

컴퓨터과학기초

14주차

순서논리회로

인하공업전문대학 컴퓨터정보과

이수정 교수



지난 시간

Ch.8 플립플롭

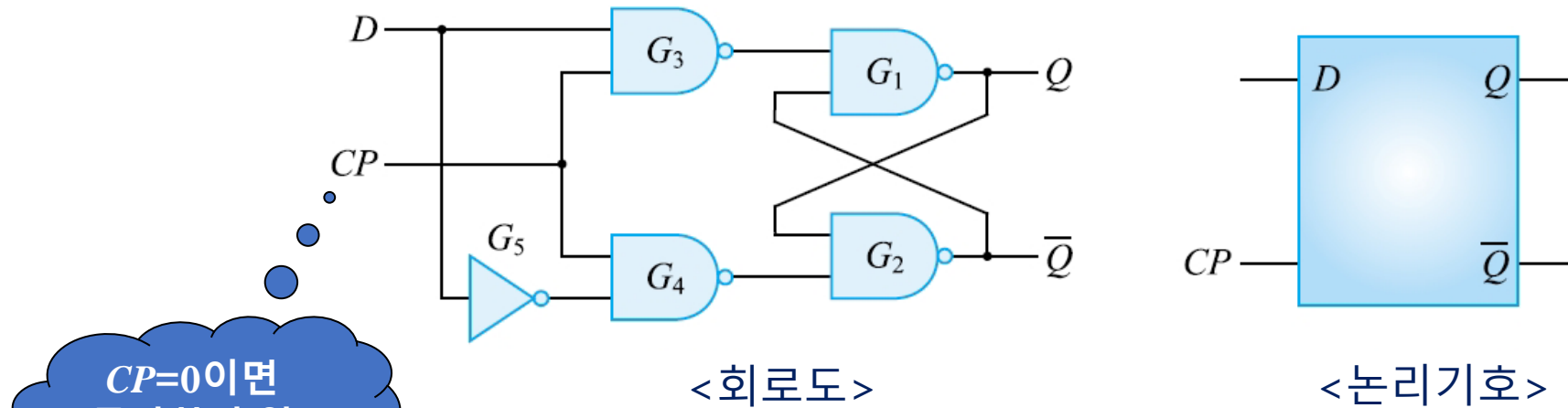
1. 기본적인 플립플롭
2. SR 플립플롭
3. D 플립플롭
4. JK 플립플롭
5. T 플립플롭
6. 비동기 입력
7. 플립플롭의 동작 특성
8. 멀티바이브레이터

3. D 플립플롭

1) 클록형 D 플립플롭

- 클록형 SR 플립플롭에서 원하지 않는 상태($S=R=1$)를 제거하는 한 가지 방법
- 클록형 D 플립플롭(Clocked D Flip-Flop)은 클록형 SR 플립플롭을 변형한 것
- 입력신호 D가 CP에 동기되어 그대로 출력에 전달되는 특성을 가지고 있음
- D 플립플롭이라는 이름은 데이터(Data)를 전달하는 것과 지연(Delay)하는 역할에서 유래

3. D 플립플롭



CP=0이면
동작하지 않
음

$CP=1, D=1$

G_3 의 출력은 0, G_4 의 출력은 1이 된다. 따라서 SR 래치의 입력은 $S=0, R=1$ 이 되므로 결과적으로 $Q=1$ 을 얻는다.

$CP=1, D=0$

G_3 의 출력은 1, G_4 의 출력은 0이 된다. 따라서 SR 래치의 입력은 $S=1, R=0$ 이 되므로 결과적으로 $Q=0$ 을 얻는다.

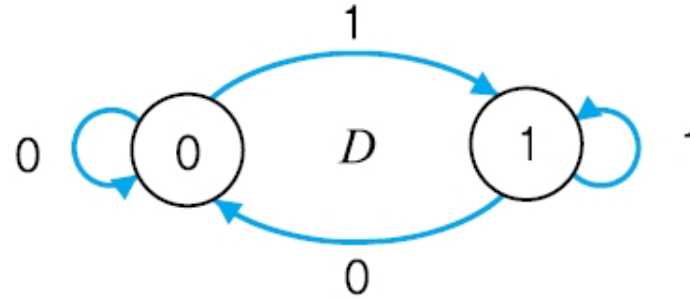
3. D 플립플롭

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

<D 플립플롭의 진리표>

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

<D 플립플롭의 특성표>



<D 플립플롭의 상태도>

$D \backslash Q(t)$	0	1
0		1
1		1

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

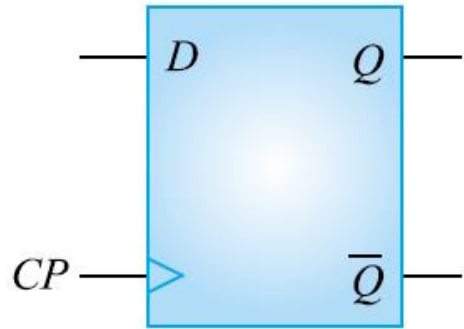
특성 방정식
(characteristic equation)

* $Q(t)$: 현재 상태
* $Q(t+1)$: 다음 상태

3. D 플립플롭

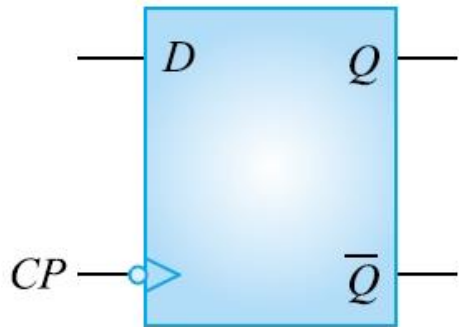
2) 에지 트리거 D 플립플롭

- 클록형 D 플립플롭의 클록펄스 입력에 펄스전이검출기를 추가하여 구성



CP	D	Q(t+1)
↑	0	0
↑	1	1

<상승에지 트리거 D 플립플롭의 논리기호 및 진리표>



CP	D	Q(t+1)
↓	0	0
↓	1	1

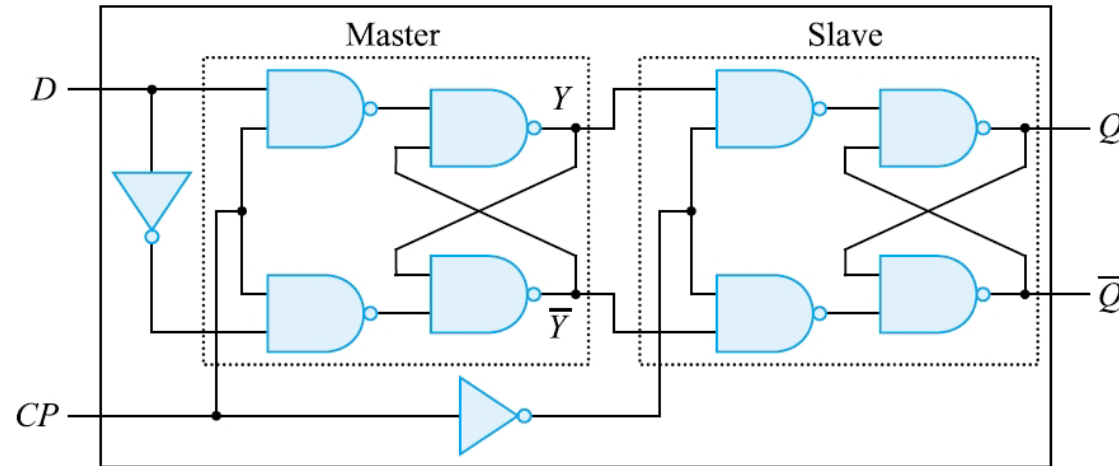
<하강에지 트리거 D 플립플롭의 논리기호 및 진리표>

100100010000011000010011
001100001100001000010001
001100001100001000010001

3. D 플립플롭

3) 주종형 D 플립플롭

- Master 플립플롭의 클럭입력은 클럭펄스가 그대로 입력되고, Slave 플립플롭 부분의 클럭입력에는 반전된 클럭펄스가 입력되도록 구성



$CP=1$	외부의 D 입력이 Master 플립플롭에 전달 Slave 플립플롭은 $CP=0$ 이므로 동작하지 않음.
$CP=0$	Slave 플립플롭이 동작하여 $Q = Y, \bar{Q} = \bar{Y}$ Master 플립플롭은 $CP=0$ 이므로 동작하지 않음.

4. JK 플립플롭

1) 클록형 JK 플립플롭

- JK 플립플롭은 SR 플립플롭에서 $S=1, R=1$ 인 경우 출력이 불안정한 상태가 되는 문제점을 개선하여 $S=1, R=1$ 에서도 동작하도록 개선한 회로
- JK 플립플롭의 J는 S(set)에, K는 R(reset)에 대응하는 입력
- $J=1, K=1$ 인 경우 JK 플립플롭의 출력은 이전 출력의 보수 상태로 변화
- JK 플립플롭은 플립플롭 중에서 가장 많이 사용되는 플립플롭이다.

- JK 플립플롭에서 J와 K의 어원에 대한 정확한 근거는 없으나, 미국의 물리학자 잭 킬비(Jack S. Kilby, 1923~2005)의 이름 이니셜이라는 설이 있다. Texas Instruments사의 엔지니어였던 잭 킬비는 1958년 집적회로를 발명했고, 2000년에 노벨 물리학상을 수상했다.
- 또 다른 설은 가장 흔한 미국 남녀 이름인 John과 Kate에서 따온 말이라고도 하지만 정확한 것은 알려져 있지 않다.

4. JK 플립플롭

CP	J	K	$Q(t+1)$
1	0	0	$Q(t)$ (불변)
1	0	1	0
1	1	0	1
1	1	1	$\overline{Q(t)}$ (toggle)

<JK 플립플롭의 진리표>

특성 방정식
(characteristic equation)

- * $Q(t)$: 현재 상태
- * $Q(t+1)$: 다음 상태

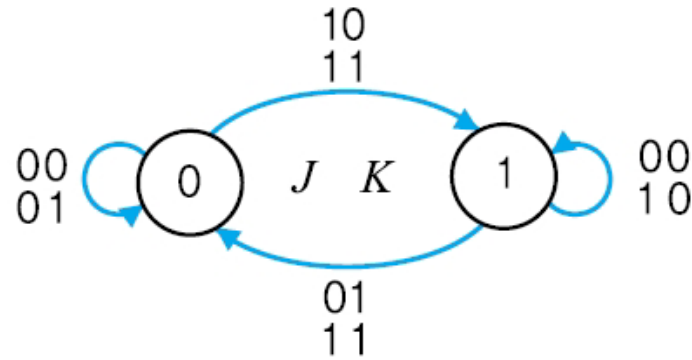
$Q(t)$	J	K	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

<JK 플립플롭의 특성표>

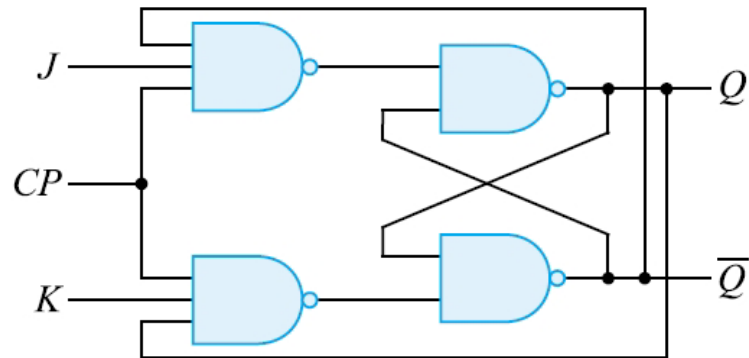
		JK			
		00	01	11	10
$Q(t)$	0			1	1
	1	1			1

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

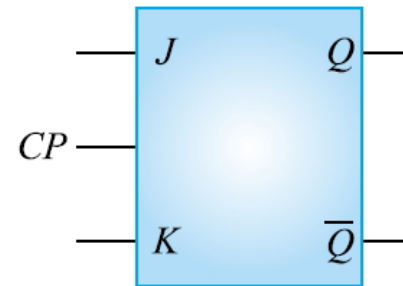
4. JK 플립플롭



<JK 플립플롭의 상태도>



<회로도>



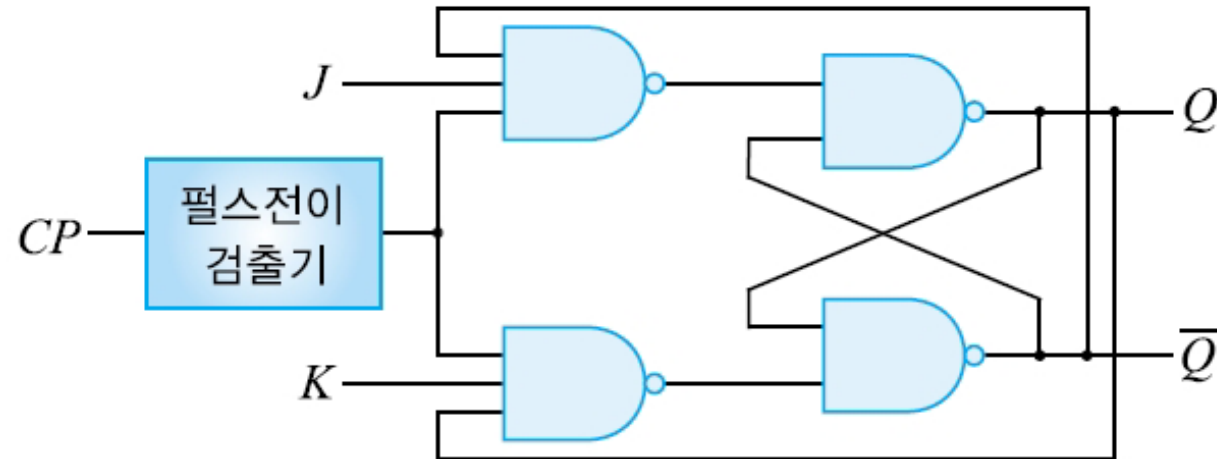
<논리기호>

클록형 JK 플립플롭(NAND 게이트형)

4. JK 플립플롭

2) 에지 트리거 JK 플립플롭

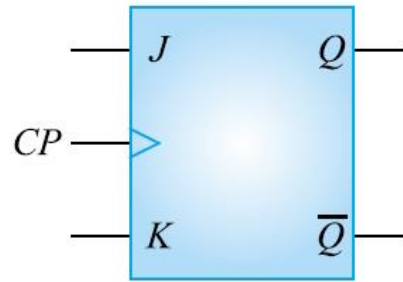
- 클록형 JK 플립플롭의 클록펄스 입력에 펄스전이 검출기를 추가하여 구성



<에지 트리거 JK 플립플롭의 구조>

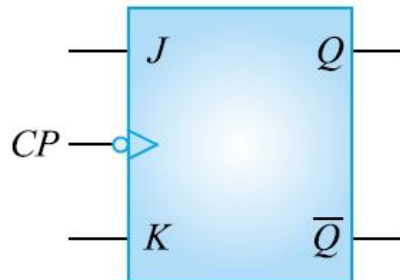
4. JK 플립플롭

■ 에지 트리거 JK 플립플롭의 논리기호와 진리표



CP	J K	$Q(t+1)$
↑	0 0	$Q(t)$ (불변)
↑	0 1	0
↑	1 0	1
↑	1 1	$\bar{Q}(t)$ (toggle)

<상승 에지 트리거 JK 플립플롭의 논리기호 및 진리표>



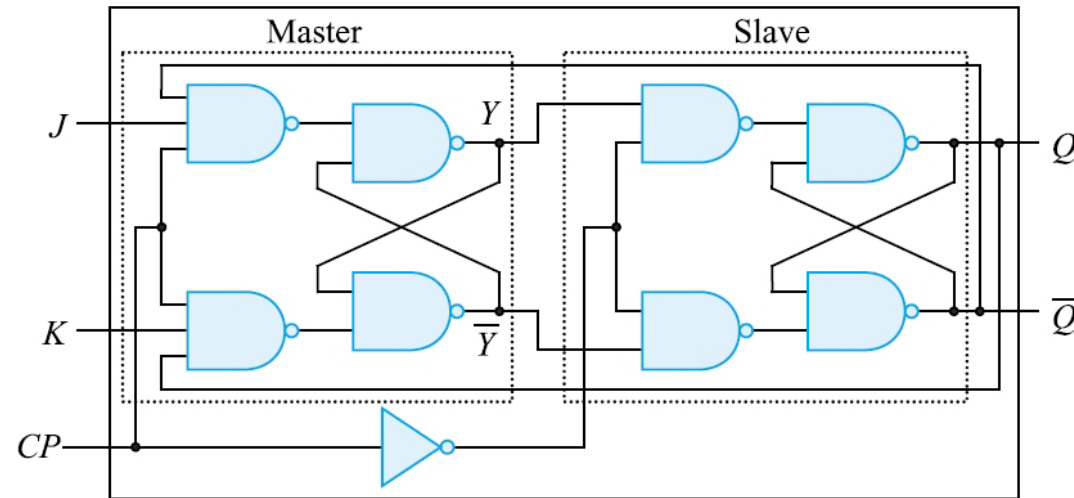
CP	J K	$Q(t+1)$
↓	0 0	$Q(t)$ (불변)
↓	0 1	0
↓	1 0	1
↓	1 1	$\bar{Q}(t)$ (toggle)

<하강 에지 트리거 JK 플립플롭의 논리기호 및 진리표>

4. JK 플립플롭

3) 주종형 JK 플립플롭

- Master 플립플롭의 클럭입력은 클럭펄스가 그대로 입력되고, Slave 플립플롭 부분의 클럭입력에는 반전된 클럭펄스가 입력되도록 구성



$CP=1$

외부의 J 와 K 의 입력이 **Master** 플립플롭에 전달
Slave 플립플롭은 $CP=0$ 이므로 동작하지 않음.

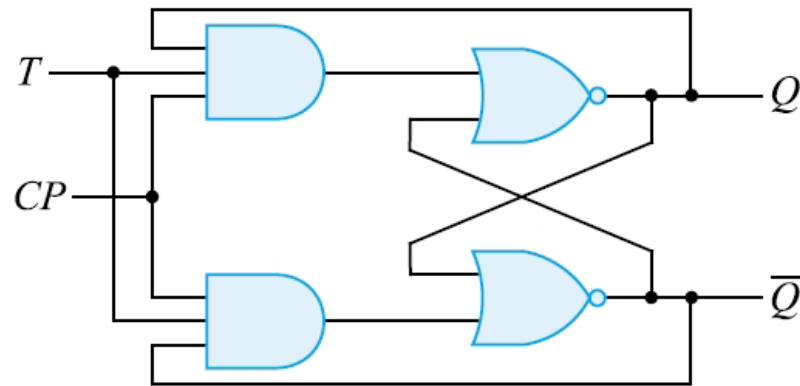
$CP=0$

Slave 플립플롭이 동작하여 $Q = Y, \bar{Q} = \bar{Y}$
Master 플립플롭은 $CP=0$ 이므로 동작하지 않음.

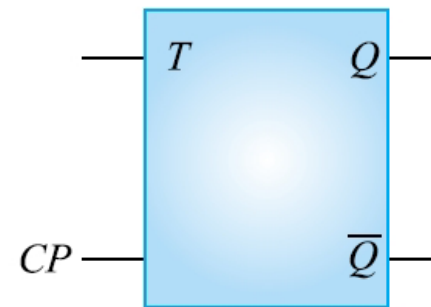
5. T 플립플롭

1) 클록형 T 플립플롭

- JK 플립플롭의 J와 K 입력을 묶어서 하나의 입력신호 T로 동작시키는 플립플롭
- JK 플립플롭의 동작에서 입력이 모두 0이거나 1인 경우만을 이용하는 플립플롭
- T 플립플롭의 입력 $T=0$ 이면, T 플립플롭은 $J=0, K=0$ 인 JK 플립플롭과 같이 동작하므로 출력은 변하지 않는다. $T=1$ 이면, $J=1, K=1$ 인 JK 플립플롭과 같이 동작하므로 출력은 보수가 된다.



<회로도>

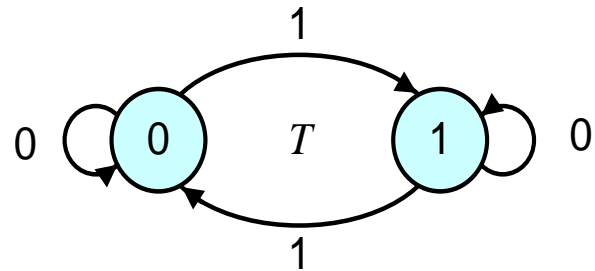


<논리기호>

5. T 플립플롭

CP	T	$Q(t+1)$
1	0	$Q(t)$
1	1	$\bar{Q}(t)$

<T 플립플롭의 진리표>



<T 플립플롭의 상태도>

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

<T 플립플롭의 특성표>

$Q(t) \backslash T$	0	1
0		1
1	1	

- * $Q(t)$: 현재 상태
- * $Q(t+1)$: 다음 상태

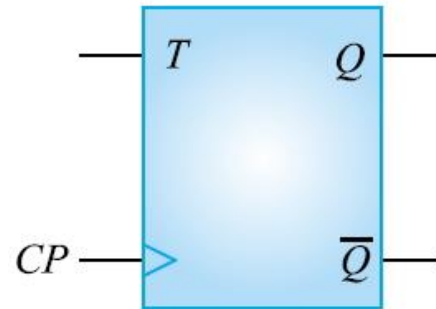
특성 방정식
(characteristic equation)

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

5. T 플립플롭

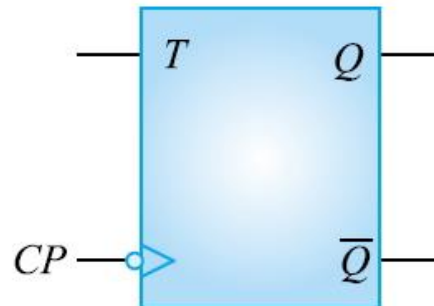
2) 에지 트리거 T 플립플롭

- 클록형 T 플립플롭의 클록펄스 입력에 펄스 전이 검출기를 추가하여 구성



CP	T	$Q(t+1)$
↑	0	$Q(t)$
↑	1	$\overline{Q}(t)$

<상승에지 트리거 T 플립플롭>



CP	T	$Q(t+1)$
↓	0	$Q(t)$
↓	1	$\overline{Q}(t)$

<하강에지 트리거 T 플립플롭>

8. 멀티바이브레이터

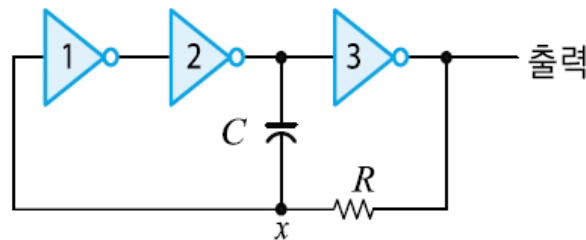
- 멀티바이브레이터(multivibrator)는 디지털 시스템에서 매우 중요하게 사용되는 것 중의 하나
- 기본적으로 두 개의 인버터(inverter)로 구성되어 있고 각각의 출력을 궤환(feedback)시켜서 서로 상대 인버터를 입력으로 한다.
- 이와 같은 형태의 인버터는 한쪽 인버터의 출력이 0이면 다른 한쪽 인버터의 출력은 반드시 1이어서 동시에 같은 상태에 있을 수는 없다.
- 멀티바이브레이터는 디지털 시스템에서 2진수를 저장하고, 펄스 수를 세며, 연산을 동기화하고 그 외 여러 가지 중요한 기능을 수행
- 회로 구성에 따른 멀티바이브레이터의 종류
 - 무안정 멀티바이브레이터(astable multivibrator)
 - 단안정 멀티바이브레이터(monostable multivibrator)
 - 쌍안정 멀티바이브레이터(bistable multivibrator, 플립플롭과 같음)

8. 멀티바이브레이터

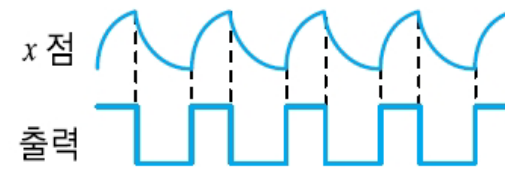
1) 무안정 멀티바이브레이터

- 무안정(또는 비안정, 불안정) 멀티바이브레이터는 불안정한 두 가지 상태 High 또는 Low 상태를 가지며, 한 쪽 상태에 머무르지 못하고 두 상태를 왔다 갔다 하는 것으로서 일종의 발진기(oscillator)다.
- 외부 입력 없이 스스로 주기적인 구형파를 발생시킨다.

■ NOT 게이트를 이용한 무안정 멀티바이브레이터 회로



<회로도>



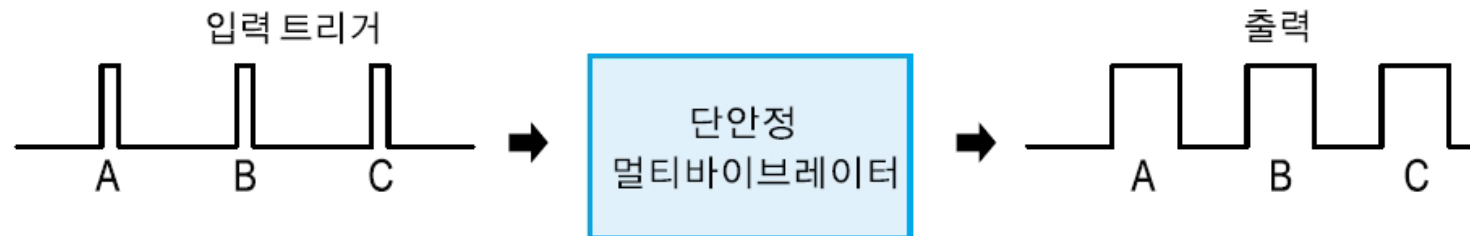
<출력파형>

$$\text{발진 주파수 : } f = \frac{0.455}{RC}$$

8. 멀티바이브레이터

2) 단안정 멀티바이브레이터

- 단안정(one-shot) 멀티바이브레이터는 입력에 트리거 신호(짧은 펄스)가 가해질 때마다 일정한 폭을 갖는 하나의 구형 펄스를 발생시키는 회로
- 트리거 신호에 의하여 일단 준 안정상태(quasi-stable)를 유지하다가 곧 안정된 상태로 복귀
- 단안정 멀티바이브레이터의 종류
 - retriggerable 단안정 회로(74122, 74123)
 - non-retriggerable 단안정 회로(74121, 74221)



<단안정 멀티바이브레이터 동작 개념도>

차례

Ch.9 동기 순서논리회로

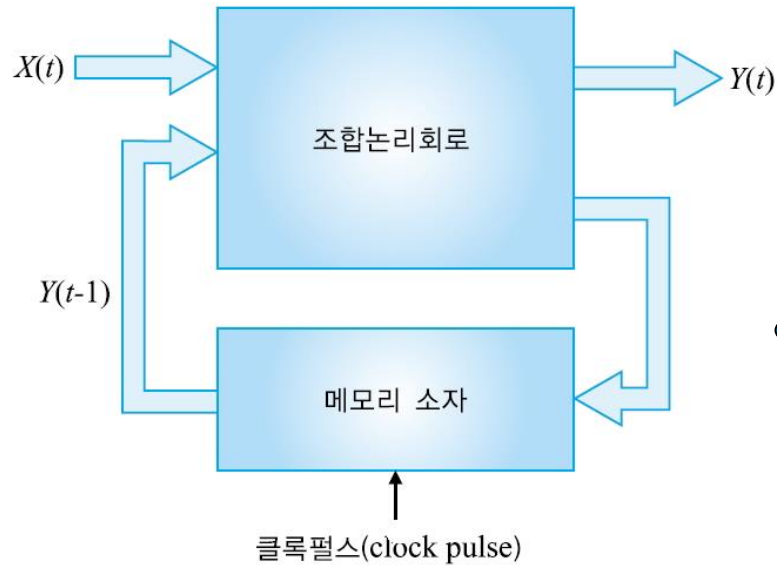
1. 동기 순서논리회로 개요
2. 동기 순서논리회로의 해석 과정
3. 플립플롭의 여기표
4. 동기 순서논리회로의 설계 과정
5. 동기 순서논리회로의 설계 예

조합논리회로와 순서논리회로

조합논리회로 (combinational logic circuit)	<ul style="list-style-type: none">• 출력이 현재의 입력에 의해서만 결정되는 논리회로
순서논리회로 (sequential logic circuit)	<ul style="list-style-type: none">• 현재의 입력과 이전의 출력상태에 의해서 출력이 결정되는 논리회로• 신호의 타이밍(timing)에 따라 동기 순서논리회로와 비동기 순서논리회로로 분류• 동기 순서회로는 클록펄스가 들어오는 시점에서 상태가 변화하는 회로• 클록펄스에 의해서 동작하는 회로를 동기순서논리회로 또는 단순히 동기순서회로라 한다.• 비동기 순서회로는 시간에 관계없이 단지 입력이 변화하는 순서에 따라 동작하는 논리회로

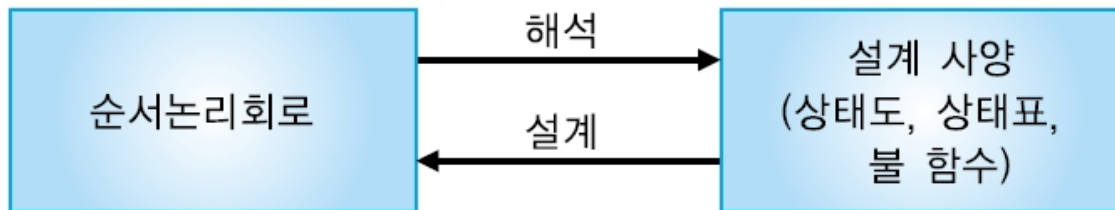
1. 동기 순서논리회로 개요

■ 순서논리회로의 블록도



출력 $Y(t)$ 는 현재 상태의 입력 $X(t)$ 와 이전 상태의 출력 $Y(t-1)$ 에 의하여 결정

■ 순서논리회로의 해석과 설계 관계



2. 동기 순서논리회로의 해석 과정

- 순서논리회로의 동작은 입력과 출력 및 플립플롭의 현재 상태에 의해 결정
- 출력과 다음 상태는 현재 상태의 함수가 된다.
- 순서논리회로의 해석은 입력과 출력 및 현재 상태에 의해 결정되는 다음 상태의 시간순서를 상태표나 상태도로 나타냄으로써 해석이 가능

■ 순서논리회로의 해석 과정

[단계 1] 회로 입력과 출력에 대한 변수 명칭 부여

[단계 2] 조합논리회로가 있으면 조합논리회로의 부울대수식 유도

[단계 3] 회로의 상태표 작성

[단계 4] 상태표를 이용하여 상태도 작성

[단계 5] 상태방정식 유도

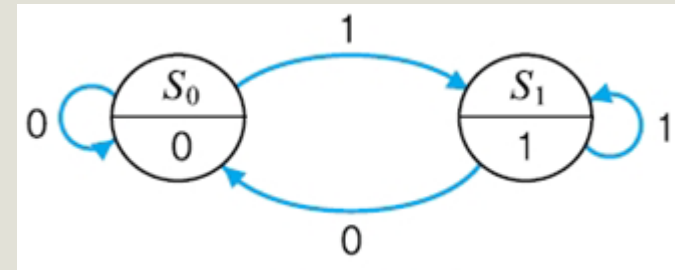
[단계 6] 상태표와 상태도를 분석하여 회로의 동작 설명

2. 동기 순서논리회로의 해석 과정

■ 출력값이 결정되는 방법에 따른 분류

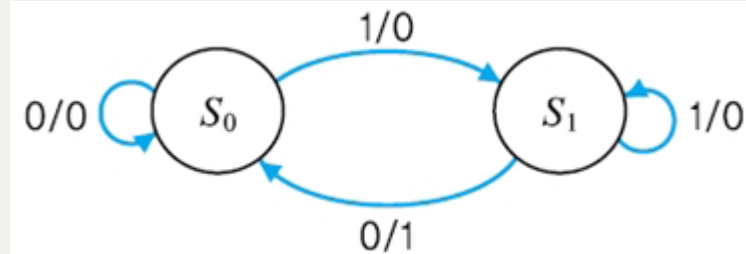
무어머신
(Moore
machine)

- 순서논리회로의 출력이 플립플롭들의 현재 상태만의 함수인 회로
- 출력이 상태 내에 결합되어 표시



밀리머신
(Mealy
machine)

- 출력이 현재 상태와 입력의 함수인 회로
- 출력은 상태 사이를 지나가는 화살선의 위에 표시



2. 동기 순서논리회로의 해석 과정

1) 변수 명칭 부여

- 입력변수 : x
- 출력 변수 : y
- FF-A 플립플롭의 입력 : J_A, K_A
- FF-B 플립플롭의 입력 : J_B, K_B
- FF-A 플립플롭의 출력 : $A (= Q_A)$
- FF-B 플립플롭의 출력 : $B (= Q_B)$

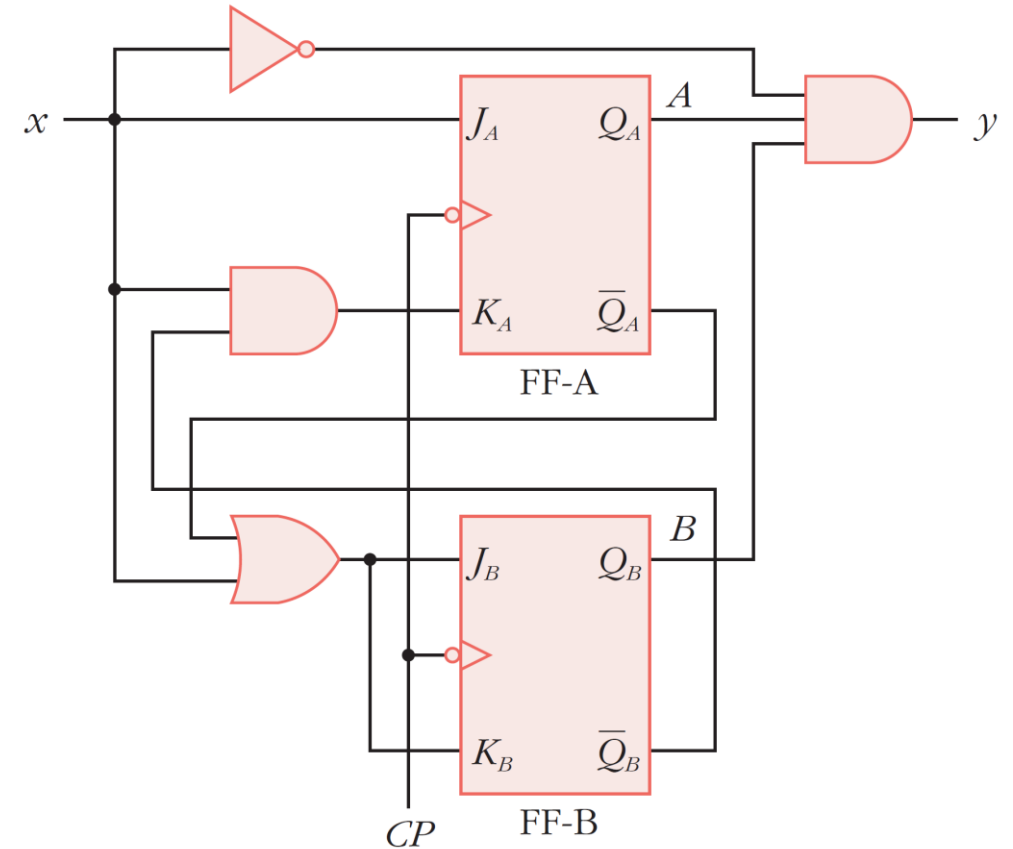
2) 불 대수식 유도

- F-F A 플립플롭의 입력
- F-F B 플립플롭의 입력
- 시스템 출력

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

$$J_B = \bar{A} + x, \quad K_B = \bar{A} + x$$

$$y = AB\bar{x}$$



2. 동기 순서논리회로의 해석 과정

3) 상태표 작성

- 상태표(state table) : 현재 상태와 외부 입력의 변화에 따라 다음 상태와 출력의 변화를 정의한 것
- 현재 상태 : 클록펄스(*CP*) 인가 전 상태
- 다음 상태 : 클록펄스의 인가 후 상태

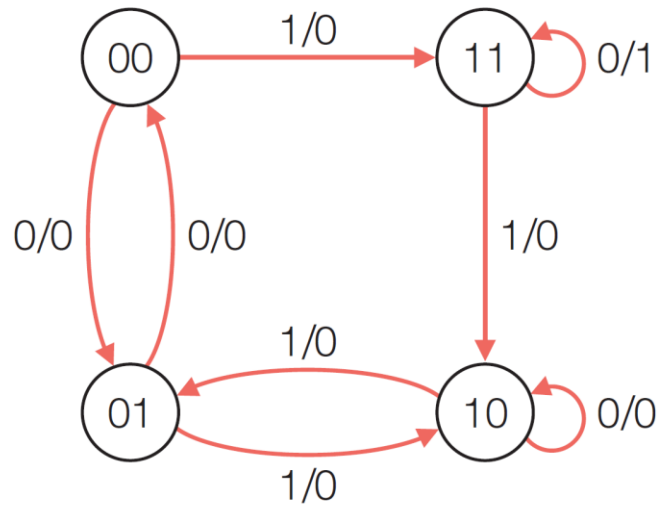
현재 상태		입력	다음 상태		출력
A	B	x	A	B	y
0	0	0	0	1	0
0	0	1	1	1	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	0	1	0
1	1	0	1	1	1
1	1	1	1	0	0

<상태표>

2. 동기 순서논리회로의 해석 과정

4) 상태도 작성

- 상태표로부터 상태도를 그린다.



<상태도>

x/y ← 출력
↑ 입력

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

<상태표>

2. 동기 순서논리회로의 해석 과정

5) 상태방정식 유도

- 상태방정식(state equation): 플립플롭 상태 전이에 대한 조건을 지정하는 대수식
- 상태표로부터 플립플롭 A 와 B 가 논리 1이 되는 상태방정식을 구한다.

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

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

- 카르노 맵을 이용하여 간소화한 상태방정식

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

$$A(t+1) = \overline{A}x + AB + A\overline{x}$$

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

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

2. 동기 순서논리회로의 해석 과정

- JK 플립플롭의 특성방정식과 비교

$$\begin{aligned} A(t+1) &= \bar{A}x + AB + A\bar{x} \\ &= x\bar{A} + (B + \bar{x})A = x\bar{A} + (\overline{\bar{B}x})A \end{aligned}$$



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

$$\begin{aligned} B(t+1) &= \bar{A}\bar{B} + \bar{B}x + AB\bar{x} \\ &= (\bar{A} + x)\bar{B} + A\bar{x}B = (\bar{A} + x)\bar{B} + (\overline{\bar{A} + x})B \end{aligned}$$



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

$$\begin{aligned} J_A &= x & K_A &= \bar{B}x \\ J_B &= \bar{A} + x & K_B &= \bar{A} + x \end{aligned}$$

[그림 9-4] 회로와 일치

6) 회로의 동작 설명

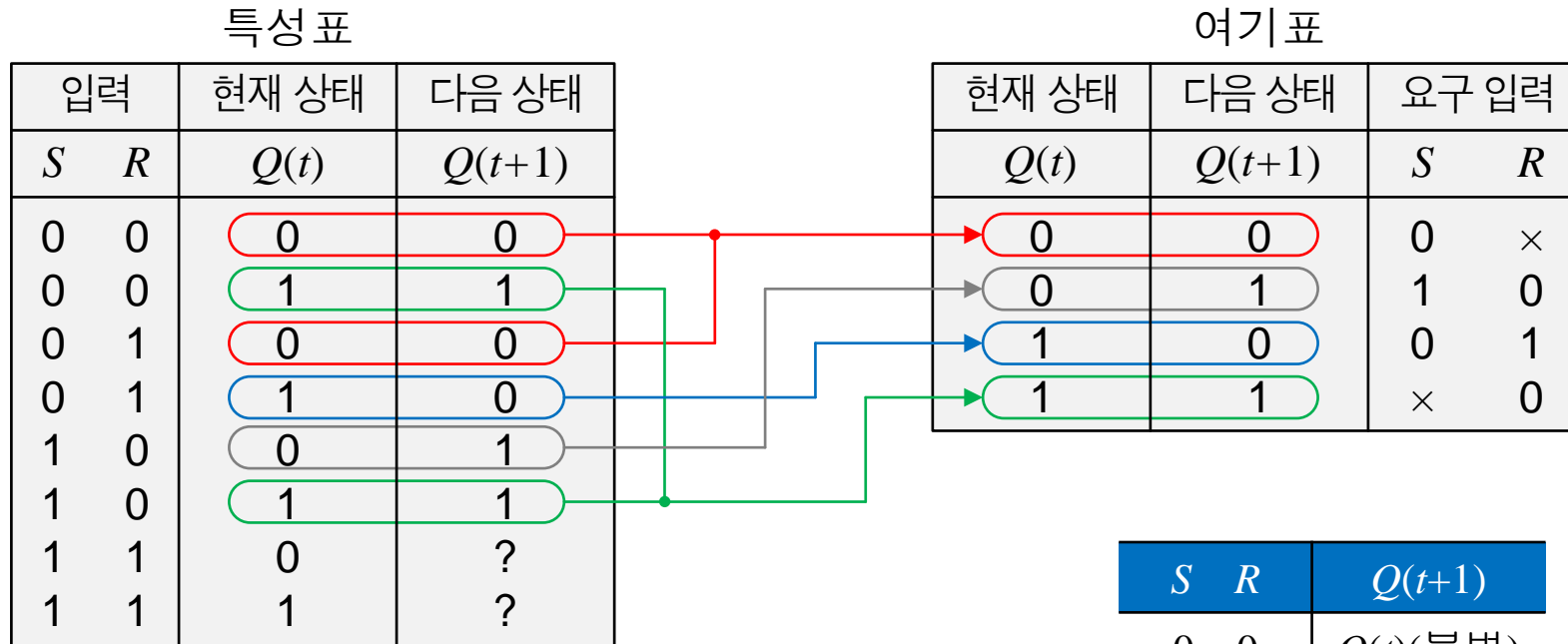
- 순서논리회로의 동작은 상태도나 상태표를 이용하여 설명 가능
- 입력 x 의 값에 따라 클럭펄스가 한번씩 인가될 때마다 순차적으로 동작하는 순서논리회로

3. 플립플롭의 여기표

- 플립플롭의 **특성표** : 현재 상태와 입력값이 주어졌을 때, 다음 상태가 어떻게 변하는가를 나타내는 표
- 플립플롭의 **여기표**(excitation table) : 현재 상태에서 다음 상태로 변했을 때, 플립플롭의 입력 조건이 어떤 상태인가를 나타내는 표
- 플립플롭의 여기표는 순서논리회로를 설계할 때 자주 사용

3. 플립플롭의 여기표

1) SR 플립플롭의 여기표

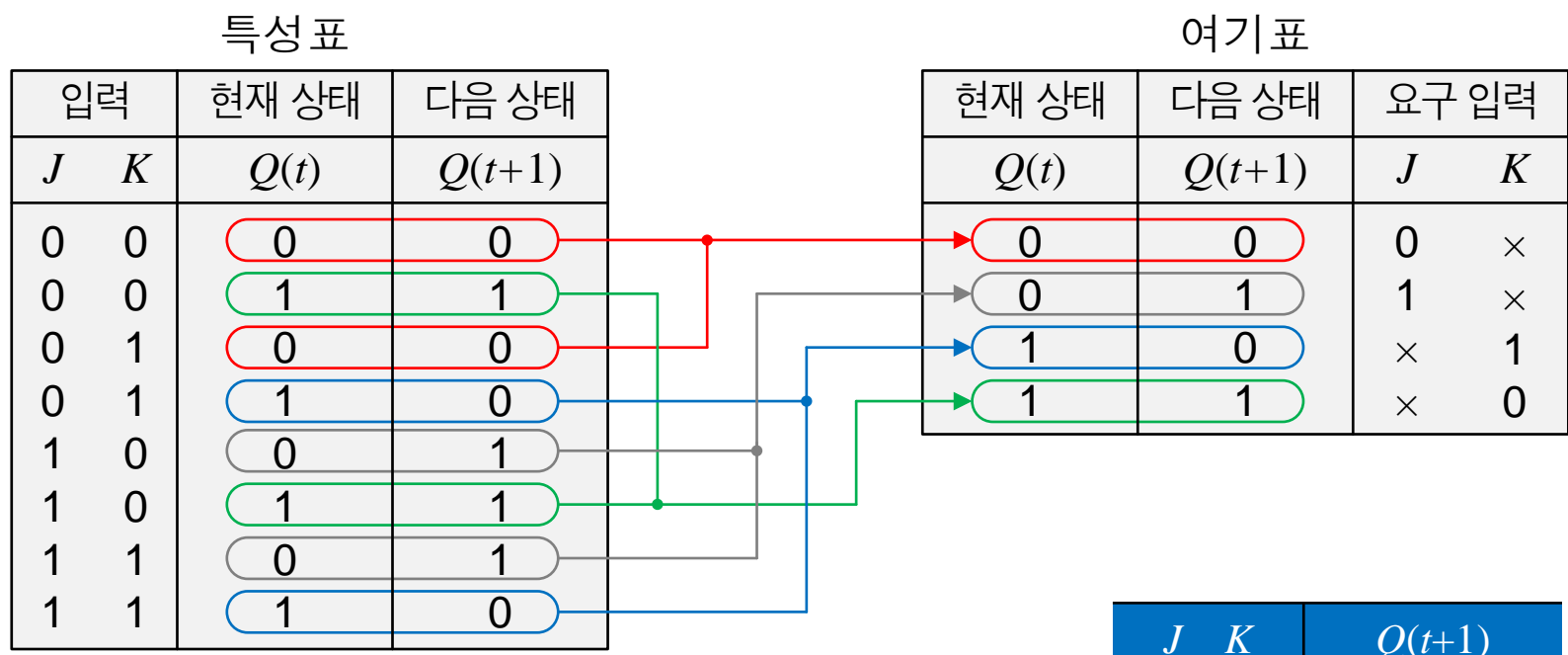


S	R	$Q(t+1)$
0	0	$Q(t)$ (불변)
0	1	0
1	0	1
1	1	(부정)

<SR 플립플롭 진리표>

3. 플립플롭의 여기표

2) JK 플립플롭의 여기표



J	K	Q(t+1)
0	0	Q(t)(불변)
0	1	0
1	0	1
1	1	$\bar{Q}(t)$ (토글)

<JK 플립플롭 진리표>

3. 플립플롭의 여기표

3) D 플립플롭의 여기표

특성 표			여기 표		
입력	현재 상태	다음 상태	현재 상태	다음 상태	요구 입력
D	$Q(t)$	$Q(t+1)$	$Q(t)$	$Q(t+1)$	D
0	0	0	0	0	0
0	1	0	0	1	1
1	0	1	1	0	0
1	1	1	1	1	1

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

D 플립플롭 진리표

4) T 플립플롭의 여기표

특성 표			여기 표		
입력	현재 상태	다음 상태	현재 상태	다음 상태	요구 입력
T	$Q(t)$	$Q(t+1)$	$Q(t)$	$Q(t+1)$	T
0	0	0	0	0	0
0	1	1	0	1	1
1	0	1	1	0	1
1	1	0	1	1	0

T	$Q(t+1)$
0	$Q(t)$ (불변)
1	$\bar{Q}(t)$ (토글)

T 플립플롭 진리표

4. 동기 순서논리회로의 설계 과정

■ 순서논리회로의 설계 과정

단계 1

회로 동작 기술(상태도 작성)

단계 2

정의된 회로의 상태표 작성

단계 3

필요한 경우 상태 축소 및 상태 할당

단계 4

플립플롭의 수와 플립플롭의 종류 결정

단계 5

플립플롭의 입력, 출력 및 각각의 상태에 문자기호 부여

단계 6

상태표를 이용하여 회로의 상태 여기표 작성

단계 7

간소화 방법을 이용하여 출력 함수 및 플립플롭의 입력 함수 유도

단계 8

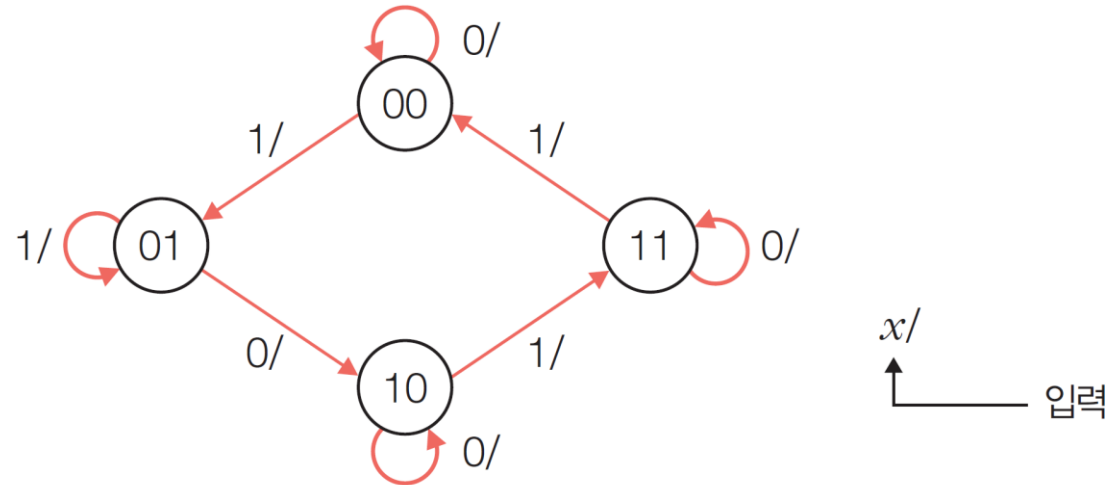
순서논리회로도 작성

4. 동기 순서논리회로의 설계 과정

■ JK 플립플롭을 이용한 순서논리회로 설계

1) 회로 동작 기술

- 입력변수만 있고 출력변수는 없는 상태에서 상태변화가 일어난다.



동기 순서논리회로에 대한 상태도(JK 플립플롭을 이용하는 경우)

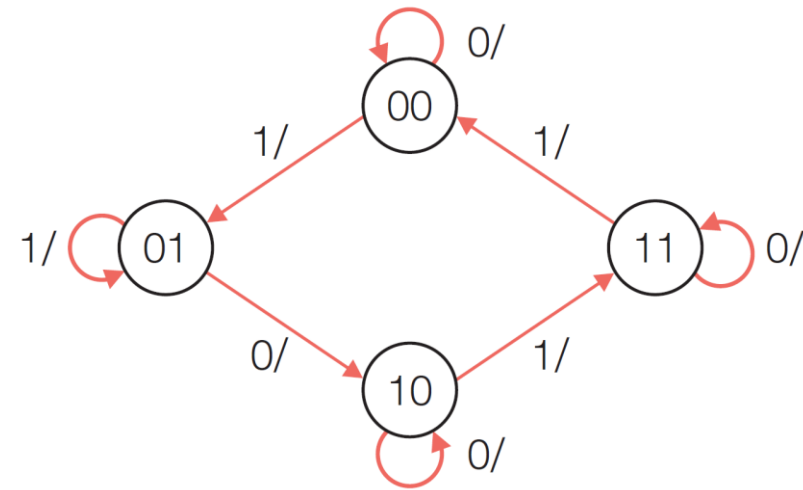
4. 동기 순서논리회로의 설계 과정

2) 상태표 작성

- 상태도로부터 상태표 유도

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

상태표



상태도

4. 동기 순서논리회로의 설계 과정

3) 플립플롭의 수와 형태 결정

■ 플립플롭의 수

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

$$n=16\text{이면, } \lceil \log_2 16 \rceil = 4 \log_2 2 = 4$$

$$n=4\text{이면, } \lceil \log_2 4 \rceil = 2 \log_2 2 = 2$$

$$n=5\text{이면, } \lceil \log_2 5 \rceil = \lceil 2.3219 \rceil = 3$$

- 상태의 수가 4가지인 경우에는 2개의 플립플롭이 필요

■ 플립플롭의 형태

- 설계할 회로 특성에 알맞고 구현이 용이한 플립플롭을 선택해야 함
- 카운터를 설계할 경우 : 회로의 특성상 주로 JK 플립플롭이나 T 플립플롭을 이용하는 것이 유리

4. 동기 순서논리회로의 설계 과정

4) 상태 여기표 유도

현재 상태		입력	다음 상태		플립플롭 입력			
A	B	x	A	B	J _A	K _A	J _B	K _B
0	0	0	0	0	0	×	0	×
0	0	1	0	1	0	×	1	×
0	1	0	1	0	1	×	×	1
0	1	1	0	1	0	×	×	0
1	0	0	1	0	×	0	0	×
1	0	1	1	1	×	0	1	×
1	1	0	1	1	×	0	×	0
1	1	1	0	0	×	1	×	1

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

JK 플립플롭의 여기표

4. 동기 순서논리회로의 설계 과정

5) 플립플롭의 입력함수 및 회로의 출력함수 유도

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

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

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

$$K_A = Bx$$

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

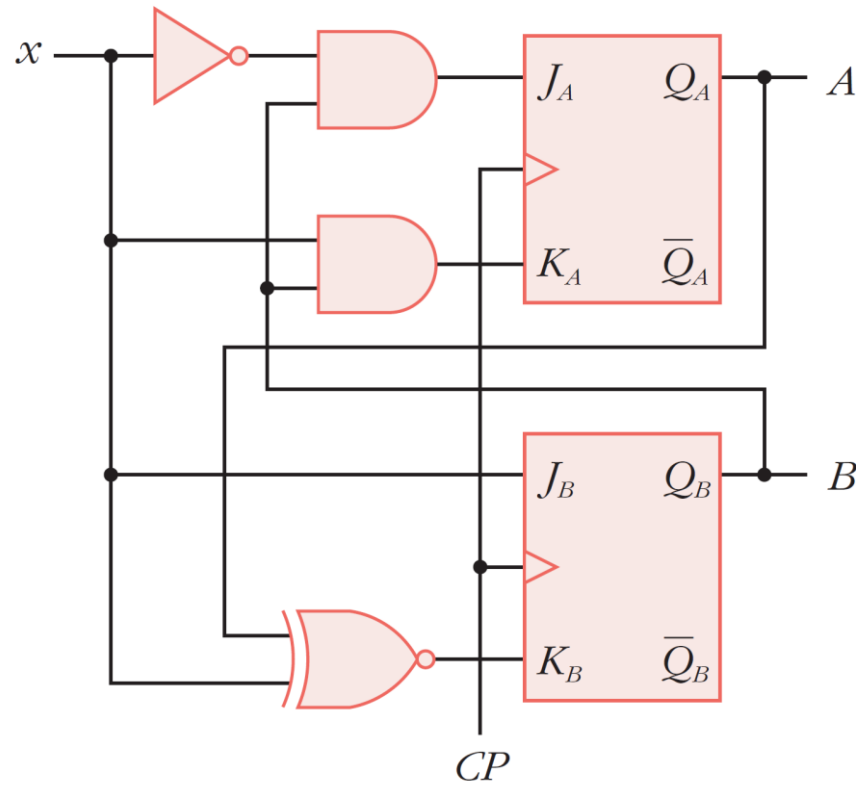
$$J_B = x$$

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

$$K_B = Ax + \overline{A}x = \overline{A} \oplus x = A \odot x$$

4. 동기 순서논리회로의 설계 과정

6) 논리회로의 구현



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

$$K_A = Bx$$

$$J_B = x$$

$$K_B = A \odot x$$

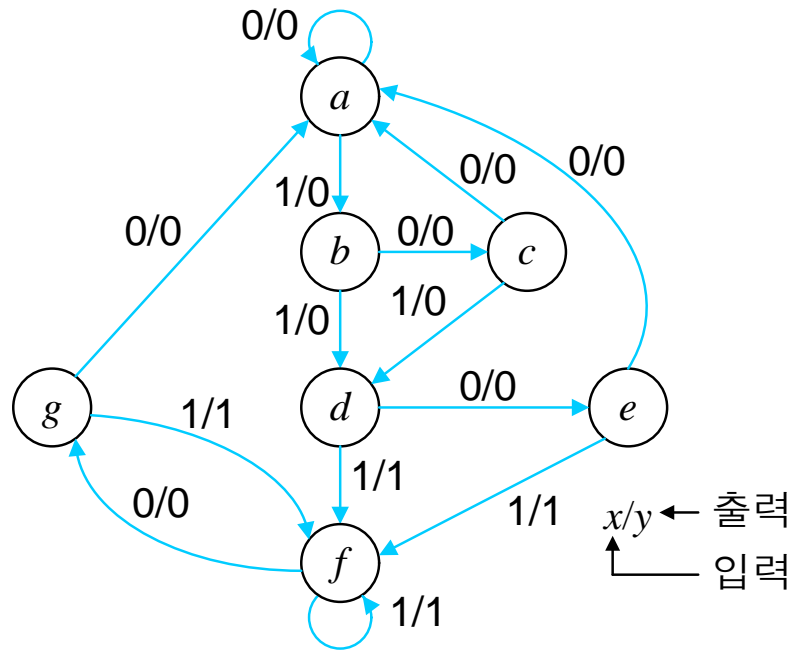
5. 동기 순서논리회로의 설계 예

- 문자 기호로 표시된 상태를 가진 상태도로부터 간소화된 상태표를 유도하기 위한 절차에 대해서 알아보기로 한다.
- 상태도로부터 얻은 상태표는 불필요한 상태(redundant state)를 가질 수 있다.
- 축소된 최소 상태표(minimal state table)를 유도하기 위한 과정은 상태 축소와 상태 할당의 2단계에 의해서 수행된다.

■ 상태 축소

- 순서논리회로에서 플립플롭의 수를 줄이는 것
- 플립플롭의 수가 m 이라 가정하면, 이때 요구되는 상태는 2^m 이 되므로 상태의 수를 줄임으로써 플립플롭의 수를 줄일 수 있다.
- 경우에 따라 상태의 수는 감소되지만 플립플롭의 수는 변화하지 않는 경우도 있다.

5. 동기 순서논리회로의 설계 예



<상태도>

현재 상태	다음 상태		출력	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

<상태표>

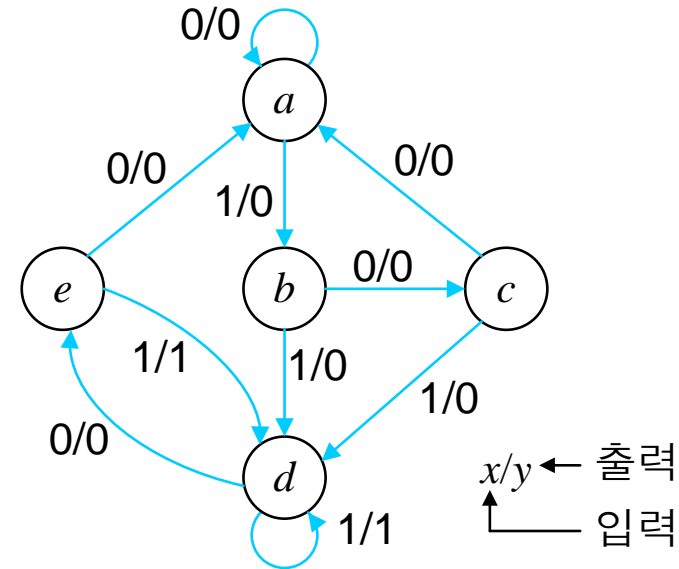
<상태 축소를 설명하기 위한 상태도>

5. 동기 순서논리회로의 설계 예

현재 상태	다음 상태		출력	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	fd	0	1
e	a	fd	0	1
f	ge	f	0	1
g	a	f	0	1

현재 상태	다음 상태		출력	
	$x=0$	$x=1$	$x=0$	$x=1$
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	d	0	1
e	a	d	0	1

<최종 상태표>



<축소된 상태도>

5. 동기 순서논리회로의 설계 예

■ 상태 할당

- 기호 형태로 표현된 각각의 상태에 대해서 2진수(2진 코드)의 값을 할당하는 과정

상태	할당1	할당2	할당3
<i>a</i>	0 0 0	0 0 0	0 0 0
<i>b</i>	0 0 1	0 1 0	1 0 0
<i>c</i>	0 1 0	0 1 1	0 1 0
<i>d</i>	0 1 1	1 0 1	1 0 1
<i>e</i>	1 0 0	1 1 1	0 1 1

현재 상태	다음 상태		출력	
	<i>x</i> =0	<i>x</i> =1	<i>x</i> =0	<i>x</i> =1
0 0 0	0 0 0	0 0 1	0	0
0 0 1	0 1 0	0 1 1	0	0
0 1 0	0 0 0	0 1 1	0	0
0 1 1	1 0 0	0 1 1	0	1
1 0 0	0 0 0	0 1 1	0	1

<할당 1에 의한 최소 상태표>

5. 동기 순서논리회로의 설계 예

■ 플립플롭의 수와 형태 결정

- 제어하려는 상태의 수는 5가지이므로 플립플롭 3비트가 필요

$$n=5 \text{이면 } , [\log_2 5] = [2.3219] = 3$$

- 3개의 *SR* 플립플롭을 순서대로 *A*, *B*, *C*라고 정의
- 현재 상태 *a*, *b*, *c*, *d*, *e*에 각각 000, 001, 010, 011, 100을 할당

5. 동기 순서논리회로의 설계 예

■ 상태 여기표의 유도

	현재상태	외부입력	다음상태	플립플롭의입력						외부출력
	<i>A B C</i>	<i>x</i>	<i>A B C</i>	<i>S_A</i>	<i>R_A</i>	<i>S_B</i>	<i>R_B</i>	<i>S_C</i>	<i>R_C</i>	<i>y</i>
<i>a</i>	0 0 0	0	0 0 0	0	x	0	x	0	x	0
	0 0 0	1	0 0 1	0	x	0	x	1	0	0
<i>b</i>	0 0 1	0	0 1 0	0	x	1	0	0	1	0
	0 0 1	1	0 1 1	0	x	1	0	x	0	0
<i>c</i>	0 1 0	0	0 0 0	0	x	0	1	0	x	0
	0 1 0	1	0 1 1	0	x	x	0	1	0	0
<i>d</i>	0 1 1	0	1 0 0	1	0	0	1	0	1	0
	0 1 1	1	0 1 1	0	x	x	0	x	0	1
<i>e</i>	1 0 0	0	0 0 0	0	1	0	x	0	x	0
	1 0 0	1	0 1 1	0	1	1	0	1	0	1
don't care	1 0 1	0	x x x	x	x	x	x	x	x	x
	1 0 1	1	x x x	x	x	x	x	x	x	x
	1 1 0	0	x x x	x	x	x	x	x	x	x
	1 1 0	1	x x x	x	x	x	x	x	x	x
	1 1 1	0	x x x	x	x	x	x	x	x	x
	1 1 1	1	x x x	x	x	x	x	x	x	x

<i>Q(t)</i>	<i>Q(t+1)</i>	<i>S</i>	<i>R</i>
0	0	0	x
0	1	1	0
1	0	0	1
1	1	x	0

SR 플립플롭의 여기표

5. 동기 순서논리회로의 설계 예

■ 플립플롭의 입력함수 및 회로의 출력함수 유도

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

$$S_A = BC\bar{x}$$

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

$$R_A = A$$

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

$$S_B = Ax + \bar{B}C$$

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

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

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

$$S_C = x$$

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

$$R_C = \bar{x}$$

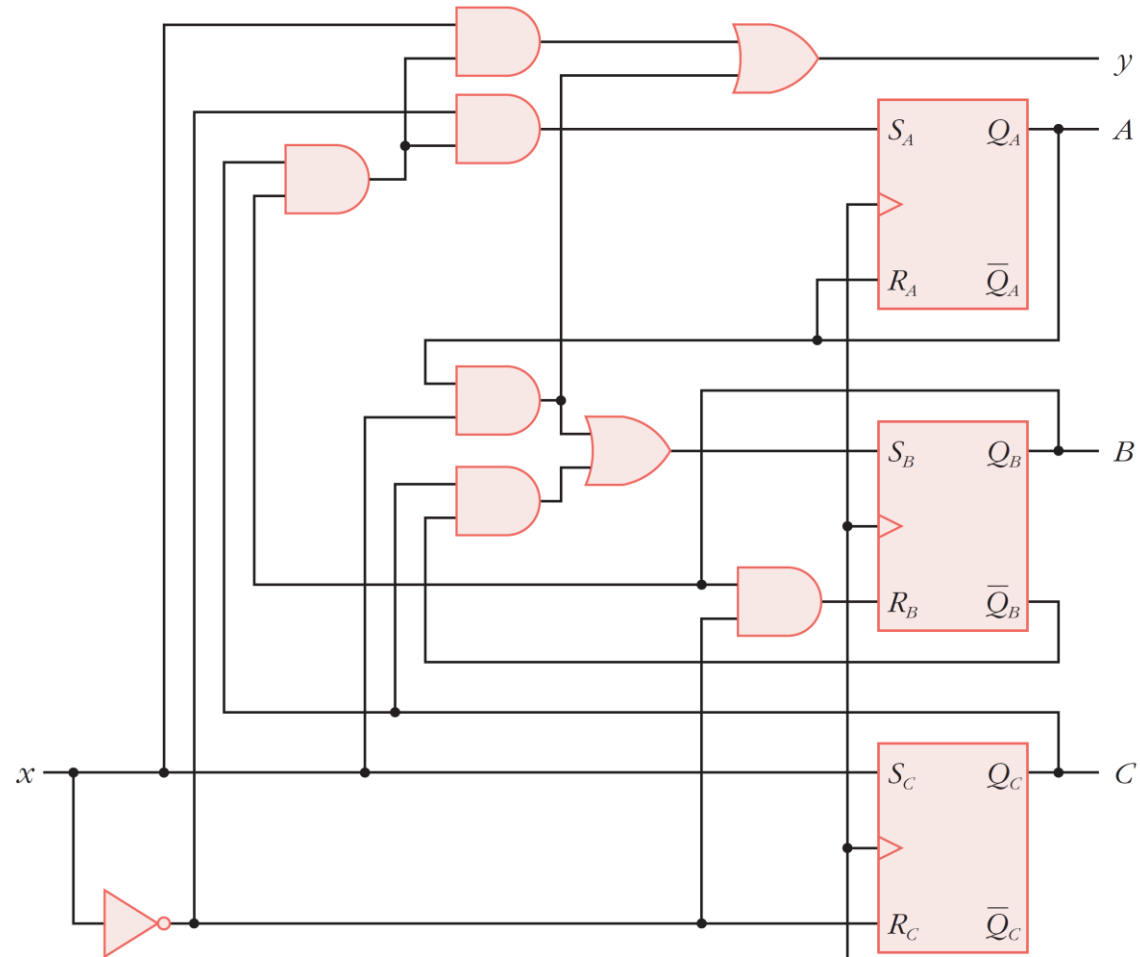
5. 동기 순서논리회로의 설계 예

■ 순서 논리회로의 구현

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

$$y = Ax + BCx$$

$$\begin{aligned} S_A &= BC\bar{x} & R_A &= A \\ S_B &= Ax + \bar{B}C & R_B &= B\bar{x} \\ S_C &= x & R_C &= \bar{x} \\ y &= Ax + BCx \end{aligned}$$



<순서 제어회로의 논리회로>

수고했습니다. 감사합니다.

