

컴퓨터공학실험II

Counter

20182186 김승원
20192124 황유진

목차

1. 카운터

- 1) 카운터란
- 2) 비동기식/동기식 카운터

2. 비동기식 카운터

- 1) 2진 비동기식 카운터
- 2) 2비트 2진 비동기식 카운터
- 3) 4비트 2진 비동기식 카운터
- 4) BCD 비동기식 카운터

3. 동기식 카운터

- 1) 4 비트 동기식 2진 카운터
- 2) n 비트 동기식 2진 카운터 설계 과정
- 3) 상향/하향 2진 카운터
- 4) BCD 카운터

4. 비동기식/동기식 카운터 비교

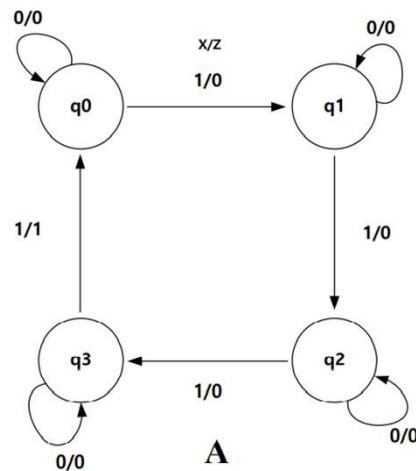
5. Finite-state machine(유한 상태 기계)

- 1) 밀리 모델
 - 2) 무어 모델
-

1. 카운터

1. 카운터란

- 1) 입력 펄스에 따라 정해진 순서대로 상태 변화가 진행되는 논리 회로를 말함
- 2) 상태의 변화 순서는 2진수 순서 또는 임의의 순서를 따름
- 3) 상태도



<Figure 1> 2 비트 2진 카운터 상태도

1. 카운터

1. 카운터란

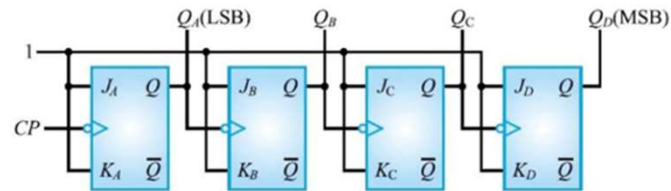
4) 2진 카운터: 2진수의 순서를 따르는 카운터

5) n 비트 2진 카운터: n 개의 플립플롭으로 구성되고,
0에서 $(2^n - 1)$ 까지 2진으로 카운트함.

1. 카운터

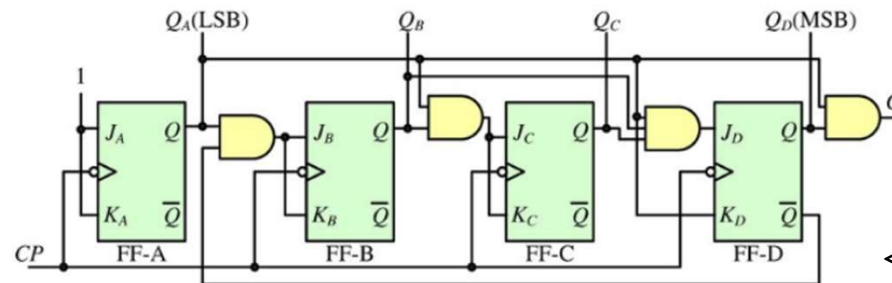
2. 비동기식 카운터/동기식 카운터

- 1) 비동기식 카운터(리플 카운터): 순차적으로 플립플롭의 출력의 변화에 의해 다음 플립플롭이 트리거되어 총 전파 지연이 크다.



<Figure 2> 4 비트 비동기식 2진 카운터

- 2) 동기식 카운터: 모든 플립플롭의 클럭 입력이 공통된 하나의 클럭 신호로 동기화되어 트리거됨

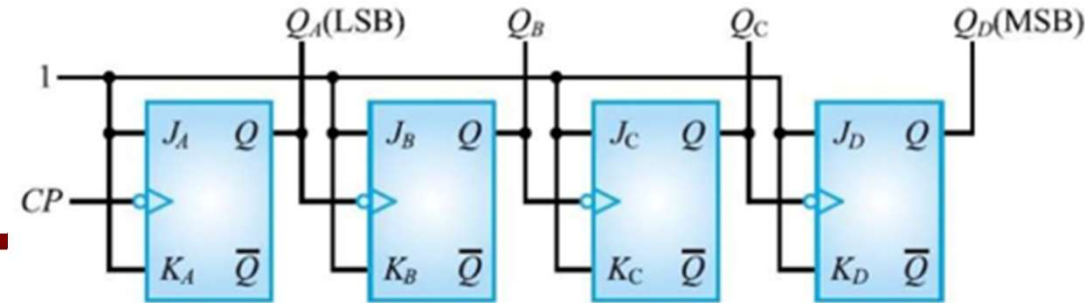


<Figure 3> BCD 동기식 카운터

2. 비동기식 카운터

1. 2진 비동기식 카운터

- 1) 보수 플립플롭들이 직렬로 연결됨
- 2) 플립플롭의 출력이 다음 상위 플립플롭의 C 입력에 연결됨
- 3) 최하위 비트를 저장하는 플립플롭은 입력 카운트 펄스와 연결됨
- 4) JK 플립플롭, T 플립플롭, 또는 D 플립플롭 사용

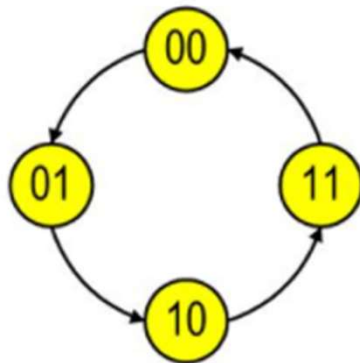


<Figure 2> 4 비트 비동기식 2진 카운터

2. 비동기식 카운터

2. 2 비트 2진 비동기식 카운터

- 1) 2진값 00, 01, 10, 11 순으로 매 입력 펄스마다 반복적으로 진행되는 순차 회로
- 2) 2 bit는 0부터 $2^2-1=3$ 까지 상태를 계수할 수 있음



<Figure 4> 2 비트 2진 카운터 상태도

Present State Q	Next State Q*	
	input x = 0	input x = 1
00	00	01
01	01	10
10	10	11
11	11	00

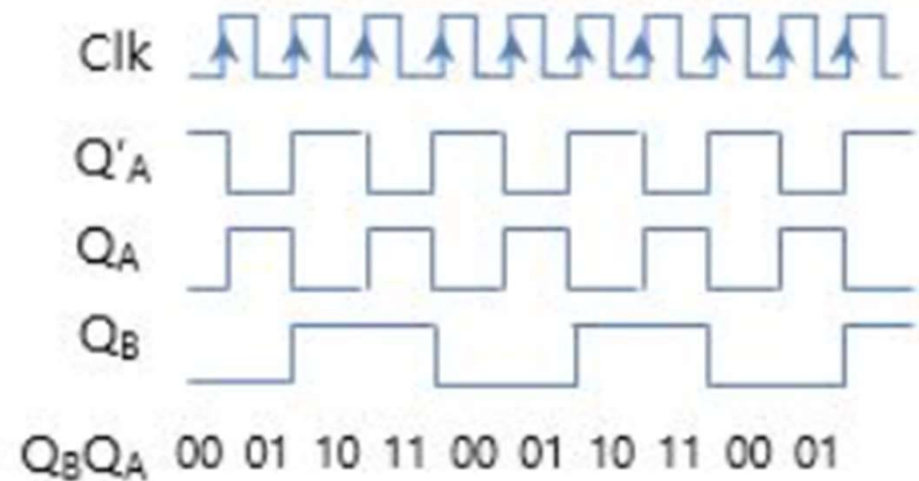
<Figure 5> 2 비트 2진 카운터 상태표

2. 비동기식 카운터

2. 2 비트 2진 비동기식 카운터



<Figure 6> 2비트 2진 비동기식 카운터 논리 회로 (JK 플립플롭)

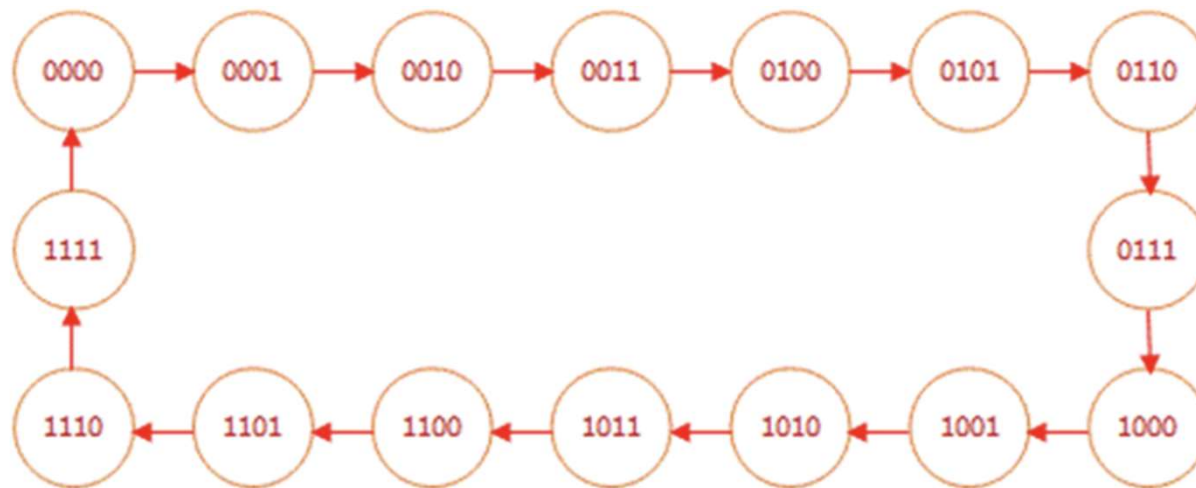


<Figure 7> 2비트 2진 비동기식 카운터 timing diagram

2. 비동기식 카운터

3. 4 비트 2진 비동기식 카운터

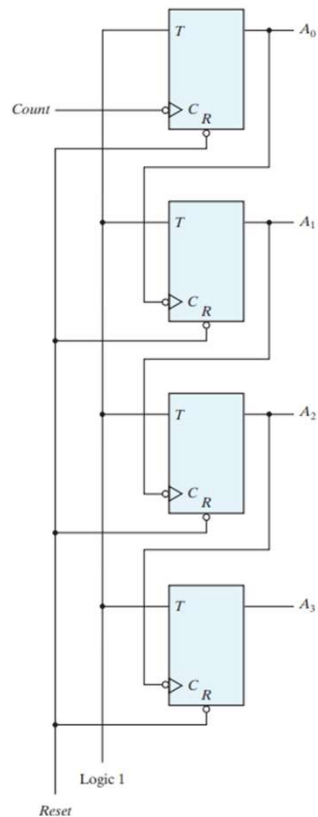
- 1) 2진값 0000에서 1111 까지 순차적으로 증가하며 반복적으로 진행하는 순차 회로
- 2) 4 bit는 0부터 $2^4-1=15$ 까지 상태를 계수할 수 있음



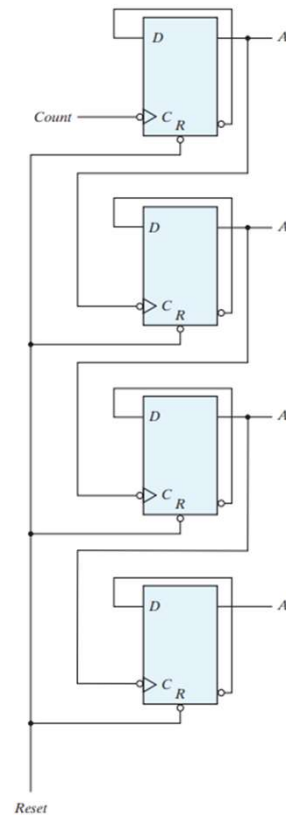
<Figure 8> 4비트 2진 상향 카운터 상태도

2. 비동기식 카운터

3. 4 비트 2진 비동기식 카운터



<Figure 9> T 플립플롭을 사용한 4 비트 이진 비동기식 카운터



<Figure 10> D 플립플롭을 사용한 4 비트 이진 비동기식 카운터

2. 비동기식 카운터

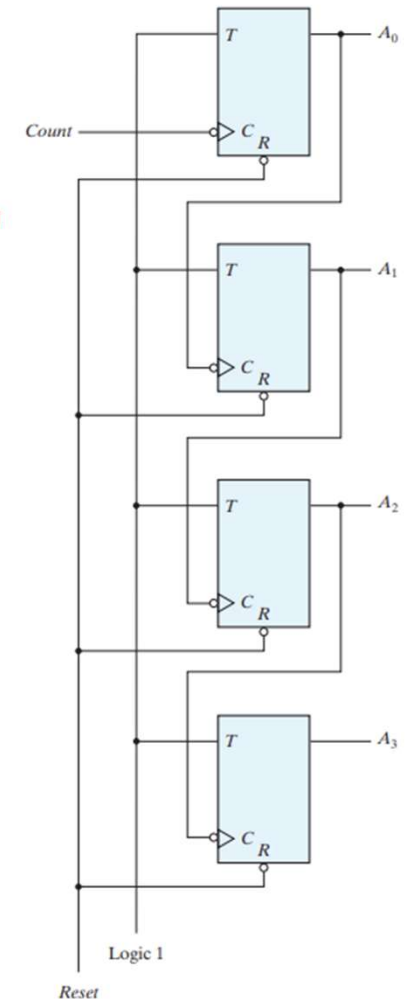
3. 4 비트 2진 비동기식 카운터

3) 동작 과정

A_3	A_2	A_1	A_0
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

<Figure 11> 2진 카운터 순서

- * 최하위 비트 A_0 : 각 카운트 펄스 입력마다 보수화됨
- * A_1 : A_0 가 1에서 0으로 바뀔 때마다 A_1 보수화
- * A_2 : A_1 가 1에서 0으로 바뀔 때마다 A_2 보수화
- * A_3 : A_2 가 1에서 0으로 바뀔 때마다 A_3 보수화



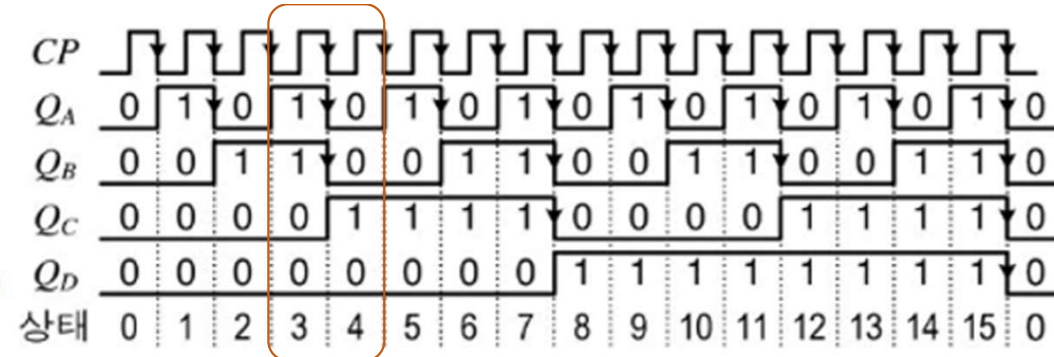
<Figure 9> T 플립플롭을 사용한 4 비트 이진 비동기식 카운터

2. 비동기식 카운터

3. 4 비트 2진 비동기식 카운터

ex) 0011 -> 0100

- 카운트 펄스에 의해 A_0 가 보수화(0011->0010)
- A_0 가 1에서 0으로 바뀌면서 A_1 보수화(0010->0000)
- A_1 이 1에서 0으로 바뀌면서 A_2 보수화(0000->0100)
- A_2 가 0에서 1로 바뀌면서 상승 천이
- 이때 플립플롭은 하향 천이에서 응답하므로 A_3 은 트리거되지 않음(보수화 X)



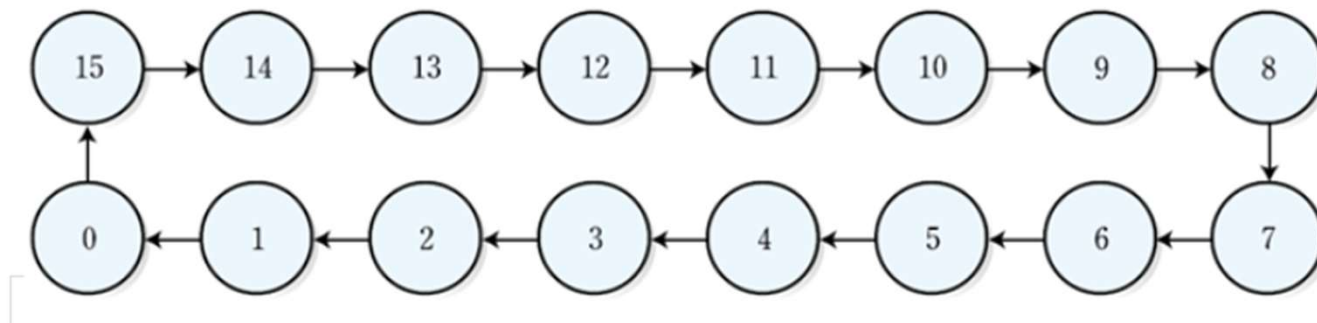
<Figure 12> 4 비트 이진 비동기식 카운터의 timing diagram

2. 비동기식 카운터

3. 4 비트 2진 비동기식 카운터

4) 하향 카운터

- * 매 입력 카운트 펄스마다 2진 카운터가 1씩 감소
- * 4 비트 2진 하향 카운터는 15에서 시작하여 14, 13, ...0 순으로 카운트하고 다시 15로 돌아감. (1111->0000)



<Figure 13> 4비트 2진 하향 카운터의 상태도

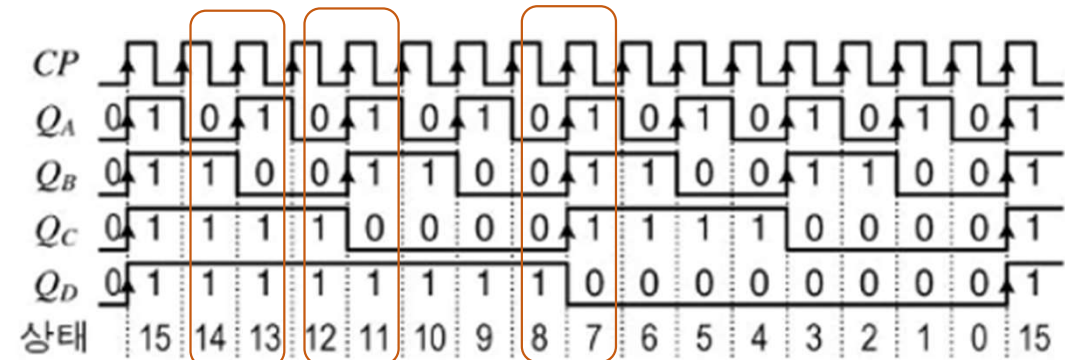
2. 비동기식 카운터

3. 4 비트 2진 비동기식 카운터

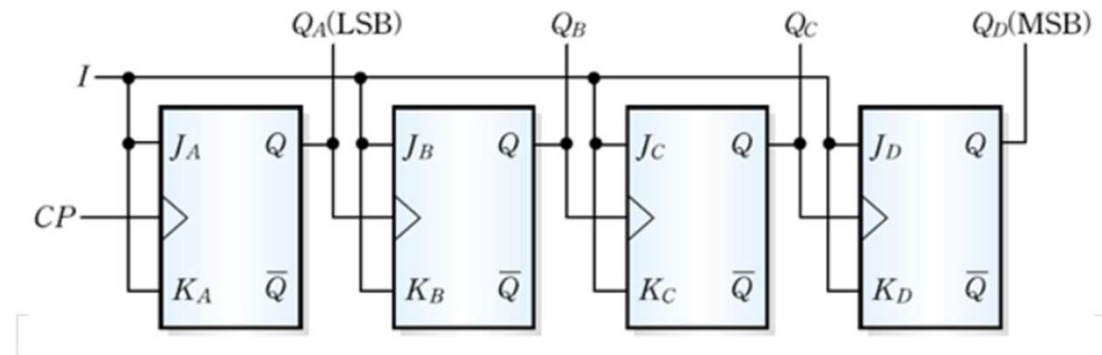
4) 하향 카운터

카운트 되는 순서를 보면 각 비트는
그 아래의 하위 비트가 0에서 1로
바뀔 때 보수화되는 것을 알 수 있음

⇒ 모든 플립플롭이 상향 에지에서
트리거되면 하향 카운터로 작동



<Figure 14> 4 비트 2진 하향 카운터 순서 timing diagram

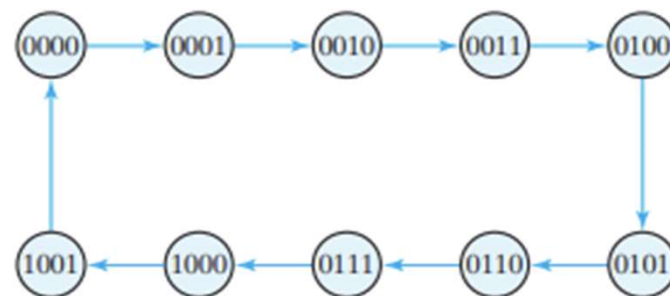


<Figure 15> JK 플립플롭을 사용한 4 비트 2진 하향 카운터 논리 회로

2. 비동기식 카운터

4. BCD 비동기식 카운터

- 1) 10 개의 상태를 0에서 9까지 카운트하고 다시 0으로 돌아옴
- 2) 4 비트의 2진 코드로 표현되는데, 16 개의 상태 중 10 개의 상태만 사용



<Figure 16> BCD 카운터의 상태도

2. 비동기식 카운터

4. BCD 비동기식 카운터

3) 동작 과정

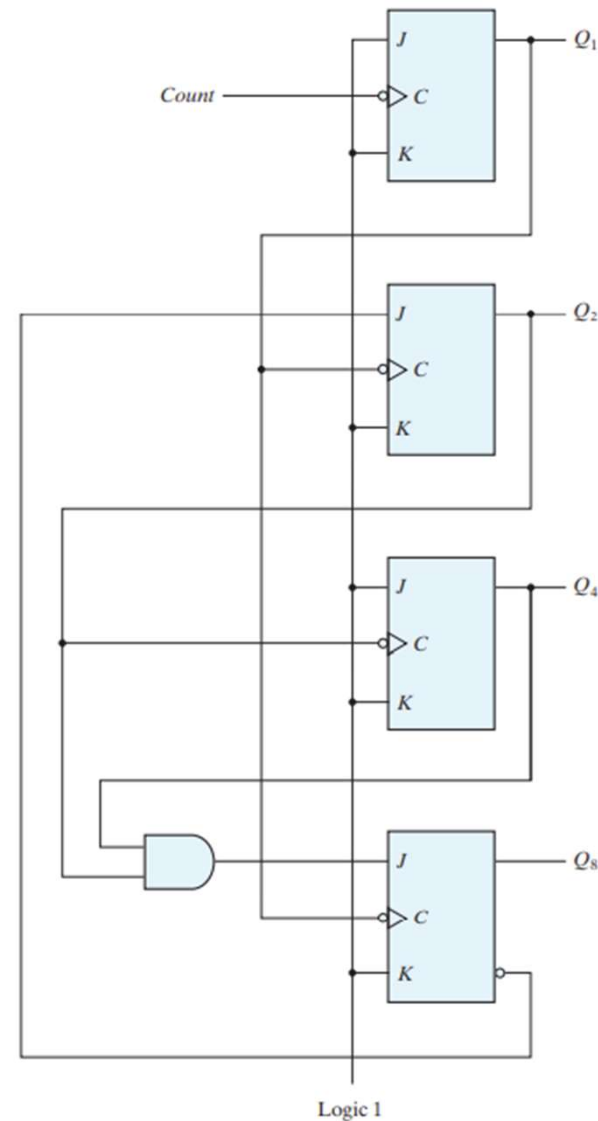
(JK 플립플롭을 사용한 BCD 비동기식 카운터)

* 출력(4 개): Q_1, Q_2, Q_4, Q_8

*

J	K	q^*
0	0	q
0	1	0
1	0	1
1	1	q'

<Figure 17> JK 플립플롭의 진리표



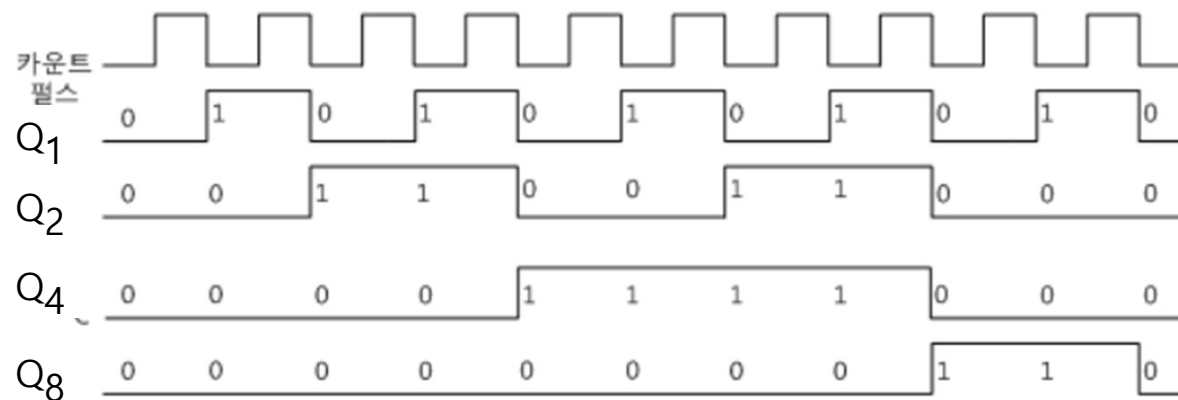
<Figure 18> JK 플립플롭을 사용한 BCD 비동기식 카운터의 논리 회로

2. 비동기식 카운터

4. BCD 비동기식 카운터

3) 동작 과정

(JK 플립플롭을 사용한 BCD 비동기식 카운터)



<Figure 19> JK 플립플롭을 사용한 BCD 비동기식 카운터의 timing diagram

* Q_1 : 매 클럭 펄스마다 상태가 변함

* Q_2 : $Q_8 = 0$ 일 때, Q_1 이 1에서 0으로 바뀔 때 Q_2 보수화됨

* Q_4 : Q_2 가 1에서 0으로 바뀔 때 마다 Q_4 보수화됨

* Q_8 : Q_2 또는 Q_4 가 0일 때 0으로 유지됨, Q_2 와 Q_4 가 모두 1이면 Q_1 이 1에서 0으로 바뀔 때 보수화됨, Q_1 의 다음 1에서 0으로 바뀌는 지점에서 Q_8 클리어됨

-

<Figure 20> 4비트 동기식 2진 카운터

3. 동기식 카운터

1. 4 비트 동기식 2진 카운터

1) 최하위 자리에 있는 플립플롭이 매 펄스마다 보수화

2) 다른 자리에 있는 플립플롭들은 하위의 모든 비트가 1이 되면 보수화

Ex) $A_3A_2A_1A_0 = 0011$

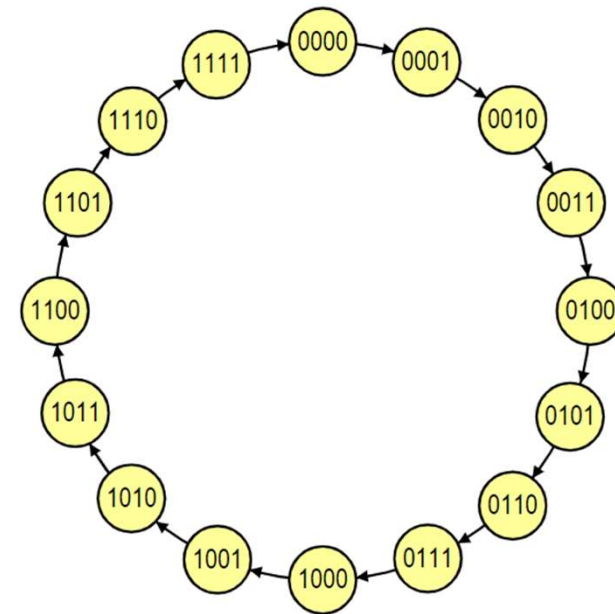
① A_0 는 항상 보수화

② A_1 은 현재 상태가 1이기에 보수화

③ A_2 도 현재 상태가 11이니까 보수화

④ A_3 는 현재상태가 011이니까 보수화되지
않음.

-> 모든 하위 비트들이 1이 되어야 보수화가
됨

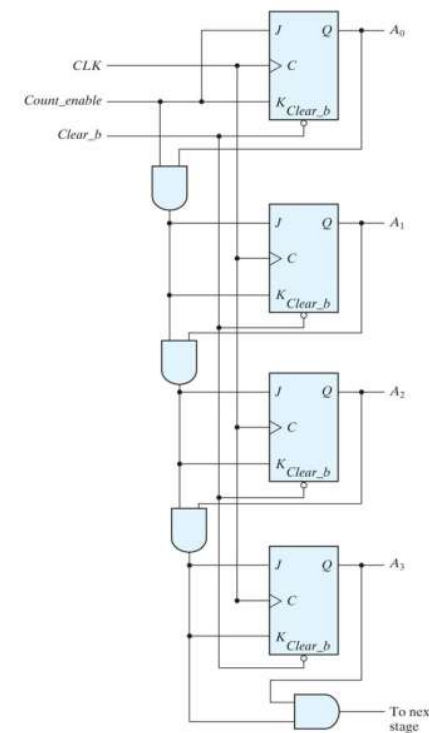


<Figure 21> 4비트 동기식 2진 카운터 상태도

3. 동기식 카운터

1. 4비트 동기식 2진 카운터

- 1) 규칙적인 하드웨어 패턴을 가지며, 보수화 플립플롭과 게이트를 이용해 구성
- 2) 2진 카운터의 보수화 플립플롭은 JK, T, D 플립플롭 중 하나를 사용하여 구성 가능
- 3) 동기식 카운터는 상향이나 하향 엣지 둘 중 어떤 것을 사용해도 트리거 가능



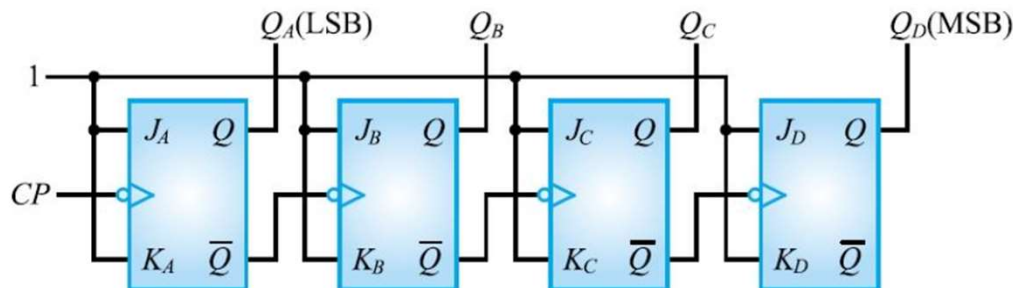
<Figure 20> 4비트 동기식 2진 카운터

3. 동기식 카운터

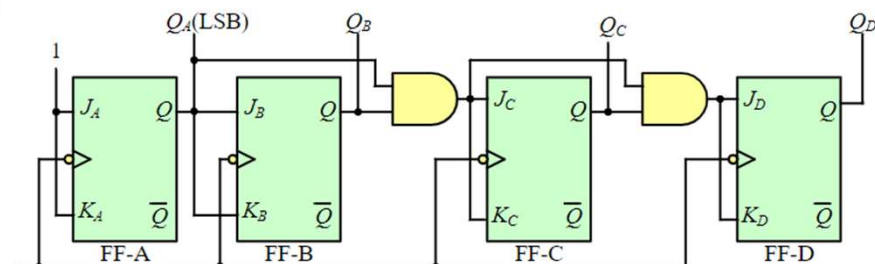
1. 4 비트 동기식 2진 카운터

1) 비동기식과는 다른 CP의 연결 형태

- 동시에 동기가 되어야 하기 때문에 각각의 플립플롭에 클럭이 연결되어 있다.



<Figure 22> 비동기식 카운터



<Figure 23> 동기식 카운터

3. 동기식 카운터

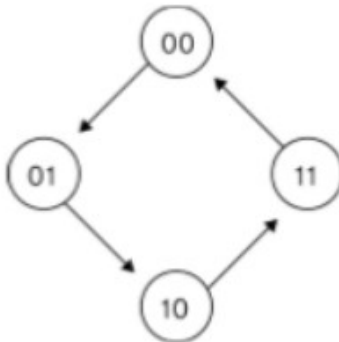
2. **n 비트** 동기식 2진 카운터 **설계 과정**

- 1) 상태분석 및 다이어그램 작성
- 2) 상태표, 여기표 작성
- 3) 입력함수 유도
- 4) 회로 구성

3. 동기식 카운터

2. **n 비트 동기식 2진 카운터 설계 과정**

1) 상태분석 및 다이어그램 작성



<Figure 24> 2비트 동기식 2진 카운터 상태도

3. 동기식 카운터

2. n 비트 동기식 2진 카운터 설계 과정

- 2) 상태표, 여기표 작성
- 3) 입력함수 유도(카르노 맵 등)

<table 1> 2비트 동기식 2진 카운터 상태 여기표

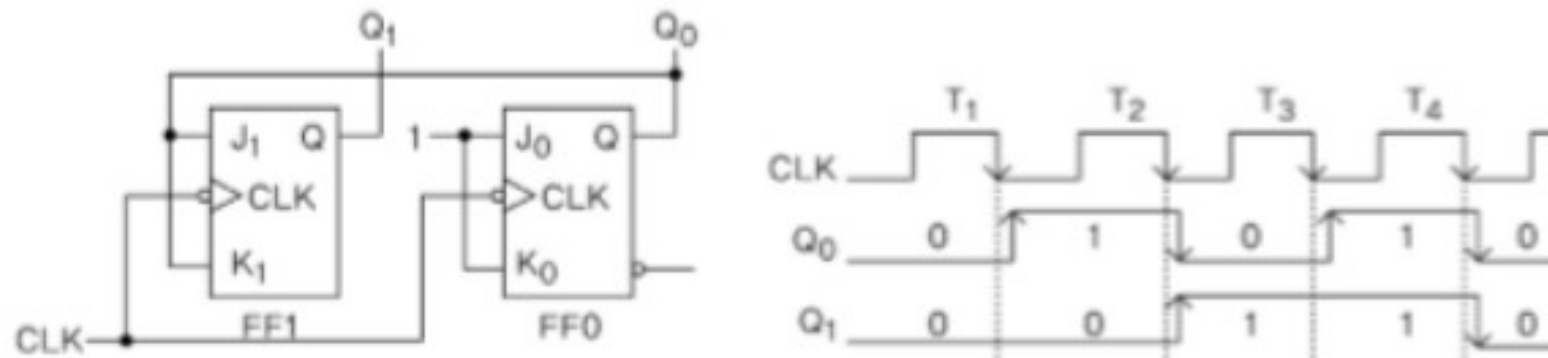
현재 상태		다음상태		플립플롭 입력			
Q_1	Q_0	$Q_1(t+1)$	$Q_0(t+1)$	J_1	K_1	J_0	K_0
0	0	0	1	0	x	1	x
0	1	1	0	1	x	x	1
1	0	1	1	x	0	1	x
1	1	0	0	x	1	x	1

$$\begin{array}{ll}
 J_0 = 1 & K_0 = 1 \\
 J = Q_0 & K_1 = Q_0
 \end{array}$$

3. 동기식 카운터

2. n 비트 동기식 2진 카운터 설계 과정

4) 회로 구성

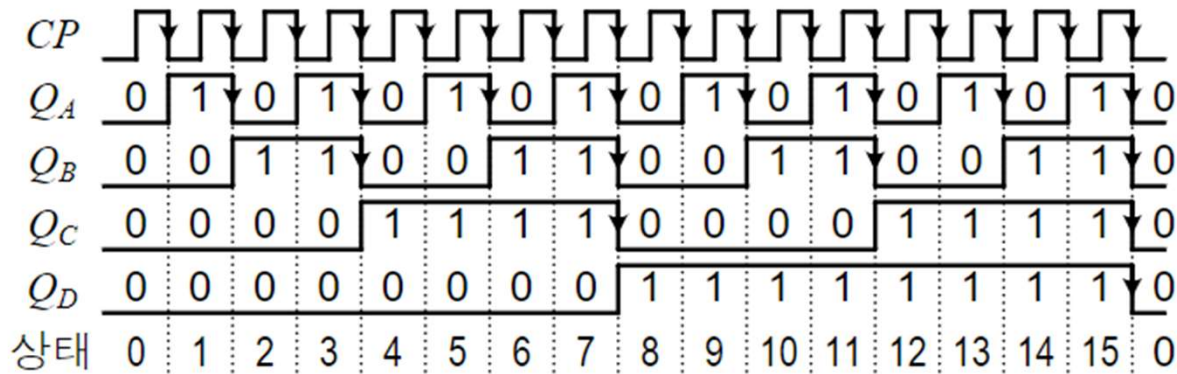


<Figure 25> 2비트 동기식 2진 카운터의 순차 논리 회로와 타이밍 다이어그램

3. 동기식 카운터

2. n 비트 동기식 2진 카운터 설계 과정

- 4비트 동기식 2진 카운터



$$J_A = 1$$

$$K_A = 1$$

$$K_B = Q_A$$

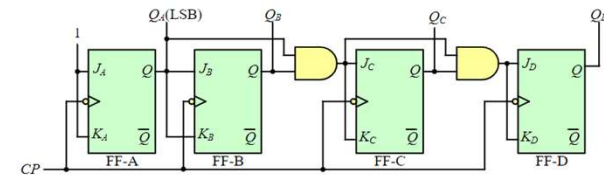
$$J_B = Q_A$$

$$J_C = Q_B Q_A$$

$$K_C = Q_B Q_A$$

$$J_D = Q_C Q_B Q_A$$

$$K_D = Q_C Q_B Q_A$$

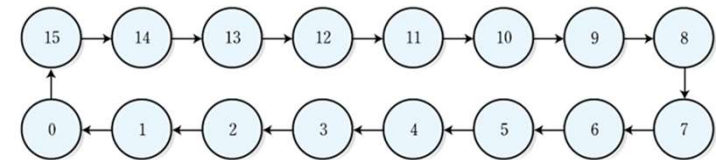


<Figure 26> 4비트 동기식 2진 카운터의 순차 논리 회로와 타이밍 다이어그램

3. 동기식 카운터

3. 상향/하향 2진 카운터

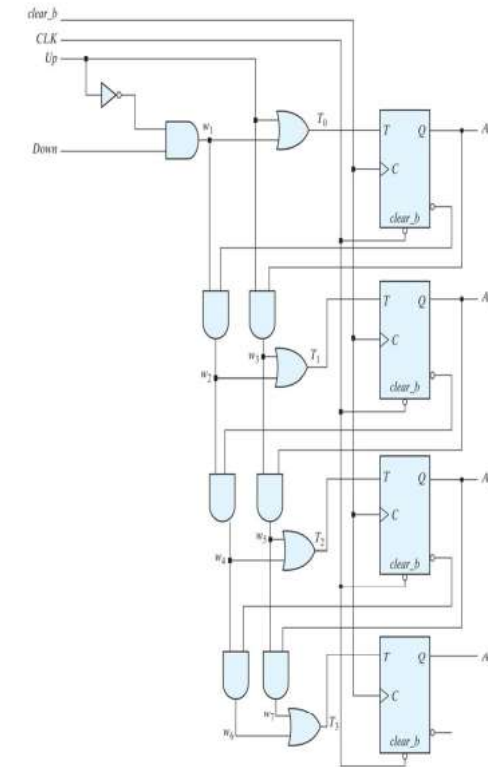
- 1) 동기식 하향 2진 카운터는 1111부터 0000까지 2진 상태를 역순으로 진행시키다가 0000이 되면 다시 1111로 돌아가 카운트를 반복
- 2) 최하위 비트는 매 펄스마다 반전, 이외의 비트는 하위의 다른 모든 비트들이 0일때 보수화
- 3) 0100의 다음상태는 0011, 0100의 최하위 비트는 항상 보수화되고, 두 번째 하위 비트는 첫 번째 비트가 0이기 때문에 보수화된다. 세 번째도 보수화되고, 네 번째는 세 번째 비트가 1이기 때문에 바뀌지 않음



3. 동기식 카운터

3. 상향/하향 2진 카운터

- 1) 상향/하향 모두 가능한 카운터
- 2) 상향, 하향 제어 입력을 모두 가지고 있다.
- 상향 입력이 1이면, 상향으로 카운터
- 하향 입력이 1, 상향 입력이 0이면, 회로는 하향으로 카운트
- 상향과 하향 입력이 모두 1이면, 회로는 상향으로 카운트
- 1) 위의 조건들은 주어진 조건 중 하나의 작업만 수행함(상향입력이 하향 입력보다 우선됨)



<Figure 27> 4비트 상하향 2진 카운터

3. 동기식 카운터

4. BCD 카운터

- 1) BCD 카운터는 0000에서 1001까지 카운트하고 다시 0000으로 되돌아감.
-> 규칙적인 패턴을 갖지 않음.
- 2) Y는 카운터의 현재 상태가 1001이면 1이 된다.

<Table 2> State table of BCD counter

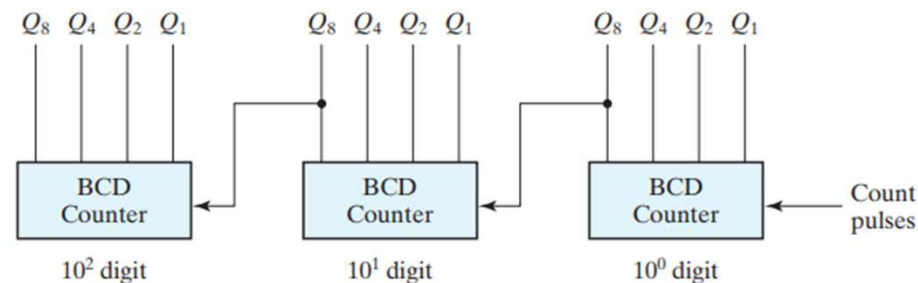
Present State				Next State				Output	Flip-Flop Inputs			
Q_8	Q_4	Q_2	Q_1	Q_8	Q_4	Q_2	Q_1	y	T_{Q8}	T_{Q4}	T_{Q2}	T_{Q1}
0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	1	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	1	1	0	0	0	0	1
0	0	1	1	0	1	0	0	0	0	1	1	1
0	1	0	0	0	1	0	1	0	0	0	0	1
0	1	0	1	0	1	1	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	0	0	1
0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	0	0	1	1	0	0	1

$$\begin{aligned}
 T_{Q1} &= 1 \\
 T_{Q2} &= Q_8'Q_1 \\
 T_{Q4} &= Q_2Q_1 \\
 T_{Q8} &= Q_8Q_1 + Q_4Q_2Q_1 \\
 y &= Q_8Q_1
 \end{aligned}$$

4. 비동기식/동기식 카운터

<BCD 카운터>

- 1) BCD 카운터의 직렬 연결을 통해 어떤 임의의 길이의 10진수에 대해서도 카운터 구성이 가능하다.
- 2) 동기식 카운터의 경우 Table 1의 출력 y가 다음 상위 10진수 자리의 카운트 입력에 연결되어야 한다는 차이가 있다.



<Figure 28> 3자리수 10진수를 위한 10진식 BCD카운터의 블록도

4. 비동기식/동기식 카운터

1) 비동기식과 동기식은 동작면에서 어떻게 다른가?

→ 동기식 카운터의 모든 플립플롭은 공통된 클럭펄스를 수신해서 동기화

→ 비동기식 카운터는 단지 첫 단계에서만 클럭을 수신.

→ 동기식 카운터는 공통 클럭에 의해 모든 단계들이 동시에 갱신

→ 비동기식 카운터는 한 번에 한 단계 씩 갱신

2) 즉, 동기식 카운터가 더 빠른 특징이 있다.

3. 동기식 카운터

플립플롭에서의 전파지연 시간: t_{PD}

N개의 플립플롭을 종속 연결한 비동기 카운터의
전체 전파 지연 시간: $N \times t_{PD}$

★ 동기식 카운터를 사용하는 이유?

-> 지연 시간을 줄이기 위함

5. Finite-state machine(유한 상태 머신)

1) 동기적 순차 회로를 기술하는 추상화 모델

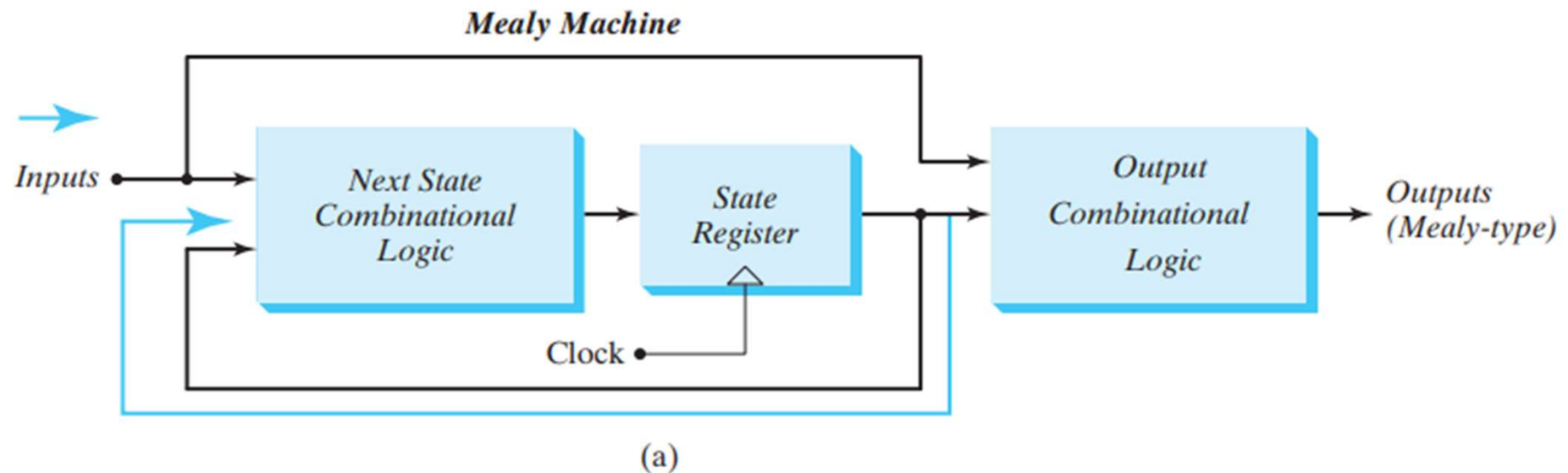
- 카운터와 같은 회로에서 각각의 상태를 구분하여 입력에 따른 상태의 변화를 도식화하여 그림이나 도표로 나타냄

5. Finite-state machine(유한 상태 머신)

- 1) 순차 회로의 일반적인 모델 - 입력, 출력, 내부상태로 구성
- 2) 순차 회로의 구별
 - 밀리 모델
 - 무어 모델

5. Finite-state machine(유한 상태 머신)

1) **밀리 모델**은 입력과 현재 상태 모두 출력에 영향



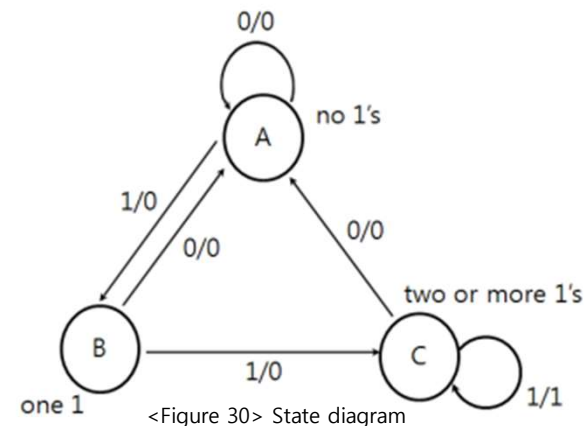
<Figure 29> Mealy Machine

5. Finite-state machine(유한 상태 머신)

1) **밀리 모델** EX) A system with one input x and one output z such that $z = 1$ iff x has been 1 for the last three clock times

<Table 3> State table

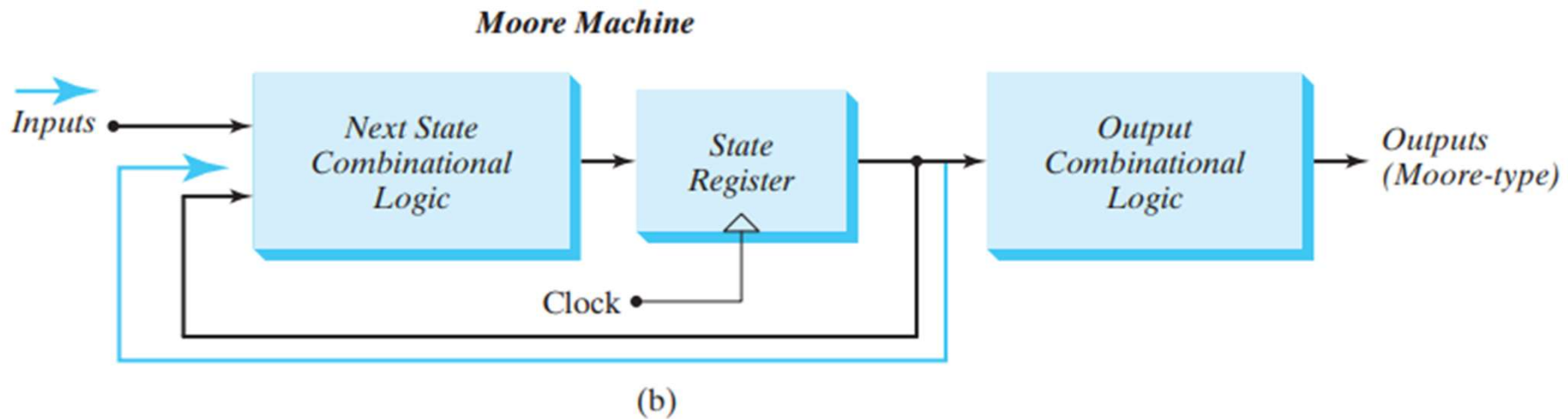
q	q*		z	
	x=0	x=1	x=0	x=1
A	A	B	0	0
B	A	C	0	0
C	A	C	0	1



x	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1	1	0	0		
q	?	A	→B	C	A	B	C	C	A	A	B	A	B	A	B	C	C	C	C	A	A
z	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0

5. Finite-state machine(유한 상태 머신)

2) 무어 모델은 현재상태만 출력에 영향을 줌



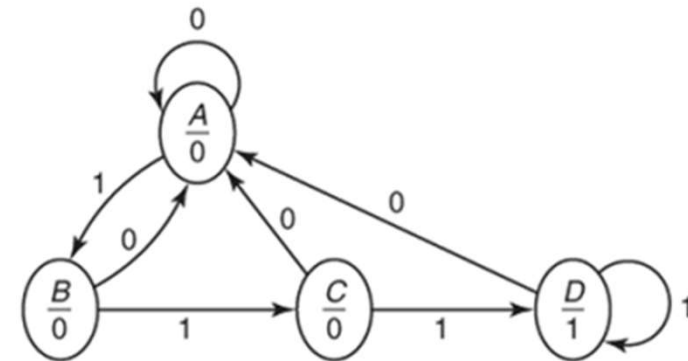
<Figure 31> Moore Machine

5. Finite-state machine(유한 상태 머신)

2) 무어모델 EX) A system with one input x and one output z such that $z = 1$ iff x has been 1 for the last three clock times

<Table 4> State table

Present state	Next state		Output
	$x = 0$	$x = 1$	
A	A	B	0
B	A	C	0
C	A	D	0
D	A	D	1

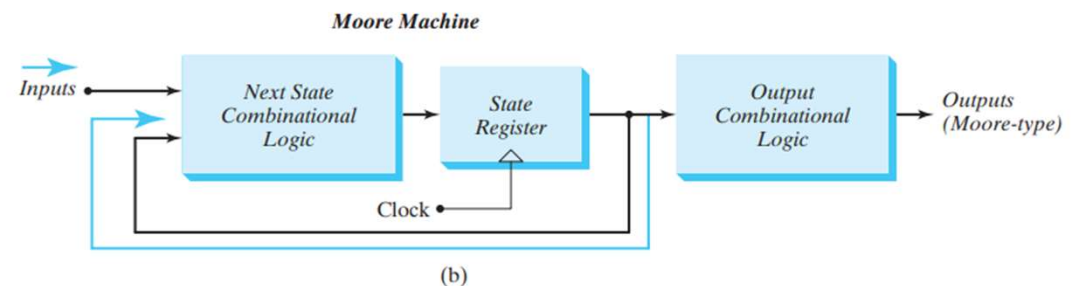
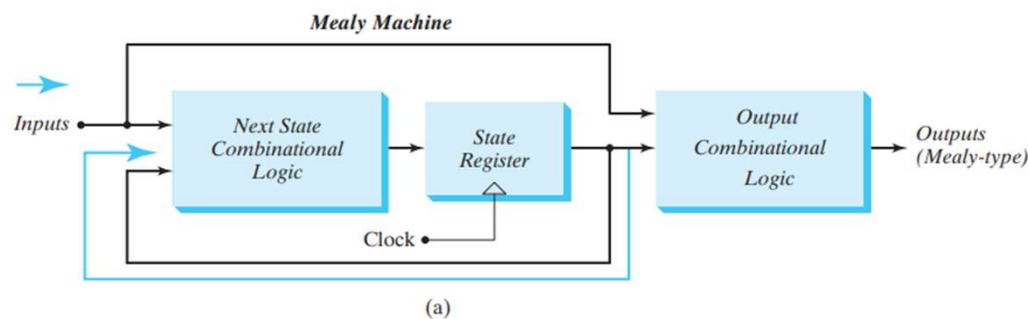


<Figure 32> State diagram

x	0	1	1	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0			
q	?	A	→ B	C	A	B	C	D	A	A	B	A	B	C	D	D	D	A	A	?	
z	?	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0

5. Finite-state machine(유한 상태 머신)

1) 두 가지 모델의 순차회로를 Finite state machine이라고 부름



<Figure 33> Mealy, Moore Machine

참조문헌 및 팀원별 기여도

참조문헌

- 1) M Morris Mano, Michael D. Ciletti. "Digital design: with an introduction to the verilog HDL, VHDL, and system Verilog." (2017).
- 2) Alan B. Marcovitz, "Introduction to logic and Computer Design(International edition)" (2008).
- 3) Rafiquzzaman, Mohamed, and Steven A. McNinch. *Digital Logic: With an Introduction to Verilog and FPGA-Based Design*. John Wiley & Sons, 2019.
- 4) Charles B. Kime and M. Morris Mano, "Logic and computer Design Fundamentals (4th edition)" (2008).

<팀원별 기여도>

김승원 50%
황유진 50%