

컴퓨터공학실험II

2024

Week 12

PRESENTATION

20231591 이하은

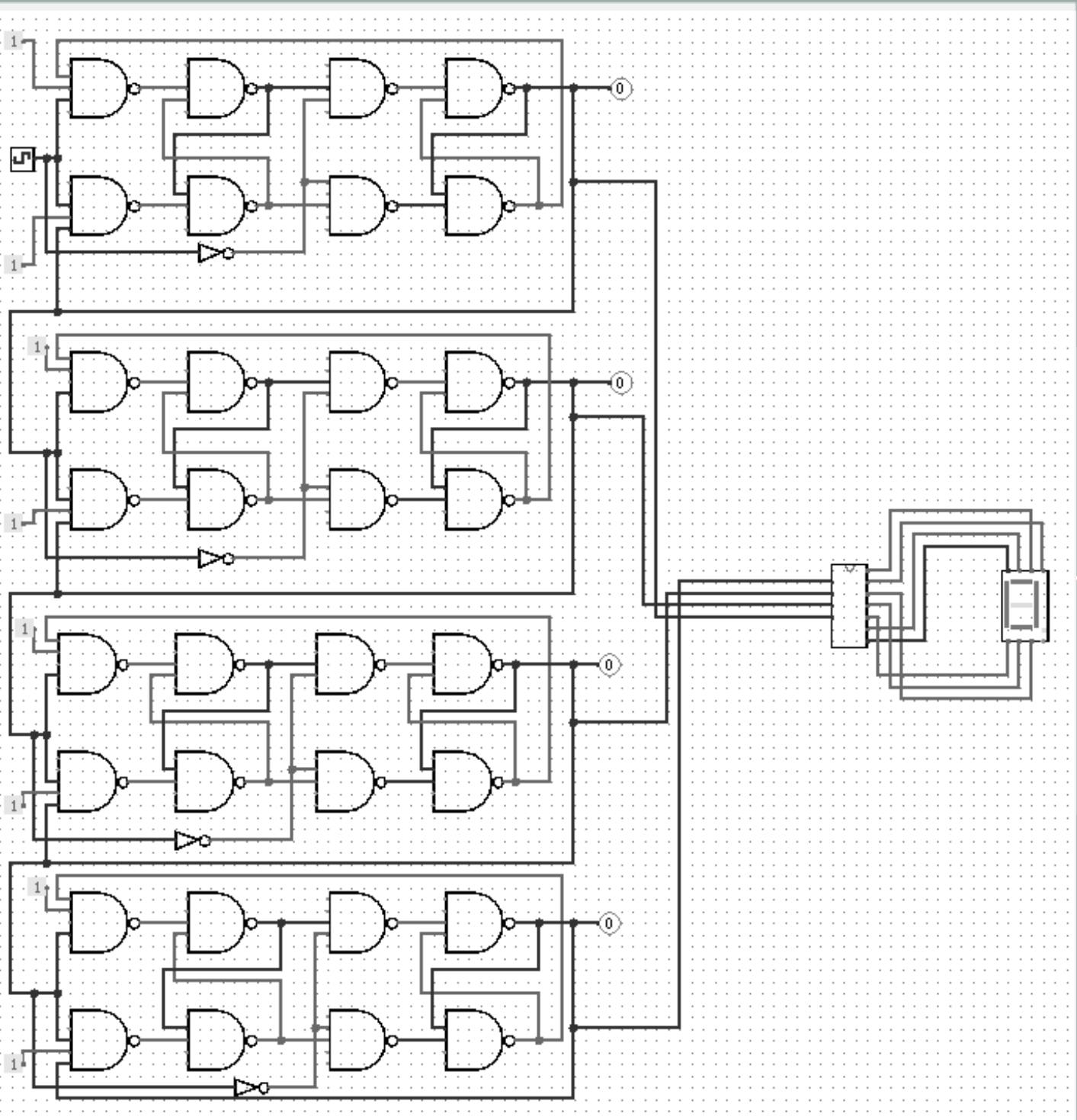
20231579 이명진

20231523 김민정

Sogang University

Counter

카운터 개념



정의 Definition

특정한 상태들의 고정된 순서를 클럭 신호에 따라 진행
주기적인 이벤트를 세거나 클럭을 기반으로 시간을 측정하는 데 사용
데이터 입력 X, 플립플롭의 상태와 출력이 동일

D
U

주요 용도 Use

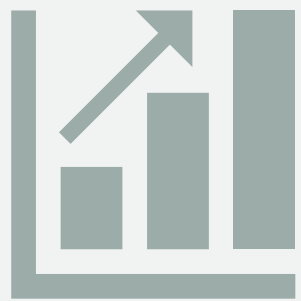
특정 신호의 발생 횟수를 셈
클럭을 기준으로 일정 시간 간격을 계산
신호를 원하는 주파수로 분할

Counter

01

대표적인 종류

업 카운터



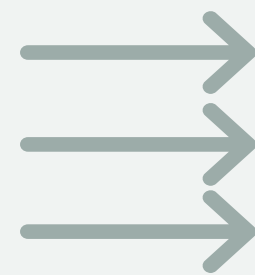
- 클럭 신호가 들어올 때마다 카운터 값이 증가
- 00→01→10→11
- 이벤트 카운터, 시간 카운터 등에 활용

다운 카운터



- 클럭 신호가 들어올 때마다 카운터 값이 감소
- 11→10→01→00
- 카운트다운 타이머, 감소 기반 프로세스 제어 등에 활용

동기식 카운터



- 뒤에서 후술

비동기식 카운터



- 뒤에서 후술

Counter

예시

비표준 카운터 (Non-Standard)

임의의 상태 전환 순서를 가지는 카운터
자연스러운 이진수 순서가 아닌, 임의의 순서

■ 0 3 2 4 1 5 7 and repeat

q_1	q_2	q_3	q_1^\star	q_2^\star	q_3^\star
0	0	0	0	1	1
0	0	1	1	0	1
0	1	0	1	0	0
0	1	1	0	1	0
1	0	0	0	0	1
1	0	1	1	1	1
1	1	0	X	X	X
1	1	1	0	0	0

SR 플립플롭과 T 플립플롭의 진리표

S_1	R_1	S_2	R_2	S_3	R_3	T_1	T_2	T_3
0	X	1	0	1	0	0	1	1
1	0	0	X	X	0	1	0	0
1	0	0	1	0	X	1	1	0
0	X	X	0	0	1	0	0	1
0	1	0	X	1	0	1	0	1
X	0	1	0	X	0	0	1	0
X	X	X	X	X	X	X	X	X
0	1	0	1	0	1	1	1	1

Counter

예시

비표준 카운터 (Non-Standard)

임의의 상태 전환 순서를 가지는 카운터
자연스러운 이진수 순서가 아닌, 임의의 순서

01

$q_2 q_3$		q_1	
		0	1
00			
01	1	X	
11			
10	1	X	

S_1

$q_2 q_3$		q_1	
		0	1
00	1		
01			1
11	X		
10			X

S_2

$q_2 q_3$		q_1	
		0	1
00	1	1	
01	X	X	
11			
10			X

S_3

$q_2 q_3$		q_1	
		0	1
00	X	1	
01			
11	X	1	
10			X

R_1

$q_2 q_3$		q_1	
		0	1
00			X
01	X		
11			1
10	1	X	

R_2

$q_2 q_3$		q_1	
		0	1
00			
01			
11	1	1	
10	X	X	

R_3

$$S_1 = q_2'q_3 + q_2q_3'$$

$$R_1 = q_2'q_3' + q_2q_3 = S_1'$$

$$= q_2'q_3' + q_1q_2$$

$$S_2 = q_1'q_2'q_3' + q_1q_2'q_3$$

$$R_2 = q_1q_2 + q_2q_3'$$

$$S_3 = q_2'$$

$$R_3 = q_2$$

SR 플립플롭의 논리식 간소화

Counter

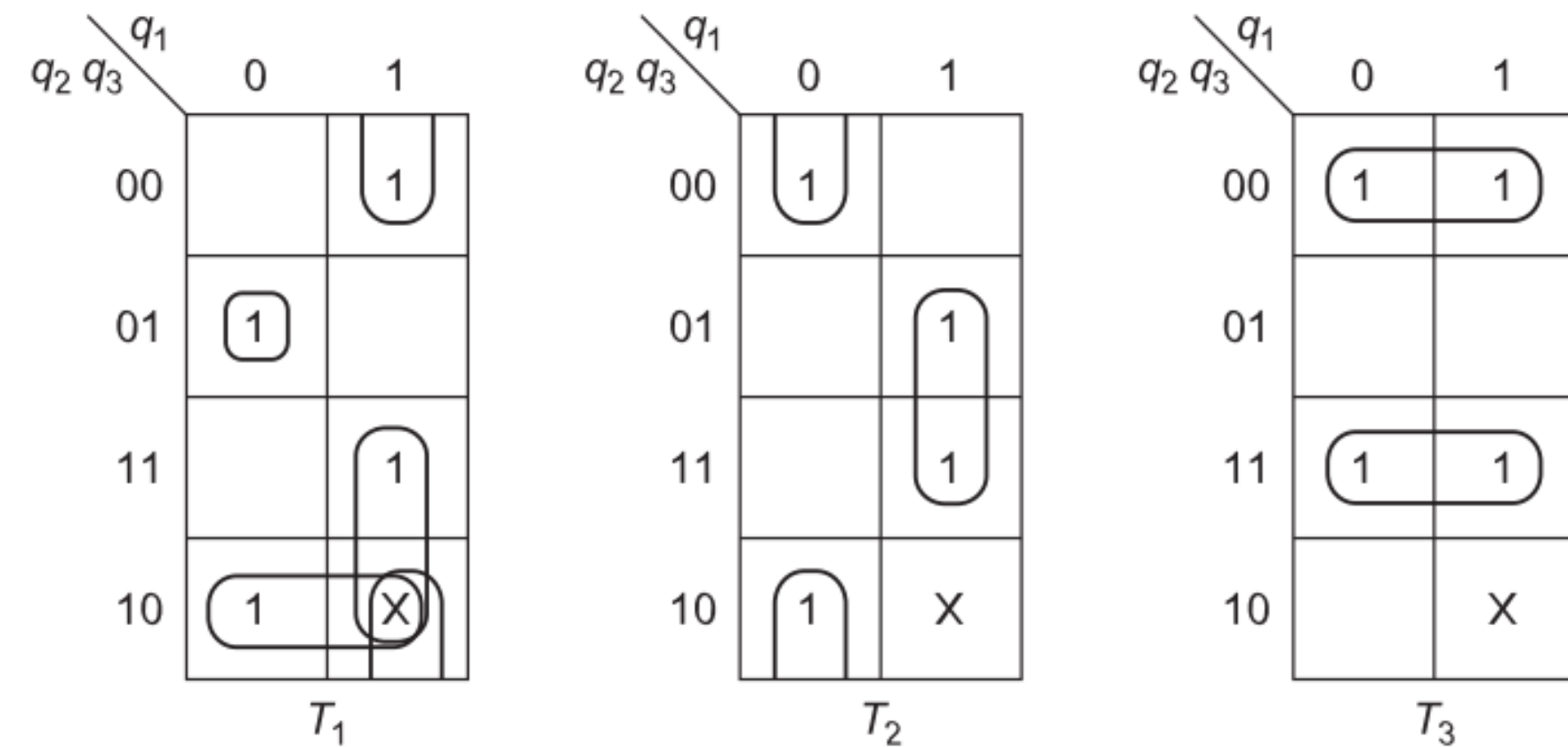
예시

비표준 카운터 (Non-Standard)

임의의 상태 전환 순서를 가지는 카운터
자연스러운 이진수 순서가 아닌, 임의의 순서

01

T 플립플롭의 논리식 간소화

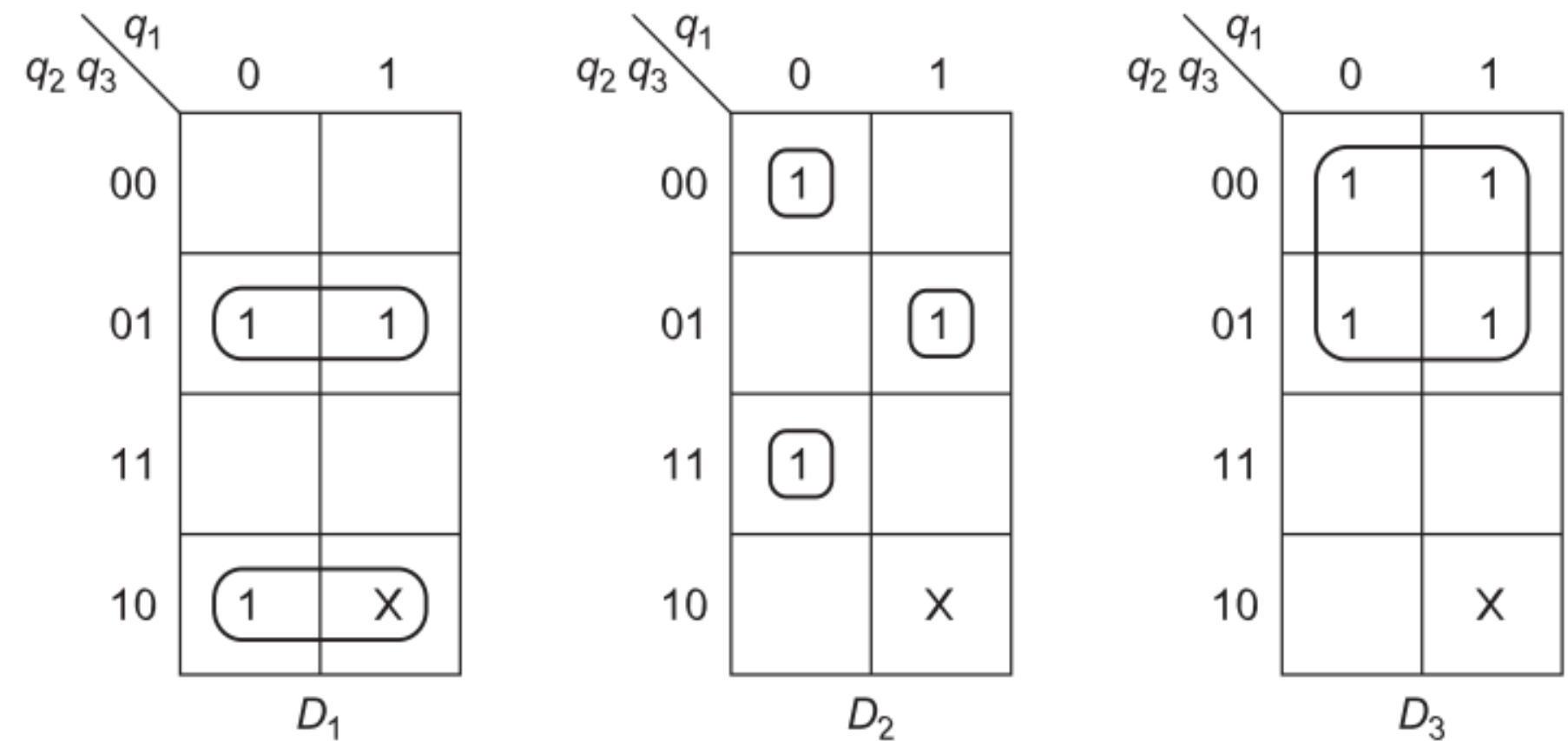


$$T_1 = q_1'q_2'q_3 + q_2q_3' + q_1q_2 + q_1q_3'$$

$$T_2 = q_1'q_3' + q_1q_3$$

$$T_3 = q_2'q_3' + q_2q_3$$

D 플립플롭의 논리식 간소화



$$D_1 = q_2'q_3 + q_2q_3'$$

$$D_2 = q_1'q_2'q_3' + q_1'q_2q_3 + q_1q_2'q_3$$

$$D_3 = q_2'$$

Counter

예시

비표준 카운터 (Non-Standard)

임의의 상태 전환 순서를 가지는 카운터
자연스러운 이진수 순서가 아닌, 임의의 순서

01

Quick method를 이용한 JK 플립플롭의 논리식 간소화

$q_2 q_3$		q_1	
		0	1
00			
01		1	1
11			
10		1	X

q_1^*

$q_2 q_3$		q_1	
		0	1
00		1	
01			1
11		1	
10			X

q_2^*

$q_2 q_3$		q_1	
		0	1
00		1	1
01		1	1
11			
10			X

q_3^*

$$J_1 = q_2'q_3 + q_2q_3'$$

$$J_2 = q_1'q_3' + q_1q_3$$

$$J_3 = q_2'$$

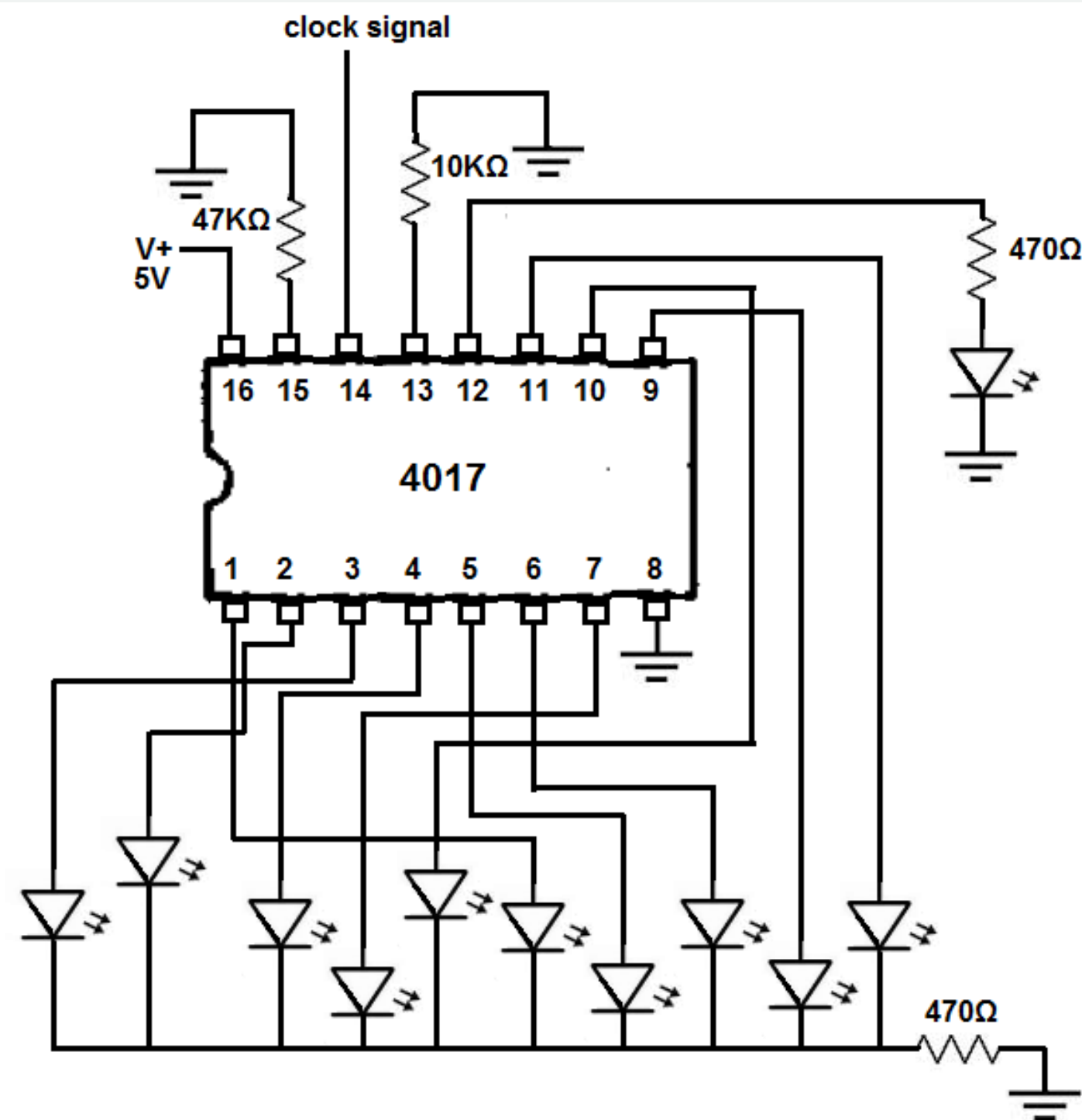
$$K_1 = q_3' + q_2$$

$$K_2 = q_1 + q_3'$$

$$K_3 = q_2$$

Decade Counter

Decade 카운터 개념



개념 Concept

0에서 9까지의 상태를 갖는 순환 카운터
클럭 신호에 따라 $0 \rightarrow 1 \rightarrow 2 \rightarrow \dots \rightarrow 9 \rightarrow 0$ 순서로 상태가 진행
BCD(Binary-Coded Decimal) 형식으로 출력

C

U

주요 용도 Use

디지털 시계, 타이머
계산기, 카운팅 디스플레이 장치 등의 디스플레이 드라이버
이외 0~9까지의 십진수가 활용되는 다양한 부문

Decade Counter

예시

0 1 2 3 4 5 6 7 8 9 and repeat

D	C	B	A	D [★]	C [★]	B [★]	A [★]
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	0	0	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Quick method를 이용한 JK 플립플롭의 논리식 간소화

D C		B A			
		00	01	11	10
00				X	1
01				X	
11			1	X	X
10				X	X

D[★]

D C		B A			
		00	01	11	10
00			1	X	
01			1	X	
11	1			X	X
10		1		X	X

C[★]

D C		B A			
		00	01	11	10
00				X	
01	1	1		X	
11				X	X
10	1	1		X	X

B[★]

D C		B A			
		00	01	11	10
00	1	1		X	1
01				X	
11				X	X
10	1	1		X	X

A[★]

$J_D = CBA$ $K_D = A$

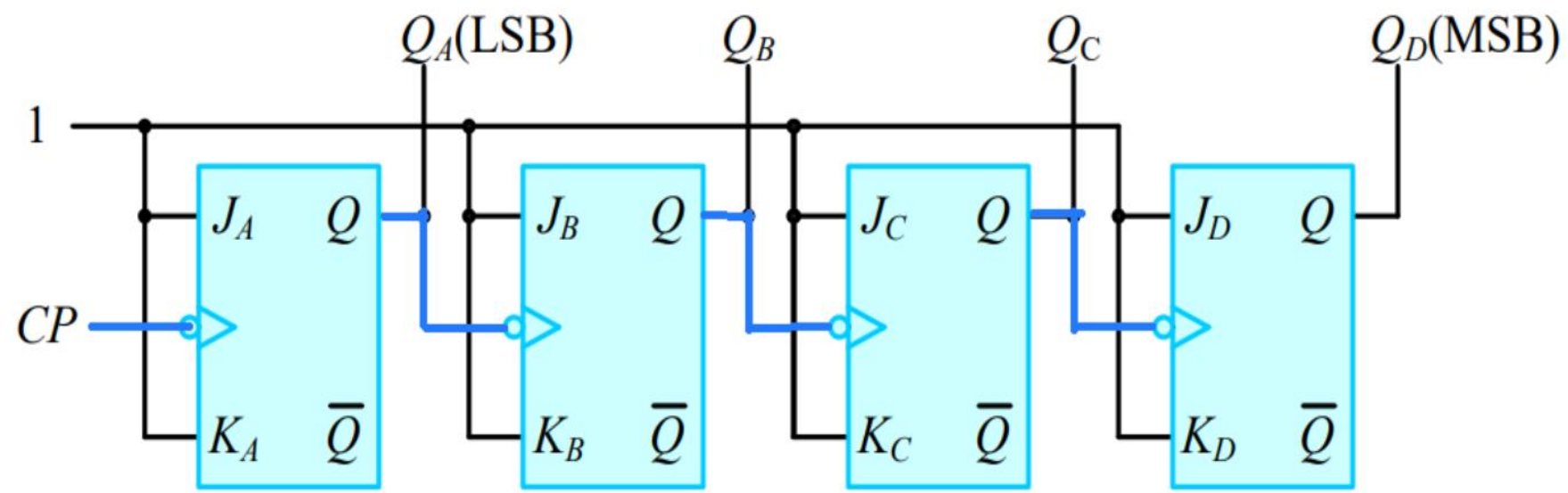
$J_C = K_C = BA$

$J_B = D'A$ $K_B = A$

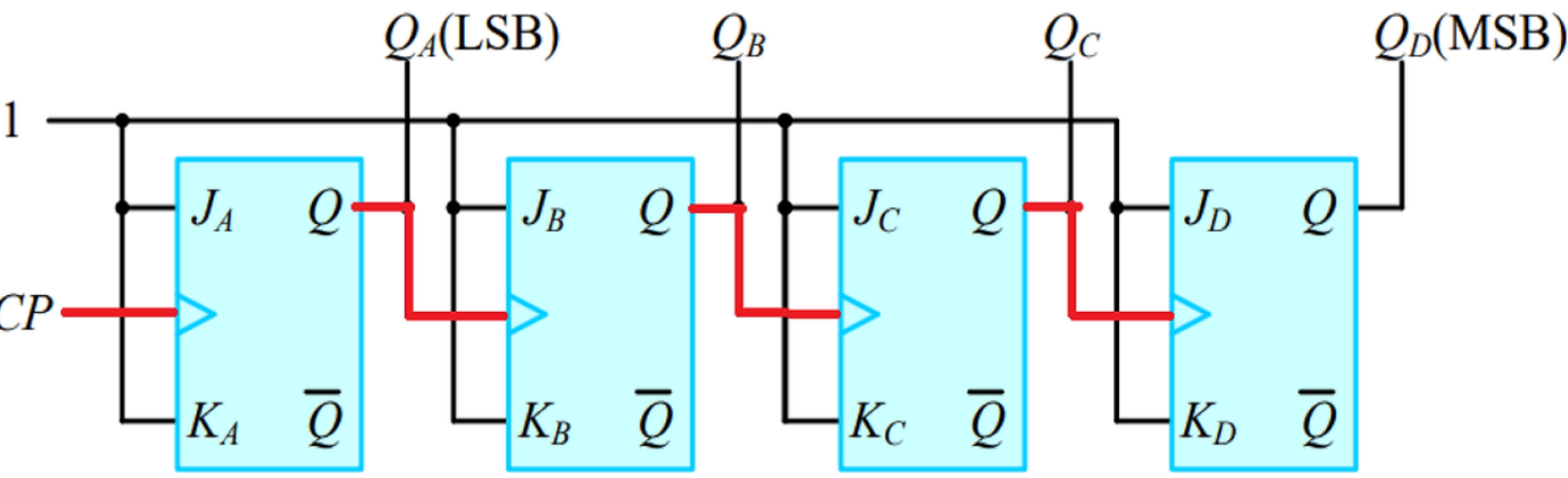
$J_A = K_A = 1$

비동기식 Counter

정의



- 4비트 2진 비동기식 상향 카운터



- 4비트 2진 비동기식 하향 카운터

개념 Concept

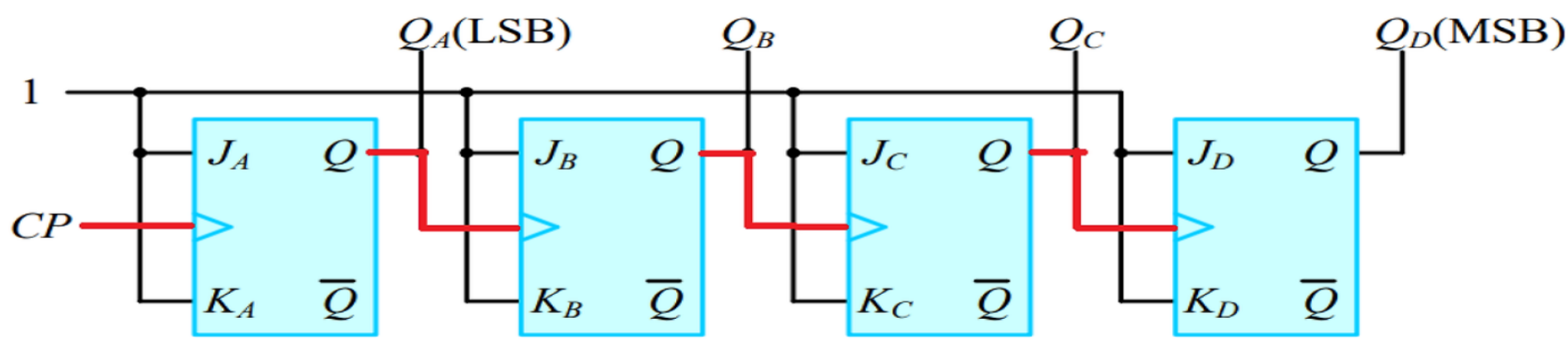
첫 번째 플립플롭의 cp 입력에만 클럭 펄스가 인가되고, 다른 플립플롭은 자신의 플립플롭의 클럭 펄스를 이전 플립플롭의 출력으로 하는 카운터

특징 Characteristic

리플 카운터라고도 부름
Jk 플립플롭, t 플립플롭, d 플립플롭 모두 사용 가능함

비동기식 Counter

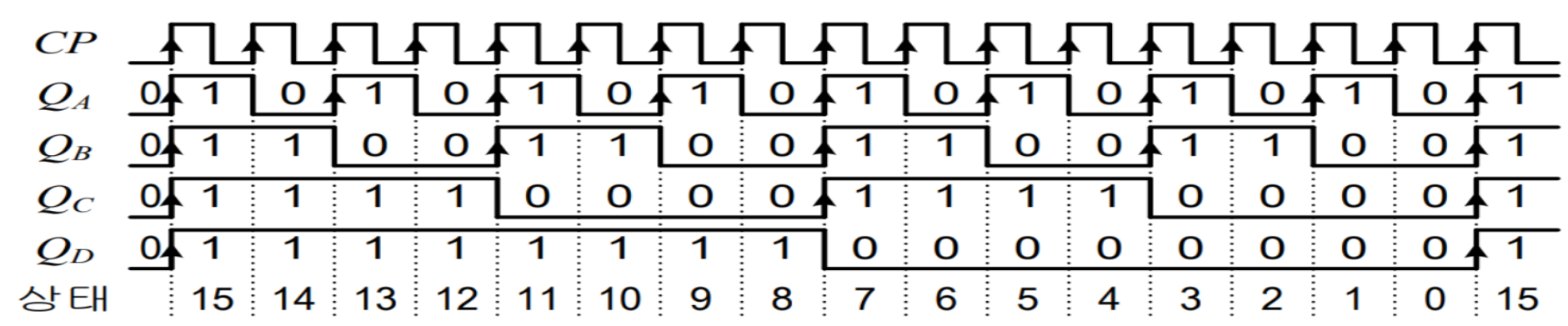
작동 방식 - 4비트 2진 비동기식 하향 카운터



- 회로에 not 게이트가 연결되어 있어 신호가 내려갈 때 플립플롭이 트리거됨
- 각 플립플롭들이 순서대로 하강 에지에서 동작하여 카운터 숫자가 증가함
- 첫 번째 플립플롭은 cp가 하강 에지를 만날 때 상태가 바뀜
- 두 번째 플립플롭은 첫 번째 플립플롭의 출력이 하강 에지를 만들 때 상태가 바뀜
- 세 번째, 네 번째 플립플롭도 이전 플립플롭의 출력이 하강 에지를 만들 때 상태가 바뀜

클럭펄스	Q _D	Q _C	Q _B	Q _A	10진수	클럭펄스	Q _D	Q _C	Q _B	Q _A	10진수
1	1	1	1	1	15	9	0	1	1	1	7
2	1	1	1	0	14	10	0	1	1	0	6
3	1	1	0	1	13	11	0	1	0	1	5
4	1	1	0	0	12	12	0	1	0	0	4
5	1	0	1	1	11	13	0	0	1	1	3
6	1	0	1	0	10	14	0	0	1	0	2
7	1	0	0	1	9	15	0	0	0	1	1
8	1	0	0	0	8	16	0	0	0	0	0

4비트 2진 비동기식 하향 카운터의 상태표



클럭 펄스에 따른 상태 변화

비동기식 Counter

장점과 단점

장점

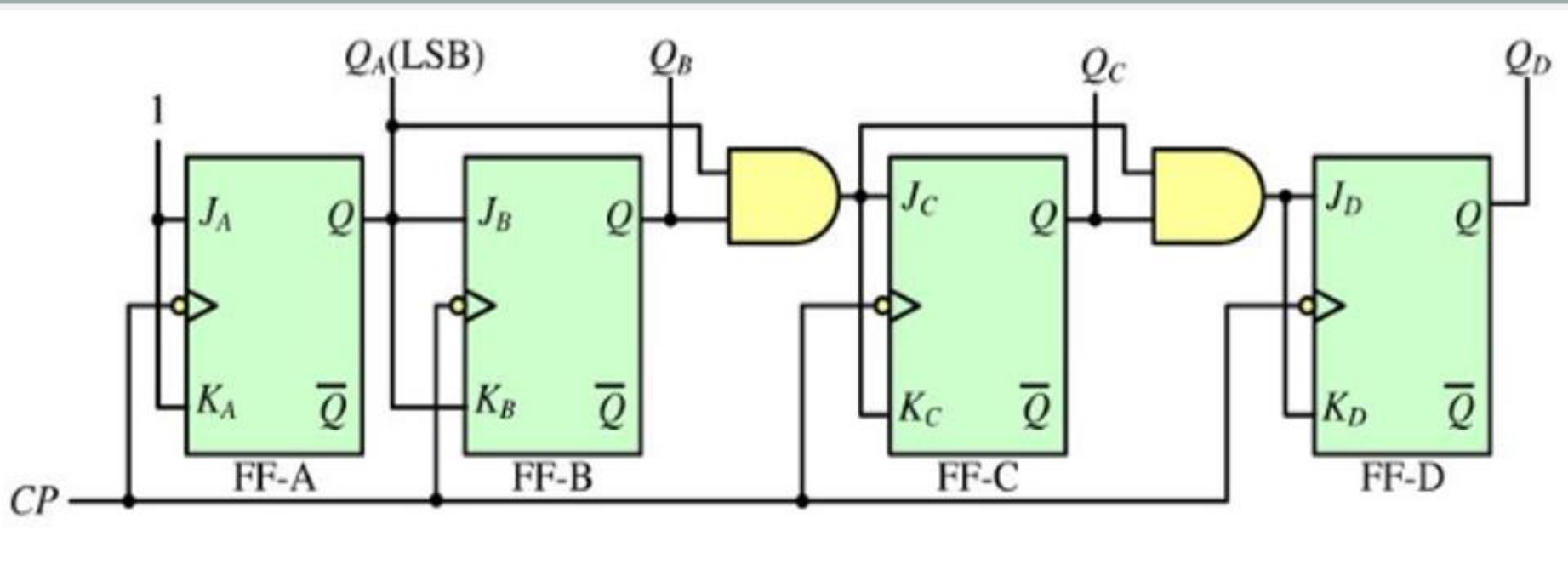
- 순차적 연결로 설계가 간단함
- 추가 연산이 필요 없어 전력 소비가 적음
- 비교적 단순한 설계로 회로의 크기가 작음

단점

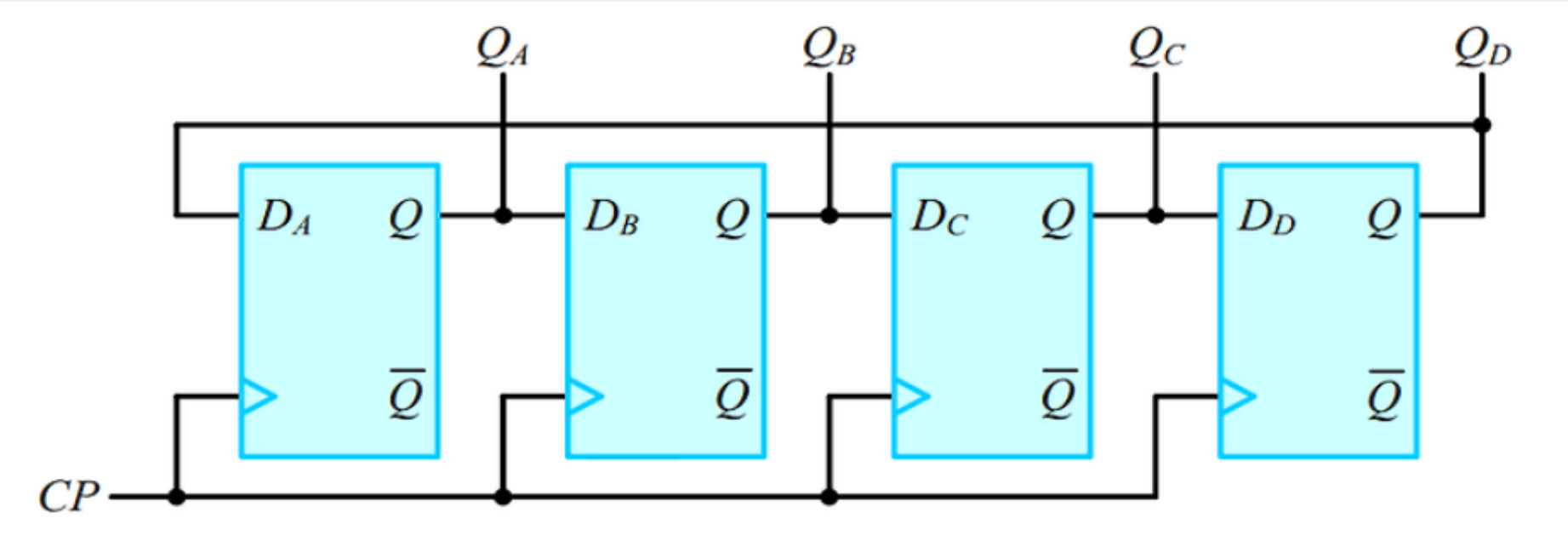
- 플립플롭이 순차적으로 작동하면서 전파 지연이 누적돼 속도가 느림
- 플립플롭의 순차적 작동으로 출력 신호의 타이밍이 불규칙해 질 수 있음

동기식 Counter

정의



- 4비트 2진 동기식 상향 카운터



- 4비트 링카운터

개념 Concept

입력 cp를 모든 플립플롭에 공통으로 인가하여
모든 플립플롭의 상태가 같은 시점에서 변화하는 카운터

특징 Characteristic

모든 플립플롭을 하나의 cp에 연결함
현재 상태에서 다음 상태를 요구하는 입력인 여기표가 필요함

여기표(Excitation table)

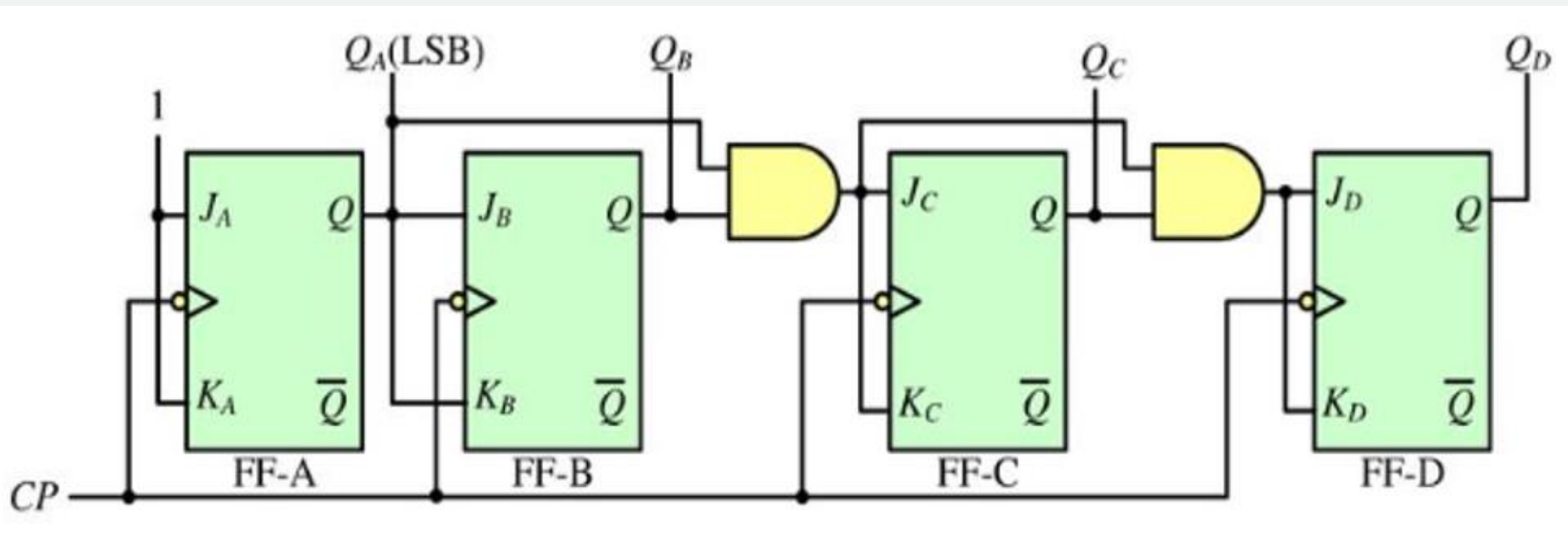
특성표			여기표		
입력	현재상태	차기상태	현재상태	차기상태	요구 입력
S R	$Q(t)$	$Q(t+1)$	$Q(t)$	$Q(t+1)$	S R
0 0	0	0	0	0	0 x
0 0	1	1	0	1	1 0
0 1	0	0	1	0	0 1
0 1	1	0	1	1	x 0
1 0	0	1			
1 0	1	1			
1 1	0	?			
1 1	1	?			

- SR 플립플롭의 여기표

- 현재 상태에서 다음 상태를 요구하는 입력
- 현재 상태와 입력 값이 주어졌을 때 다음 상태의 변화를 나타내는 특성표를 통해 작성
- 동기식 카운터를 설계할 때, 정해진 순서대로 플립플롭의 상태를 변화시키기 위해 설계하려는 카운터의 순서를 나열하고, 그에 대한 현재 상태와 다음 상태를 표시하여 해당 변화에 맞는 여기표를 적용시킴

동기식 Counter

작동 방식 - 4비트 2진 동기식 상향 카운터



클록 펄스	Q3	Q2	Q1	Q0	클록 펄스	Q3	Q2	Q1	Q0
0	0	0	0	0	8	1	0	0	0
1	0	0	0	1	9	1	0	0	1
2	0	0	1	0	10	1	0	1	0
3	0	0	1	1	11	1	0	1	1
4	0	1	0	0	12	1	1	0	0
5	0	1	0	1	13	1	1	0	1
6	0	1	1	0	14	1	1	1	0
7	0	1	1	1	15	1	1	1	1

- 모든 플립플롭이 하나의 cp에 연결되어 동작하도록 추가적인 연산이 필요함
- 하위 모든 출력이 1이 될 때 현재 플립플롭의 상태가 0에서 1, 1에서 0이 되기 위해 J와 K 입력이 모두 1이 되어야 함
- 플립플롭의 입력 함수는 하위 비트의 and 연산 결과가 됨
- 하위 비트가 모두 1이면 현재 플립플롭의 입력값을 모두 1로 만들어 다음 cp에서 비트를 반전시킴

동기식 Counter

장점과 단점

장점

- 모든 플립플롭이 하나의 cp에서 동작하므로 전파 지연이 적어 비동기식 카운터에 비해 빠름
- 클럭 신호에 동기화되어 작동하므로 신호가 왜곡되거나 비정상적인 타이밍 문제가 발생하지 않음

단점

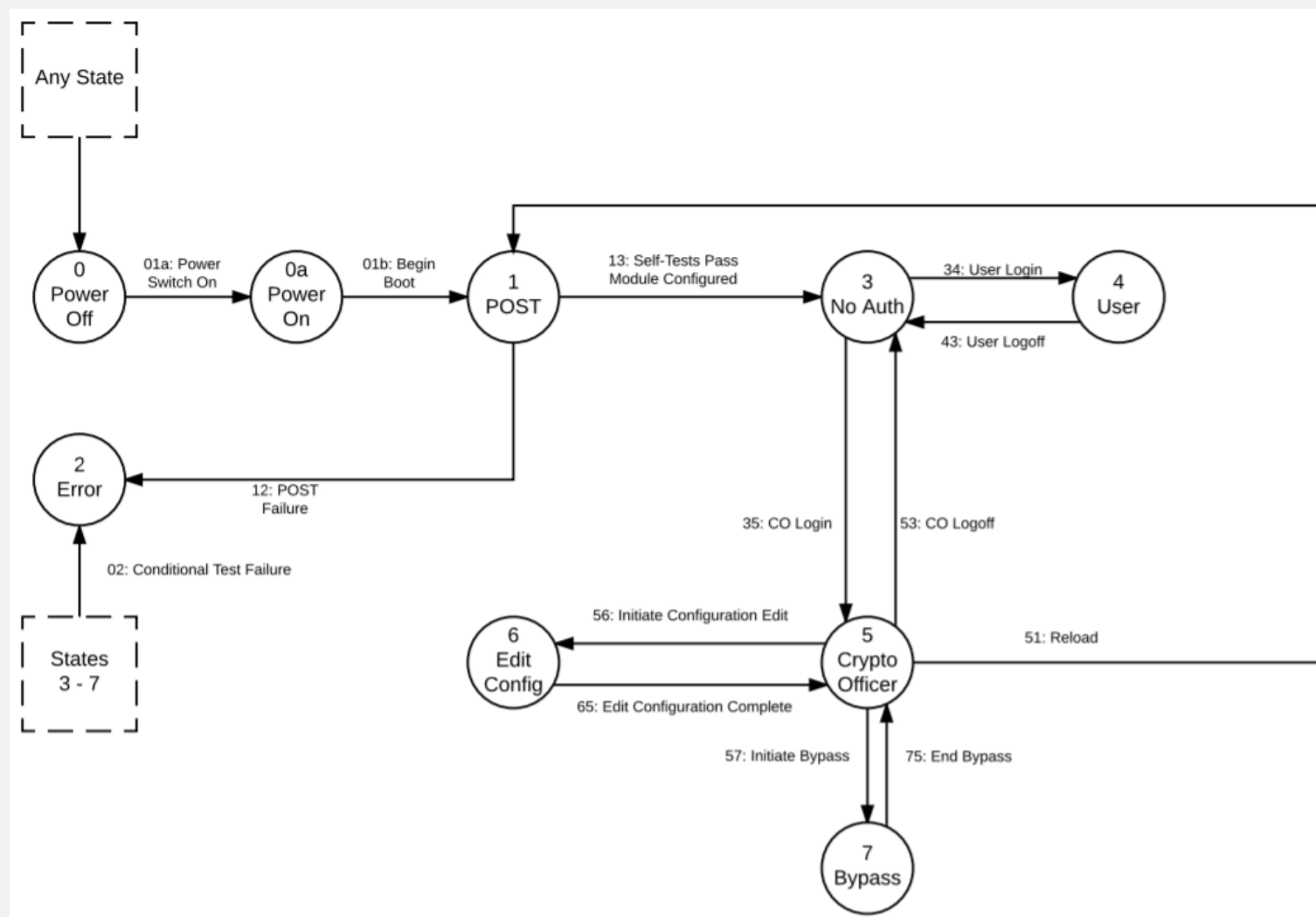
- 각 플립플롭의 입력을 제어하기 위해 추가적인 논리 게이트가 필요해져 회로 설계가 복잡해짐
- 비동기식 카운터에 비해 전력 소비가 많을 수 있음

FSM

Finitie-State-Machine

04

FSM 예시 사진



정의

- 유한한 수의 state를 가지는 컴퓨터의 이론적 모형
- FSM은 동시에 여러 state를 지닐 수 없다
-> 오직 한 번에 하나의 state를 가질 수 있다
- transition(전이) :
현재 state에 새로운 input이 들어오게 되면 다른 state로 변화 가능

구성

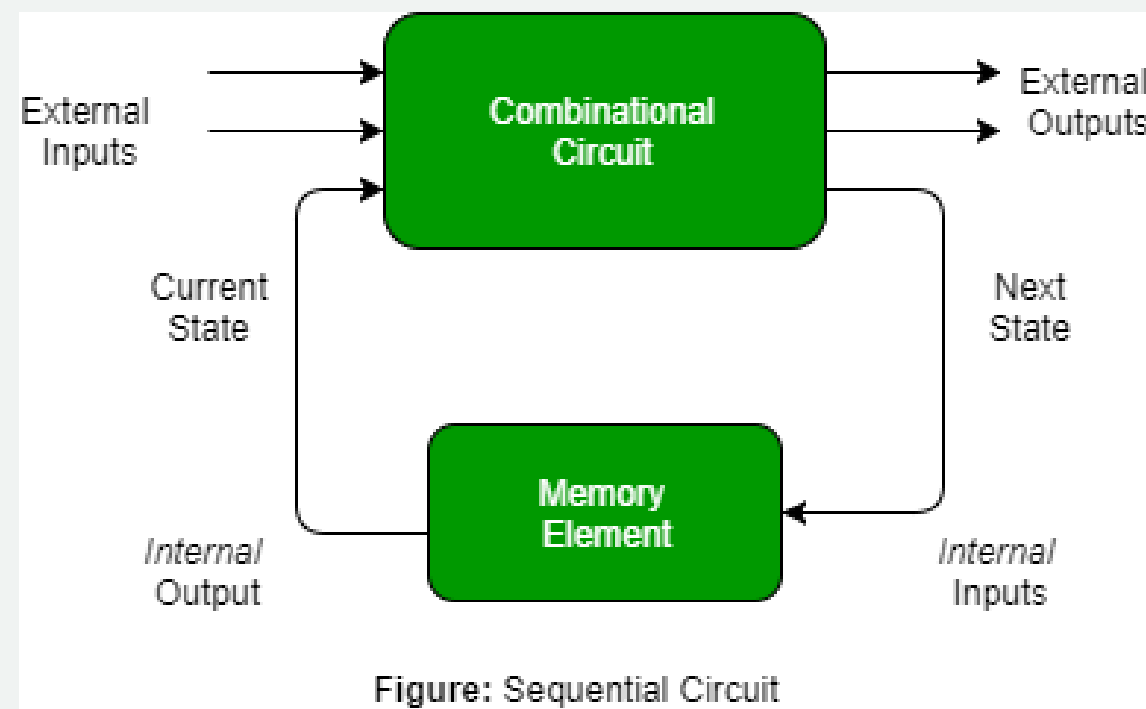
- state
=> N개의 bit가 존재
-> 2^N 개의 state 표현가능
eg) 3 bit 존재
-> 최대 8개 state 표현 가능
(000, 001, 010, 011,
100, 101, 110, 111)
- State가 되기 위한 조건들
eg) 000->001 조건은 input=1 이어야함

작동

1. Clock 발생
2. State transition 발생
3. 입력 + 현재 state -> 다음 state 정함
-> 이는 sequential circuit의 특성

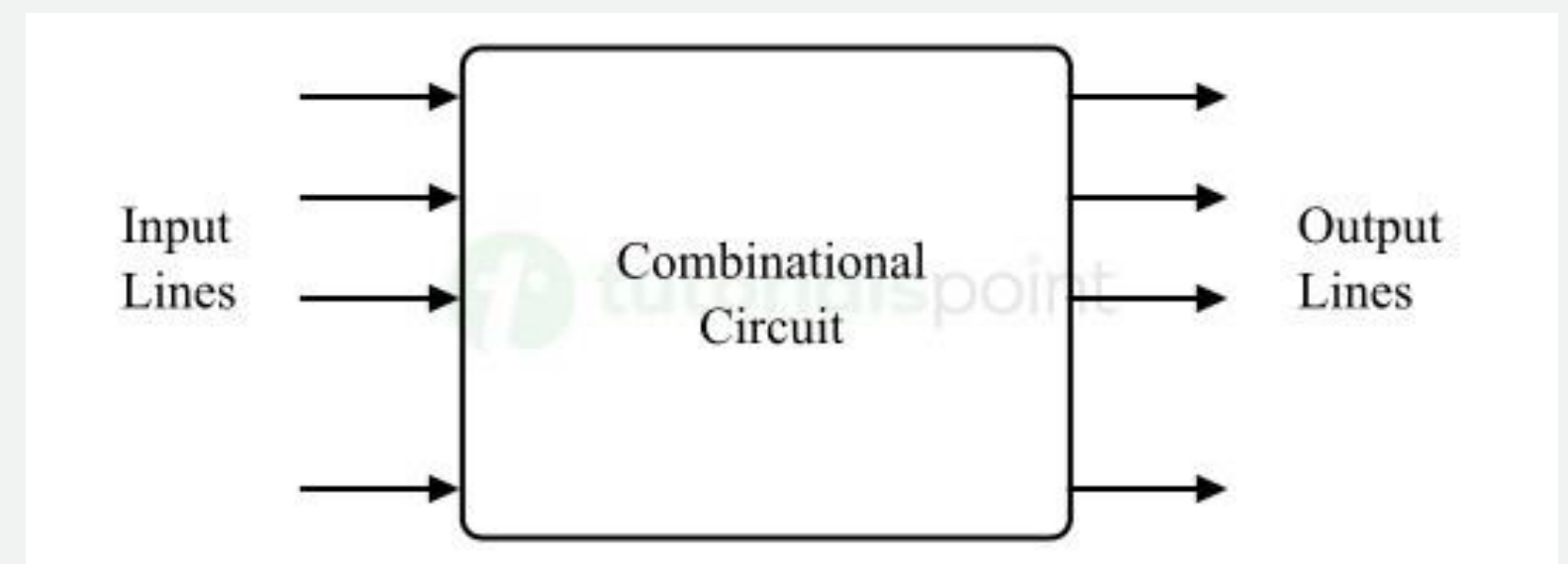
Sequential circuit

- 현재의 input + 과거의 history -> output
=> 과거의 history
= memory에 저장된 state



Combinational circuit

- 오직 현재의 input -> output



FSM

Finitie-State-Machine

04

Moore Machine

- 현재의 state => output
- Input을 받아서 state를 변화시키고 변화시킨 state를 통해 output을 도출함
-> 순차적으로 output이 도출됨.
eg) 111이 나오고 다음 turn에 output이 1

input	0	1	1	0	1	1	1	0	0	1	0	1	1	1	1	1	0	0		
state	?	A	B	C	A	B	C	D	A	A	B	A	B	C	D	D	D	A	A	?
output	?	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0

Mealy Machine

- 현재의 input + 현재의 state => output
- Input을 받아서 output을 내는 것과 state를 변화시키는 것은 병렬적으로 발생함.
eg) 111에서 마지막 1이 input으로 들어가자마자 바로 output이 1

input	0	1	1	0	1	1	1	0	0	1	0	1	0	1	1	1	1	1	0	0	
state	?	A	B	C	A	B	C	C	A	A	B	A	B	A	B	C	C	C	C	A	A
output	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	0

THANK YOU😊

20231591 이하은(33.3%)
20231579 이명진(33.3%)
20231523 김민정 (33.3%)