

# 디지털 시스템

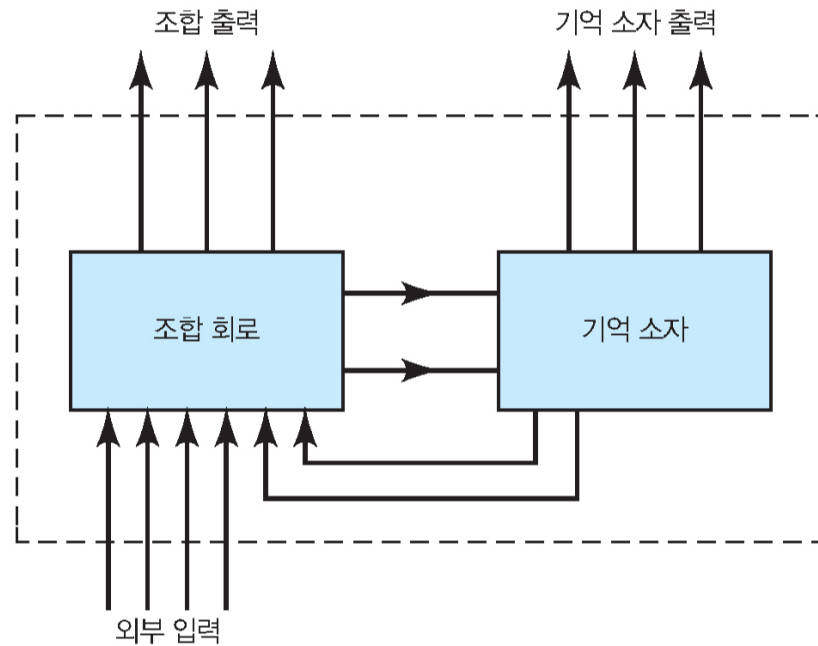
(CHAPTER 5. 플립플롭과 관련소자)

5

우 송 대 학 교  
철도전기시스템학과  
최 상 성

# Flip-Flop 개요

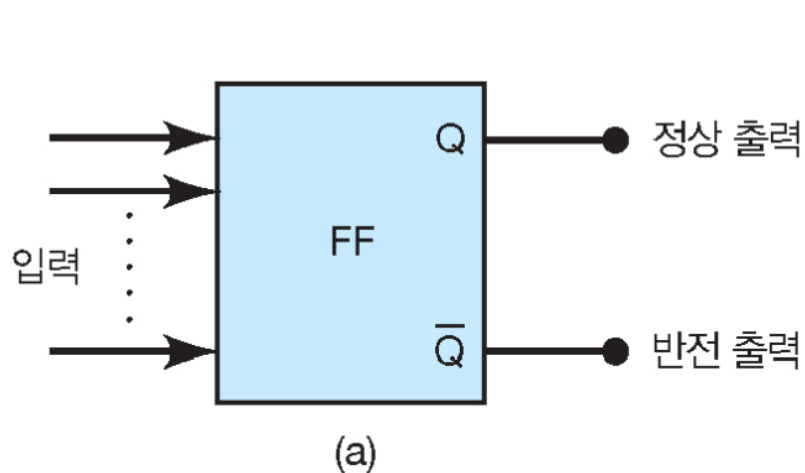
- **디지털 시스템은 조합 논리 회로와 기억 소자의 결합으로 구성**
  - 조합 회로는 외부입력과 기억소자의 출력을 입력 받아 여러 가지 출력 신호를 만든다
  - 조합회로의 출력 신호의 일부는 기억 소자에 저장되는 2진수 값을 결정하는데 사용
  - 기억 소자의 몇몇 출력들은 조합회로의 입력으로 피드백



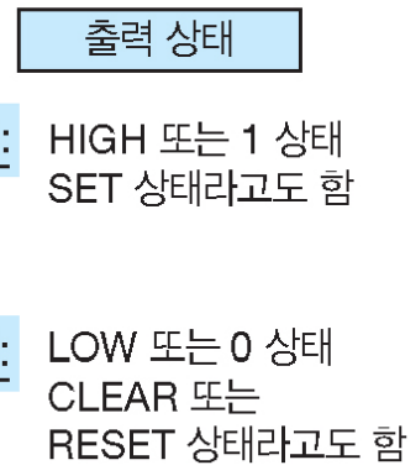
# Flip-Flop 개요 - 계속

## ➤ 플립플롭(Flip-Flop)

- 논리 게이트 자신은 저장 능력이 없으나 몇 개를 연결하면 정보 저장 가능
- 플립플롭은 논리 게이트 조합으로 피드백 개념을 적용해서 만들어 지는 기억 소자
- 가장 기본적인 플립플롭은 2개의 NAND 게이트 또는 NOR 게이트로 구성
- 래치(latch) 또는 쌍안정 멀티바이브레이터(Bi-Stable Multivibrator)



(a)

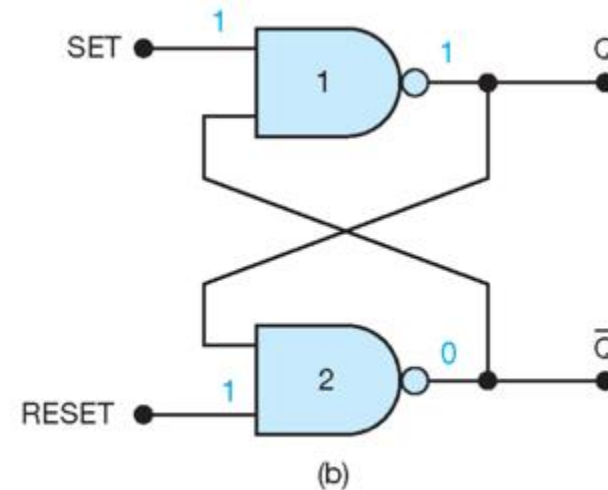
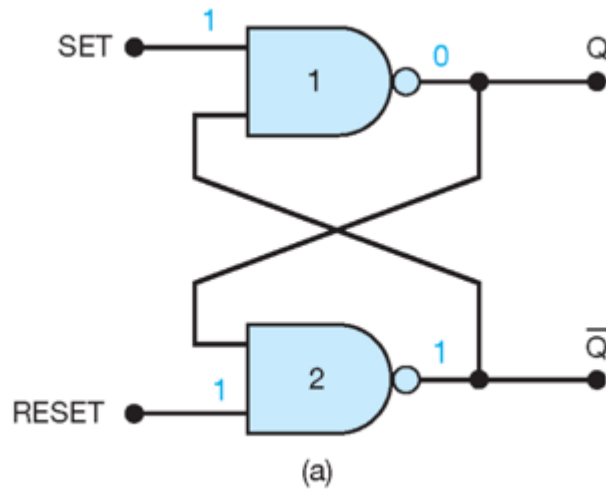


(b)

# 5-1 NAND 게이트 래치

## ➤ NAND 게이트를 이용한 래치(Latch)

- NAND-1의 출력이 NAND-2의 입력 중 하나에 연결
- NAND-2의 출력은 NAND-1의 입력 중 하나에 연결

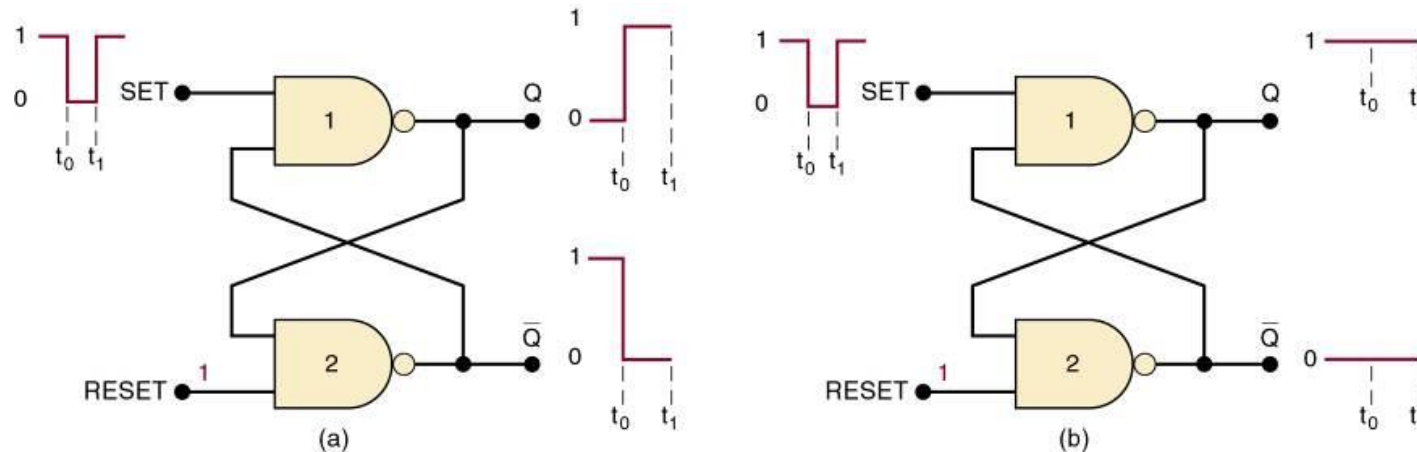


NAND 게이트 래치에서 SET와 RESET 입력들은 보통 HIGH 상태를 유지하고 있고, 래치 출력들을 변화 시키고자 할 때에는 두 입력중 하나에 LOW 펄스를 인가 한다

# 5-1 NAND 게이트 래치

## ➤ 래치(FF)의 세트

- RESET 입력이 HIGH로 유지 하면서, SET 입력에 LOW 펄스를 인가할 경우
  - (a) SET에 펄스가 인가 되기 전 Q=0인 경우
  - (b) SET에 펄스가 인가 되기 전 Q=1인 경우

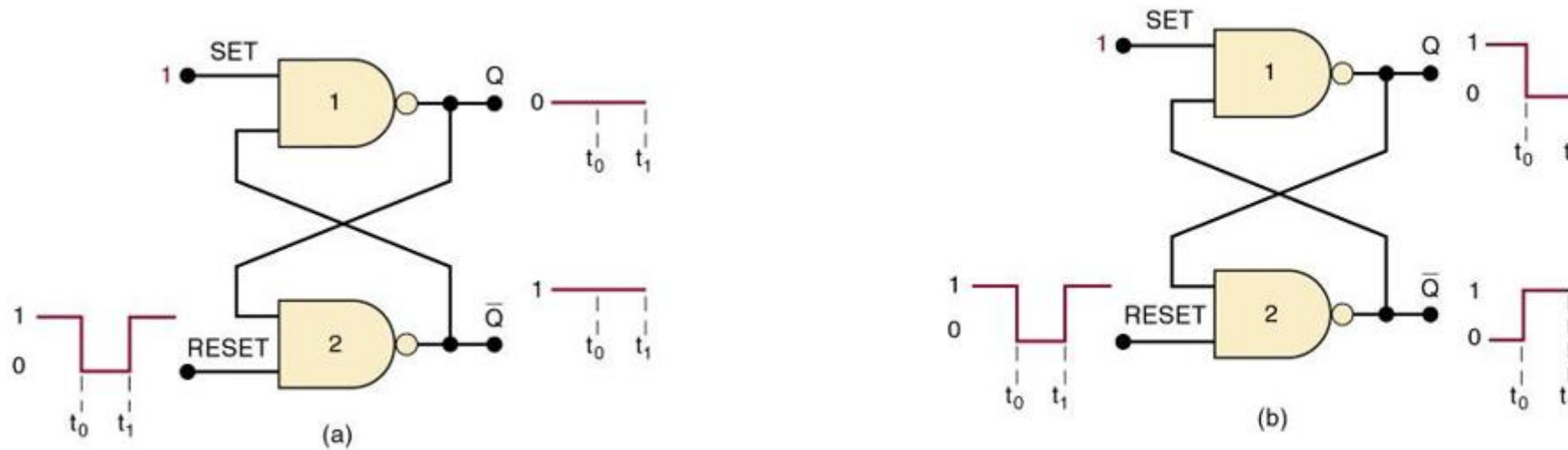


두 경우 모두 SET 입력이 원래 상태로 돌아와도 Q는 HIGH 상태를 유지  
→ SET 입력에 LOW 펄스를 인가 하면 항상 래치는 Q=1인 상태가 된다  
→ 이러한 동작을 래치의 세트(set)라고 한다

# 5-1 NAND 게이트 래치

## ➤ 래치(FF)의 리세트

- SET 입력이 HIGH로 유지 하면서 RESET 입력에 LOW 펄스를 인가할 경우
  - (a) RESET에 펄스가 인가 되기 전 Q=0인 경우
  - (b) RESET에 펄스가 인가 되기 전 Q=1인 경우



두 경우 모두 SET 입력이 원래 상태로 돌아와도 Q는 LOW 상태를 유지  
→ RESET 입력에 LOW 펄스를 인가 하면 항상 래치는 Q=0인 상태가 된다  
→ 이러한 동작을 래치의 클리어(clear) 또는 리세트(reset)라고 한다

# 5-1 NAND 게이트 래치

## ➤ 동시적인 세트와 리세트

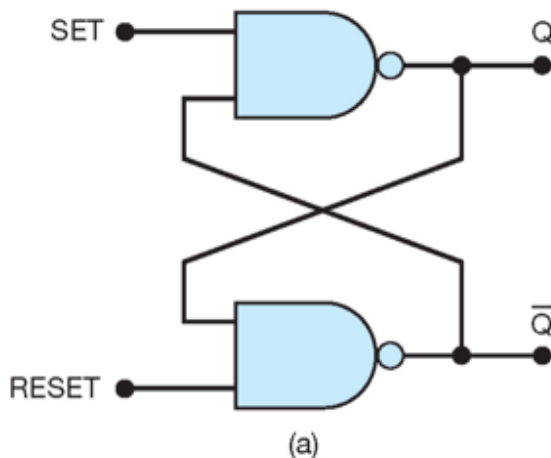
- SET와 RESET 입력에 동시에 LOW 펄스를 인가할 경우
  - 2개의 NAND 출력이 HIGH( $Q=\bar{Q}=1$ )인 경우
    - 2개의 출력이 서로 반대가 되어야 한다는 가정에 어긋나므로 원하지 않는 상태
    - SET와 RESET 입력이 다시 HIGH가 될 때 출력 값 예측 불능

**SET=RESET=0 인 조건은 일반적으로 NAND 래치에서는 사용되지 않는다**

# 5-1 NAND 게이트 래치

## ➤ NAND 래치의 요약

1. **SET=RESET=1인 경우** 부동 상태(resting state)로 출력 상태에 아무런 영향을 미치지 않는다. Q와  $\bar{Q}$  출력은 입력이 들어가기 전 상태를 유지 한다
2. **SET=0, RESET=1인 경우** 출력을 항상 Q=1인 상태로 만들며, 다시 SET 입력이 HIGH가 되어도 출력은 변하지 않는다. 이것을 **래치의 세트** 상태라 한다
3. **SET=1, RESET=0인 경우** 출력을 항상 Q=0인 상태로 만들며, RESET 입력이 다시 HIGH로 변하여도 출력은 변하지 않는다. 이것을 **래치의 클리어** 또는 **리세트**라고 한다
4. **SET=RESET=0인 조건**은 래치의 세트와 리세트를 동시에 하는 것으로 Q= $\bar{Q}$ =1의 출력을 낸다. 입력이 동시에 1로 돌아 가면 출력 값을 예측할 수 없어 이 조건은 사용하지 않는다.



Set	Reset	출력
1	1	변화없음
0	1	Q = 1
1	0	Q = 0
0	0	사용 안됨*

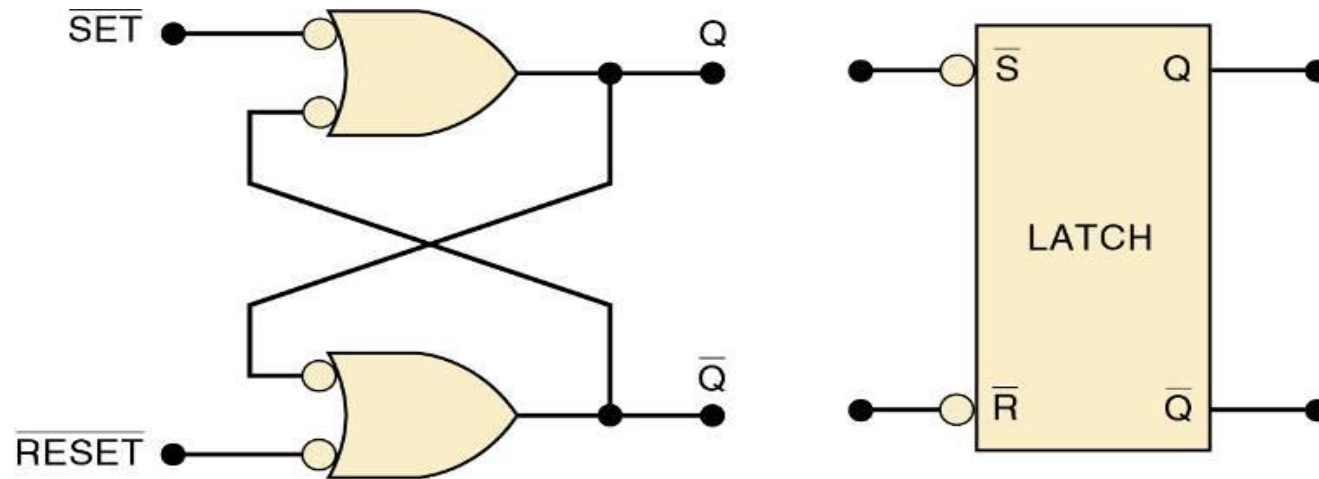
\*Q =  $\bar{Q}$  = 1을 생성

(b)

# 5-1 NAND 게이트 래치

## ➤ 등가 표현

- SET가 LOW 상태로 될 때  $Q=1$ , RESET가 LOW 상태로 될 때  $Q=0$
- 입력에 표시된 방울과 입력신호 이름  $\overline{SET}$ ,  $\overline{RESET}$ 는 이들 입력이 LOW에서 활성화됨을 나타낸다

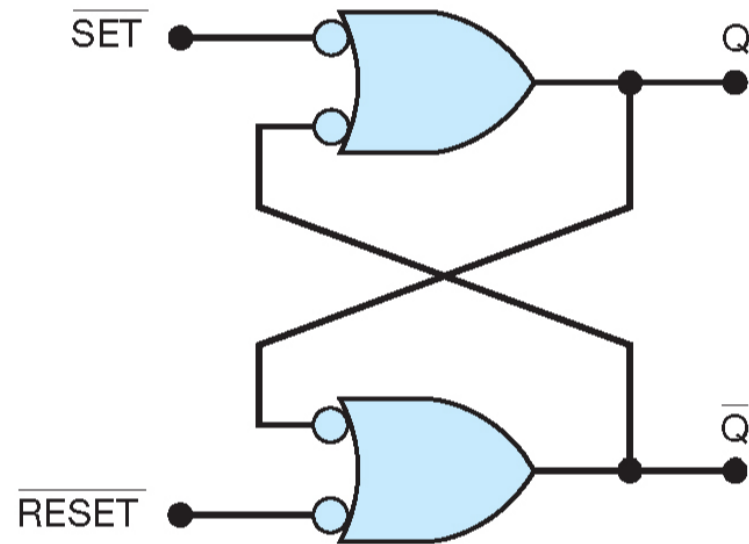
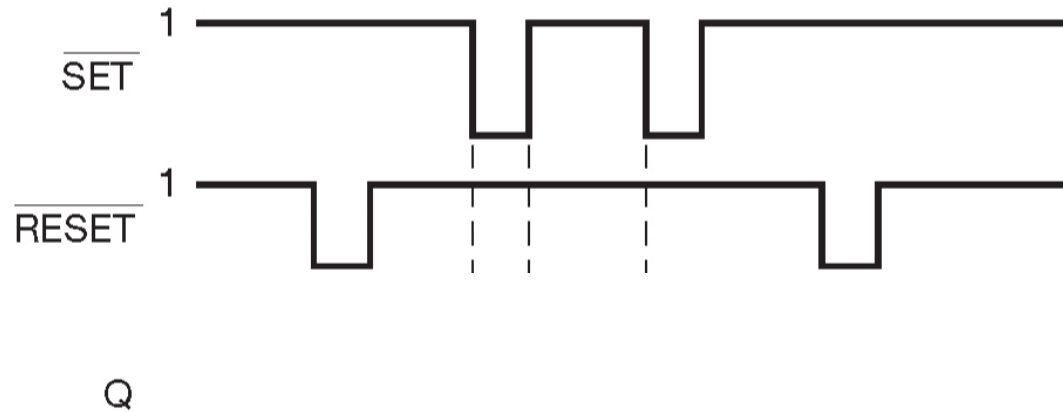


NAND latch equivalent representations and simplified block diagram

# 5-1 NAND 게이트 래치

## ➤ 예제 5-1

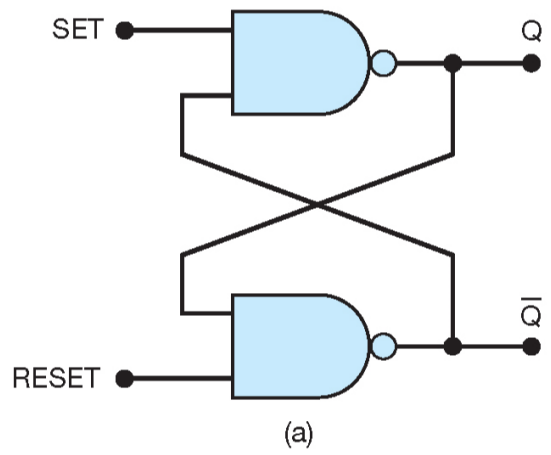
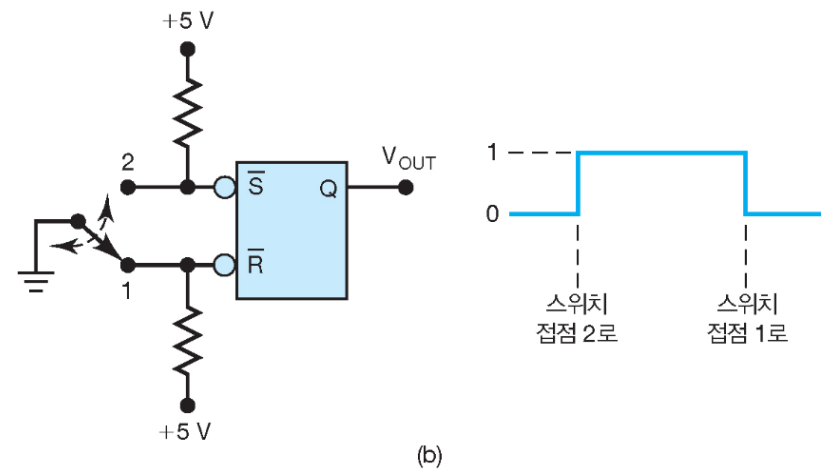
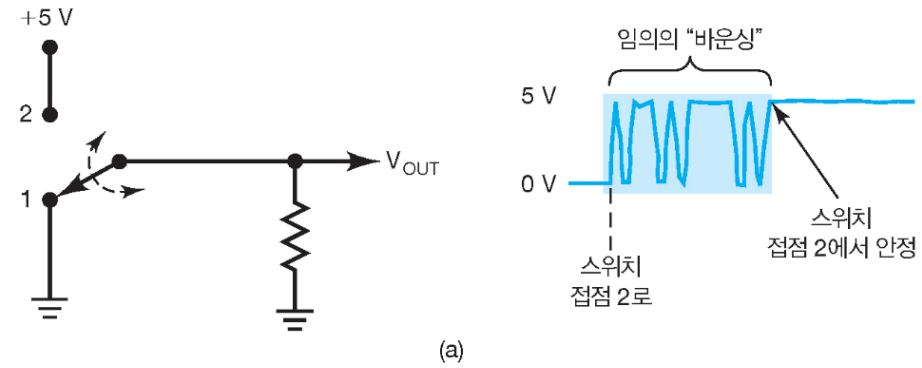
그림 5-8의 파형이 그림 5-7의 래치의 입력으로 인가 되었다. 초기값을  $Q=0$ 으로 가정하고  $Q$ 의 파형을 구하라



# 5-1 NAND 게이트 래치

## 예제 5-2

그림 5-9(b) 스위치 디바운싱(switch debouncing) 회로의 동작에 대해 설명하라



Set	Reset	출력
1	1	변화없음
0	1	Q = 1
1	0	Q = 0
0	0	사용 안됨*

\*Q = Q-bar = 1을 생성

(b)

# 5-1 NAND 게이트 래치

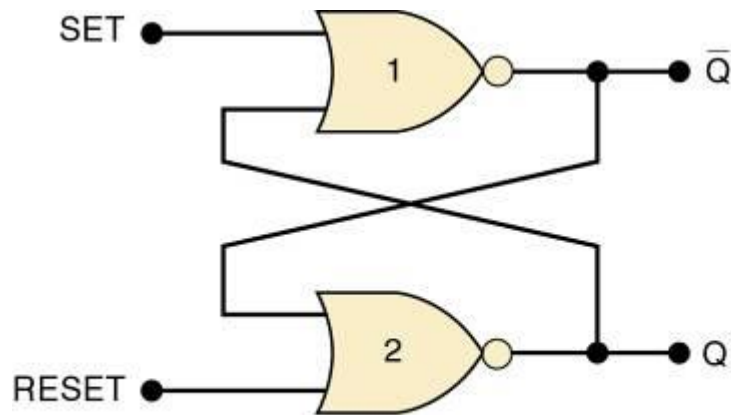
## ➤ 학습성과 평가문제

1. 래치의 SET와 RESET 입력의 평소 상태는 무엇인가? 각 입력의 활성화 상태는 무엇인가?
2. 플립플롭이 리세트(클리어)된 후 Q와  $\bar{Q}$ 의 상태는 무엇인가?
3. 참 또는 거짓: SET 입력은 결코 Q=0로 만들 수 없다
4. 어떤 플립플롭에 전원을 처음 인가 하였을 때 출력 Q와  $\bar{Q}$ 의 초기 상태를 예측하는 것은 불가능하다. NAND 래치가 항상 Q=1의 초기 상태에서 동작 하도록 하기 위한 방법은 무엇인가?

# 5-2 NOR 게이트 래치

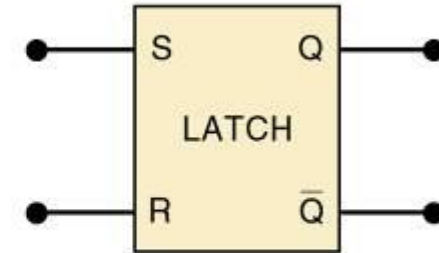
## ➤ NOR 게이트를 이용한 래치(Latch)

- NOR-1의 출력이 NOR-2의 입력 중 하나에 연결
- NOR-2의 출력은 NOR-1의 입력 중 하나에 연결



Set	Reset	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

\*Produces  $Q = \bar{Q} = 0$ .

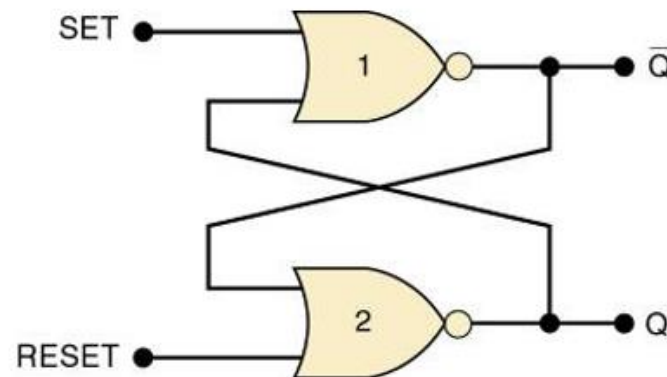


NOR 게이트 래치에서 SET와 RESET 입력들은 보통 LOW 상태를 유지하고 있고, 래치 출력들을 변화 시키고자 할 때에는 두 입력중 하나에 HIGH 펄스를 인가 한다

# 5-2 NOR 게이트 래치

## ➤ NOR 래치의 요약

1. **SET=RESET=0인 경우** 부동 상태(resting state)로 출력 상태에 아무런 영향을 미치지 않는다. Q와  $\bar{Q}$  출력은 입력이 들어가기 전 상태를 유지 한다
2. **SET=1, RESET=0 인 경우** 출력을 항상 Q=1인 상태로 만들며, 다시 SET 입력이 LOW가 되어도 그대로 그 값을 유지한다
3. **SET= 0, RESET=1인 경우** 출력을 항상 Q=0인 상태로 만들며, RESET 입력이 다시 LOW로 변하여도 그 값을 그대로 유지한다
4. **SET=RESET=1인 조건은** 동시에 래치를 세트와 리셋을 하게 되고 Q= $\bar{Q}$ =0을 만약 입력들이 발생 시킨다. 동시에 0으로 돌아오면, 그 결과 출력 상태는 예측할 수 없다. 그러므로 이러한 입력 조건은 사용하지 않는다.



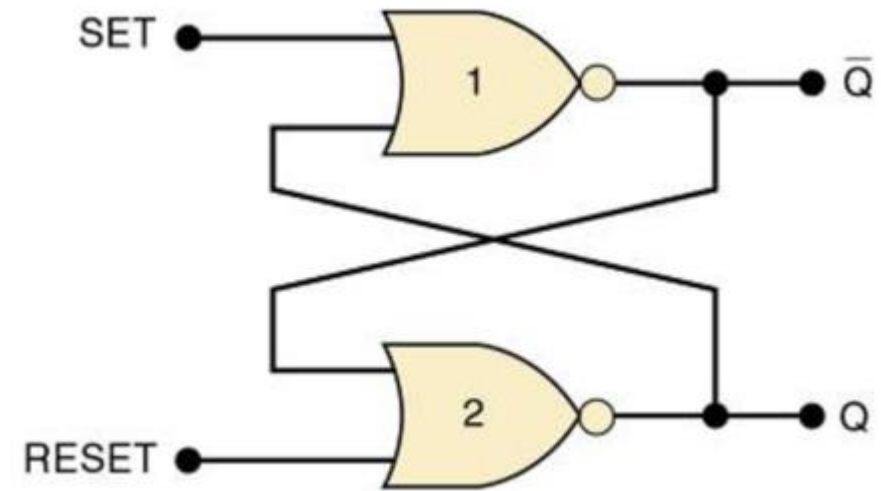
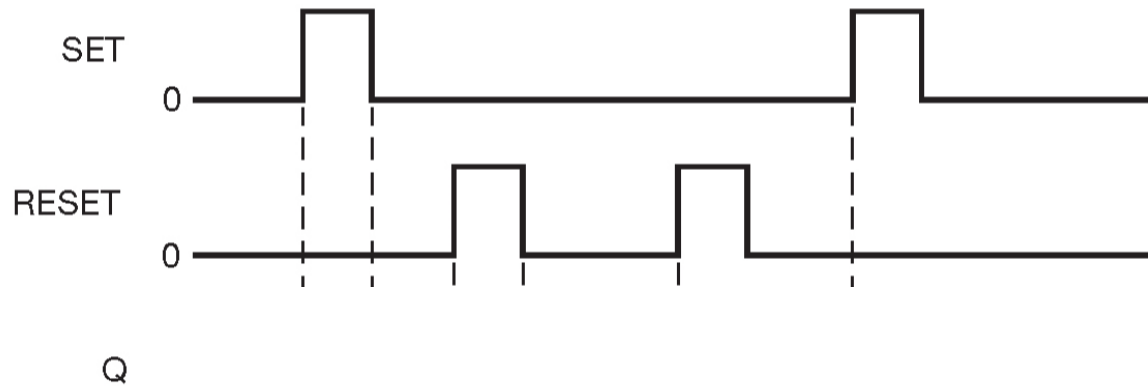
Set	Reset	Output
0	0	No change
1	0	Q = 1
0	1	Q = 0
1	1	Invalid*

\*Produces Q =  $\bar{Q}$  = 0.

# 5-2 NOR 게이트 래치

## 예제 5-3

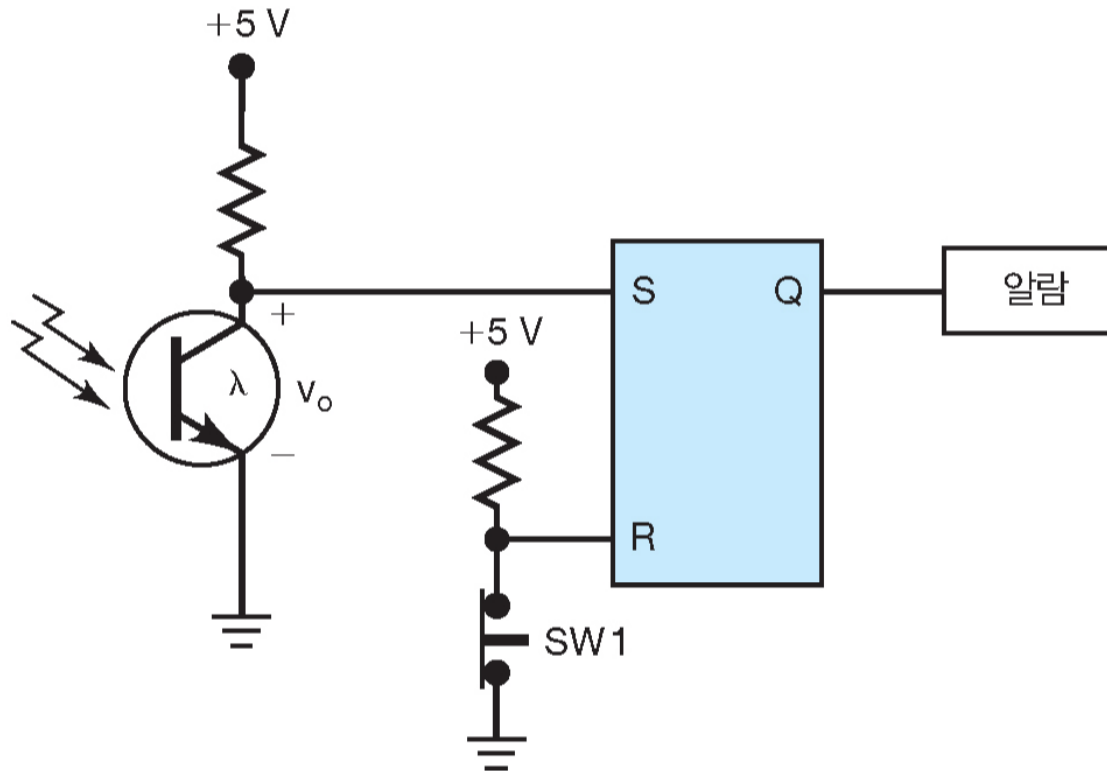
초기값을  $Q=0$ 으로 가정하고 그림 5-11의 NOR 래치 입력에 대한 Q의 파형을 구하라



## 5-2 NOR 게이트 래치

### ➤ 예제 5-4

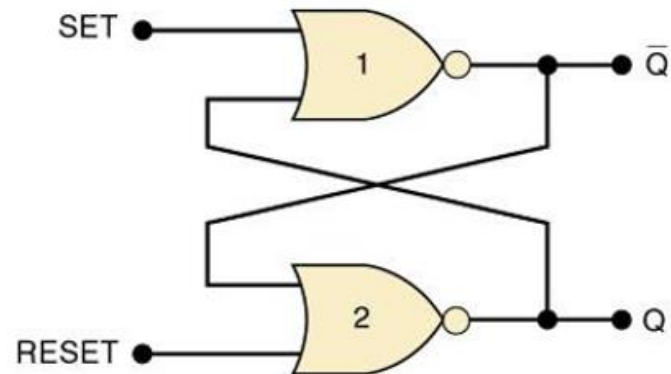
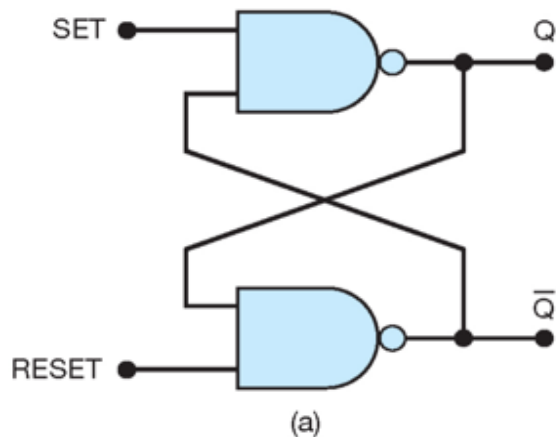
SW1을 잠시 열어서 래치의 초기 상태를 0으로 만들었다고 하자. 이때 만약 빛이 잠시 차단된다면 어떤 현상이 일어나는가?



## 5-2 NOR 게이트 래치

### ➤ 전원을 인가하였을 때 플립플롭의 상태

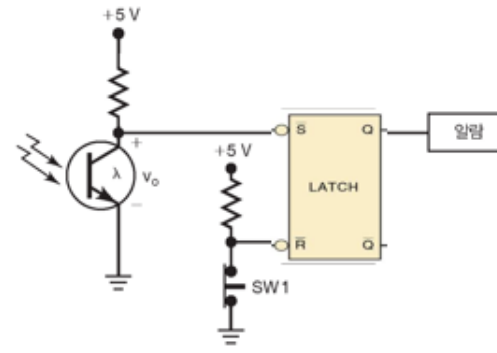
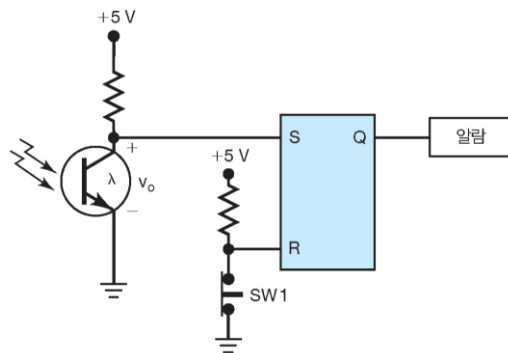
- 회로에 전압을 인가 하였을 때 SET와 RESET 입력이 비활성 상태에 있다면 플립플롭 출력의 초기 상태를 예측할 수 없다
  - NAND 래치의 경우  $S=R=1$  일 때 비활성화
  - NOR 래치의 경우  $S=R=0$  일 때 비활성화
- 플립플롭 회로의 적절한 동작을 시작하기 위해서 초기 상태를 만들어야 한다면 회로 동작 초기에 **순간적으로 SET나 RESET 입력을 활성화** 시켜 그 상태를 맞추어야 한다
  - 이것은 입력에 적당한 펄스를 인가하므로 구현 된다



# 5-2 NOR 게이트 래치

## ➤ 학습성과 평가문제

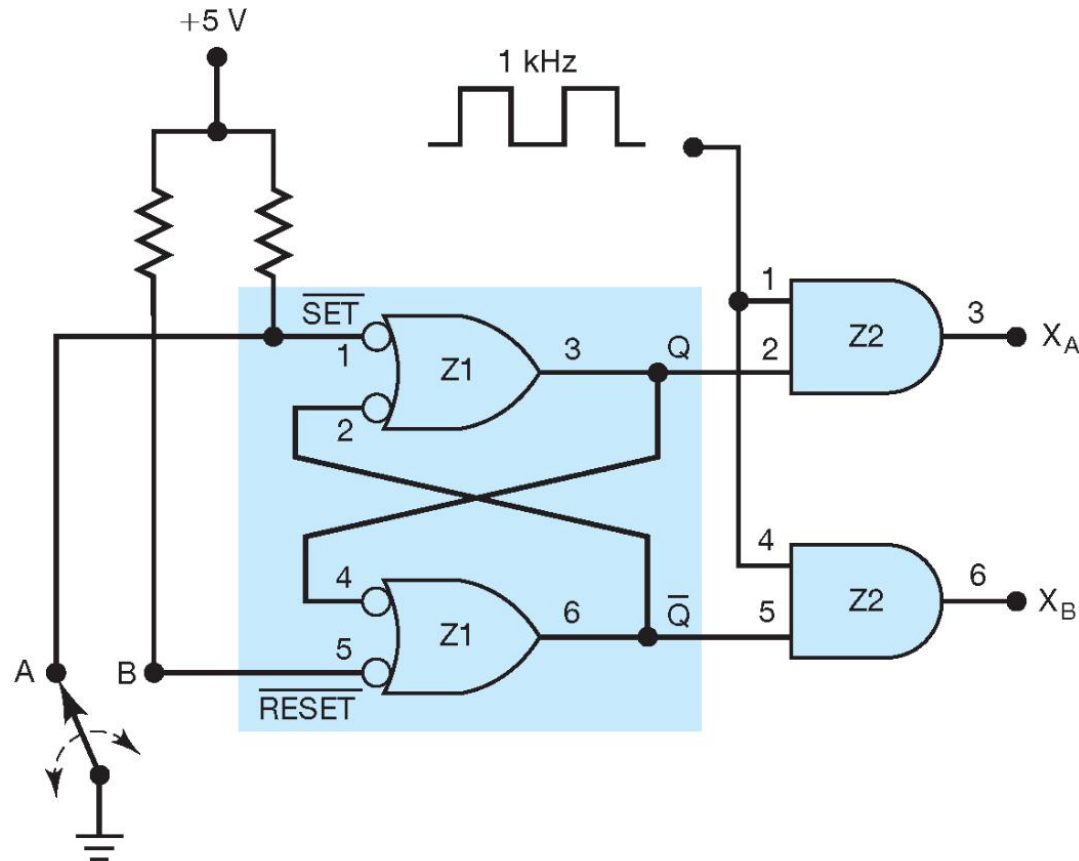
1. NOR 래치 입력의 부동 상태는 무엇인가? 그리고 활성 상태는 무엇인가?
2. 래치가 세트 상태로 되면 Q와  $\bar{Q}$ 의 상태는 무엇인가?
3. NOR 래치에서 출력 Q를 1에서 0으로 변화 시키는 방법은 무엇인가?
4. 그림 5-12의 NOR 래치가 NAND 래치로 바뀌었을 때 회로가 잘 동작하지 않는 이유는 무엇인가?



# 5-3 고장 진단 연구

## ➤ 예제 5-5

아래 회로의 동작을 기술하고 분석하라



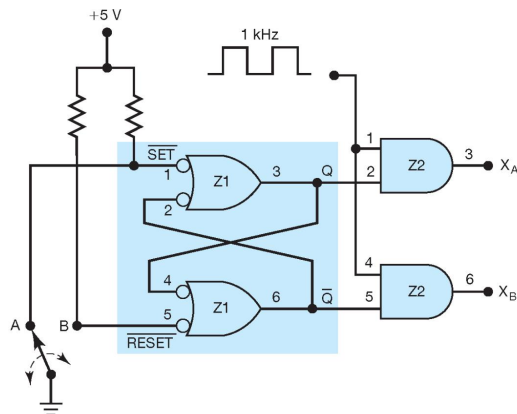
스위치 위치	$X_A$	$X_B$
A	펄스	LOW
B	LOW	펄스

# 5-3 고장 진단 연구

## 예제 5-6

스위치가 접점 B에 있을 때에는 정확히 동작  
 스위치가 접점 A에 있을 때에는 래치 Q=1 상태로 세트 되지 않았음을 인지  
 이러한 오작동을 일으키는 원인은 무엇인가?

Switch Position	SET (Z1-1)	RESET (Z1-5)	Q (Z1-3)	$\bar{Q}$ (Z1-6)	$X_A$ (Z2-3)	$X_B$ (Z2-6)
A	LOW	HIGH	LOW	HIGH	LOW	Pulses
B	HIGH	LOW	LOW	HIGH	LOW	Pulses

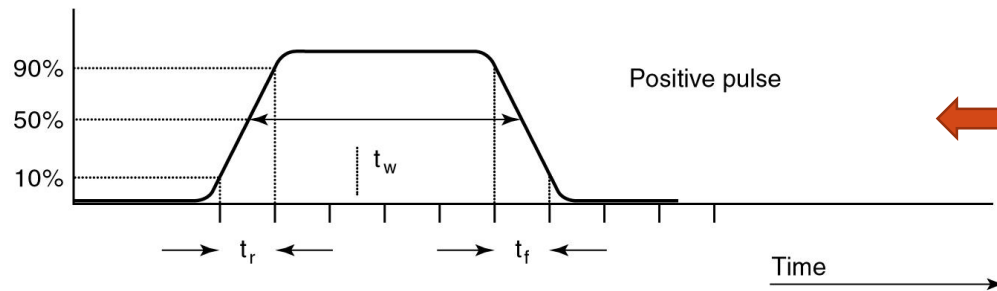


스위치 위치	$X_A$	$X_B$
A	펄스	LOW 펄스
B	LOW	펄스

# 5-4 디지털 펄스

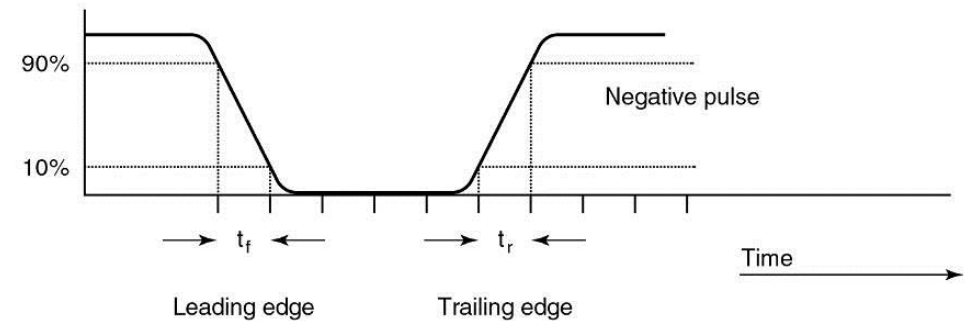
## ➤ 펄스 파형(Pulse Waveform)

- 양 펄스(Positive Pulse)
- 음 펄스(Negative Pulse)



← HIGH로 변할 때 원하는 기능을 수행하는 펄스

LOW로 변할 때 원하는 기능을 수행하는 펄스 →

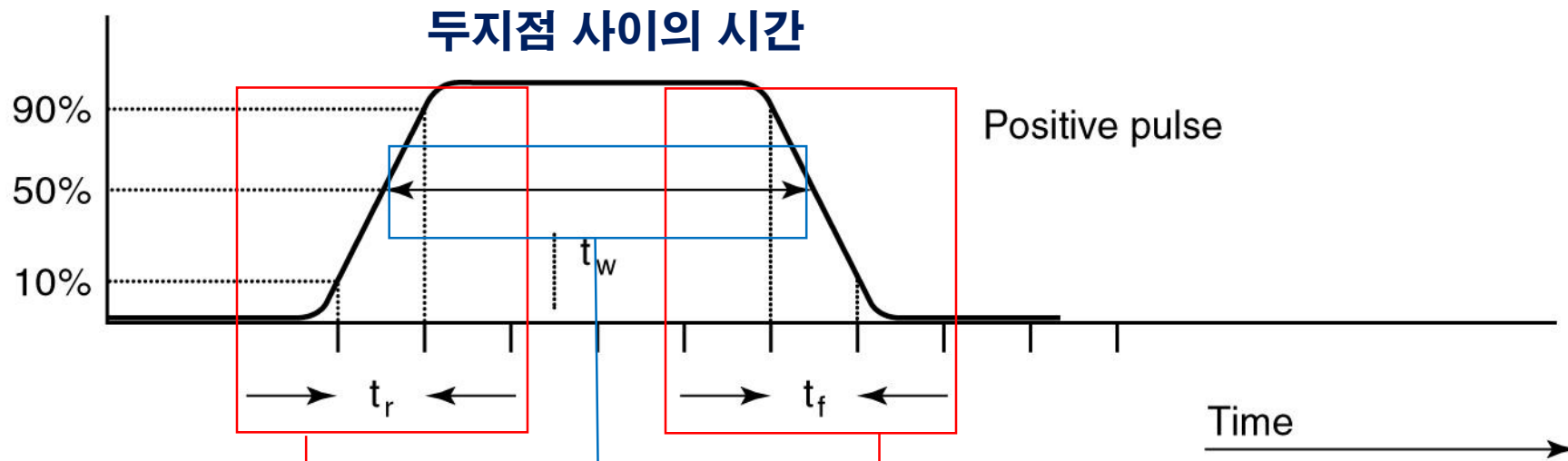


활성 상태와 비활성 상태 사이를 전환하는 신호를 펄스 파형이라 부른다

# 5-4 디지털 펄스

## ➤ 천이 시간(Transition Time)

- 상승시간(rise time) : LOW에서 HIGH로 천이 되는 시간( $T_r$ )
- 하강시간(fall time): HIGH에서 LOW로 천이 되는 시간( $T_f$ )
- 펄스 유지시간(펄스폭:  $t_w$ ): 리딩 에지와 트레일링 에지가 HIGH 레벨 전압의 50%가 되는



Measured between the 10% and 90% points on the *leading* edge of the voltage waveform.

The time between the points when the leading and trailing edges are at 50% of the HIGH level voltage.

Measured between the 10% and 90% points on the *trailing* edge of the voltage waveform.

## 5-4 디지털 펄스

### ➤ 예제 5-7

$\overline{RD}$  펄스는 LOW 활성 펄스, 펄스폭  $t_w=50\text{ns}$ , 상승시간  $t_r=15\text{ns}$ , 하강시간  $t_f=10\text{ns}$   
스케일에 맞춰 펄스를 그려라

## 5-4 디지털 펄스

### ➤ 학습성과 평가문제

1. 다음을 정의해 보라. 상승시간, 하강시간, 상승에지 하강에지, 리딩 에지, 트레일 에지, Positive 펄스, Negative 펄스, 펄스폭
2. 펄스폭은 어디에서 측정하는가?
3. 상승시간은 어디에서 측정하는가?
4. 하강시간은 어디에서 측정하는가?

# 5-5 클럭 신호와 클럭 입력을 갖는 플립플롭

## ➤ Synchronous/Asynchronous System

### ▪ Asynchronous system(비동기 시스템)

- 시스템 출력은 입력이 변경 될 때마다 상태를 변경할 수 있다
- 동기 시스템에 비해 설계나 고장 진단이 어렵다

### ▪ Synchronous System(동기 시스템):

- 시스템 출력은 클럭 사이클의 특정 시간에만 상태를 변경할 수 있다
- 설계나 고장 진단이 쉬워 대부분의 디지털 시스템은 동기 시스템이다

## ➤ Clock 신호

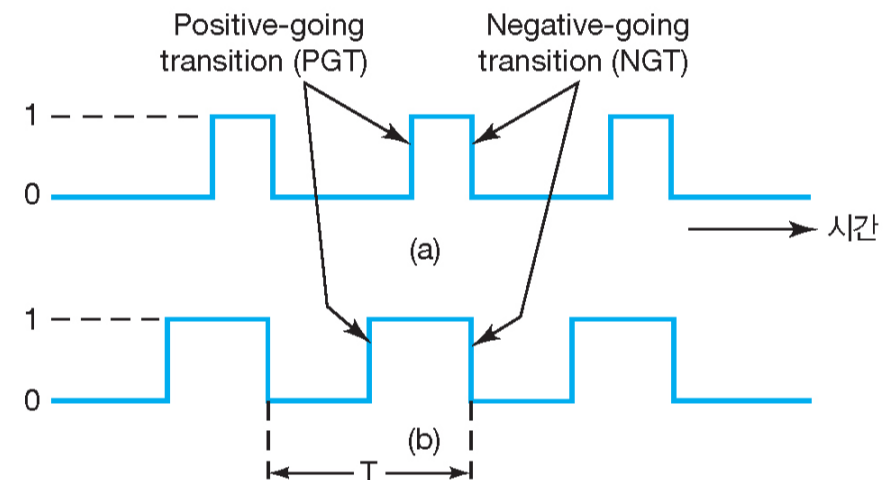
### ▪ Positive-Going Transition(PGT)

- 클럭이 0에서 1로 변할때

### ▪ Negative-Going Transition(NGT)

- 클럭이 1에서 0으로 변할 때

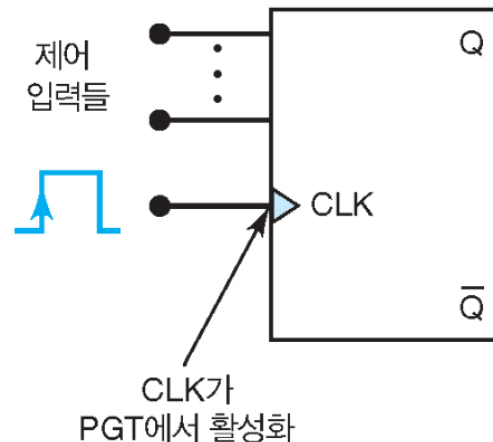
클럭 신호의 동기는 클럭의 두가지 천이 중 하나에서 그 상태를 변화시키는 클럭 입력을 갖는 플립플롭을 이용하여 구현할 수 있다



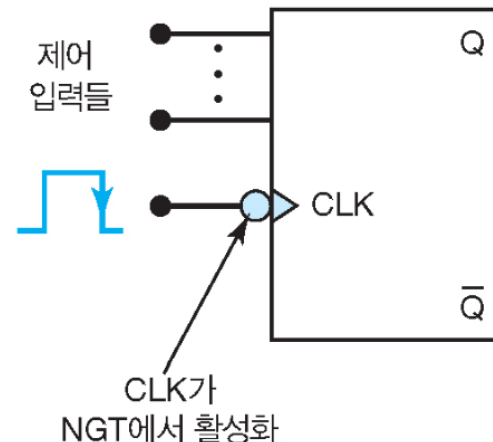
# 5-5 클럭 신호와 클럭 입력을 갖는 플립플롭

## ➤ 클럭 입력을 갖는 플립플롭(Clocked Flip-Flop)

- CLK, CK 또는 CP라고 명명된 클럭 입력을 가진다
- 하나 이상의 제어 입력(control input)을 가지며, 기능에 따라 여러 가지 이름을 갖는다
  - CLK 입력에 인가된 신호에 플립플롭이 동기 된다
    - 동기 제어 입력(Synchronous Control Input)
- 제어 입력은 플립플롭의 출력을 변화할 준비를 시키고 실제 트리거 변화는 CLK 입력의 활성 천이에 의해 가동 된다



(a)

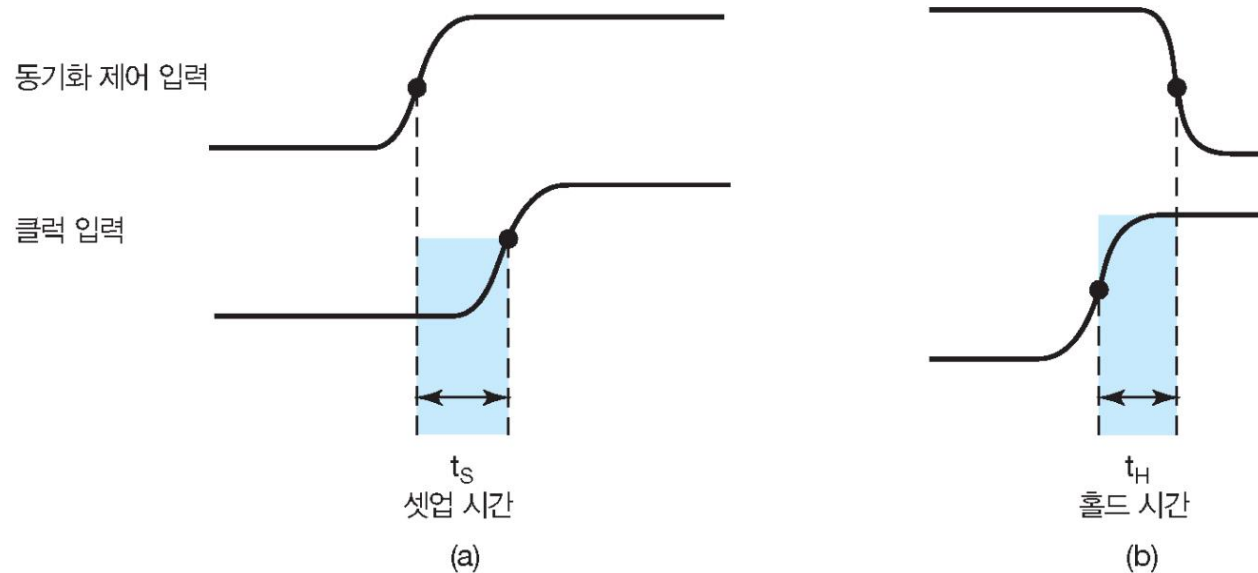


(b)

# 5-5 클럭 신호와 클럭 입력을 갖는 플립플롭

## ➤ 셋업(setup) 과 홀드(hold) 시간

- **셋업 시간( $t_s$ )** : CLK 신호의 활성 천이가 발생하기 전 제어 입력이 일정한 레벨을 유지해 주어야 하는 시간
- **홀드 시간( $t_H$ )** : CLK 신호의 활성 천이가 발생한 다음 제어 입력이 일정한 레벨을 유지해 주어야 하는 시간



# 5-5 클럭 신호와 클럭 입력을 갖는 플립플롭

## ➤ 셋업(setup) 과 홀드(hold) 시간

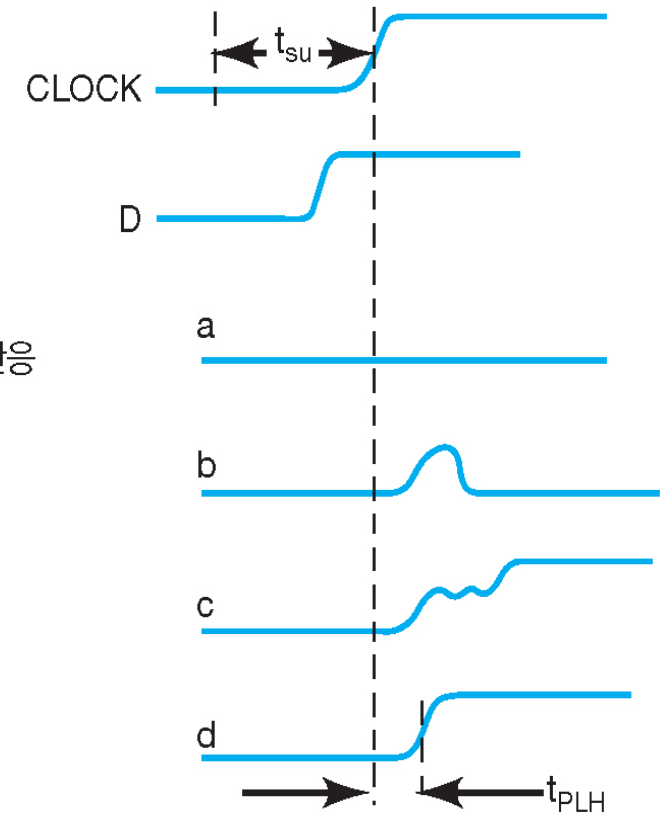
▪ 플립플롭에서 제시된 셋업 시간과 홀드 시간을 어겼을 때 나타나는 현상

(a) 출력이 HIGH 입력에 전혀 반응 하지 않는다

(b) 출력이 HIGH로 변화 하려고 하다가 다시 LOW로 돌아 온다

(c) 출력이 HIGH로 변화 하다가 중간에 잘못된 영역에 멈췄다가 HIGH 변화 한다

(d) 플립플롭이 원래 의도한 대로 HIGH로 간 경우 Q의 가능한 반응



(a)(b)(c) 경우를 **준안정 상태(metastable state)**라고 한다  
준안정 상태는 다른 논리 회로들을 혼란스럽게 만들어  
**시스템의 오동작을 일으킨다**

# 5-5 클럭 신호와 클럭 입력을 갖는 플립플롭

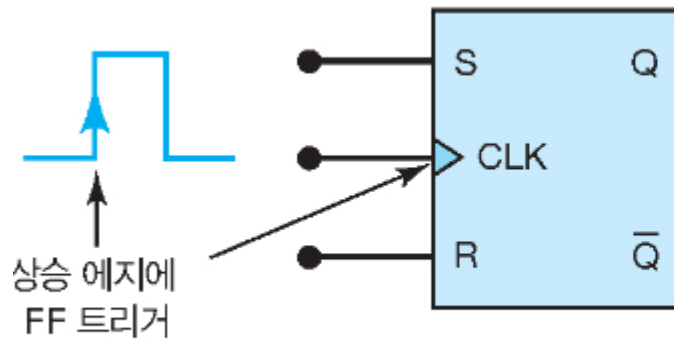
## ➤ 학습성과 평가문제

1. 클럭 입력을 갖는 플립플롭의 두 가지 종류의 입력에는 어떤 것들이 있는가?
2. 에지 트리거란 용어는 무엇을 의미 하는가>
3. 참 또는 거짓: CLK 입력은 제어 입력에 활성 천이가 발생 했을 때만 플립플롭의 출력에 영향을 미친다
4. 플립플롭에서 셋업 시간과 홀드 시간에 대하여 정의 하라
5. 참 또는 거짓: 준 안정 상태는 클럭 플립플롭을 사용하는 중요한 장점이다
6. 무엇이 플립플롭을 준 안정 상태로 만드는가?

# 5-6 클럭 입력을 갖는 S-R 플립플롭

## ➤ S-R 플립플롭

- NAND 래치와 NOR 래치를 공통적으로 S-R 래치라고 한다
- 클럭 신호의 PGT에 의해서 동작하는 S-R 플립플롭
  - S-R 입력은 NOR 게이트 래치와 같은 방법으로 출력 상태를 결정
  - 클럭 입력에 인가된 신호가 0에서 1로 변할 때 만 플립플롭의 상태가 변할 수 있다
    - 클럭의 PGT가 발생할 때 까지는 이들 입력에 대해 플립플롭의 출력은 응답하지 않는다
  - 진리표의 위를 향한 화살표(↑)는 CLK 신호에 PGT가 필요하다는 의미



(a)

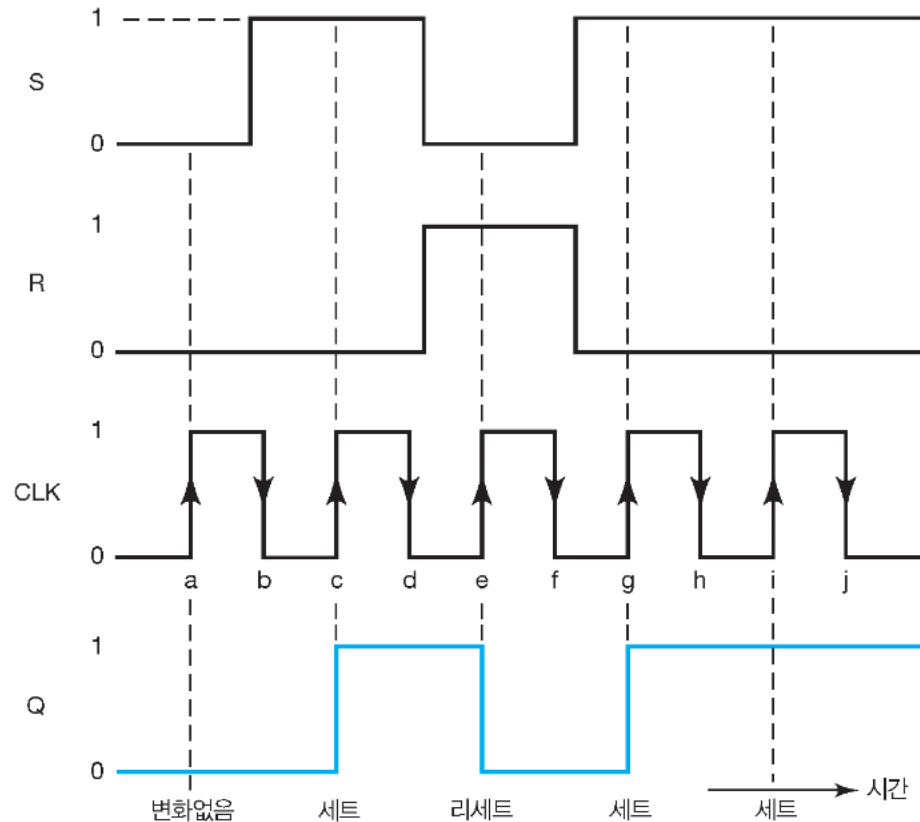
입력			출력
S	R	CLK	Q
0	0	↑	Q <sub>0</sub> (변화없음)
1	0	↑	1
0	1	↑	0
1	1	↑	모호한 상태

Q<sub>0</sub>는 CLK의 상승 에지전의 출력 레벨이다.  
CLK 하강 에지에는 아무런 Q 값의 변화가 없다.

(b)

# 5-6 클럭 입력을 갖는 S-R 플립플롭

## ➤ 클럭 입력을 갖는 S-R 플립플롭의 동작



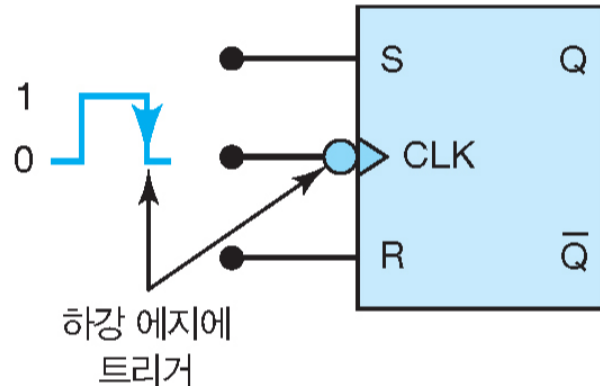
(c)

- 초기에 입력은 0 상태이고 Q 출력도 0 상태라고 가정
- 시점 a에서 첫번째 클럭 펄스의 PGT가 발생 하였을 때, S와 R 입력 모두가 0 상태 이므로 플립플롭 출력에는 영향을 미치지 않아 Q=0인 상태 유지 한다
- 시점 c에서 두 번째 클럭 펄스의 PGT 가 발생하였을 때, S 입력이 HIGH 이고, R 입력은 LOW이다. 그러므로 플립플롭은 이 클럭 펄스의 상승 에지에서 1 상태로 세트된다
- 시점 e에서 세번째 클럭 펄스의 상승 에지가 발생할 때 S=0이고, R=1이므로 플립플롭은 0 상태로 클리어 된다
- 시점 g에서 S=1, R=0이고, 네번째 클럭 펄스의 상승 에너지가 발생하므로 플립플롭은 다시 Q=1 상태로 세트 시킨다
- 다섯번째 펄스의 PGT 가 발생할 때 S=1, R=0이다. 그러나 이미 Q=1이므로 그 상태를 그대로 유지 한다
- S=R=1인 조건은 플립플롭의 출력 상태를 모호하게 만들기 때문에 사용하면 안된다

# 5-6 클럭 입력을 갖는 S-R 플립플롭

## ➤ S-R 플립플롭

- 클럭 신호의 **NGT에 의해서 동작하는 S-R 플립플롭**
  - S-R 입력은 NOR 게이트 래치와 같은 방법으로 출력 상태를 결정
  - 클럭 입력에 인가된 신호가 1에서 0로 변할 때 만 플립플롭의 상태가 변할 수 있다
    - 클럭의 NGT 가 발생할 때 까지는 이들 입력에 대해 플립플롭의 출력은 응답하지 않는다
  - 진리표의 위를 향한 화살표(↓)는 CLK 신호에 NGT가 필요하다는 의미
  - CLK 입력단의 작은 동그라미 와 삼각형 기호는 CLK 입력이 1에서 0으로 변화 될때만 플립플롭이 트리거 됨을 의미
  - 클럭 펄스의 하강 에지(b,d,f,h,j)에서 출력의 논리 상태가 변환하는 것을 제외하고 PGT 경우와 동일

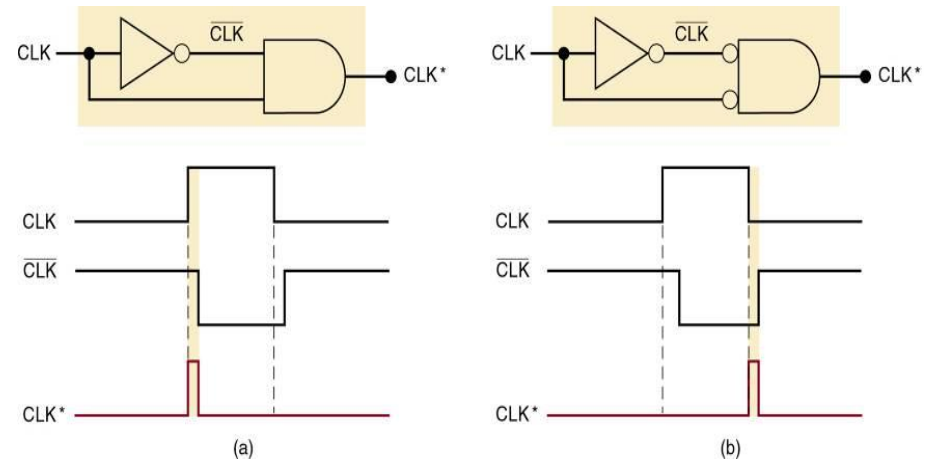
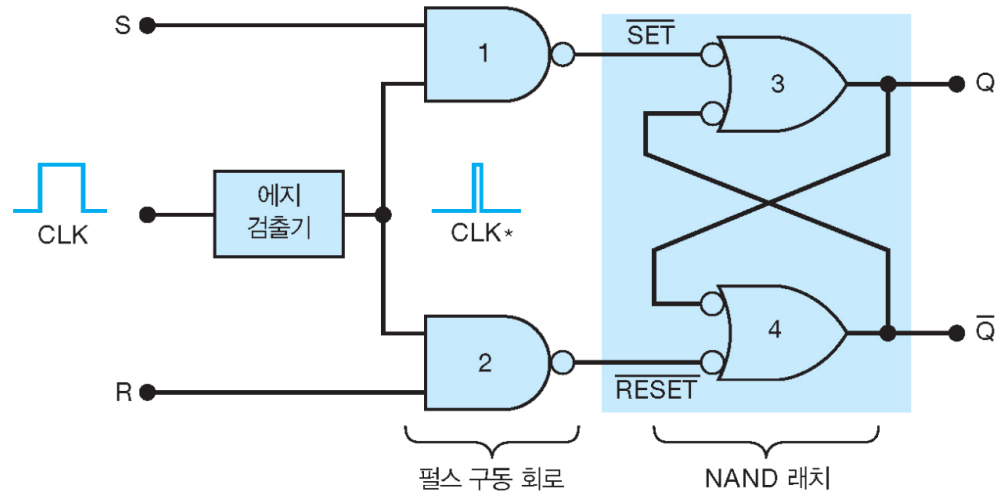


입력			출력
S	R	CLK	Q
0	0	↓	Q <sub>0</sub> (변화없음)
1	0	↓	1
0	1	↓	0
1	1	↓	모호한 상태

# 5-6 클럭 입력을 갖는 S-R 플립플롭

## ➤ 에지 플립플롭의 내부 회로

- NAND-3와 NAND-4 로 구성된 기본적인 NAND 게이트 래치
- NAND-1과 NAND-2에 의한 펄스 구동 회로(pulse-steering circuit)
- 에지 검출 회로(edge-detector circuit)

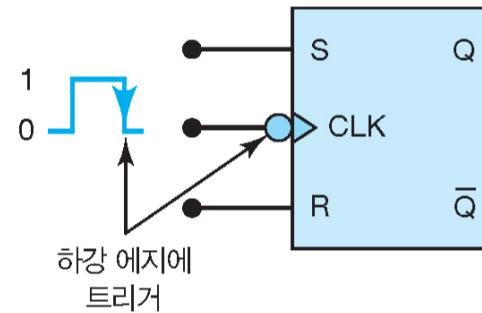
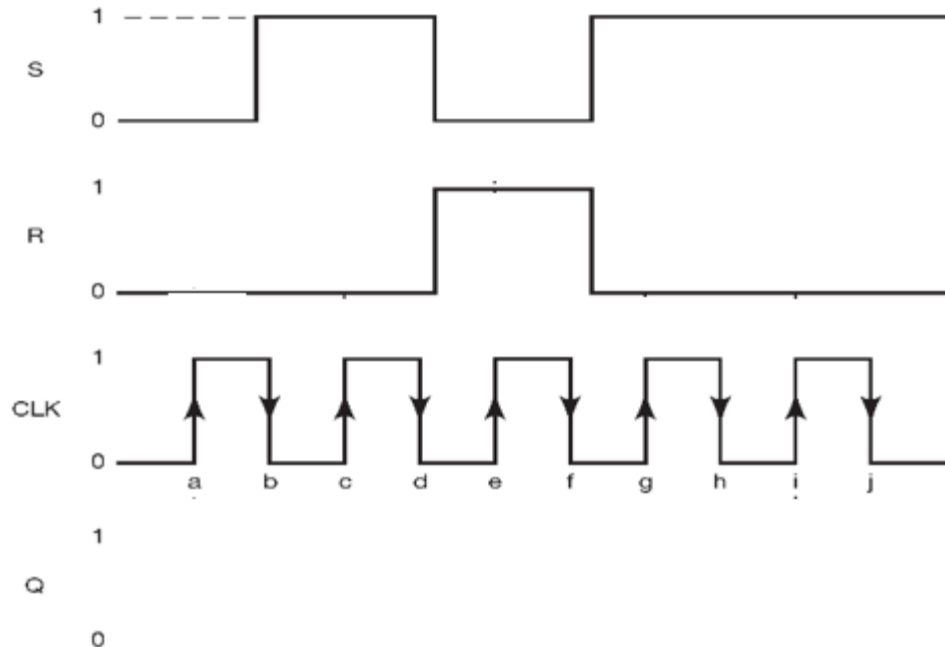


The duration of the  $CLK^*$  pulses is typically 2–5 ns.

# 5-6 클럭 입력을 갖는 S-R 플립플롭

## ➤ 학습성과 평가문제

1. 그림 5-20(C) 파형이 그림 5-21의 플립플롭 입력에 적용된다면, 시점 b에서 Q는 어떻게 되겠는가? 시점 f와 h에서는 어떻게 되는가?

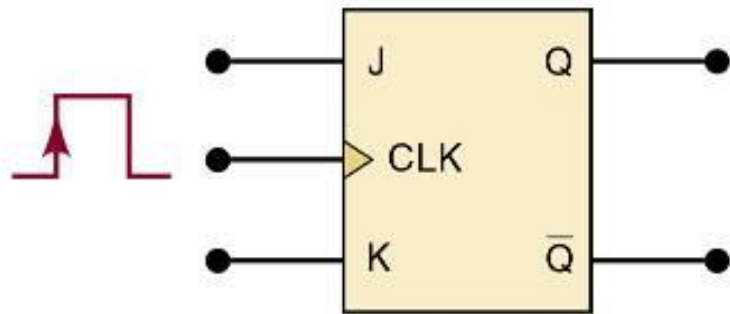


입력			출력
S	R	CLK	Q
0	0	↓	$Q_0$ (변화없음)
1	0	↓	1
0	1	↓	0
1	1	↓	모호한 상태

2. S와 R의 입력이 CLK의 활성 천이 동안만 Q에 영향을 미치는 이유를 설명하라.

# 5-7 클럭 입력을 갖는 J-K 플립플롭

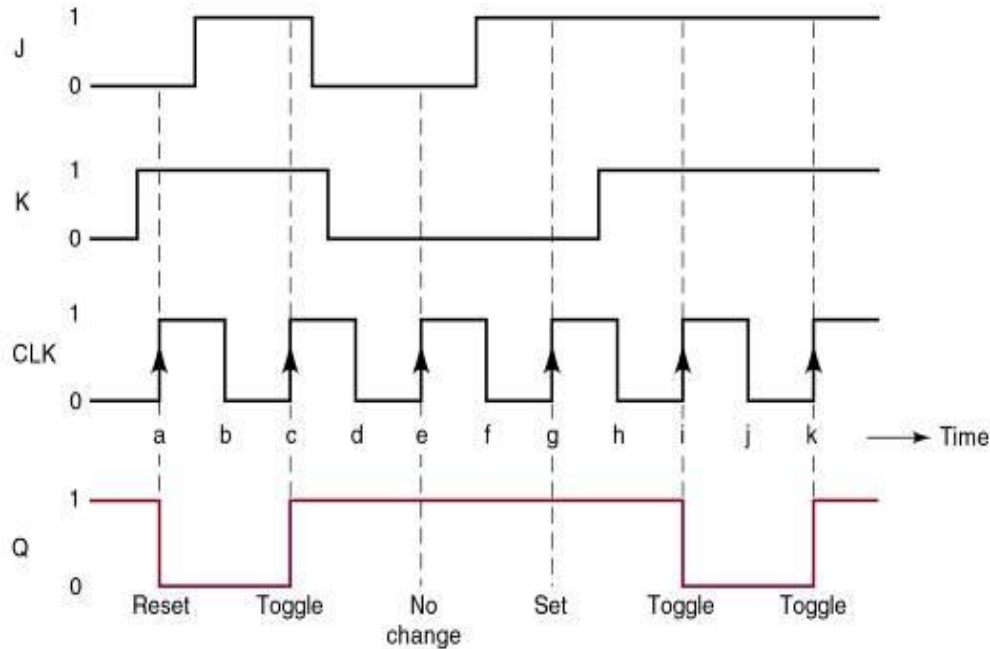
- 클럭 펄스의 PGT에서 J-K 플립플롭의 논리상태 변화
  - S-R 플립플롭과 같이 작동
    - J는 SET, K는 CLEAR
  - J=K=1인 조건에서 모호한 출력 상태를 갖지 않는다
    - J와 K가 모두 HIGH이면 출력은 반대 상태로 반전(Toggle)
  - 플립플롭의 토글 모드(Toggle Mode)
    - J=K=1 일 때  $Q = \overline{Q}_0$ 
      - ➔ 새로운 값 Q는 PGT 발생하기 전 값을 반전한 값



J	K	CLK	Q
0	0	↑	$Q_0$ (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)

# 5-7 클럭 입력을 갖는 J-K 플립플롭

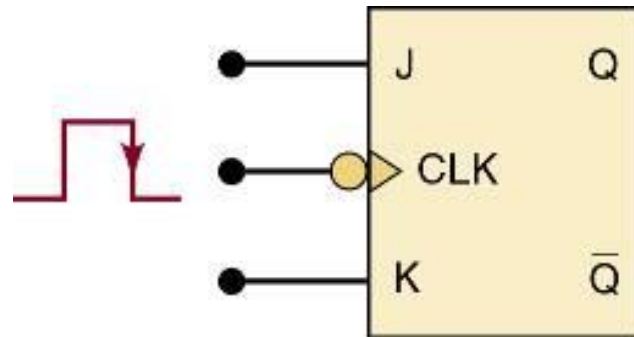
## ➤ 클럭 펄스의 PGT에서 J-K 플립플롭의 동작



- 초기에 입력은 0 이고 출력 Q는 1 이라 가정( $Q_0=0$ )
- 시점 a에서  $J=0, K=1$  조건이므로 첫 번째 클럭 펄스의 PGT 가 발생하면 플립플롭은  $Q=0$  상태로 클리어 된다
- 시점 c에서  $J=K=1$  이므로 두번째 클럭 펄스의 PGT가 발생하면, 플립플롭은  $Q=1$ 의 상태로 토글 된다
- 시점 e에서  $J=K=0$  이므로 플립플롭은 클럭 신호의 에지에서 스 상태를 변화 시키지 않는다
- 시점 g에서  $J=1, K=0$  조건이므로 Q를 1 상태로 세트시킨다 그러나 이미 1 상태이므로 그 값은 변하지 않는다
- 시점 i에서  $J+K+1$ 이고 플립플롭은 반대 방향으로 토글 된다
- 이와 같은 과정은 시점 k에서도 발생

# 5-7 클럭 입력을 갖는 J-K 플립플롭

- 클럭 펄스의 **NGT**에서 **J-K 플립플롭**의 논리상태 변화
  - 작동 클럭 펄스의 PGT 발생하는 순간을 제외하고 플립플롭의 논리 상태는 변하지 않는다
  - CLK 입력단의 작은 원은 CLK 입력이 1에서 0 상태로 변할 때 플립플롭이 트리거 됨을 의미  
플립플롭은 클럭 신호의 NGT(시점 b,d,f,h,j)에서 출력 상태가 변화
    - ➔ PGT에서 동작하는 플립플롭과 유사하게 동작

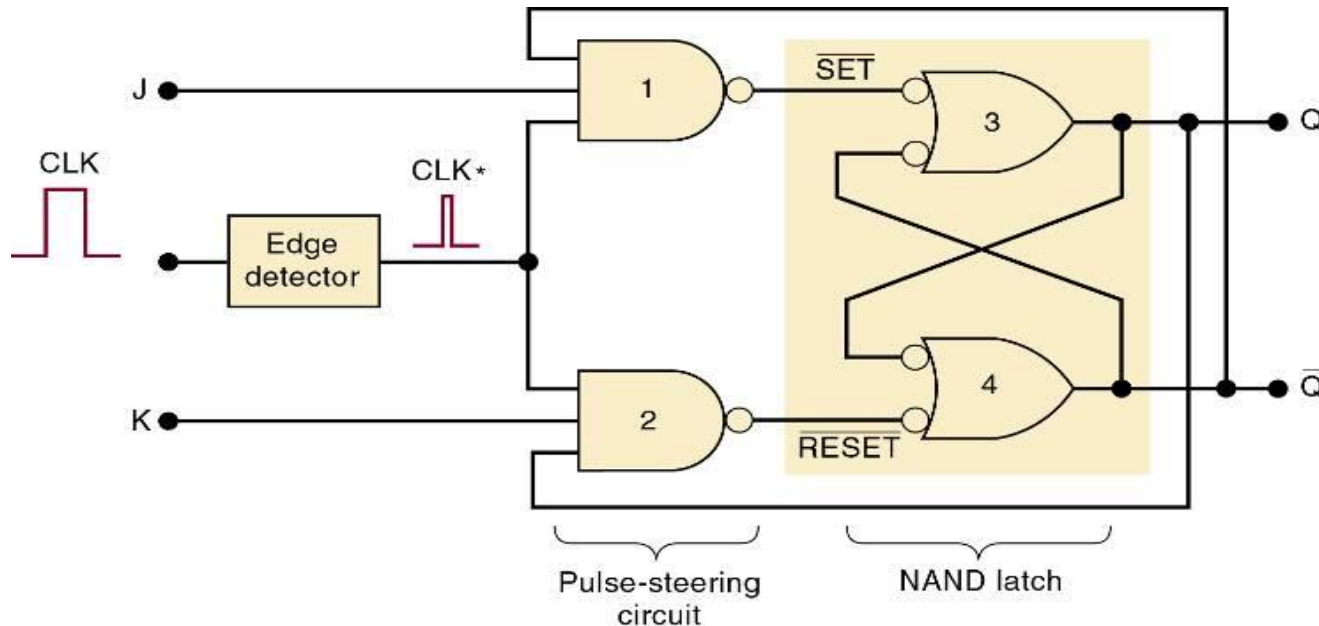


J	K	CLK	Q
0	0	↓	$Q_0$ (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\overline{Q_0}$ (toggles)

# 5-7 클럭 입력을 갖는 J-K 플립플롭

## ➤ 에지 트리거 J-K 플립플롭의 내부 회로

- NAND-3와 NAND-4 로 구성된 기본적인 NAND 게이트 래치
- NAND-1과 NAND-2에 의한 펄스 구동 회로(pulse-steering circuit)
- 에지 검출 회로(edge-detector circuit)



출력 Q와  $\bar{Q}$ 가 펄스 구동  
NAND 게이트로 귀환 연결  
→ 이 귀환 연결이 J=K=1인  
조건에서 J-K 플립플롭이  
토글될 수 있게 한다

# 5-7 클럭 입력을 갖는 J-K 플립플롭

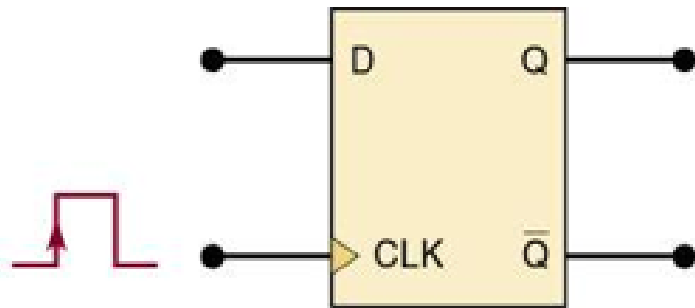
## ➤ 학습성과 평가문제

1. 참 또는 거짓 : J-K 플립플롭은 S-R 플립플롭으로 사용할 수 있으나, S-R 플립플롭은 J\_K 플립플롭으로 사용할 수 없다
2. J-K 플립플롭은 애매모호한 입력 상태를 가지는가?
3. J-K 플립플롭의 어떤 입력 조건이 활성 CLK 입력이 발생할 때 Q를 항상 세트 시키는가?

# 5-8 클럭 입력을 갖는 D 플립플롭

## ➤ D 플립플롭의 논리상태 변화

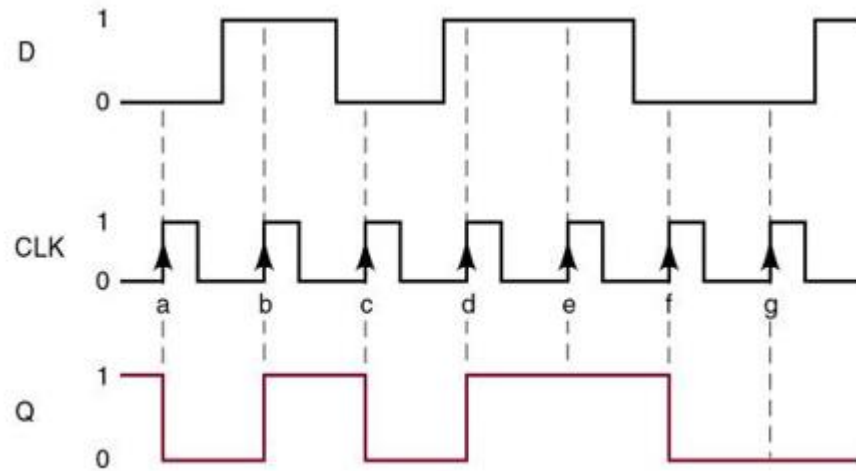
- 오직 하나의 동기식 제어 입력 D를 갖는다
  - D는 Data를 의미
- Q는 PGT의 CLK 입력이 발생하였을 때 D 입력의 상태와 같은 값을 출력한다
  - PGT가 발생하는 순간의 D 논리값이 플립플롭에 저장



D	CLK	Q
0	↑	0
1	↑	1

# 5-8 클럭 입력을 갖는 D 플립플롭

## ➤ 클럭 펄스의 PGT에서 D 플립플롭의 동작

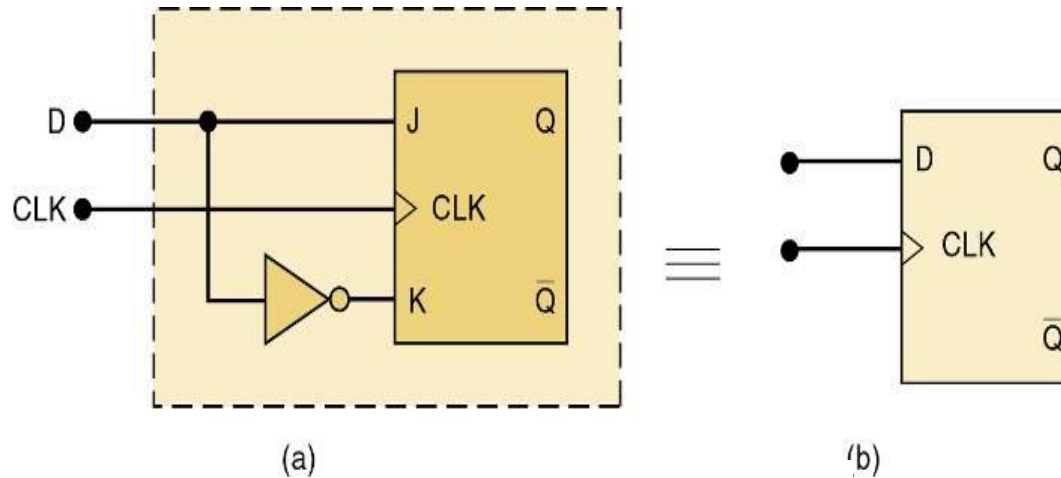


- Q의 초기값을 HIGH라 가정( $Q_0=1$ )
- 첫 번째 PGT가 시점 a에서 발생하였을 때 D 입력은 LOW. 그러므로 출력 Q는 0상태가 된다
- 입력 D의 논리 상태가 시점 a와 b 사이에서 변하더라도 Q에는 아무런 영향을 미치지 않는다
  - 시점 b에서 Q의 값을 HIGH로 변화하도록 하는 PGT가 발생할 때 까지 Q는 LOW를 저장
- PGT가 b에서 발생하였을 때 D가 HIGH 이므로 Q는 high 상태로 변한다
  - 시점 c에서 Q의 값을 LOW로 변화하도록 하는 PGT가 발생할 때 까지 Q는 HIGH를 저장
- 이와 같은 과정은 시점 d,e,f,g에서 PGT가 발생하면 그 시점의 D 값이 출력 Q로 전송된다

# 5-8 클럭 입력을 갖는 D 플립플롭

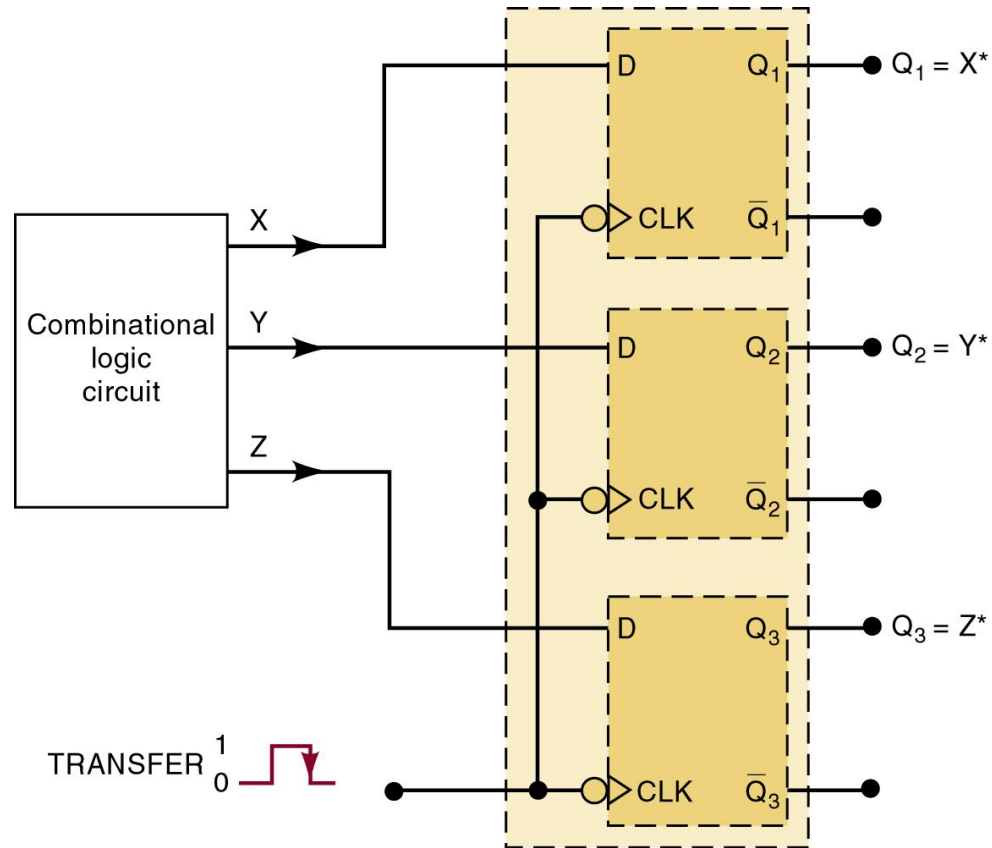
## ➤ D 플립플롭의 구현

- J-K 플립플롭에 1개의 인버터를 더함으로써 쉽게 구현
- 같은 방법으로 S-R 플립플롭으로도 D 플립플롭을 만들 수 있다



# 5-8 클럭 입력을 갖는 D 플립플롭

## ➤ 병렬 데이터 전송



\*After occurrence of NGT

- 논리회로의 출력 X, Y, Z가 각각 플립플롭의 Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>에 전달되어 저장되는 회로
  - 3 비트의 X, Y, Z가 동시에 전송
  - D 플립플롭을 사용하므로 공통 CLK 입력에 Transfer 펄스가 인가 될 때 X, Y, Z 값이 Q<sub>1</sub>, Q<sub>2</sub>, Q<sub>3</sub>에 전달
  - 플립플롭은 다음 단계를 위해 그 값을 저장할 수 있다

## 5-8 클럭 입력을 갖는 D 플립플롭

### ➤ 학습성과 평가문제

1. 입력 D가 영구적으로 LOW로 묶여 있다면, 그림 5-27(b) 파형 Q는 어떻게 되겠는가?
2. 참 또는 거짓: 출력 Q 는 모든 시간에 걸쳐 입력 D와 같은 값을 가질 수 있다
3. 병렬 전송을 위해 J-K 플립플롭을 사용할 수 있는가?

# 5-9 D 래치(투과성 래치)

## ➤ D 래치 구조 및 논리 상태

- 에지 트리거 D 플립플롭

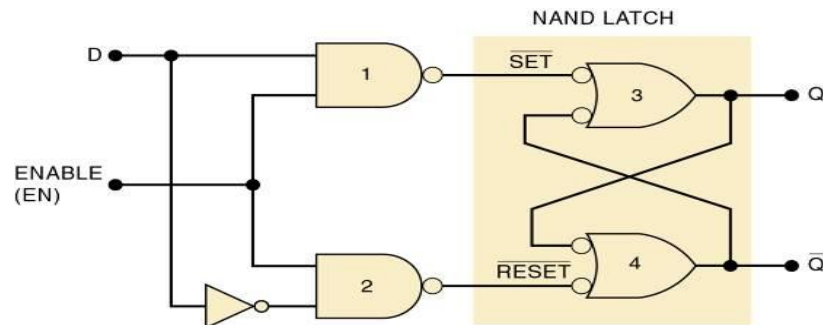
  - 클럭이 인가 될때만 입력 D에 의하여 출력이 응답하도록 에지 검출회로 사용

- D-래치는 에지 검출 회로를 사용하지 않는다

  - NAND 래치와 에지 검출회로가 없는 펄스 구동 NAND 게이트 1과 2로 구성

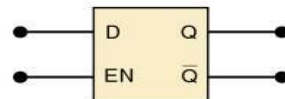
  - 구동 게이트의 공통 입력 신호는 **인에이블(Enable) 입력**이라 부른다

→ 이 신호가 Q와  $\bar{Q}$ 에 영향을 주는 것이 에지가 발생하는 순간에만 국한되지 않는다



Inputs		Output
EN	D	Q
0	X	$Q_0$ (no change)
1	0	0
1	1	1

"X" indicates "don't care."  
 $Q_0$  is state Q just prior to EN going LOW.

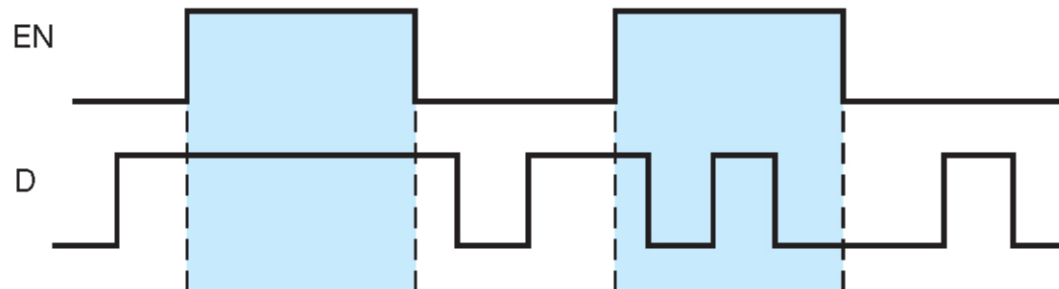


# 5-9 D 래치(투과성 래치)

## ➤ 예제 5-8

초기값  $Q=0$ 라고 가정

그림 5-31의 입력 EN과 D를 갖는 D-래치에 대한 Q의 변화를 구하라



# 5-9 D 래치(투과성 래치)

## ➤ 학습성과 평가문제

1. D-래치의와 에지 트리거 D-플립플롭의 동작 차이를 설명하라.
2. 참 또는 거짓: D-래치는  $EN=0$ 에서 투과 모드가 된다
3. 참 또는 거짓: D-래치에서 0 입력은  $EN=1$ 일 때만 Q에 영향을 준다

# 5-10 비동기 입력

## ➤ 동기 입력

- 플립플롭에서 제어입력 S, R, J, K 및 D 입력들은 CLK 입력에 동기 되어서 플립플롭에 영향

## ➤ 비동기 입력

- 동기입력과 클럭 입력에 독립적으로 동작
- 임의의 시간에서 플립플롭을 1 또는 0 상태로 세트 또는 클리어 시키기 위하여 사용
- 대부분의 clocked flip-flop들은 동기 입력과 하나 또는 그 이상의 비동기 입력을 갖는다
  - 비동기 입력은 PRE & CLR 라벨을 사용하고 라벨 상단에 윗줄을 그어 표시
  - 플립플롭 입력단의 방울은 LOW 활성화 입력을 표시

# 5-10 비동기 입력

## ➤ 2개의 비동기 입력을 갖춘 J-K 플립플롭

- $\overline{PRESET} = \overline{CLEAR} = 1$

비동기 입력들은 비활성이고 플립플롭 J, K 및 CLK 신호들에 의하여 동작

- $\overline{PRESET} = 0, \overline{CLEAR} = 1.$

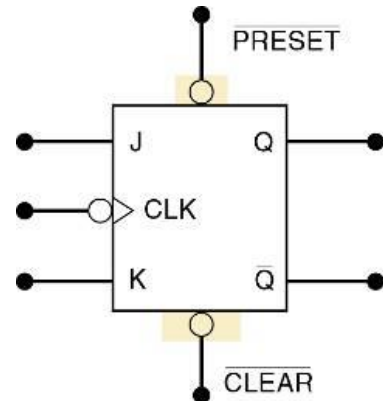
$\overline{PRESET}$  입력이 활성화 되고 출력 Q는 J, K 및 CLK의 현재 조건에 무관하게 즉시 1로 세트

- $\overline{PRESET} = 1, \overline{CLEAR} = 0$

$\overline{CLEAR}$  입력이 활성화 되고 출력 Q는 J, K 및 CLK의 현재 조건에 무관하게 즉시 0로 세트

- $\overline{PRESET} = 0, \overline{CLEAR} = 0$

이 조건은 애매모호한 결과를 발생 시키므로 사용하지 않는다

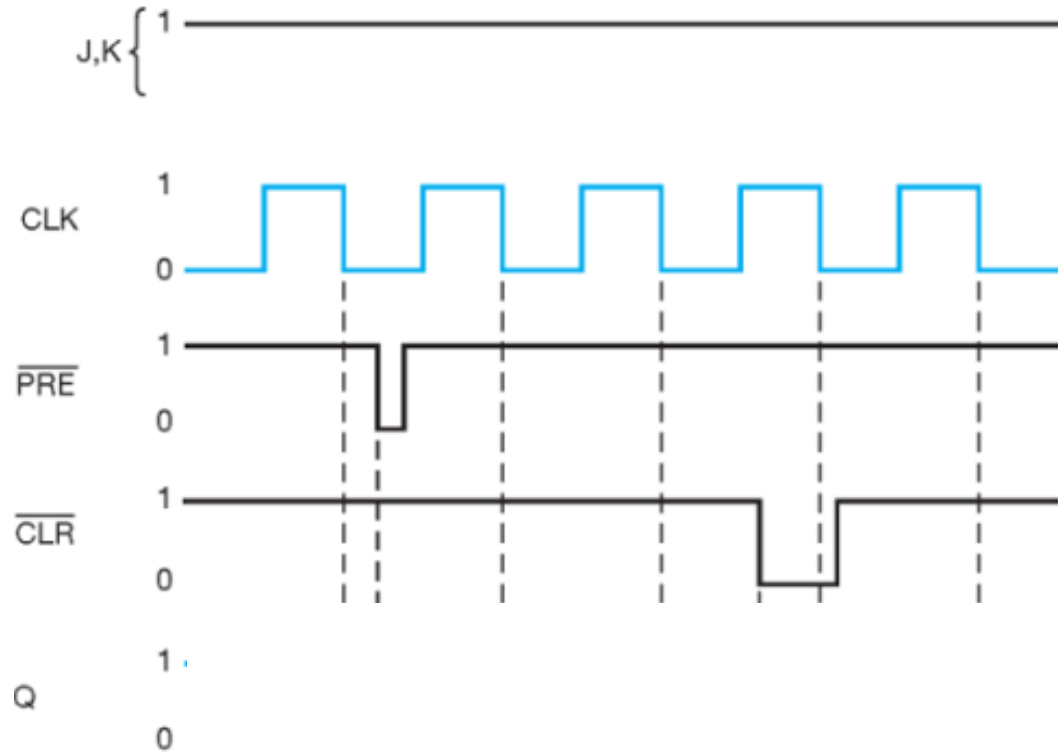
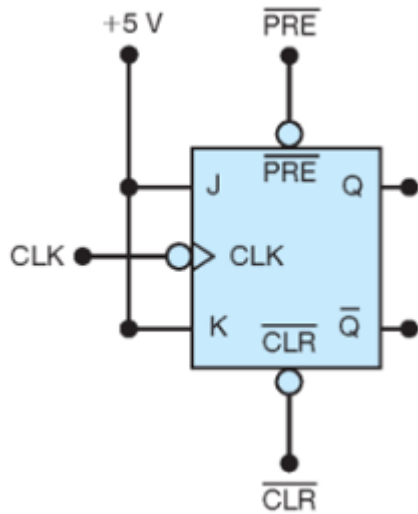


J	K	Clk	$\overline{PRE}$	$\overline{CLR}$	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\overline{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)

# 5-10 비동기 입력

## 예제 5-9

LOW 활성 NGT J-K 플립플롭  
입력파형에 대해 출력 Q?



(a)

# 5-10 비동기 입력

## ➤ 학습성과 평가문제

1. 비동기 입력은 동기입력과 어떻게 다르게 동작하는가?
2. D-플립플롭은  $\overline{PRE} = 1$ 일 때, 클럭 입력에 대하여 반응할 수 있는가?
3. LOW활성 비동기 입력을 갖고 PGT에서 동작하는 J-K 플립플롭이 반대 상태로 토글하기 위해 필요한 조건들을 써라

# 5-11 플립플롭의 타이밍 고찰

## ➤ 중요 타이밍 파라미터

- 셋업과 홀드 시간
- 전파 지연
- 최대 클럭 주파수
- 클럭 펄스의 HIGH와 LOW 시간
- 비동기 펄스의 폭
- 클럭의 천이시간

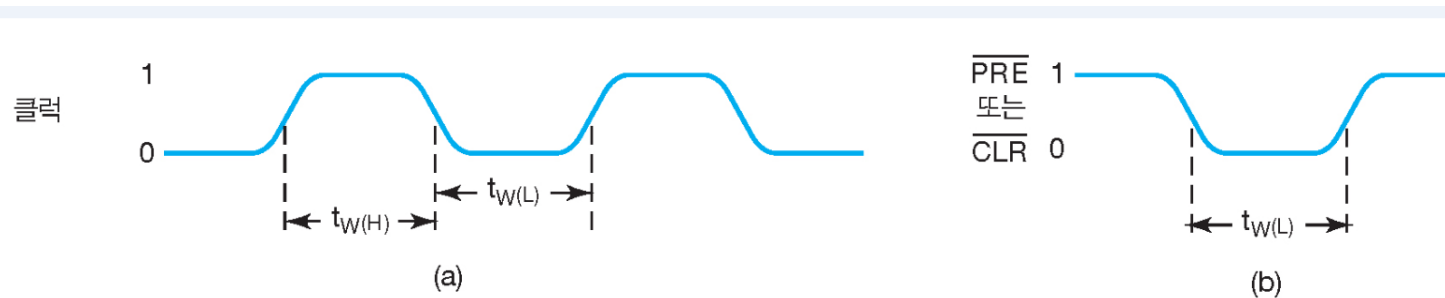
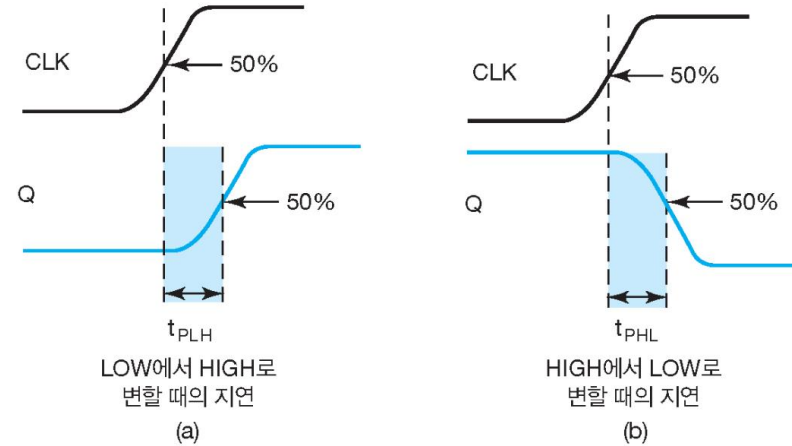


그림 5-35 (a) 클럭 LOW와 HIGH 시간. (b) 비동기 펄스의 폭.

# 5-11 플립플롭의 타이밍 고찰

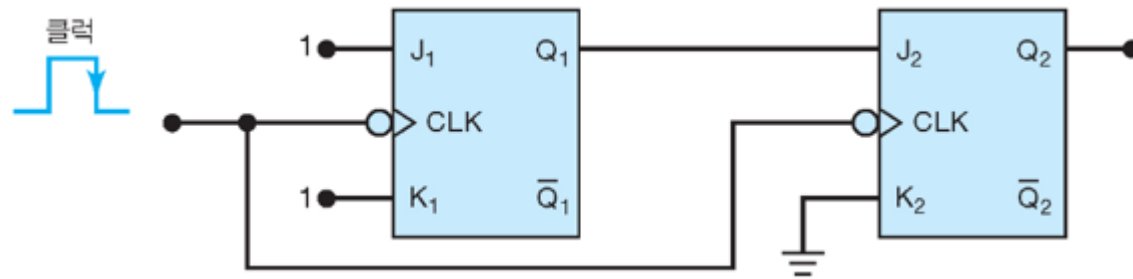
## ➤ 학습성과 평가문제

1. 입력에 의하여 출력 Q가 반응하는 시간을 나타내는 플립플롭의 타이밍 파라미터들은 어느 것인가?
2. 참 또는 거짓: 25MHz  $f_{max}$ 를 갖는 플립플롭은 25MHz 아래의 주파수를 갖는 CLK 펄스에 의하여 신뢰성 있게 트리거 될 수 있다
3. 클럭 에지가 발생하기 전에 제어 입력이 안정된 상태를 유지해야 하는 최소시간을 무엇이라 부르는가?
4. 클럭 에지가 발생한 후에 제어 입력이 안정된 상태를 유지해야 하는 최소시간을 무엇이라 부르는가?
5. 참 또는 거짓: 클럭의 파형의 상승시간이나 하강시간이 너무 길면 플립플롭을 안정되게 트리거 할 수 없다

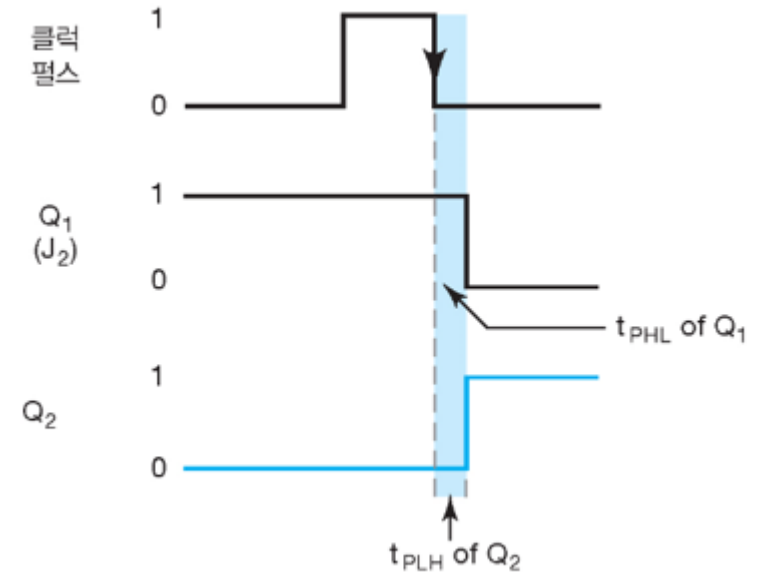
# 5-12 플립플롭 회로에서의 잠재적인 타이밍 문제

## ▶ 잠재적 타이밍 문제

- 플립플롭의 출력은 다른 플립플롭의 입력에 직접 또는 논리 제어 게이트를 통하여 연결
- 이 플립플롭들은 같은 클럭 신호에 의하여 트리거



플립플롭의 출력은 활성 클럭 천이가 나타나기 직전에 동기제어 입력에 인가되어 있는 논리신호에 의해 결정된다

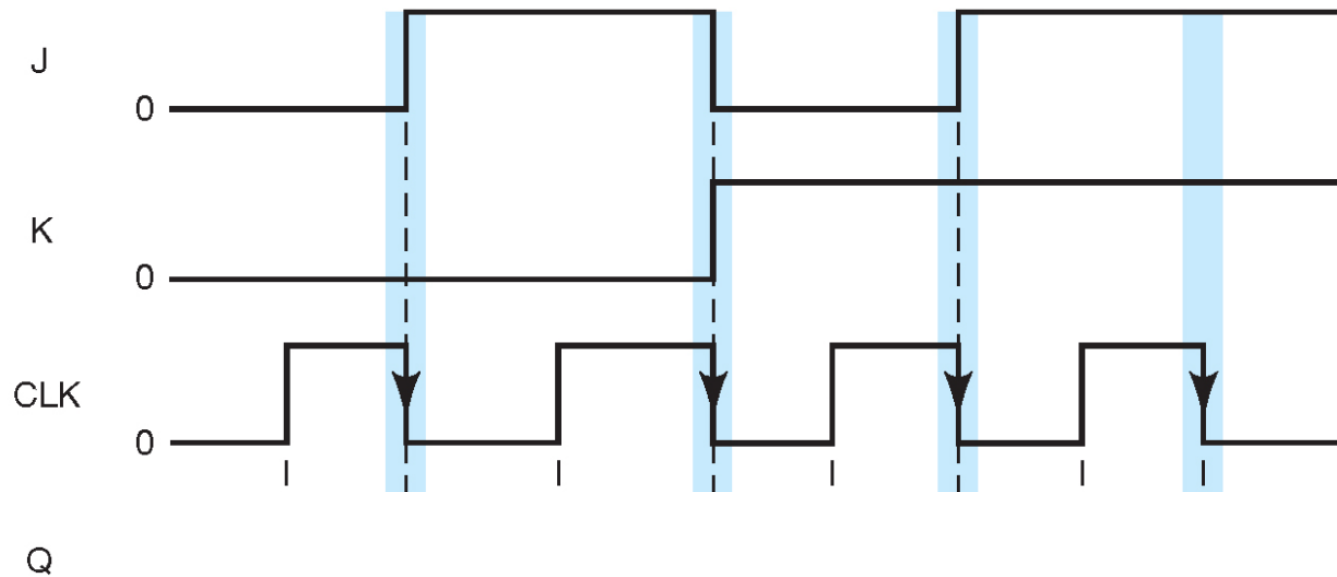


# 5-12 플립플롭 회로에서의 잠재적인 타이밍 문제

## ➤ 예제 5-10

아래 입력 신호에 대한 NGT J-K 플립플롭의 출력 Q 값을 구하라.

단  $t_H=0$ 이고 초기에  $Q=0$ 이다



# 5-12 플립플롭 회로에서의 잠재적인 타이밍 문제

## ➤ 학습성과 평가문제

1. 참 또는 거짓: 동기 회로는 출력이 활성 클럭 엣지에서 변하므로 언제나 세트와 홀드 시간 조건을 위반한다.
2. 참 또는 거짓: 동기 플립플롭이 안정되게 동작하려면 전파지연 시간이 홀드 시간 조건보다 길어야 한다
3. 동기 회로를 분석하기 위해서 출력 변화를 정하는 제어입력은 어디서 찾아야 하는가?

# 5-13 플립플롭의 응용

## ➤ 에지 트리거 플립플롭의 응용분야

- 계수, 2진 데이터 저장, 2진 데이터를 한곳에서 다른 곳으로 전송 등의 응용분야에 사용
- 이런 대부분의 응용에서는 플립플롭의 클럭 동작을 사용한다  
→ 순차회로라 불리는 응용분야에 해당

## ➤ 순차 회로

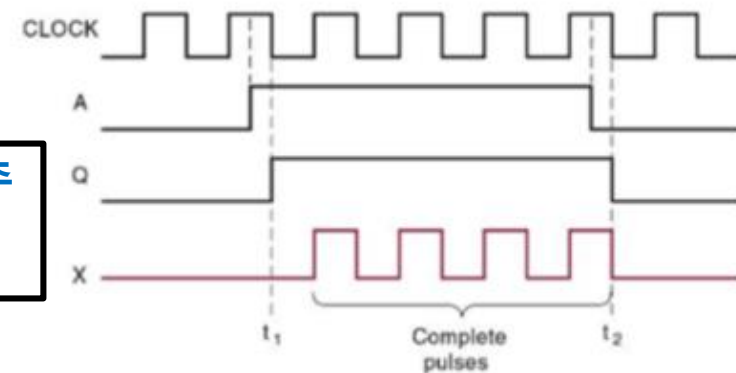
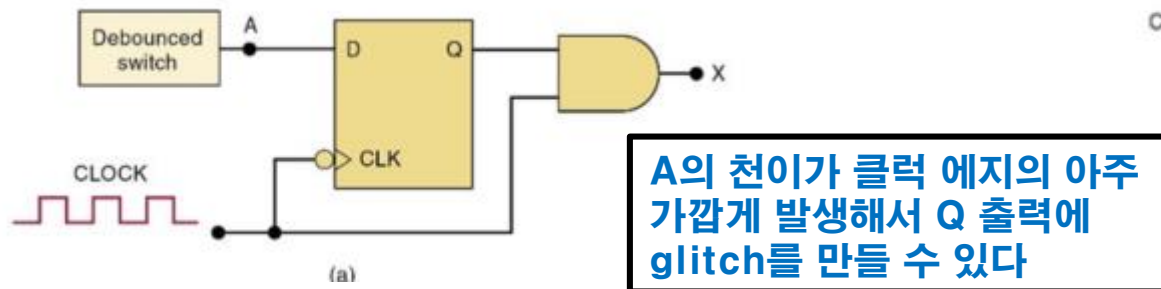
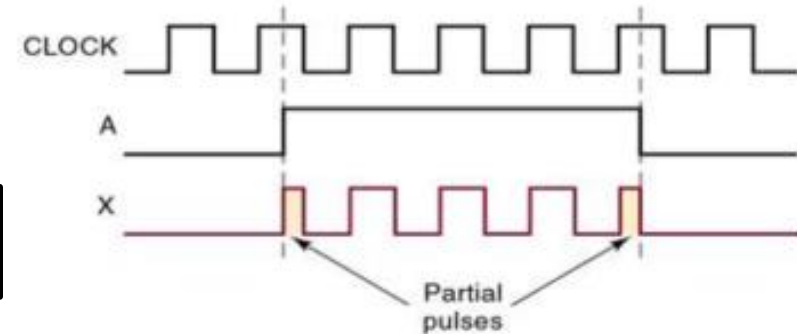
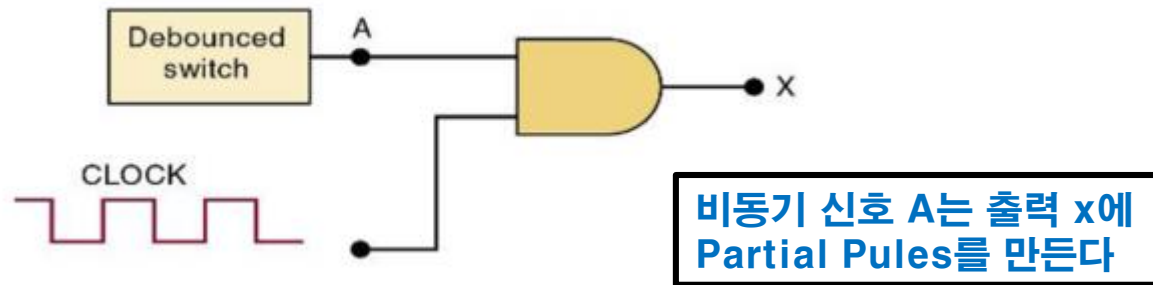
- 미리 정해진 상태를 따라 순차적으로 출력이 변화  
→ 클럭이 발생할 때 마다 새로운 상태로 변화 2진 데이터 저장
- 다시 피드백의 개념이 적용되지만 FF 메모리 소자 안드는 것에만 해당하는 것이 아니다  
→ FF 출력은 FF 동작을 제어하는 순차회로의 게이트들의 입력으로 피드백한다  
→ 그래서 다음 클럭 사이클에 다른 상태가 결정되게 된다

# 5-14 플롭의 동기화

## 예제 5-11

입력신호 A는 운영자가 플립 스위치를 올렸을 때 HIGH가 되고 내렸을 때 LOW가 된다

입력신호 A는 클럭 신호가 AND 게이트를 통과하는 것을 제어하기 위하여 사용되며, 입력회로 A가 HIGH인 동안에만 신호가 출력 X에 나타난다



# 5-14 플롭의 동기화

## ➤ 학습성과 평가문제

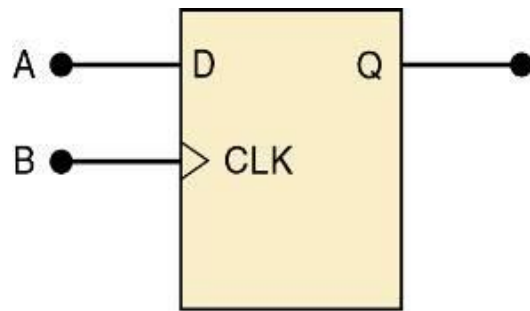
1. 동기화 회로는 어느 경우에 사용되어야 하는가?

2. 신호들을 동기 시키기 위해 DFF를 어떻게 사용하는가?

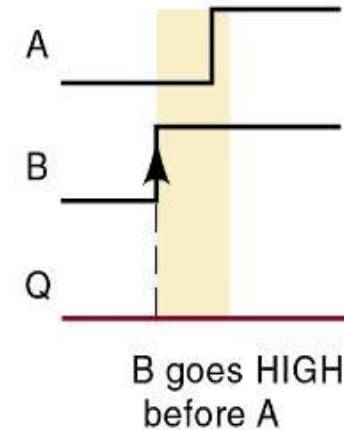
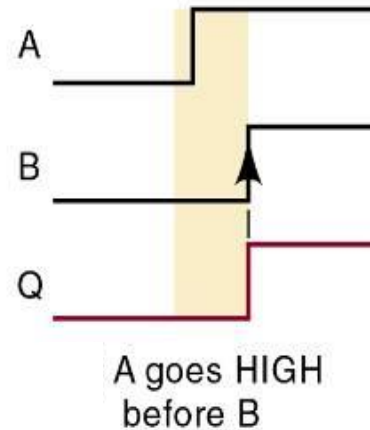
# 5-15 입력 순차 검출

## ➤ 입력의 특정 순서에 따라 반응하는 D-FF

- 회로가 적절히 동작하기 위해서는 A가 B 보다 적어도 플립플롭의 셋업 시간 만큼은 먼저 HIGH가 되어야 한다



제대로 작동하려면 A가 B보다 먼저 FF 설정 시간과 같은 시간만큼 HIGH가 되어야 한다



# 5-15 입력 순차 검출

## 예제 5-12

DFF를 사용하여 어느 방향으로 축이 회전하는지 정하시오

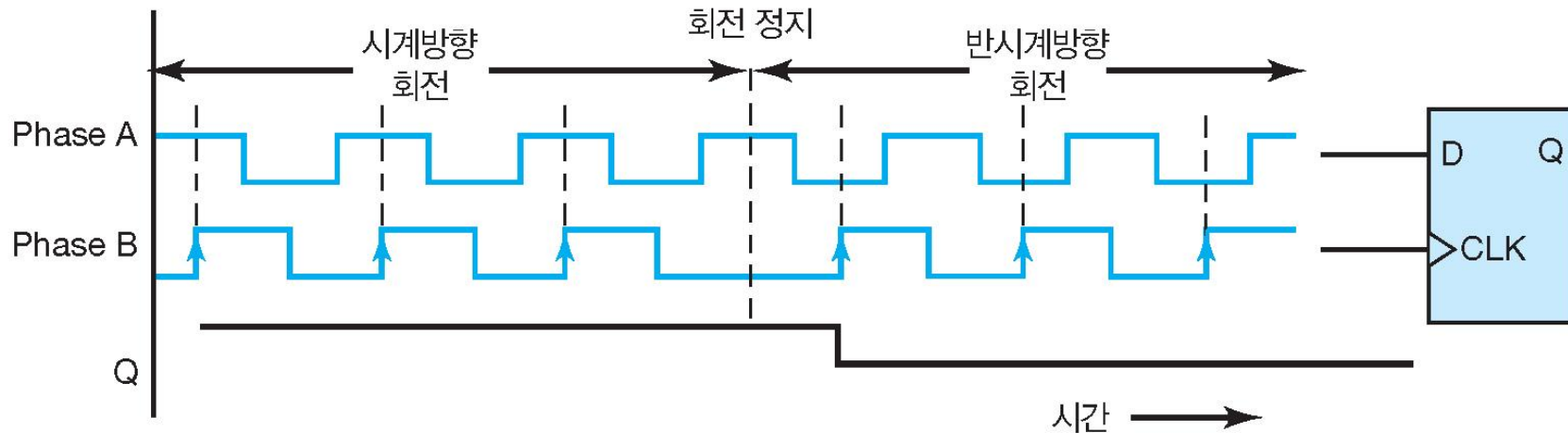
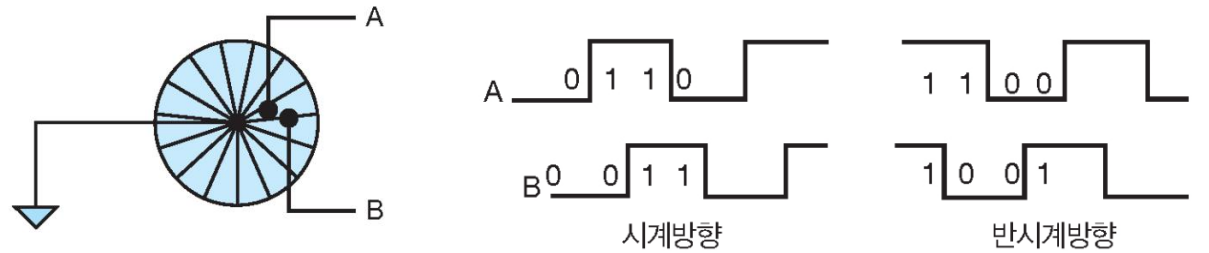


그림 5-41 회전 방향을 정하는 인코더 신호 열. DFF이 이 열을 탐지한다.

# 5-15 입력 순차 검출

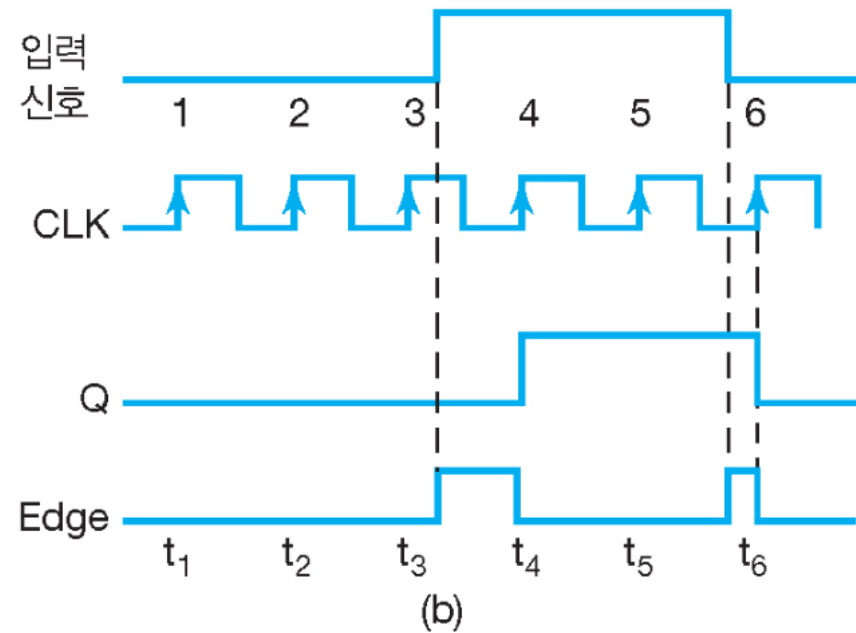
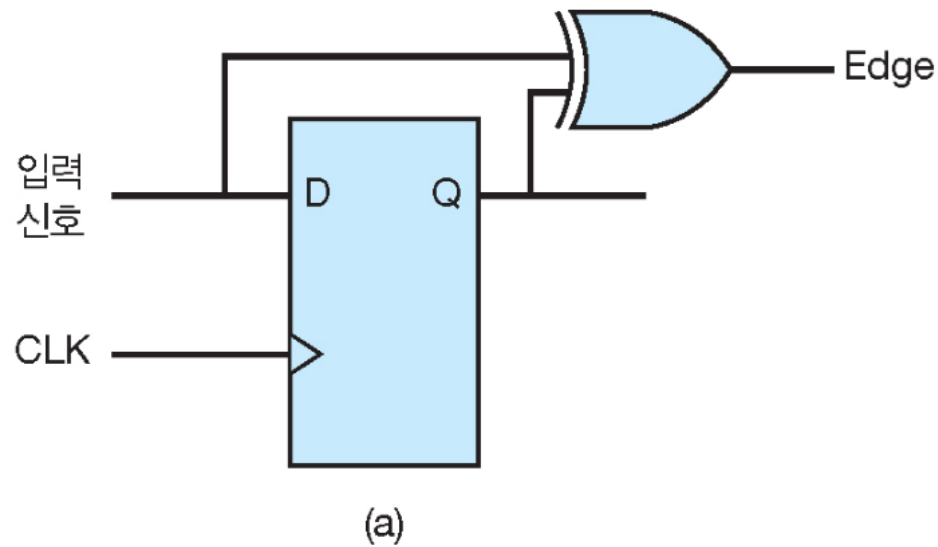
## ➤ 학습성과 평가 문제

1. DFF은 어느 입력이 먼저 변하는지를 어떻게 확인 하는가?

# 5-16 천이 또는 이벤트 검출

## ➤ D-FF과 XOR 게이트를 사용한 이벤트 검출

- 입력신호가 지난번 클럭 에지 후에 변화할 때 마다 HIGH 출력을 내는 회로가 필요
- 입력의 논리 레벨과 Q 출력의 논리 레벨을 비교



# 5-16 천이 또는 이벤트 검출

## ➤ 학습성과 평가 문제

1. 이벤트 탐지회로에서 XOR 게이트의 목적은 무엇인가?
2. CLK가 1MHz이고 듀티비가 50%라고 가정하자. 만일 입력 신호가 CLK의 하강에지에서 상태가 바뀌었다면 출력 EDGE 신호는 어떻게 나올까?

# 5-17 데이터의 저장과 전송

## ➤ 데이터 저장

- 플립플롭의 가장 일반적인 사용처는 데이터나 정보 저장이다
- 데이터는 일반적으로 레지스터라고 부르는 플립플롭의 그룹에 저장된다.

## ➤ 데이터 전송

- 데이터를 플립플롭 또는 레지스터에 저장하기 위하여 수행하는 동작을 데이터 전송이라 한다

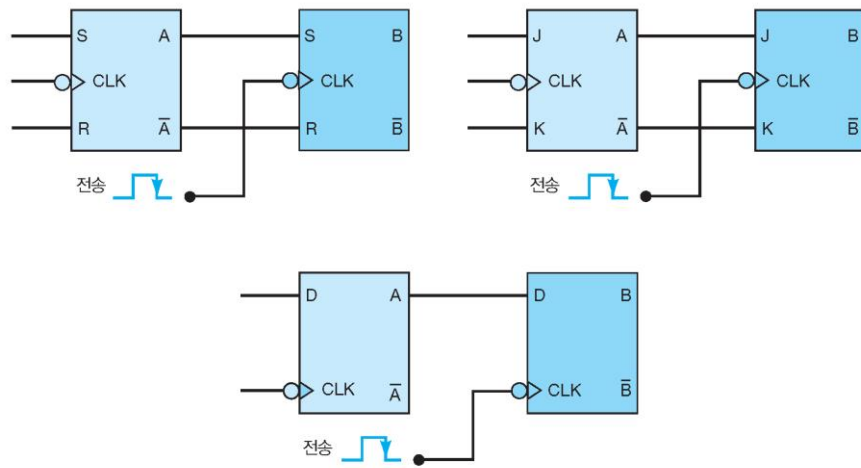


그림 5-43 클럭 입력을 갖는 여러 플립플롭에 의해 실행되는 동기 데이터 전송.

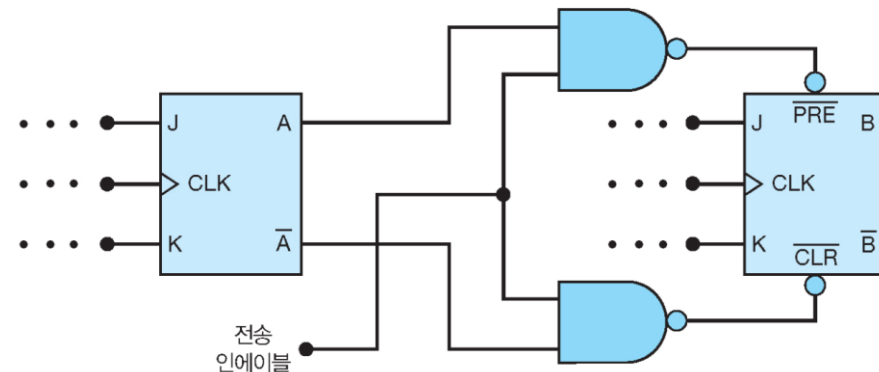


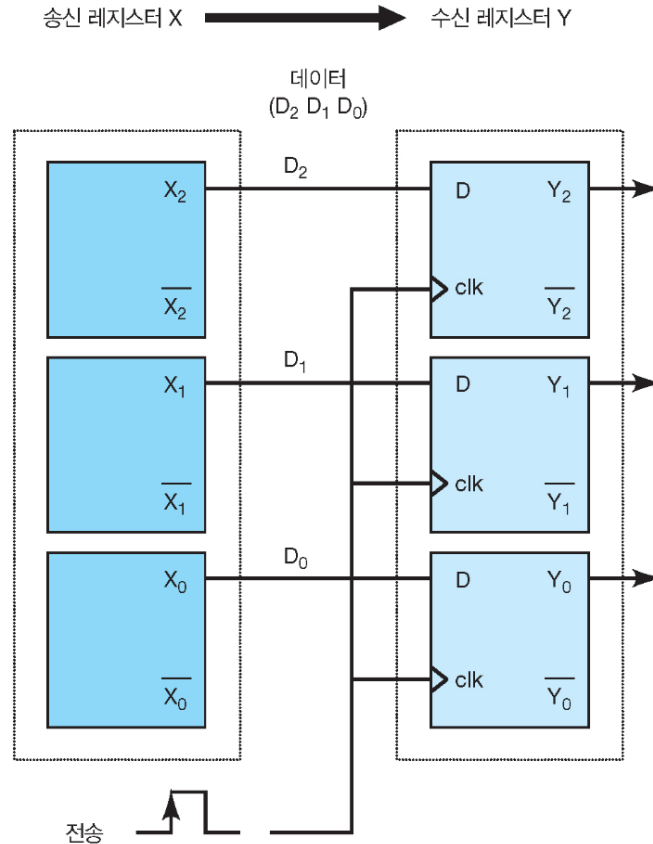
그림 5-44 비동기 데이터 전송 동작

# 5-17 데이터의 저장과 전송

## ➤ 병렬 데이터 전송

- 데이터를 플립플롭 또는 레지스터에 저장하기 위하여 수행하는 동작을 데이터 전송이라 한다

그림 5-45 레지스터 X에서  
레지스터 Y로의 병렬 전송.



# 5-17 데이터의 저장과 전송

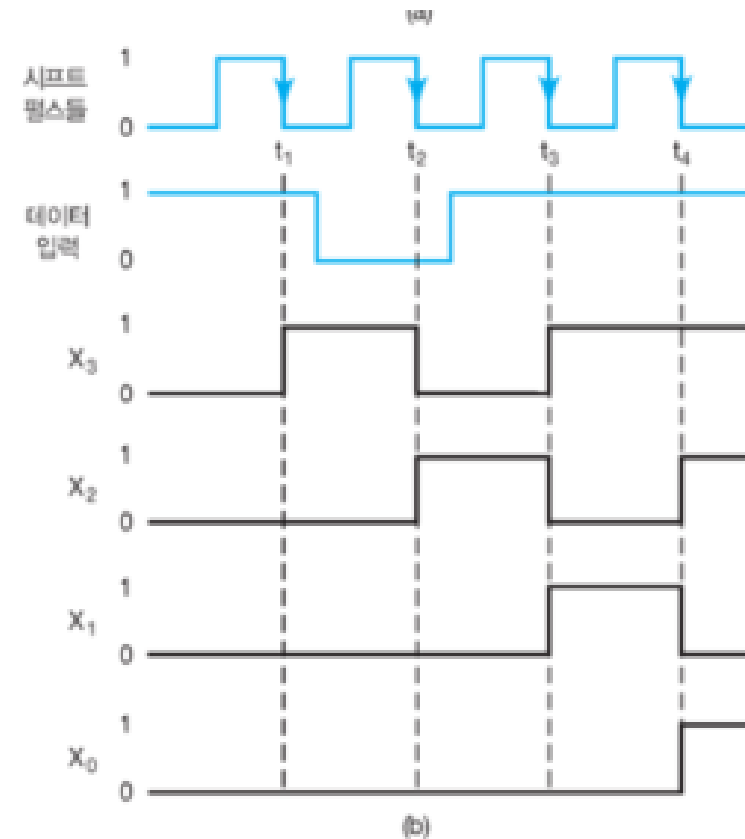
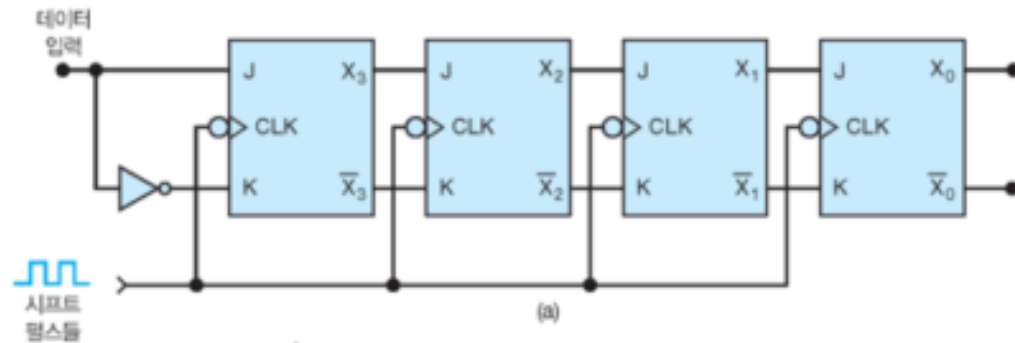
## ➤ 학습성과 평가 문제

1. 참 또는 거짓: 비동기 데이터 전송은 CLK 입력을 이용한다
2. 동기 전송에서 하나의 플립플롭에서 다른 플립플롭으로 연결선을 최소화하기에 적합한 플립플롭은 어떤것인가?
3. 그림 5-45의 레지스터에서 J-K 플립플롭을 이용한다면 레지스터 X에서 레지스터 Y로 얼마나 많은 결선이 필요한가?
4. 참 또는 거짓: 동기 전송은 비동기 전송보다 회로가 적다

# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 시프트 레지스터

- 플립플롭에 저장된 2진수를 매 클럭 마다 하나의 플립플롭에서 다른 플립플롭으로 이동하도록 결선된 플립플롭의 집합이다
- 4 비트 시프트 레지스터

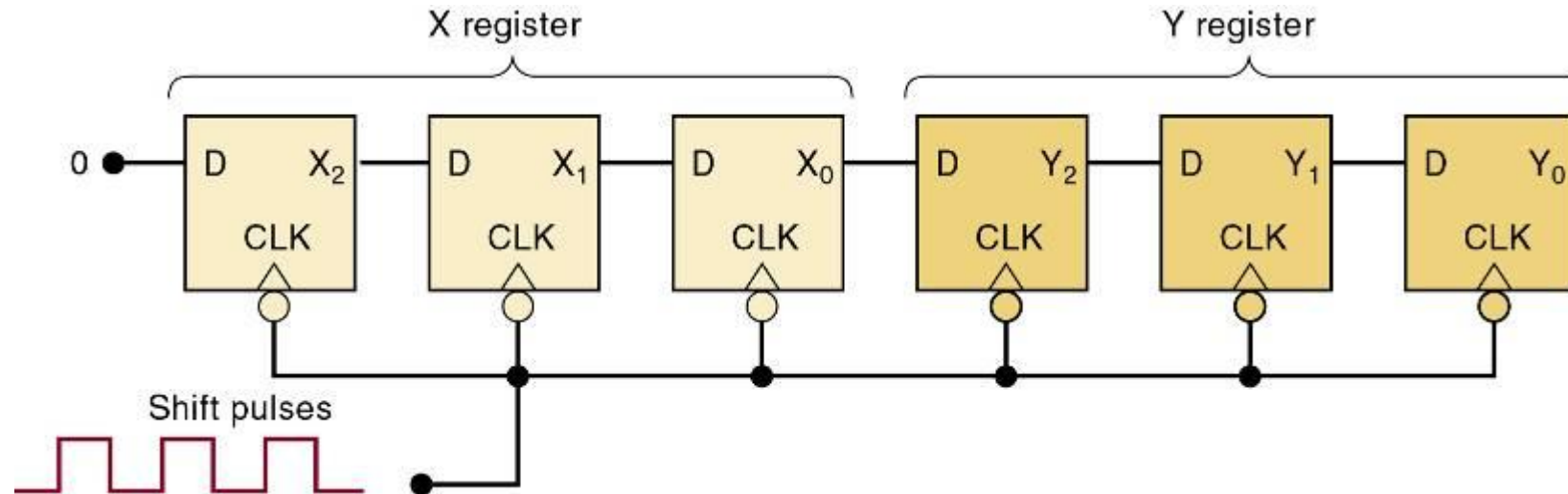


J, K 입력이 CLK 천이와 거의 같은 시간에 변하는 경우가 있기 때문에 플립플롭들은 매우 작은 홀드시간을 가질 필요가 있다.

## 5-18 직렬 데이터 전송, 시프트 레지스터

### ➤ 레지스터 사이의 직렬 전송

- X 레지스터의 내용은 레지스터 Y로 순차적으로 전송 (이동)된다



각 시프트 레지스터의 D 플립 플롭은 J-K 플립 플롭보다 적은 연결을 필요로 한다.

# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 레지스터 사이의 직렬 전송

3 비트 데이터의 완전한 전송은 3 개의 시프트 펄스 요구한다.

$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$
1	0	1	0	0	0
0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	1	0	1

← Before pulses applied

# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 레지스터 사이의 직렬 전송

- 2 개의 연결된 3 비트 시프트 레지스터

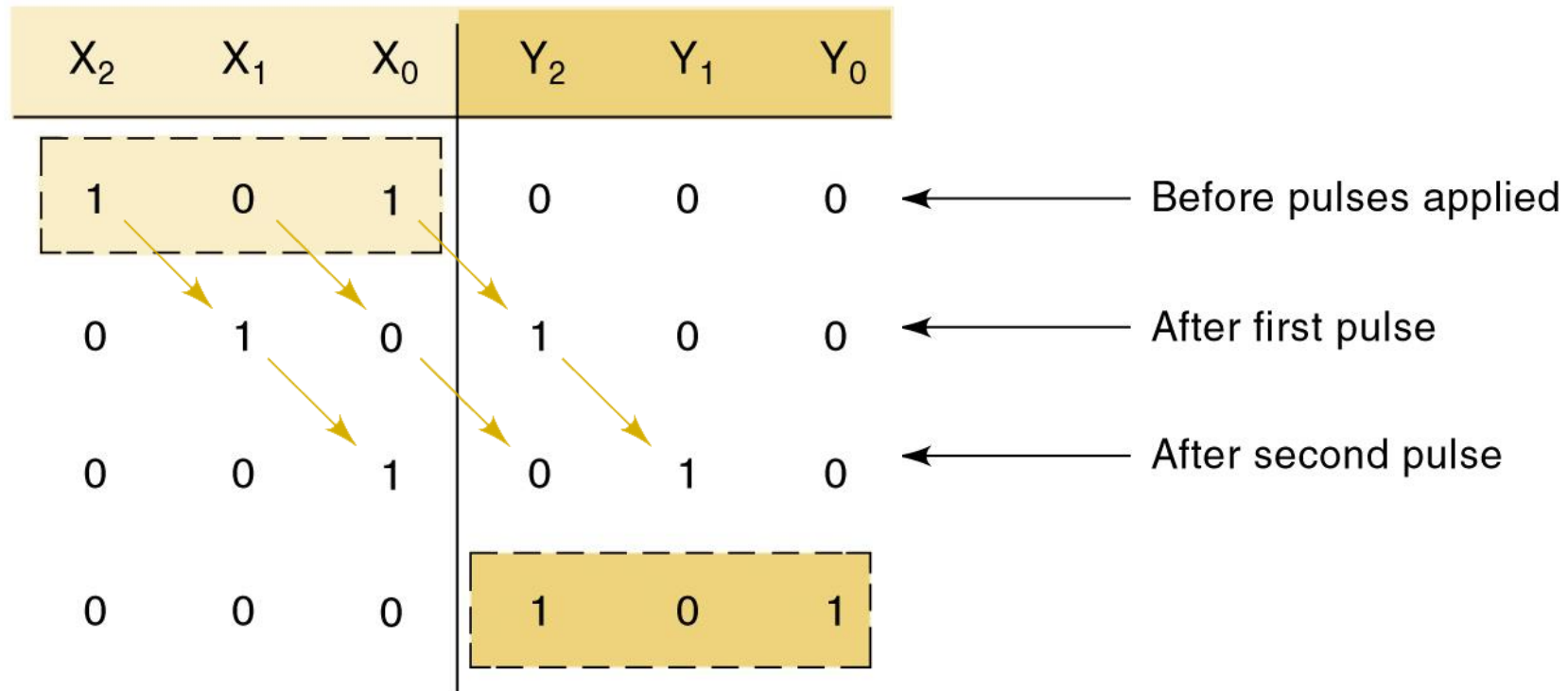
- 각 펄스 NGT에서 각 FF는 펄스 이전에 왼쪽에 있는 FF에 저장된 값을 취한다.

$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$	
1	0	1	0	0	0	← Before pulses applied
0	1	0	1	0	0	← After first pulse
0	0	1	0	1	0	
0	0	0	1	0	1	

# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 레지스터 사이의 직렬 전송

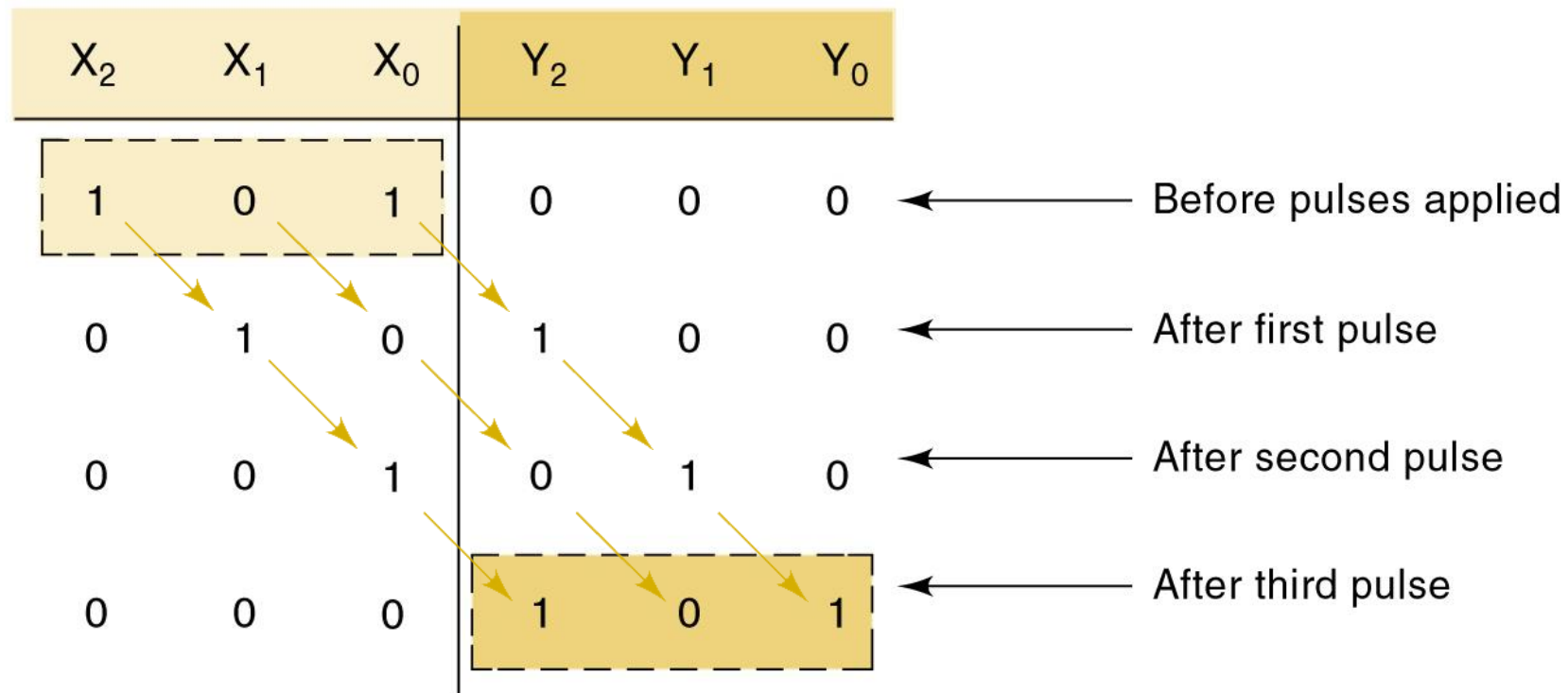
- 두 개의 연결된 3 비트 시프트 레지스터
  - 각 펄스 NGT에서 각 FF는 펄스 이전에 왼쪽에있는 FF에 저장된 값을 사용한다



# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 레지스터 사이의 직렬 전송

- 2 개의 연결된 3 비트 시프트 레지스터
  - 각 펄스 NGT에서 각 FF는 펄스 이전에 왼쪽에있는 FF에 저장된 값을 사용한다



# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 레지스터 사이의 직렬 전송

- X 레지스터에 저장된 101은 이제 Y 레지스터로 이동되었다
- X 레지스터는 원래 데이터를 잃었으며 000에 있다.
- 2 개의 연결된 3 비트 시프트 레지스터

3 펄스 후

- 처음에 X2의 1은 Y2에
- 처음에 X1의 0은 Y1에
- 처음에 X0에있는 1은 Y0

X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
1	0	1	0	0	0
0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	1	0	1

# 5-18 직렬 데이터 전송, 시프트 레지스터

## ➤ 예제 5-13

그림 5-47에서와 같은 초기값을 갖는 레지스터 X와 Y를 가정하자. 여섯번째의 시프트 펄스가 발생한 후의 각 플립플롭은 어떤값을 갖는가?

# 5-18 직렬 데이터 전송, 시프트 레지스터

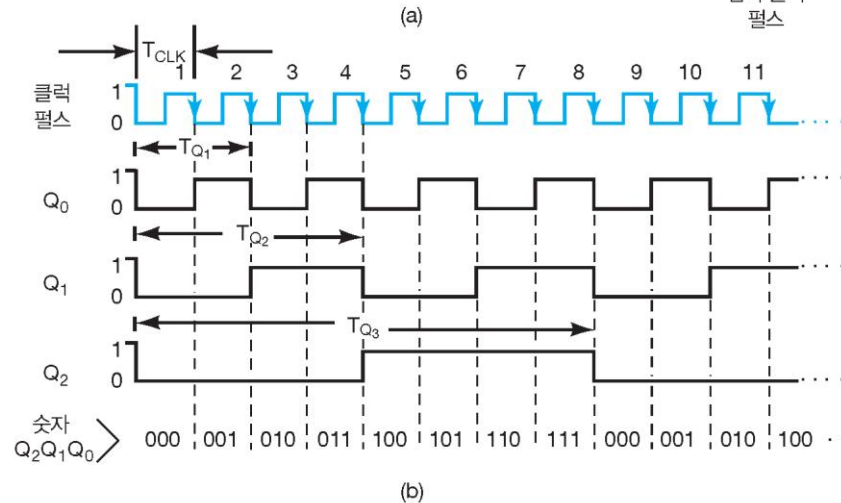
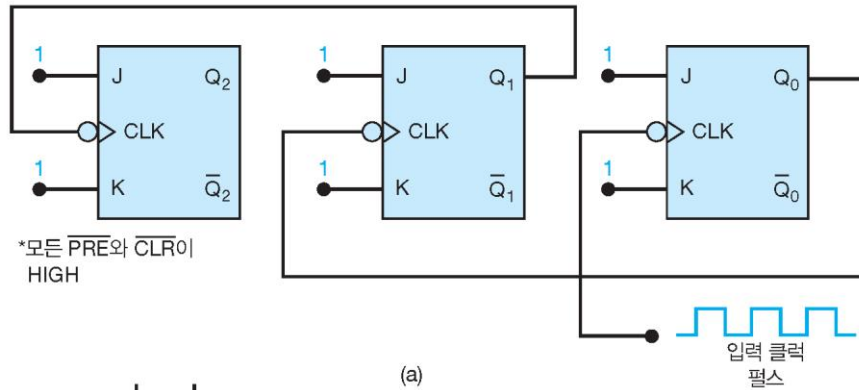
## ➤ 학습성과 평가 문제

1. 참 또는 거짓: 하나의 레지스터에서 다른 레지스터로 데이터를 가장 빠르게 전송하는 방법은 병렬 전송이다
2. 병렬 전송에 비해 직렬 전송의 가장 큰 장점은 무엇인가?
3. 그림 5-47에서 레지스터의 초기값이  $X_2=0, x_1=1, x_0=0, Y_2=1, Y_1=1, Y_0=0$ 이고  $X_2$ 의 D 입력이 HIGH라고 가정하자. 네번째 시프트 펄스가 발생한 후 각 플립플롭의 출력을 구하라
4. 데이터 전송 형태 중에서 송신측이 데이터 값을 잃어버리지 않는 전송 형태는 무엇인가?

# 5-19 주파수 분할과 카운터

## ➤ 3비트 2진 카운터로 결선된 J-K 플리플롭

그림 5-48 3비트 2진 카운터 (MOD-8)로 결선된 J-K 플리플롭.



- 각 FF는 입력 주파수를 2로 나눈다
- 출력 주파수는 입력 (클럭) 주파수의 1/8이다
- 네 번째 FF는 클럭의 1/16 주파수를 만든다
- 이 회로는 바이너리 카운터 역할을 한다
- 출력은 000에서 111 또는 010에서 710까지 계산된다
- 카운터에서 가능한 상태 수는 모듈러스 또는 MOD 수이다

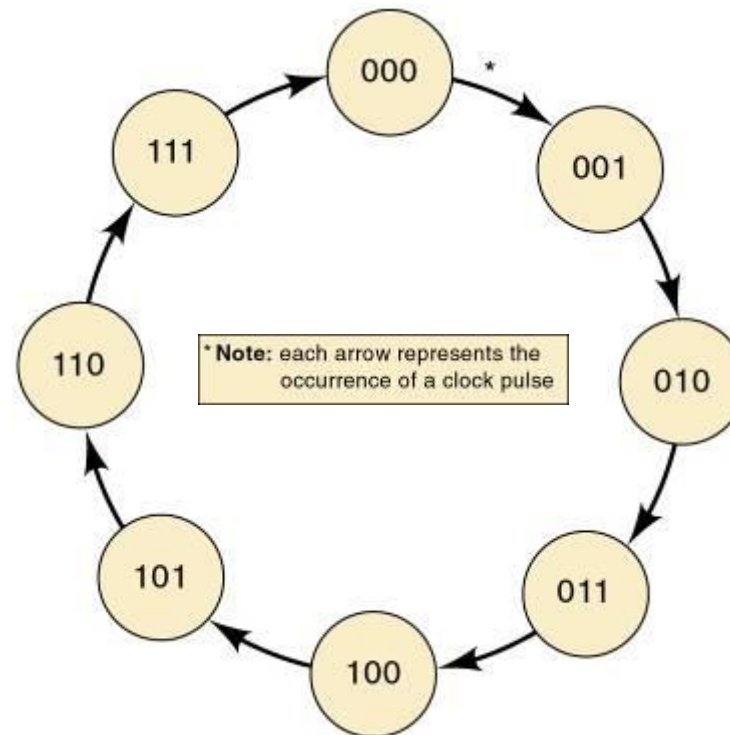
# 5-19 주파수 분할과 카운터

## ➤ 상태 천이도

### ▪ MOD-8 (2<sup>3</sup>) 카운터

- 다른 FF가 추가되면 MOD-16 (2<sup>4</sup>) 카운터가 된다

2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>	
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	
0	0	0	Before applying clock pulses
0	0	1	After pulse #1
0	1	0	After pulse #2
0	1	1	After pulse #3
1	0	0	After pulse #4
1	0	1	After pulse #5
1	1	0	After pulse #6
1	1	1	After pulse #7
0	0	0	After pulse #8 recycles to 000
0	0	1	After pulse #9
0	1	0	After pulse #10
0	1	1	After pulse #11
.	.	.	.
.	.	.	.
.	.	.	.



# 5-19 주파수 분할과 카운터

## ➤ 예제 5-14

그림 5-48 MOD-8 카운터가 101 상태에 있다고 가정하자. 13번째의 펄스가 인가된후 카운터의 상태는 어떠한가

# 5-19 주파수 분할과 카운터

## ➤ 예제 5-15

6개의 플립플롭(Q5 Q4 Q3 Q2 Q1 Q0)을 이용하여 그림 5-48과 같이 구성된 회로를 고려해 보자

(a) 카운터의 모드 수를 구하라

(b) 1 MHz의 입력 클럭 주파수를 인가 하였을때 마지막 플립플롭(Q5)의 출력 주파수를 구하라

(c) 이 카운터의 카운트 상태의 범위는?

(d) 초기 상태가 000000라고 할 때, 129번째 펄스후의 카운터 상태는 무엇인가?

# 5-19 주파수 분할과 카운터

## ➤ 학습성과 평가문제

1. 20 KHz 클럭 신호를  $J+K+1$ 인 J-K 플립플롭에 인가하였을 때 플립플롭의 출력 주파수는 얼마인가?
2. 0에서  $255_{10}$  까지 카운트 하는데 필요한 플립플롭의 수는 몇 개인가?
3. 2번 질문의 카운터 모드 수는 얼마인가?
4. 입력 클럭 주파수가 512KHz 일 때 여덟번째 플립플롭의 출력 주파수는 얼마인가?
5. 이 카운터가 00000000 상태에서 출발할 때 520번째 펄스 후의 논리 상태는 무엇인가?

# 5-20 타이밍 제약이 있는 플립플롭의 응용

- 회전하는 샤프트를 추적하기 위해 간단한 D 플립 플롭을 사용하는 것은 부적절합니다
- 목표는 항상 샤프트의 물리적 위치를 나타내는 카운터의 출력에 2 진수를 생성하는 것입니다.
- 이것은 시계 방향으로 회전하는 동안 카운트 업하고 그림 5-51과 같이 시계 반대 방향으로 회전 할 때 카운트 다운하여 개념적으로 달성 할 수 있습니다
- 그러나 실제적인 현실은 이 시점에서 샤프트가 반전되어 카운트와 샤프트 위치가 오정렬 될 수 있다는 것입니다.

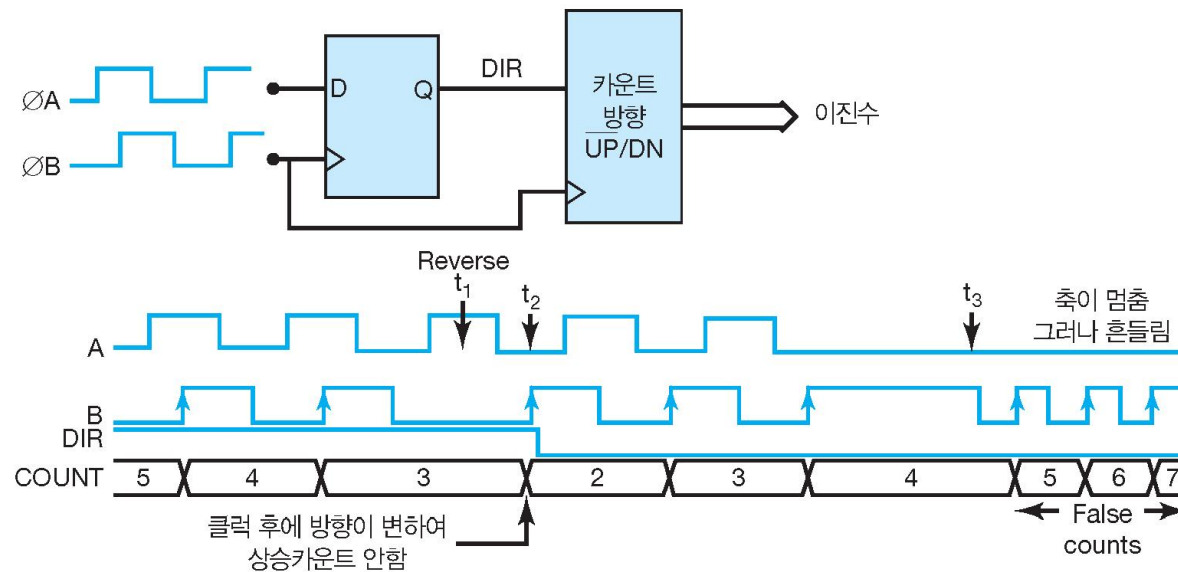
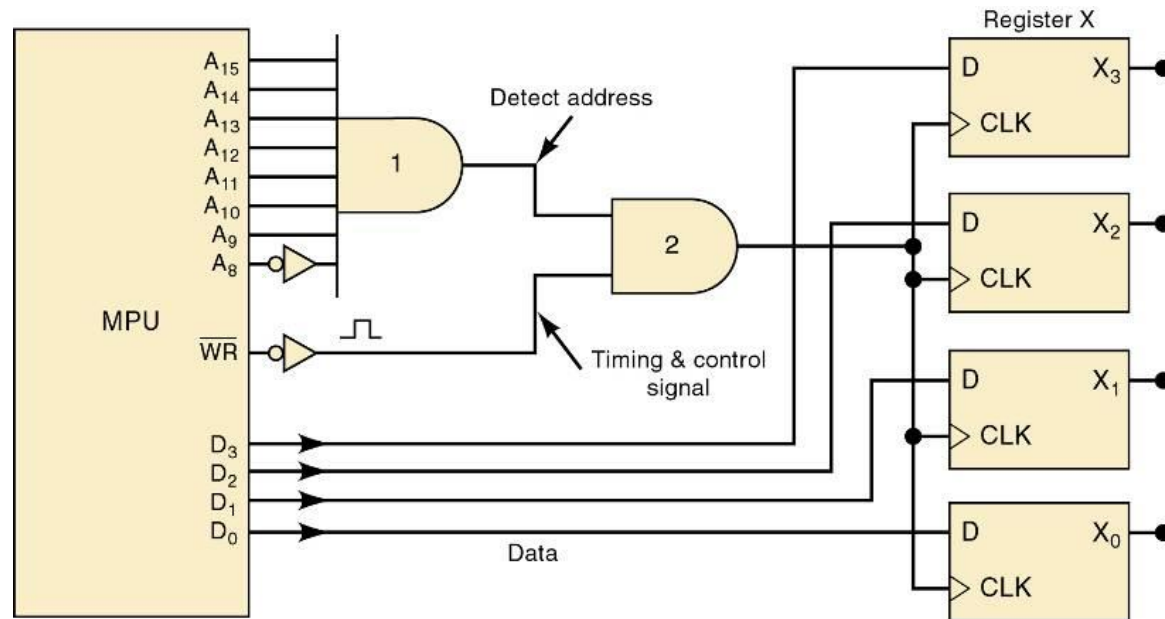


그림 5-51 단순한 시퀀스 탐지기를 사용하여 절대 축 인코더를 만들 때 문제.

# 5-21 마이크로 컴퓨터의 응용

## ➤ 외부 레지스터로 2진 데이터를 전송하는 마이크로프로세서

- 2진수를 D3에서 D0 까지의 데이터를 출력포트로 출력 한다
- 데이터를 수신하는 레지스터 X를 선택하기 위하여 적당한 주소를 A15에서 A8 까지의 포트로 출력 한다
- 데이터와 주소 출력이 안정화 되면 MPU는 외부 레지스터 X로 쓰기 펄스 WR을 발생시켜 이 레지스터에 클럭을 제공하고, 레지스터 X로 데이터를 병렬 전송한다



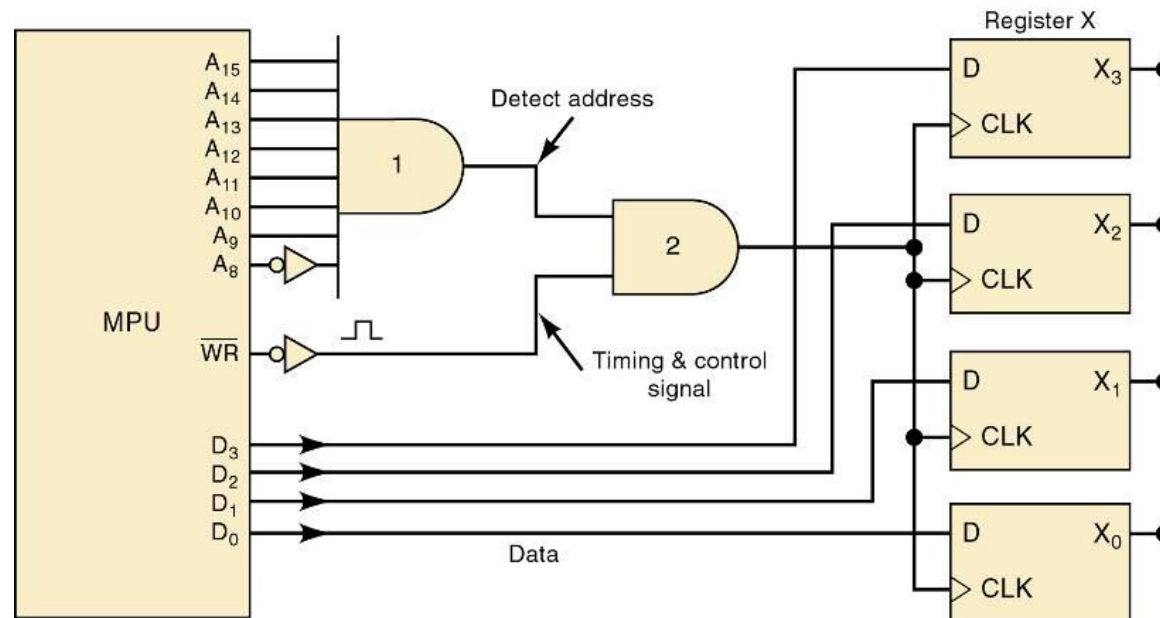
# 5-21 마이크로 컴퓨터의 응용

## ➤ 예제 5-16

(a) 그림 5-58dml MCU가 데이터를 레지스터 X로 전송하기 위하여 발생하는 주소 코드는?

(b)  $X_3-X_0 = 0110$ ,  $A_{15}\sim A_8 = 11111111$ ,  $D_3\sim D_0 = 1011$ 이라고 가정하자

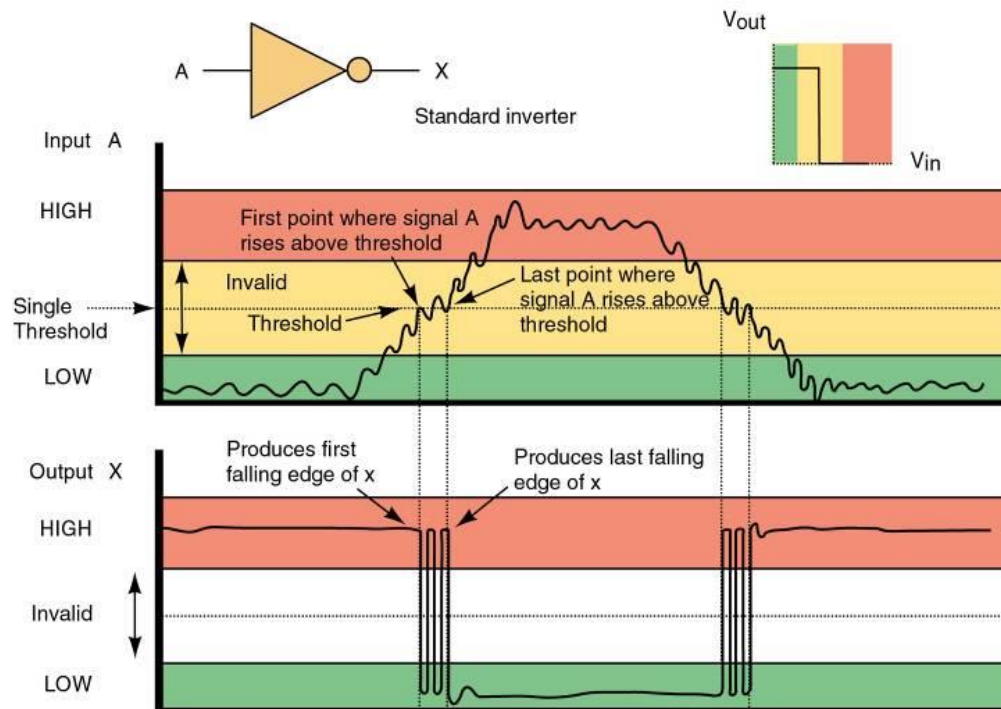
$\overline{WR}$  펄스 발생한 후의 레지스터 X의 값은 무엇인가?



# 5-22 슈미트-트리거 소자

## ➤ 슈미트-트리거 회로

- 플립플롭으로 분류되지 않는다
- 어떠한 특수한 상황에서 유용하게 쓰일 수 있는 일종의 기억 특성을 가지고 있다

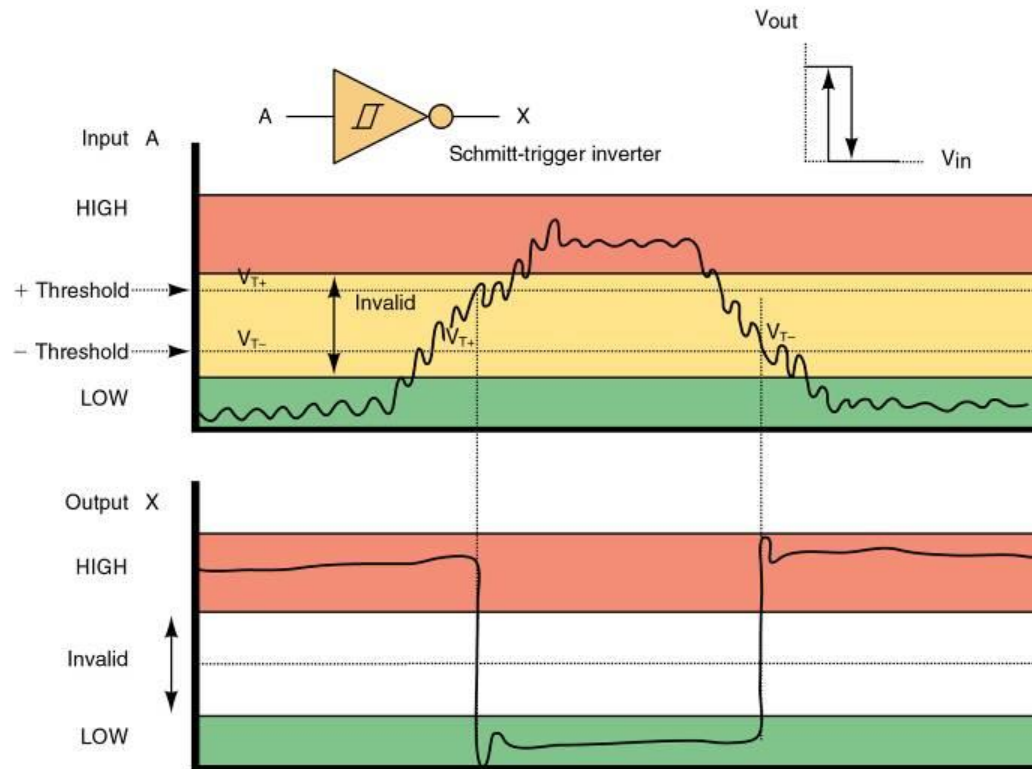


- 표준 인버터가 비교적 느린 천이속도를 갖는 논리 입력으로 구동된다
- 이러한 신호의 천이가 최대 허용치를 초과하면, 입력이 결정되지 않은 범위를 통과할 때 논리 게이트와 인버터의 출력이 발진하게 된다
- 이런 입력 조건은 플립플롭에도 적절치 못한 트리거를 일으킬 수 있다.

# 5-22 슈미트-트리거 소자

## ➤ 슈미트-트리거 회로

- 입력단에 슈미트-트리거를 갖는 소자는 천천히 변하는 신호를 입력 받을 수 있도록 설계
- 발진하지 않는 출력을 발생한다



- 상승 문턱 전압(Positive going threshold voltage)  $V_{T+}$  을 초과할 때 까지 출력이 HIGH에서 LOW로 변하지 않는다
- 출력이 일단 LOW로 변하면 입력이  $V_{T+}$  아래로 떨어지더라도 출력은 변하지 않으며
- 입력이 하강 문턱 전압(Negative going threshold voltage)  $V_{T-}$  까지 내려가야 변한다

# 5-22 슈미트-트리거 소자

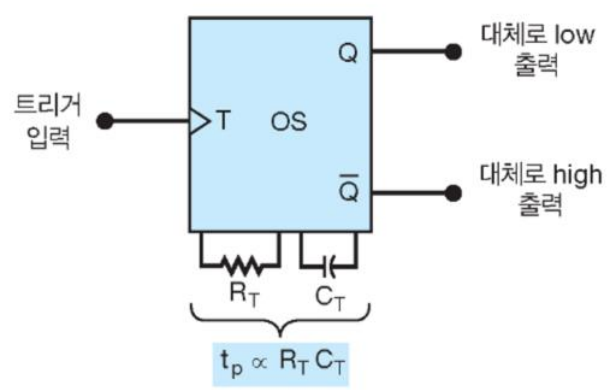
## ➤ 학습성과 평가문제

1. 느린 속도로 변하는 입력을 표준 논리 IC에 인가 하였을 때 어떤 일이 발생 하는가?
2. 슈미트-트리거 논리 소자는 표준 논리소자와 어떻게 다르게 동작 하는가?

# 5-23 단안정 멀티 바이브레이터

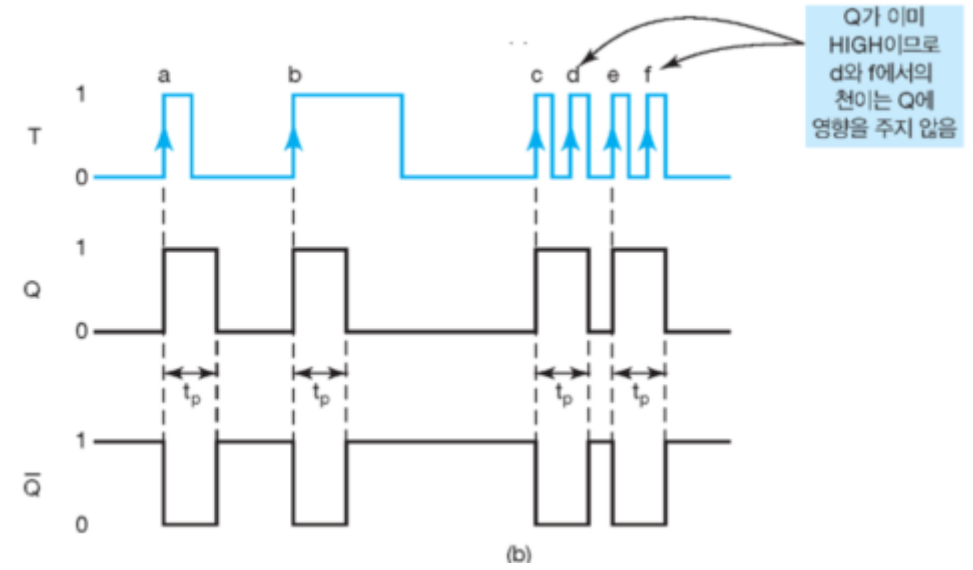
## ➤ One-Shot: OS (단안정 멀티바이브레이터)

- OS는 단지 하나의 안정된 출력 상태를 가지고 입력 신호에 의해 트리거 되기까지 그 상태를 유지한다
- 입력이 한번 트리거 되면 OS의 출력은 반대 상태로 변한다
- 출력은 준안정상태에 정해진 시간( $t_p$ ) 동안 머무는데 이  $t_p$  값은 OS 외부에 연결된 소자에 의해 정해지는 R C 시상수에 의해 결정된다



**안정 상태**  
 $Q = 0, \bar{Q} = 1$

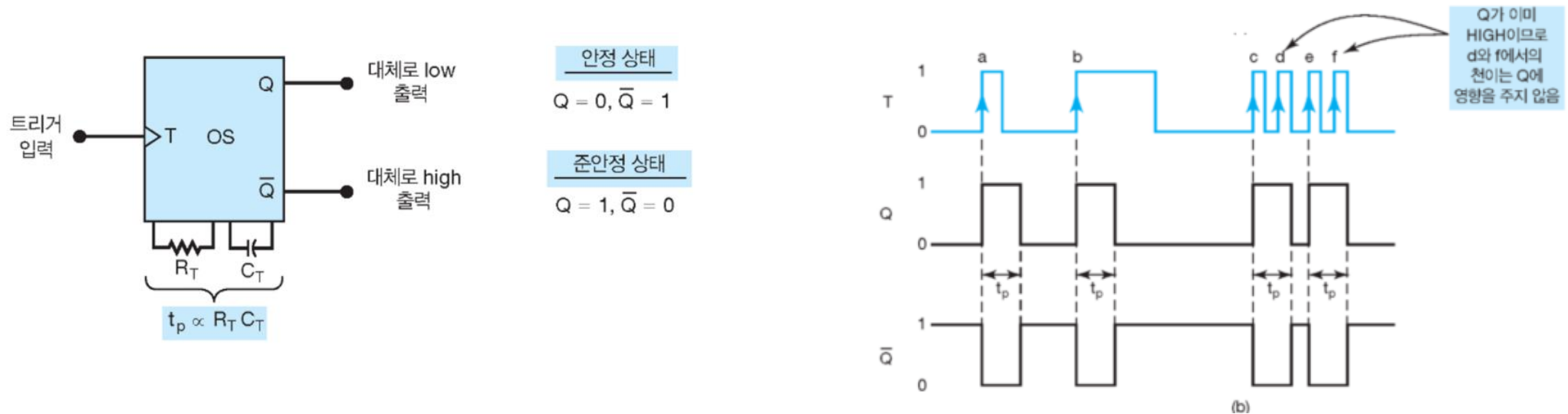
**준안정 상태**  
 $Q = 1, \bar{Q} = 0$



# 5-23 단안정 멀티 바이브레이터

## ➤ 재트리거 할 수 없는 OS

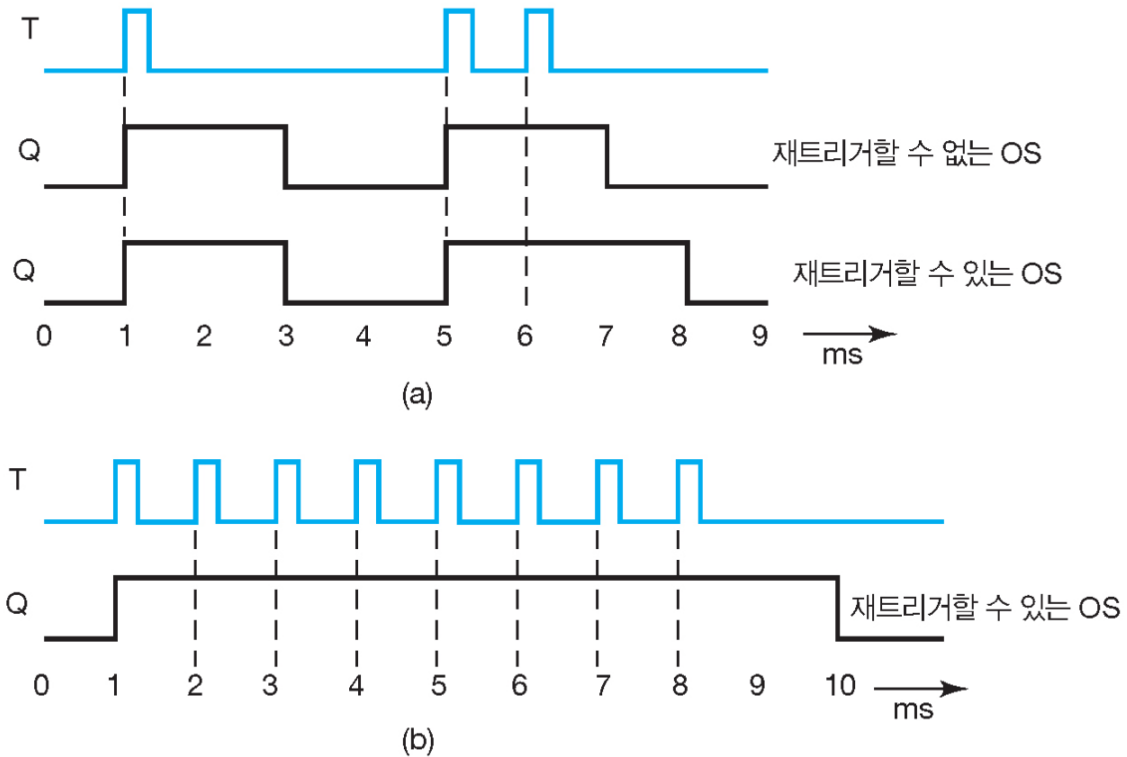
- a,b,c,e, 시점의 PGT는  $t_p$  시간 동안 OS를 준안정 상태로 트리거 시키고, 다시 안정 상태로 돌아 간다
- D와 f 시점에서의 PGT는 이미 OS가 준안정 상태에 있기 때문에 OS에 아무런 영향을 미치지 않는다. OS는 트리거 되기전에 반드시 안정 상태로 되돌아 와 있어야 한다
- OS 출력 펄스의 폭은 이력펄스의 폭에 무관하게 언제나 일정하다  
일반적으로 OS의  $t_p$ 는  $0.693 R_T C_T$ 로 주어 진다



# 5-23 단안정 멀티 바이브레이터

## ➤ 재트리거 할 수 있는 OS

- 준안정 상태에서도 다시 트리거 될 수 있고, 준안정 상태는 새로운  $t_p$  기간 동안 연장 된다

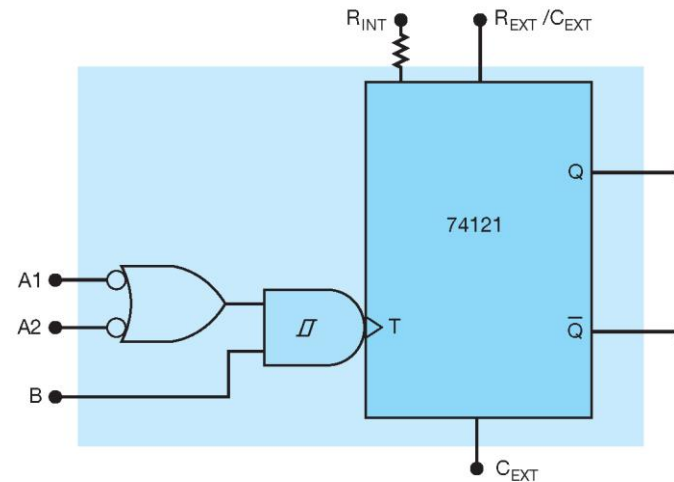


# 5-23 단안정 멀티 바이브레이터

## ➤ 실제 디바이스

- 여러 가지 방법으로 OS를 트리거 하기 위한 입력  $A_1$   $A_2$
- 입력은 느린 천이 시간을 갖는 신호로도 OS를 확실히 트리거 할 수 있는 슈미트-트리거 입력단이다
- $R_{INT}$   $R_{EXT}/C_{EXT}$   $C_{EXT}$  핀은 원하는 출력 펄스 길이를 구현하기 위한 외부 저항과 캐패시터를 연결하기 위해 사용

그림 5-62 74121 재트리거 할 수 없는 OS의 논리 기호.



# 5-23 단안정 멀티 바이브레이터

## ➤ 학습성과 평가문제

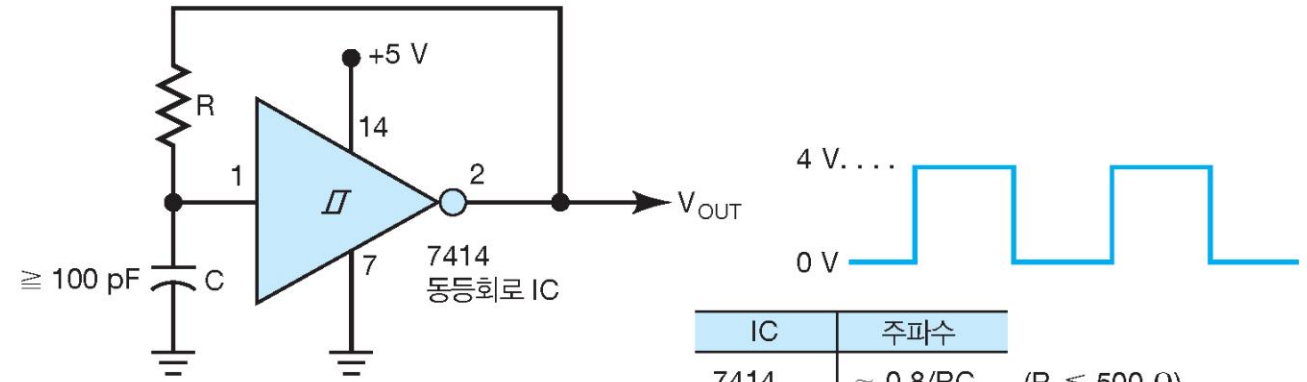
1. 트리거 펄스가 없을 때 OS 출력 상태는 어떠한가?
2. 참 또는 거짓: 재트리거 할 수 없는 OS가 준안정 상태에서 트리거 되면 출력에는 아무런 영향을 미치지 않는다
3. 무엇이 OS의  $t_p$  값을 결정하는가?
4. 재트리거 할 수 있는 OS 가 재트리거 할 수 없는 OS와 어떻게 달리 동작하는지 설명하라

# 5-24 클럭 발생 회로

## ➤ 슈미트-트리거 발진기

- 슈미트-트리거 인버터로 발진기 구성
- $V_{out}$  신호는 R과 C 값에 의해 주파수가 결정되는 구형파이다
  - 각 디바이스에 대한 저항값에 최대 제한값이 있음에 유의
  - 만약 R이 범위를 넘는다면 회로는 제대로 동작하지 않는다

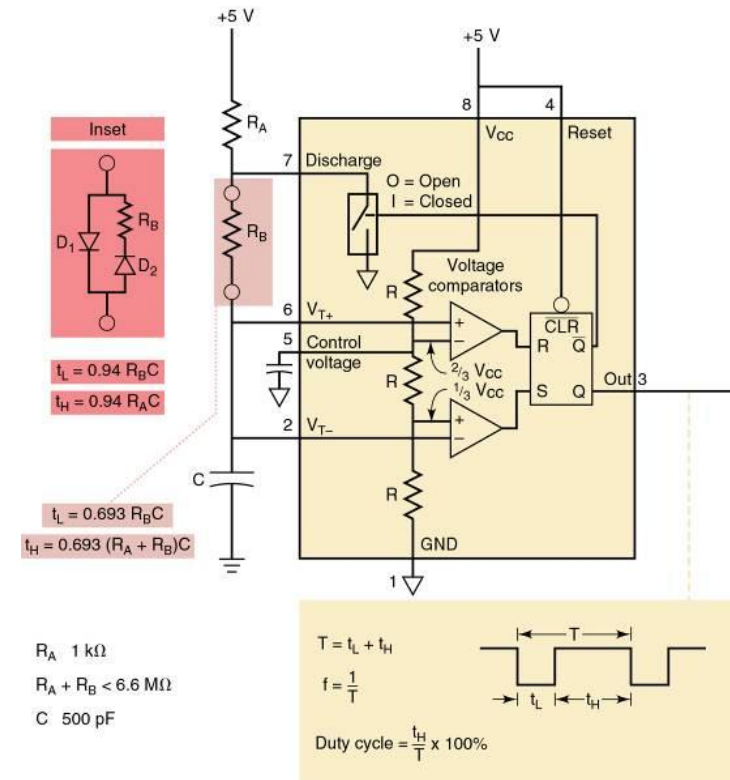
그림 5-63 7414 인버터를 이용한 슈미트-트리거 발진기. 7413 슈미트-트리거 NAND를 사용할 수도 있다.



IC	주파수
7414	$\approx 0.8/RC$ ( $R \leq 500 \Omega$ )
74LS14	$\approx 0.8/RC$ ( $R \leq 2 \text{ k}\Omega$ )
74HC14	$\approx 1.2/RC$ ( $R \leq 10 \text{ M}\Omega$ )

# 5-24 클럭 발생 회로

- 비안정 멀티바이브레이터로 사용되는 555 타이머
  - 2개의 전압 비교기와 1개의 S-R 래치로 구성
    - 전압 비교기는 1입력 전압이 2입력 전압보다 크면 HIGH를 출력하는 소자
  - 전압 비교기 출력이 HIGH가 되면 S-R 래치가 리셋 되고 출력핀 (3)이 LOW 가 된다



# 5-24 클럭 발생 회로

## ➤ 예제 5-17

$C=0.001\mu\text{F}$ ,  $R_1=2.2\text{k}\Omega$ ,  $R_B=100\text{k}\Omega$  일때 555 비안정 멀티바이브레이터 출력의 듀티 사이클과 주파수를 계산하라

# 5-24 클럭 발생 회로

## ➤ 예제 5-18

그림 5-64 처럼  $R_B$ 와 함께 다이오드를 사용하여 555을의 출력 파형이 1KHz에 25% 듀티 사이클이 되도록  $R_A$ 와  $R_B$  값을 계산하라. C는 0.1uF 이다

# 5-24 클럭 발생 회로

## ➤ 수정-제어 클럭 발진기

- 수정 진동자라는 매우 안정되고 정확한 소자를 사용
- 10KHz에서 80MHz까지의 주파수를 발생할 수 있다
- 수정 진동자 클럭 발진 회로는 모든 마이크로 프로세서 시스템과 마이크로 컴퓨터 그리고 정확한 시간 간격으로 발생하도록 클럭 신호가 쓰이는 응용분야에 사용된다

# 5-24 클럭 발생 회로

## ➤ 학습성과 평가문제

1.  $R=10\text{Kohm}$ ,  $C=0.05\mu\text{F}$ 와 74HC14를 이용한 슈미트-트리거 발진기의 발진 주파수를 구하라
2.  $R_A=R_B=2.2\text{kohm}$ ,  $C=2000\text{pF}$ 를 갖는 555발진기의 주파수와 듀티 사이클을 계산하라
3. 수정 진동자 클럭 발진기가 RC 제어 발진기에 비해 장점은 무엇인가?

# 감사합니다 !



for details about this presentation, please email to [sschoi@wsu.ac.kr](mailto:sschoi@wsu.ac.kr)