



# [J02122] 컴퓨터구조

2022년 1학기

상명대학교 소프트웨어학과 박희민

5.1 래치

5.2 플립플롭

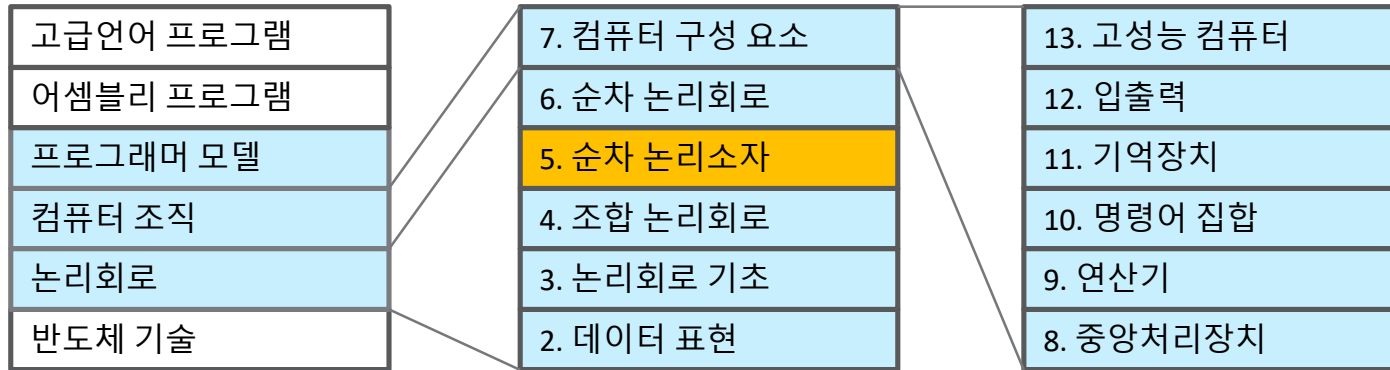
5.3 레지스터와 카운터

5.4 요약

2022-03-30

CHAP05 순차논리소자

# 5. 순차 논리소자

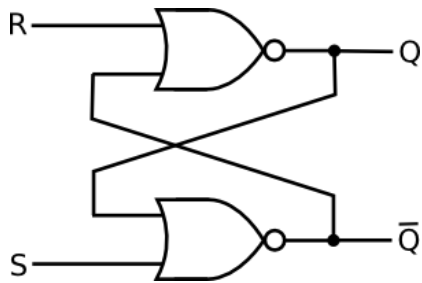


- 학습 목표
  - 데이터를 저장하는 기억소자의 기능을 설명할 수 있다.
  - 레지스터, 시프트 레지스터, 카운터의 기능을 설명할 수 있다.
- 내용
  - 5.1 래치
  - 5.2 플립플롭
  - 5.3 레지스터와 카운터
  - 5.4 요약

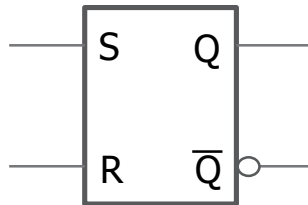
# 5.1 래치

- 래치(걸쇠, latch)
  - 제어 입력에 따라 0 또는 1 상태를 유지하는 기억 소자
- 내용
  - 5.1.1 NOR 래치
  - 5.1.2 NAND 래치
  - 5.1.3 클럭 제어 래치

# 5.1.1 NOR 래치

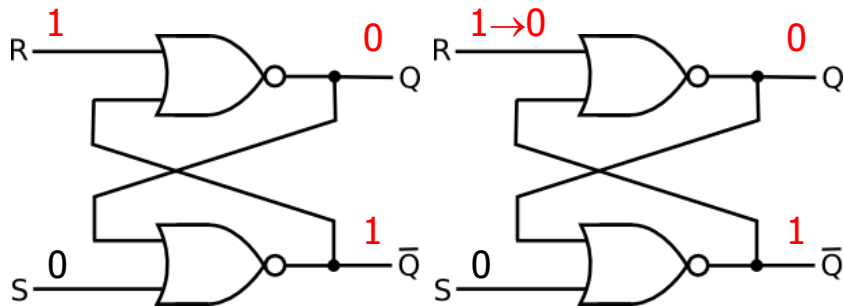


(a) NOR 래치 구조



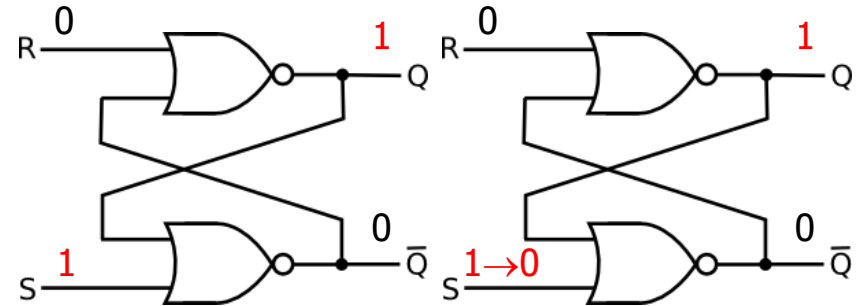
(b) 블록도

S	R	Q	동작 설명
0	0		
0	1		
1	0		
1	1		



(a) 리셋

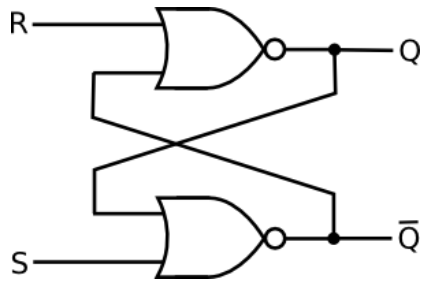
(b) 리셋 유지



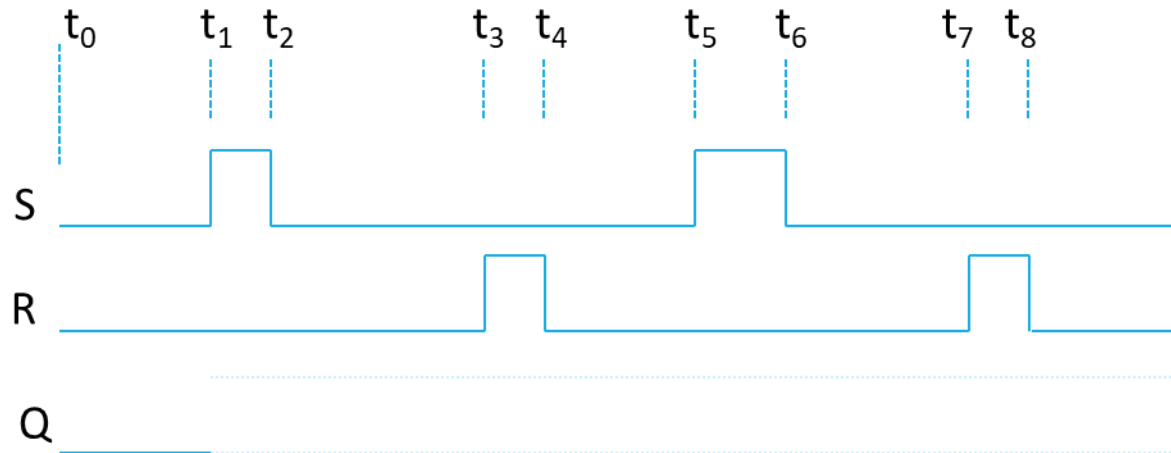
(c) 셋

(d) 셋 유지

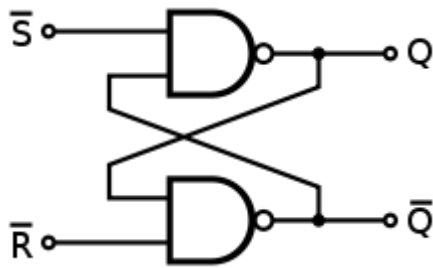
# NOR 래치 동작 예



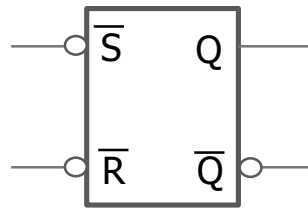
S	R	Q	동작 설명
0	0	$Q_0$	유지
0	1	0	리셋
1	0	1	셋
1	1	NA	사용 불허



# 5.1.2 NAND 래치

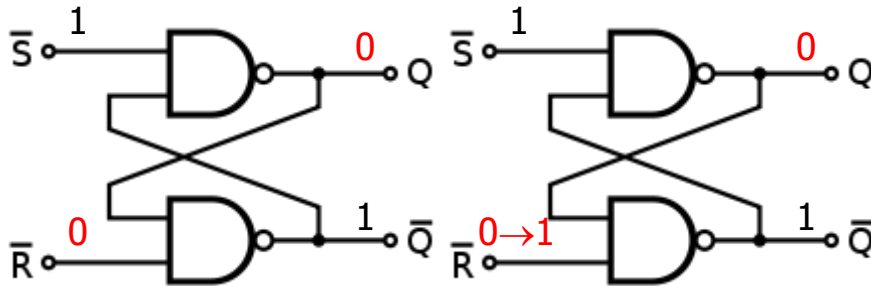


(a) NAND 래치 구조



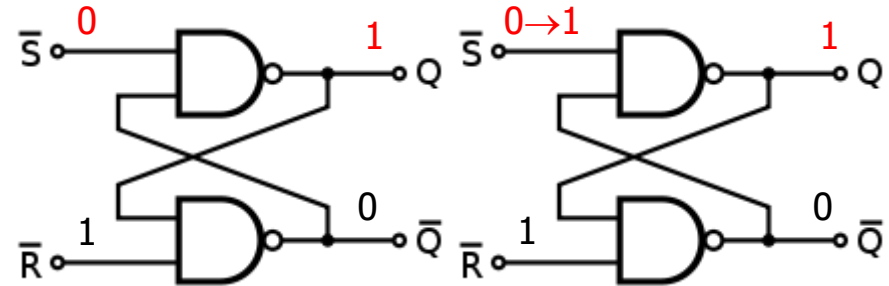
(b) 블록도

/S	/R	Q	동작 설명
1	1		
1	0		
0	1		
0	0		



(a) 리셋

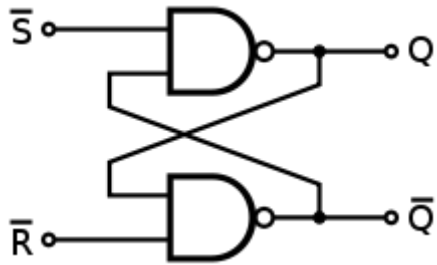
(b) 리셋 유지



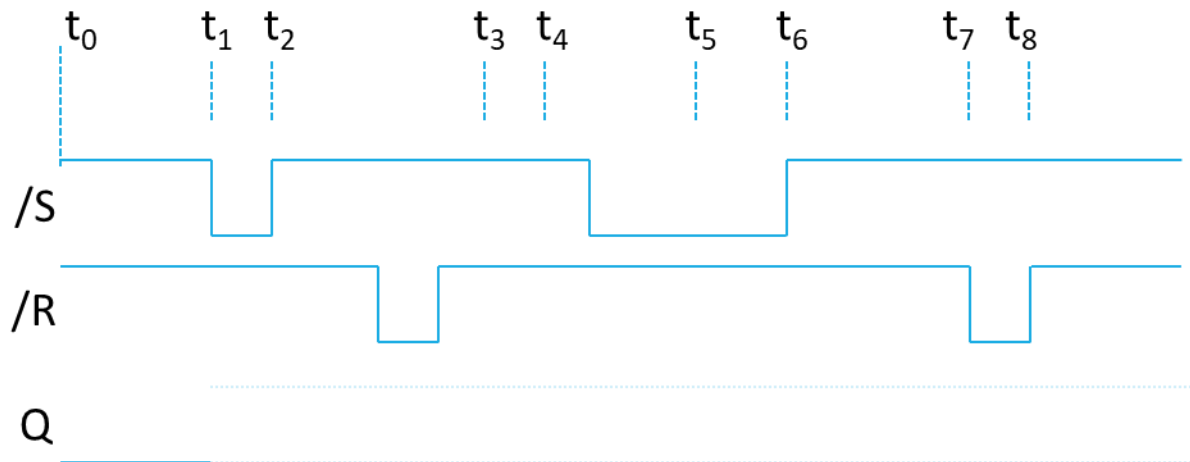
(c) 셋

(d) 셋 유지

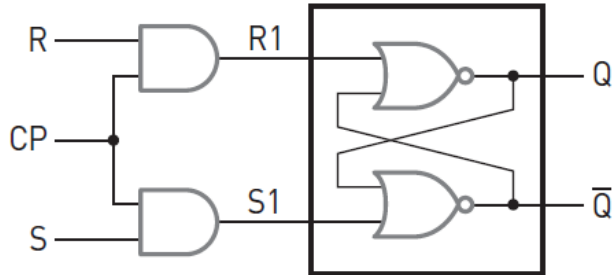
# NAND 래치 동작 예



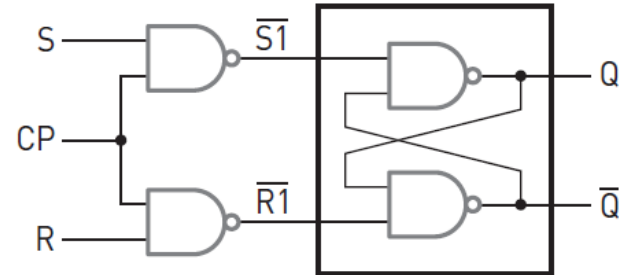
/S	/R	Q	동작 설명
1	1	$Q_0$	유지
1	0	0	리셋
0	1	1	셋
0	0	NA	사용 불허



# 5.1.3 클럭 제어 래치



(a) 클럭 제어 NOR 래치



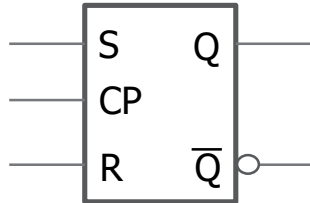
(b) 클럭 제어 NAND 래치

CP	S	R	S1	R1	Q
0	x	x			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

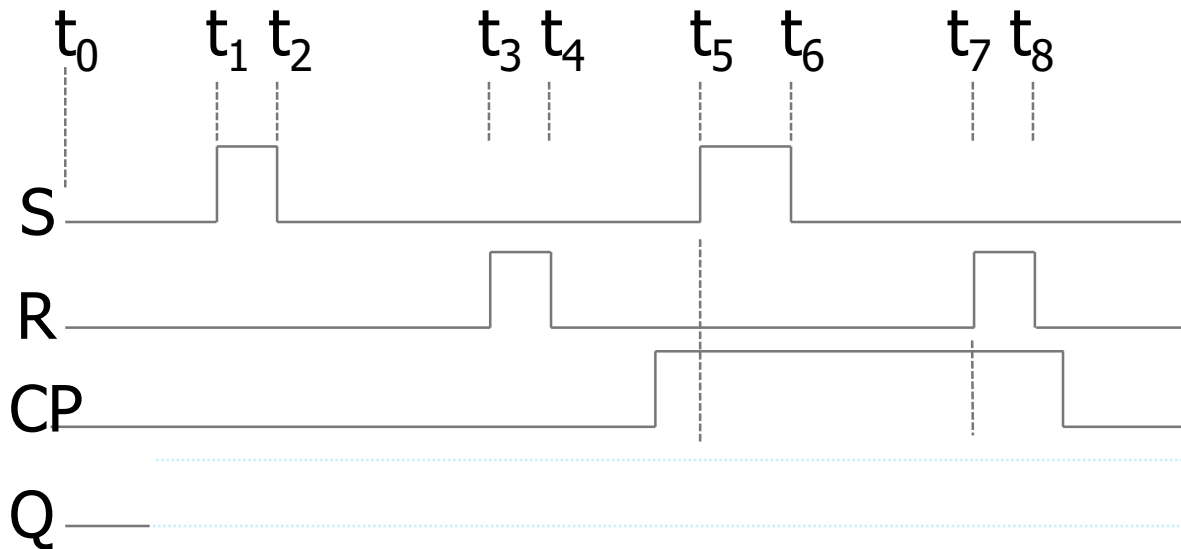
CP	S	R	/S1	/R1	Q
0	x	x			
1	0	0			
1	0	1			
1	1	0			
1	1	1			



# 클럭 제어 SR latch의 동작



CP	S	R	Q
0	x	x	$Q_0$
1	0	0	$Q_0$
1	0	1	0
1	1	0	1
1	1	1	사용 불허



# 5.1 래치 요약

- 래치
  - 한 비트 데이터를 저장하는 기억 소자
  - NOR 래치
  - NAND 래치
- 클럭 제어 래치
  - 클럭 펄스가 1일 때만 래치로 동작

# 5.2 플립플롭

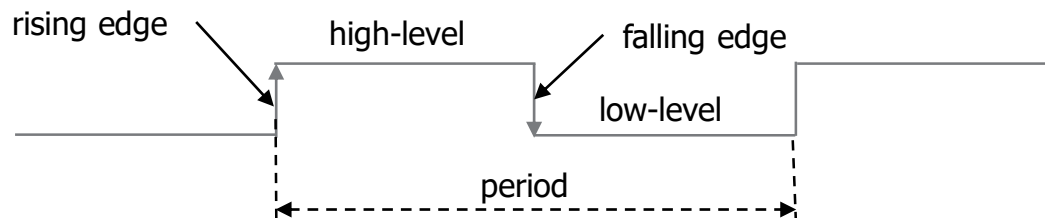
- 래치와 플립플롭
  - 래치: 클럭의 높은 레벨에서 출력 변경
  - 플립플롭: 클럭 에지에서 출력 변경
- 학습 목표
  - 플립플롭의 동작 원리를 설명할 수 있다.
  - 클럭 펄스와 비동기 리셋 신호에 따른 플립플롭의 출력을 해석할 수 있다.
- 내용
  - 5.2.1 클럭 펄스
  - 5.2.2 플립플롭
  - 5.2.3 플립플롭 종류
  - 5.2.4 비동기 리셋 입력

## 5.2.1 클럭 펄스

- 발진기 (oscillator)
  - 디지털 시스템에 클럭 펄스를 공급하는 소자
- 클럭 펄스
  - 0과 1을 무한 반복하는 신호
  - 주기(period): 한 개의 구간. 단위는 시간, 초(second)
  - 주파수(frequency) =  $1/\text{주기}$  Hz

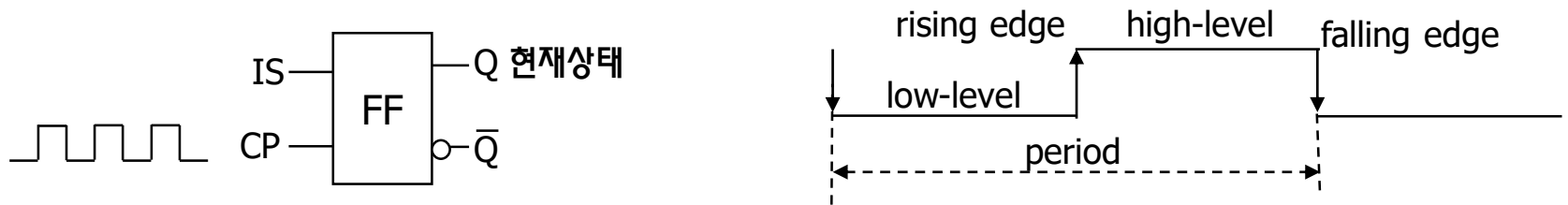


(a) 클럭 펄스



(b) 한 주기

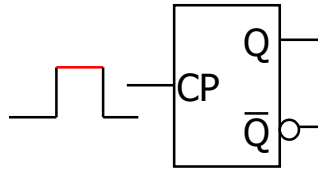
## 5.2.2 플립플롭



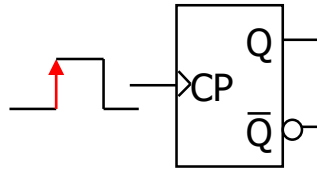
### 플립플롭의 일반 형태와 클럭 펄스(CP) 클럭 펄스의 한 주기

- 플립플롭(flip-flop)
  - 한 비트의 값을 저장하는 기억 소자
  - $0 \leftrightarrow 1$ 로 값이 변하는 모습
  - 출력 2개: Q, Q'. (항상 반대 값)
- 플립플롭의 동작을 결정하는 요인
  - 출력 변경 시간(타이밍): 클럭 펄스에 의해 결정
  - 입력 신호: 종류와 값에 따라 출력 값(다음상태, next state) 결정

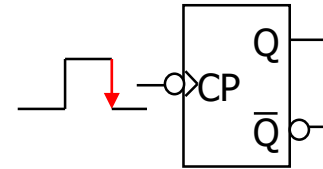
# 플립플롭 동작 타이밍



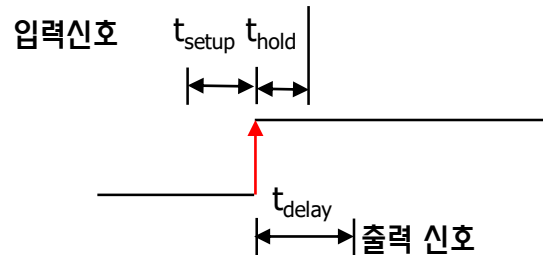
(a) Latch or Level Trigger



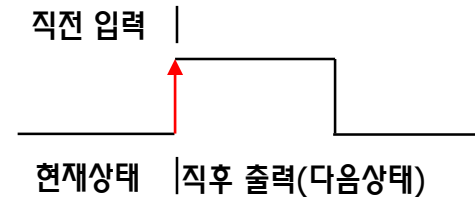
(b) Positive Edge Trigger



(c) Negative Edge Trigger



(a) 미시적 관점

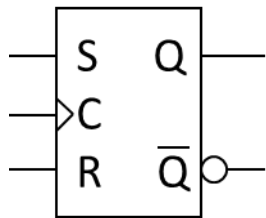


(b) 거시적 관점

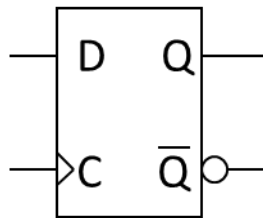
- 플립플롭의 현재상태(current state)는
  - 클럭이 인가되기 직전 입력 신호의 값에 따라
  - 클럭이 인가된 직후에 다음상태(next state)로 변한다.

## 5.2.3 플립플롭 종류

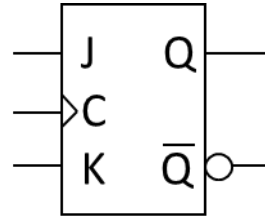
- 플립플롭 종류
  - SR-FF, D-FF, JK-FF, T-FF
- 동작 타이밍
  - 상승 에지 플립플롭과 하강 에지 플립플롭이 있을 수 있다.
  - 상승 에지 플립플롭으로 설명한다.



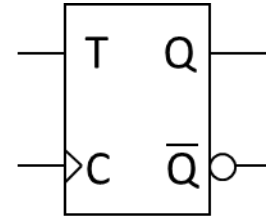
(a) SR-FF



(b) D-FF

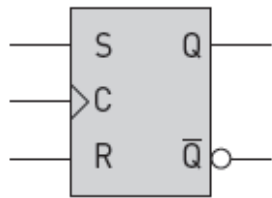


(c) JK-FF



(d) T-FF

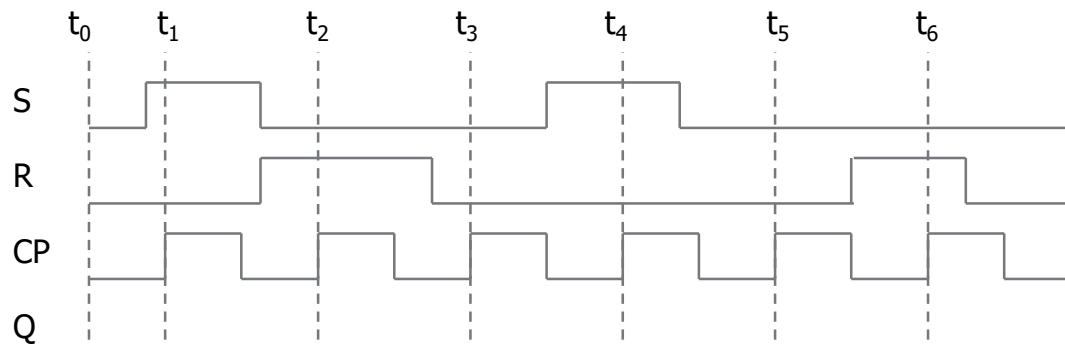
# SR-플립플롭



[그림 5-13] SR-플립플롭

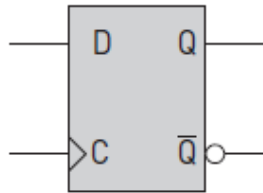
## SR-FF 특성표

CP	S	R	다음상태 Q	설명
↑	0	0		
↑	0	1		
↑	1	0		
↑	1	1		





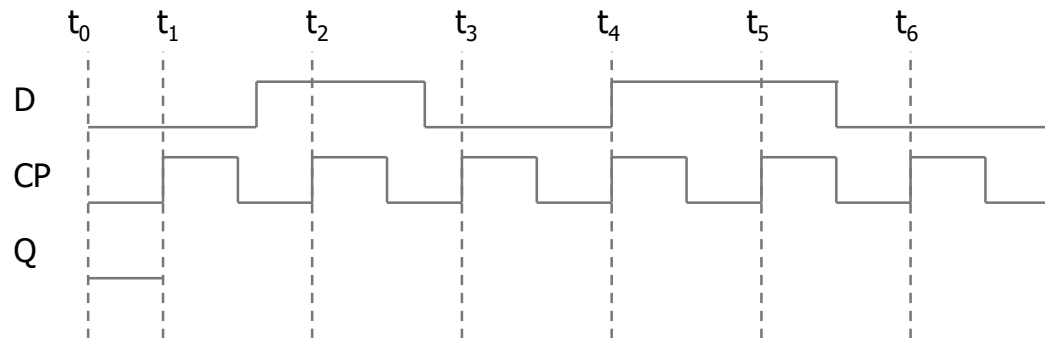
# D-플립플롭



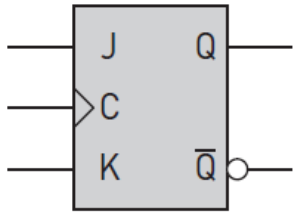
[그림 5-15] D-플립플롭

## D-FF 특성표

CP	D	다음상태 Q	설명
↑	0		
↑	1		



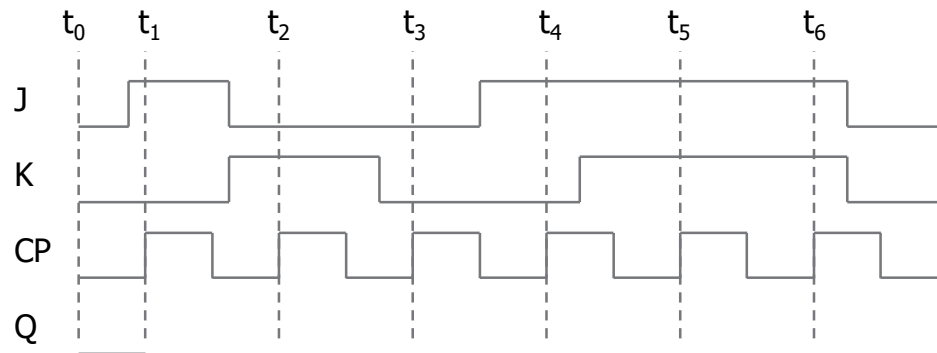
# JK-플립플롭



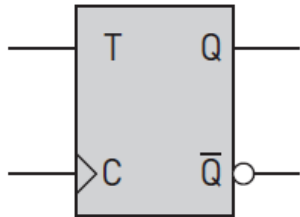
[그림 5-17] JK-플립플롭

## JK-FF 특성표

CP	J	K	다음상태 Q	설명
↑	0	0		
↑	0	1		
↑	1	0		
↑	1	1		



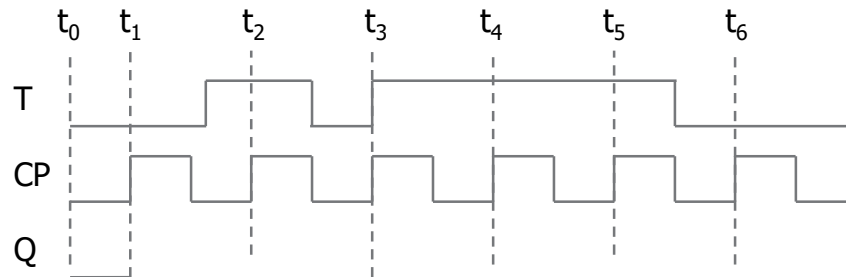
# T-플립플롭



[그림 5-19] T-플립플롭

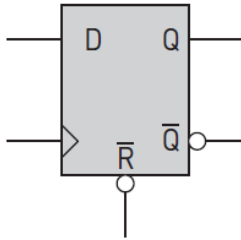
## T-FF 특성표

CP	T	다음상태 Q	설명
↑	0		
↑	1		



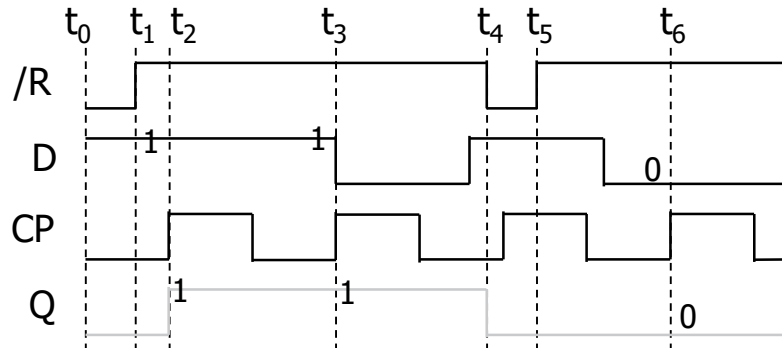
## 5.2.4 비동기 리셋 입력

- 플립플롭 제어 신호
  - 동기(synchronous): 클럭 신호에 맞춰 동작
  - 비동기(asynchronous): 클럭과 관계없이 동작

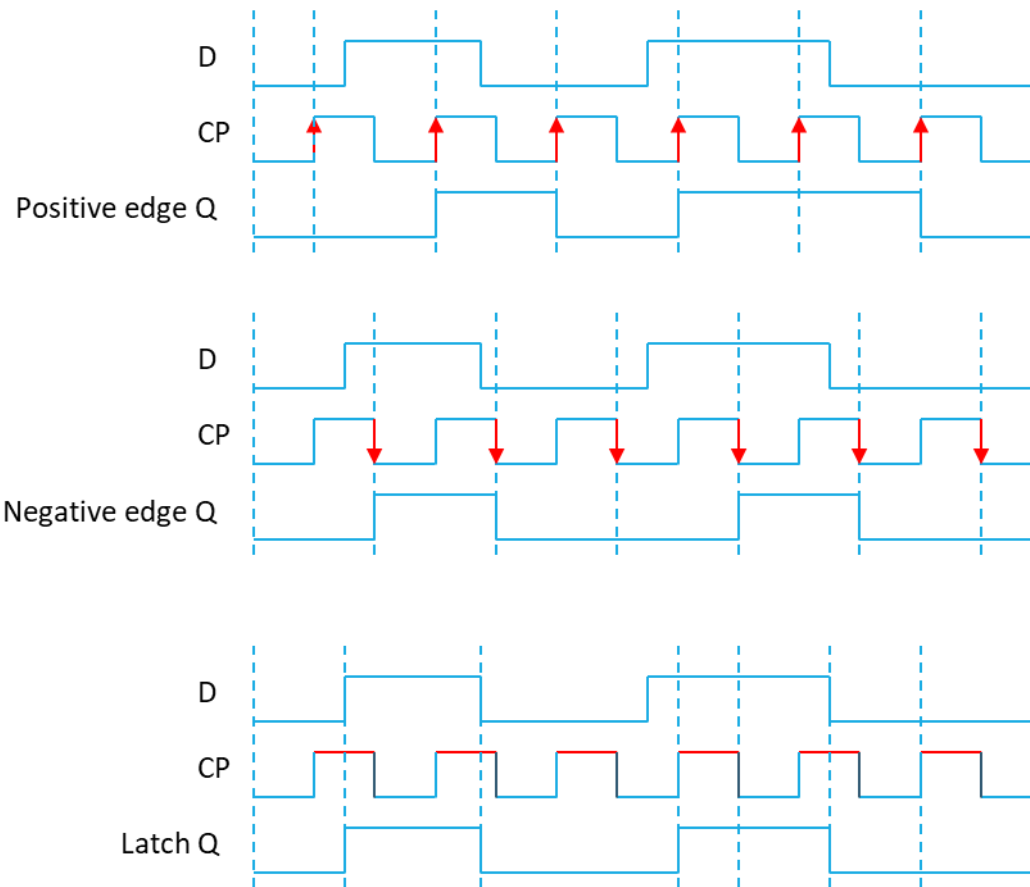


/R	CP	D	Q	설명
0	x	x	0	비동기 리셋
1	↑	0	0	클리어
1	↑	1	1	세트

[그림 5-21] 리셋 기능이 있는 D-플립플롭



# [예제 5-2] 래치와 플립플롭 출력 비교



## 5.2 플립플롭 요약

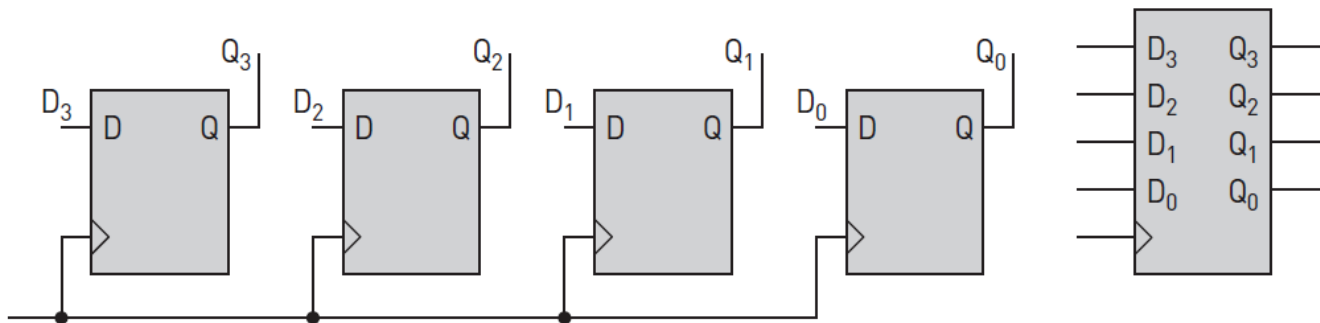
- 플립플롭
  - 클럭 펄스의 에지에서 출력이 변하는 기억소자
  - 출력:
    - 클럭펄스가 인가되는 직전 입력 신호의 값에 따라
    - 클럭펄스가 인가된 직후에 출력이 변경된다.
- 입력 신호에 따른 플립플롭 종류
  - SR-FF: 세트, 리셋
  - D-FF: 지연(delay)
  - JK-FF: 세트, 리셋, 토글
  - T-FF: 토글
- 비동기 리셋 입력
  - 클럭 펄스와 관계없이 플립플롭의 출력을 0으로 만든다.

# 5.3 레지스터와 카운터

- 레지스터
  - 여러 비트를 저장하는 기억 소자
- 학습 목표
  - 여러 가지 레지스터의 동작 원리를 이해한다.
  - 입력 신호에 대한 레지스터의 출력 신호를 해석할 수 있다.
- 내용
  - 5.3.1 레지스터
  - 5.3.2 시프트 레지스터
  - 5.3.3 카운터

## 5.3.1 레지스터

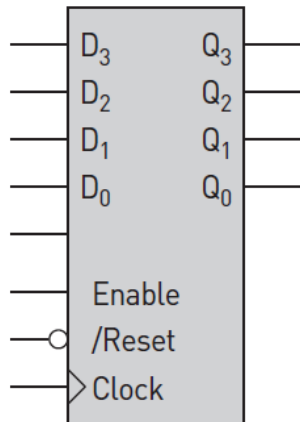
- 레지스터(register)
  - 여러 비트의 이진 정보를 저장하는 소자
  - 클럭을 공유하여 동시에 동작



[그림 5-23] 4비트 레지스터

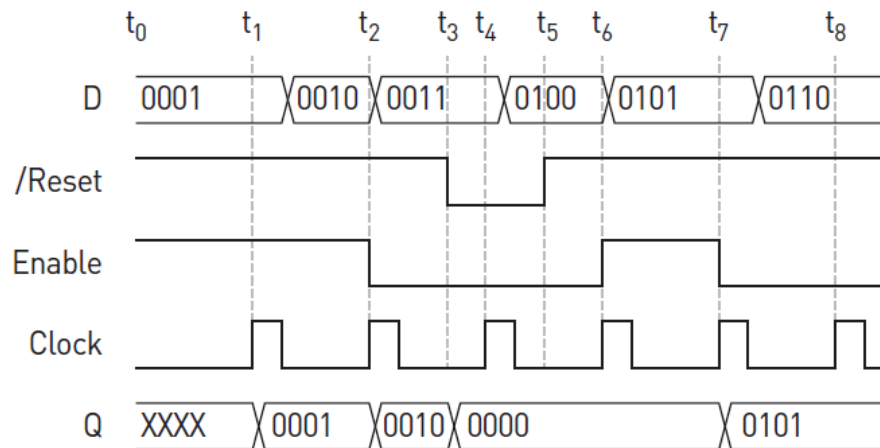


# 레지스터 동작 예



/Reset	Enable	Clock	Q	기능
0	X	X	0	비동기 리셋
1	0	X	Q <sub>0</sub>	현재상태 유지
1	1	↑	D	동기 적재

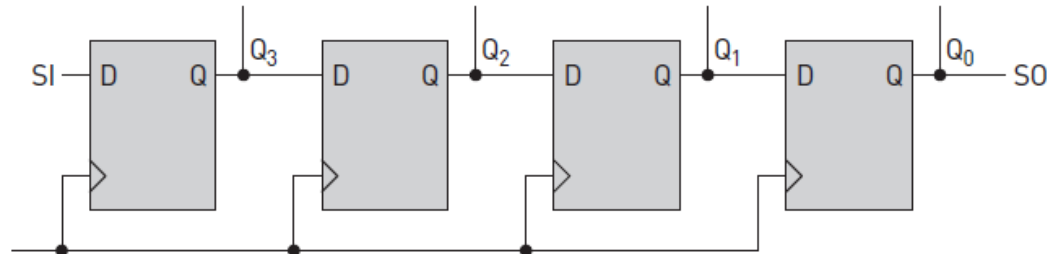
[그림 5-24] 비동기 리셋과 동기 적재 기능이 있는 4비트 레지스터



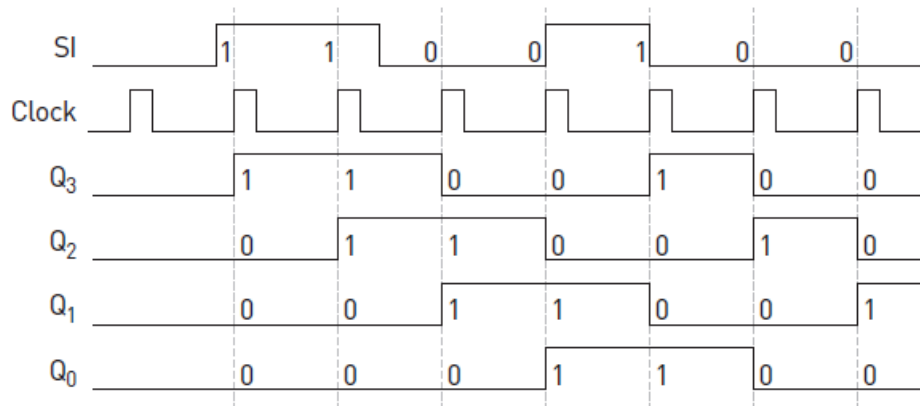
[그림 5-25] 4비트 레지스터의 동작

## 5.3.2 시프트 레지스터

- 시프트 레지스터(shift register)
  - 연결 방법에 따라  
데이터를 왼쪽 또는 오른쪽으로 자리이동하는 레지스터

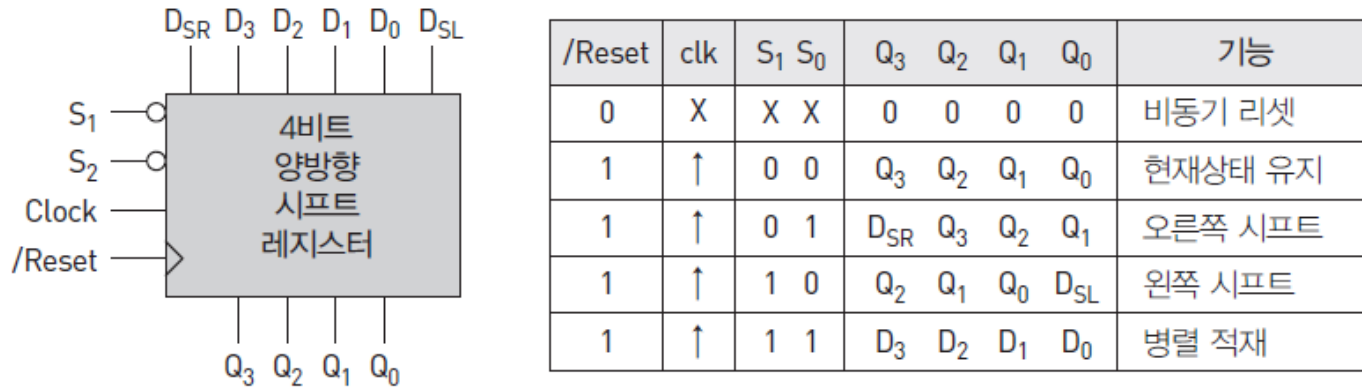


[그림 5-26] 간단한 시프트 레지스터



[그림 5-27] 시프트 레지스터의 동작

# 양방향 시프트 레지스터



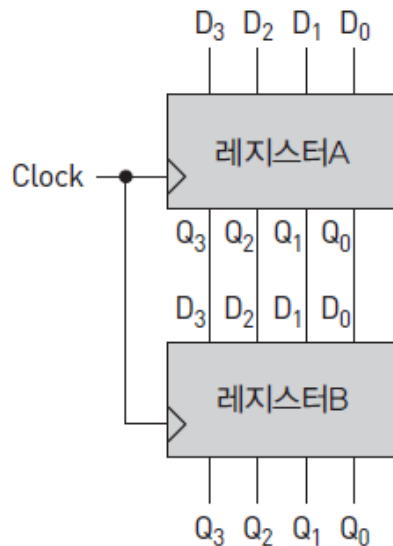
[그림 5-28] 4비트 양방향 시프트 레지스터

- [예제 5-3] 입력 조건에 따른 양방향 시프트 레지스터의 다음상태는?

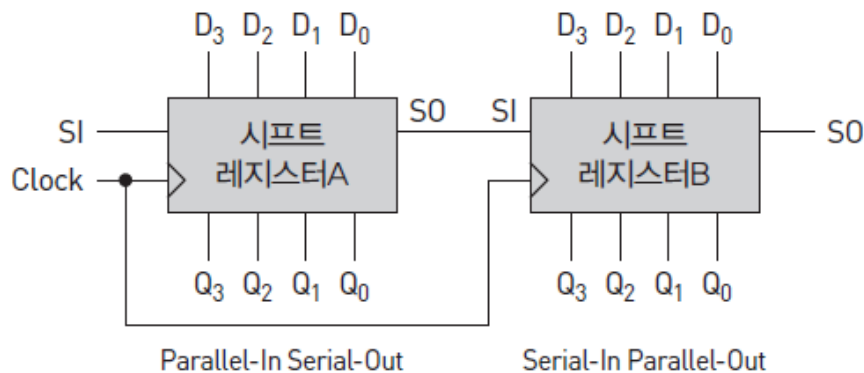
- 1) R = 1, S<sub>1</sub>S<sub>0</sub> = 00, D<sub>SR</sub> D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub> D<sub>SL</sub> = 1\_0110\_1, Q = 1001      Q = \_\_\_\_\_
- 2) R = 1, S<sub>1</sub>S<sub>0</sub> = 01, D<sub>SR</sub> D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub> D<sub>SL</sub> = 1\_0110\_1, Q = 1001      Q = \_\_\_\_\_
- 3) R = 1, S<sub>1</sub>S<sub>0</sub> = 10, D<sub>SR</sub> D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub> D<sub>SL</sub> = 1\_0110\_1, Q = 1001      Q = \_\_\_\_\_
- 4) R = 1, S<sub>1</sub>S<sub>0</sub> = 11, D<sub>SR</sub> D<sub>3</sub> D<sub>2</sub> D<sub>1</sub> D<sub>0</sub> D<sub>SL</sub> = 1\_0110\_1, Q = 1001      Q = \_\_\_\_\_

# 레지스터 전송

- 병렬 전송: 한 클럭에 데이터 전송
- 직렬 전송: 한 클럭에 한 비트씩 전송



(a) 병렬 전송



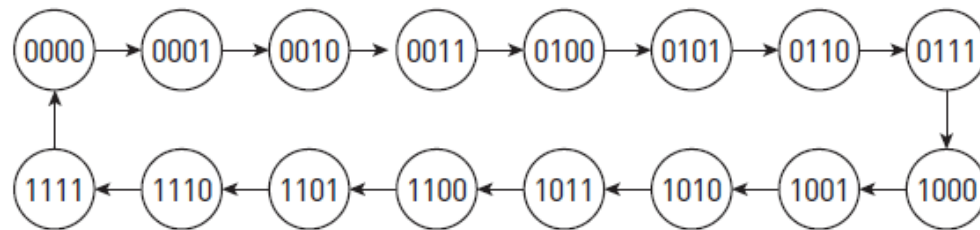
(b) 직렬 전송

[그림 5-29] 레지스터 전송

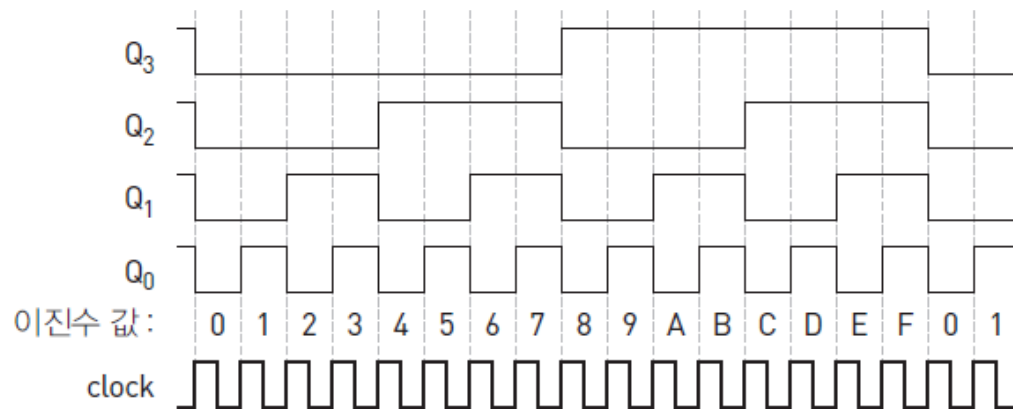
## 5.3.3 카운터

- 카운터(counter)
  - 일정한 순서로 상태를 반복하는 레지스터
  - 2진 카운터: 2진수 순서로 증가
  - N진 카운터(modulo-N counter): 0~N-1까지 세는 카운터

(a) 상태도

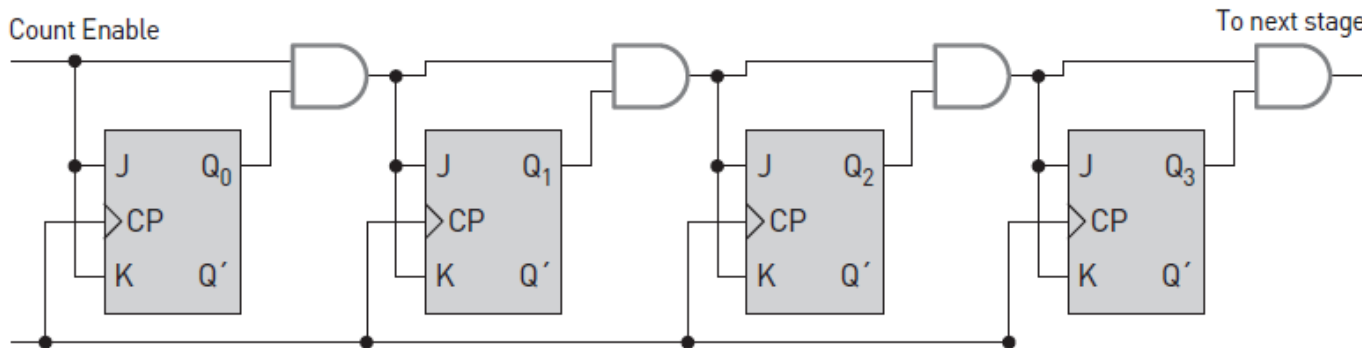
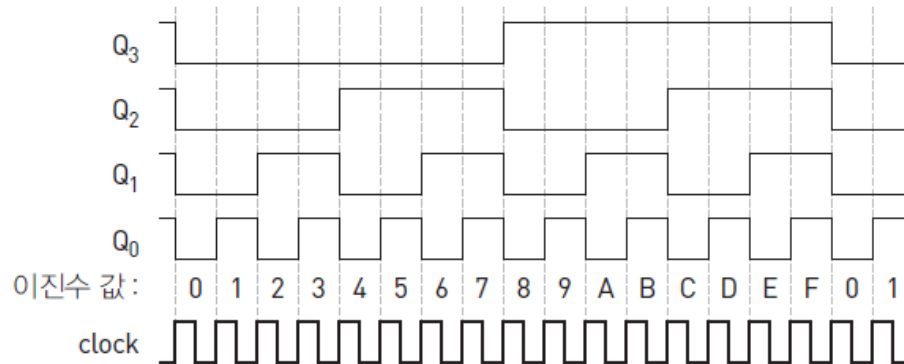


(b) 출력 파형



[그림 5-30] 4비트 2진 카운터

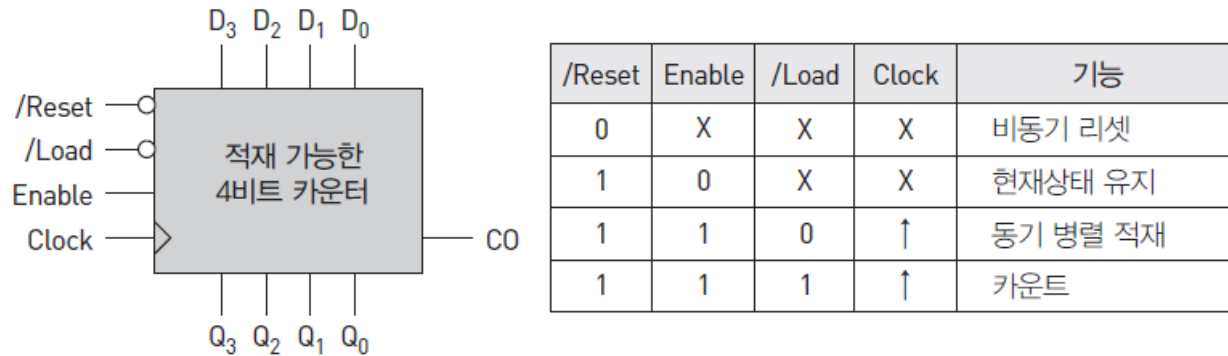
# JK-FF 이진 카운터



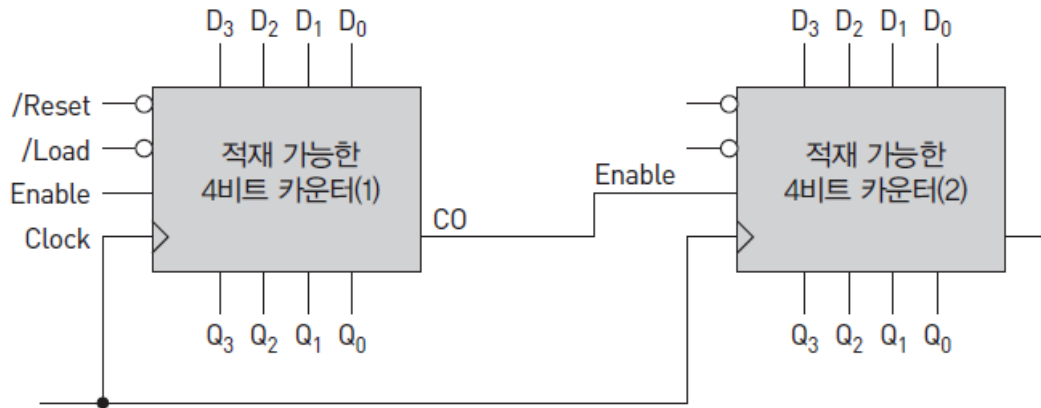
[그림 5-31] JK-플립플롭으로 만든 4비트 2진 카운터

- 상향 카운터(up counter): 이전 단의 값이 모두 1일 때, 출력 토글
- 하향 카운터(down counter): 이전 단의 값이 모두 0일 때, 출력 토글
- 상하향 카운터(up-down counter): 제어선으로 상향/하향 선택

# 병렬 적재 4 비트 카운터



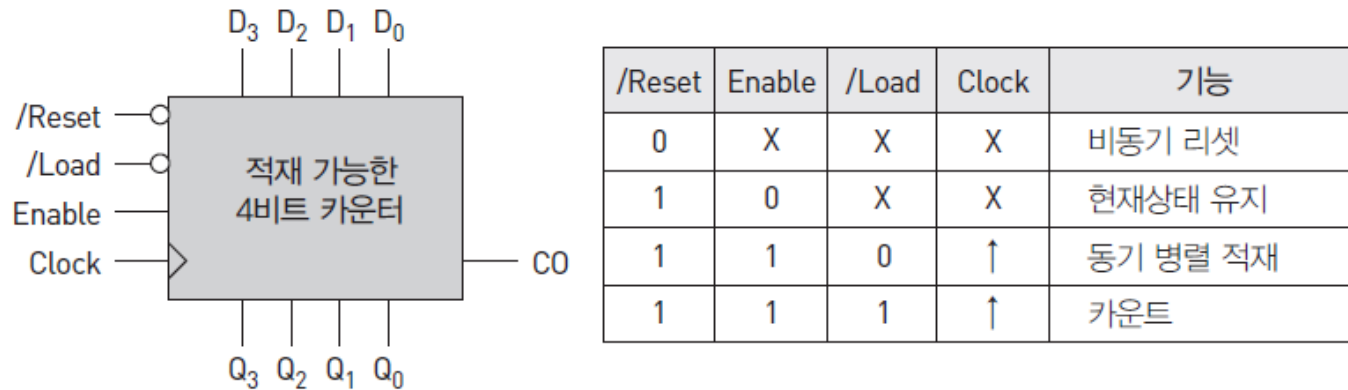
(a) 입출력 신호 및 기능



(b) 카운터 비트 확장

[그림 5-32] 병렬 적재 가능한 4비트 카운터

# 병렬 적재 4 비트 카운터



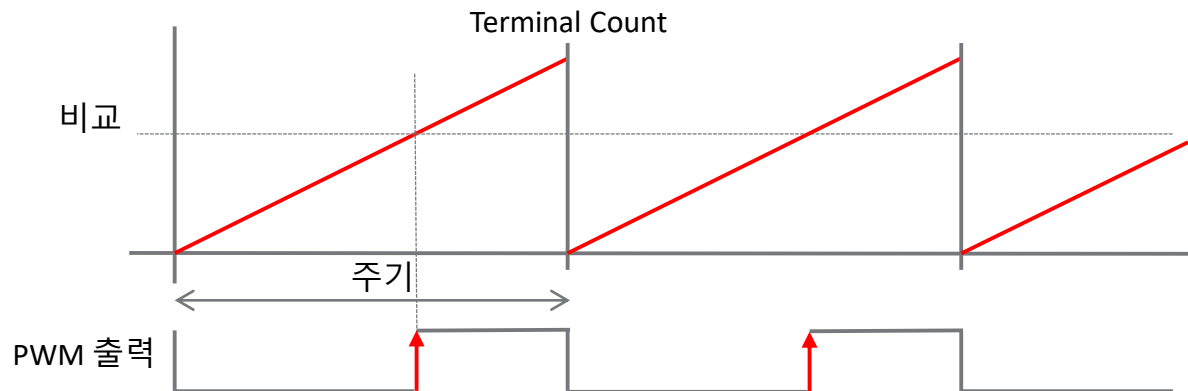
• [예제 5-4] 병렬 적재 4비트 카운터의 다음상태는?

- 1)  $/R = 0$ ,  $Enable = x$ ,  $/Load = x$ ,  $D_3D_2D_1D_0 = 1100$ ,  $Q_3Q_2Q_1Q_0 = 0101$ ,  $CO = 0$
- 2)  $/R = 1$ ,  $Enable = 0$ ,  $/Load = x$ ,  $D_3D_2D_1D_0 = 1100$ ,  $Q_3Q_2Q_1Q_0 = 0101$ ,  $CO = 0$
- 3)  $/R = 1$ ,  $Enable = 1$ ,  $/Load = 0$ ,  $D_3D_2D_1D_0 = 1100$ ,  $Q_3Q_2Q_1Q_0 = 0101$ ,  $CO = 0$
- 4)  $/R = 1$ ,  $Enable = 1$ ,  $/Load = 1$ ,  $D_3D_2D_1D_0 = 1100$ ,  $Q_3Q_2Q_1Q_0 = 0101$ ,  $CO = 0$
- 5)  $/R = 1$ ,  $Enable = 1$ ,  $/Load = 1$ ,  $D_3D_2D_1D_0 = 1100$ ,  $Q_3Q_2Q_1Q_0 = 1110$ ,  $CO = 0$



# 카운터 활용

- 카운터(counter)
  - 이벤트 발생 횟수 카운트: 펄스를 센다.
- 타이머(timer)
  - 시간 측정: 주기가 일정한 펄스를 센다.
- PWM(Pulse Width Modulation): 펄스의 듀티비 조정



## 5.3 레지스터 요약

- 레지스터
  - 플립플롭 여러 개를 연결한 논리소자.
  - 2진 데이터 저장
- 시프트 레지스터
  - 데이터를 왼쪽 또는 오른쪽으로 자리이동하는 레지스터
  - 데이터 직렬, 병렬 변환
- 카운터
  - 2진수 순서대로 상태를 변환하는 레지스터
  - 외부에서 발생한 이벤트의 발생 수를 카운트하거나, 시간을 측정한다.

# 5.4 요약

- 5.1 래치
  - NOR Latch, NAND latch
  - Gated SR Latch
- 5.2 플립플롭
  - 클록 직전 입력 값에 따라 클록 직후 출력 변화
  - SR-FF, D-FF, JK-FF, T-FF
  - 비동기 리셋: 출력 초기화
- 5.3 레지스터
  - 레지스터: n비트 데이터 저장 소자
  - 시프트 레지스터: 직렬 병렬 변환
  - 카운터: 펄스 카운트. 시간 측정
- 제6장 순차 논리회로
  - 순차 논리회로의 동작 표현 방법과 디지털 시스템의 동작