

컴퓨터과학기초

12주차

# 플립플롭

인하공업전문대학 컴퓨터정보과

이수정 교수





지난 시간

## Ch.7 조합논리회로

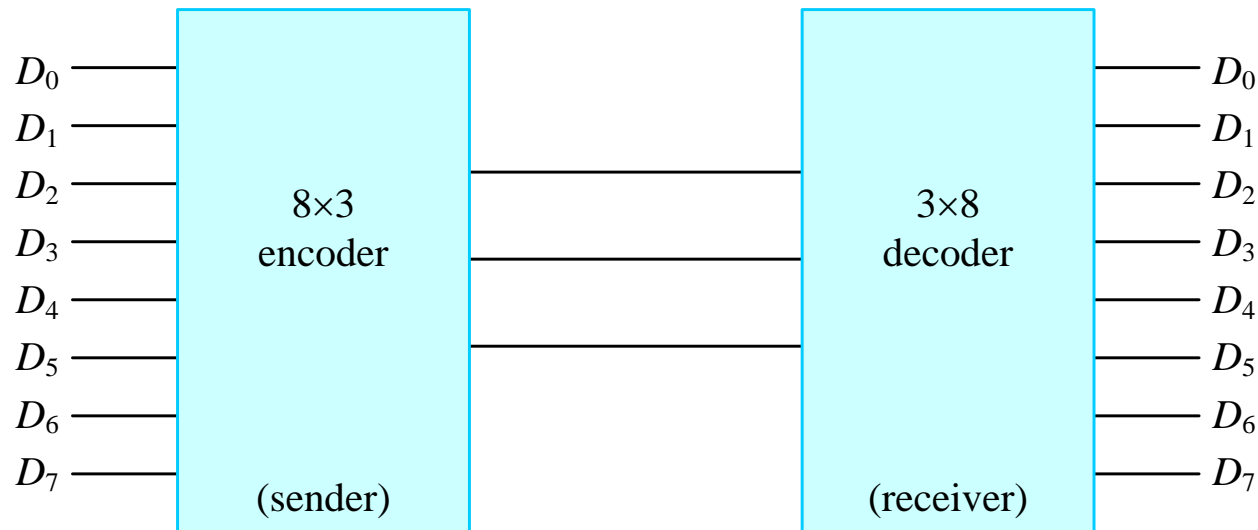
1. 가산기
2. 비교기
3. 디코더
4. 인코더
5. 멀티플렉서
6. 디멀티플렉서
7. 코드 변환기
8. 패리티 발생기/검출기



# 3. 디코더

## ■ 디코더(decoder)

- 입력선에 나타나는  $n$ 비트의 2진 코드를 최대  $2^n$ 개의 서로 다른 정보로 바꿔주는 조합논리회로
- 인에이블(enable)단자를 가지고 있는 경우는 디멀티플렉서의 기능도 수행



<디코더와 인코더의 기능>

# 3. 디코더

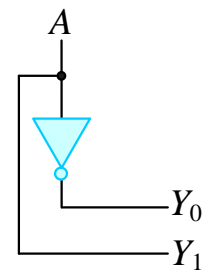
## 1) 1×2 디코더

- 1개의 입력에 따라서 2개의 출력 중 하나가 선택

입력	출력	
A	$Y_1$	$Y_0$
0	0	1
1	1	0

$$Y_0 = \bar{A} \quad Y_1 = A$$

<진리표와 논리식>



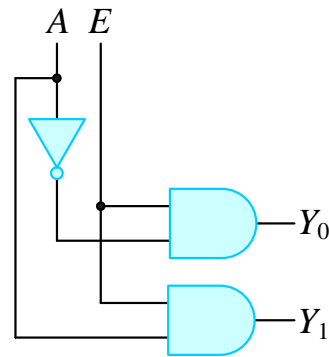
<회로도>

- 인에이블이 있는 1×2 디코더

입력		출력	
E	A	$Y_1$	$Y_0$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	0

$$Y_0 = E\bar{A} \quad Y_1 = EA$$

<진리표와 논리식>



<회로도>

# 3. 디코더

## 2) 2×4 디코더

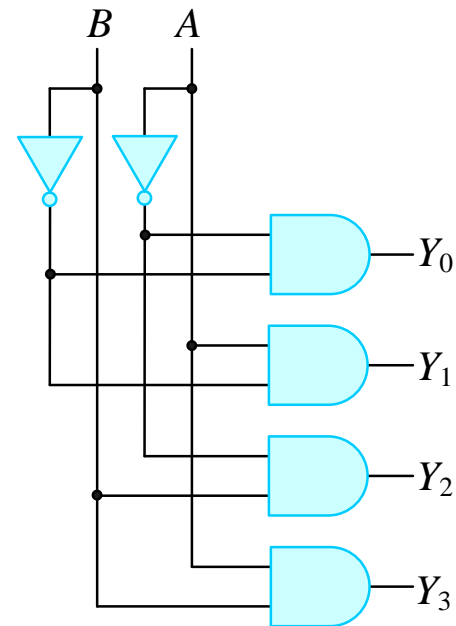
- 2개의 입력에 따라서 4개의 출력 중 하나가 선택

입력		출력			
B	A	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_0 = \overline{B}\overline{A} \quad Y_1 = \overline{B}A$$

$$Y_2 = B\overline{A} \quad Y_3 = BA$$

<진리표와 논리식>



<회로도>

### 3. 디코더

#### ■ 인에이블이 있는 2×4 디코더

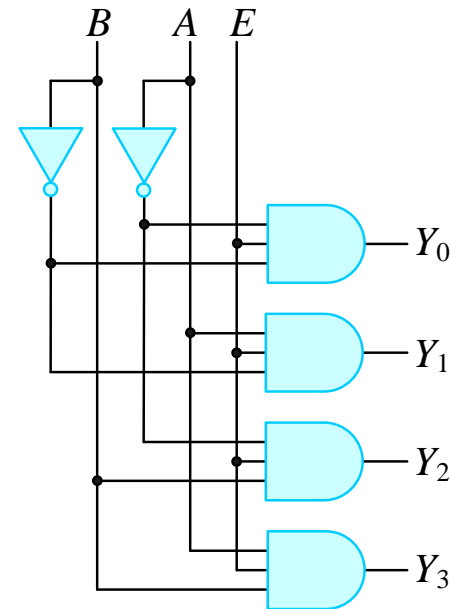
- 대부분의 IC 디코더들은 인에이블(enable) 입력이 있어서 회로를 제어한다.
- E=1일 때만 출력이 동작

입력			출력			
E	B	A	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	×	×	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

$$Y_0 = E\bar{B}\bar{A} \quad Y_1 = E\bar{B}A$$

$$Y_2 = EB\bar{A} \quad Y_3 = EBA$$

<진리표와 논리식>



<회로도>



# 3. 디코더

## 3) 3×8 디코더

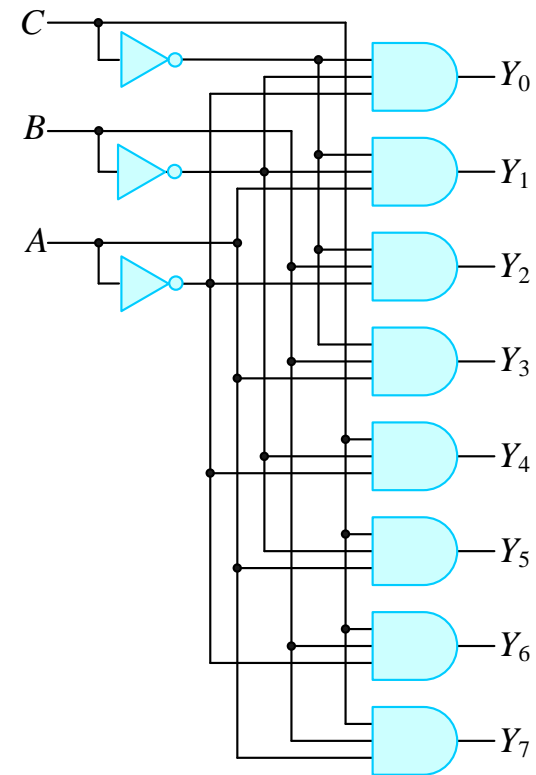
- 3개의 입력에 따라서 8개의 출력 중 하나가 선택

입력			출력							
$C$	$B$	$A$	$Y_7$	$Y_6$	$Y_5$	$Y_4$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

$$Y_0 = \overline{C}\overline{B}\overline{A}, \quad Y_1 = \overline{C}\overline{B}A, \quad Y_2 = \overline{C}B\overline{A}, \quad Y_3 = \overline{C}BA$$

$$Y_4 = C\overline{B}\overline{A}, \quad Y_5 = C\overline{B}A, \quad Y_6 = CB\overline{A}, \quad Y_7 = CBA$$

<진리표와 논리식>



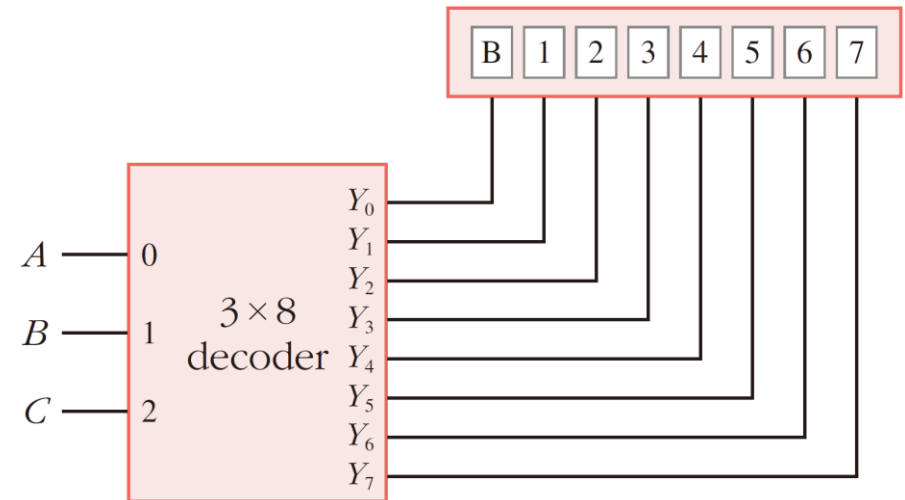
<회로도>

### 3. 디코더

#### 예제 7-3

지하1층, 지상7층으로 이루어진 8층 건물에 설치된 엘리베이터에서 현재 층을 가리키는 램프 (lamp)를 켜주는 회로를 구성하려고 한다. 엘리베이터가 위치한 층을 표시하는 디스플레이는 B, 1, 2, 3, 4, 5, 6, 7로 배열된 글자판이 있으며, 각 글자 뒤에 있는 램프가 켜지면 해당 글자판이 밝게 비추어져 층을 표시한다고 가정한다. 또한 회로의 입력으로는 각 층의 센서로부터 해당 층을 나타내는 3비트 2진수(000 : 지하층, 001 : 1층, 010 : 2층, 011 : 3층, 100 : 4층, 101 : 5층, 110 : 6층, 111 : 7층)가 입력된다고 가정한다.

**풀이** 그림과 같이 현재의 엘리베이터 위치를 나타내는 2진수를 3x8 디코더로 입력시킨다. 그리고 디코더의  $Y_0$  출력을 디스플레이의 글자 B 뒤에 위치한 램프에 연결한다. 마찬가지로  $Y_1$ 은 1,  $Y_2$ 는 2,  $Y_3$ 은 3,  $Y_4$ 는 4,  $Y_5$ 는 5,  $Y_6$ 은 6,  $Y_7$ 은 7로 연결하면 된다. 예를 들어, 엘리베이터가 5층에 있으면 3x8 디코더의 입력으로 101이 들어오고, 디코더의 출력  $Y_5$ 만이 인에이블 되어 5층이 표시된다.



End of Example



# 3. 디코더

## 4) 4×16 디코더

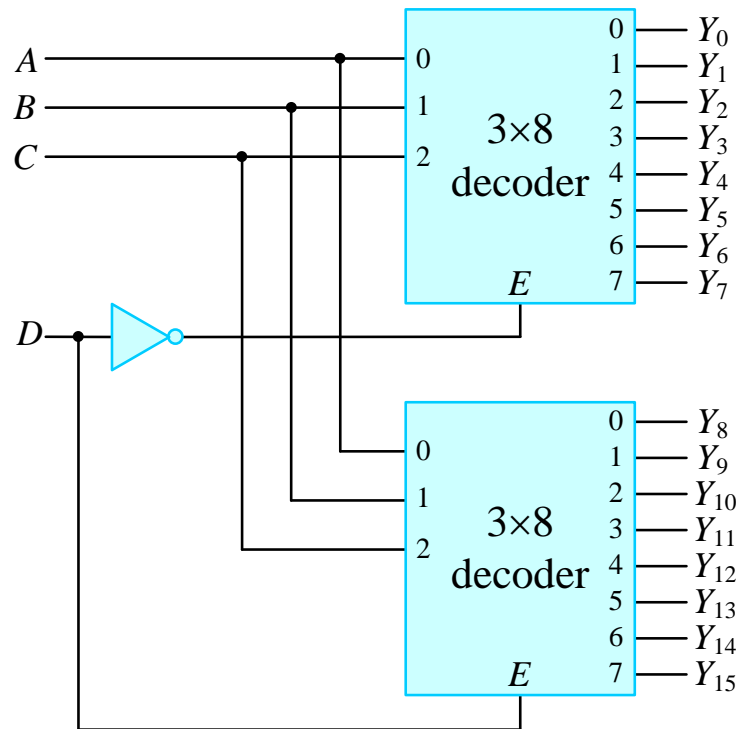
<i>D C B A</i>	<i>Y<sub>15</sub></i>	<i>Y<sub>14</sub></i>	<i>Y<sub>13</sub></i>	<i>Y<sub>12</sub></i>	<i>Y<sub>11</sub></i>	<i>Y<sub>10</sub></i>	<i>Y<sub>9</sub></i>	<i>Y<sub>8</sub></i>	<i>Y<sub>7</sub></i>	<i>Y<sub>6</sub></i>	<i>Y<sub>5</sub></i>	<i>Y<sub>4</sub></i>	<i>Y<sub>3</sub></i>	<i>Y<sub>2</sub></i>	<i>Y<sub>1</sub></i>	<i>Y<sub>0</sub></i>
0 0 0 0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0 0 0 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0 0 1 0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0 0 1 1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0 1 0 0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0 1 0 1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
0 1 1 0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
0 1 1 1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1 0 0 0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1 0 0 1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1 0 1 0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
1 0 1 1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
1 1 0 0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1 1 0 1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1 1 1 0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1 1 1 1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<4×16 디코더 진리표>

### 3. 디코더

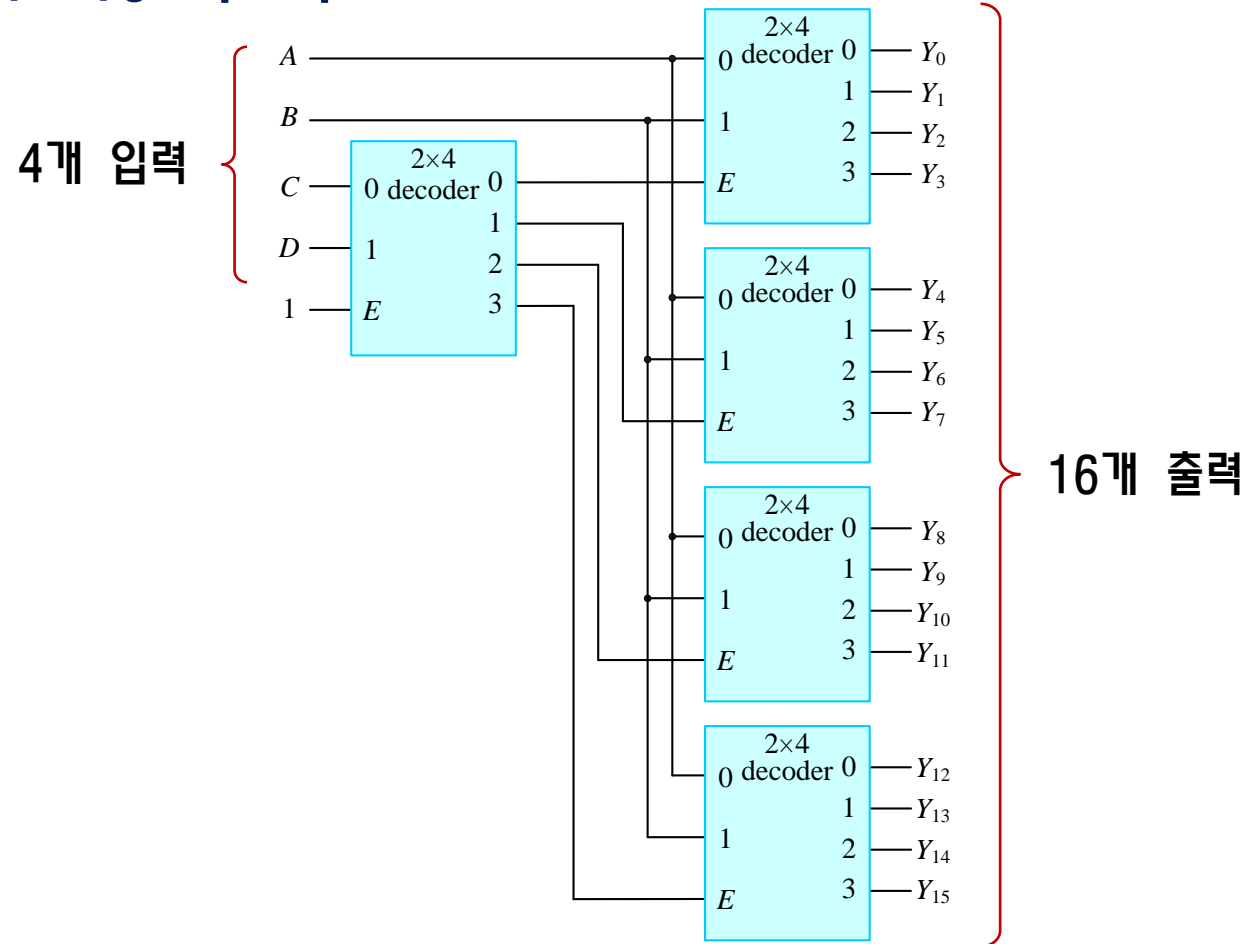
#### ■ 2개의 3×8 디코더로 4×16 디코더를 구성

$D=0$	상위 디코더만 enable되어 출력은 $Y_0 \sim Y_7$ 중의 하나가 1로 되고, 하위 디코더 출력들은 모두 0이 된다.
$D=1$	하위 디코더만 enable 되어 출력은 $Y_8 \sim Y_{15}$ 중의 하나가 1로 되고, 상위 디코더 출력들은 모두 0이 된다.



### 3. 디코더

#### <2×4 디코더 5개를 이용한 4×16 디코더>



0010010011000100  
1010010011000100

- [illegible]



### <7-세그먼트의 숫자 표시>



# 3. 디코더

<7-세그먼트 디코더 진리표>

입력				출력						
$D$	$C$	$B$	$A$	$\bar{a}$	$\bar{b}$	$\bar{c}$	$\bar{d}$	$\bar{e}$	$\bar{f}$	$\bar{g}$
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	1	1	0	0	0	0	0
0	1	1	1	0	0	0	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	0	0
1	0	1	0	×	×	×	×	×	×	×
1	0	1	1	×	×	×	×	×	×	×
1	1	0	0	×	×	×	×	×	×	×
1	1	0	1	×	×	×	×	×	×	×
1	1	1	0	×	×	×	×	×	×	×
1	1	1	1	×	×	×	×	×	×	×

## 4. 인코더

### ■ 인코더(encoder)

- 디코더의 반대기능을 수행하는 조합논리회로
- $2^n$ 개의 입력신호로부터  $n$ 개의 출력신호를 만든다.
- $2^n$ 개 중 활성화된 하나의 1비트 입력 신호를 받아서 그 숫자에 해당하는  $n$  비트 2진 정보를 출력

# 4. 인코더

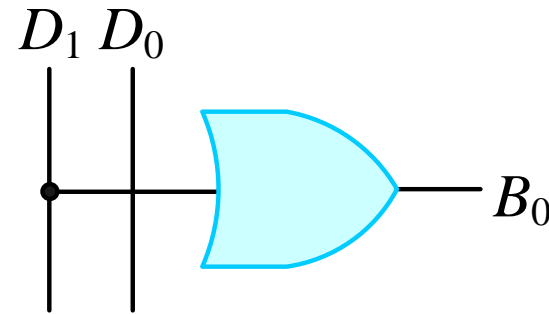
## 1) 2×1 인코더

- 입력의 신호에 따라 2개의 2진 조합(0 또는 1)을 출력

입력		출력
$D_1$	$D_0$	$B_0$
0	1	0
1	0	1

$$B_0 = D_1$$

<진리표와 논리식>



<회로도>

# 4. 인코더

## 3) 8×3 인코더

- 8(=2<sup>3</sup>)개의 입력과 3개의 출력을 가지며, 입력의 신호에 따라 3개의 2진 조합으로 출력

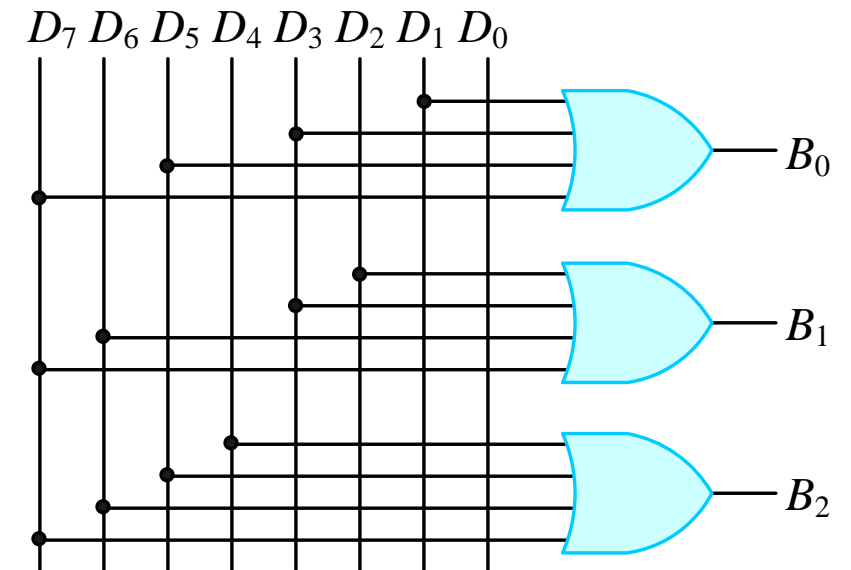
입력								출력		
$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$B_2 = D_4 + D_5 + D_6 + D_7$$

$$B_1 = D_2 + D_3 + D_6 + D_7$$

$$B_0 = D_1 + D_3 + D_5 + D_7$$

<진리표와 논리식>



<회로도>



# 5. 멀티플렉서

## ■ 멀티플렉서(Multiplexer : MUX)

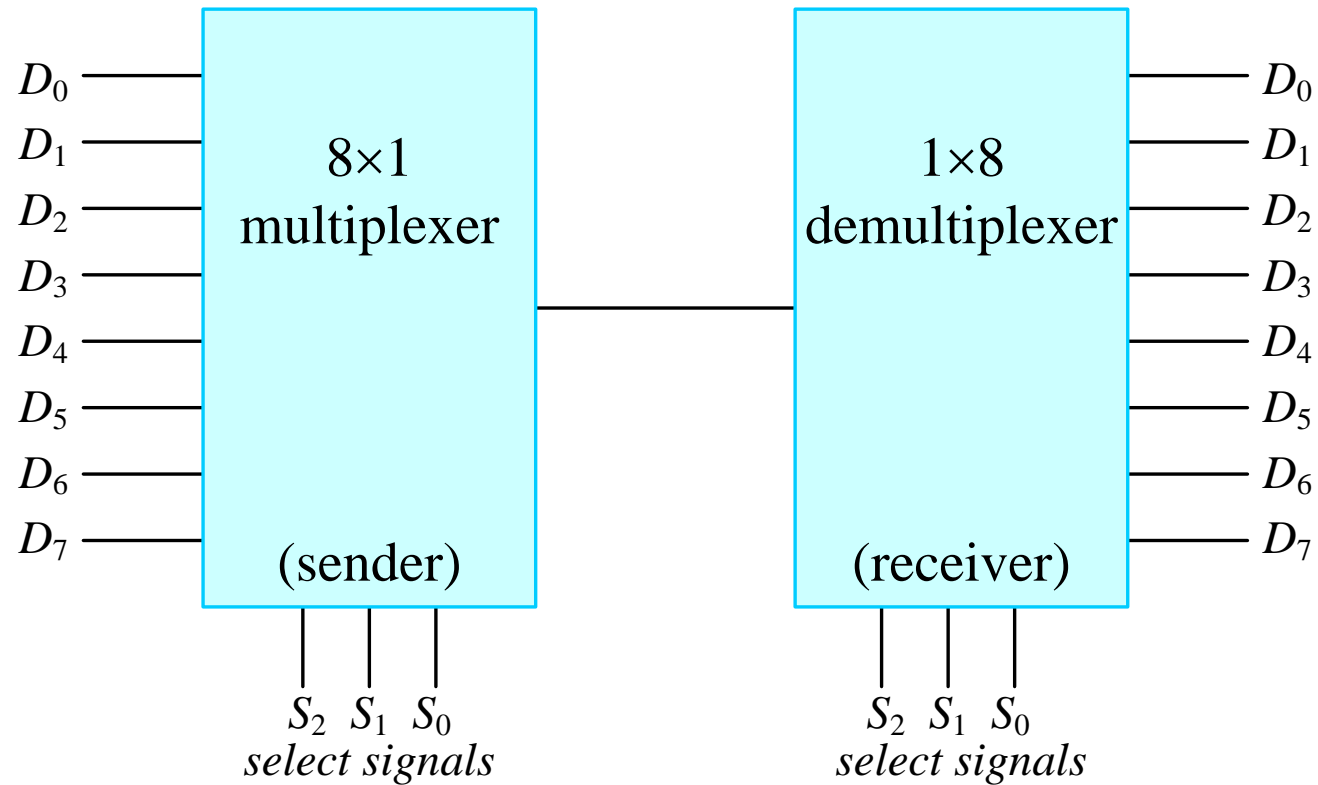
- 여러 개의 입력선들 중에서 하나를 선택하여 출력선에 연결하는 조합논리회로
- 선택선의 값에 따라서 특별한 입력선이 선택
- 데이터 선택기(data selector) : 많은 입력들 중 하나를 선택하여 선택된 입력선의 2진 정보를 출력선에 넘겨주기 때문

## ■ 디멀티플렉서(Demultiplexer : DEMUX)

- 멀티플렉서와 반대
- 정보를 한 선으로 받아서  $2^n$ 개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로
- $n$ 개의 선택선(selection line)의 값에 의해 하나의 출력선이 선택

# 5. 멀티플렉서

## ■ 멀티플렉서와 디멀티플렉서의 역할



# 5. 멀티플렉서

## 1) 2×1 멀티플렉서

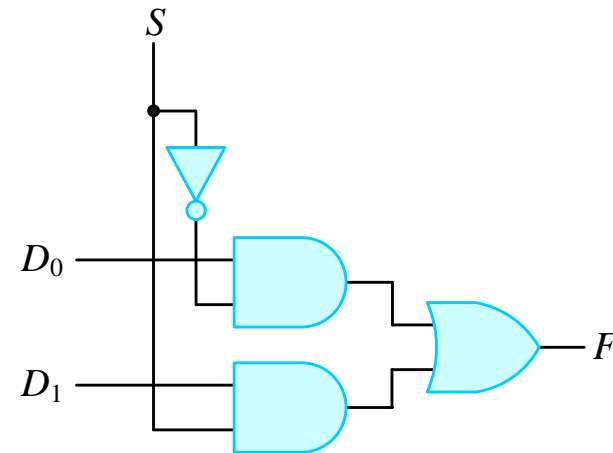
- 2( $=2^1$ )개의 입력중의 하나를 선택선 S에 입력된 값에 따라서 출력으로 보내주는 조합회로

선택선	출력
S	F
0	$D_0$
1	$D_1$

<진리표>

$$F = \bar{S}D_0 + SD_1$$

<논리식>



<회로도>

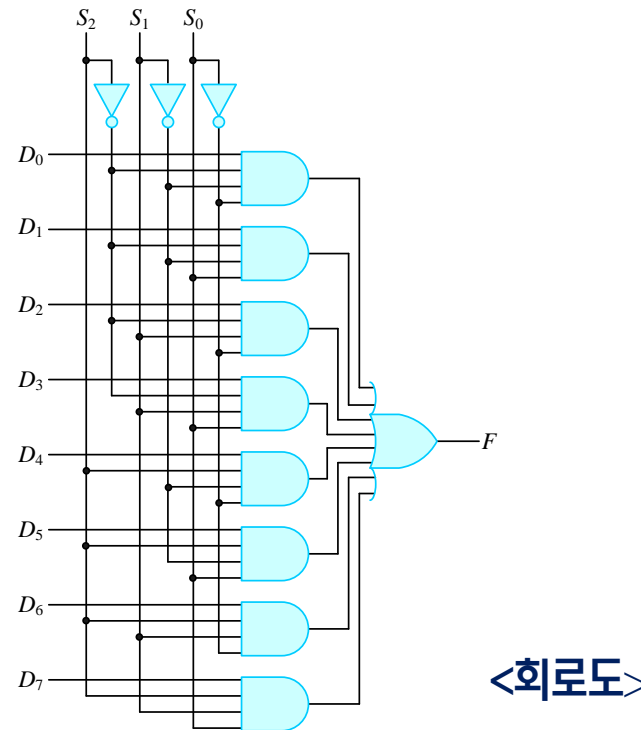
# 5. 멀티플렉서

## 3) 8×1 멀티플렉서

- 8( $=2^3$ )개의 입력중의 하나를 출력으로 보내주는 조합논리회로

선택선			출력
$S_2$	$S_1$	$S_0$	$F$
0	0	0	$D_0$
0	0	1	$D_1$
0	1	0	$D_2$
0	1	1	$D_3$
1	0	0	$D_4$
1	0	1	$D_5$
1	1	0	$D_6$
1	1	1	$D_7$

<진리표>



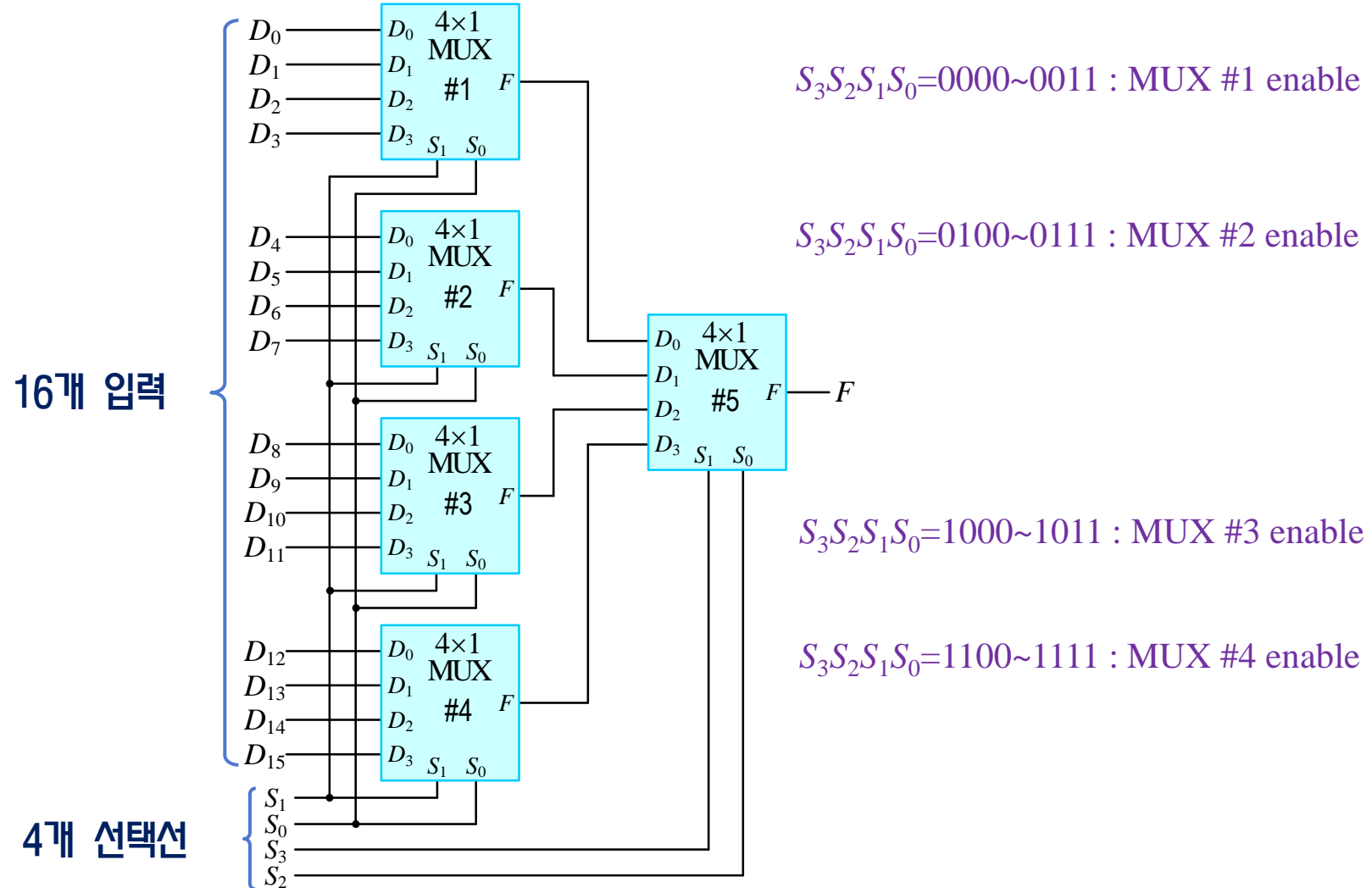
<회로도>

<논리식> 
$$F = \overline{S_2}\overline{S_1}\overline{S_0}D_0 + \overline{S_2}\overline{S_1}S_0D_1 + \overline{S_2}S_1\overline{S_0}D_2 + \overline{S_2}S_1S_0D_3 + S_2\overline{S_1}\overline{S_0}D_4 + S_2\overline{S_1}S_0D_5 + S_2S_1\overline{S_0}D_6 + S_2S_1S_0D_7$$



# 5. 멀티플렉서

- 4×1 멀티플렉서 5개를 이용한 16×1 멀티플렉서



# 5. 멀티플렉서

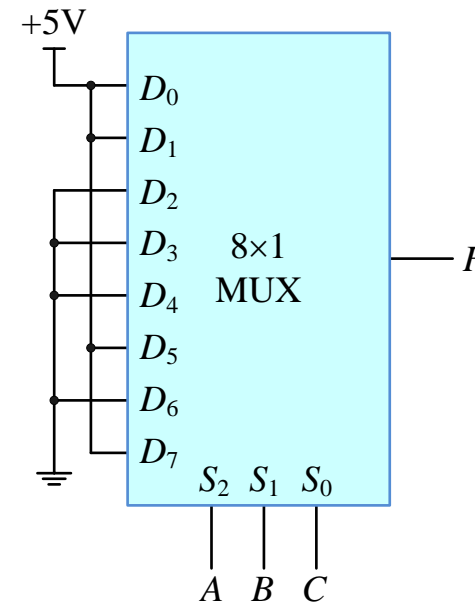
## 4) 멀티플렉서를 이용한 조합회로 구현

■  $F(A, B, C) = \sum m(0, 1, 5, 7)$ 를 8×1 멀티플렉서로 구현하는 경우

☞ 3개의 선택선을 입력  $A, B, C$ 로 사용

$A$	$B$	$C$	$F$
0	0	0	1 ( $D_0$ )
0	0	1	1 ( $D_1$ )
0	1	0	0 ( $D_2$ )
0	1	1	0 ( $D_3$ )
1	0	0	0 ( $D_4$ )
1	0	1	1 ( $D_5$ )
1	1	0	0 ( $D_6$ )
1	1	1	1 ( $D_7$ )

<진리표>



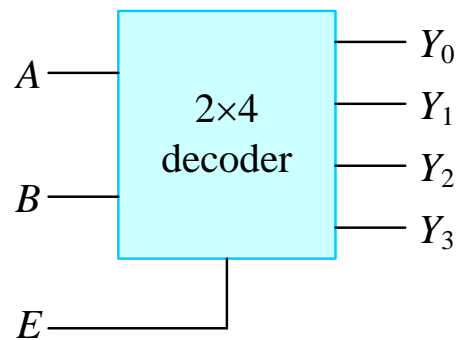
<회로도>

# 6. 디멀티플렉서

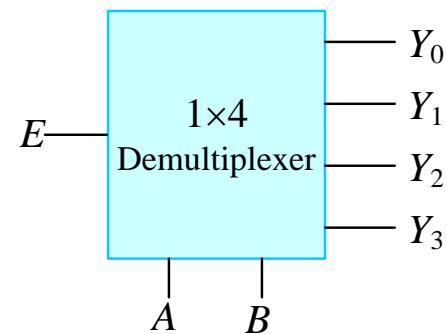
## ■ 디멀티플렉서

- 정보를 한 선으로 받아서  $2^n$ 개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로
- $n$ 개의 선택선(selection line)들을 이용하여 출력을 제어

## ■ 1개의 인에이블 입력을 가지고 있는 디코더는 디멀티플렉서로서의 기능을 수행



<2×4 디코더>



<1×4 디멀티플렉서>

# 7. 코드 변환기

## 1) 2진 코드-그레이 코드 변환

<2진 코드-그레이 코드 변환 진리표>

2진 코드(입력) $B_3 B_2 B_1 B_0$				그레이 코드(출력) $G_3 G_2 G_1 G_0$				2진 코드(입력) $B_3 B_2 B_1 B_0$				그레이 코드(출력) $G_3 G_2 G_1 G_0$			
0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	0	1
0	0	1	0	0	0	1	1	1	0	1	0	1	1	1	1
0	0	1	1	0	0	1	0	1	0	1	1	1	1	0	0
0	1	0	0	0	1	1	0	1	1	0	0	1	0	1	0
0	1	0	1	0	1	1	1	1	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1	1	1	1	0	1	0	0	1
0	1	1	1	0	1	0	0	1	1	1	1	1	0	0	0



# 7. 코드 변환기

$B_1B_0$		00	01	11	10
$B_3B_2$	00				
	01				
	11	1	1	1	1
	10	1	1	1	1

$$G_3 = B_3$$

$B_1B_0$		00	01	11	10
$B_3B_2$					
00					
01	1	1	1	1	
11					
10	1	1	1	1	

$$G_2 = \bar{B}_3B_2 + B_3\bar{B}_2 = B_3 \oplus B_2$$

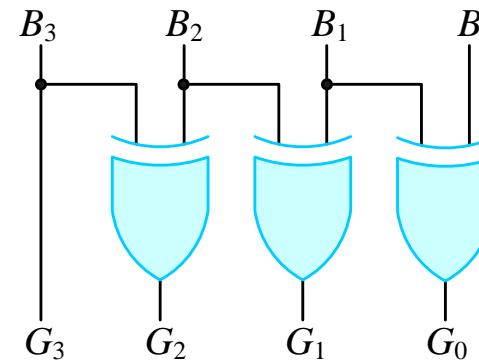
$B_1B_0$		00	01	11	10
$B_3B_2$					
00				1	1
01	1	1			
11	1	1			
10				1	1

$$G_1 = \bar{B}_2B_1 + B_2\bar{B}_1 = B_2 \oplus B_1$$

$B_1B_0$		00	01	11	10
$B_3B_2$			1		1
00			1		1
01			1		1
11			1		1
10			1		1

$$G_0 = \bar{B}_1B_0 + B_1\bar{B}_0 = B_1 \oplus B_0$$

카르노 맵



회로도

# 7. 코드 변환기

## 3) BCD 코드-3초과 코드 변환

- BCD는 10개의 숫자만 가지므로 1010 이후의 6개의 코드는 BCD에 존재하지 않는 코드
- 입력으로서 사용될 수 없어 무관항으로 처리

BCD 코드(입력)				3초과 코드(출력)			
$B_3$	$B_2$	$B_1$	$B_0$	$E_3$	$E_2$	$E_1$	$E_0$
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	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
1	1	1	1	x	x	x	x

<진리표>

## 8. 패리티 발생기/검출기

### ■ 패리티 발생

- 원래의 데이터에 1비트 패리티를 추가하여 1의 개수를 짝수 또는 홀수로 맞추는 것
- XOR는 1의 개수가 홀수일때 1이 발생됨. 원래의 데이터에 XOR의 출력을 추가하면 1의 개수가 짝수가 됨

### ■ 패리티 검출

- 패리티가 추가된 데이터에서 1의 개수가 짝수인지 홀수인지 검사하는 것

## 차례

---

### 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



# 조합논리회로와 순서논리회로

## 조합논리회로 (combinational logic circuit)

- 현재 입력의 조합에 의해서만 출력이 결정되는 논리회로

## 순서논리회로 (sequential logic circuit)

- 현재의 입력과 이전의 출력상태에 의해서 출력이 결정되는 논리회로
- 회로의 상태를 기억하는 기억 소자가 필요



# 1. 기본적인 플립플롭

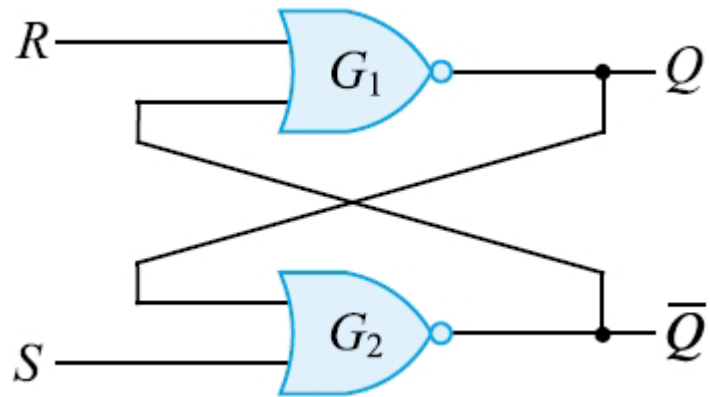
- 플립플롭(flip-flop)과 래치(latch)는 두 개의 안정된(bi-stable) 상태 중 하나를 가지는 1비트 기억소자
- 플립플롭 : 동기식 순서 논리 소자. 클록 신호에 따라 정해진 시점에서의 입력을 샘플하여 출력에 저장
- 래치 : 비동기식 순서 논리 소자. 클록 신호에 관계없이 언제든지 출력을 변화시킴
- 플립플롭과 래치도 게이트로 구성되지만 조합논리회로와 달리 궤환(feed back)이 있다.
- 래치회로는 근본적으로는 플립플롭과 유사한 기능을 수행



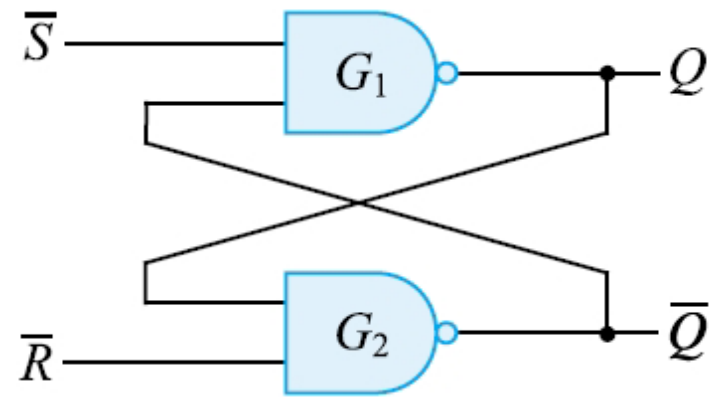
# 1. 기본적인 플립플롭

## ■ 기본적인 플립플롭 회로

- NOR 게이트 래치 또는 NAND 게이트 래치



<NOR 래치회로>



<NAND 래치회로>

# 1. 기본적인 플립플롭

## 1) NOR 게이트로 구성된 SR 래치

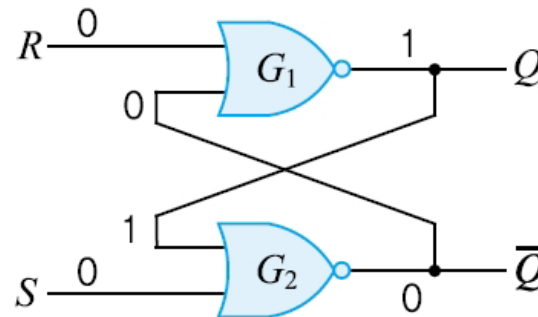
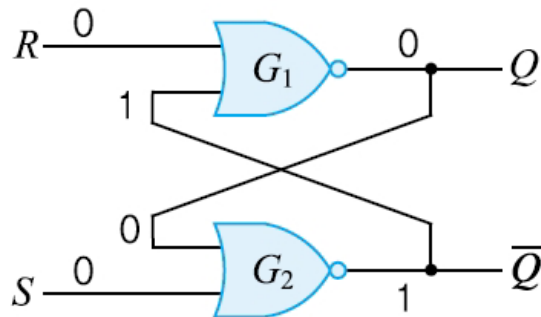
$S$	$R$	$Q(t+1)$
0	0	$Q(t)$ (불변)
0	1	0
1	0	1
1	1	(부정)

<진리표>

$A$	$B$	$F$
0	0	1
0	1	0
1	0	0
1	1	0

<NOR 진리표>

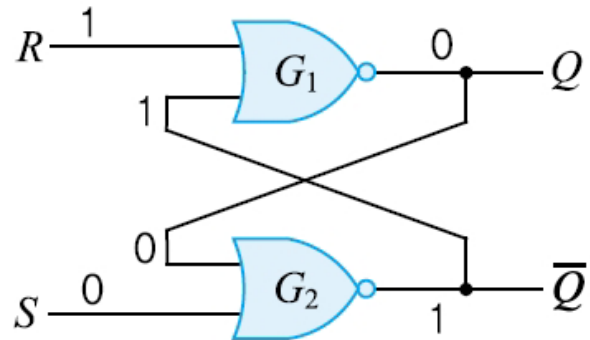
(1)  $S = 0, R = 0$  일 때



👉 출력은 현재상태 유지

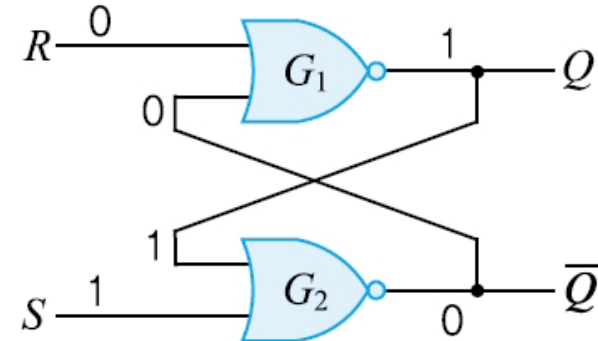
# 1. 기본적인 플립플롭

(2)  $S = 0, R = 1$  일 때



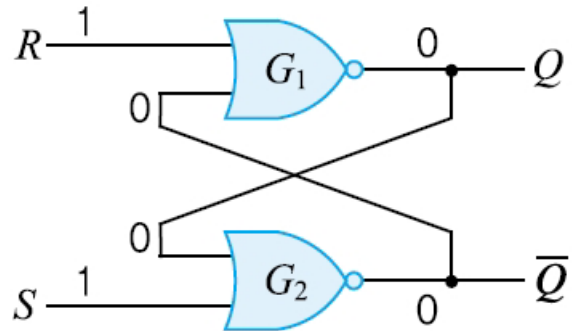
👉 출력 :  $Q = 0$

(3)  $S = 1, R = 0$  일 때



👉 출력 :  $Q = 1$

(4)  $S = 1, R = 1$  일 때



👉 출력 : 부정 ( $Q = 0, \bar{Q} = 0$ )

# 1. 기본적인 플립플롭

## 2) NAND 게이트로 구성된 SR 래치

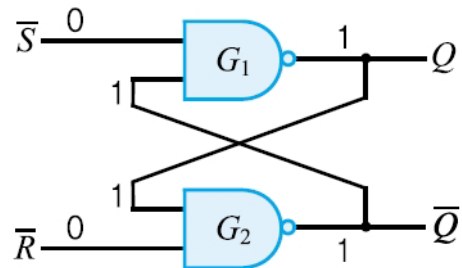
$\bar{S}$	$\bar{R}$	$Q(t+1)$
0	0	(부정)
0	1	1
1	0	0
1	1	$Q(t)$ (불변)

<진리표>

A	B	F
0	0	1
0	1	1
1	0	1
1	1	0

<NAND 진리표>

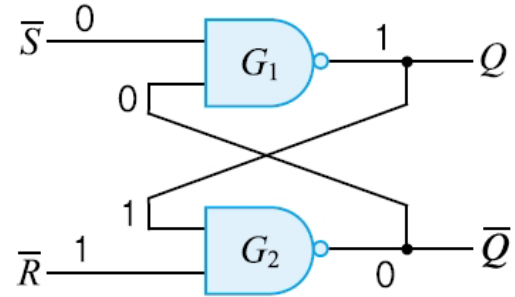
(1)  $\bar{S} = 0, \bar{R} = 0$ 일 때



👉 출력 : 부정 ( $Q=1, \bar{Q}=1$ )

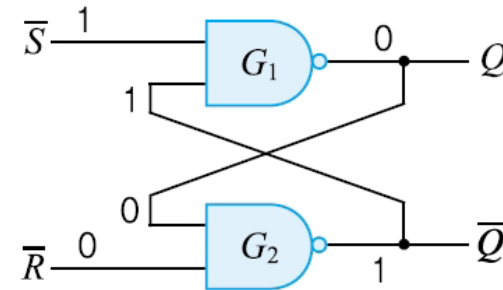
# 1. 기본적인 플립플롭

(2)  $\bar{S} = 0, \bar{R} = 1$  일 때



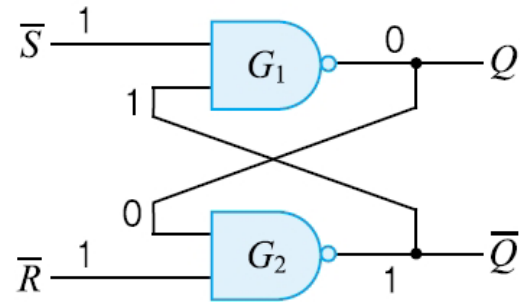
👉 출력 :  $Q = 1$

(3)  $\bar{S} = 1, \bar{R} = 0$  일 때

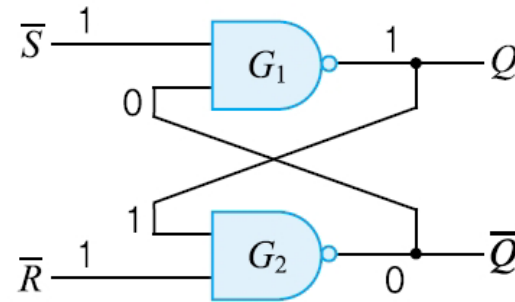


👉 출력 :  $Q = 0$

(4)  $\bar{S} = 1, \bar{R} = 1$  일 때



👉 출력은 현재상태 유지



# 1. 기본적인 플립플롭

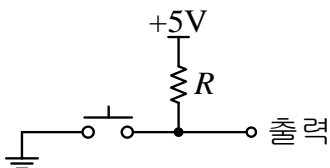
## 3) SR 래치의 응용 예

- 래치 : 기계적인 스위치에서 일어나는 접점(contact)의 바운싱(bouncing) 영향을 제거하는데 사용
- 바운싱 : 기계적인 스위치 내부에 존재하는 스프링의 탄성과 접점면의 불균일성 때문에 스위치를 개폐하는 경우 여러 번 붙었다가 떨어지는 현상

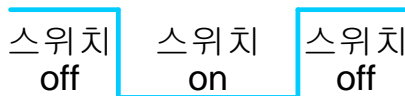


# 1. 기본적인 플립플롭

- 스위치 접점에서의 바운싱을 제거하기 위해 사용한 SR래치



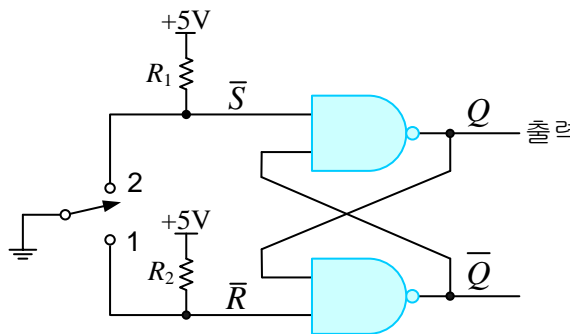
<스위치 회로>



<이상적인 출력>



<실제의 출력>



<래치를 부가한 스위치 회로>

\*풀-업 저항:  $R_1, R_2$



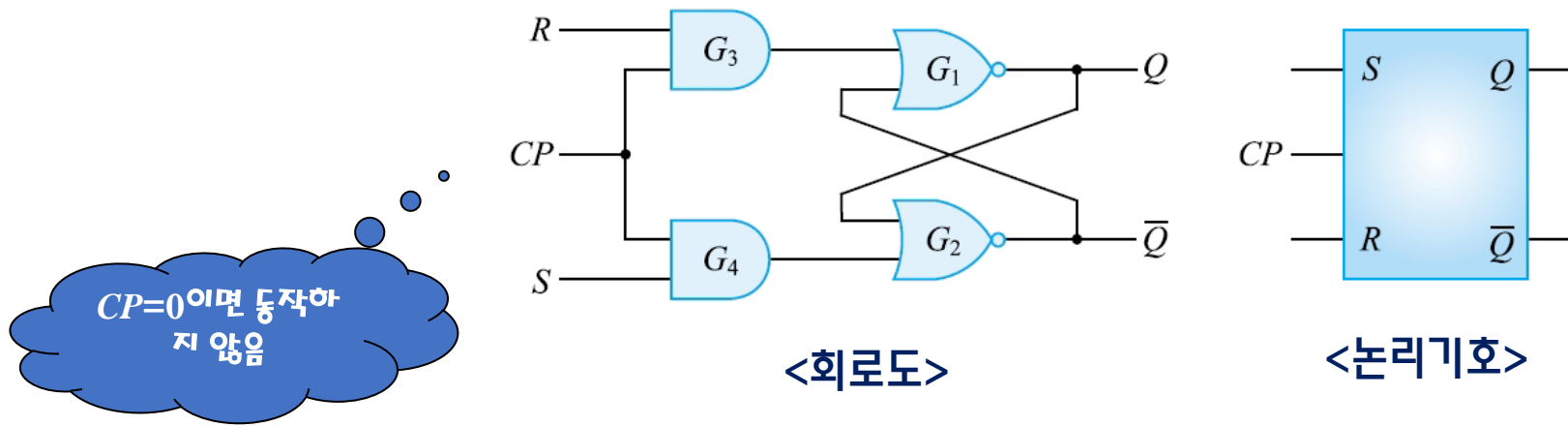
<래치가 없을 때의 출력(Q)>



<래치가 있을 때의 출력(Q)>

## 2. SR 플립플롭

### 1) 클록형 SR 플립플롭



CP=0인 경우	S와 R의 입력에 관계없이 앞단의 AND 게이트 $G_3$ 과 $G_4$ 의 출력이 항상 0이므로 플립플롭의 출력은 불변
CP=1인 경우	S와 R의 입력이 회로 후단의 NOR 게이트 $G_1$ 과 $G_2$ 의 입력으로 전달되어 앞에서 설명한 SR 래치와 같은 동작을 수행

\* 클록형 SR 플립플롭을 Gated SR 래치라고도 한다.

## 2. SR 플립플롭

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

<SR 플립플롭의 진리표>

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

<SR 플립플롭의 특성표>

특성 방정식  
(characteristic equation)

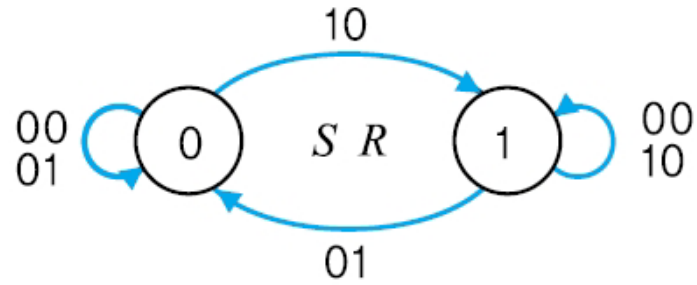
$Q(t) \backslash SR$	00	01	11	10
0			X	1
1	1		X	1

$$Q(t+1) = S + \bar{R}Q(t), \quad SR = 0$$

\*  $Q(t)$  : 현재 상태

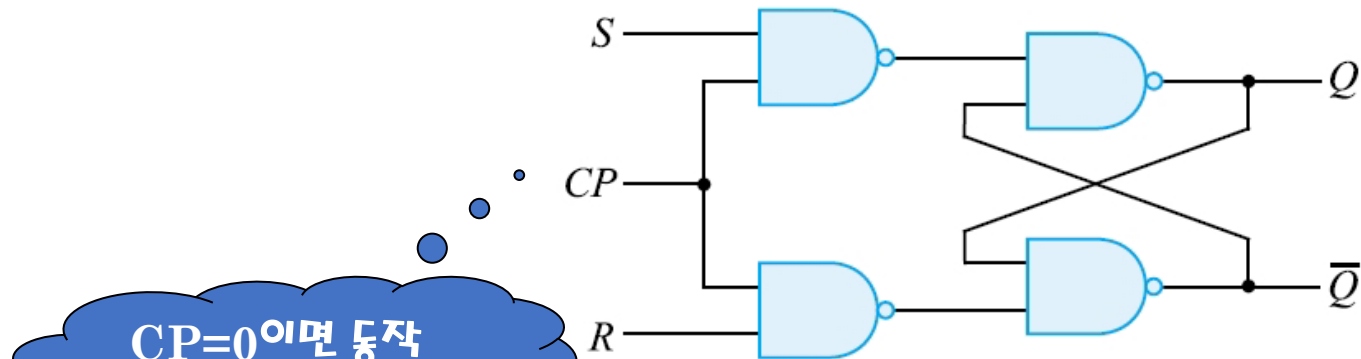
\*  $Q(t+1)$  : 다음 상태

## 2. SR 플립플롭



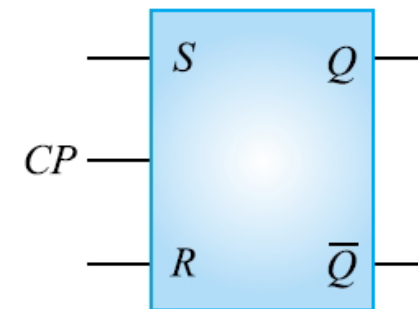
<SR 플립플롭의 상태도>

### ■ 클럭형 SR 플립플롭(NAND형)



CP=0이면 동작  
하지 않음

<회로도>

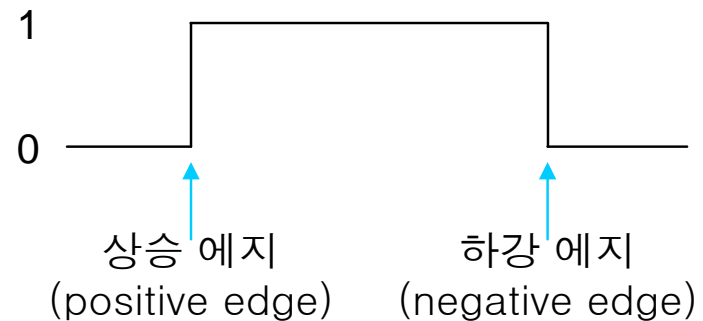


<논리기호>

## 2. SR 플립플롭

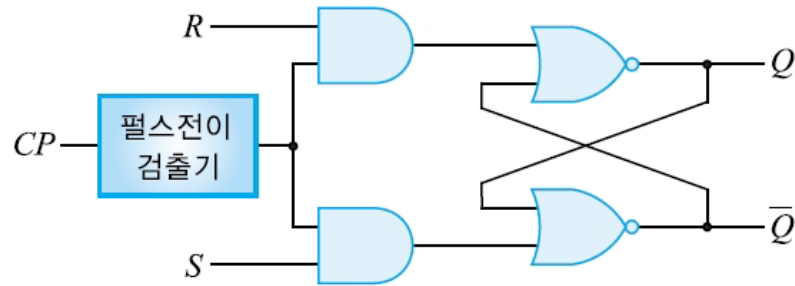
### 2) 에지 트리거 SR 플립플롭

- 클록형 SR 플립플롭은 클록펄스가 1인 상태에서 모든 동작이 수행
- 플립플롭의 동작시간보다도 클록펄스의 지속시간이 길면 플립플롭은 여러 차례 동작이 수행 되기 때문에 예측치 못한 동작을 할 여지가 충분
- 에지 트리거(edge trigger)를 이용
- 트리거 종류 : 레벨(level) 트리거, 에지(edge) 트리거
- 클록형 플립플롭은 레벨 트리거로 동작
- 에지 트리거는 플립플롭의 내부 구조를 바꾸어 클록이 0에서 1로 변하거나 1에서 0으로 변할 때의 순간에만 입력을 받아들이게 하는 방법

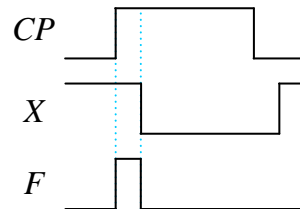
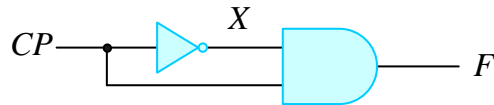


## 2. SR 플립플롭

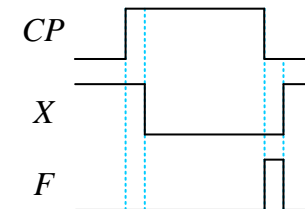
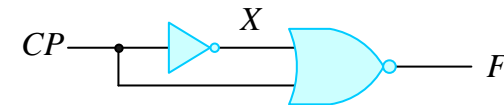
### ■ 에지 트리거 SR 플립플롭



<논리회로도>



<상승에지 펄스전이검출기>



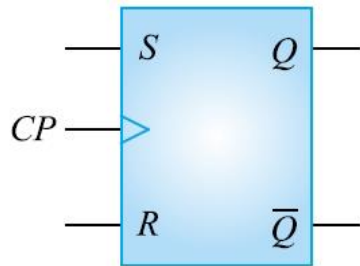
<하강에지 펄스전이검출기>

- S와 R입력을 동기입력(synchronous input)이라 한다.



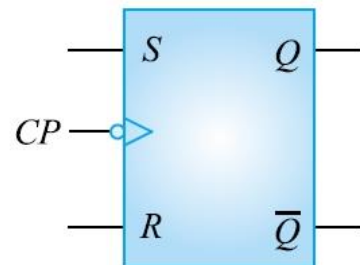
## 2. SR 플립플롭

### ■ 에지 트리거 SR 플립플롭의 논리기호와 진리표



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

<상승에지 트리거 SR 플립플롭>



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

<하강에지 트리거 SR 플립플롭>



다음 시간

## 13주차 : 플립플롭(2)

