



이론, 실습, 시뮬레이션 

# 디지털 논리회로

개정 3판

## Chapter 07. 조합논리회로

## 학습목표 및 목차

- 반가산기, 전가산기, 고속가산기 및 비교기 회로를 설계할 수 있다.
- 디코더와 인코더 동작 원리를 이해하고 응용회로를 설계할 수 있다.
- 멀티플렉서와 디멀티플렉서의 동작 원리를 이해하고 응용회로를 설계할 수 있다.
- 각종 코드를 변환하는 회로를 설계할 수 있다.
- 패리티 발생기와 검출기의 동작 원리를 이해하고 응용회로를 설계할 수 있다.

01. 가산기/감산기

02. 비교기

03. 디코더

04. 인코더

05. 멀티플렉서

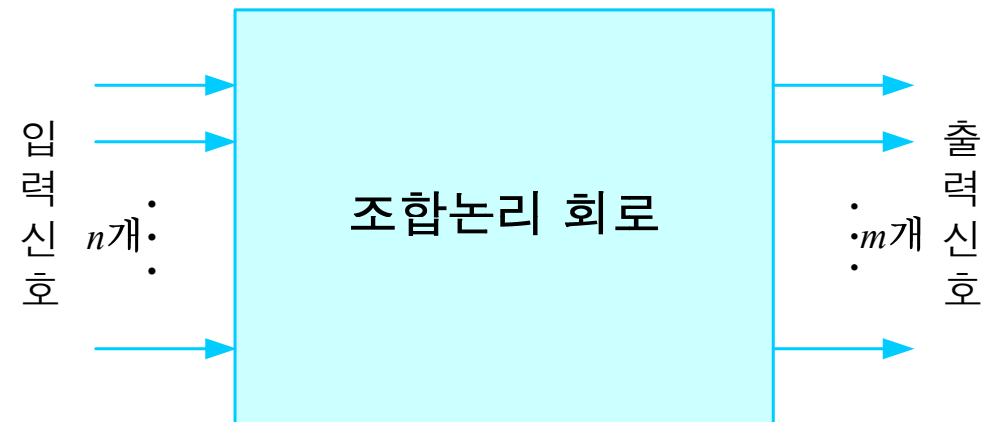
06. 디멀티플렉서

07. 코드 변환기

08. 패리티 발생기/검출기

# 개요

- 개요 조합논리회로는 논리곱(AND), 논리합(OR), 논리 부정(NOT)의 세 가지 기본 논리 회로를 조합하여 구성한 논리 회로
- 조합논리회로는 입력변수, 논리 게이트, 그리고 출력변수들로 구성



<조합논리회로 블록도>

# 01 가산기

## 1. 반가산기(half-adder, HA)

$$\begin{array}{r}
 A \\
 + B \\
 \hline
 C \ S
 \end{array}
 \quad
 \begin{array}{r}
 0 \\
 + 0 \\
 \hline
 0 \ 0
 \end{array}
 \quad
 \begin{array}{r}
 0 \\
 + 1 \\
 \hline
 0 \ 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 + 0 \\
 \hline
 0 \ 1
 \end{array}
 \quad
 \begin{array}{r}
 1 \\
 + 1 \\
 \hline
 1 \ 0
 \end{array}$$

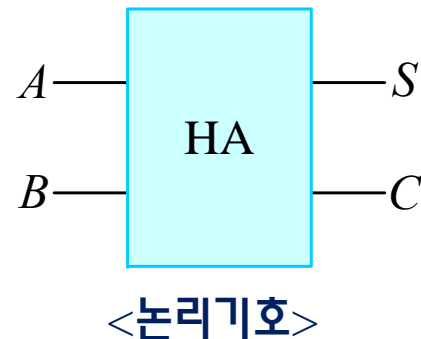
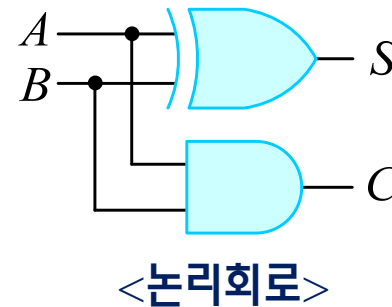
$S$  : sum  
 $C$  : carry

입력		출력	
$A$	$B$	$S$	$C$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = A \cdot B$$

<진리표와 논리식>



# 01 가산기

## 2. 전가산기(full-adder, FA)

- 자리 올림수(carry)를 고려하여 만든 덧셈 회로

$C_{in}$	0	1	0	1	0	1	0	1
$A$	0	0	1	1	0	0	1	1
$+ B$	$+ 0$	$+ 0$	$+ 0$	$+ 0$	$+ 1$	$+ 1$	$+ 1$	$+ 1$
$C_{out} S$	0 0	0 1	0 1	1 0	0 1	1 0	1 0	1 1

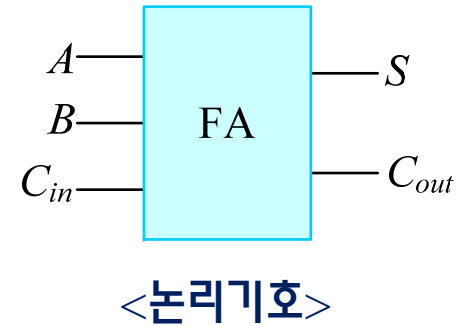
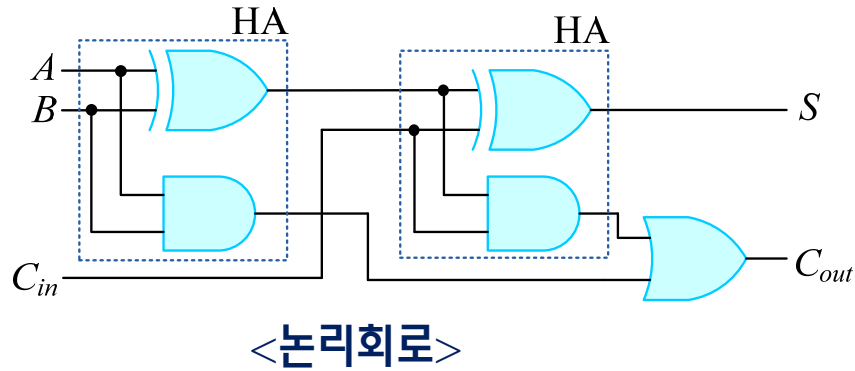
입력			출력	
$A$	$B$	$C_{in}$	$S$	$C_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$$\begin{aligned}
 S &= \overline{A}\overline{B}C_{in} + \overline{A}B\overline{C_{in}} + A\overline{B}\overline{C_{in}} + ABC_{in} \\
 &= \overline{A}(\overline{B}C_{in} + B\overline{C_{in}}) + A(\overline{B}\overline{C_{in}} + BC_{in}) \\
 &= \overline{A}(B \oplus C_{in}) + A(\overline{B \oplus C_{in}}) \\
 &= A \oplus (B \oplus C_{in}) = (A \oplus B) \oplus C_{in}
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C_{in}} + ABC_{in} \\
 &= C_{in}(\overline{A}B + A\overline{B}) + AB(\overline{C_{in}} + C_{in}) \\
 &= C_{in}(A \oplus B) + AB
 \end{aligned}$$

<진리표와 논리식>

# 01 가산기



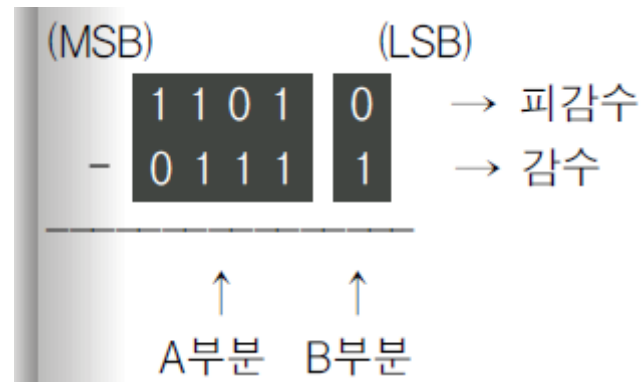
$$S = A \oplus B \oplus C_{in}$$

$$C_{out} = C_{in}(A \oplus B) + AB$$

- 전가산기는 **반가산기 2개**와 **OR 게이트**를 이용하여 구성

# 01-1 감산기

n개의 비트로 구성된 두 개의 2진수 감산



## ■ B부분

- 최하위 비트
- borrow를 빌려올 수는 있어도 하위 비트가 존재하지 않으므로 borrow를 빌려줄 수는 없다.

## ■ A부분

- borrow를 상위 비트에서 빌려올 수도 있으며, 하위 비트에 빌려줄 수도 있다.

# 01-1 감산기

## 1. 반감산기(half-subtractor, HS)

- 임의의 입력 비트 두 개 : A와 B
- 출력
  - 두 비트의 차인 D(difference)와 상위 비트로부터의 borrow를 빌려왔는지의 여부에 따라 Bo(output borrow)로 표시
- 반감산기 진리표

입 력		출 력	
A	B	$B_0$	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$\begin{array}{r} B_0 \swarrow \\ A \\ -B \\ \hline D \end{array}$$

- 출력 D와 Bo에 대한 부울식

$$D = \overline{A}B + A\overline{B} = A \oplus B$$

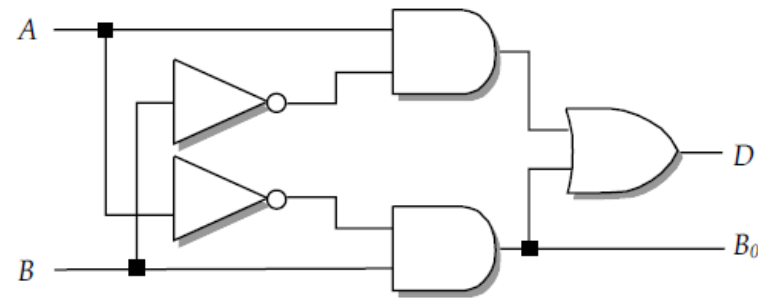
$$B_0 = \overline{A}B$$



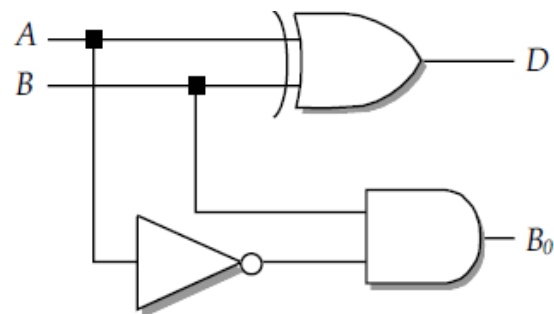
# 01-1 감산기

## 1. 반감산기(half-subtractor, HS)

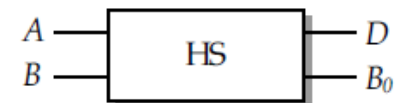
- 회로도와 논리기호



(a) 반감산기 회로



(b) 등가회로



(c) 논리기호

# 01-1 감산기

## 2. 전감산기(full-subtractor, FS)

- 입력
  - 피감수 비트와 감수 비트인 두 개의 비트  $A$ ,  $B$ 와 하위 비트에 1을 빌려주었는지를 고려한 입력 빌림수  $B_i$ (input borrow)를 포함
- 출력
  - 차를 나타내는  $D$ (difference)와 상위 비트에서 borrow가 있었는지를 고려한 출력 빌림수  $B_o$ (output borrow)로 구성
- 진리표

입 력			출 력	
$A$	$B$	$B_i$	$B_o$	$D$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

# 01-1 감산기

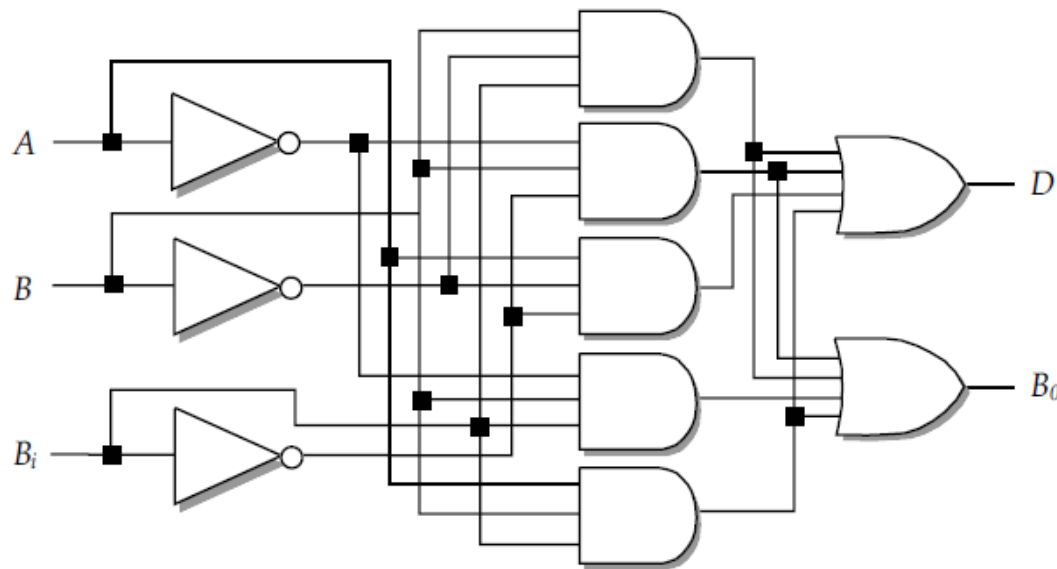
## 2. 전감산기(full-subtractor, FS)

- 출력 D와  $B_0$ 에 대해 각각 부울식으로 표현

$$D = \overline{A}\overline{B}B_i + \overline{A}B\overline{B}_i + A\overline{B}\overline{B}_i + AB\overline{B}_i$$

$$B_0 = \overline{A}\overline{B}B_i + \overline{A}B\overline{B}_i + \overline{A}BB_i + A\overline{B}B_i$$

- 전체 논리 회로도 구성



# 01-1 감산기

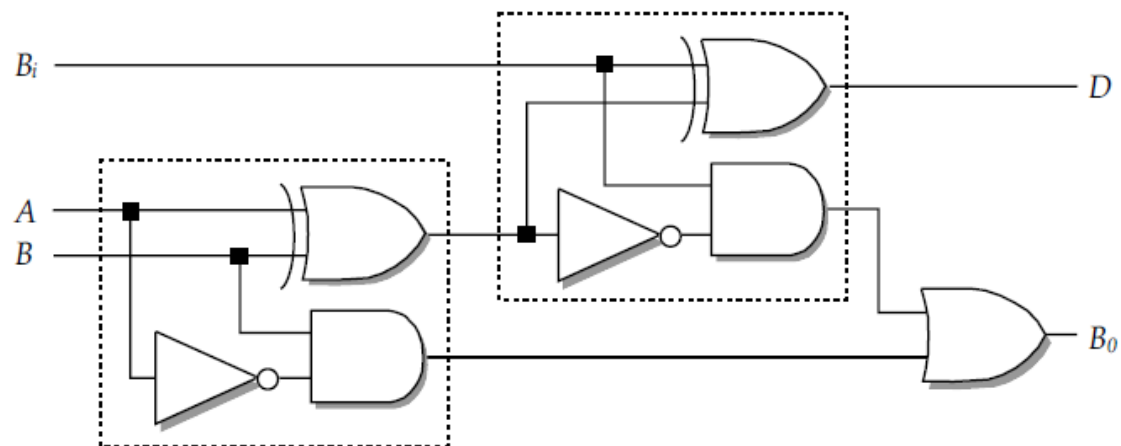
## 2. 전감산기(full-subtractor, FS)

- 부울식 D와 B0를 간소화

$$\begin{aligned} D &= \overline{A}\overline{B}B_i + \overline{A}B\overline{B}_i + A\overline{B}\overline{B}_i + AB B_i \\ &= \overline{A}(\overline{B}B_i + B\overline{B}_i) + A(\overline{B}\overline{B}_i + BB_i) \\ &= \overline{A}(B \oplus B_i) + A(\overline{B} \oplus \overline{B}_i) \\ &= A \oplus (B \oplus B_i) \end{aligned}$$

$$\begin{aligned} B_0 &= \overline{A}\overline{B}B_i + \overline{A}B\overline{B}_i + \overline{A}B B_i + AB B_i \\ &= B_i(\overline{A}\overline{B} + AB) + \overline{A}B(\overline{B}_i + B_i) \\ &= B_i(\overline{A} \oplus B) + \overline{A}B \end{aligned}$$

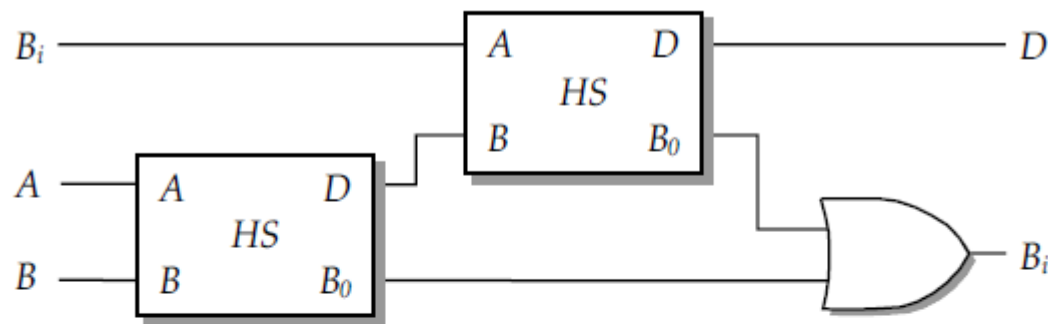
- 간소화된 부울식에 대한 전감산기 회로



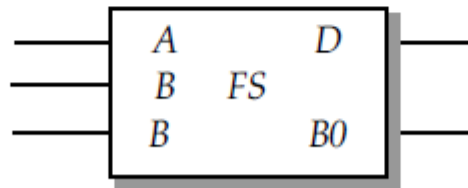
# 01-1 감산기

## 2. 전감산기(full-subtractor, FS)

- 두 개의 반감산기를 이용한 전감산기 회로와 논리기호



(a) 반감산기를 이용한 전감산기 회로

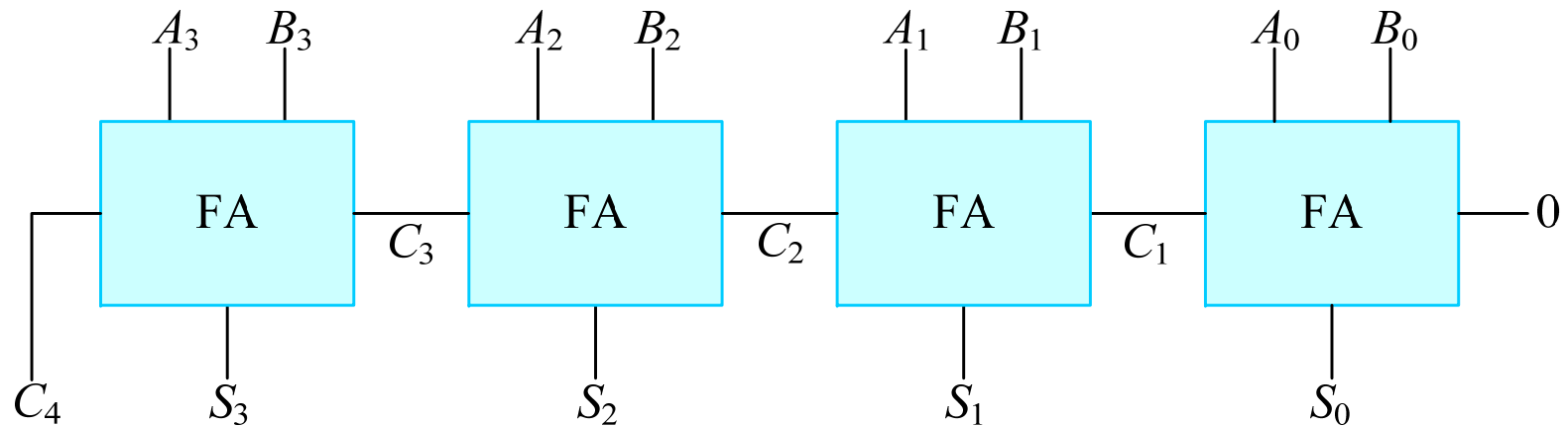


(b) 논리기호

# 01 가감산기

## 3. 병렬가감산기(parallel-adder/subtractor)

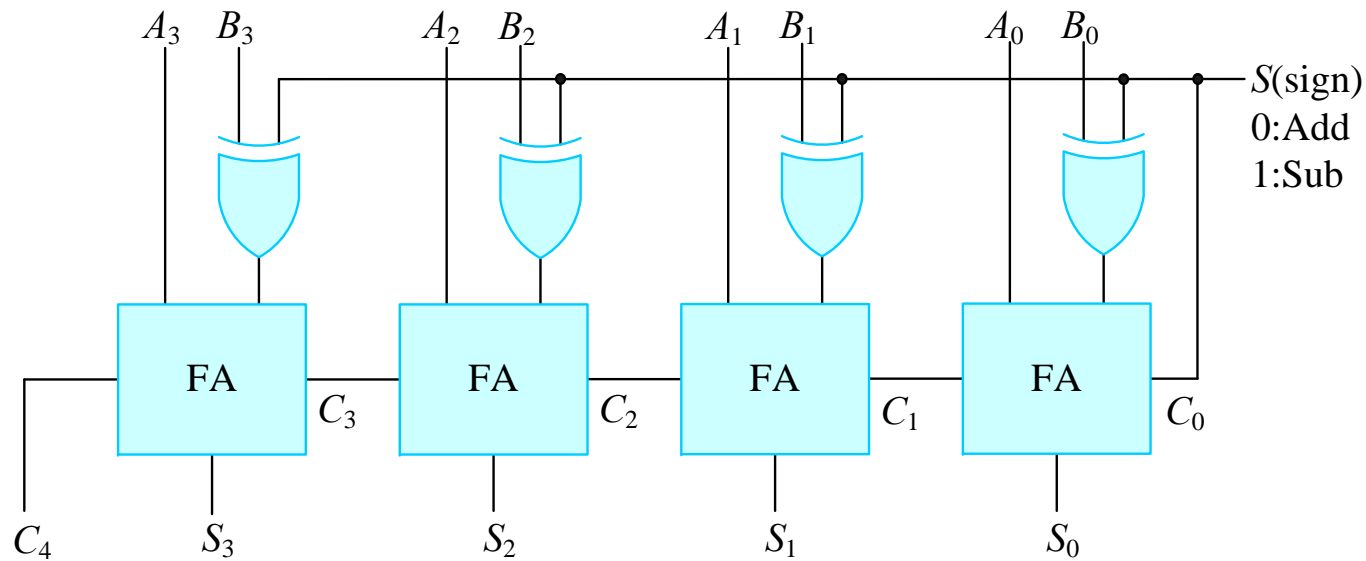
- 병렬가산기 : 전가산기 여러 개를 병렬로 연결한 회로



<전가산기를 이용한 병렬가산기>

# 01 가산기

- 병렬가감산기 : 병렬가산기의  $B$ 입력을 부호  $S(\text{sign})$ 와 XOR하여 전가산기의 입력으로 사용함으로써 덧셈과 뺄셈이 모두 가능한 회로

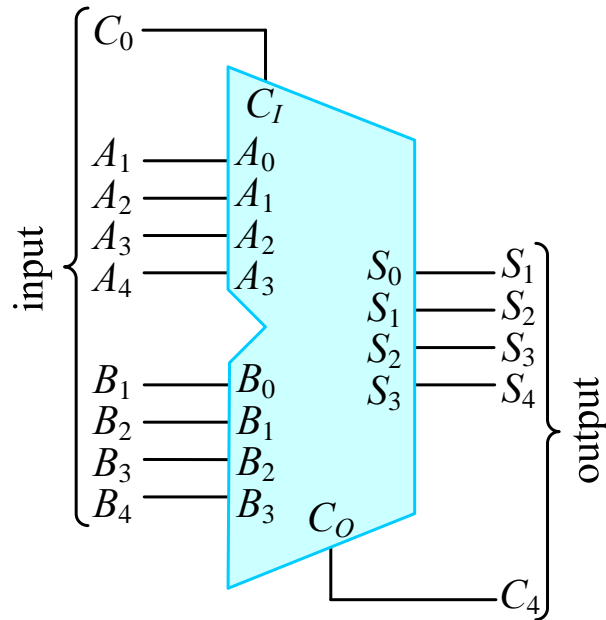


<병렬가감산기>

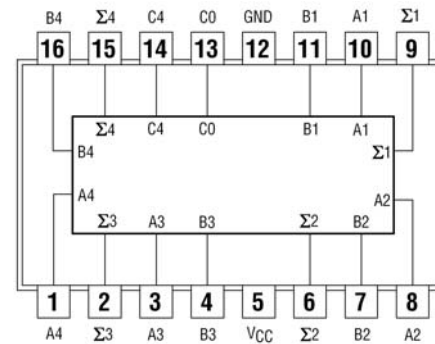
# 01 가산기

## ■ IC 7483/74283

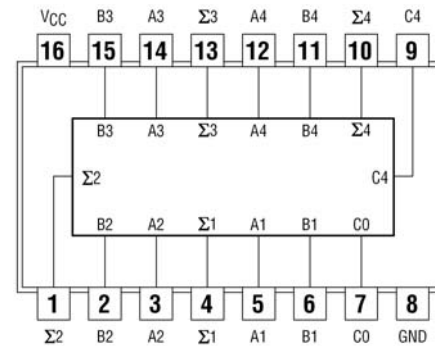
- 4비트 2진 전가산기이며, 내부에 carry look ahead 회로 내장



<개념도>



<IC 7483 핀 배치도>



<IC 74283 핀 배치도>



# 01 가산기

## 5. BCD 가산기

- BCD 코드는 2진수와 달리 표현범위가 0에서 9까지이다.
- 그러므로 BCD 계산을 하려면 결과를 보정해 주어야 한다.
- 2진수 합의 결과가 1010~1111인 경우 보정
- 6+7=13인 경우

$$\begin{array}{r} 0110 \\ + 0111 \\ \hline 1101 \end{array}$$

→  
보정 +6

$$\begin{array}{r} 1101 \\ + 0110 \\ \hline 10011 \end{array}$$

# 01 가산기

<BCD 덧셈표>

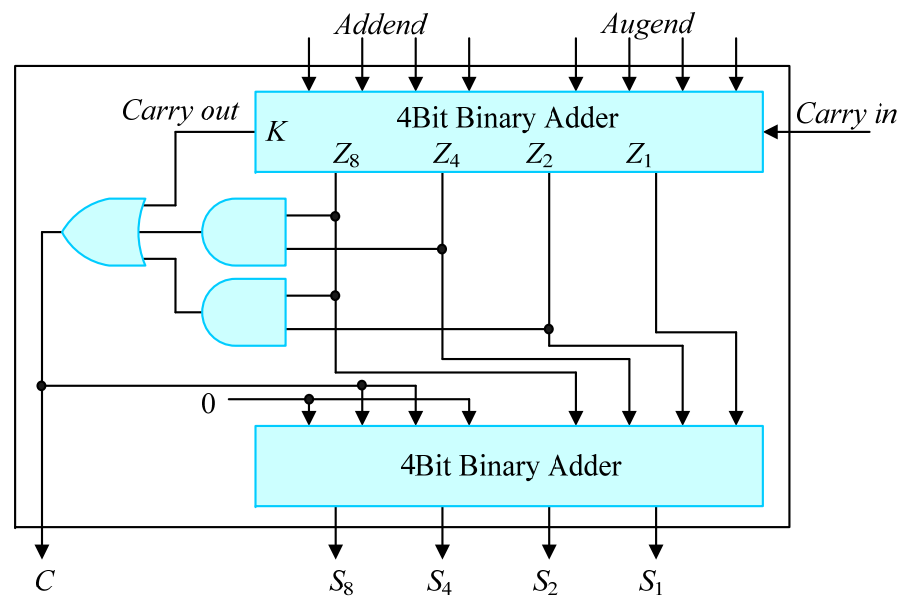
2진 합					BCD 합					10진 값
$K$	$Z_8$	$Z_4$	$Z_2$	$Z_1$	$C$	$S_8$	$S_4$	$S_2$	$S_1$	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	10
0	1	0	1	1	1	0	0	0	1	11
0	1	1	0	0	1	0	0	1	0	12
0	1	1	0	1	1	0	0	1	1	13
0	1	1	1	0	1	0	1	0	0	14
0	1	1	1	1	1	0	1	0	1	15
1	0	0	0	0	1	0	1	1	0	16
1	0	0	0	1	1	0	1	1	1	17
1	0	0	1	0	1	1	0	0	0	18
1	0	0	1	1	1	1	0	0	1	19

# 01 가산기

$Z_8Z_4 \backslash Z_2Z_1$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

$$C = K + Z_8Z_4 + Z_8Z_2$$

<BCD 합에서 캐리를 만들어 주어야 하는 경우의 논리식>



<BCD 가산기>

## 02 비교기

- 2진 비교기(comparator) : 두 개의 2진수의 크기를 비교하는 회로

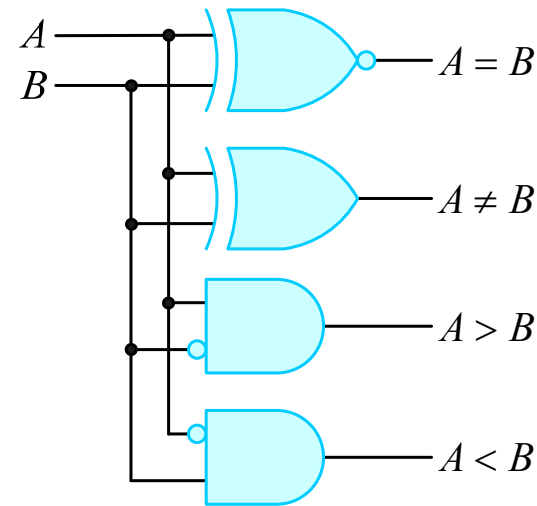
### ■ 1비트 비교기

입력		출력			
$A$	$B$	$A=B$ $F_1$	$A \neq B$ $F_2$	$A > B$ $F_3$	$A < B$ $F_4$
0	0	1	0	0	0
0	1	0	1	0	1
1	0	0	1	1	0
1	1	1	0	0	0

$$F_1 = \overline{A \oplus B}, \quad F_2 = A \oplus B,$$

$$F_3 = A \overline{B}, \quad F_4 = \overline{A} B$$

<진리표와 논리식>



<회로도>

## 02 비교기

### ■ 2비트 비교기

입력		출력			
$A$	$B$	$A=B$	$A \neq B$	$A > B$	$A < B$
$A_1A_2$	$B_1B_2$	$F_1$	$F_2$	$F_3$	$F_4$
0 0	0 0	1	0	0	0
	0 1	0	1	0	1
	1 0	0	1	0	1
	1 1	0	1	0	1
0 1	0 0	0	1	1	0
	0 1	1	0	0	0
	1 0	0	1	0	1
	1 1	0	1	0	1
1 0	0 0	0	1	1	0
	0 1	0	1	1	0
	1 0	1	0	0	0
	1 1	0	1	0	1
1 1	0 0	0	1	1	0
	0 1	0	1	1	0
	1 0	0	1	1	0
	1 1	1	0	0	0

<진리표>

## 02 비교기

$B_2B_1 \backslash A_2A_1$	00	01	11	10
00	1			
01		1		
11			1	
10				1

$$F_1 = (A_1 \oplus B_1)(A_2 \oplus B_2)$$

$B_2B_1 \backslash A_2A_1$	00	01	11	10
00		1	1	1
01	1		1	1
11	1	1		1
10	1	1	1	

$$F_2 = (A_1 \oplus B_1) + (A_2 \oplus B_2)$$

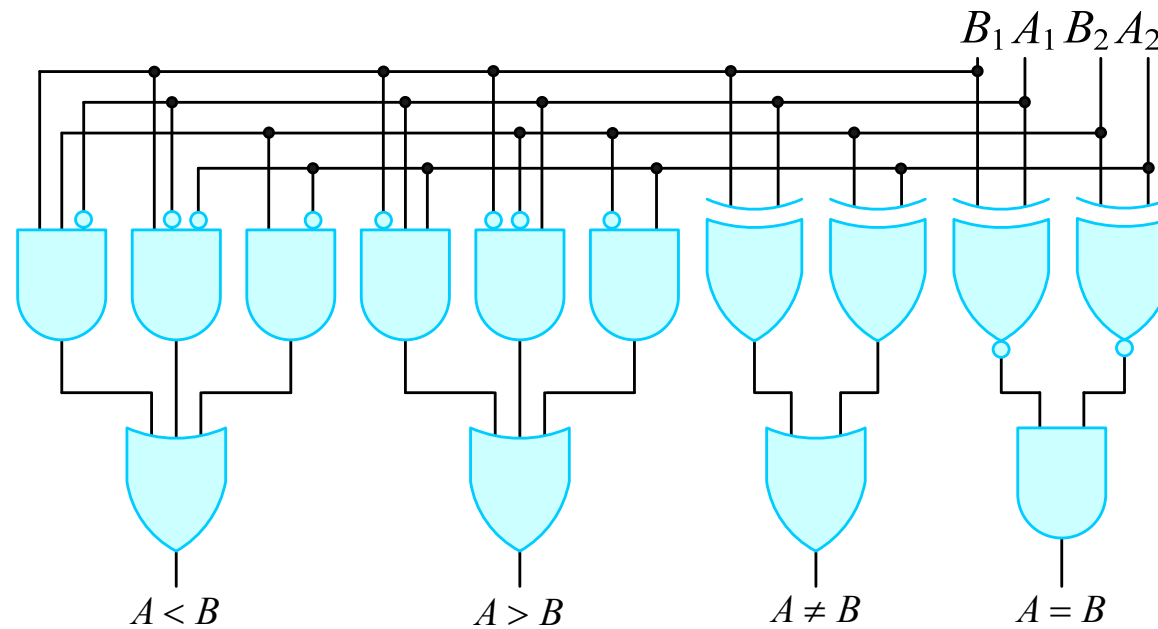
$B_2B_1 \backslash A_2A_1$	00	01	11	10
00				
01	1			
11	1	1		1
10	1	1		

$$F_3 = A_1 \overline{B_1} + A_2 \overline{B_1} B_2 + A_1 A_2 \overline{B_2}$$

$B_2B_1 \backslash A_2A_1$	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

$$F_4 = \overline{A_1} B_1 + \overline{A_1} \overline{A_2} B_2 + \overline{A_2} B_1 B_2$$

## 02 비교기

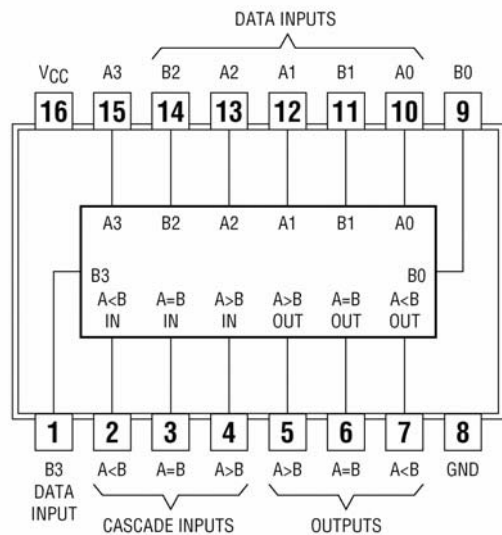


<2비트 비교기 회로>

## 02 비교기

### ■ IC 7485(4비트 비교기)

- 7485는  $A_3 \sim A_0$ 와  $B_3 \sim B_0$ 의 크기를 비교하는 회로
- $A > B$ 일 때  $AGBO$ 의 출력이 1,  $A < B$ 일 때  $ALBO$ 의 출력이 1,  $A = B$ 일 때  $AEBO$ 의 출력이 1이 된다.
- 확장 입력  $AGBI$ ,  $ALBI$ ,  $AEBI$ 는 LSB로 입력되며, 즉, 아랫단의  $AGBO$ ,  $ALBO$ ,  $AEBO$ 의 출력이 윗단의  $AGBI$ ,  $ALBI$ ,  $AEBI$ 의 입력이 된다. 맨 아랫단의  $AGBI$ ,  $ALBI$ 는 0을  $AEBI$ 는 1을 입력한다.



<IC 7485 핀 배치도>

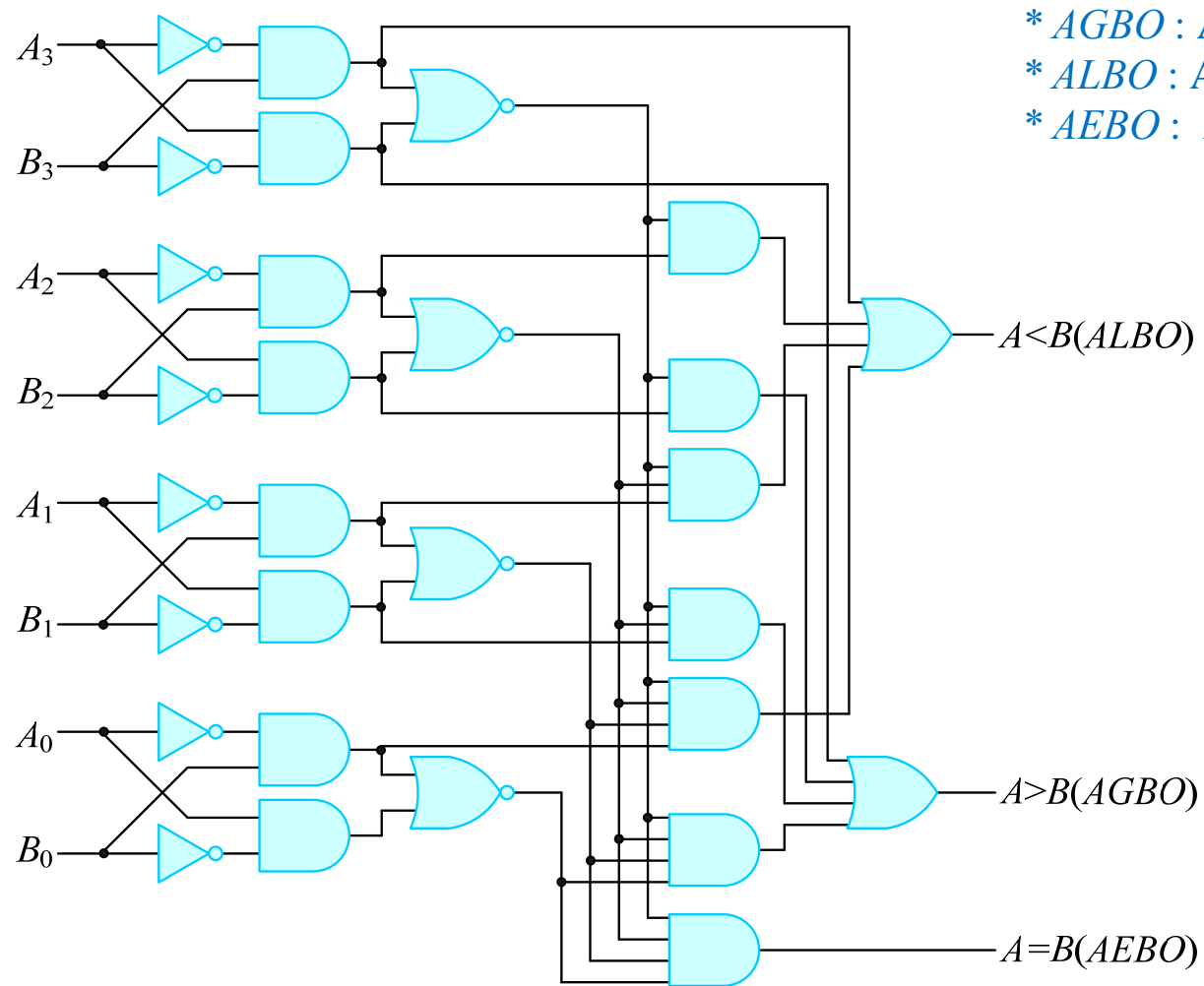


## 02 비교기

입 력							출 력		
$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$AGBI$	$ALBI$	$AEBI$	$AGBO$ $A > B$	$ALBO$ $A < B$	$AEBO$ $A = B$
$A_3 > B_3$	×	×	×	×	×	×	1	0	0
$A_3 < B_3$	×	×	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 > B_2$	×	×	×	×	×	1	0	0
$A_3 = B_3$	$A_2 < B_2$	×	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	×	×	×	×	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	×	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	×	×	×	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	×	×	×	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	0	1	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	0	0	1	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	1	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	0	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	1	0	0	1
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	1	1	0	0	0	0
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	0	0	0	1	1	0

<4비트 비교기 IC 7485 진리표>

## 02 비교기



\* *AGBO* : A Greater than B Output

\* *ALBO* : A Less than B Output

\* *AEBO* : A Equal B Output

<IC 7485 크기 비교기 회로>

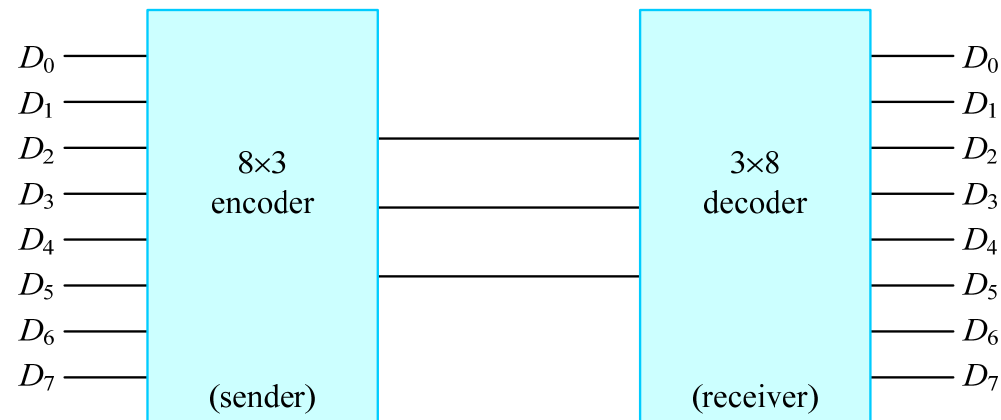


## 03 디코더

### ■ 디코더(decoder)

- 디코더 : 입력선에 나타나는  $n$ 비트의 2진 코드를 최대  $2^n$ 개의 서로 다른 정보로 바꿔주는 조합논리회로
- 인에이블(enable)단자를 가지고 있는 경우는 디멀티플렉서의 기능도 수행
- 실제 상용 IC의 경우에는 디코더와 디멀티플렉서의 기능으로 모두 사용

74138	3×8 디코더/디멀티플렉서
74139	독립된 2개의 2×4 디코더/디멀티플렉서
74154	4×16 디코더/디멀티플렉서



<디코더와 인코더의 기능>

## 03 디코더

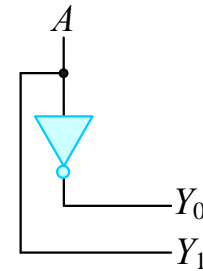
### 1. 1×2 디코더

- 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>



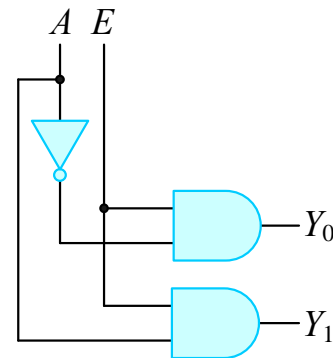
<회로도>

- 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>

## 03 디코더

### 2. $2 \times 4$ 디코더

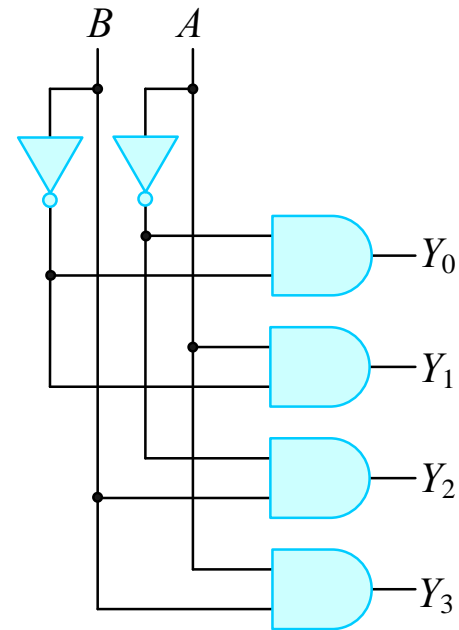
- 2개의 입력에 따라서 4개의 출력 중 하나가 선택

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>

## 03 디코더

### ■ 2×4 NAND 디코더

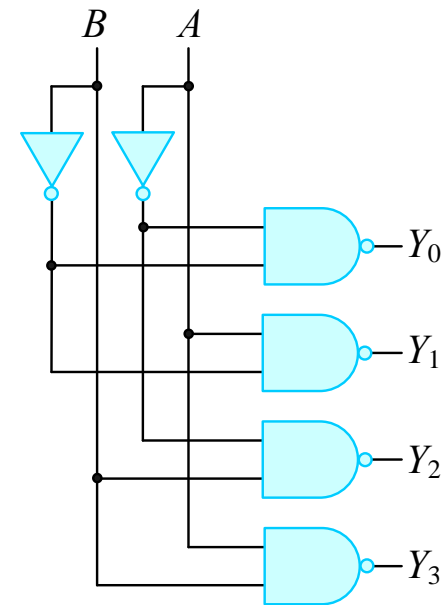
- 실제 IC들은 AND게이트가 아닌 NAND 게이트로 구성

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

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

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

<진리표와 논리식>



<회로도>

## 03 디코더

### ■ 인에이블이 있는 2×4 디코더

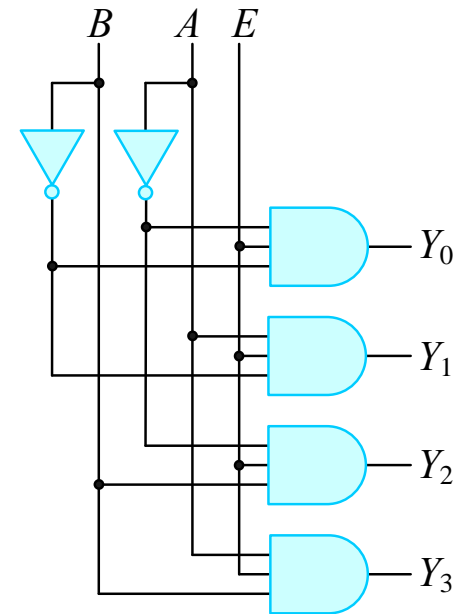
- 대부분의 IC 디코더들은 인에이블(enable) 입력이 있어서 회로를 제어한다.
- $E=1$ 일 때만 출력이 동작

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>



## 03 디코더

### ■ 인에이블이 있는 2×4 NAND 디코더

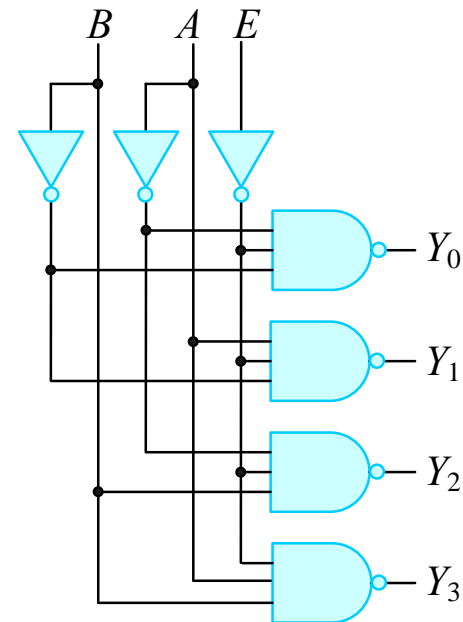
- NAND 게이트로 구성된 인에이블(enable) 입력이 있는 회로
- $E=0$ 일 때만 출력이 동작

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

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

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

<진리표와 논리식>

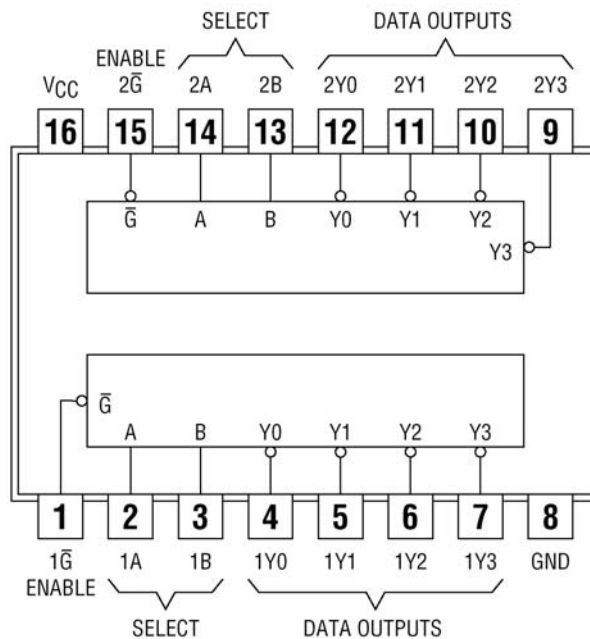


<회로도>

## 03 디코더

### ■ IC 74139 구성도

- 인에이블 단자를 갖는 2×4 디코더를 두 개 가지고 있는 IC



<IC 74139 핀 배치도>

## 03 디코더

### 3. 3×8 디코더

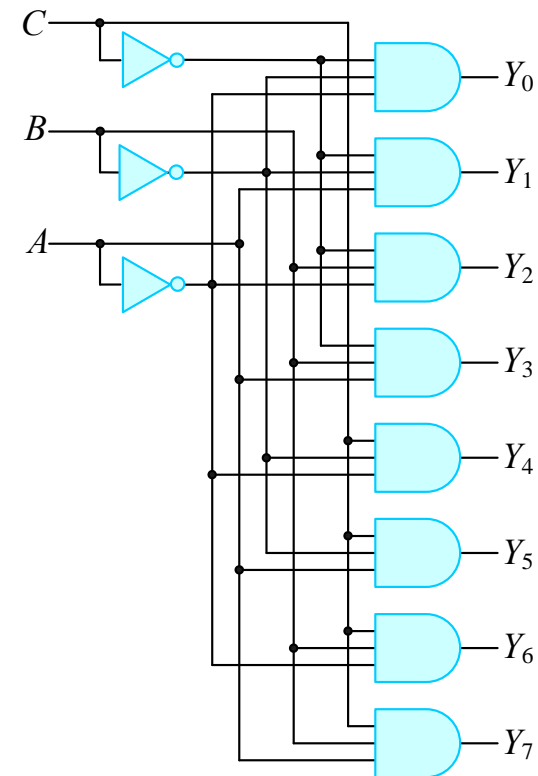
- 3개의 입력에 따라서 8개의 출력 중 하나가 선택

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>

## 03 디코더

### ■ IC 74138(3×8 디코더)

- 3개의 입력에 따라서 8개의 출력 중 하나가 선택
- 세 개의 इन에이블 단자를 가지고 있음

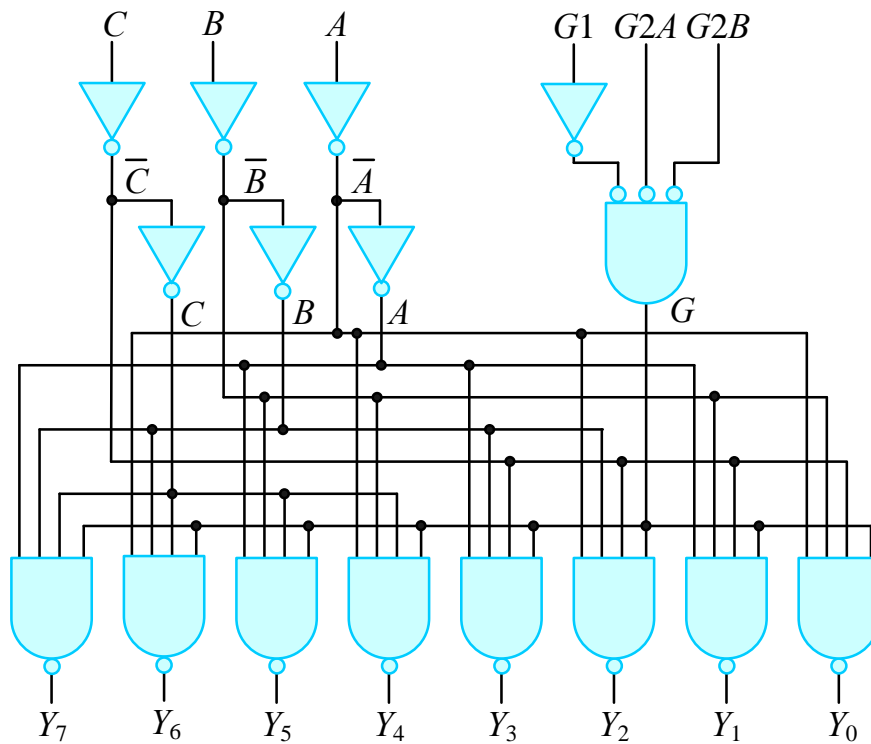
active-low

논리회로의 출력이 활성화 되었을 때 Low이거나 활성화되기 위하여 필요한 입력이 Low인 것.

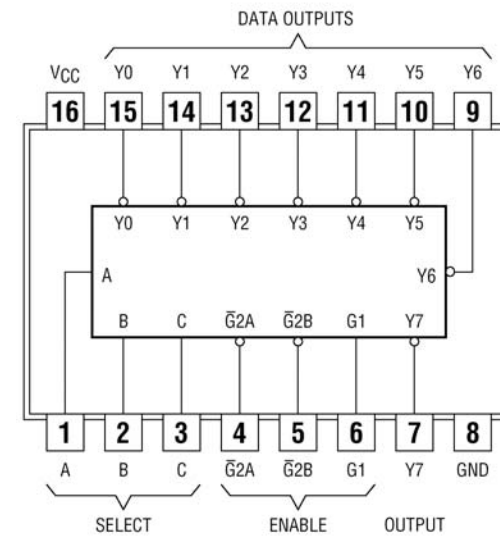
입력				출력							
<i>C B A</i>	<i>G<sub>1</sub></i>	<i>G<sub>2A</sub></i>	<i>G<sub>2B</sub></i>	<i>Y<sub>7</sub></i>	<i>Y<sub>6</sub></i>	<i>Y<sub>5</sub></i>	<i>Y<sub>4</sub></i>	<i>Y<sub>3</sub></i>	<i>Y<sub>2</sub></i>	<i>Y<sub>1</sub></i>	<i>Y<sub>0</sub></i>
0 0 0	1	0	0	1	1	1	1	1	1	1	0
0 0 1	1	0	0	1	1	1	1	1	1	0	1
0 1 0	1	0	0	1	1	1	1	1	0	1	1
0 1 1	1	0	0	1	1	1	1	0	1	1	1
1 0 0	1	0	0	1	1	1	0	1	1	1	1
1 0 1	1	0	0	1	1	0	1	1	1	1	1
1 1 0	1	0	0	1	0	1	1	1	1	1	1
1 1 1	1	0	0	0	1	1	1	1	1	1	1
× × ×	0	×	×	1	1	1	1	1	1	1	1
× × ×	×	1	×	1	1	1	1	1	1	1	1
× × ×	×	×	1	1	1	1	1	1	1	1	1

<IC 74138 진리표>

# 03 디코더



<IC 74138 내부 회로도>



<IC 74138 핀 배치도>

## 03 디코더

### 4. $4 \times 16$ 디코더

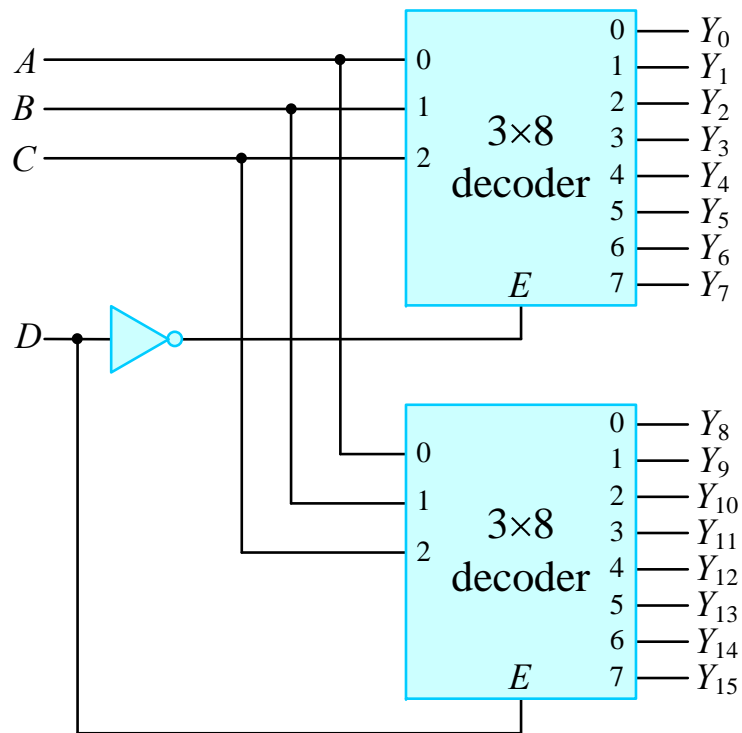
$D$	$C$	$B$	$A$	$Y_{15}$	$Y_{14}$	$Y_{13}$	$Y_{12}$	$Y_{11}$	$Y_{10}$	$Y_9$	$Y_8$	$Y_7$	$Y_6$	$Y_5$	$Y_4$	$Y_3$	$Y_2$	$Y_1$	$Y_0$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	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

< $4 \times 16$  디코더 진리표>

## 03 디코더

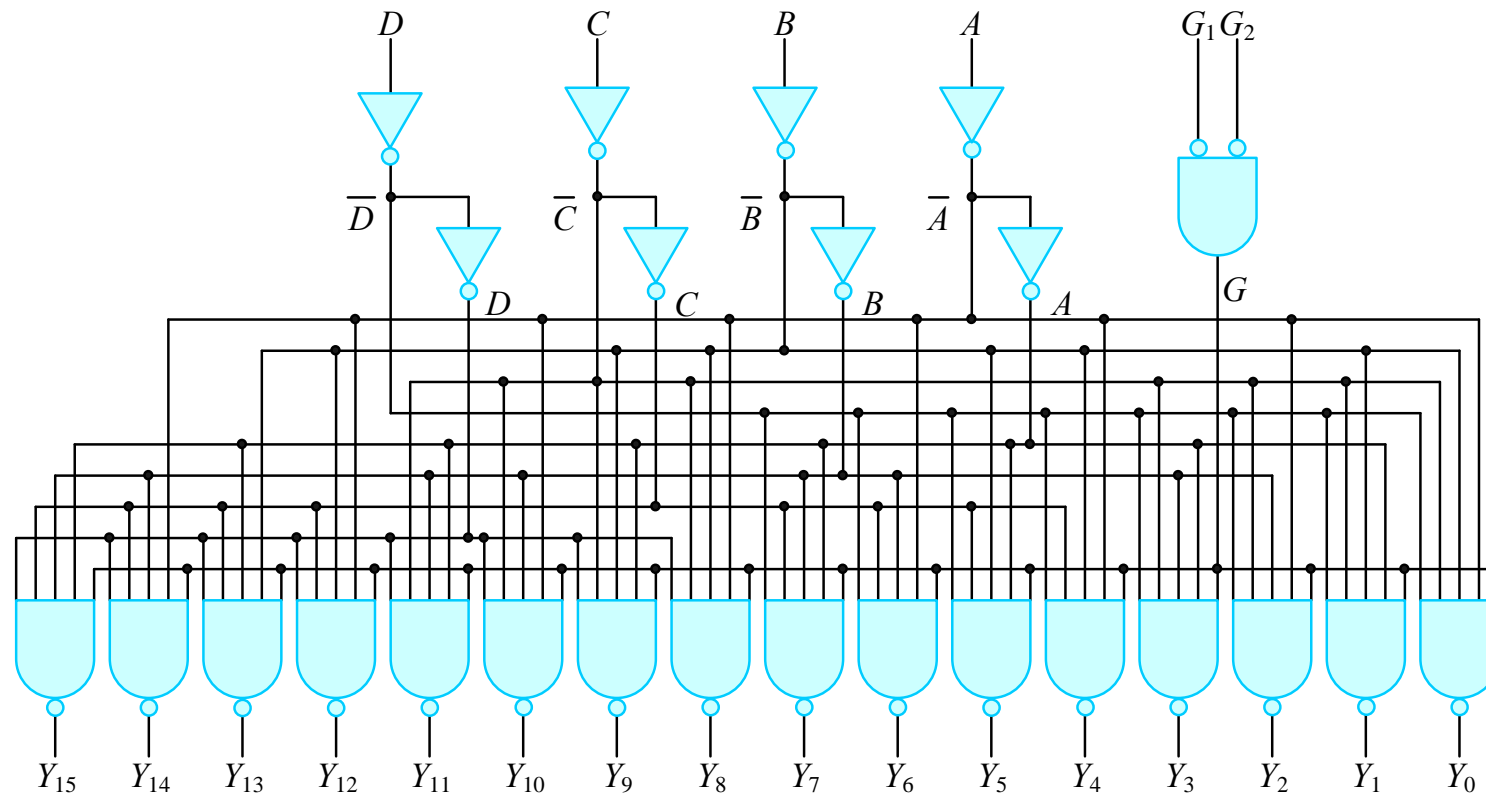
### ■ 2개의 $3 \times 8$ 디코더로 $4 \times 16$ 디코더를 구성

$D=0$	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.



## 03 디코더

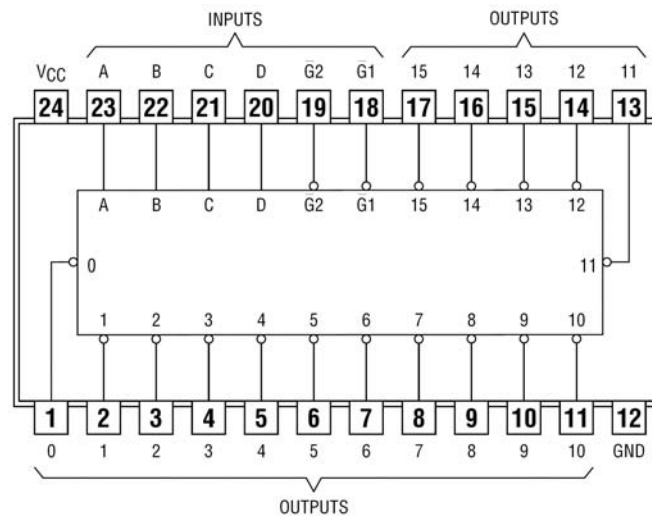
### ■ IC 74154



<IC 74154 회로도>



## 03 디코더



<IC 74154 핀 배치도>

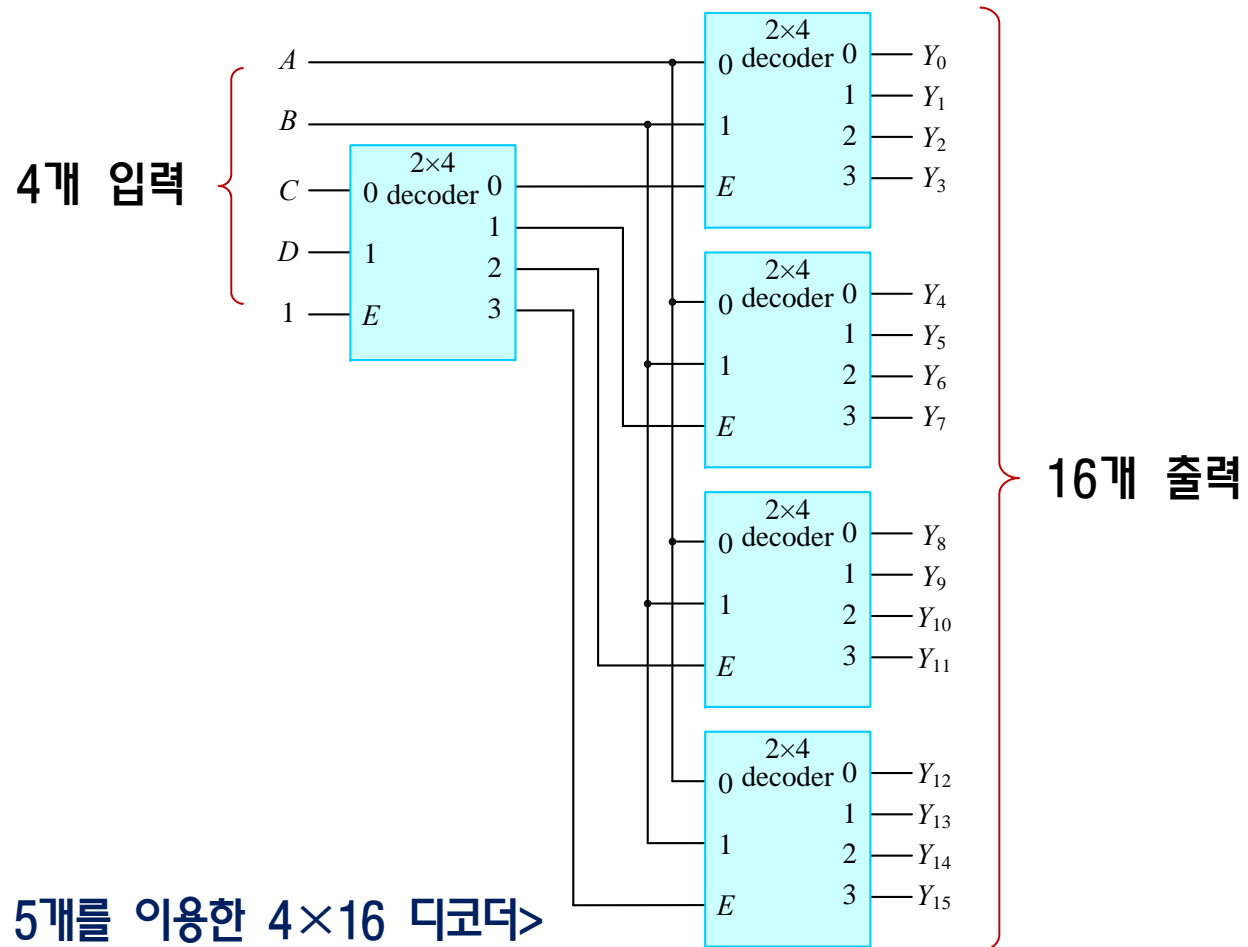
## 03 디코더

### ■ 2×4 디코더 5개를 이용한 4×16 디코더

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

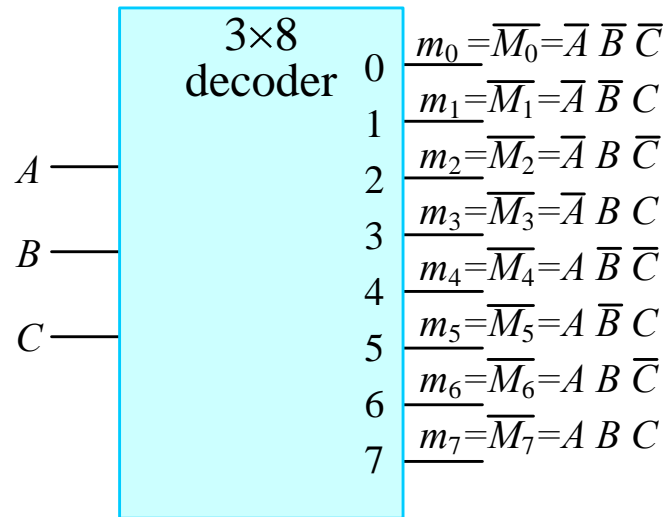
## 03 디코더



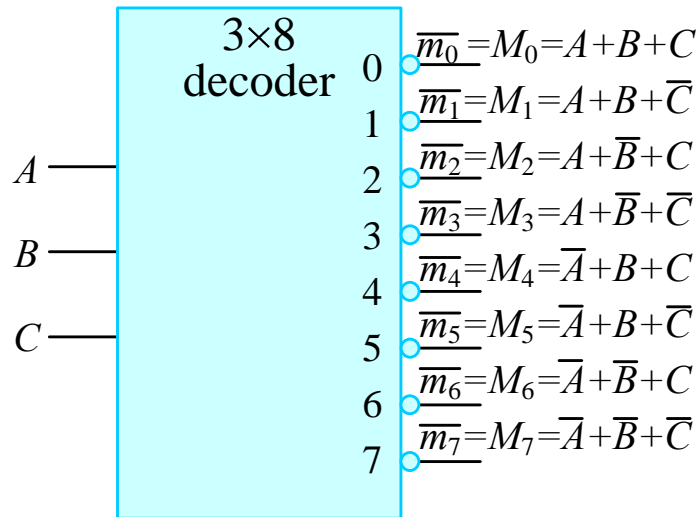
## 03 디코더

### 5. 디코더를 이용한 조합논리회로

❖ 3×8 디코더를 이용하는 경우



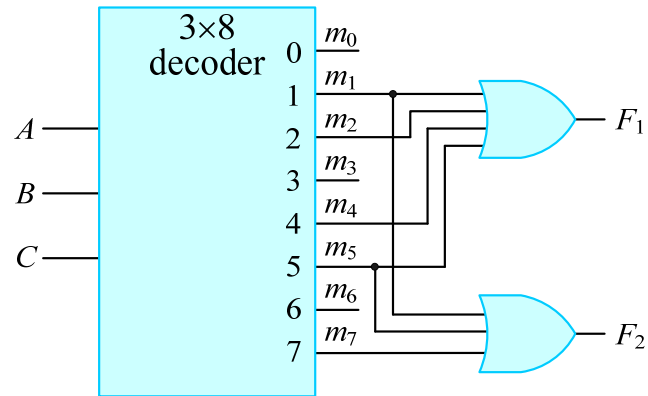
<3×8 디코더 출력>



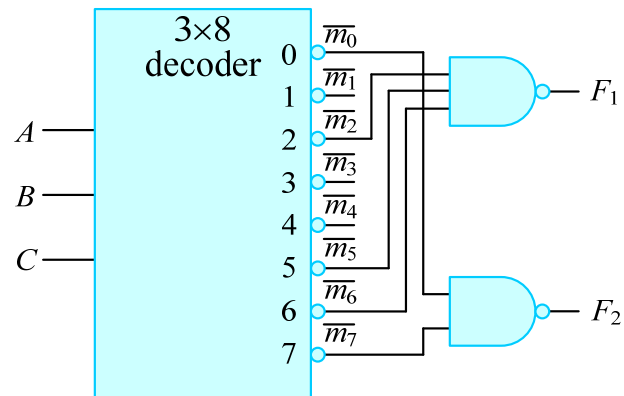
<3×8 디코더 반전출력>

## 03 디코더

❖ 3×8 디코더를 이용하는 경우의 예



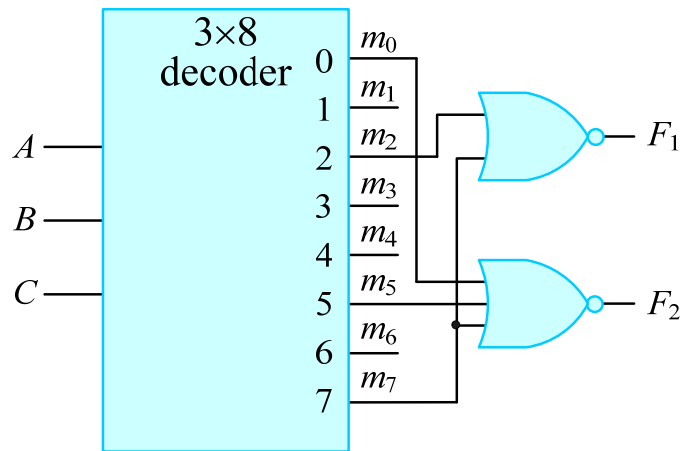
$$F_1(A, B, C) = \sum m(1, 2, 4, 5)$$
$$F_2(A, B, C) = \sum m(1, 5, 7)$$



$$F_1(A, B, C) = \sum m(2, 5, 6)$$
$$F_2(A, B, C) = \sum m(0, 7)$$

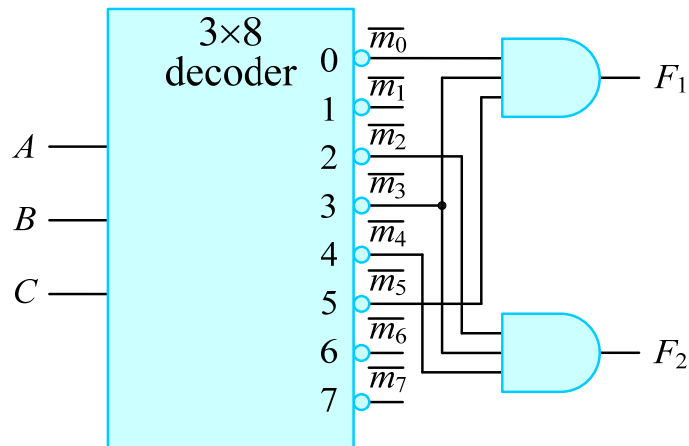
## 03 디코더

❖ 3×8 디코더를 이용하는 경우의 예



$$F_1(A, B, C) = \prod M(2, 7)$$

$$F_2(A, B, C) = \prod M(0, 5, 7)$$



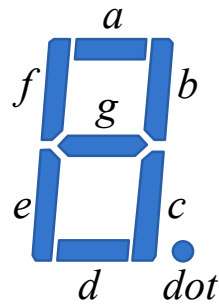
$$F_1(A, B, C) = \prod M(0, 3, 5)$$

$$F_2(A, B, C) = \prod M(2, 3, 4)$$

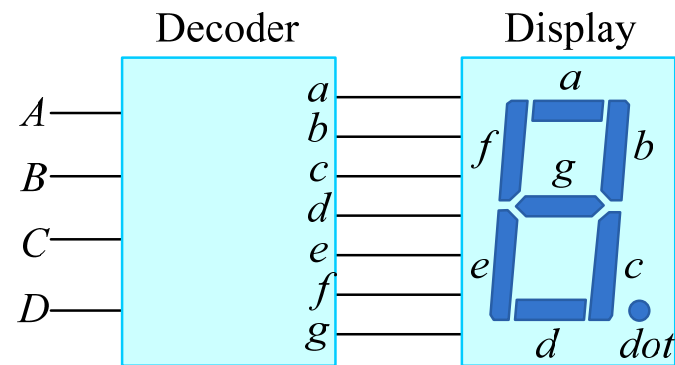
## 03 디코더

### 6. BCD-7-세그먼트 디코더

- 7 세그먼트 : 숫자 표시 전용 장치



<7-세그먼트 구성>



<7-세그먼트와 디코더의 연결>

0	1	2	3	4	5	6	7	8	9

<7-세그먼트의 숫자 표시>

## 03 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	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x



# 03 디코더

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00		1		
01	1			1
11	x	x	x	x
10			x	x

$$\bar{a} = \overline{DCBA} + C\bar{A}$$

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00				
01		1		1
11	x	x	x	x
10			x	x

$$\bar{b} = C\bar{B}A + CB\bar{A} = C(B \oplus A)$$

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00				1
01				
11	x	x	x	x
10			x	x

$$\bar{c} = \overline{CB}A$$

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00		1		
01	1		1	
11	x	x	x	x
10		1	x	x

$$\bar{d} = \overline{CB}A + C\bar{B}\bar{A} + CBA$$

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00		1	1	
01	1	1	1	
11	x	x	x	x
10		1	x	x

$$\bar{e} = A + C\bar{B}$$

$\begin{smallmatrix} BA \\ DC \end{smallmatrix}$	00	01	11	10
00		1	1	
01			1	
11	x	x	x	x
10			x	x

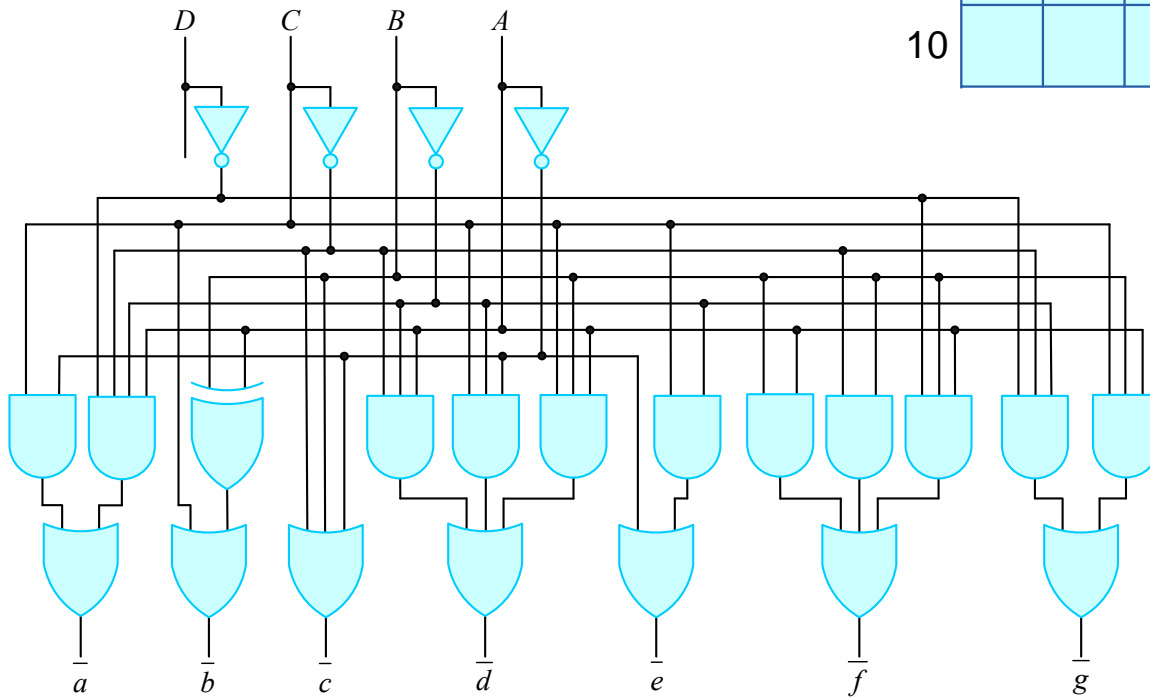
$$\bar{f} = BA + \overline{CB} + \overline{DC}A$$

<카르노 맵>

# 03 디코더

$\begin{matrix} BA \\ DC \end{matrix}$	00	01	11	10
00	1	1		
01			1	
11	x	x	x	x
10			x	x

$$\overline{g} = \overline{D}\overline{C}\overline{B} + CBA$$



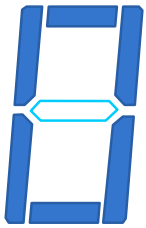
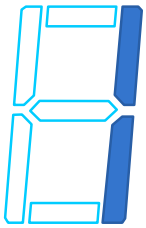
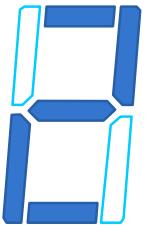


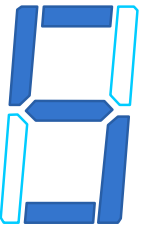
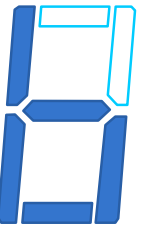
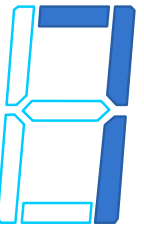


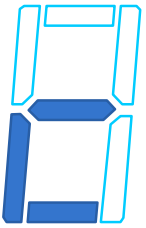

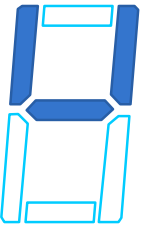
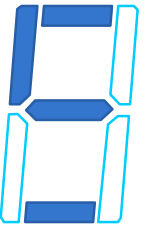
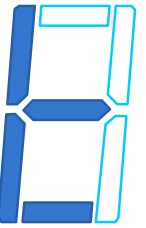
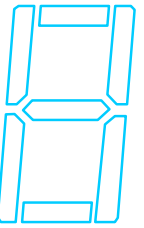
<회로도>

# 03 디코더

<7447 진리표>

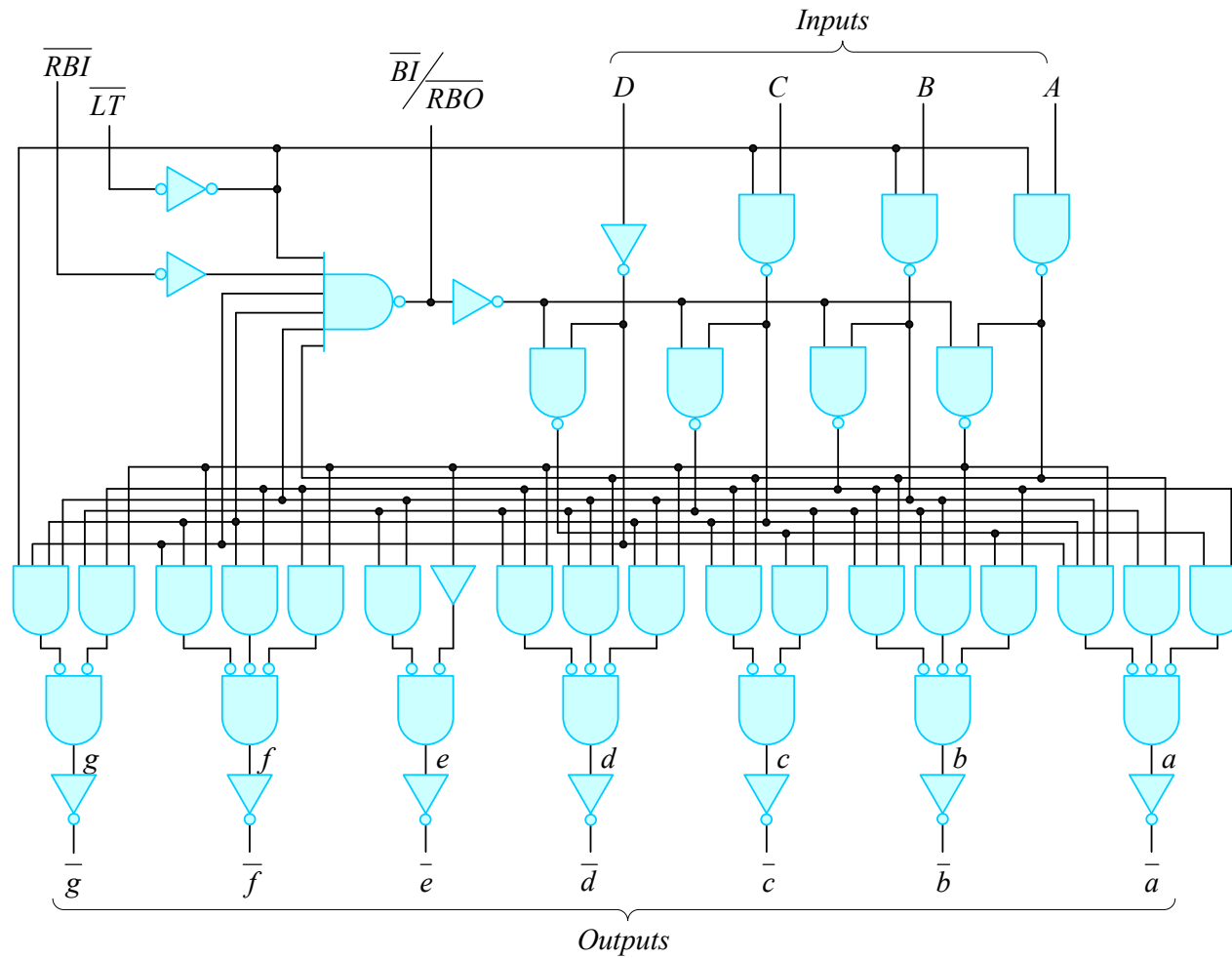
10진값 또는 기능	입력							출력						
	$\overline{LT}$	$\overline{RBI}$	$D$	$C$	$B$	$A$	$\frac{BI}{\overline{RBO}}$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	×	0	0	0	1	1	1	0	0	1	1	1	1
2	1	×	0	0	1	0	1	0	0	1	0	0	1	0
3	1	×	0	0	1	1	1	0	0	0	0	1	1	0
4	1	×	0	1	0	0	1	1	0	0	1	1	0	0
5	1	×	0	1	0	1	1	0	1	0	0	1	0	0
6	1	×	0	1	1	0	1	1	1	0	0	0	0	0
7	1	×	0	1	1	1	1	0	0	0	1	1	1	1
8	1	×	1	0	0	0	1	0	0	0	0	0	0	0
9	1	×	1	0	0	1	1	0	0	0	1	1	0	0
10	1	×	1	0	1	0	1	1	1	1	0	0	1	0
11	1	×	1	0	1	1	1	1	1	0	0	1	1	0
12	1	×	1	1	0	0	1	1	0	1	1	1	0	0
13	1	×	1	1	0	1	1	0	1	1	0	1	0	0
14	1	×	1	1	1	0	1	1	1	1	0	0	0	0
15	1	×	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{BI}$	×	×	×	×	×	×	0	1	1	1	1	1	1	1
$\overline{RBI}$	1	0	0	0	0	0	0	1	1	1	1	1	1	1
$\overline{LT}$	0	×	×	×	×	×	1	0	0	0	0	0	0	0

## 03 디코더

0	1	2	3	4	5	6	7
							
8	9	10	11	12	13	14	15
							

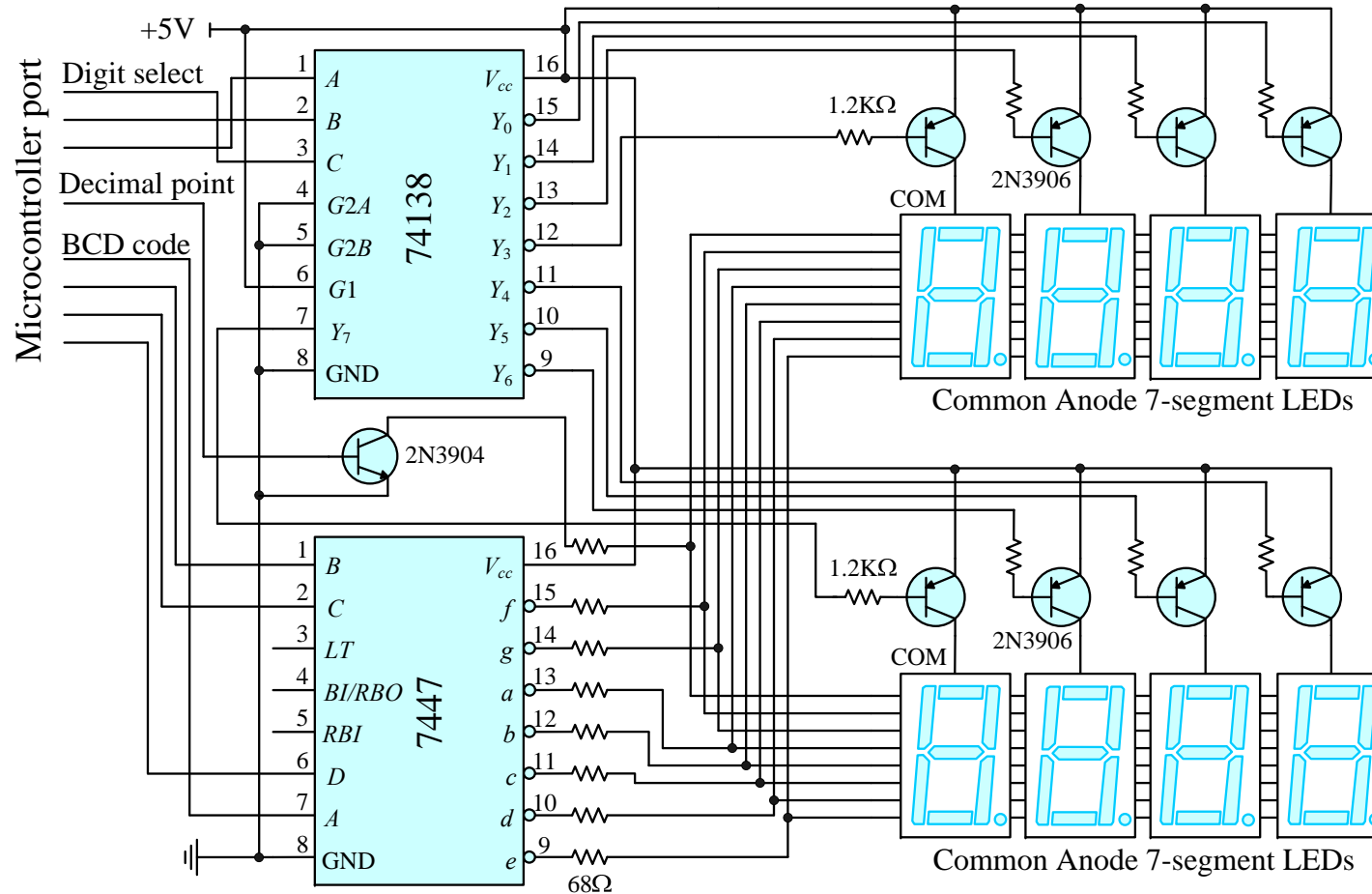
<7447의 LED 점등 패턴도>

## 03 디코더



<7447 회로도>

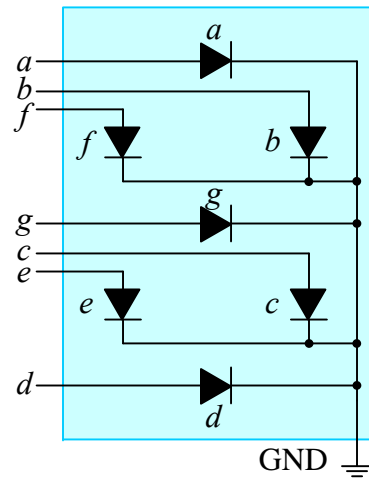
# 03 디코더



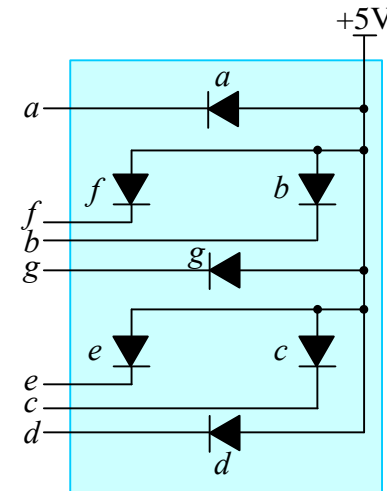
<7447을 이용한 7-세그먼트 구동 회로 예>

# 03 디코더

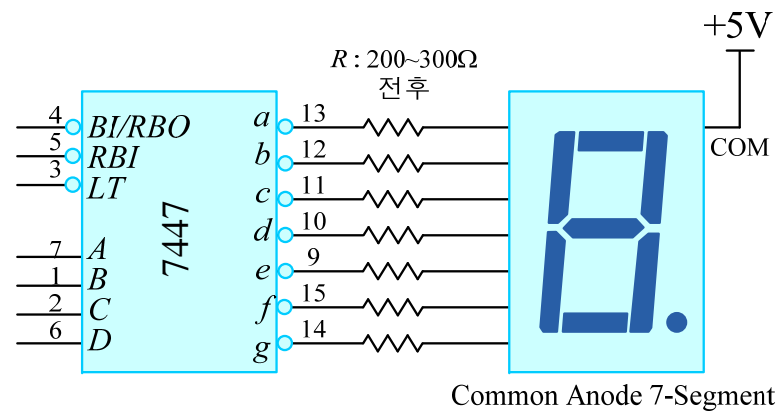
## ■ 7-세그먼트 공통 회로



<캐소드 공통>



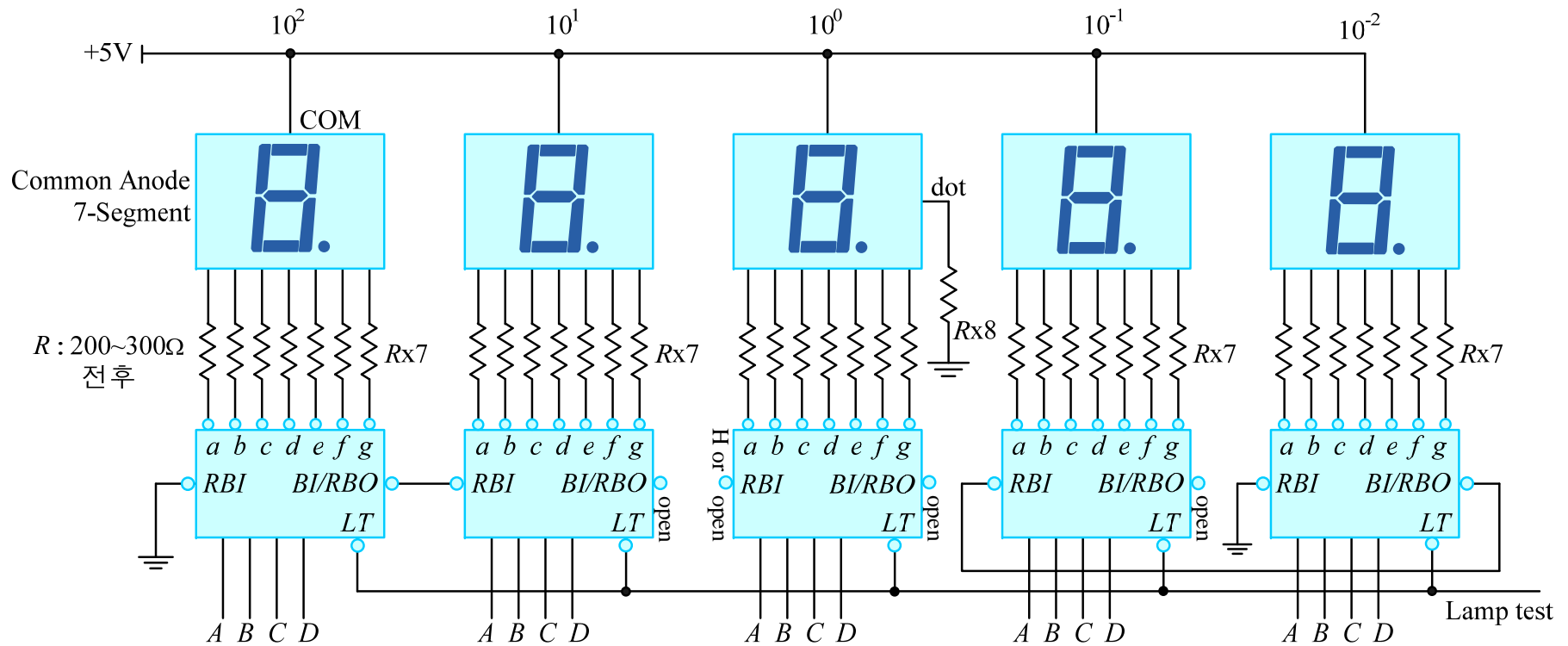
<애노드 공통>



Common Anode 7-Segment

<전류 제한 저항을 사용한 7-세그먼트 회로의 예>

## 03 디코더



<7-세그먼트의 LT, RBI, BI/RBO 사용 예>



## 04 인코더

- 인코더(encoder)는 디코더의 반대기능을 수행하는 장치로써,  $2^n$ 개의 입력신호로부터  $n$ 개의 출력신호를 만든다.
- 인코더의 역할은  $2^n$  개중 활성화된 하나의 1비트입력 신호를 받아서 그 숫자에 해당하는  $n$  비트 2진 정보를 출력한다.

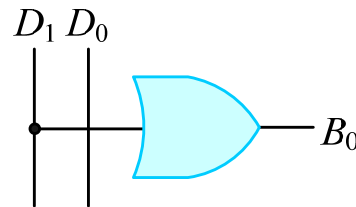
### 1. $2 \times 1$ 인코더

- 입력의 신호에 따라 2개의 2진 조합으로 출력된다.

입력		출력
$D_1$	$D_0$	$B_0$
0	1	0
1	0	1

$$B_0 = D_1$$

<진리표와 논리식>



<회로도>

# 04 인코더

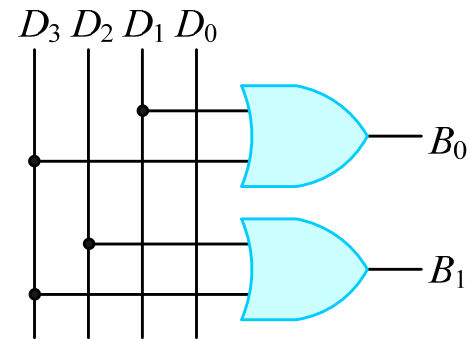
## 2. 4×2 인코더

- 입력의 신호에 따라 2개의 2진 조합으로 출력된다.

입력				출력	
$D_3$	$D_2$	$D_1$	$D_0$	$B_1$	$B_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$B_1 = D_2 + D_3, \quad B_0 = D_1 + D_3$$

<진리표와 논리식>



<회로도>

# 04 인코더

## 3. 8×3 인코더

- 8( $=2^3$ )개의 입력과 3개의 출력을 가지며, 입력의 신호에 따라 3개의 2진 조합으로 출력

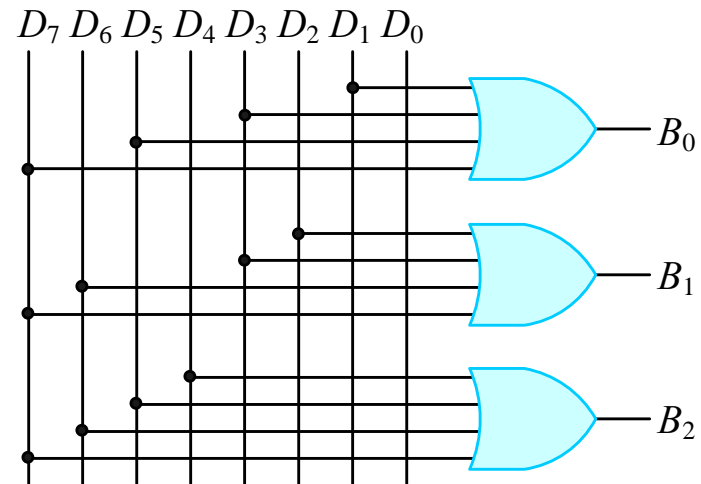
입력								출력		
$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

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

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

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

<진리표와 논리식>



<회로도>

# 04 인코더

## 4. 8×3 우선순위 인코더

- 우선순위 인코더(priority encoder)는 입력에 우선순위를 정하여 여러 개의 입력이 있을 때 우선순위가 높은 입력값에 해당되는 출력신호를 만들어 내는 회로

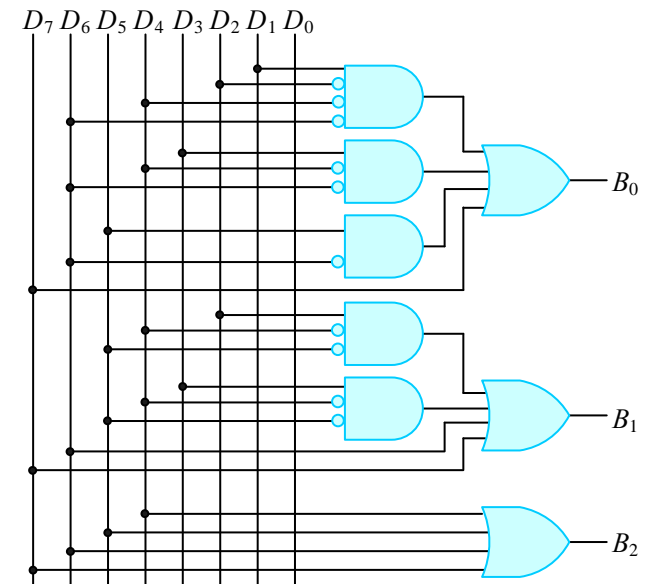
입력								출력		
$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$	$B_2$	$B_1$	$B_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	×	0	0	1
0	0	0	0	0	1	×	×	0	1	0
0	0	0	0	1	×	×	×	0	1	1
0	0	0	1	×	×	×	×	1	0	0
0	0	1	×	×	×	×	×	1	0	1
0	1	×	×	×	×	×	×	1	1	0
1	×	×	×	×	×	×	×	1	1	1

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

$$B_1 = D_7 + D_6 + \overline{D_5} \overline{D_4} D_3 + \overline{D_5} \overline{D_4} D_2$$

$$B_0 = D_7 + \overline{D_6} D_5 + \overline{D_6} \overline{D_4} D_3 + \overline{D_6} \overline{D_4} \overline{D_2} D_1$$

<진리표와 논리식>



<회로도>

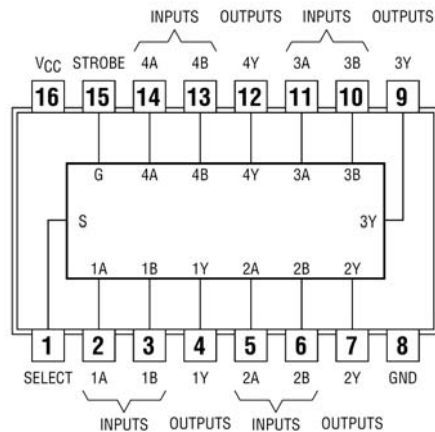
# 04 인코더

## 5. 인코더 IC

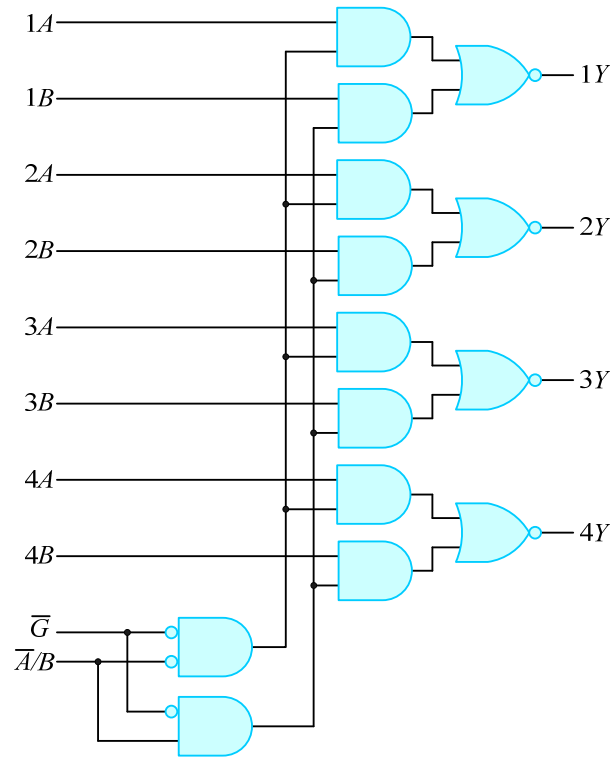
### ■ 74158 : 2×1 인코더/멀티플렉서가 4개 내장

입력		출력
<i>S</i>	<i>E</i>	<i>Y</i>
×	1	1
0	0	
1	0	

<진리표>



<핀 배치도>



<회로도>

## 04 인코더

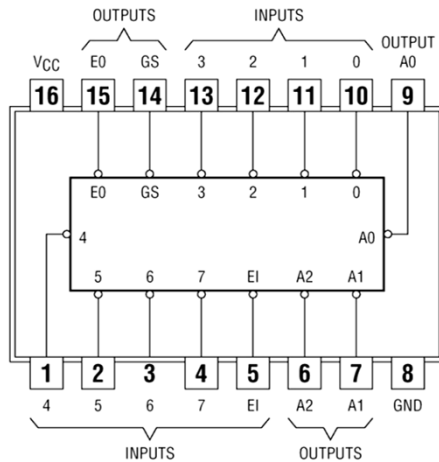
### ■ 74148(8×3 우선순위 인코더)

- 8개의 논리반전 입력(0-7)과 3개의 논리반전 출력을 가지는 우선순위 인코더이다.  
. 가장 우선순위가 높은 것은 7번이다.
- *GS*는 데이터 입력 중의 하나가 0이고 *EI*가 0일 때만 0이 된다.
- *EI*와 *EO*는 74148을 여러 개 연결할 때 사용

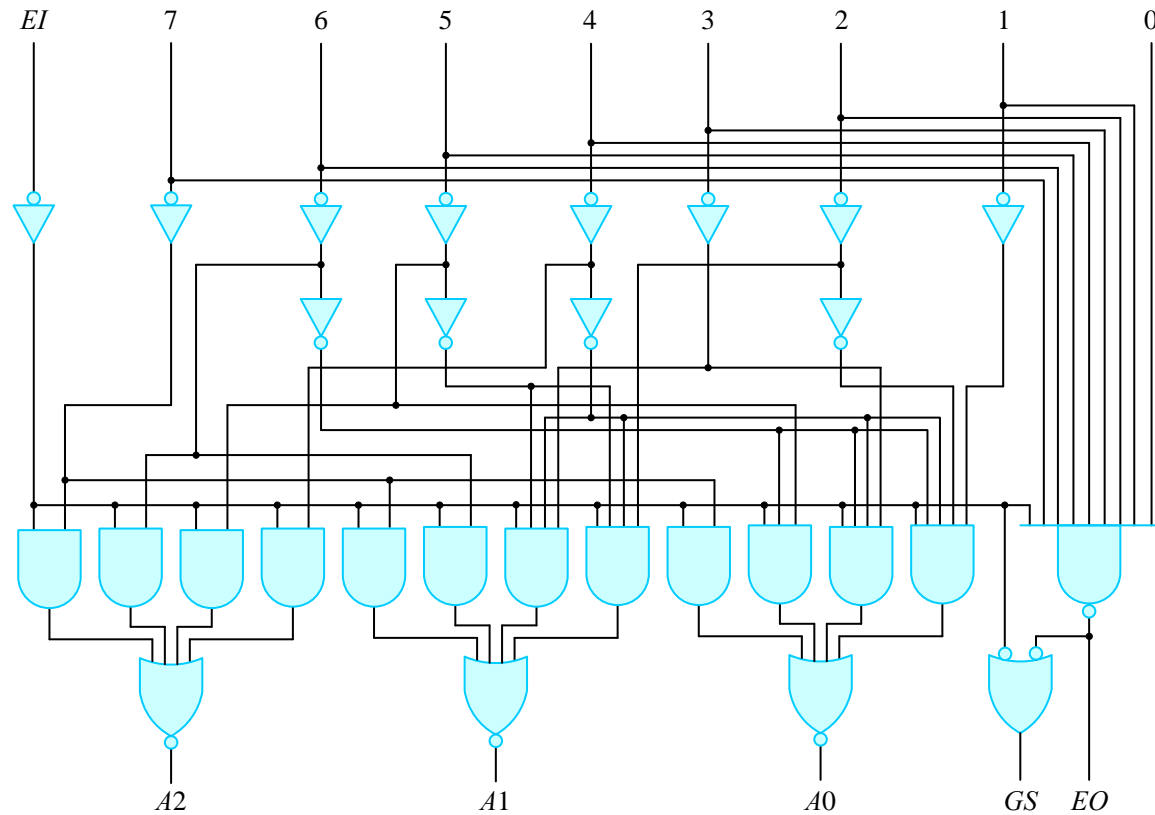
입력									출력				
<i>EI</i>	7	6	5	4	3	2	1	0	<i>A</i> <sub>2</sub>	<i>A</i> <sub>1</sub>	<i>A</i> <sub>0</sub>	<i>GS</i>	<i>EO</i>
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	1	1	0	×	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1

<진리표>

# 04 인코더



<IC 74148 핀 배치도>



<회로도>

## 04 인코더

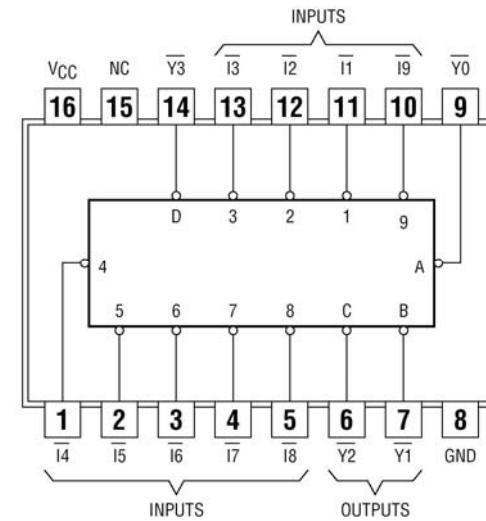
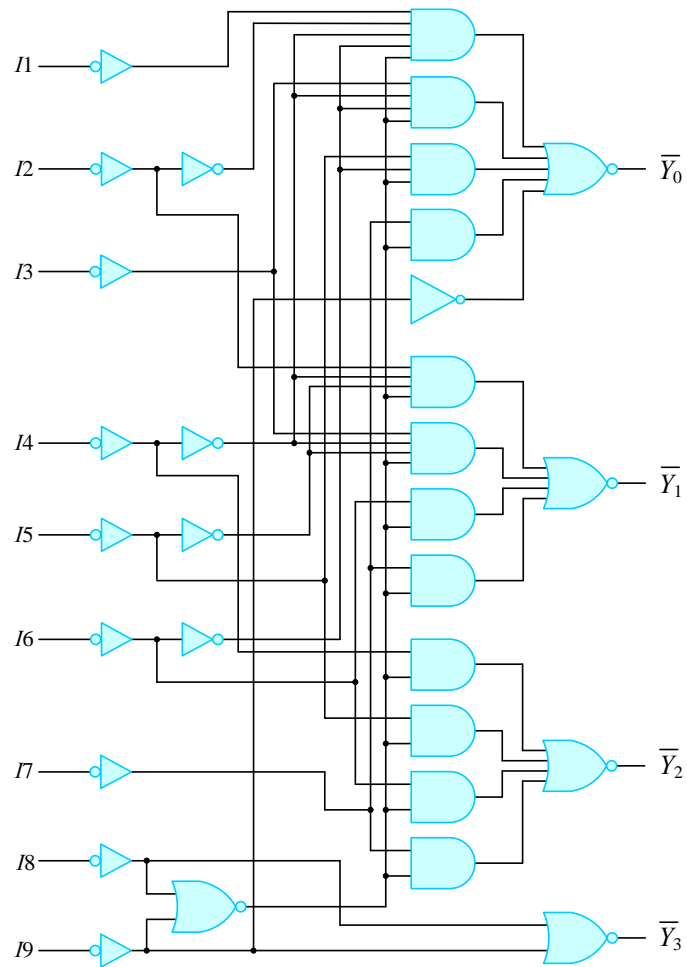
### 6. 10진-BCD 우선순위 인코더

입력									출력			
<i>I9</i>	<i>I8</i>	<i>I7</i>	<i>I6</i>	<i>I5</i>	<i>I4</i>	<i>I3</i>	<i>I2</i>	<i>I1</i>				
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	×	1	1	0	1
1	1	1	1	1	1	0	×	×	1	1	0	0
1	1	1	1	1	0	×	×	×	1	0	1	1
1	1	1	1	0	×	×	×	×	1	0	1	0
1	1	1	0	×	×	×	×	×	1	0	0	1
1	1	0	×	×	×	×	×	×	1	0	0	0
1	0	×	×	×	×	×	×	×	0	1	1	1
0	×	×	×	×	×	×	×	×	0	1	1	0

<진리표>



# 04 인코더

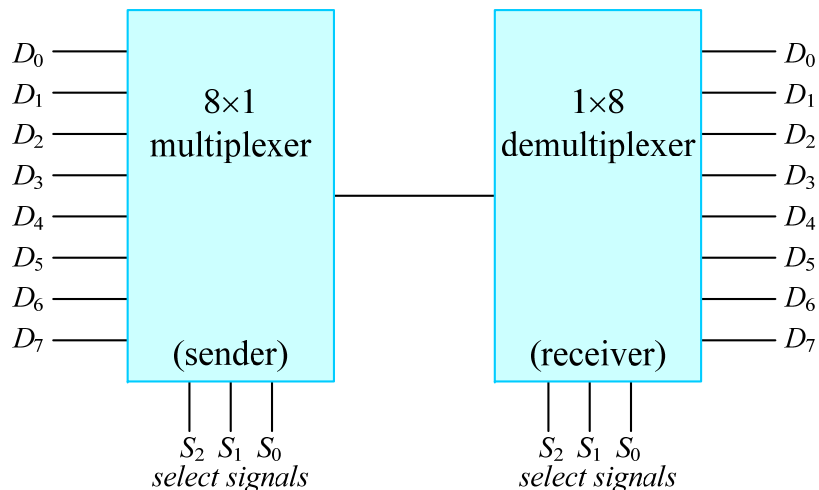


<IC 74147 핀 배치도>

<IC 74147 회로도>

## 05 멀티플렉서

- 멀티플렉서(multiplexer or selector)는 여러 개의 입력선들 중에서 하나를 선택하여 출력선에 연결하는 조합논리회로이다. 선택선들의 값에 따라서 특별한 입력선이 선택된다.
- 멀티플렉서는 많은 입력들 중 하나를 선택하여 선택된 입력선의 2진 정보를 출력선에 넘겨주기 때문에 데이터 선택기(data selector)라 부르기도 한다.
- 하나의 신호선을 이용하여 다수의 정보들을 전송할 수 있는 경제적인 장점



<멀티플렉서와 디멀티플렉서의 역할>

## 05 멀티플렉서

### 1. $2 \times 1$ 멀티플렉서

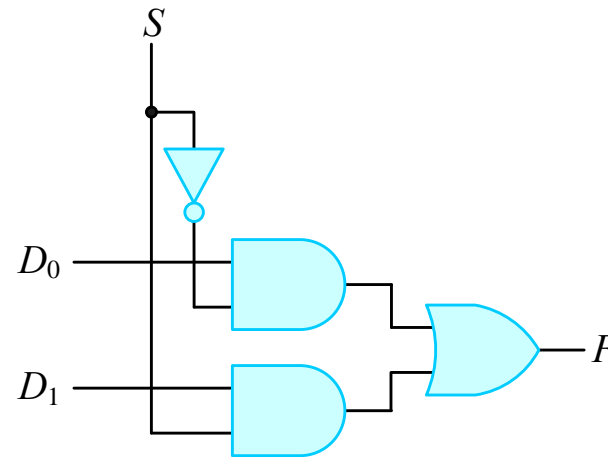
- 2( $=2^1$ )개의 입력중의 하나를 선택선  $S$ 에 입력된 값에 따라서 출력으로 보내주는 조합회로

선택선	출력
$S$	$F$
0	$D_0$
1	$D_1$

<진리표>

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

<논리식>



<회로도>

# 05 멀티플렉서

## 2. $4 \times 1$ 멀티플렉서

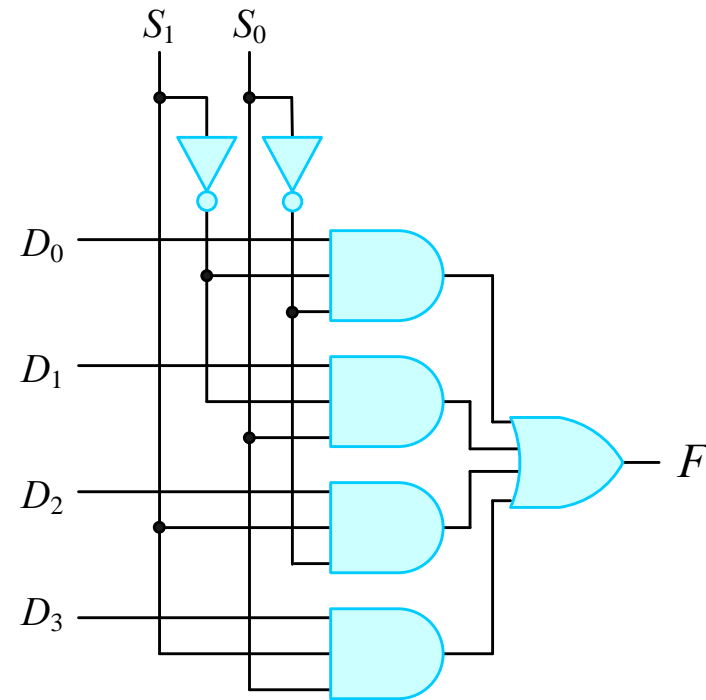
- 4( $=2^2$ )개의 입력중의 하나를 선택선  $S_1$ 과  $S_0$ 에 입력된 값에 따라서 출력으로 보내 주는 조합회로

선택선		출력
$S_1$	$S_0$	$F$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

<진리표>

$$F = \overline{S_1}\overline{S_0}D_0 + \overline{S_1}S_0D_1 + S_1\overline{S_0}D_2 + S_1S_0D_3$$

<논리식>

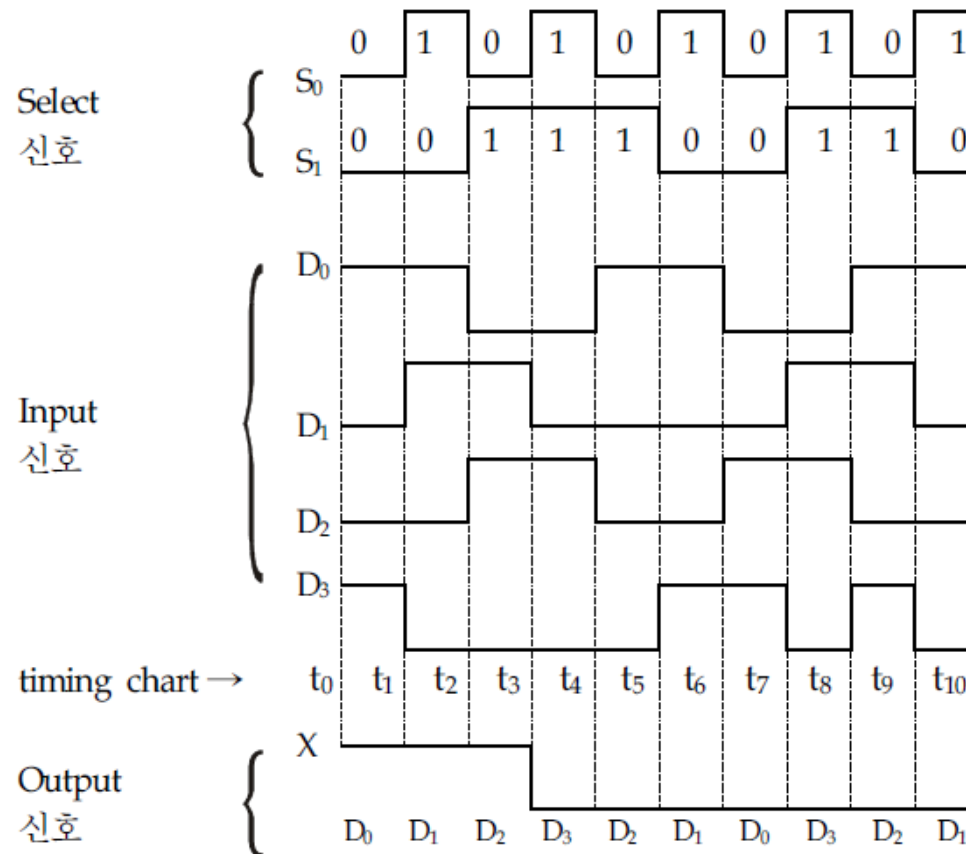


<회로도>

## 05 멀티플렉서

### 2. $4 \times 1$ 멀티플렉서

데이터 입력단자와 데이터 선택단자에 파형을 인가하였을 때 출력단자에 나타나는 파형



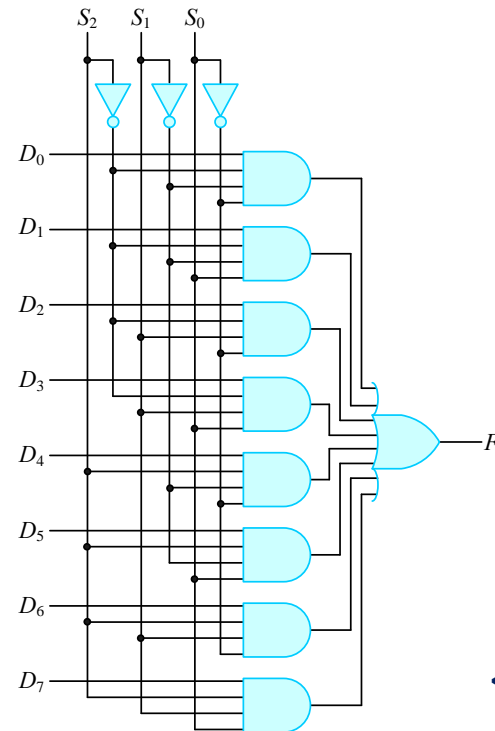
## 05 멀티플렉서

### 3. $8 \times 1$ 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>



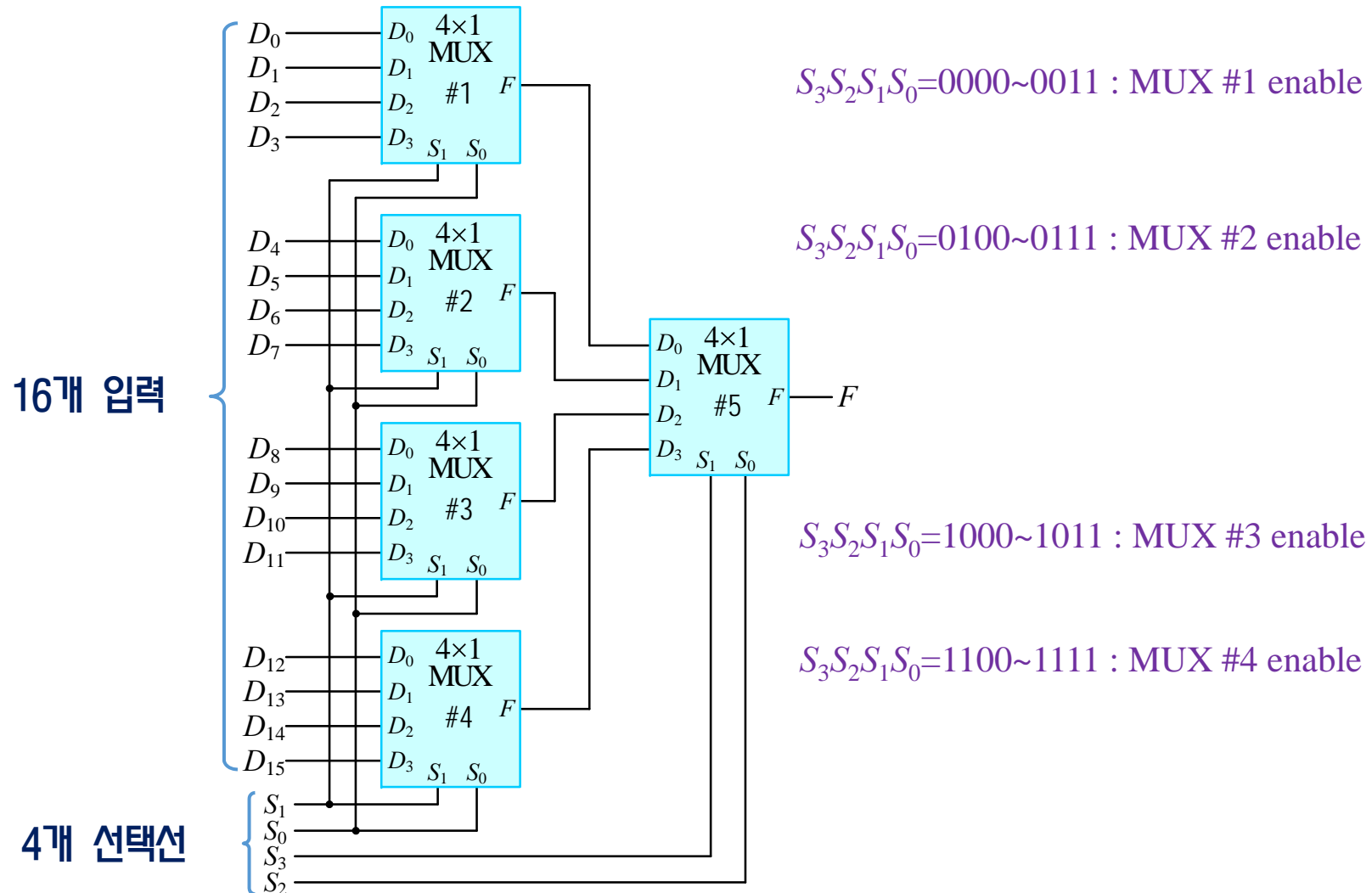
<회로도>

<논리식>

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

## 05 멀티플렉서

### ■ 4×1 멀티플렉서 5개를 이용한 16×1 멀티플렉서



## 05 멀티플렉서

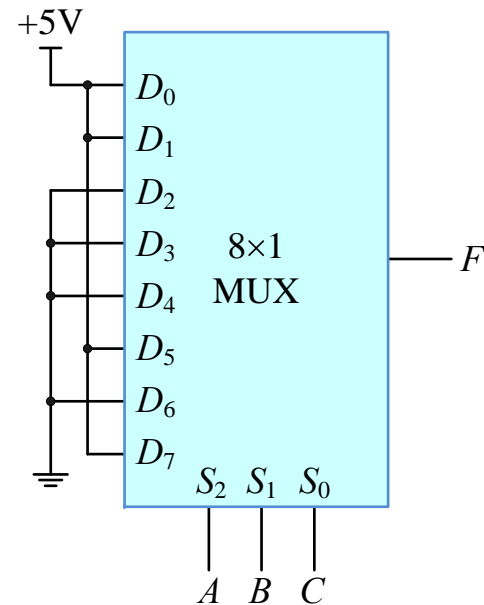
### 5. 멀티플렉서를 이용한 조합회로 구현

■  $F(A, B, C) = \sum m(0, 1, 5, 7)$  를  $8 \times 1$  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>



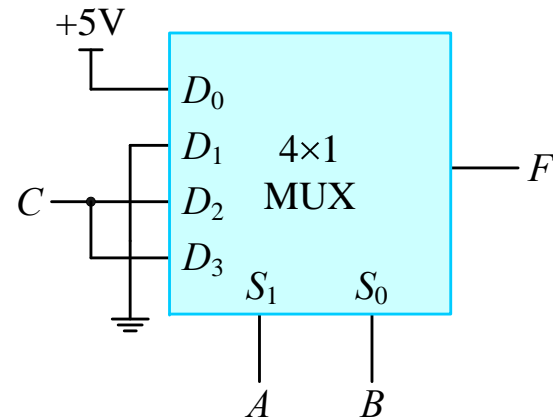
## 05 멀티플렉서

■  $F(A, B, C) = \sum m(0, 1, 5, 7)$  를  $4 \times 1$  멀티플렉서로 구현하는 경우

☞  $A, B$  는 선택선으로  $C$  는  $D_0, D_1, D_2, D_3$ 을 조합하여 사용

$A$	$B$	$C$	$F$	
0	0	0	$D_0=1$	1
		1		1
0	1	0	$D_1=0$	0
		1		0
1	0	0	$D_2=C$	0
		1		1
1	1	0	$D_3=C$	0
		1		1

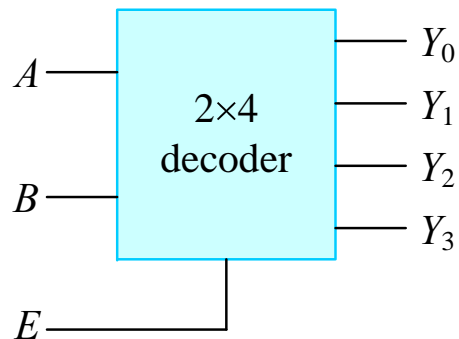
<진리표>



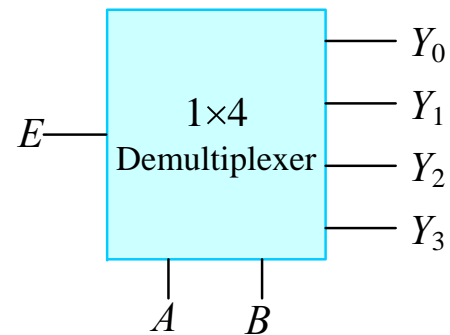
<회로도>

## 06 디멀티플렉서

- 1개의 인에이블 입력을 가지고 있는 디코더는 디멀티플렉서로서의 기능을 수행
- 디멀티플렉서는 정보를 한 선으로 받아서  $2^n$ 개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로이다. 디멀티플렉서는  $n$ 개의 선택선(selection line)들을 이용하여 출력을 제어
- 디멀티플렉서는 정보를 한 선으로 받아서  $2^n$ 개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로다. 디멀티플렉서는  $n$ 개의 선택선(selection line)의 값에 의해 하나의 출력선이 선택된다.



<2×4 디코더>



<1×4 디멀티플렉서>

## 06 디멀티플렉서

### 1. 1×4 디멀티플렉서

$S_1 = 0, S_0 = 0$ 이면 출력선  $X_0$ 가 선택되어 입력 데이터  $D$ 가 출력되며,  
 $S_1 = 0, S_0 = 1$ 이면 출력선  $X_1$ 가 선택되어 입력 데이터  $D$ 가 출력되고,  
 $S_1 = 1, S_0 = 0$ 이면 출력선  $X_2$ 가 선택되어 입력 데이터  $D$ 가 출력되며,  
 $S_1 = 1, S_0 = 1$ 이면 출력선  $X_3$ 가 선택되어 입력 데이터  $D$ 가 출력된다.

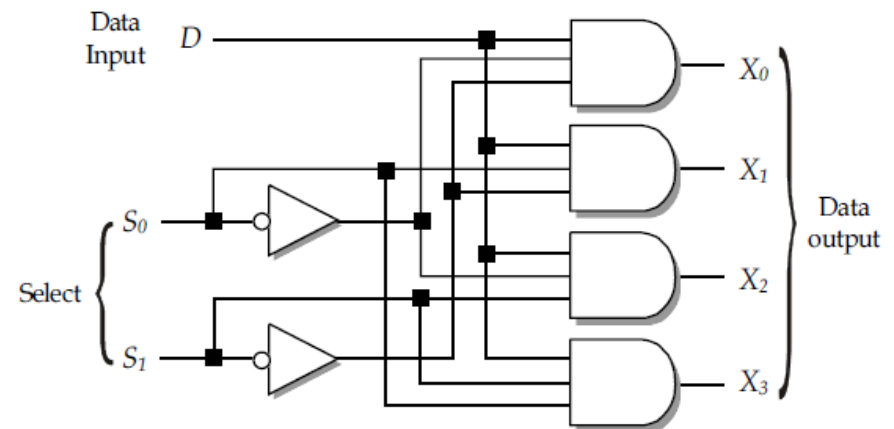
데이터 선택 단자		데이터 입력	출력
$S_1$	$S_0$		$X$
0	0	$D$	$\overline{S_1} \overline{S_0} D$
0	1	$D$	$\overline{S_1} S_0 D$
1	0	$D$	$S_1 \overline{S_0} D$
1	1	$D$	$S_1 S_0 D$

<논리식>

$$X_0 = \overline{S_1} \overline{S_0} D$$

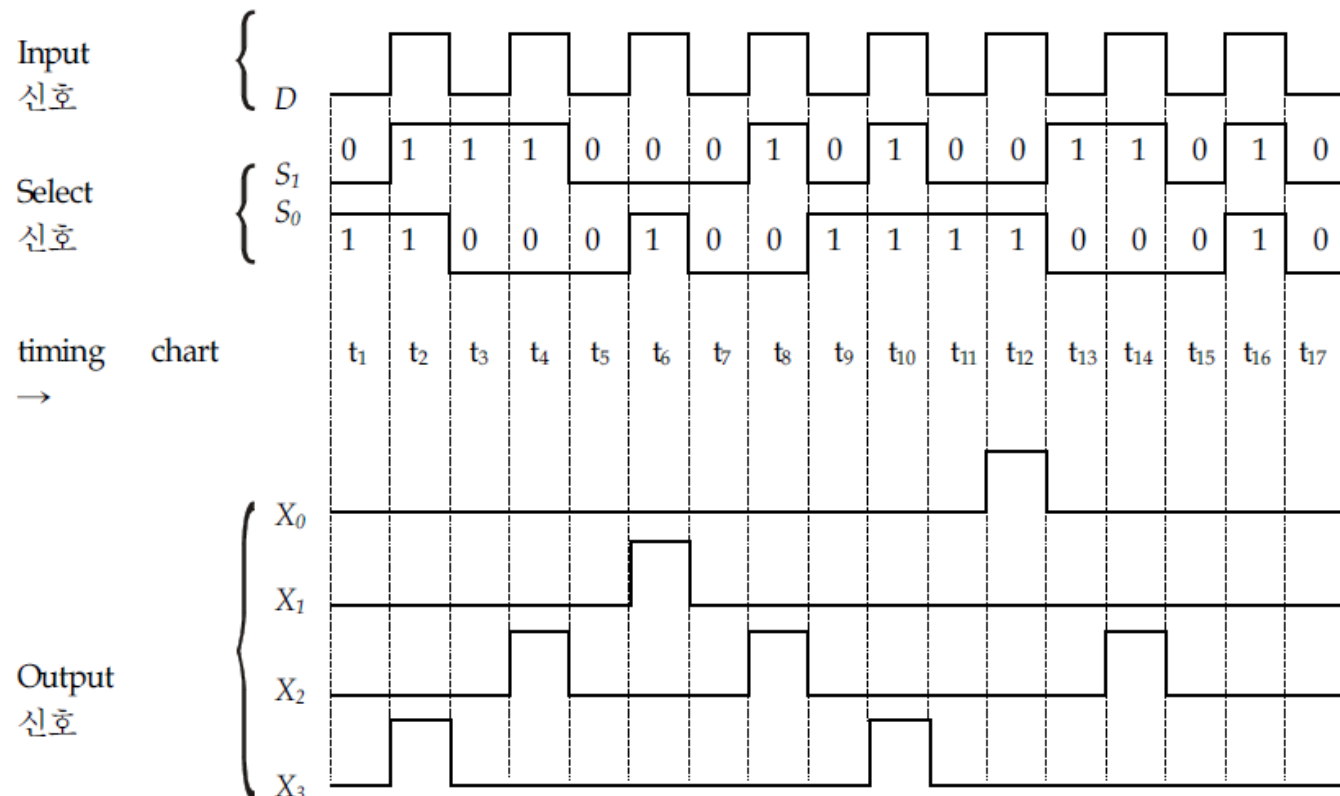
$$X_2 = S_1 \overline{S_0} D$$

<회로도>



$$X_1 = \overline{S_1} S_0 D$$

$$X_3 = S_1 S_0 D$$



## 07 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 1 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	1 0 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

# 07 코드 변환기

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

$$G_3 = B_3$$

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

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

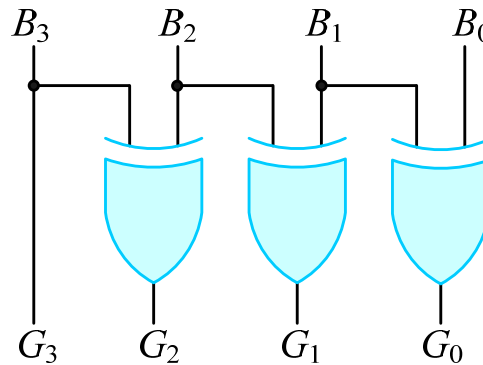
$B_3B_2 \backslash B_1B_0$	00	01	11	10
00			1	1
01	1	1		
11	1	1		
10			1	1

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

$B_3B_2 \backslash B_1B_0$	00	01	11	10
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

# 07 코드 변환기

## 2. 그레이 코드-2진 코드 변환

<그레이 코드-2진 코드 변환 진리표>

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

# 07 코드 변환기

$G_3G_2 \backslash G_1G_0$		$G_1G_0$			
		00	01	11	10
$G_3G_2$	00				
	01				
	11	1	1	1	1
	10	1	1	1	1

$$B_3 = G_3$$

$G_3G_2 \backslash G_1G_0$		$G_1G_0$			
		00	01	11	10
$G_3G_2$	00				
	01	1	1	1	1
	11				
	10	1	1	1	1

$$B_2 = \bar{G}_3G_2 + G_3\bar{G}_2 = G_3 \oplus G_2$$

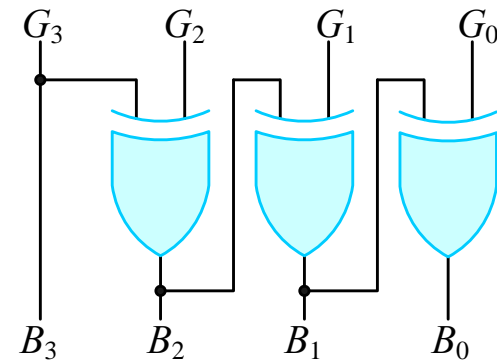
$G_3G_2 \backslash G_1G_0$		$G_1G_0$			
		00	01	11	10
$G_3G_2$	00			1	1
	01	1	1		
	11			1	1
	10	1	1		

$$B_1 = G_3 \oplus G_2 \oplus G_1 = B_2 \oplus G_1$$

$G_3G_2 \backslash G_1G_0$		$G_1G_0$			
		00	01	11	10
$G_3G_2$	00		1		1
	01	1		1	
	11		1		1
	10	1		1	

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0 = B_1 \oplus G_0$$

<카르노 맵>



<회로도>



## 07 코드 변환기

### 3. BCD 코드-3초과 코드 변환

- BCD는 10개의 숫자만 가지므로 1010 이후의 6개의 코드는 BCD에 존재하지 않는 코드이며, 입력으로서 사용될 수 없기 때문에 무관항으로 처리한다.

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

<진리표>

# 07 코드 변환기

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00				
01		1	1	1
11	x	x	x	x
10	1	1	x	x

$$E_3 = B_3 + B_2B_1 + B_2B_0$$

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00		1	1	1
01	1			
11	x	x	x	x
10		1	x	x

$$E_2 = \bar{B}_2B_1 + \bar{B}_2B_0 + B_2\bar{B}_1\bar{B}_0$$

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1		1	
01	1		1	
11	x	x	x	x
10	1		x	x

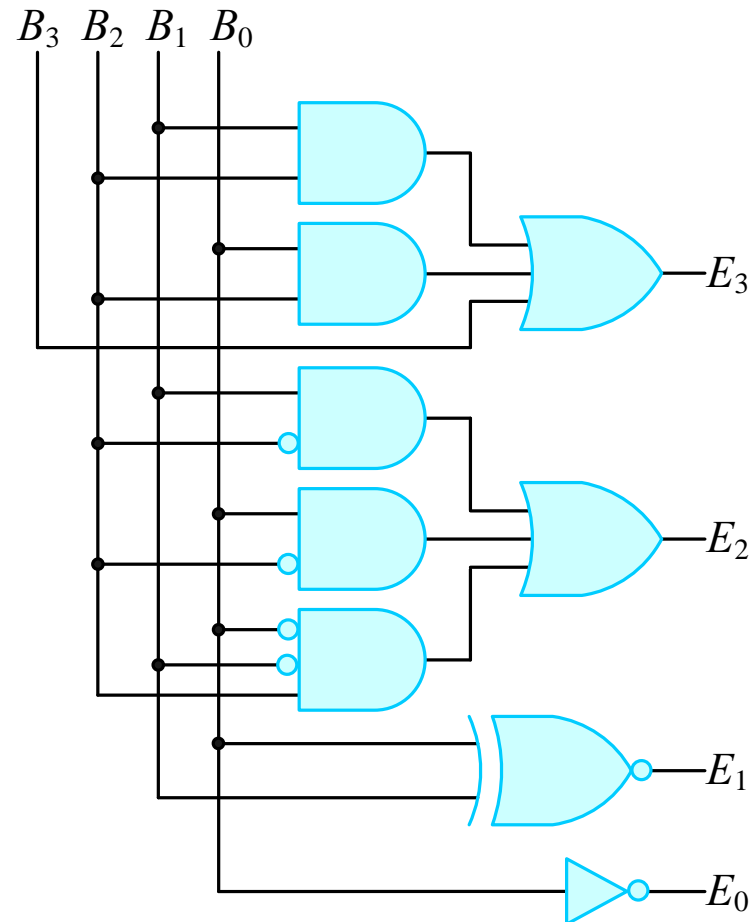
<카르노 맵>

$$E_1 = \bar{B}_1\bar{B}_0 + B_1B_0 = \bar{B}_1 \oplus B_0$$

$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1			1
01	1			1
11	x	x	x	x
10	1		x	x

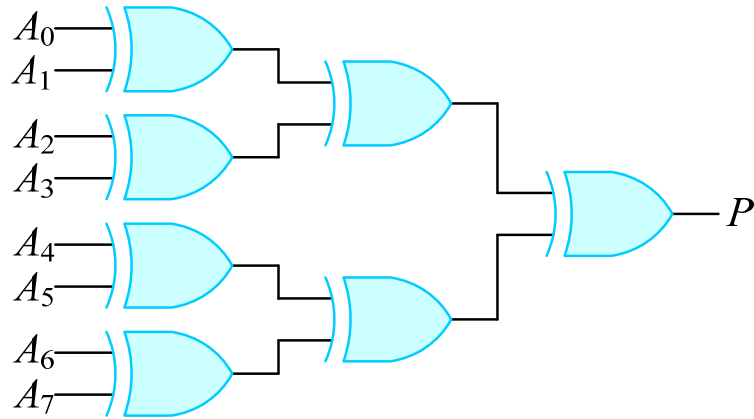
$$E_0 = \bar{B}_0$$

## 07 코드 변환기

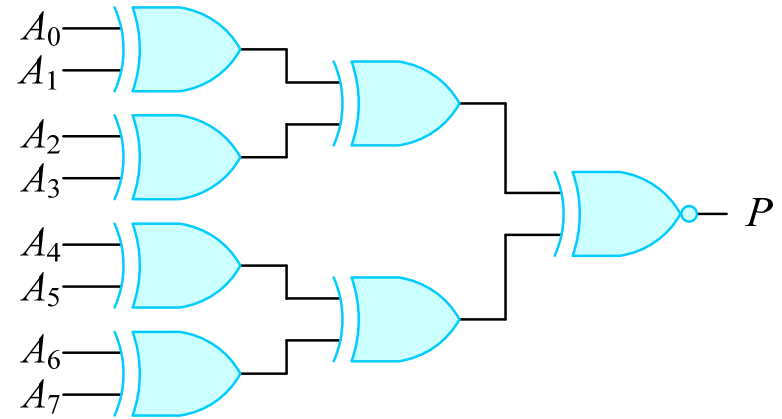


<회로도>

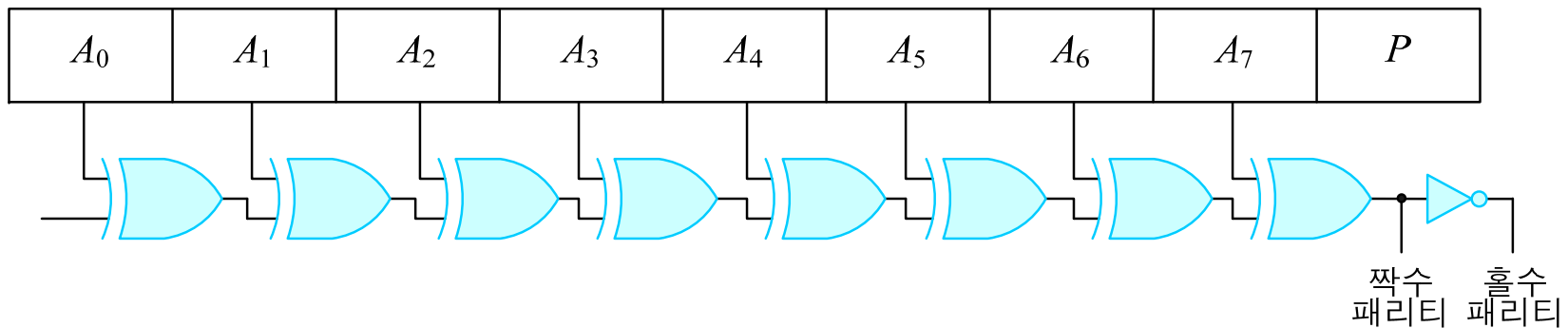
## 08 패리티 발생기/검출기



<짝수 패리티 발생회로>



<홀수 패리티 발생회로>

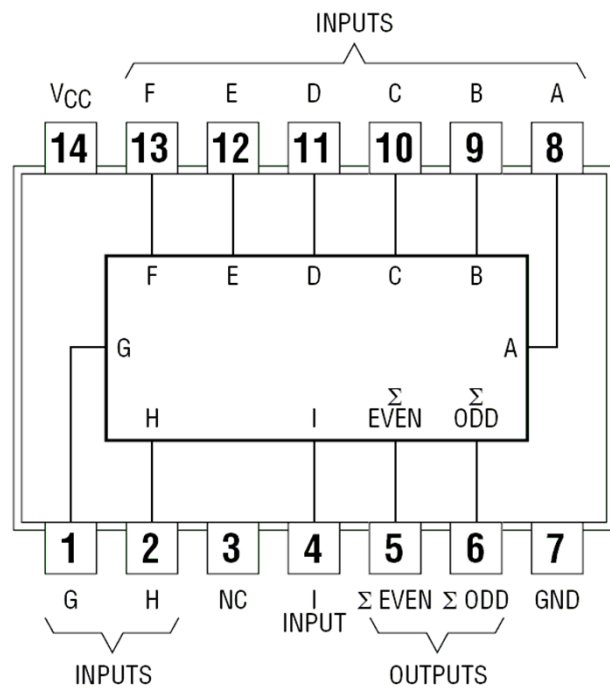


<8비트 직렬회로에서의 짝수/홀수 패리티 발생>

## 08 패리티 발생기/검출기

### ■ IC 74280

- 9비트 홀수/짝수 패리티 발생과 검출



<핀 배치도>



## 학습목표 및 목차

- 반가산기, 전가산기, 고속가산기 및 비교기 회로를 설계할 수 있다.
- 디코더와 인코더 동작 원리를 이해하고 응용회로를 설계할 수 있다.
- 멀티플렉서와 디멀티플렉서의 동작 원리를 이해하고 응용회로를 설계할 수 있다.
- 각종 코드를 변환하는 회로를 설계할 수 있다.
- 패리티 발생기와 검출기의 동작 원리를 이해하고 응용회로를 설계할 수 있다.

01. 가산기

02. 비교기

03. 디코더

04. 인코더

05. 멀티플렉서

06. 디멀티플렉서

07. 코드 변환기

08. 패리티 발생기/검출기



감사합니다 😊