이론, 실습, 시뮬레이션

디지털 논리회로

개정 3판

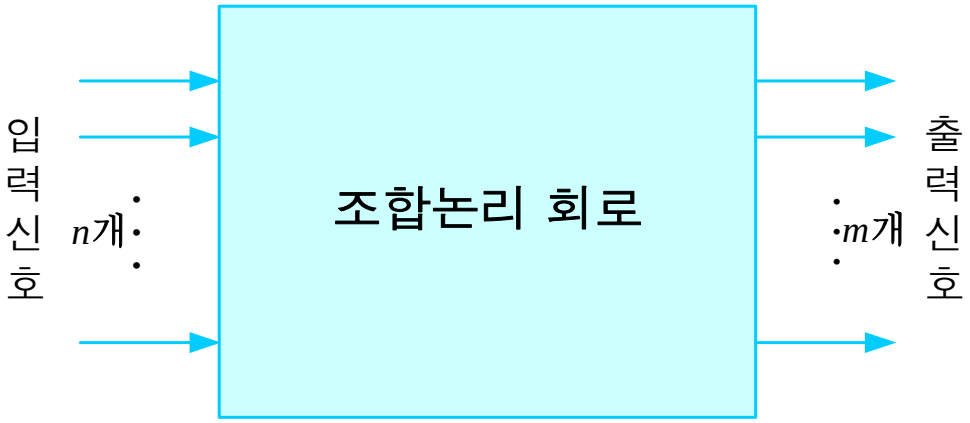
Chapter 07. 조합논리회로

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.

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)

A	0	0	1	1
$+\ B$	$+\ 0$	$+\ 1$	$+\ 0$	$+\ 1$
$C\ \ S$	$0\ \ 0$	$0\ \ 1$	$0\ \ 1$	$1\ \ 0$

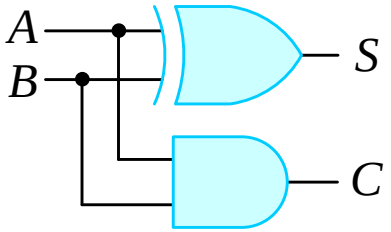
S : sum
C : carry

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

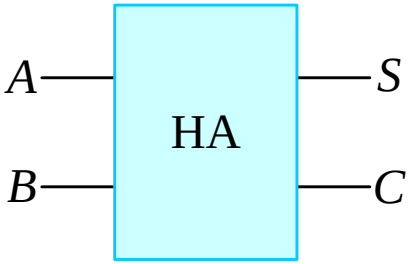
$$S = \overline{A}B + A\overline{B} = A \oplus B$$

$$C = A \cdot B$$

<진리표와 논리식>



<논리회로>



<논리기호>

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
$+ \quad B$	$+ \ 0$	$+ \ 0$	$+ \ 0$	$+ \ 0$	$+ \ 1$	$+ \ 1$	$+ \ 1$	$+ \ 1$
$C_{out} \ S$	0 0	0 1	0 1	1 0	0 1	1 0	1 0	1 1

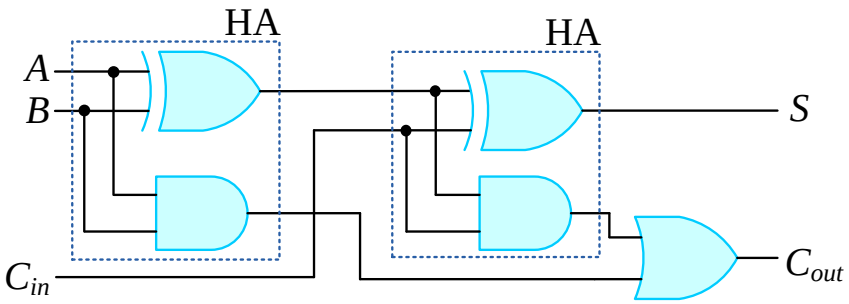
입력			출력	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

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

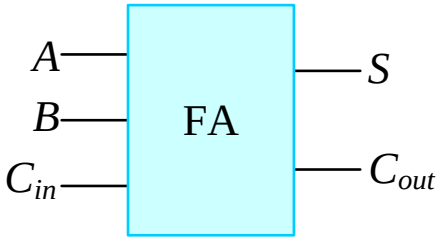
$$
\begin{aligned}
C_{out} &= \overline{A}BC_{in} + A\overline{B}C_{in} + AB\overline{C_{in}} + ABC_{in} \\
&= C_{in}(\overline{A}B + A\overline{B}) + AB(\overline{C_{in}} + C_{in}) \\
&= C_{in}(A \oplus B) + AB
\end{aligned}
$$

<진리표와 논리식>

01 가산기



<논리회로>



<논리기호>

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

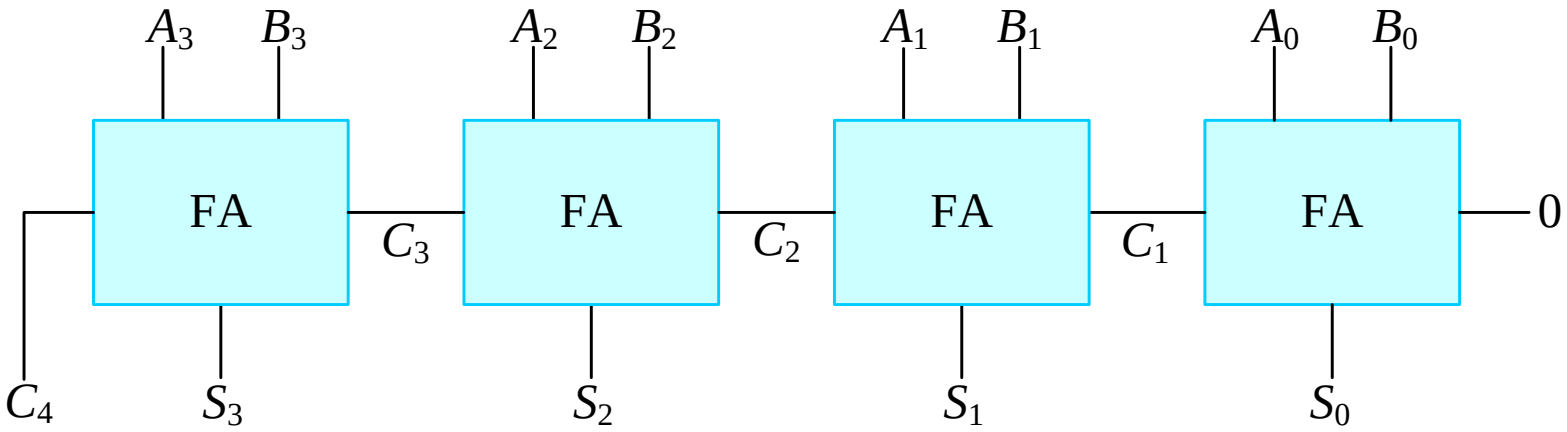
$$C_{out} = C_{in}(A \oplus B) + AB$$

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

01 가산기

3. 병렬가감산기(parallel-adder/subtracter)

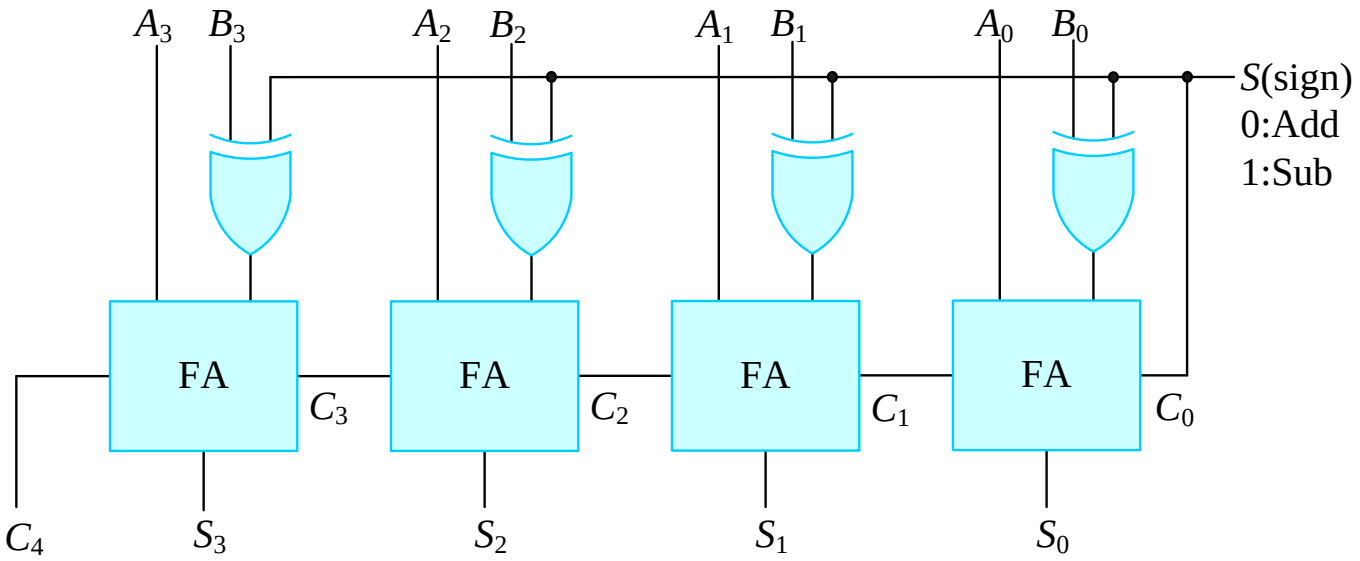
- **병렬가산기** : 전가산기 여러 개를 병렬로 연결한 회로



<전가산기를 이용한 병렬가산기>

01 가산기

- **병렬가감산기** : 병렬가산기의 B입력을 부호 S(sign)와 XOR하여 전가산기의 입력으로 사용함으로써 덧셈과 뺄셈이 모두 가능한 회로



<병렬가감산기>

01 가산기

4. 고속가산기(high-speed-adder)

- 아랫단에서 윗단으로 전달되는 자리올림수 때문에 병렬가산기는 속도가 매우 느리다는 단점이 있음
- 단점을 해결하기 위해 캐리예측가산기(carry-look-ahead-adder, CLA)를 사용
- CLA는 원리는 첫 번째는 A_i, B_i 모두가 1일 때, 또는 A_i, B_i 둘 중에 하나가 1이고 C_i 가 1일 때 캐리가 발생하므로 논리식은 다음과 같다.

$$C_{out} = C_{i+1} = A_iB_i + (A_i \oplus B_i)C_i = A_iB_i + (A_i + B_i)C_i = G_i + P_iC_i$$

where $G_i = A_iB_i$ $P_i = A_i \oplus B_i$

G: generate
P: propagate

- 4비트 가산기에서 위 식을 써보면 다음과 같다.

$$C_1 = G_0 + P_0C_0$$

$$C_2 = G_1 + P_1C_1 = G_1 + P_1G_0 + P_1P_0C_0$$

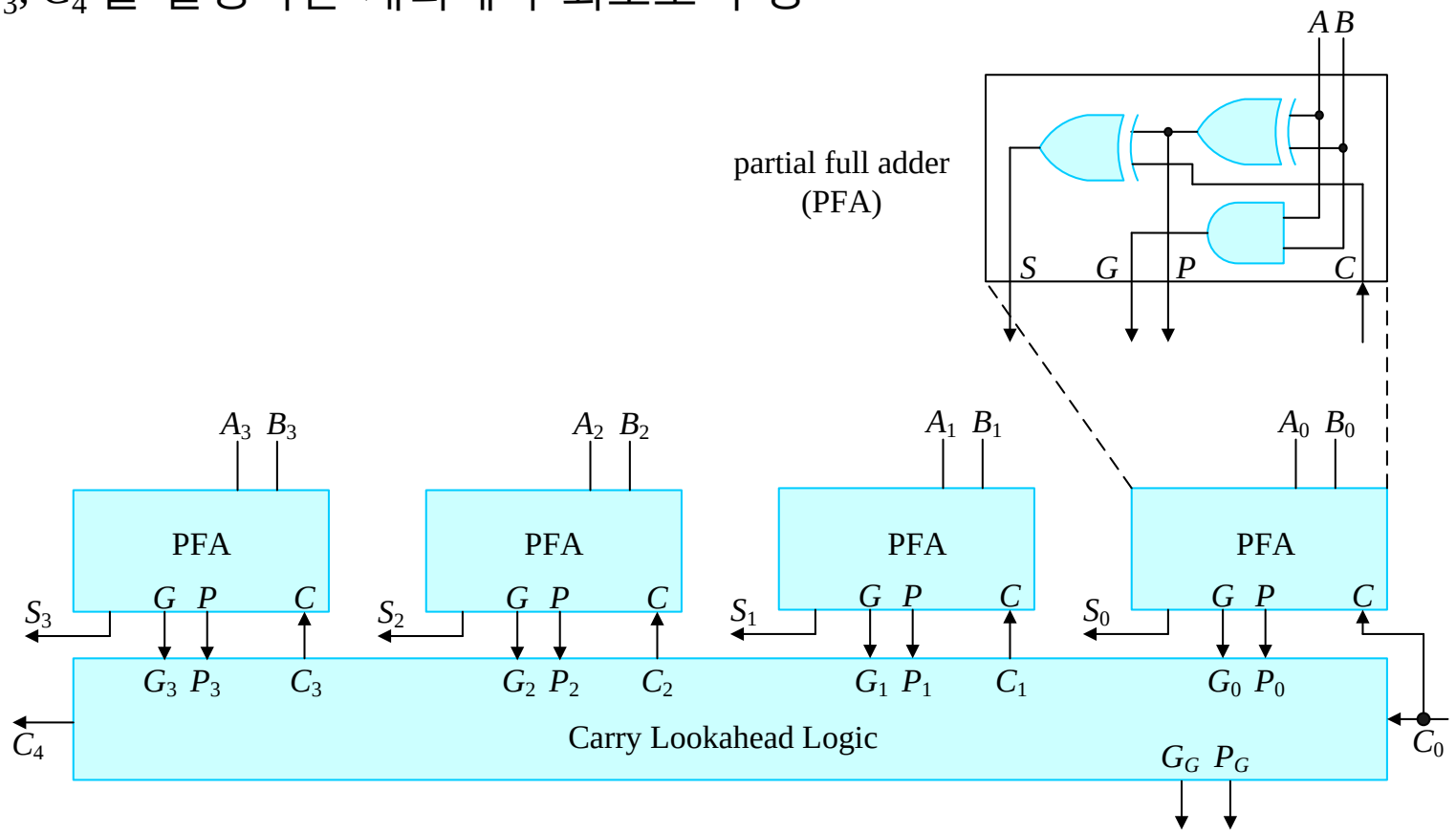
$$C_3 = G_2 + P_2C_2 = G_2 + P_2(G_1 + P_1G_0 + P_1P_0C_0) = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0$$

$$C_4 = G_3 + P_3C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

$$S_i = A_i \oplus B_i \oplus C_i = P_i \oplus C_i$$

01 가산기

- 캐리예측가산기는 S_i, P_i, G_i를 발생시키는 부분전가산기(PFA)와 위의 식 C_1, C_2, C_3, C_4 을 발생하는 캐리예측 회로로 구성



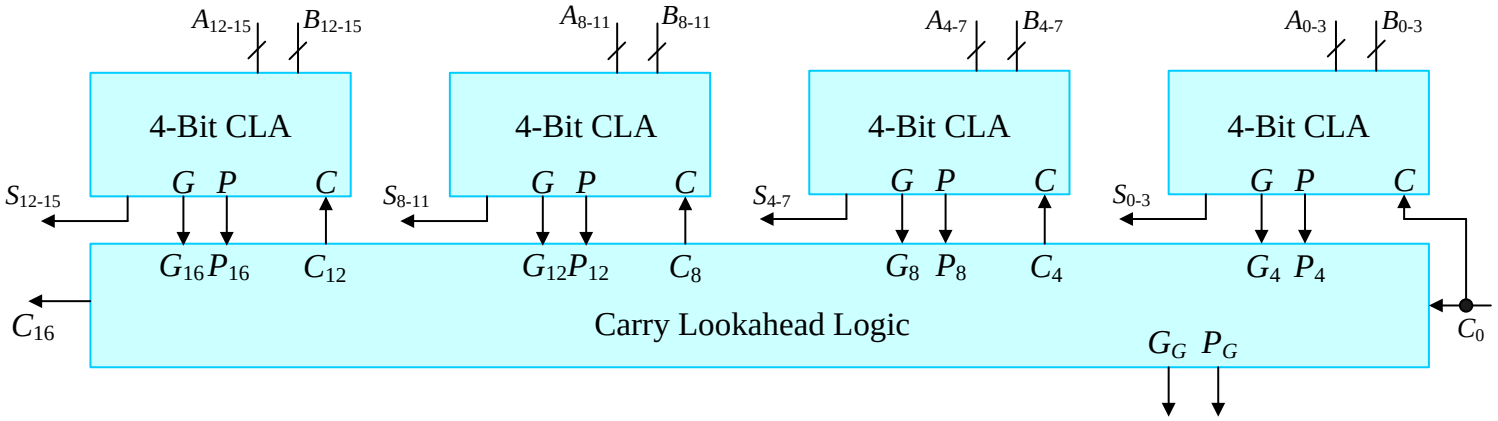
<캐리예측기를 이용한 4bit 병렬가산기>

01 가산기

- 4비트 캐리예측가산기를 하나의 모듈로 만들어서 16비트 캐리예측가산기를 만들어 사용

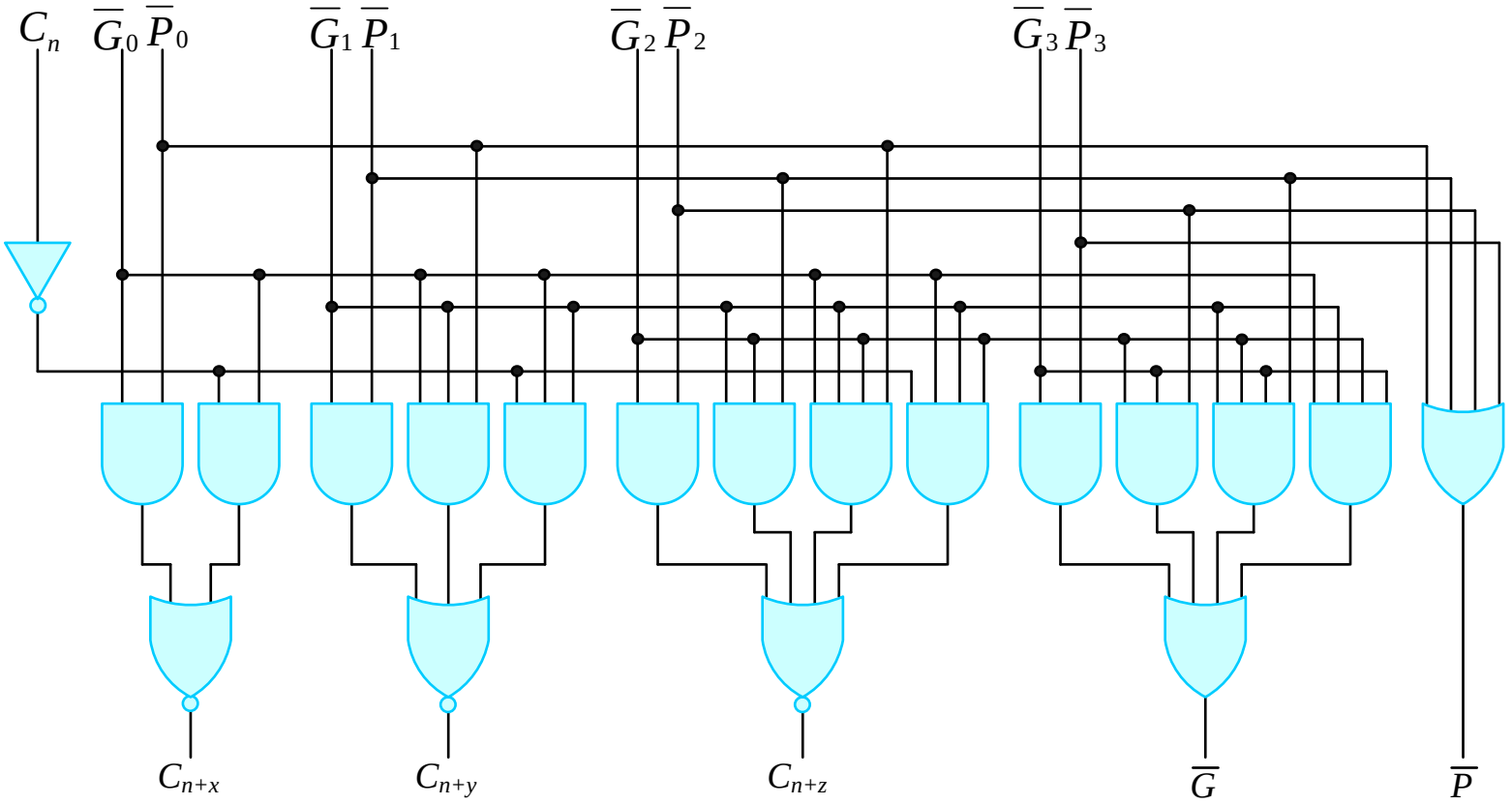
$$P_G = P_3P_2P_1P_0$$

$$G_G = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0$$



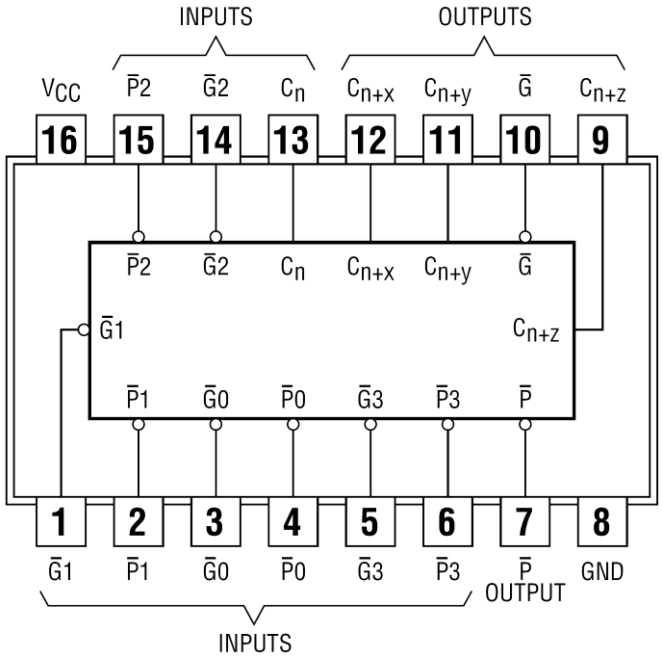
<캐리예측기를 이용한 16bit 병렬가산기>

01 가산기



<캐리예측 발생기 IC 74182의 회로>

01 가산기

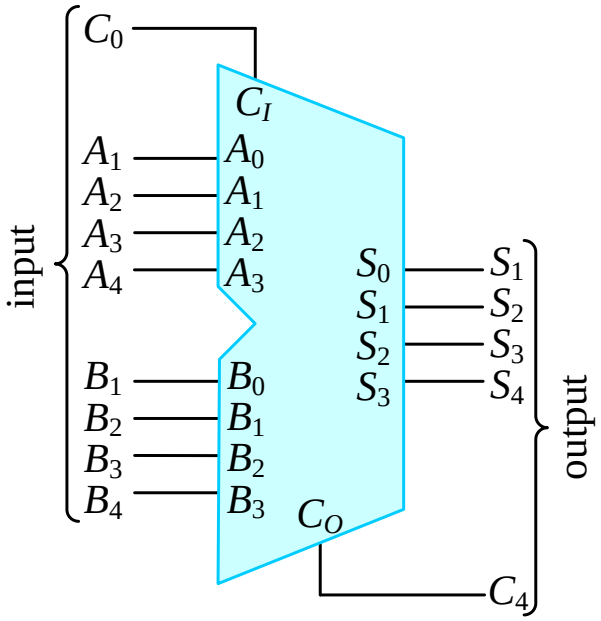


<캐리예측 발생기 IC 74182의 핀 배치도>

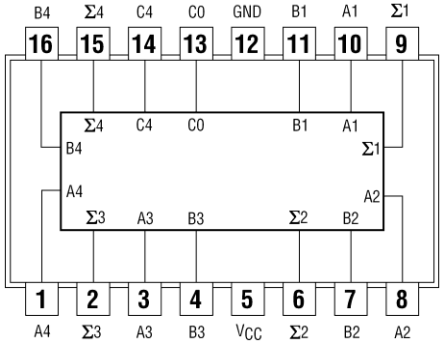
01 가산기

IC 7483/74283

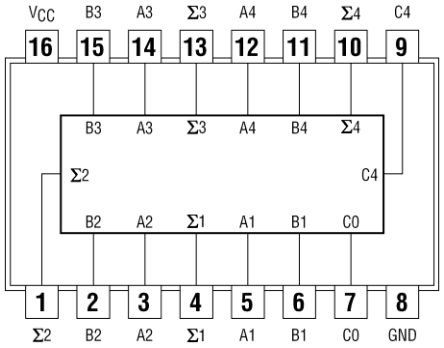
- 4비트 2진 전가산기이며, 내부에 carry look ahead 회로 내장



<개념도>



<IC 7483 핀 배치도>



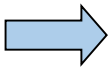
<IC 74283 핀 배치도>

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

```
  0 1 1 0
+ 0 1 1 1
---------
  1 1 0 1
```



보정 +6

```
  1 1 0 1
+ 0 1 1 0
---------
1 0 0 1 1
```

01 가산기

<BCD 덧셈표>

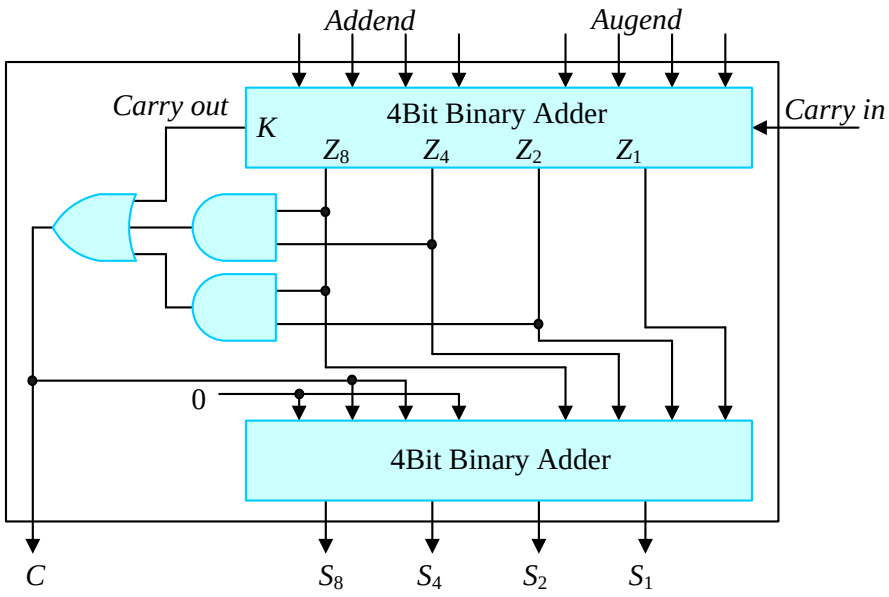
2진 합					BCD 합					10진값
K	Z_8	Z_4	Z_2	Z_1	C	S_8	S_4	S_2	S_1	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	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 \ 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

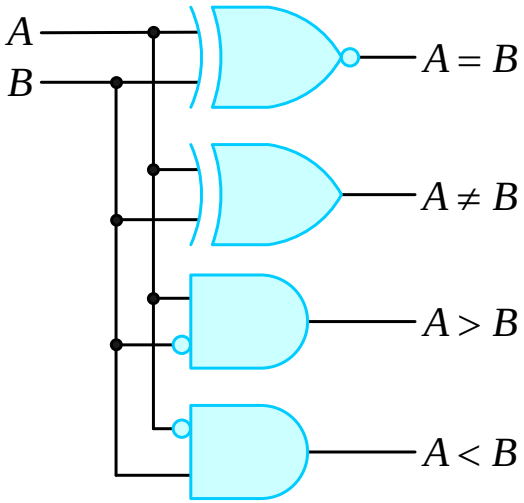
- 2진 비교기(comparator) : 두 개의 2진수의 크기를 비교하는 회로

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>

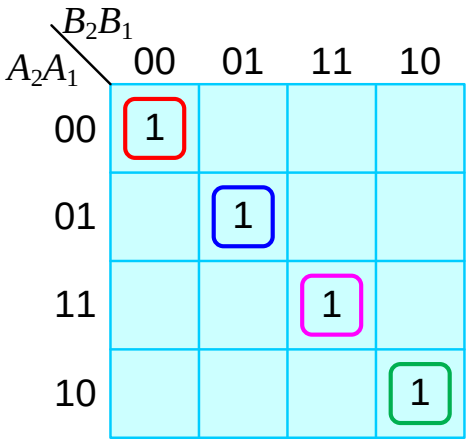
02 비교기

2비트 비교기

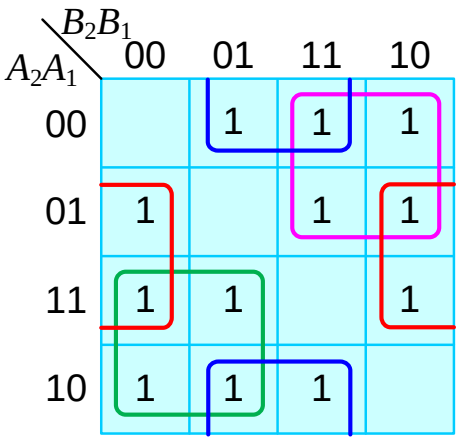
입력		출력			
A	B	$A=B$	$A\neq B$	$A>B$	$A<B$
A_1A_2	B_1B_2	F_1	F_2	F_3	F_4
0 0	0 0	1	0	0	0
	0 1	0	1	0	1
	1 0	0	1	0	1
	1 1	0	1	0	1
0 1	0 0	0	1	1	0
	0 1	1	0	0	0
	1 0	0	1	0	1
	1 1	0	1	0	1
1 0	0 0	0	1	1	0
	0 1	0	1	1	0
	1 0	1	0	0	0
	1 1	0	1	0	1
1 1	0 0	0	1	1	0
	0 1	0	1	1	0
	1 0	0	1	1	0
	1 1	1	0	0	0

<진리표>

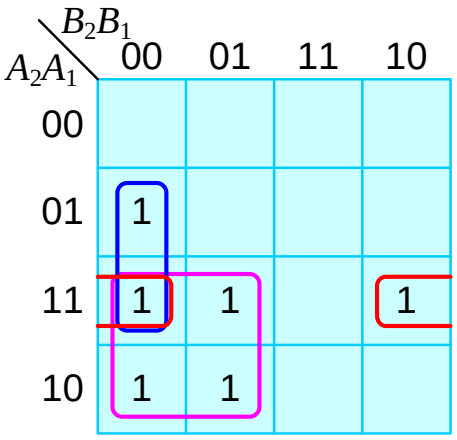
02 비교기



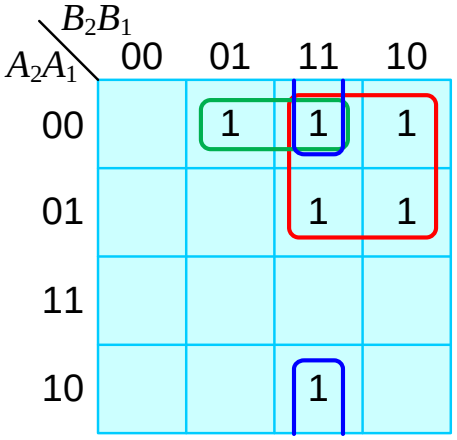
$$F_1 = (\overline{A_1 \oplus B_1})(\overline{A_2 \oplus B_2})$$



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

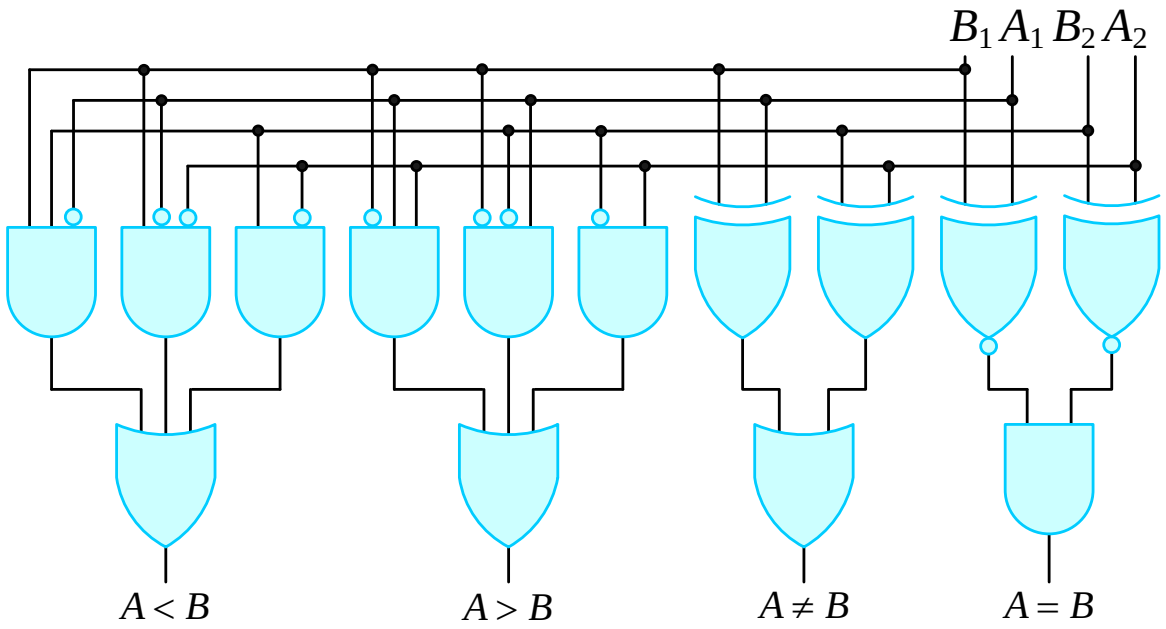


$$F_3 = A_1\overline{B_1} + A_2\overline{B_1}\,\overline{B_2} + A_1A_2\overline{B_2}$$



$$F_4 = \overline{A_1}B_1 + \overline{A_1}\,\overline{A_2}B_2 + \overline{A_2}B_1B_2$$

02 비교기

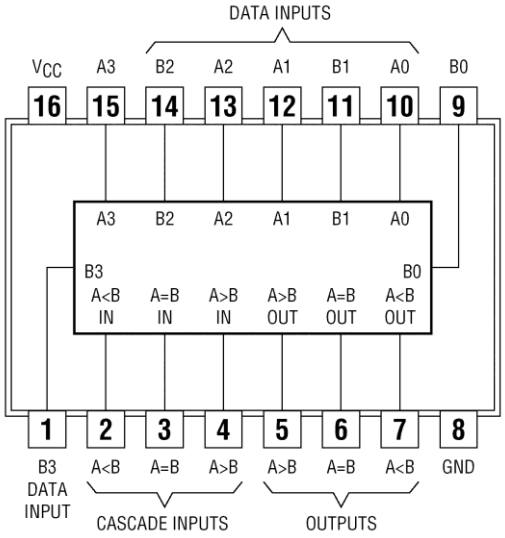


<2비트 비교기 회로>

02 비교기

IC 7485(4비트 비교기)

- 7485는 A_3~A_0와 B_3~B_0의 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.



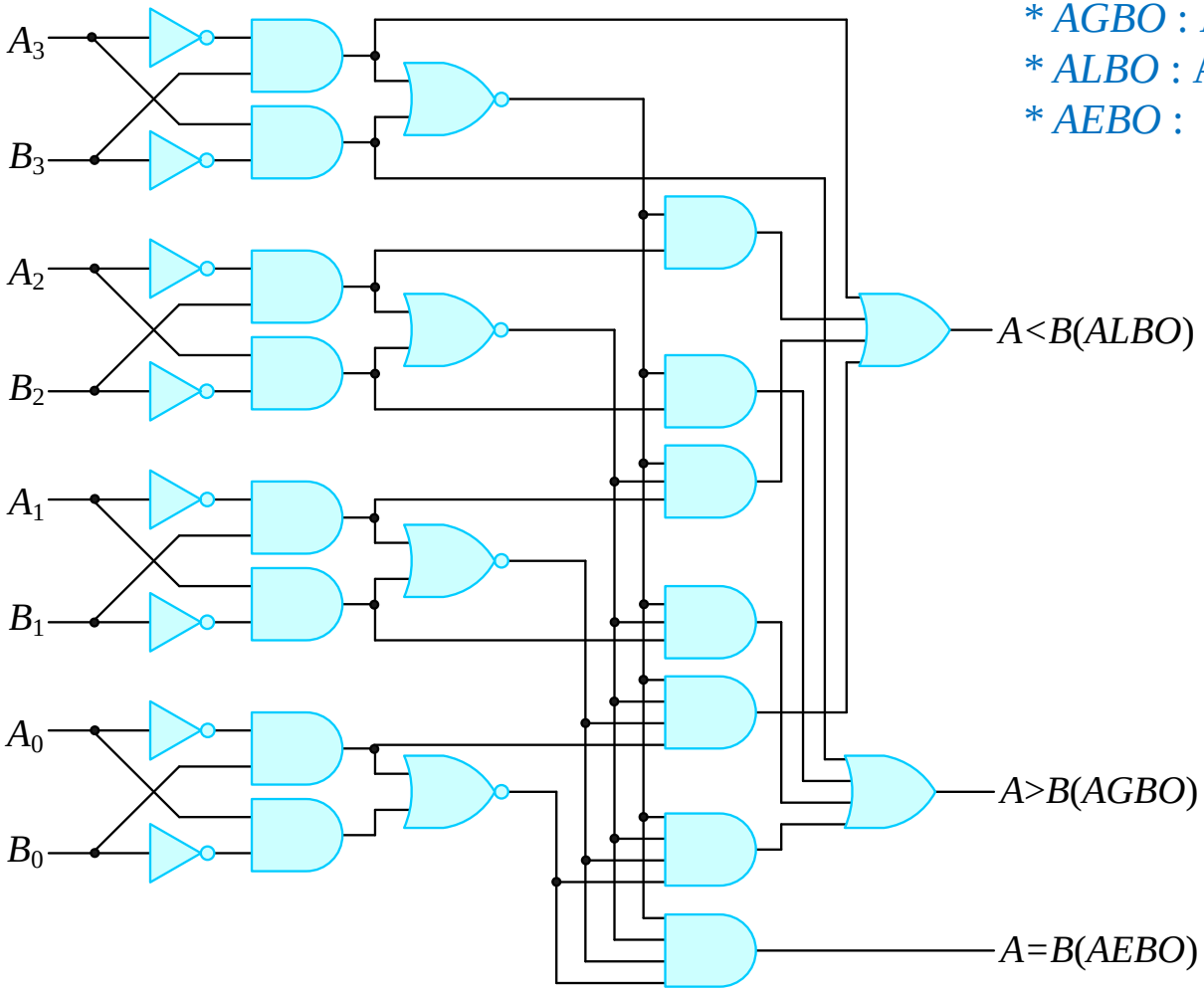
<IC 7485 핀 배치도>

02 비교기

입 력							출력		
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>

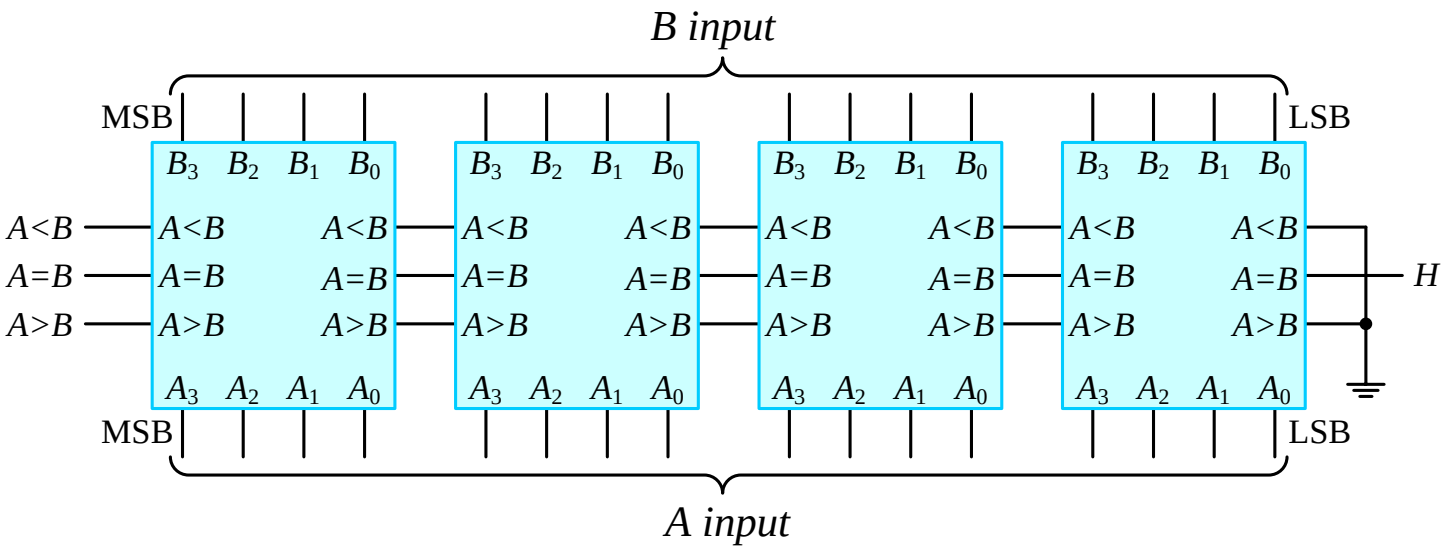
02 비교기



* *AGBO* : A Greater than B Output
* *ALBO* : A Less than B Output
* *AEBO* : A Equal B Output

<IC 7485 크기 비교기 회로>

02 비교기



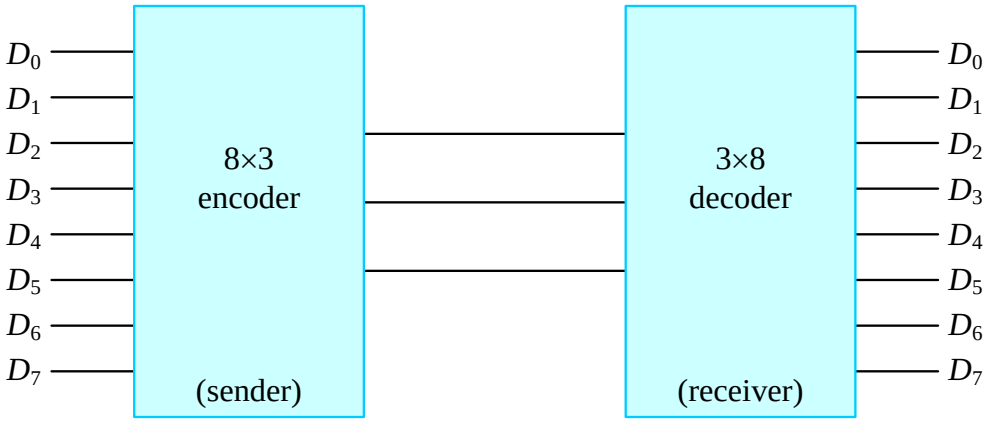
<7485 IC를 이용한 16비트 비교 회로>

03 디코더

디코더(decoder)

- 디코더 : 입력선에 나타나는 n비트의 2진 코드를 최대 2^n개의 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 인코터의 기능>

03 디코더

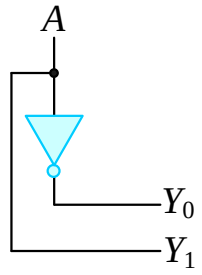
1. 1×2 디코더

- 1개의 입력에 따라서 2개의 출력 중 하나가 선택

입력	출력	
A	Y_1	Y_0
0	0	1
1	1	0

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

<진리표와 논리식>



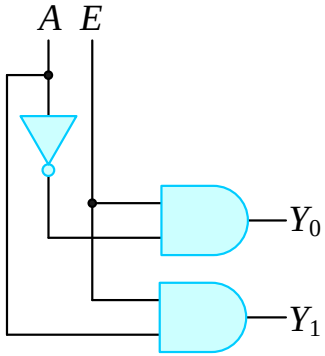
<회로도>

- 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

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

<진리표와 논리식>



<회로도>

03 디코더

2. 2×4 디코더

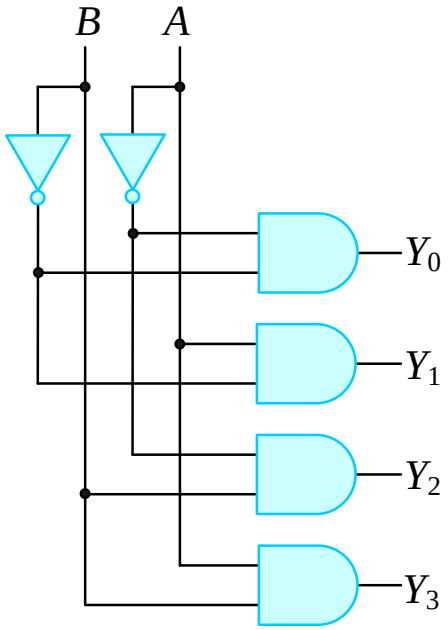
- 2개의 입력에 따라서 4개의 출력 중 하나가 선택

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>

03 디코더

2×4 NAND 디코더

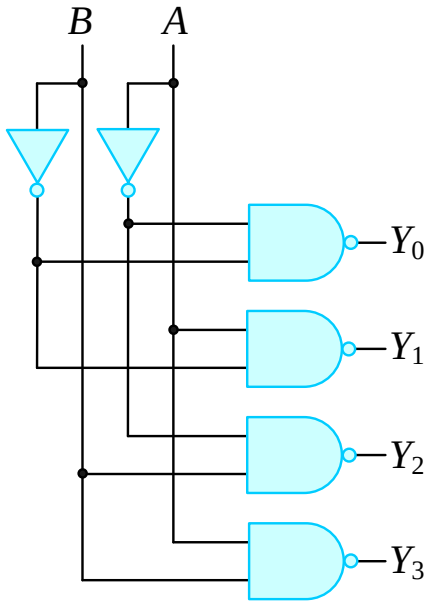
- 실제 IC들은 AND게이트가 아닌 NAND 게이트로 구성

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

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

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

<진리표와 논리식>



<회로도>

03 디코더

인에이블이 있는 2×4 디코더

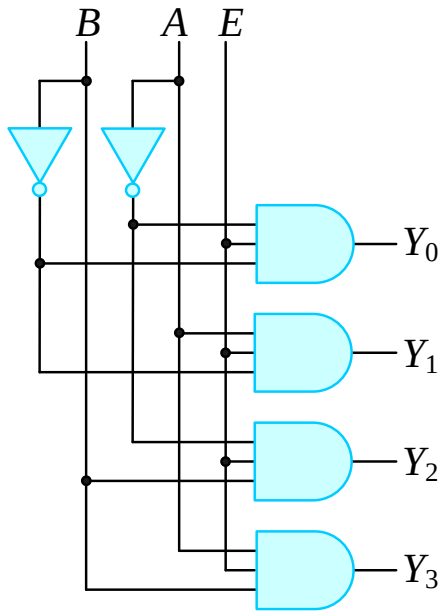
- 대부분의 IC 디코더들은 인에이블(enable) 입력이 있어서 회로를 제어한다.
- E=1일 때만 출력이 동작

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

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

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

<진리표와 논리식>



<회로도>

03 디코더

인에이블이 있는 2×4 NAND 디코더

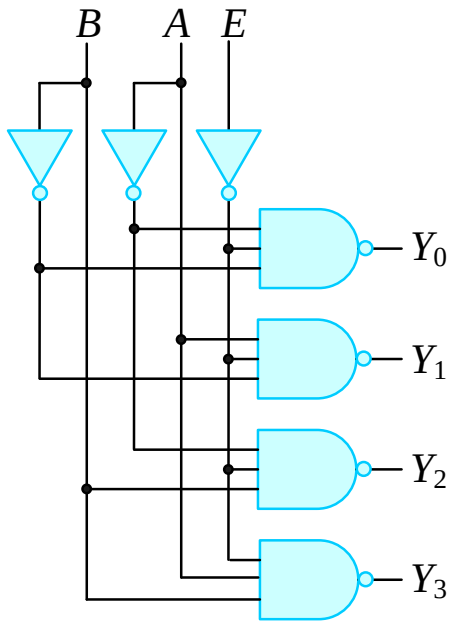
- NAND 게이트로 구성한 인에이블(enable) 입력이 있는 회로
- E=0일 때만 출력이 동작

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

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

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

<진리표와 논리식>

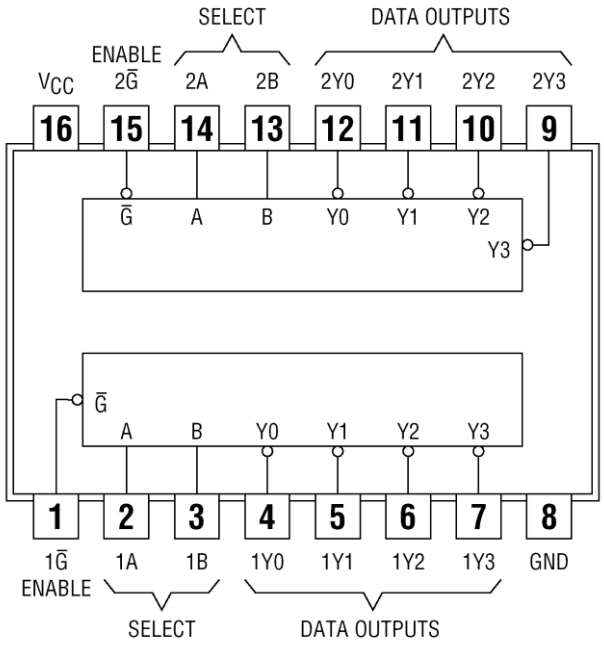


<회로도>

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

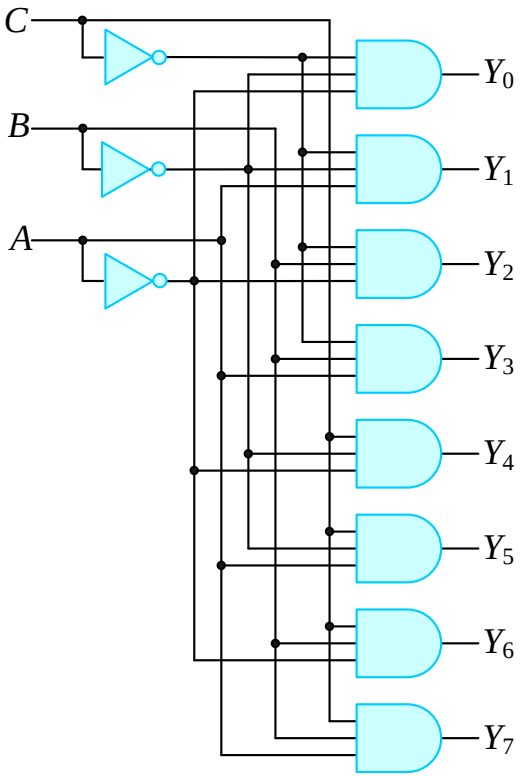
- 3개의 입력에 따라서 8개의 출력 중 하나가 선택

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)

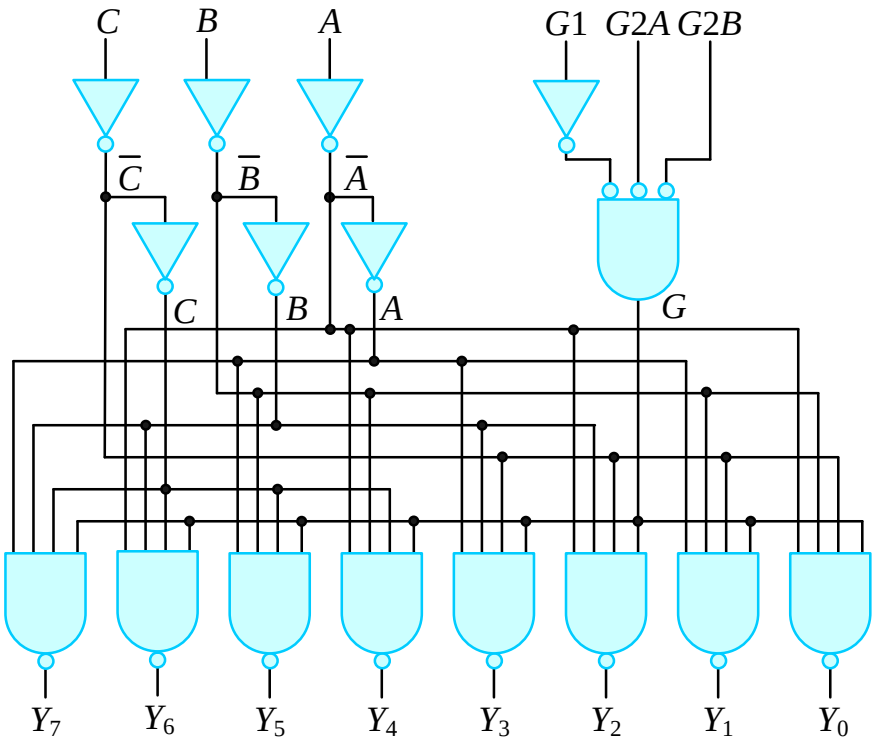
- 3개의 입력에 따라서 8개의 출력 중 하나가 선택
- 세 개의 인에이블 단자를 가지고 있음

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

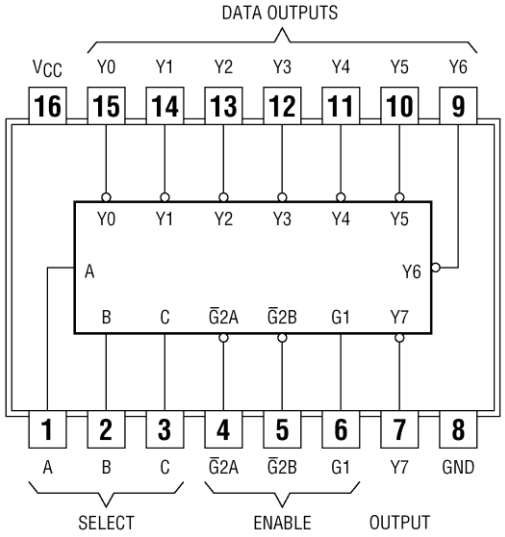
입력				출력							
C B A	G_1	G_2A	G_2B	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0 0 0	1	0	0	1	1	1	1	1	1	1	0
0 0 1	1	0	0	1	1	1	1	1	1	0	1
0 1 0	1	0	0	1	1	1	1	1	0	1	1
0 1 1	1	0	0	1	1	1	1	0	1	1	1
1 0 0	1	0	0	1	1	1	0	1	1	1	1
1 0 1	1	0	0	1	1	0	1	1	1	1	1
1 1 0	1	0	0	1	0	1	1	1	1	1	1
1 1 1	1	0	0	0	1	1	1	1	1	1	1
× × ×	0	×	×	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. 4×16 디코더

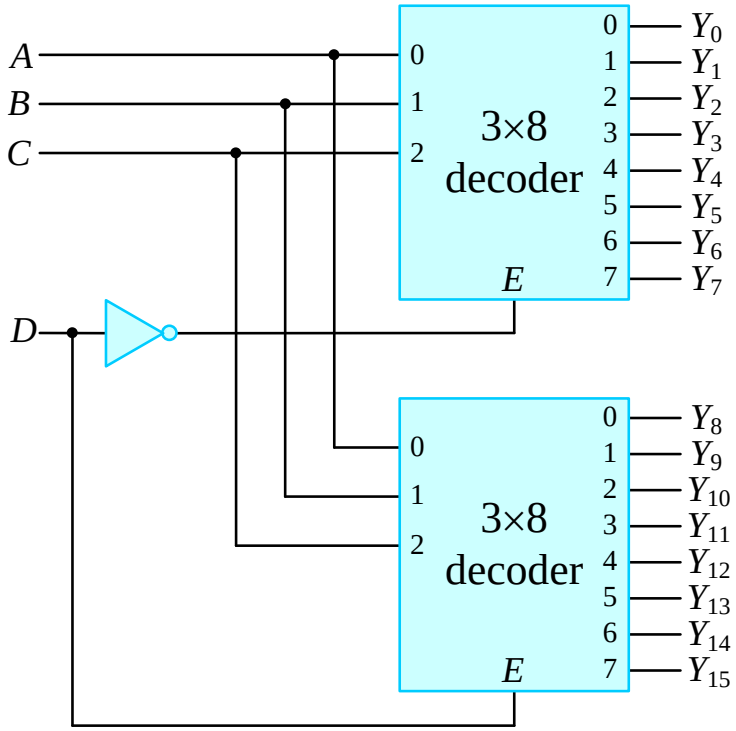
D	C	B	A	Y_{15}	Y_{14}	Y_{13}	Y_{12}	Y_{11}	Y_{10}	Y_9	Y_8	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0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

<4×16 디코더 진리표>

03 디코더

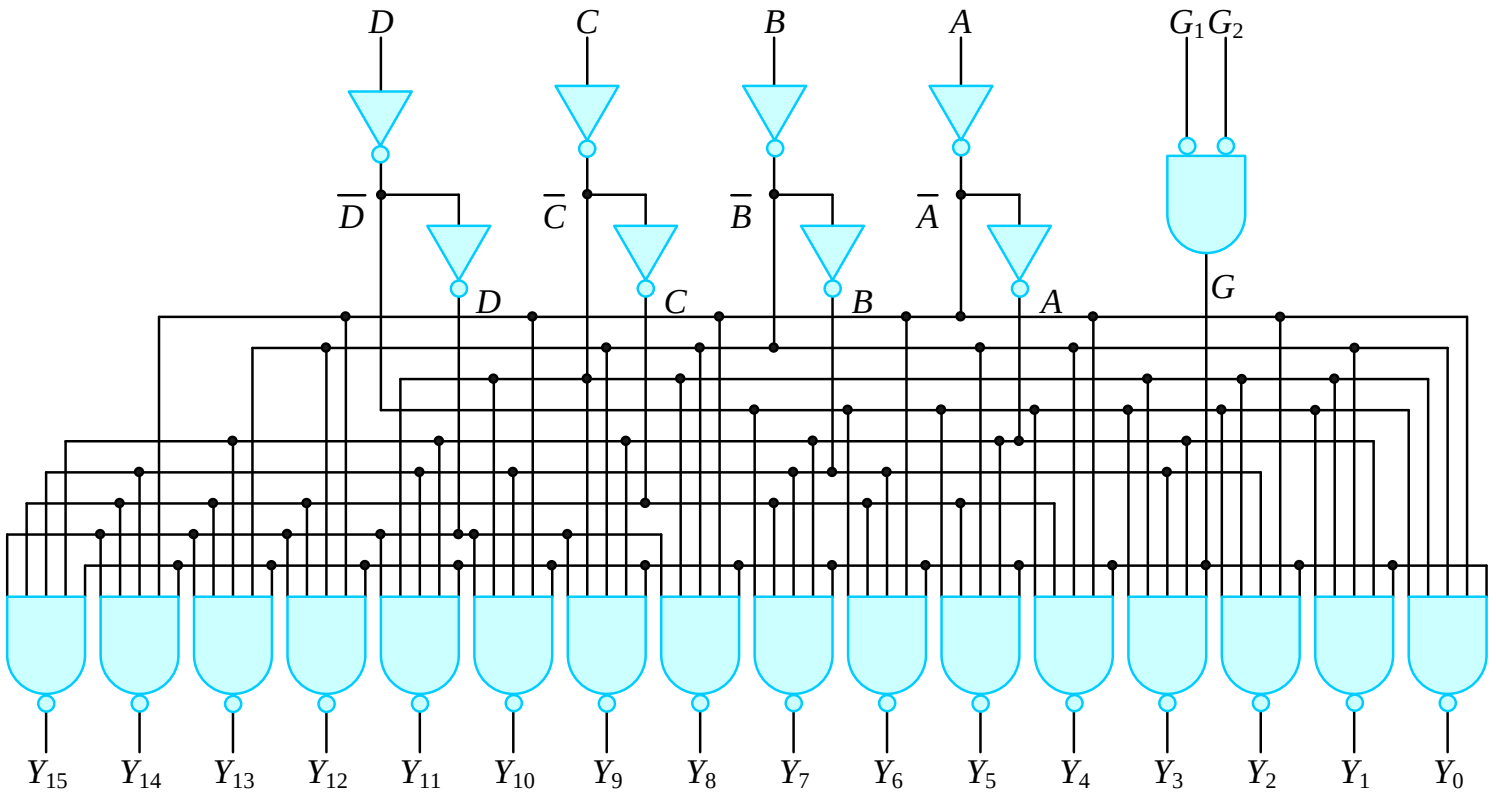
2개의 3×8 디코더로 4×16 디코더를 구성

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



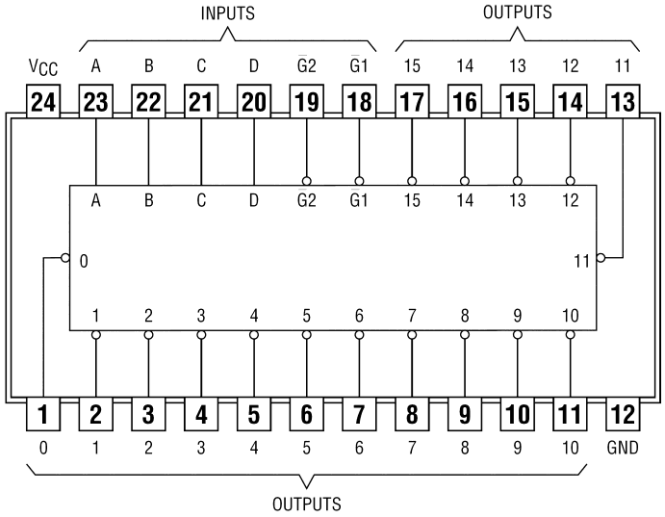
03 디코더

IC 74154



<IC 74154 회로도>

03 디코더



<IC 74154 핀 배치도>

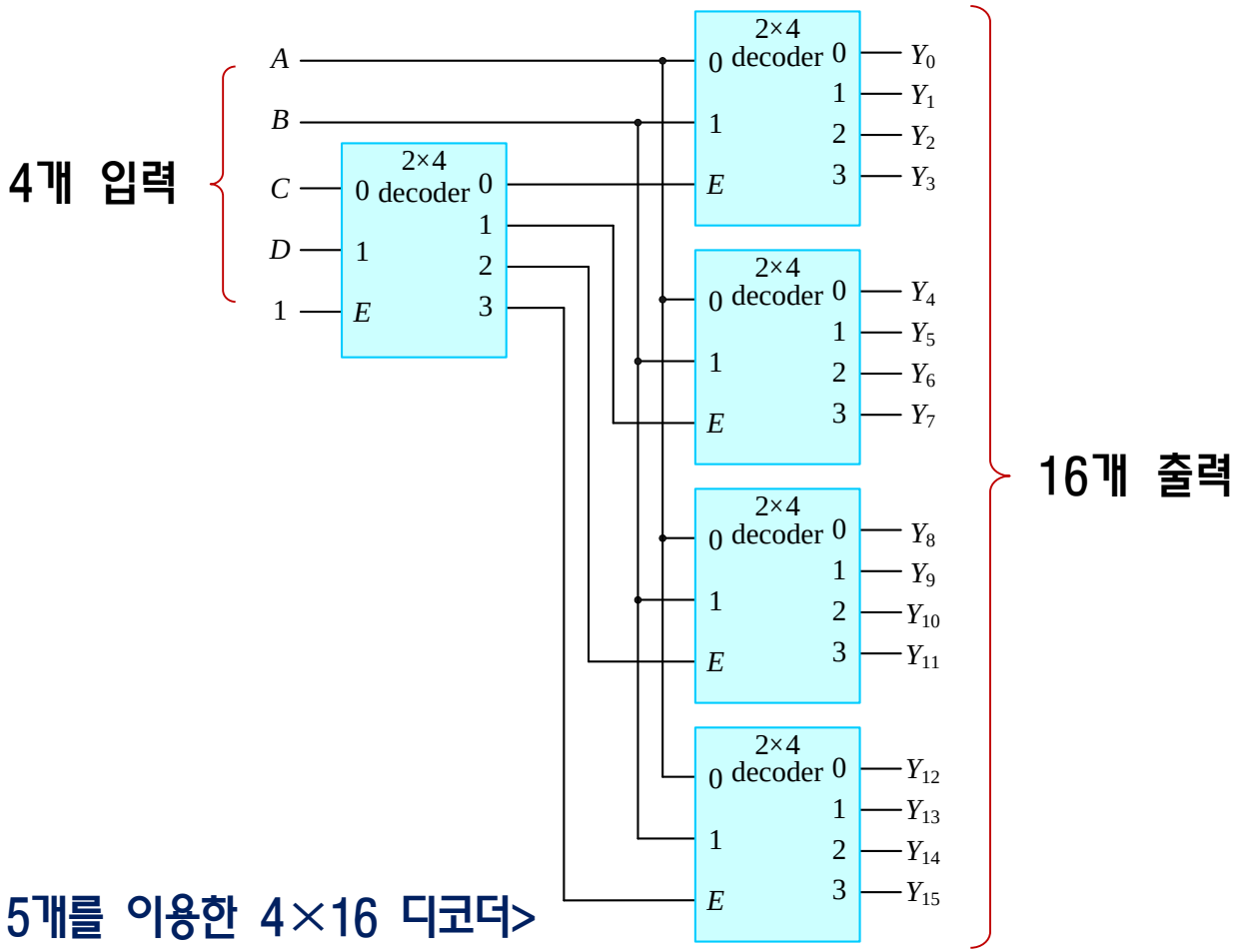
03 디코더

■ 2×4 디코더 5개를 이용한 4×16 디코더

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

<4×16 디코더 진리표>

03 디코더

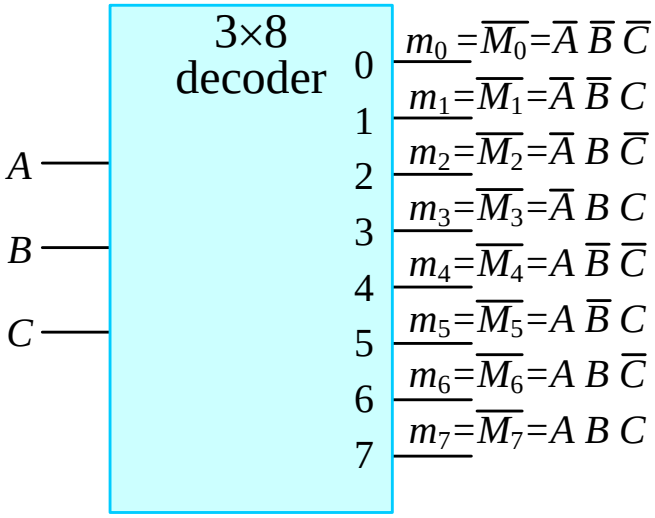


<2×4 디코더 5개를 이용한 4×16 디코더>

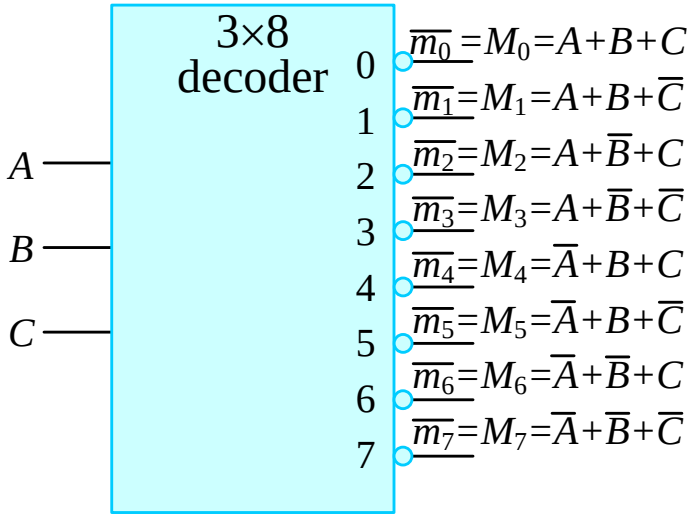
03 디코더

5. 디코더를 이용한 조합논리회로

❖ 3×8 디코더를 이용하는 경우



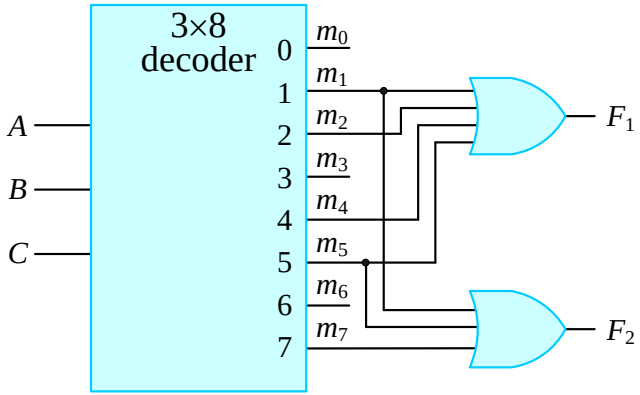
<3×8 디코더 출력>



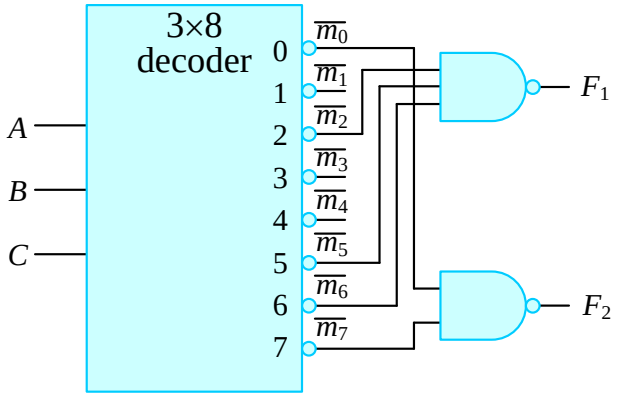
<3×8 디코더 반전출력>

03 디코더

❖ 3×8 디코더를 이용하는 경우의 예



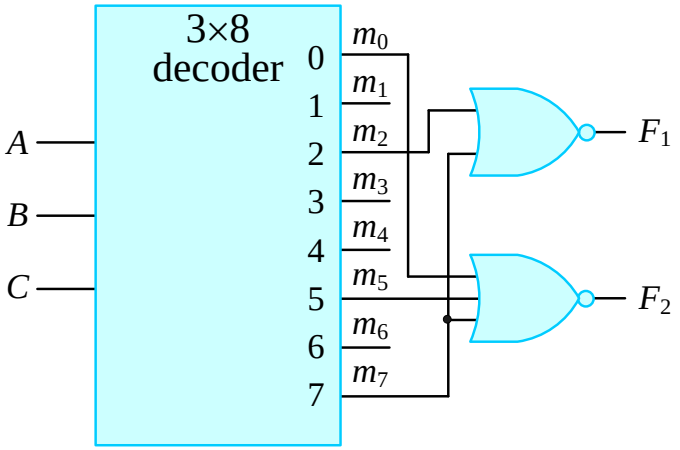
$$F_1(A,B,C) = \sum m(1, 2, 4, 5)$$
$$F_2(A,B,C) = \sum m(1, 5, 7)$$



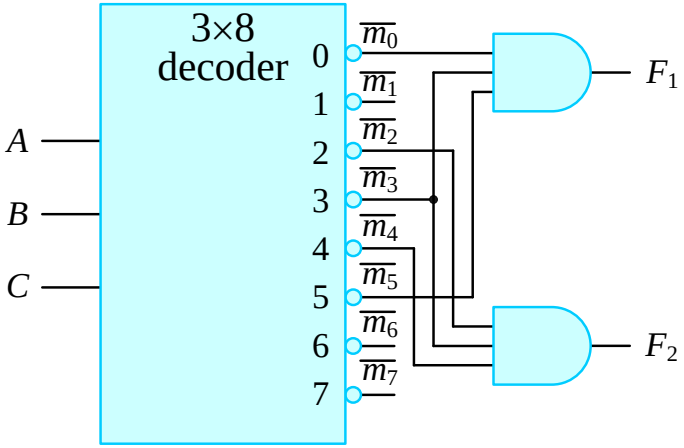
$$F_1(A,B,C) = \sum m(2, 5, 6)$$
$$F_2(A,B,C) = \sum m(0, 7)$$

03 디코더

❖ 3×8 디코더를 이용하는 경우의 예



$$F_1(A,B,C) = \prod M(2, 7)$$
$$F_2(A,B,C) = \prod M(0, 5, 7)$$

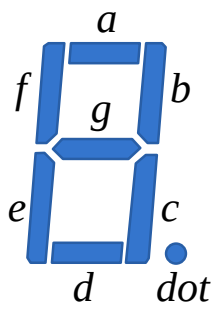


$$F_1(A,B,C) = \prod M(0, 3, 5)$$
$$F_2(A,B,C) = \prod M(2, 3, 4)$$

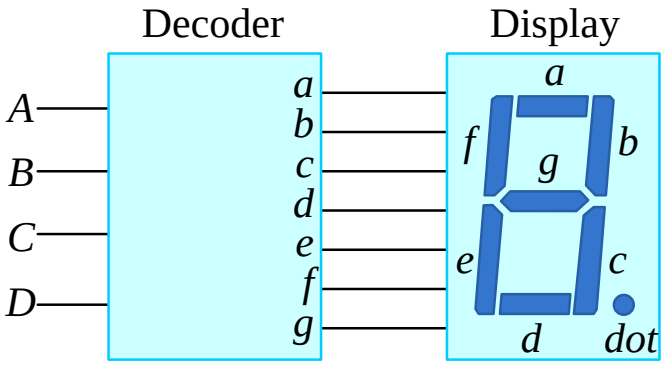
03 디코더

6. BCD-7-세그먼트 디코더

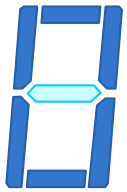
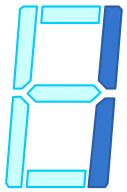
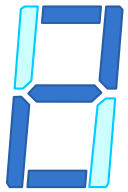
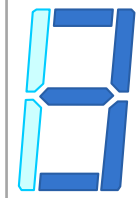
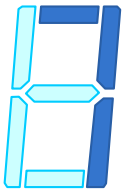
- 7 세그먼트 : 숫자 표시 전용 장치



<7-세그먼트 구성>



<7-세그먼트와 디코더의 연결>

0	1	2	3	4	5	6	7	8	9
									

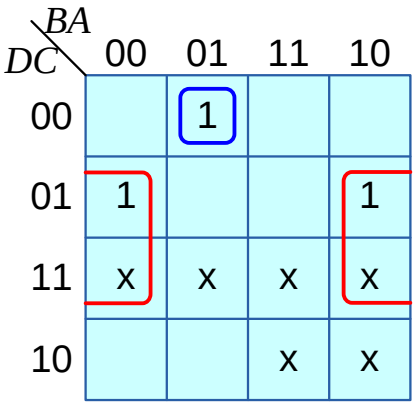
<7-세그먼트의 숫자 표시>

03 디코더

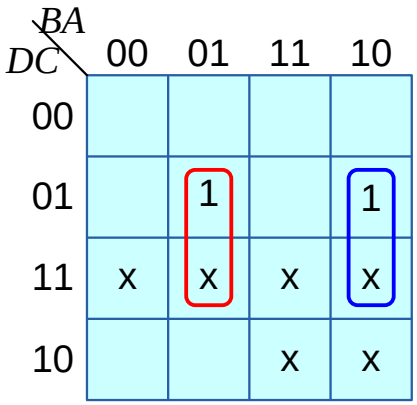
<7-세그먼트 디코더 진리표>

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

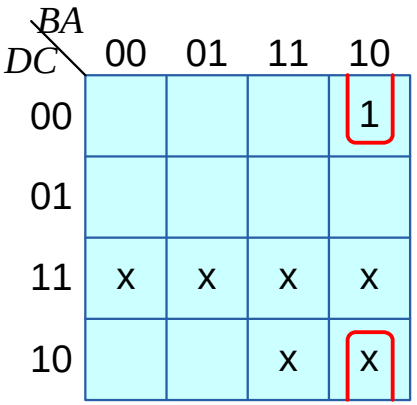
03 디코더



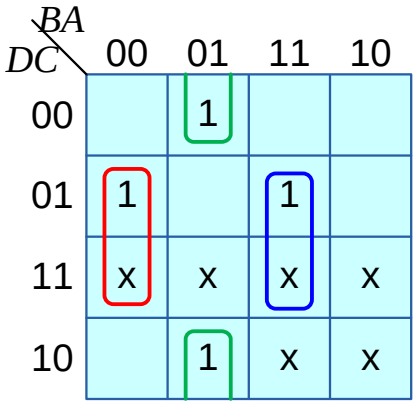
$$\overline{a} = \overline{D}\overline{C}\overline{B}A + C\overline{A}$$



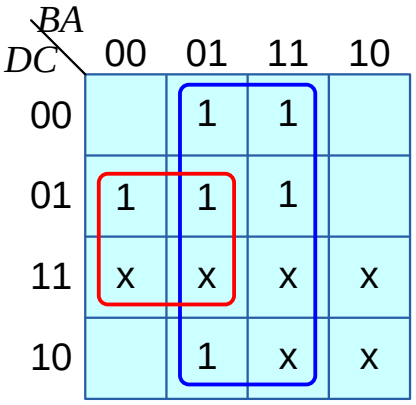
$$\overline{b} = C\overline{B}A + CB\overline{A} = C(B \oplus A)$$



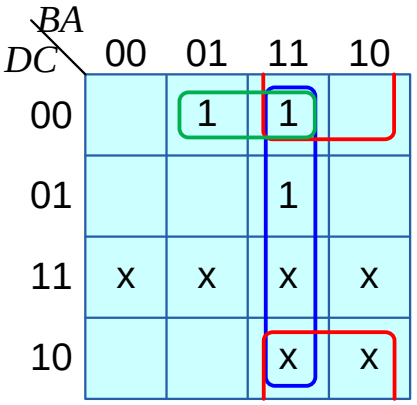
$$\overline{c} = \overline{C}B\overline{A}$$



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



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



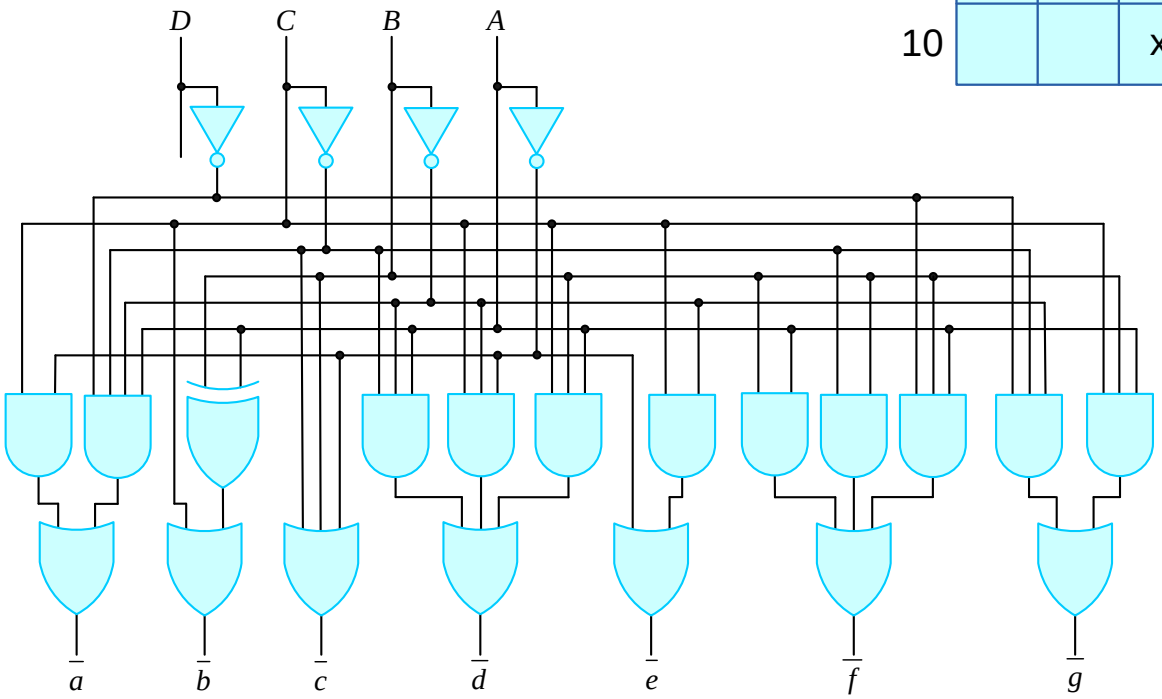
$$\overline{f} = BA + \overline{C}B + \overline{D}\overline{C}A$$

<카르노 맵>

03 디코더

DC \ BA	00	01	11	10
00	1	1		
01			1	
11	x	x	x	x
10			x	x

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



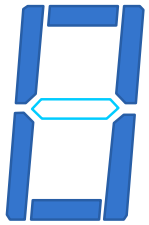
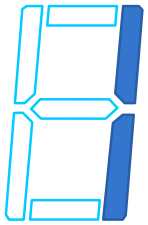
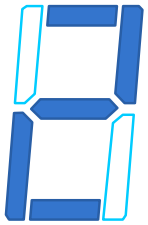
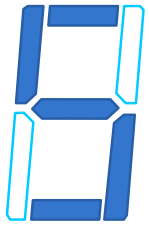
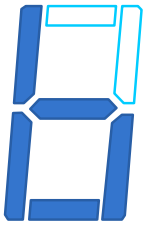
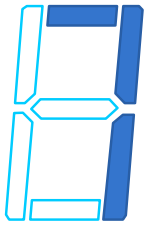
<회로도>

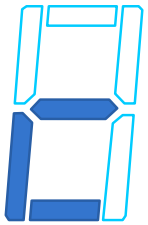
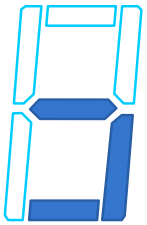
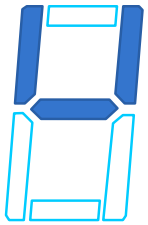
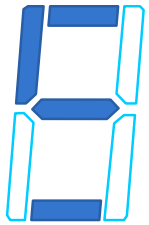
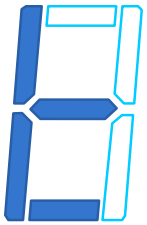
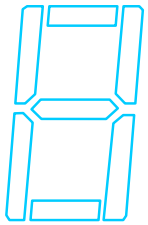
03 디코더

<7447 진리표>

10진값 또는 기능	입력							출력						
	$\overline{LT}$	$\overline{RBI}$	D	C	B	A	$BI/\overline{RBO}$	$\overline{a}$	$\overline{b}$	$\overline{c}$	$\overline{d}$	$\overline{e}$	$\overline{f}$	$\overline{g}$
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

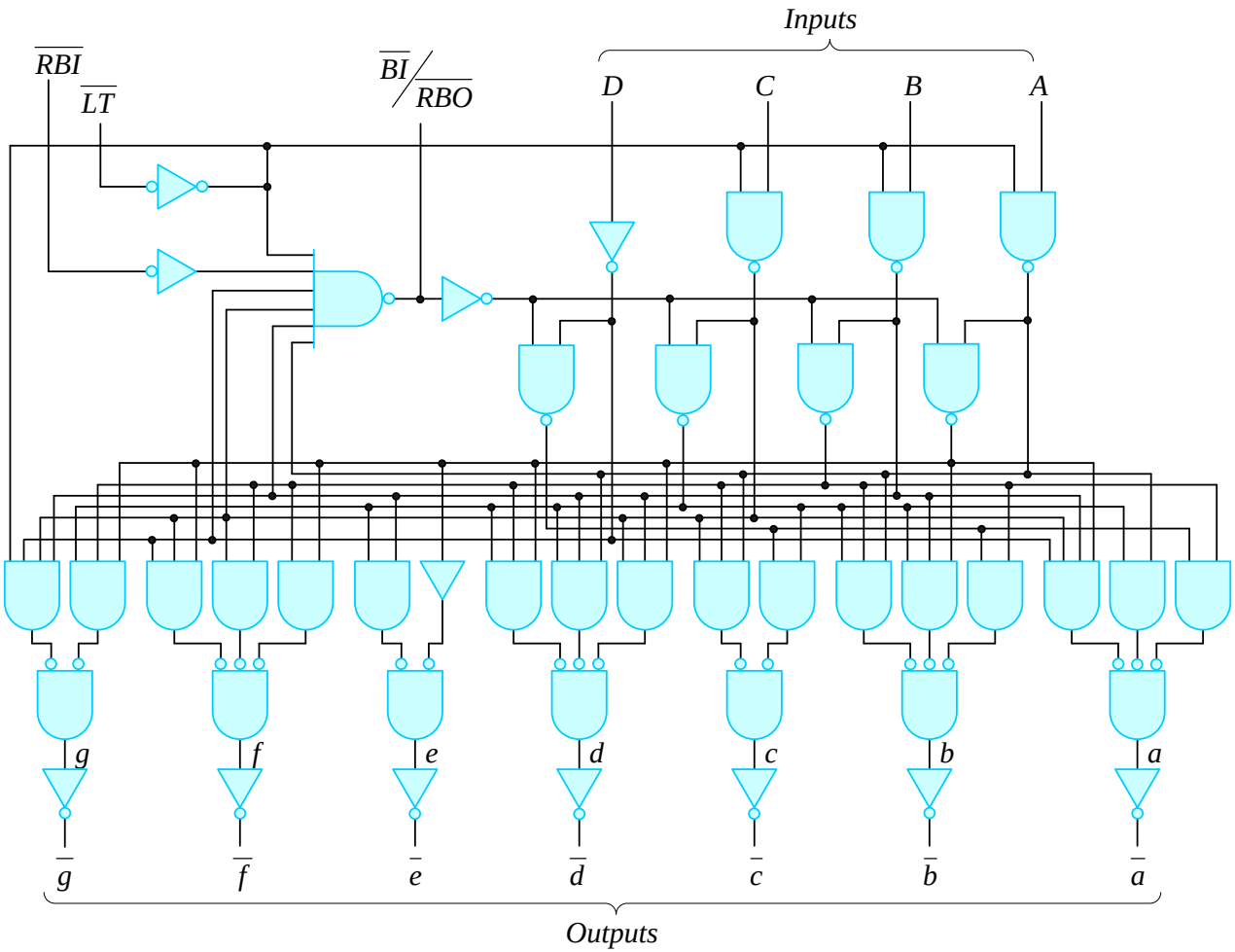
03 디코더

0	1	2	3	4	5	6	7
							

8	9	10	11	12	13	14	15
							

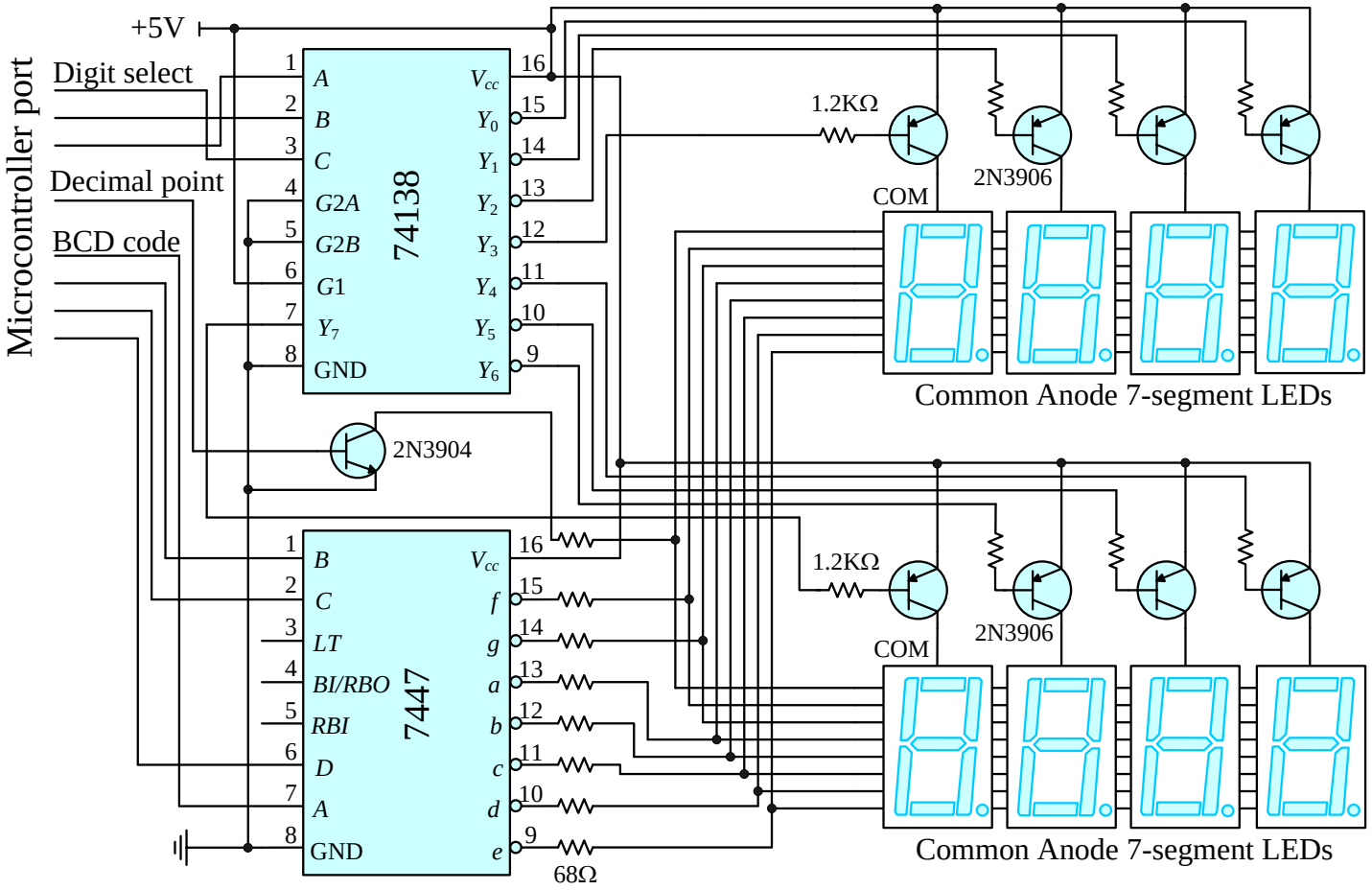
<7447의 LED 점등 패턴도>

03 디코더



<7447 회로도>

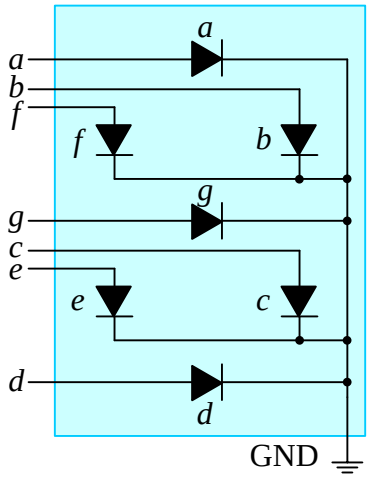
03 디코더



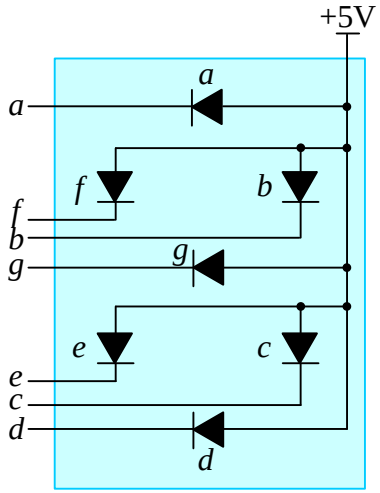
<7447을 이용한 7-세그먼트 구동 회로 예>

03 디코더

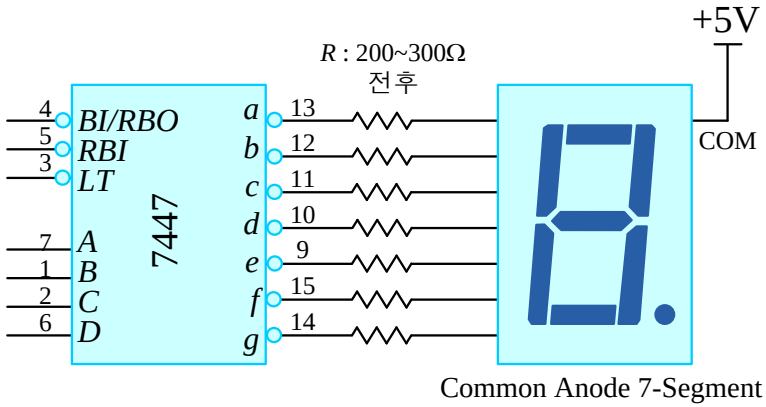
7-세그먼트 공통 회로



<캐소드 공통>



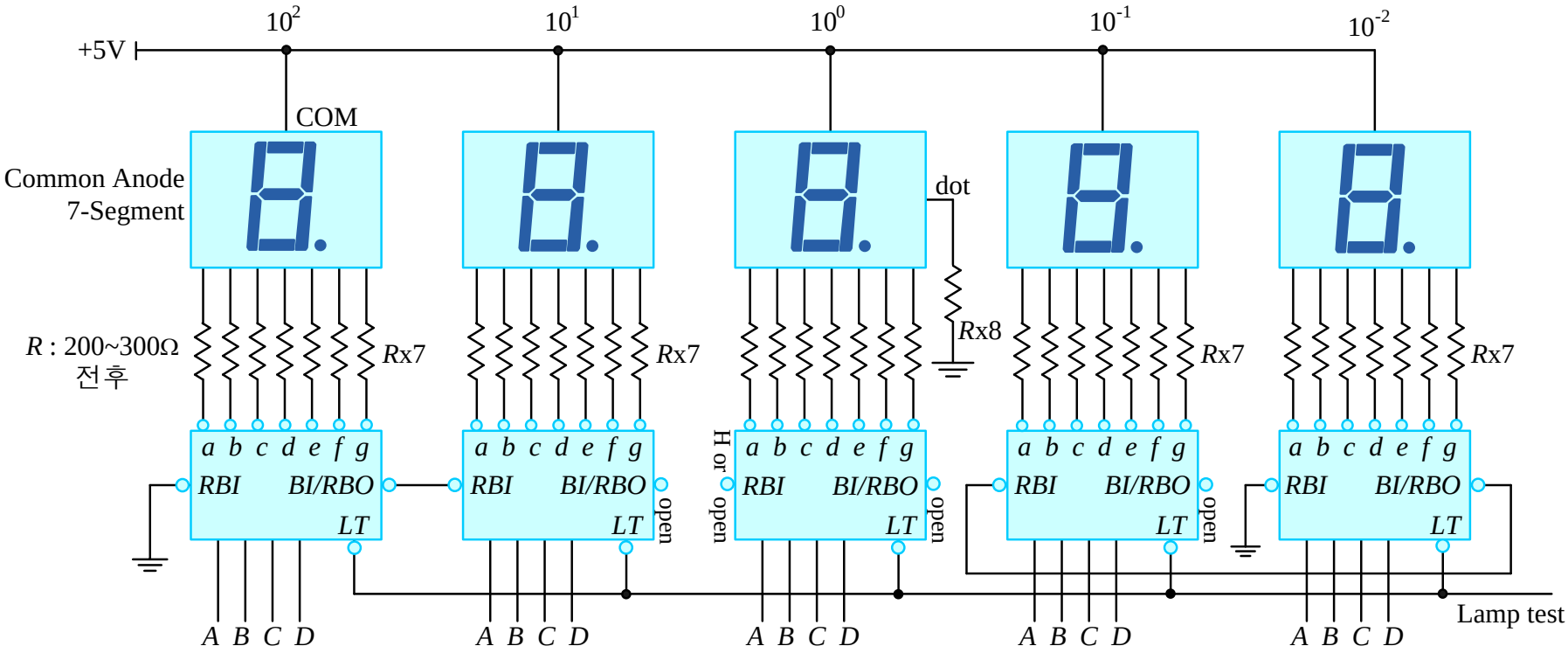
<애노드 공통>



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^n개의 입력신호로부터 n개의 출력신호를 만든다.
- 인코더의 역할은 2^n 개중 활성화된 하나의 1비트입력 신호를 받아서 그 숫자에 해당하는 n 비트 2진 정보를 출력한다.

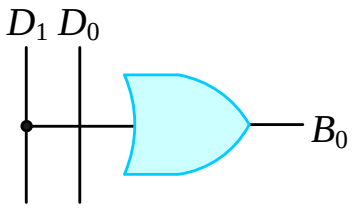
1. 2×1 인코더

- 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

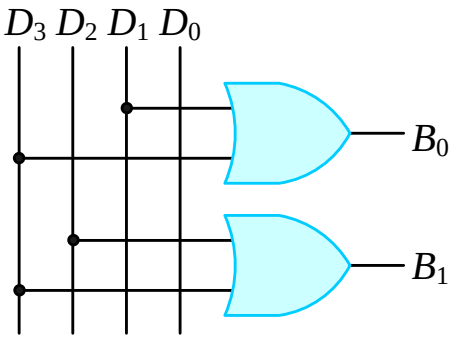
2. 4×2 인코더

- 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

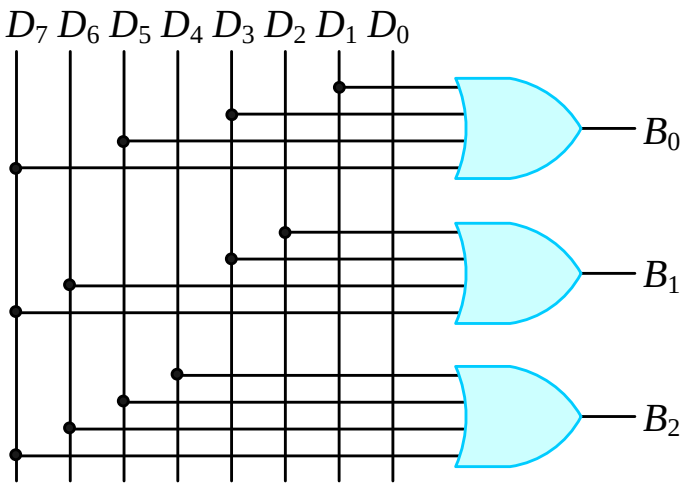
3. 8×3 인코더

- 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

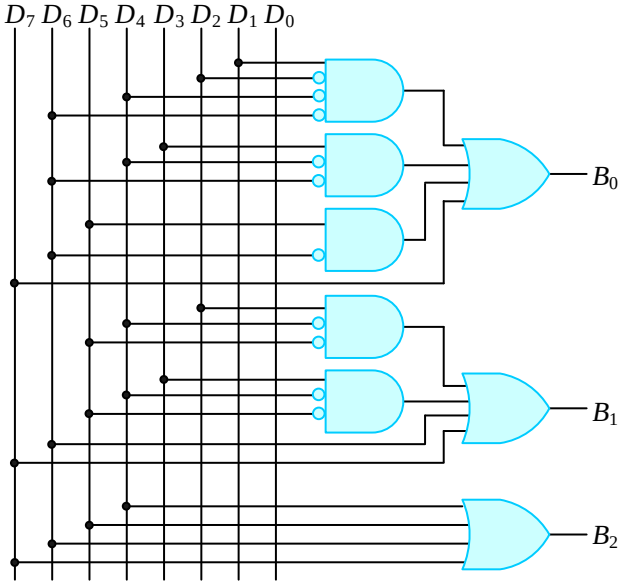
4. 8×3 우선순위 인코더

- 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

$$B_2 = D_7 + D_6 + D_5 + D_4$$
$$B_1 = D_7 + D_6 + \overline{D}_5\overline{D}_4D_3 + \overline{D}_5\overline{D}_4D_2$$
$$B_0 = D_7 + \overline{D}_6D_5 + \overline{D}_6\overline{D}_4D_3 + \overline{D}_6\overline{D}_4\overline{D}_2D_1$$

<진리표와 논리식>



<회로도>

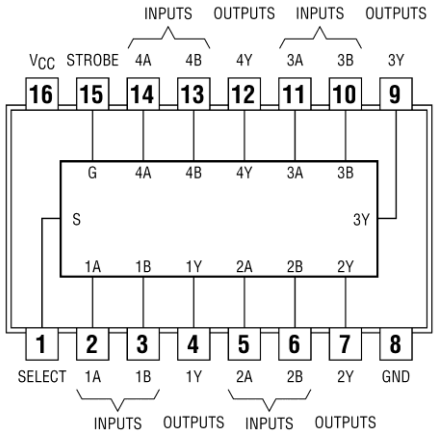
04 인코더

5. 인코더 IC

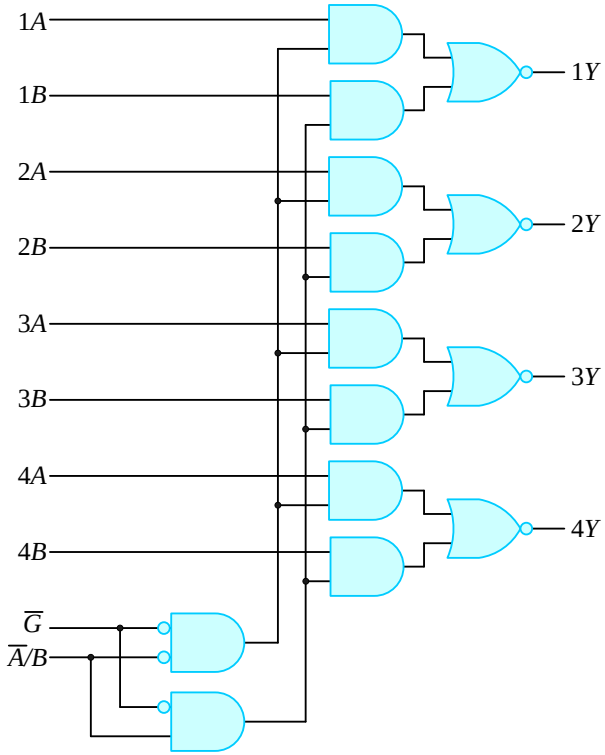
■ 74158 : 2×1 인코더/멀티플렉서가 4개 내장

입력		출력
S	E	Y
×	1	1
0	0	
1	0	

<진리표>



<핀 배치도>



<회로도>

04 인코더

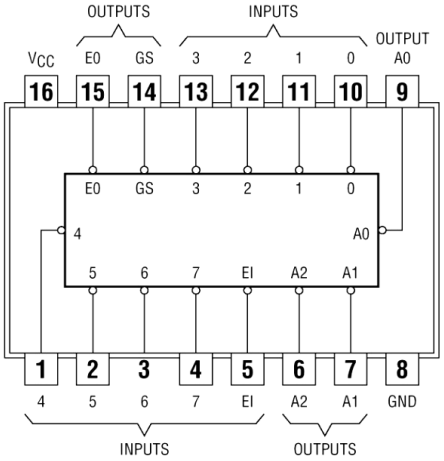
74148(8×3 우선순위 인코더)

- 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

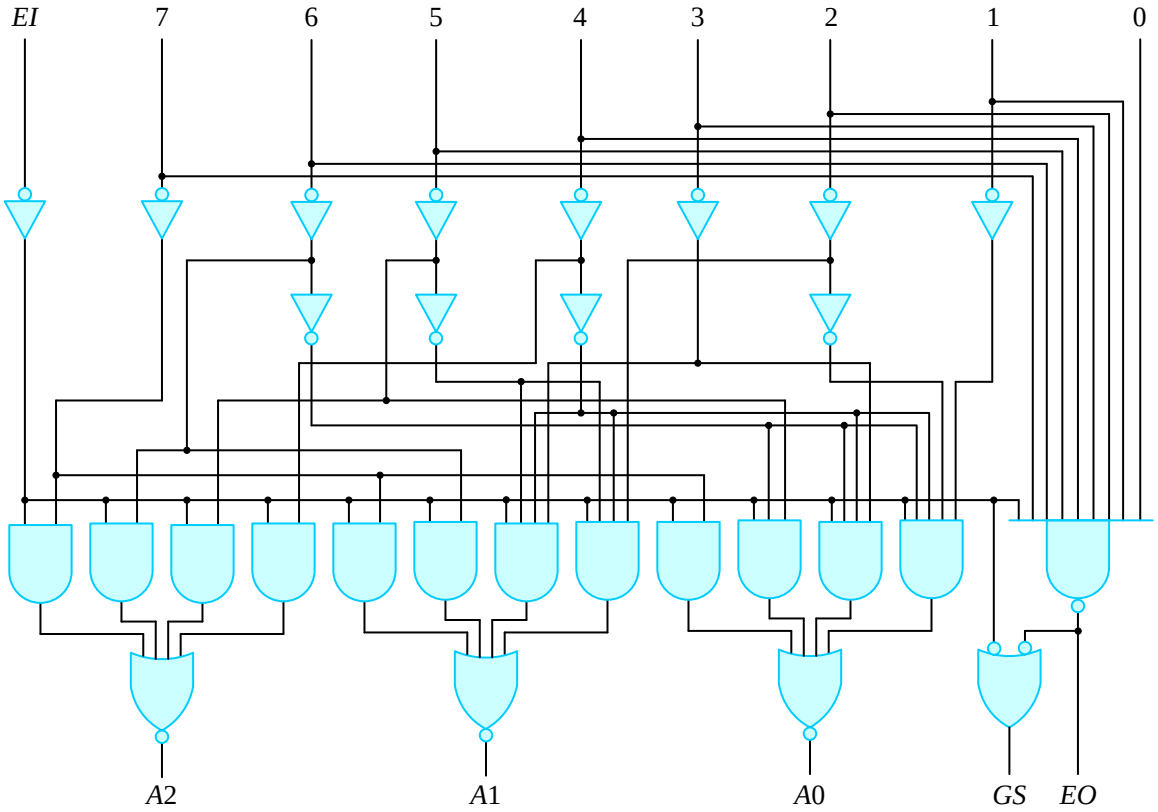
입력									출력				
EI	7	6	5	4	3	2	1	0	A_2	A_1	A_0	*GS*	*EO*
1	×	×	×	×	×	×	×	×	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0
0	0	×	×	×	×	×	×	×	0	0	0	0	1
0	1	0	×	×	×	×	×	×	0	0	1	0	1
0	1	1	0	×	×	×	×	×	0	1	0	0	1
0	1	1	1	0	×	×	×	×	0	1	1	0	1
0	1	1	1	1	0	×	×	×	1	0	0	0	1
0	1	1	1	1	1	0	×	×	1	0	1	0	1
0	1	1	1	1	1	1	0	×	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>

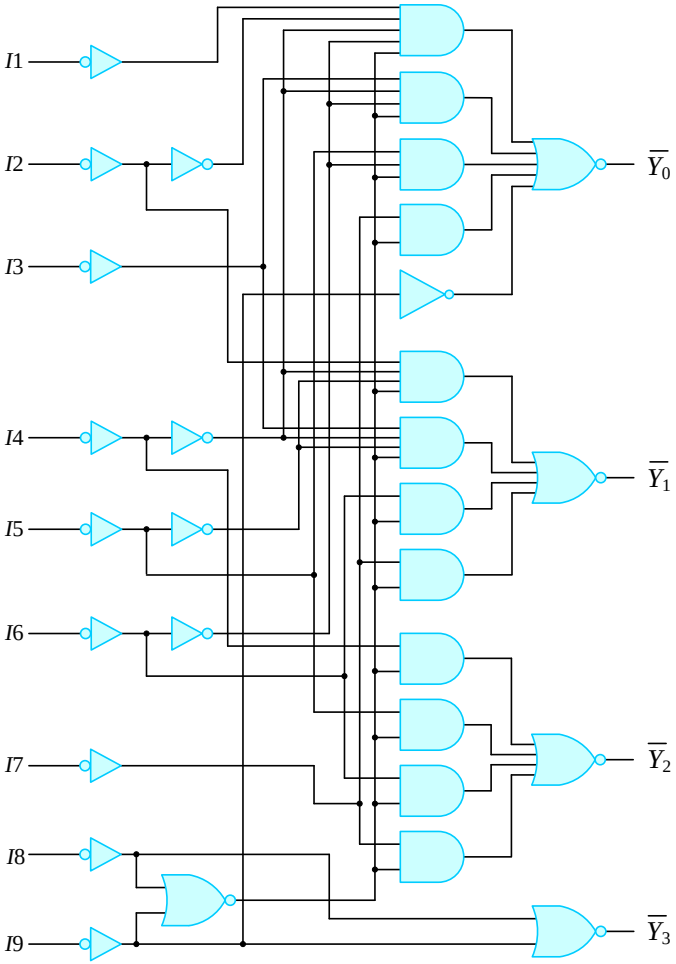
04 인코더

6. 10진-BCD 우선순위 인코더

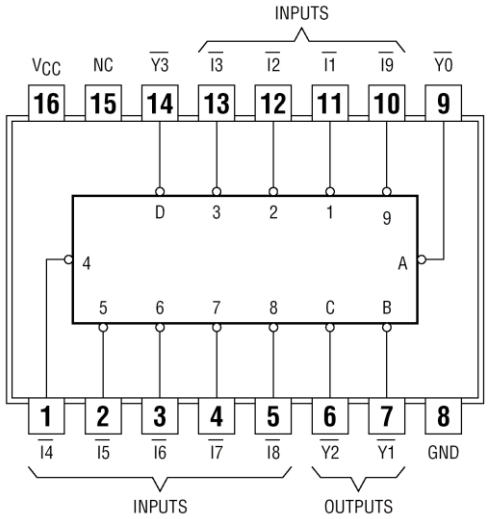
입력									출력			
*I*9	*I*8	*I*7	*I*6	*I*5	*I*4	*I*3	*I*2	*I*1				
1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	×	1	1	0	1
1	1	1	1	1	1	0	×	×	1	1	0	0
1	1	1	1	1	0	×	×	×	1	0	1	1
1	1	1	1	0	×	×	×	×	1	0	1	0
1	1	1	0	×	×	×	×	×	1	0	0	1
1	1	0	×	×	×	×	×	×	1	0	0	0
1	0	×	×	×	×	×	×	×	0	1	1	1
0	×	×	×	×	×	×	×	×	0	1	1	0

<진리표>

04 인코더



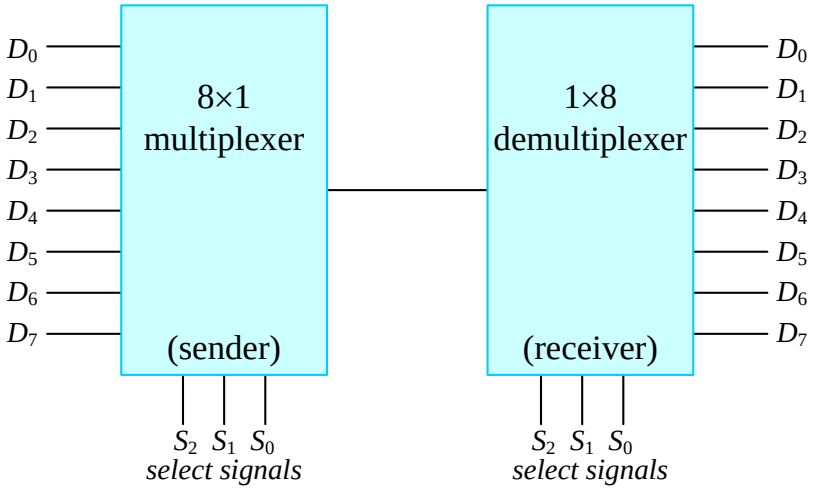
<IC 74147 회로도>



<IC 74147 핀 배치도>

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.
- 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.



<멀티플렉서와 디멀티플렉서의 역할>

05 멀티플렉서

1. 2×1 멀티플렉서

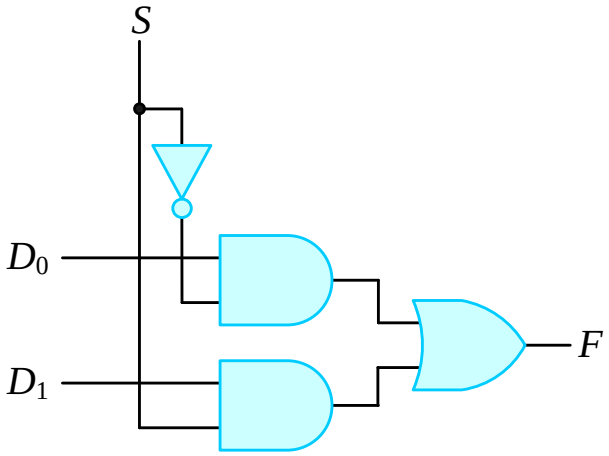
- 2(=2^1)개의 입력중의 하나를 선택선 S에 입력된 값에 따라서 출력으로 보내주는 조합회로

선택선	출력
S	F
0	D_0
1	D_1

<진리표>

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

<논리식>



<회로도>

05 멀티플렉서

2. 4×1 멀티플렉서

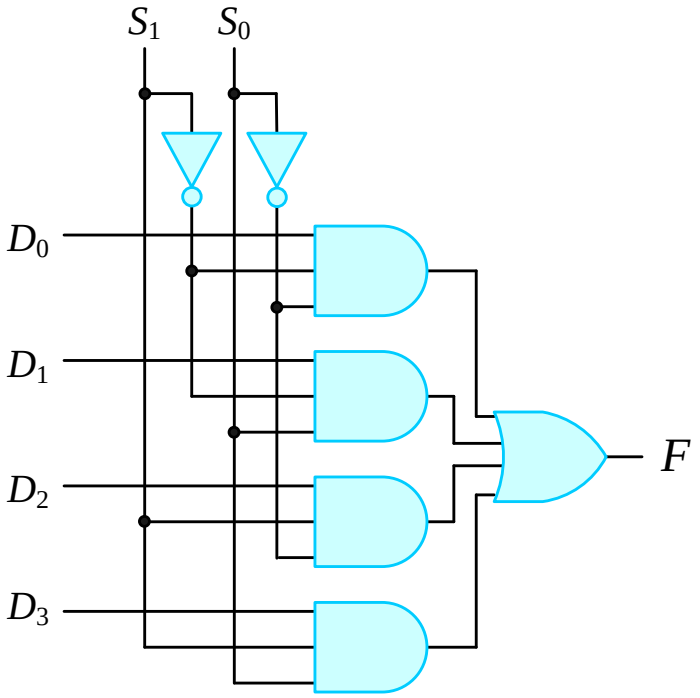
- 4(=2^2)개의 입력중의 하나를 선택선 S_1과 S_0에 입력된 값에 따라서 출력으로 보내주는 조합회로

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>

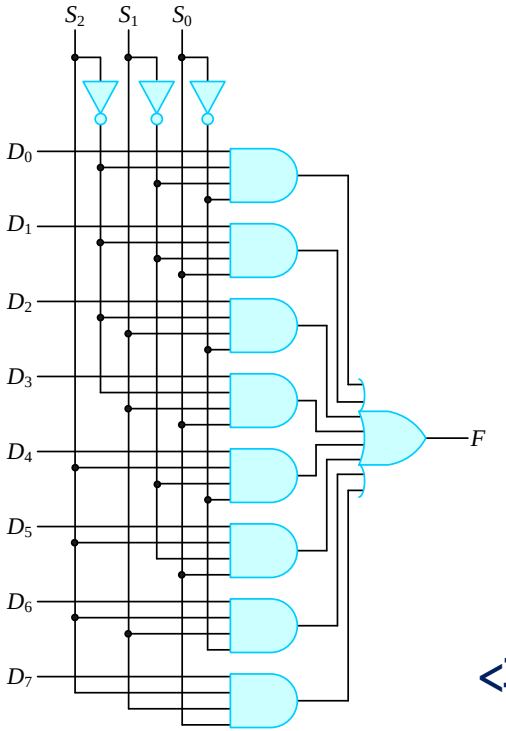
05 멀티플렉서

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>



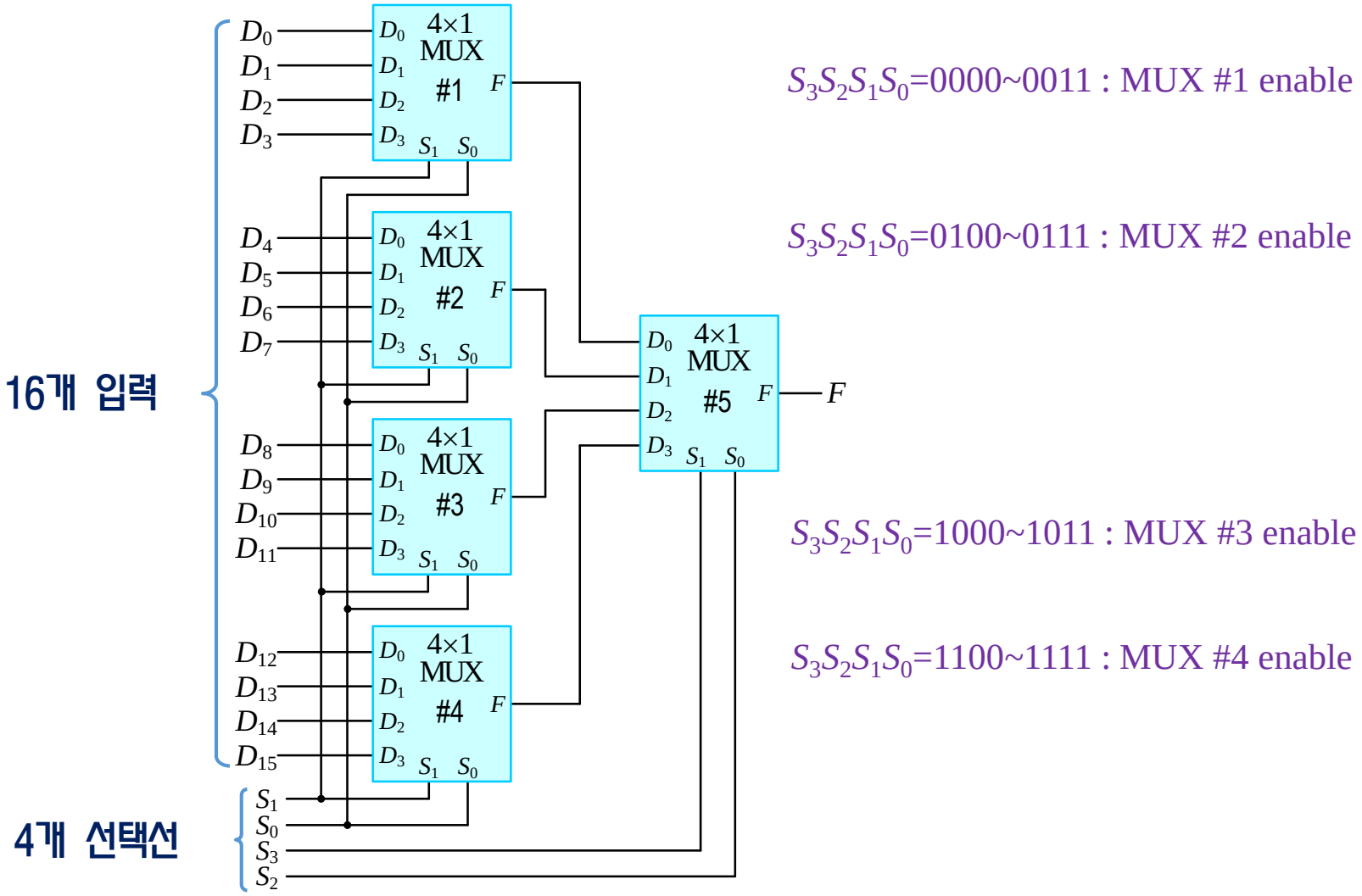
<회로도>

<논리식>

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

05 멀티플렉서

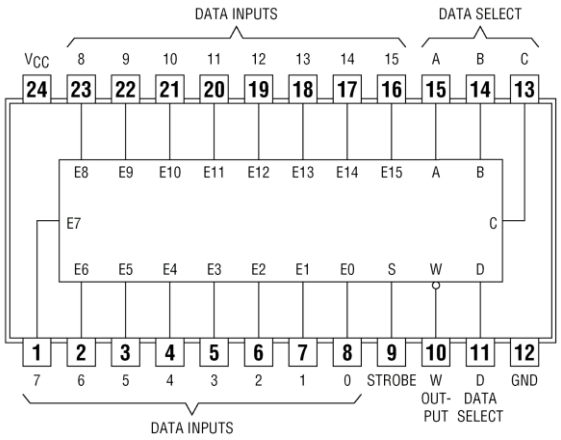
4×1 멀티플렉서 5개를 이용한 16×1 멀티플렉서



05 멀티플렉서

4. 멀티플렉서 IC

- 74150(16×1 멀티플렉서)

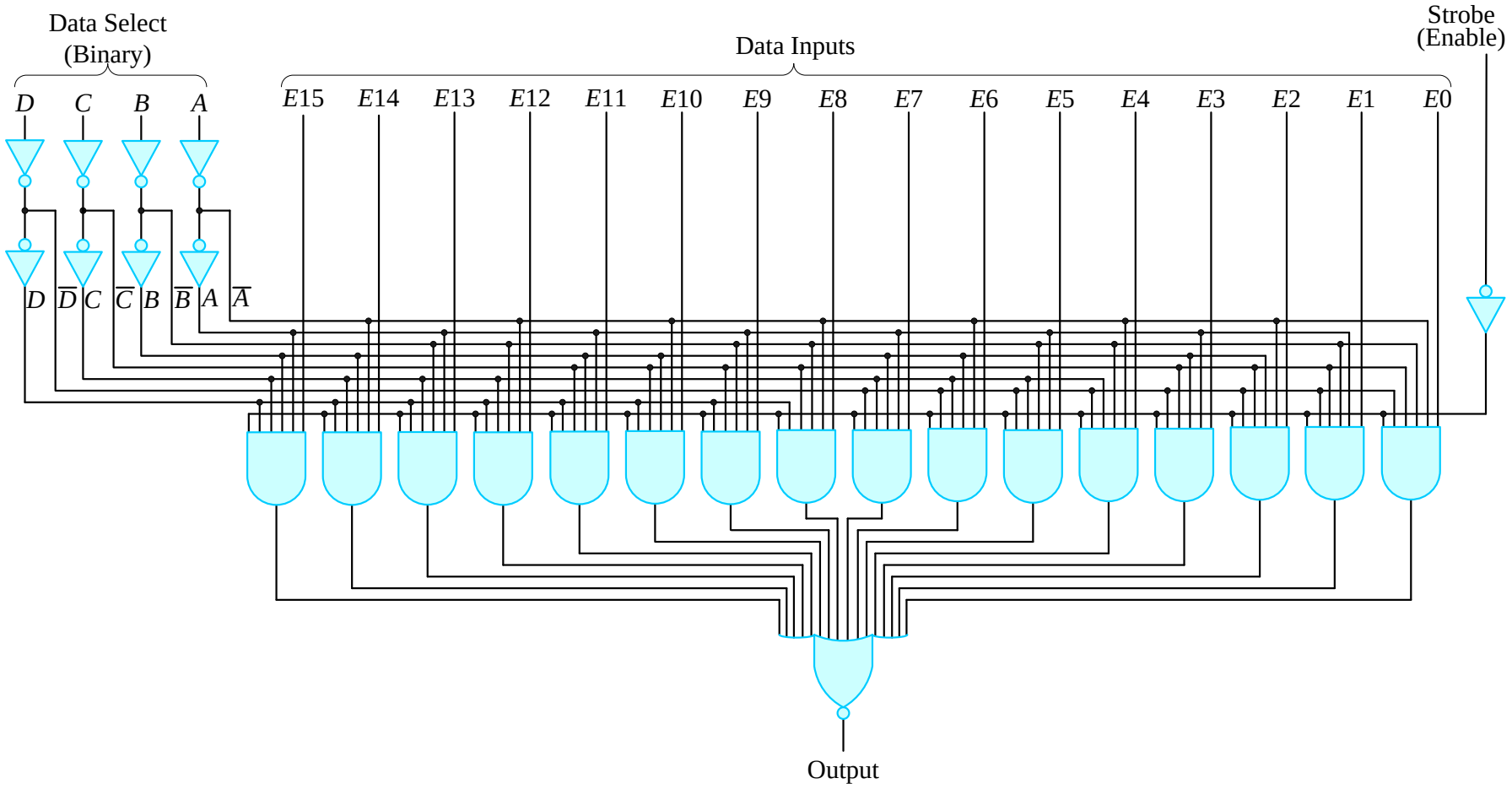


<핀 배치도>

입력					출력
Select				Strobe	W
D	C	B	A	S	
×	×	×	×	1	1
0	0	0	0	0	$\overline{E0}$
0	0	0	1	0	$\overline{E1}$
0	0	1	0	0	$\overline{E2}$
0	0	1	1	0	$\overline{E3}$
0	1	0	0	0	$\overline{E4}$
0	1	0	1	0	$\overline{E5}$
0	1	1	0	0	$\overline{E6}$
0	1	1	1	0	$\overline{E7}$
1	0	0	0	0	$\overline{E8}$
1	0	0	1	0	$\overline{E9}$
1	0	1	0	0	$\overline{E10}$
1	0	1	1	0	$\overline{E11}$
1	1	0	0	0	$\overline{E12}$
1	1	0	1	0	$\overline{E13}$
1	1	1	0	0	$\overline{E14}$
1	1	1	1	0	$\overline{E15}$

<진리표>

05 멀티플렉서


Data Select
(Binary)
Data Inputs
Strobe
(Enable)
D C B A
E15 E14 E13 E12 E11 E10 E9 E8 E7 E6 E5 E4 E3 E2 E1 E0
D D̄ C C̄ B B̄ A Ā
Output

<회로도>

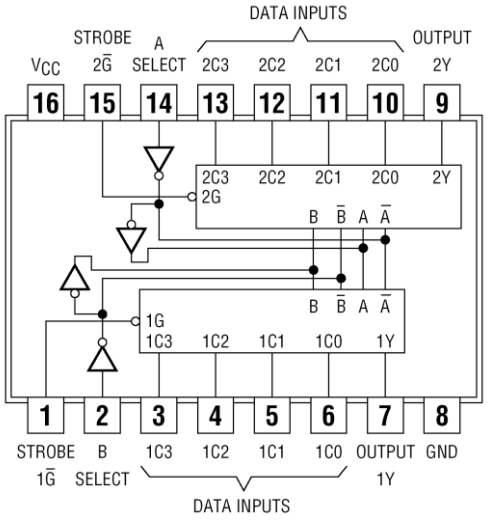
05 멀티플렉서

74153(4×1 인코더)

- 입력 $C0, C1, C2, C3$중 한 개를 선택선 입력 A, B에 따라서 출력으로 보내주는 4×1 멀티플렉서 2개 내장

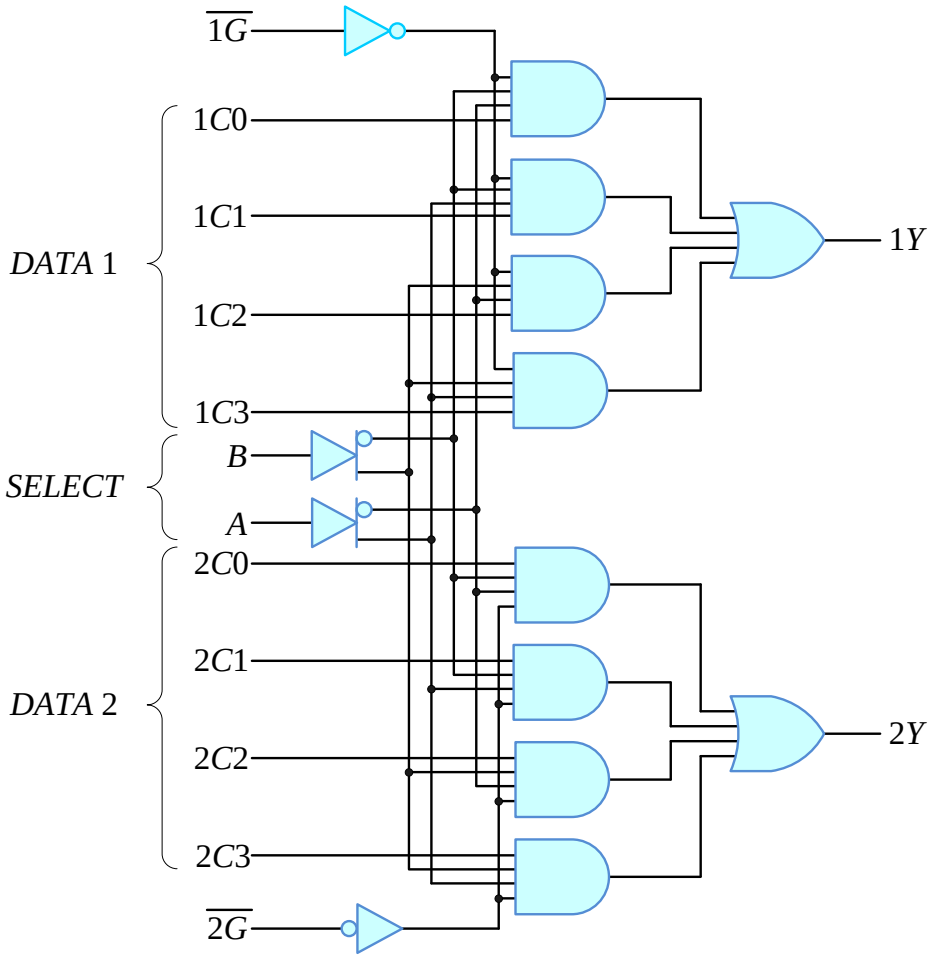
입력			출력
Select		Strobe	Y
B	A	G	
×	×	1	0
0	0	0	$C0$
0	1	0	$C1$
1	0	0	$C2$
1	1	0	C3

<진리표>



<핀 배치도>

05 멀티플렉서


1$\overline{G}$
1C0
1C1
1C2
1C3
DATA 1
B
A
SELECT
2C0
2C1
2C2
2C3
DATA 2
2$\overline{G}$
1Y
2Y

<회로도>

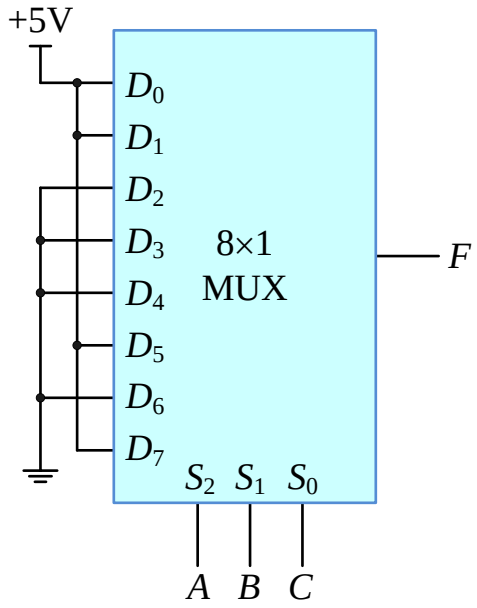
05 멀티플렉서

5. 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>



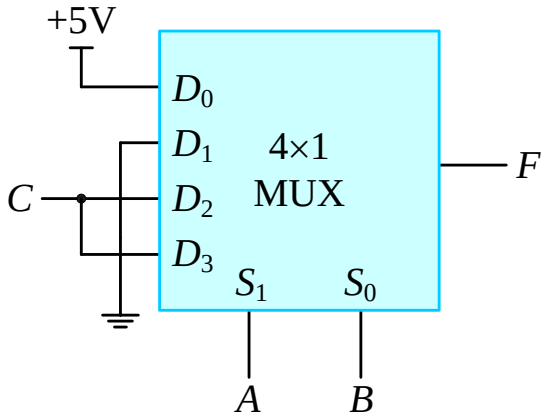
<회로도>

05 멀티플렉서

- $F(A,B,C)=\sum m(0,1,5,7)$ 를 4×1 멀티플렉서로 구현하는 경우
  - ☞ A, B 는 선택선으로 C 는 D_0, D_1, D_2, D_3을 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

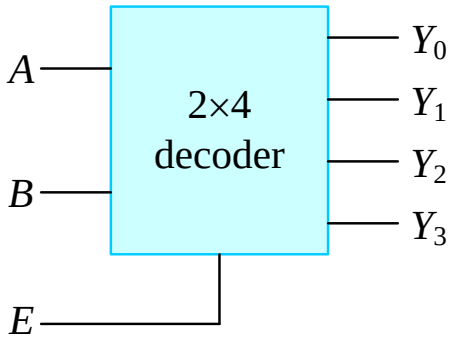
<진리표>



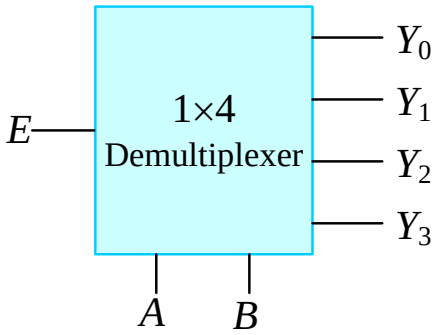
<회로도>

06 디멀티플렉서

- 1개의 인에이블 입력을 가지고 있는 디코더는 디멀티플렉서로서의 기능을 수행
- 디멀티플렉서는 정보를 한 선으로 받아서 2^n개의 가능한 출력 선들 중 하나를 선택하여, 받은 정보를 전송하는 회로이다. 디멀티플렉서는 n개의 선택선(selection line)들을 이용하여 출력을 제어



<2×4 디코더>



<1×4 디멀티플렉서>

06 디멀티플렉서

IC 74138을 디멀티플렉서로 사용하는 경우

- IC 74138의 인에이블($G1$)을 데이터 입력으로 사용하고 A, B, C를 선택선으로 사용하면 $G1$의 부정이 출력된다.

입력			출력							
C	B	A	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	0	0	1	1	1	1	1	1	1	$\overline{G1}$
0	0	1	1	1	1	1	1	1	$\overline{G1}$	1
0	1	0	1	1	1	1	1	$\overline{G1}$	1	1
0	1	1	1	1	1	1	$\overline{G1}$	1	1	1
1	0	0	1	1	1	$\overline{G1}$	1	1	1	1
1	0	1	1	1	$\overline{G1}$	1	1	1	1	1
1	1	0	1	$\overline{G1}$	1	1	1	1	1	1
1	1	1	$\overline{G1}$	1	1	1	1	1	1	1

<IC 74138을 디멀티플렉서로 사용할 때 진리표>

- 74154 IC를 디멀티플렉서로 사용할 때는 G_1과 G_2가 데이터 선이 되고 A, B, C, D가 선택선이 된다.

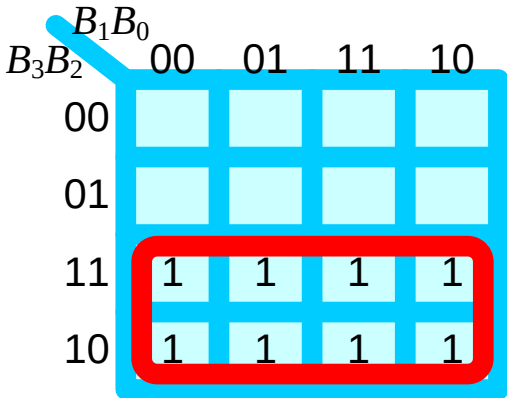
07 코드 변환기

1. 2진 코드-그레이 코드 변환

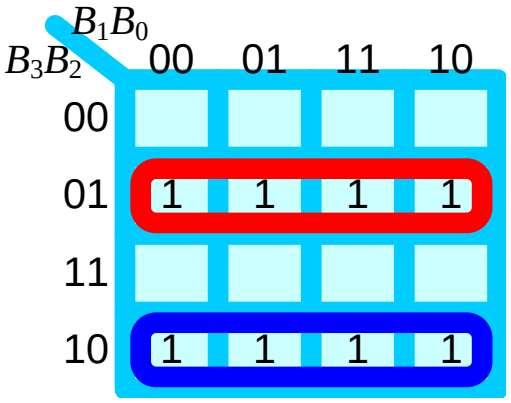
<2진 코드-그레이 코드 변환 진리표>

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

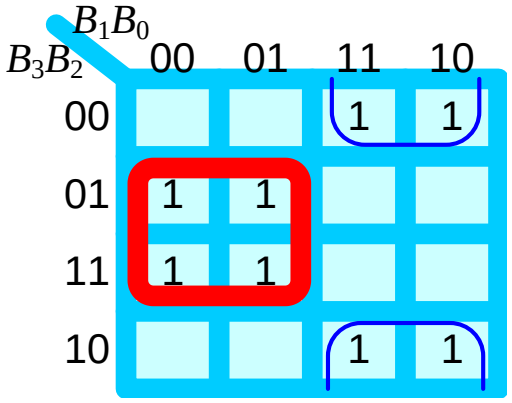
07 코드 변환기



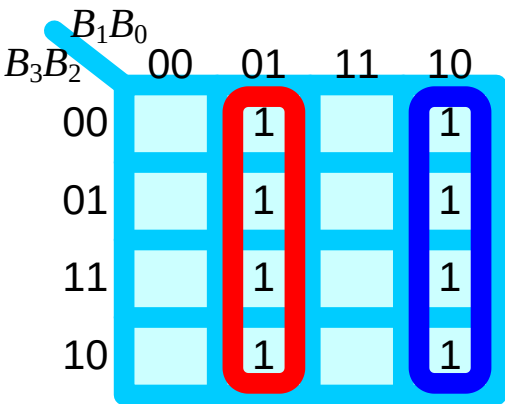
$G_3 = B_3$



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

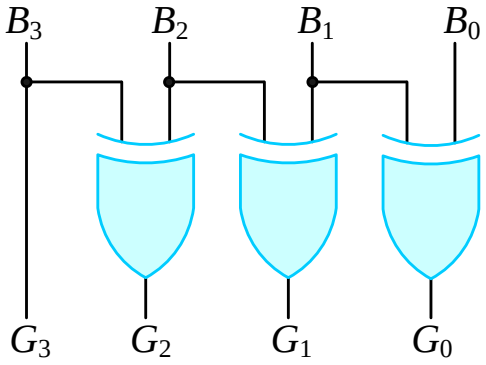


$G_1 = \bar{B}_2B_1 + B_2\bar{B}_1 = B_2 \oplus B_1$



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

카르노 맵



회로도

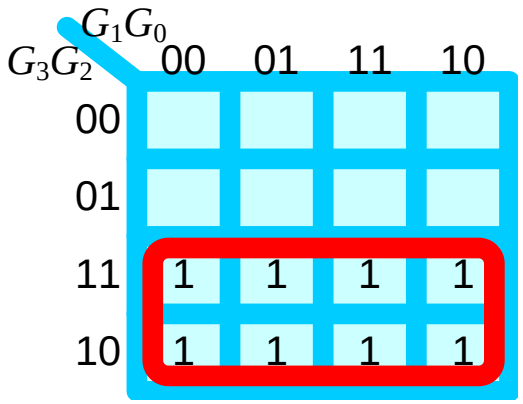
07 코드 변환기

2. 그레이 코드-2진 코드 변환

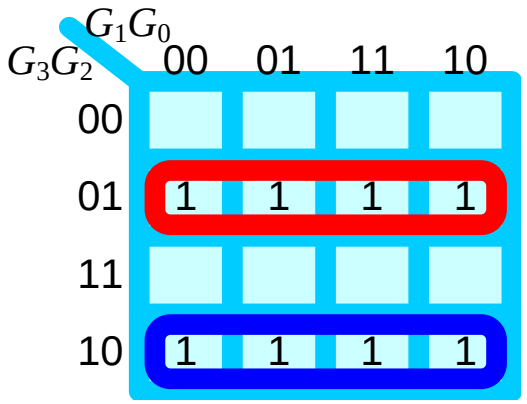
<그레이 코드-2진 코드 변환 진리표>

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

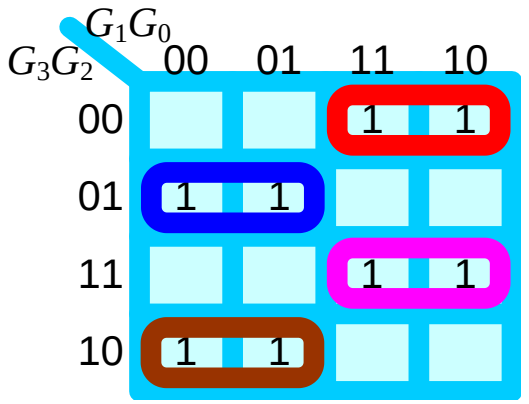
07 코드 변환기



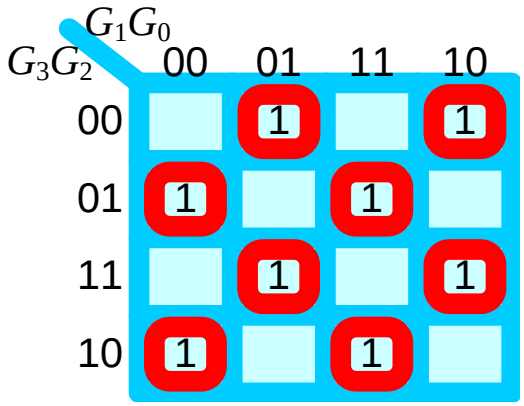
$$B_3 = G_3$$



$$B_2 = \bar{G}_3 G_2 + G_3 \bar{G}_2 = G_3 \oplus G_2$$

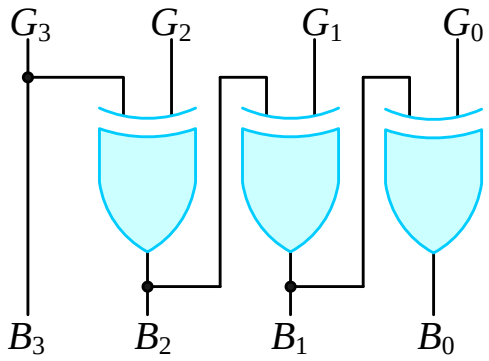


$$B_1 = G_3 \oplus G_2 \oplus G_1 = B_2 \oplus G_1$$



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

<카르노 맵>



<회로도>

07 코드 변환기

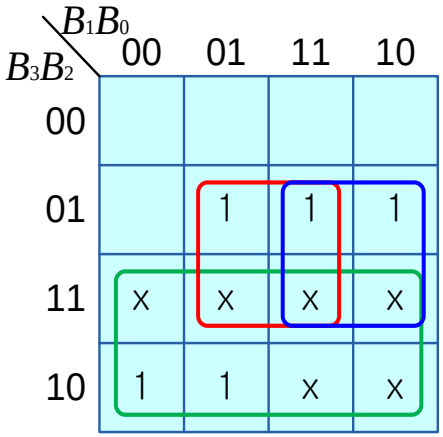
3. BCD 코드-3초과 코드 변환

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

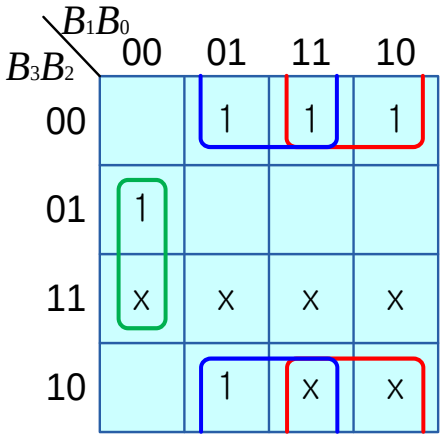
BCD 코드(입력)				3초과 코드(출력)			
B_3	B_2	B_1	B_0	E_3	E_2	E_1	E_0
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1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

<진리표>

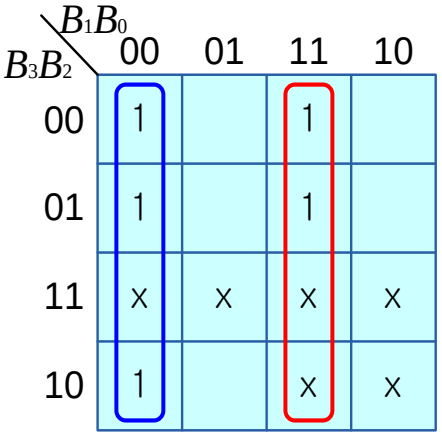
07 코드 변환기



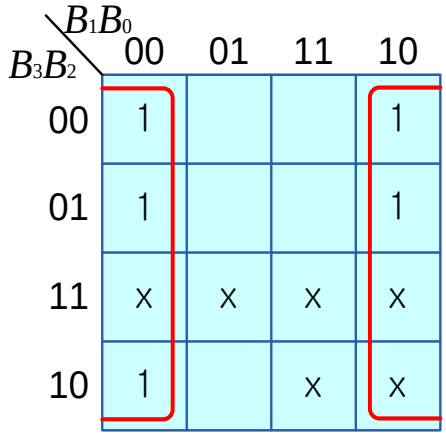
$$E_3 = B_3 + B_2B_1 + B_2B_0$$



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



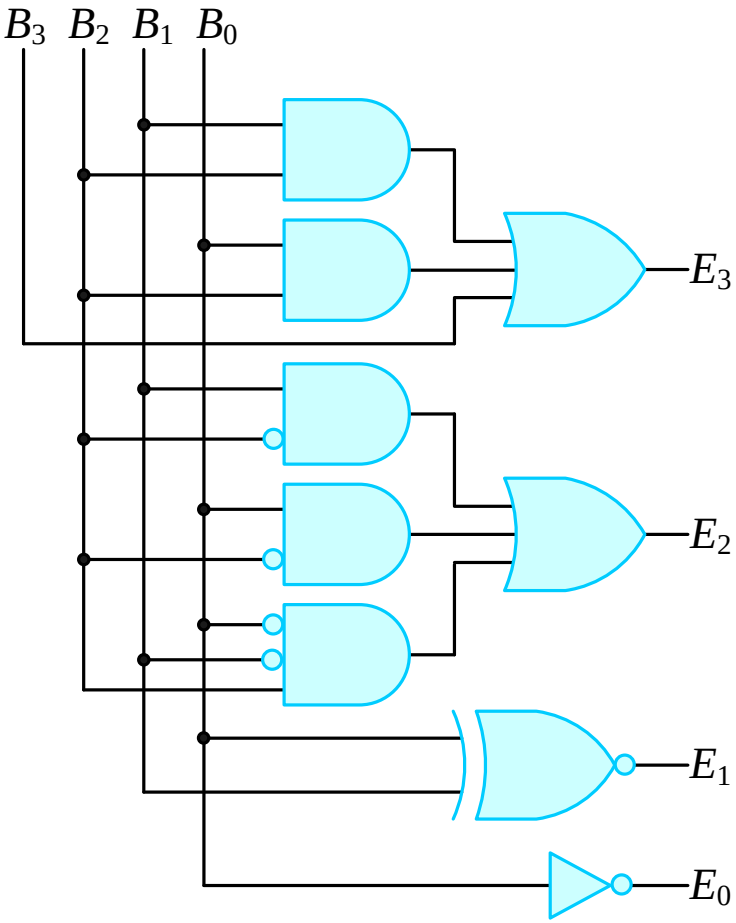
$$E_1 = \overline{B}_1\overline{B}_0 + B_1B_0 = \overline{B_1 \oplus B_0}$$



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

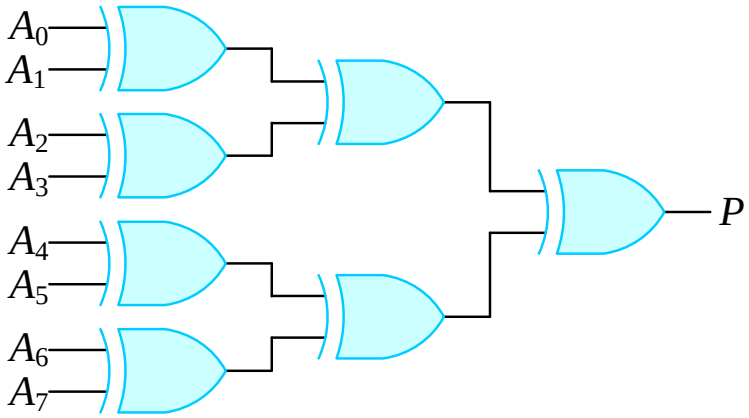
<카르노 맵>

07 코드 변환기

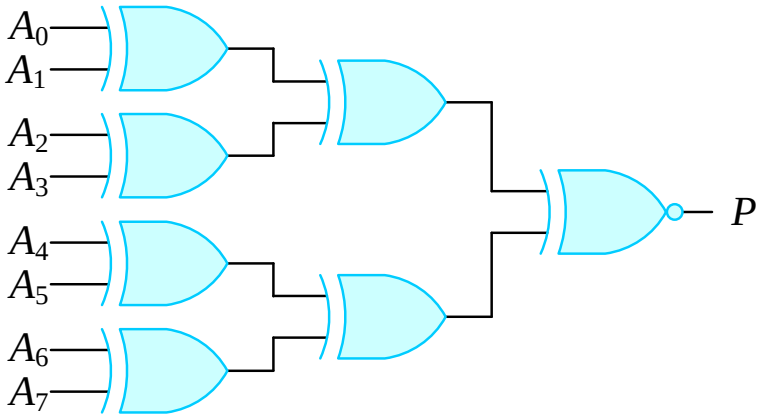


<회로도>

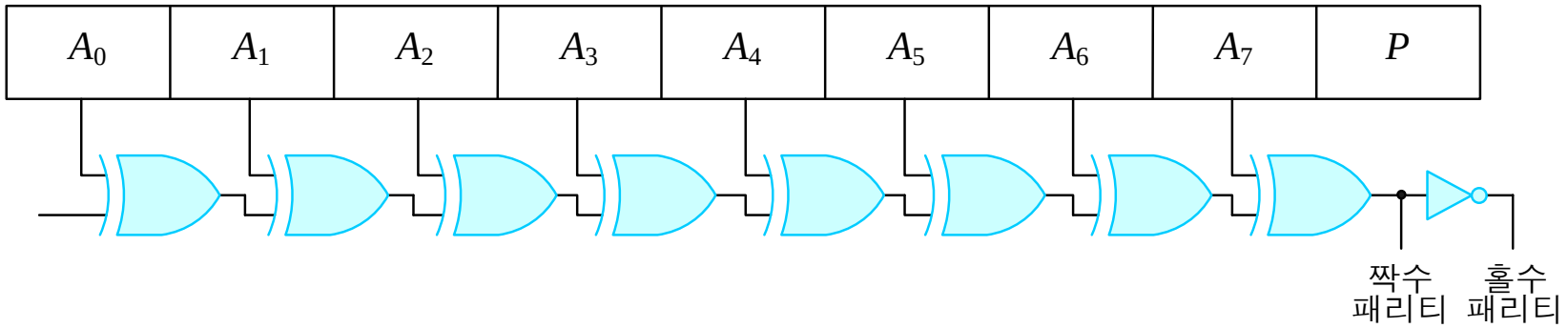
08 패리티 발생기/검출기



<짝수 패리티 발생회로>



<홀수 패리티 발생회로>

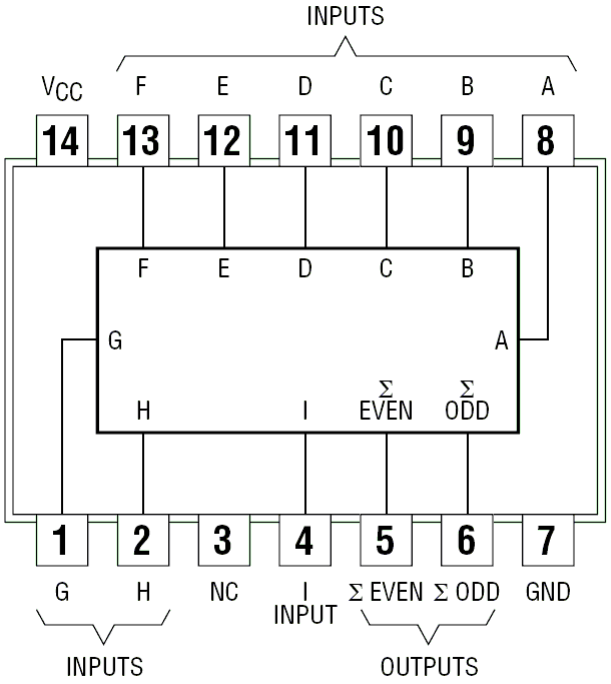


<8비트 직렬회로에서의 짝수/홀수 패리티 발생>

08 패리티 발생기/검출기

IC 74280

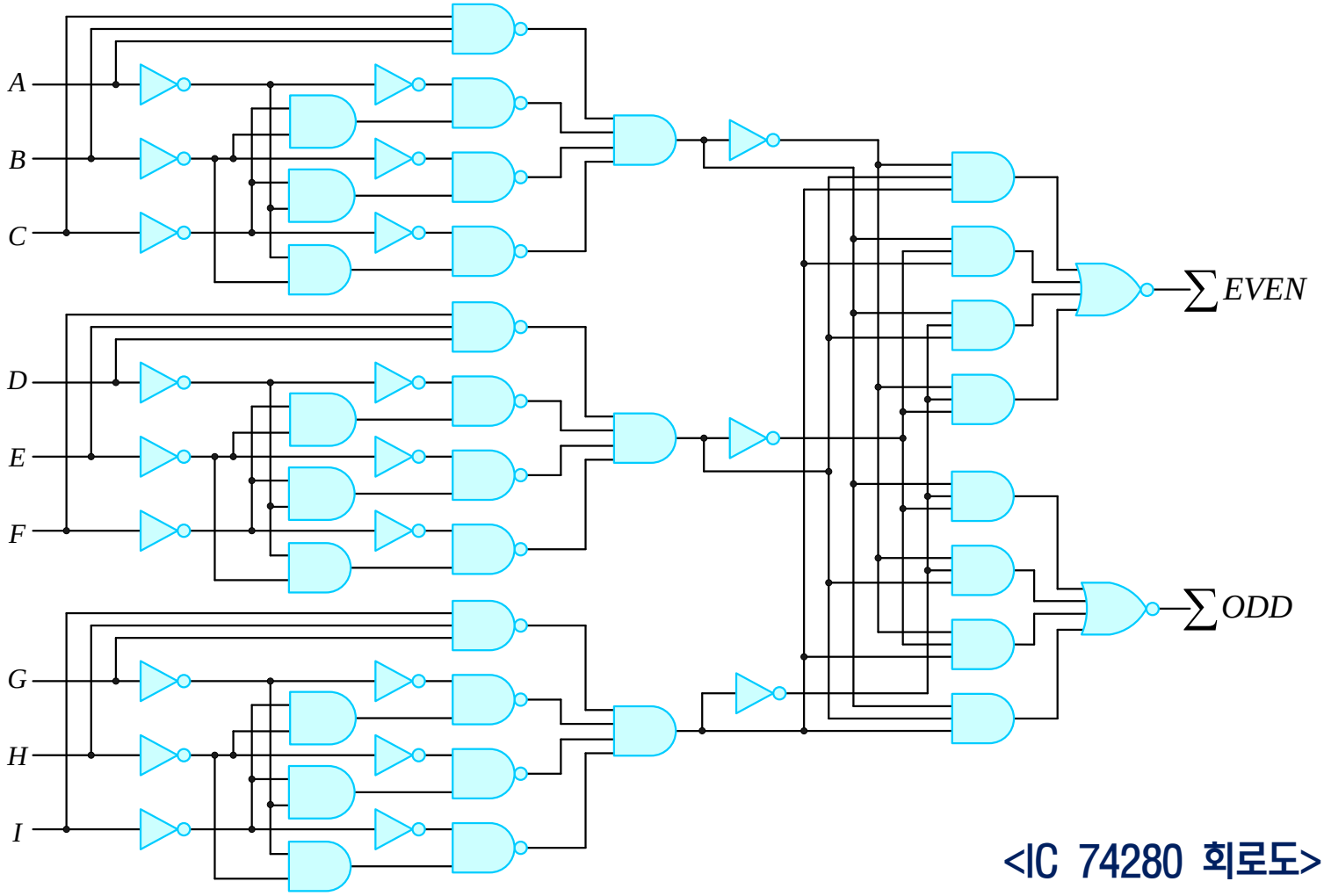
- 9비트 홀수/짝수 패리티 발생과 검출



NC – No internal connection

<핀 배치도>

08 패리티 발생기/검출기



<IC 74280 회로도>

감사합니다 ☺