

7장. 조합논리회로

01. 가산기

02. 비교기

03. 디코더

04. 인코더

05. 멀티플렉서

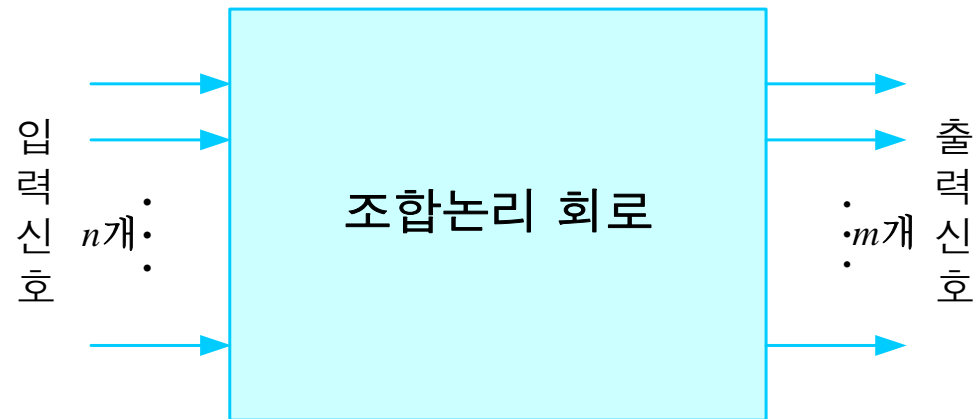
06. 디멀티플렉서

07. 코드 변환기

08. 패리티 발생기/검출기



- ❖ 조합논리회로는 **논리곱**, **논리합**, **논리 부정**의 세 가지 기본 논리 회로를 조합하여 구성한 논리 회로
- ❖ 조합논리회로는 **입력변수**, **논리 게이트**, 그리고 **출력변수**들로 구성



<조합논리회로 블록도>



반가산기(half-adder, HA)

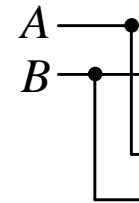
$$\begin{array}{r} A \\ + B \\ \hline C \quad S \end{array}$$

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

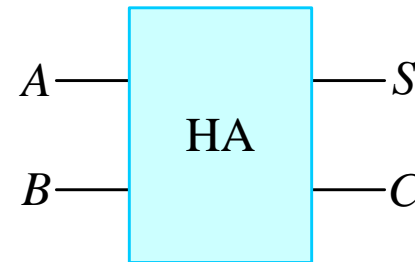
$S =$

$C =$

<진리표와 논리식>



<논리회로>



<논리기호>



전가산기(full-adder, FA)

❖ 자리 올림수(carry)를 고려하여 만든 덧셈 회로

$$\begin{array}{r} C_{in} \\ A \\ + B \\ \hline C_{out} S \end{array}$$

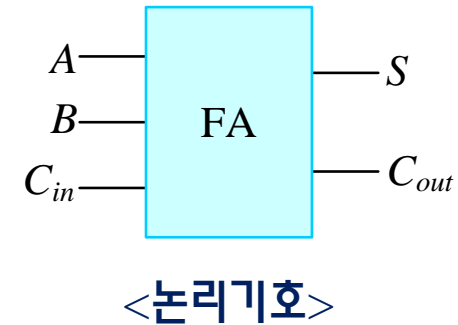
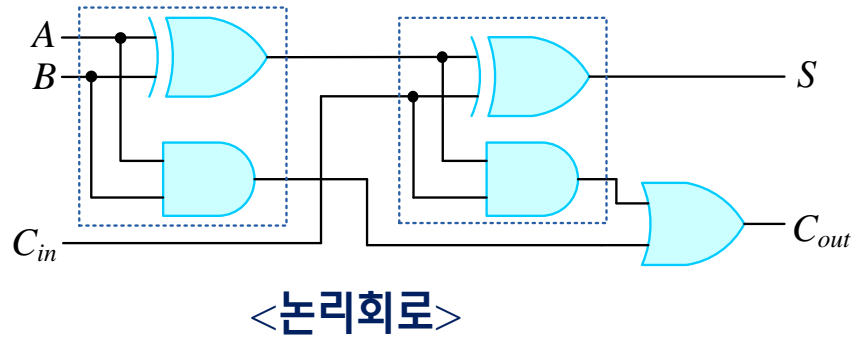


입력			출력	
A	B	C _{in}	S	C _{out}
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

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

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

진리표와 논리식



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

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

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

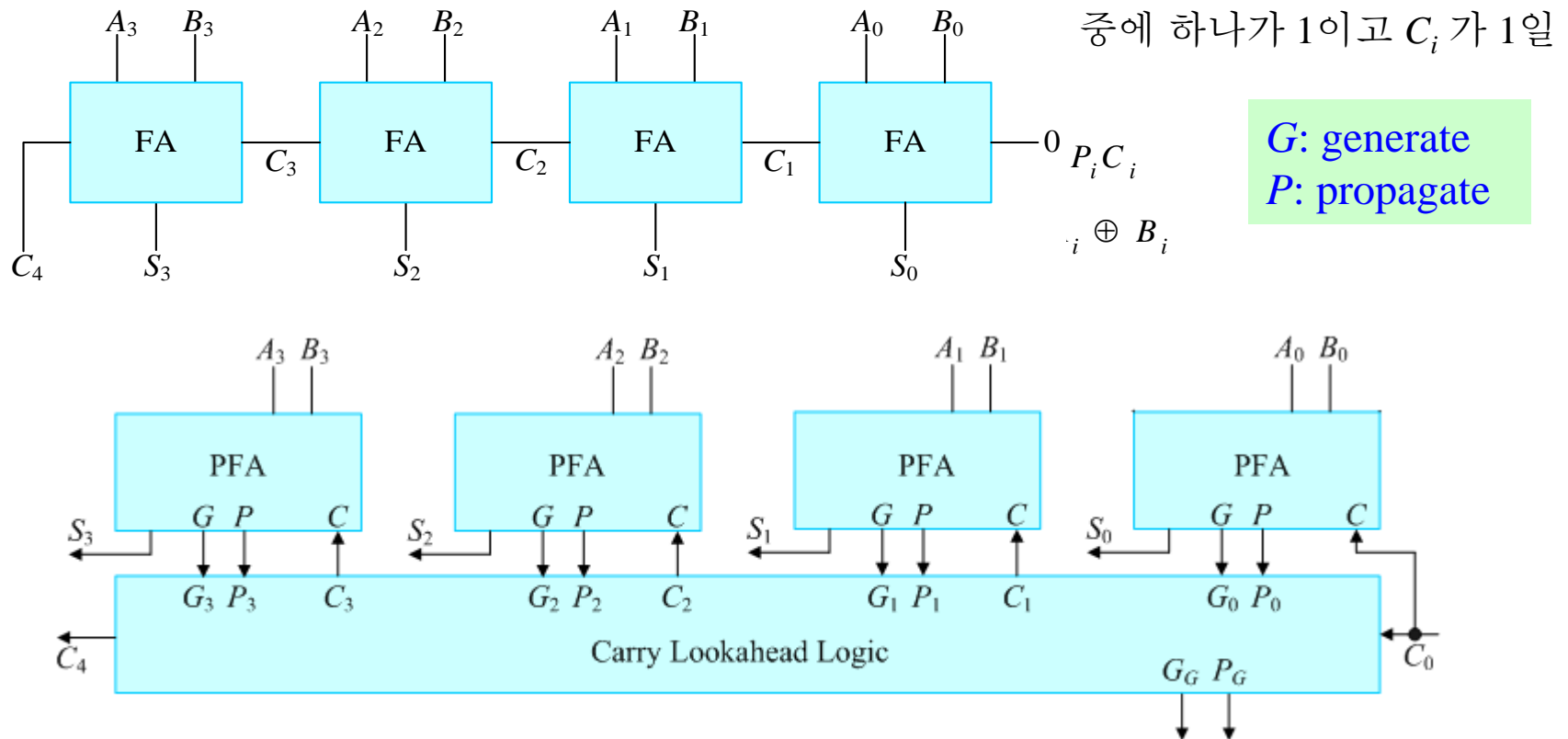
The diagram illustrates a 4-bit ripple-carry adder implemented using four Full Adders (FA) connected in a chain. The inputs are $A_0, B_0, A_1, B_1, A_2, B_2, A_3, B_3$. The carry-in to the first FA is 0. The carry-out of each FA is connected to the carry-in of the next FA, labeled C_1, C_2, C_3 . The final carry-out is C_4 . The sum outputs are S_0, S_1, S_2, S_3 .

- 6 -



고속가산기(high-speed-adder)

- ❖ 아랫단에서 윗단으로 전달되는 자리올림수 때문에 병렬가산기는 속도가 매우 느리다는 단점이 있음.
- ❖ 이것을 해결하기 위한 방법으로 캐리에측가산기(carry-look-ahead-adder, CLA)를 사용





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



BCD 덧셈표

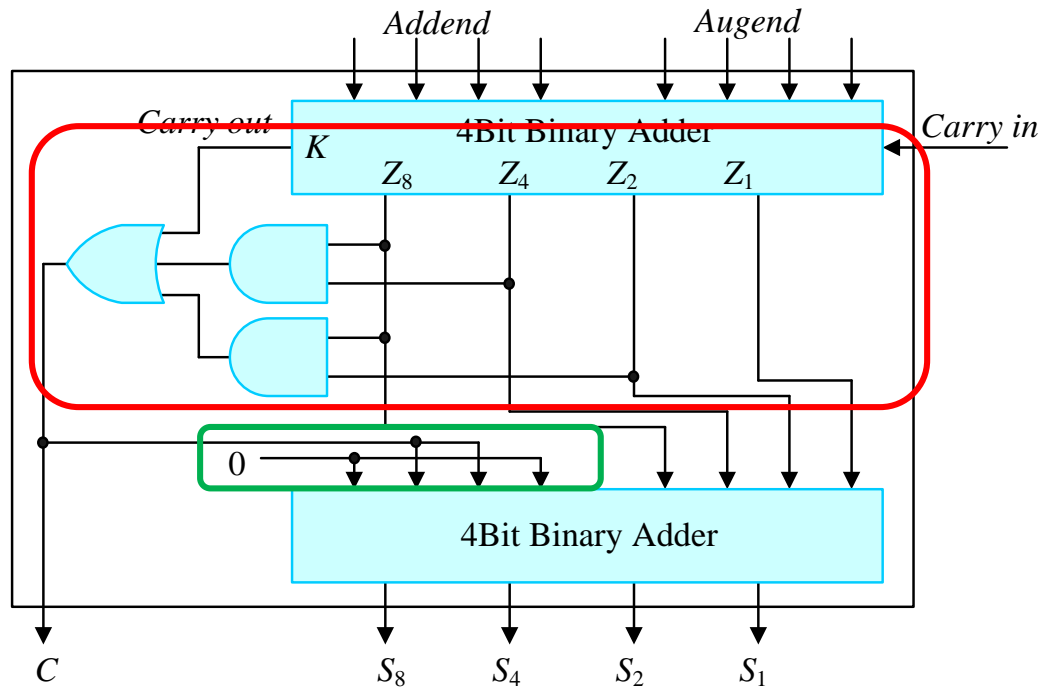
K	2진 한				C	BCD 합				10진 값
	Z ₈	Z ₄	Z ₂	Z ₁		S ₈	S ₄	S ₂	S ₁	
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	2
0	0	0	1	1	0	0	0	1	1	3
0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	1	0	0	1	0	1	5
0	0	1	1	0	0	0	1	1	0	6
0	0	1	1	1	0	0	1	1	1	7
0	1	0	0	0	0	1	0	0	0	8
0	1	0	0	1	0	1	0	0	1	9
0	1	0	1	0	1	0	0	0	0	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



❖ BCD 합에서 캐리를 만들어 주어야 하는 경우의 논리식

$$C = K + Z_8 Z_4 + Z_8 Z_2$$



$Z_8 Z_4 \backslash Z_2 Z_1$	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	1	1	1	1
10	0	0	1	1

BCD 가산기



❖ 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



□ 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

02 비교기



B_2B_1 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 A_2A_1	00	01	11	10
00		1	1	1
01	1		1	1
11	1	1		1
10	1	1	1	

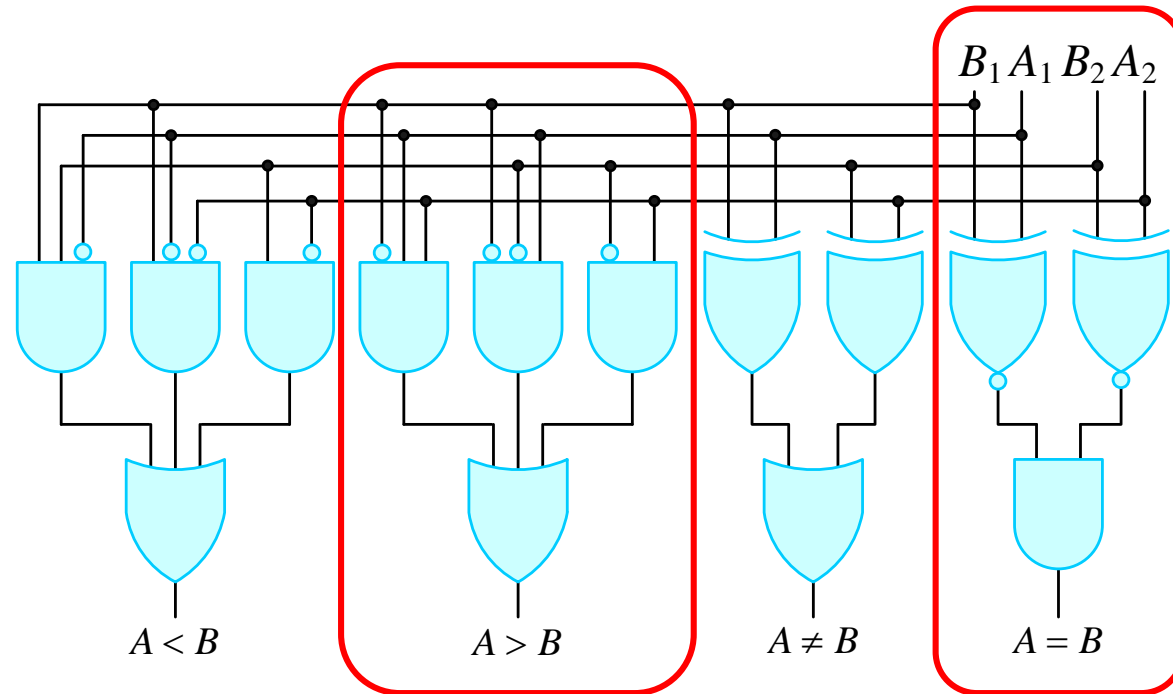
$$F_2 = (A_1 \oplus B_1) + (A_2 \oplus B_2)$$

B_2B_1 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 A_2A_1	00	01	11	10
00		1	1	1
01			1	1
11				
10			1	

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

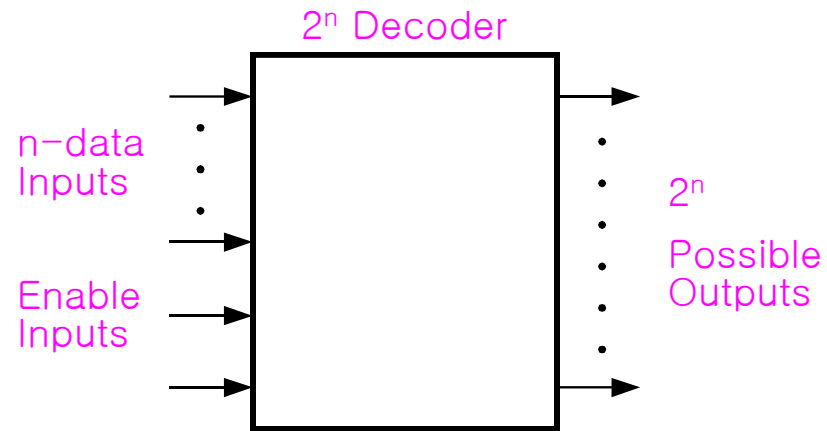


2비트 비교기 회로

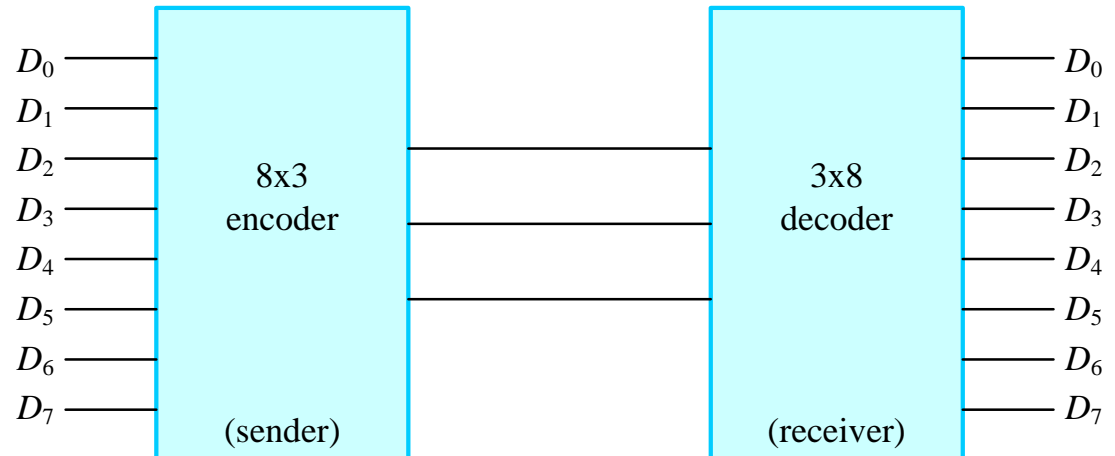


□ 디코더(decoder)

- ❖ 디코더 : 입력선에 나타나는 n 비트의 2진 코드를 최대 2^n 개의 서로 다른 정보로 바꿔주는 조합논리회로



디코더와 인코더의 기능



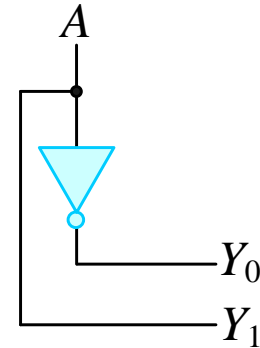


1×2 디코더

❖ 1개의 입력에 따라서 2개의 출력 중 하나가 선택

입력	출력	
A	Y_1	Y_0
0	0	1
1	1	0

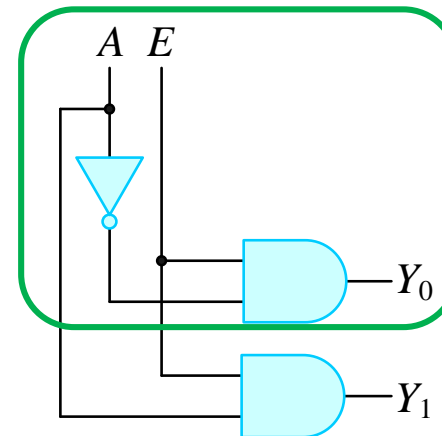
$$Y_0 = \overline{A} \quad Y_1 = A$$



□ 인에이블이 있는 1×2 디코더

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

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



인에이블(enable)

함수 또는 회로를 허락하거나 또는 동작하게 한다.

디스에이블(disable)

함수 또는 회로를 허락하지 않거나 또는 동작하지 못하게 한다.



2×4 디코더

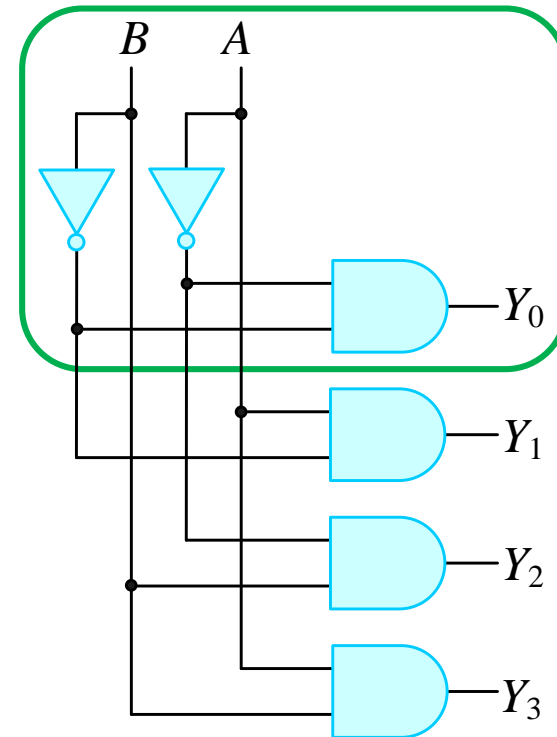
❖ 2개의 입력에 따라서 4개의 출력 중 하나가 선택

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

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

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

진리표와 논리식



회로도



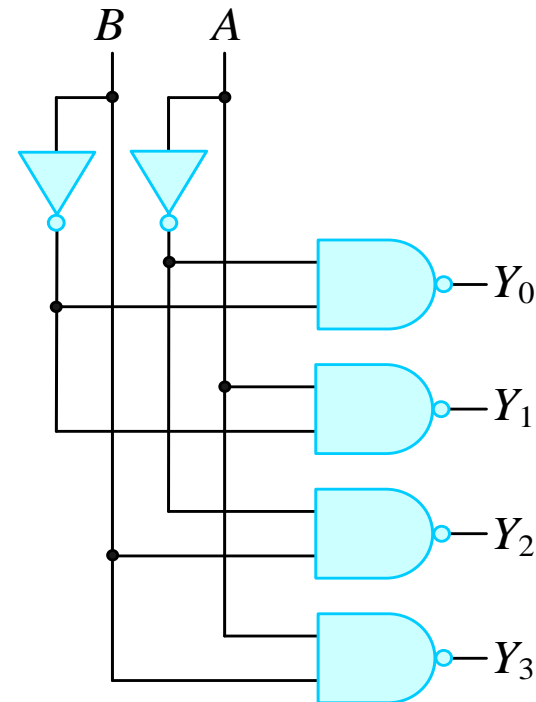
2×4 NAND 디코더

❖ 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

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

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



2×4 NAND 디코더 진리표와 논리식

2×4 NAND 디코더 회로



□ 인에이블이 있는 2×4 디코더

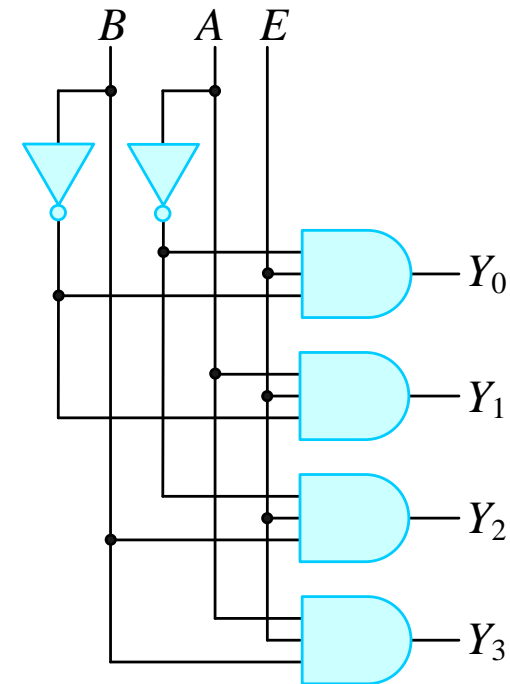
- ❖ 대부분의 IC 디코더들은 인에이블(enable) 입력이 있어서 회로를 제어한다.
- ❖ $E=1$ 일 때만 출력이 동작.

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

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

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

진리표와 논리식



회로도

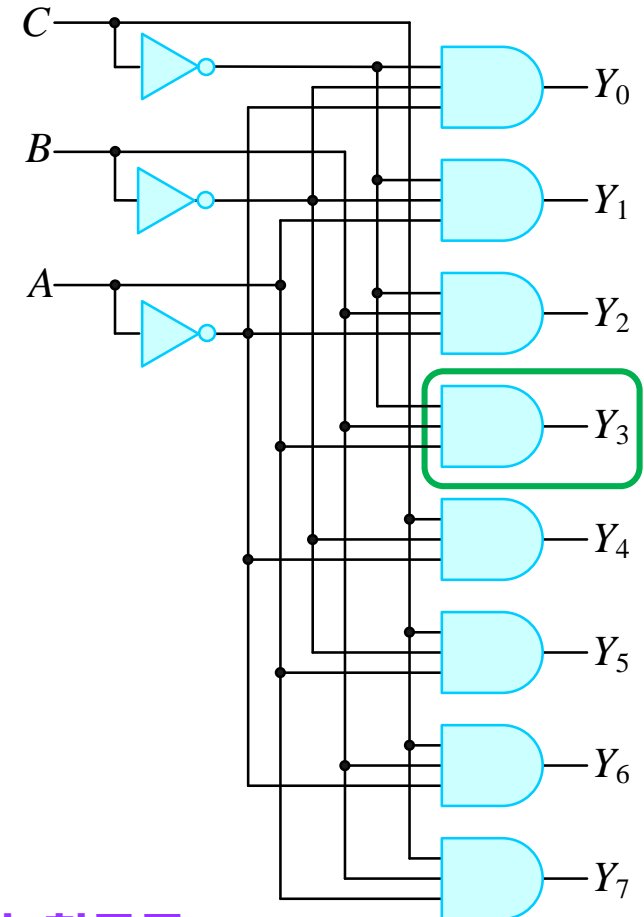


3×8 디코더

❖ 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

$$\begin{aligned}
 Y_0 &= \overline{C} \overline{B} \overline{A}, & Y_1 &= \overline{C} \overline{B} A, & Y_2 &= \overline{C} B \overline{A}, & Y_3 &= \overline{C} B A \\
 Y_4 &= C \overline{B} \overline{A}, & Y_5 &= C \overline{B} A, & Y_6 &= C B \overline{A}, & Y_7 &= C B A
 \end{aligned}$$



3×8 디코더 진리표와 논리식, 회로도



4×16 디코더

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

4×16 디코더 진리표



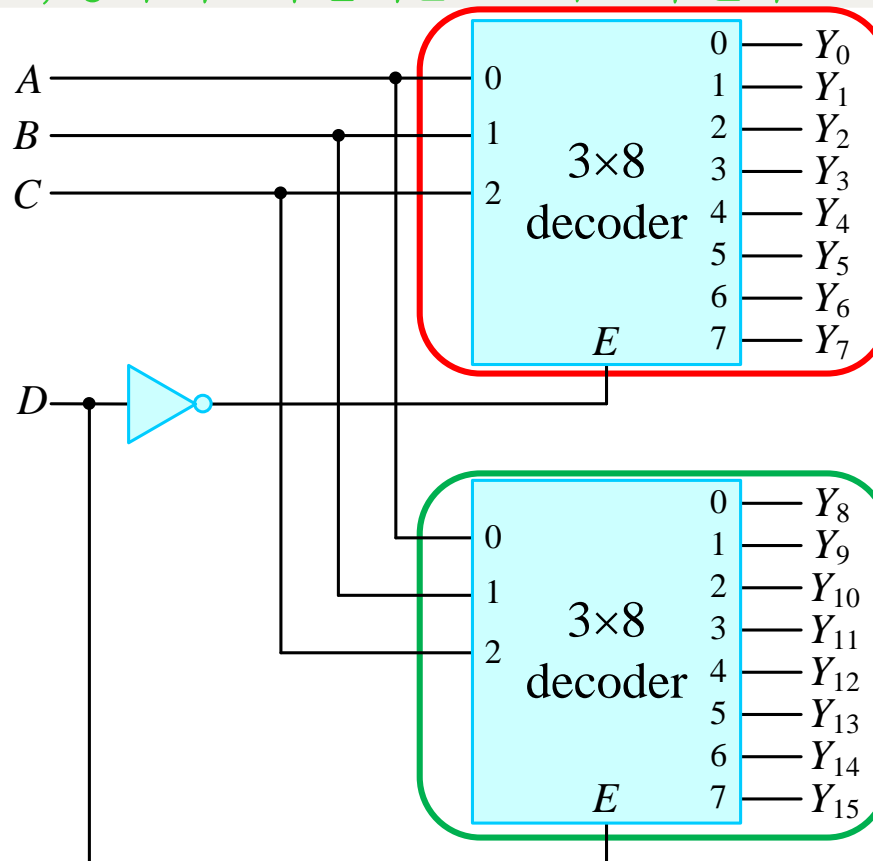
□ 2개의 3×8 디코더로 4×16 디코더를 구성

D=0

상위 디코더만 enable되어 출력은 $Y_0 \sim Y_7$ 중의 하나가 1로 되고, 아래의 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.



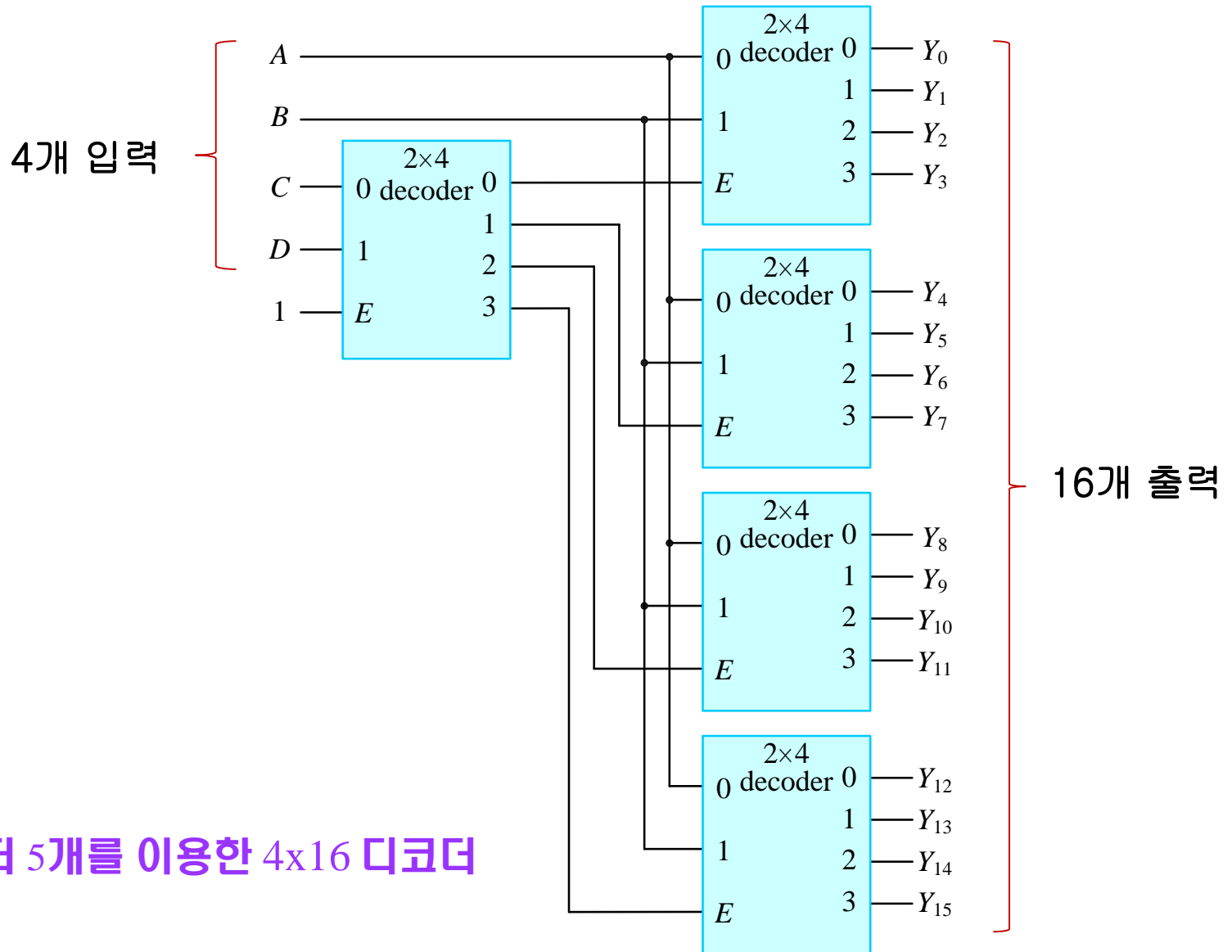


2×4 디코더 5개를 이용한 4×16 디코더

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

4×16 디코더 진리표

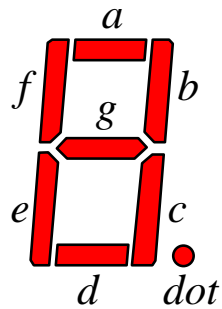
03 디코더



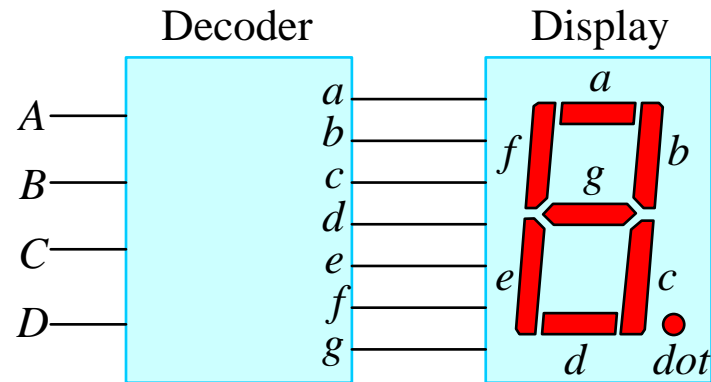


BCD-7-세그먼트 디코더

❖ 7 세그먼트 : 숫자 표시 전용 장치



7-세그먼트 구성



7-세그먼트와 디코더의 연결

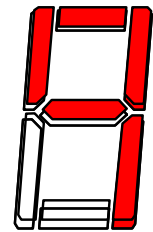
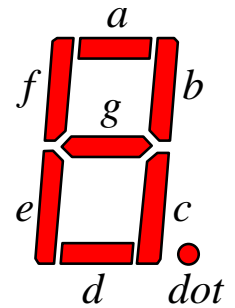
0	1	2	3	4	5	6	7	8	9

7-세그먼트의 숫자 표시



7-세그먼트 디코더 진리표

입력				출력						
<i>D</i>	<i>C</i>	<i>B</i>	<i>A</i>	\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	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



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

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

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

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

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

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

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

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

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

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

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

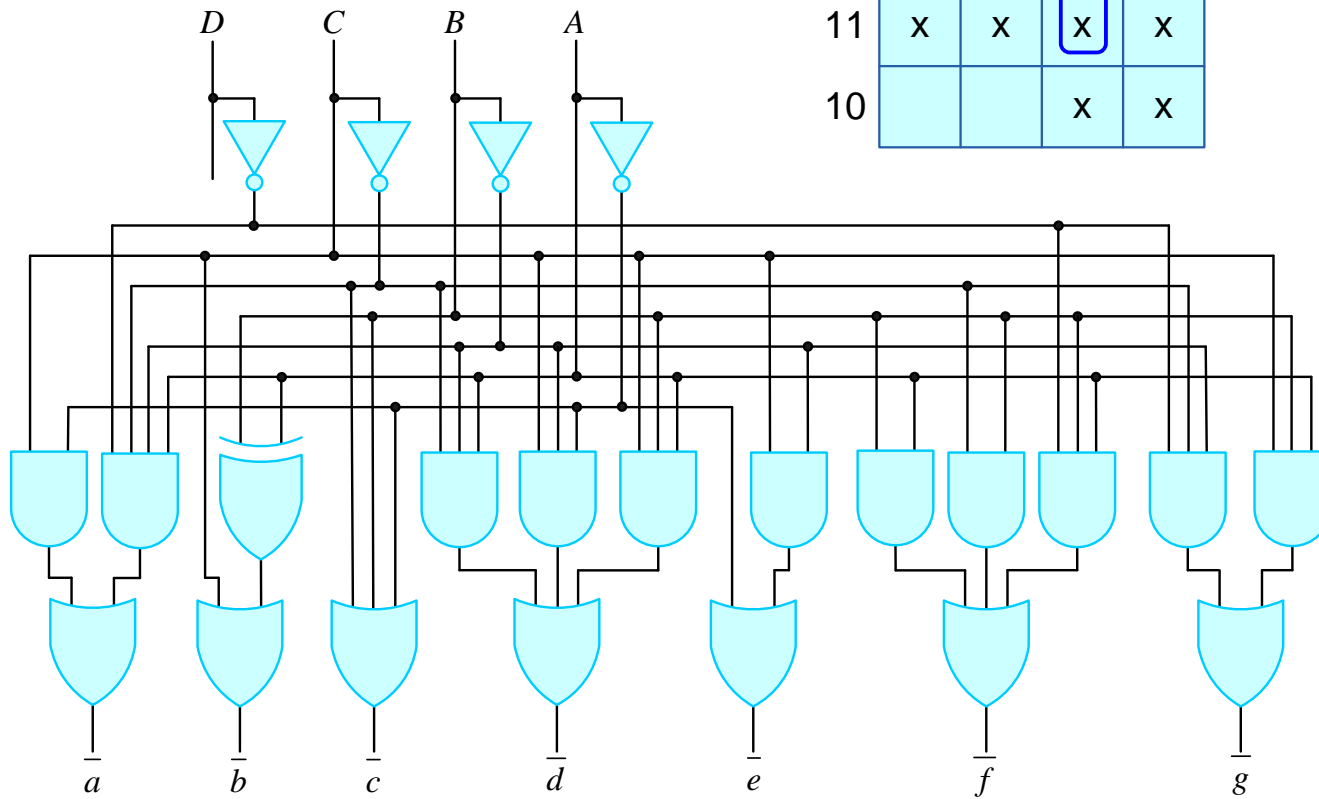
$$\bar{f} = BA + \bar{C}\bar{B} + \bar{D}\bar{C}\bar{A}$$

03 디코더



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

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



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

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

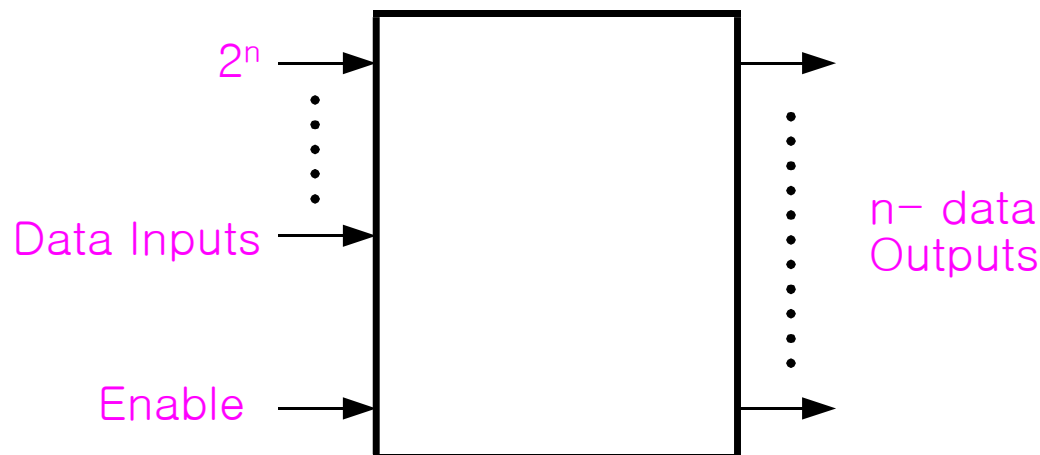
회로도



- ❖ 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.

▶ 인코더 (Encoder)

- 디코더의 역기능
- 입력 2^n 개 => 출력 n 개





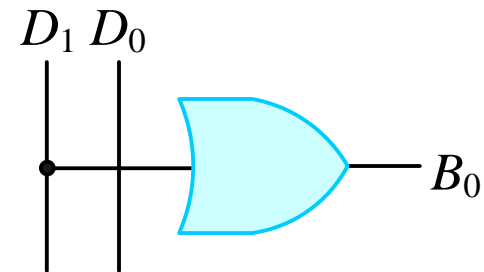
2×1 인코더

❖ 입력의 신호에 따라 1개의 2진 조합으로 출력된다.

입력		출력
D_1	D_0	B_0
0	1	0
1	0	1

$$B_0 = D_1$$

진리표와 논리식



회로도



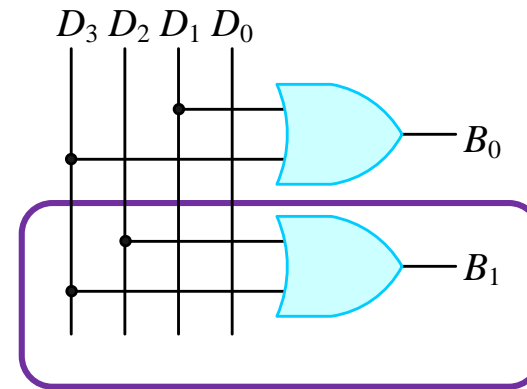
4×2 인코더

❖ 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



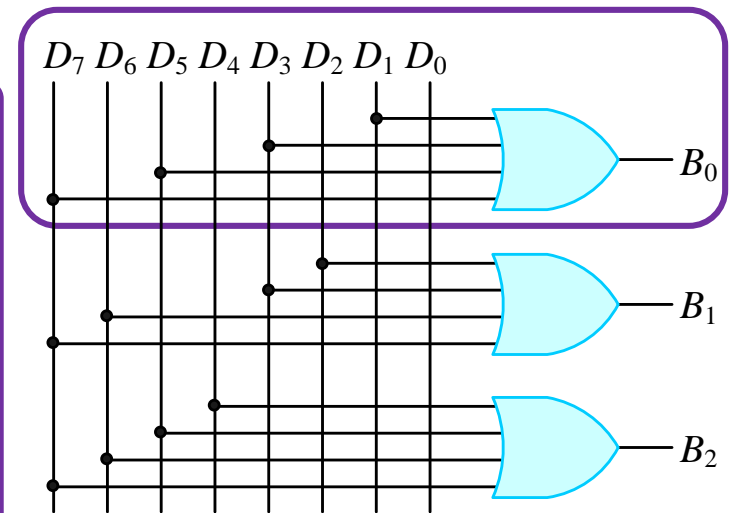
회로도



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



회로도

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

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

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

진리표와 논리식



8×3 우선순위 인코더

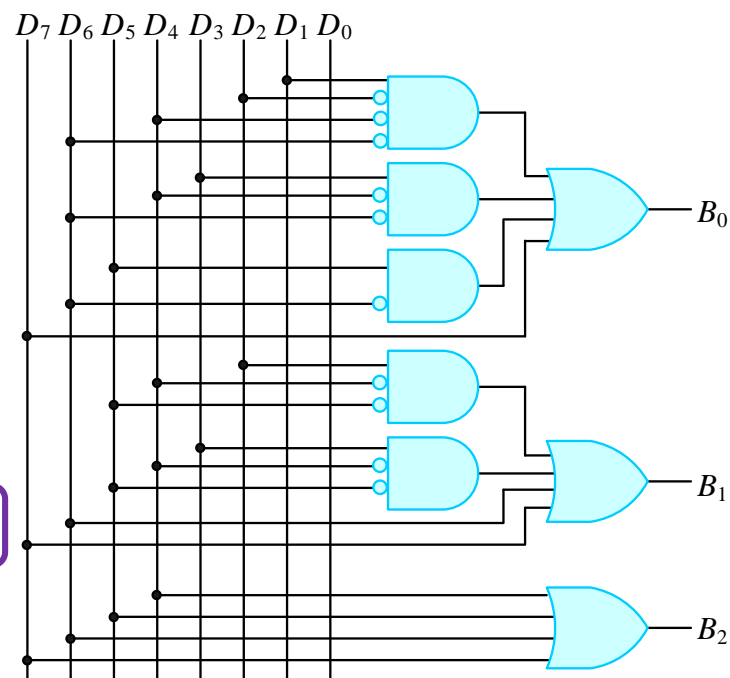
❖ **우선순위 인코더(priority encoder)**는 입력에 우선순위를 정하여 여러 개의 입력이 있을 때 우선순위가 높은 입력값에 해당되는 출력신호를 만들어 내는 회로

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	x	0	0	1
0	0	0	0	0	1	x	x	0	1	0
0	0	0	0	1	x	x	x	0	1	1
0	0	0	1	x	x	x	x	1	0	0
0	0	1	x	x	x	x	x	1	0	1
0	1	x	x	x	x	x	x	1	1	0
1	x	x	x	x	x	x	x	1	1	1

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

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

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



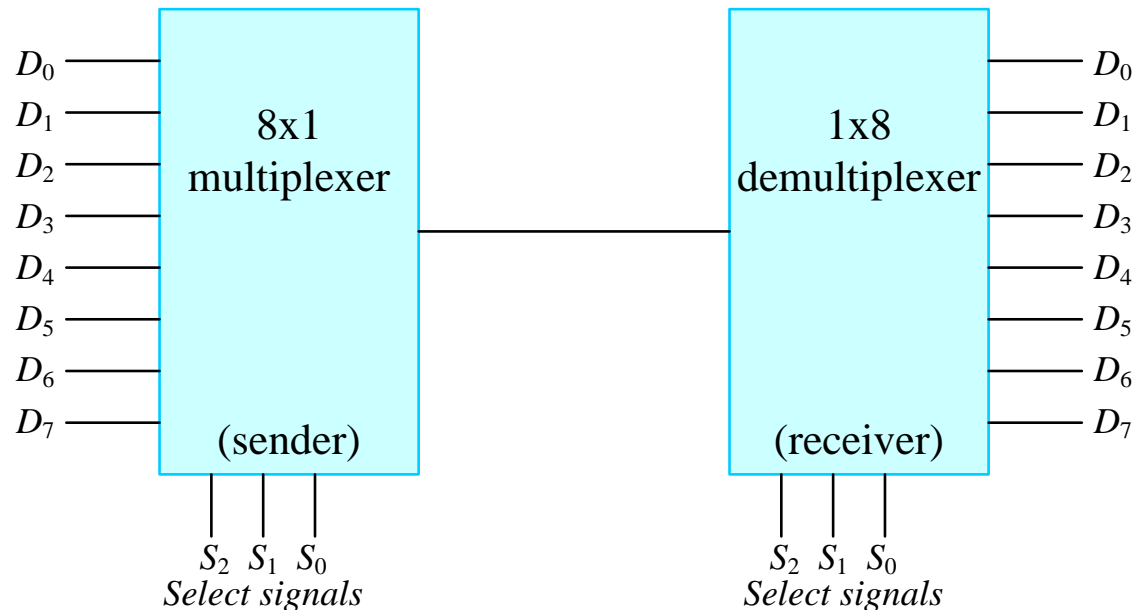
회로도

진리표와 논리식

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.
- ❖ **디멀티플렉서**는 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.





2×1 멀티플렉서

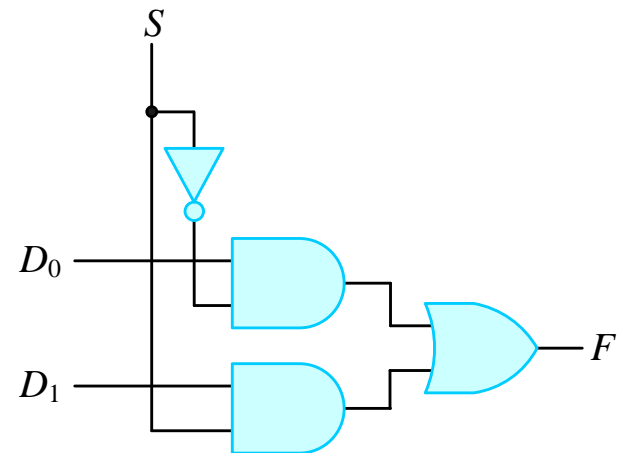
- ❖ $2(=2^1)$ 개의 입력중의 하나를 선택선 S 에 입력된 값에 따라서 출력으로 보내주는 조합회로

선택선	출력
S	F
0	D_0
1	D_1

진리표

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

논리식



회로도



4×1 멀티플렉서

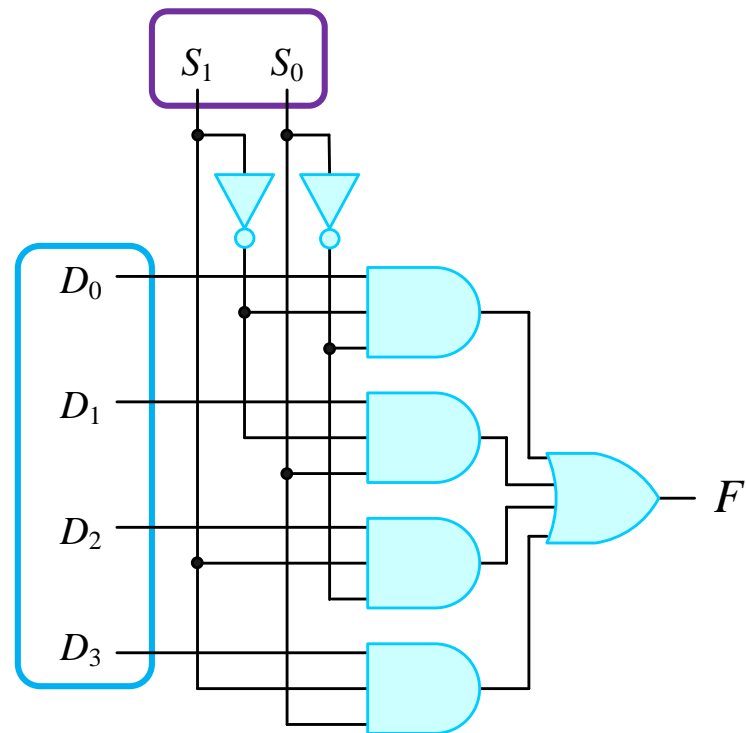
❖ 4($=2^2$)개의 입력중의 하나를 선택선 S_0 와 S_1 에 입력된 값에 따라서 출력으로 보내주는 조합회로

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

진리표

$$F = \overline{S_1} \overline{S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3$$

논리식



회로도



8×1 멀티플렉서

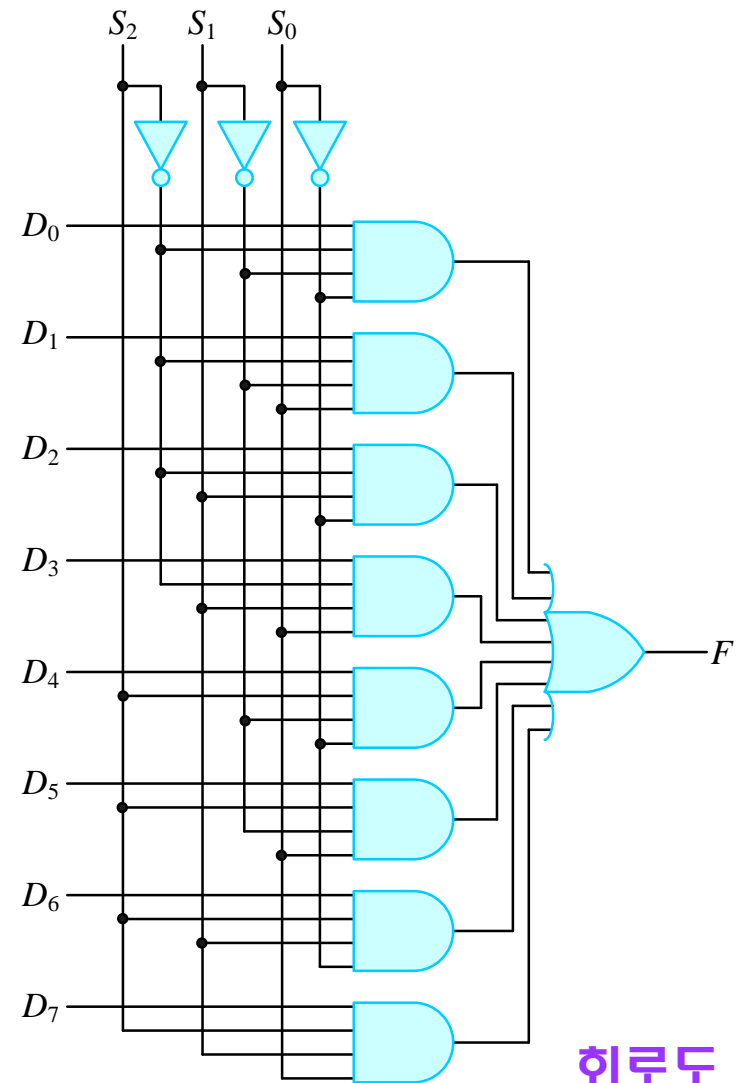
❖ 8($=2^3$)개의 입력중의 하나를 출력으로 보내주는 조합논리회로

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

논리식

$$F = \overline{S_2} \overline{S_1} \overline{S_0} D_0 + \overline{S_2} \overline{S_1} S_0 D_1 + \overline{S_2} S_1 \overline{S_0} D_2 + \overline{S_2} S_1 S_0 D_3 + S_2 \overline{S_1} \overline{S_0} D_4 + S_2 \overline{S_1} S_0 D_5 + S_2 S_1 \overline{S_0} D_6 + S_2 S_1 S_0 D_7$$

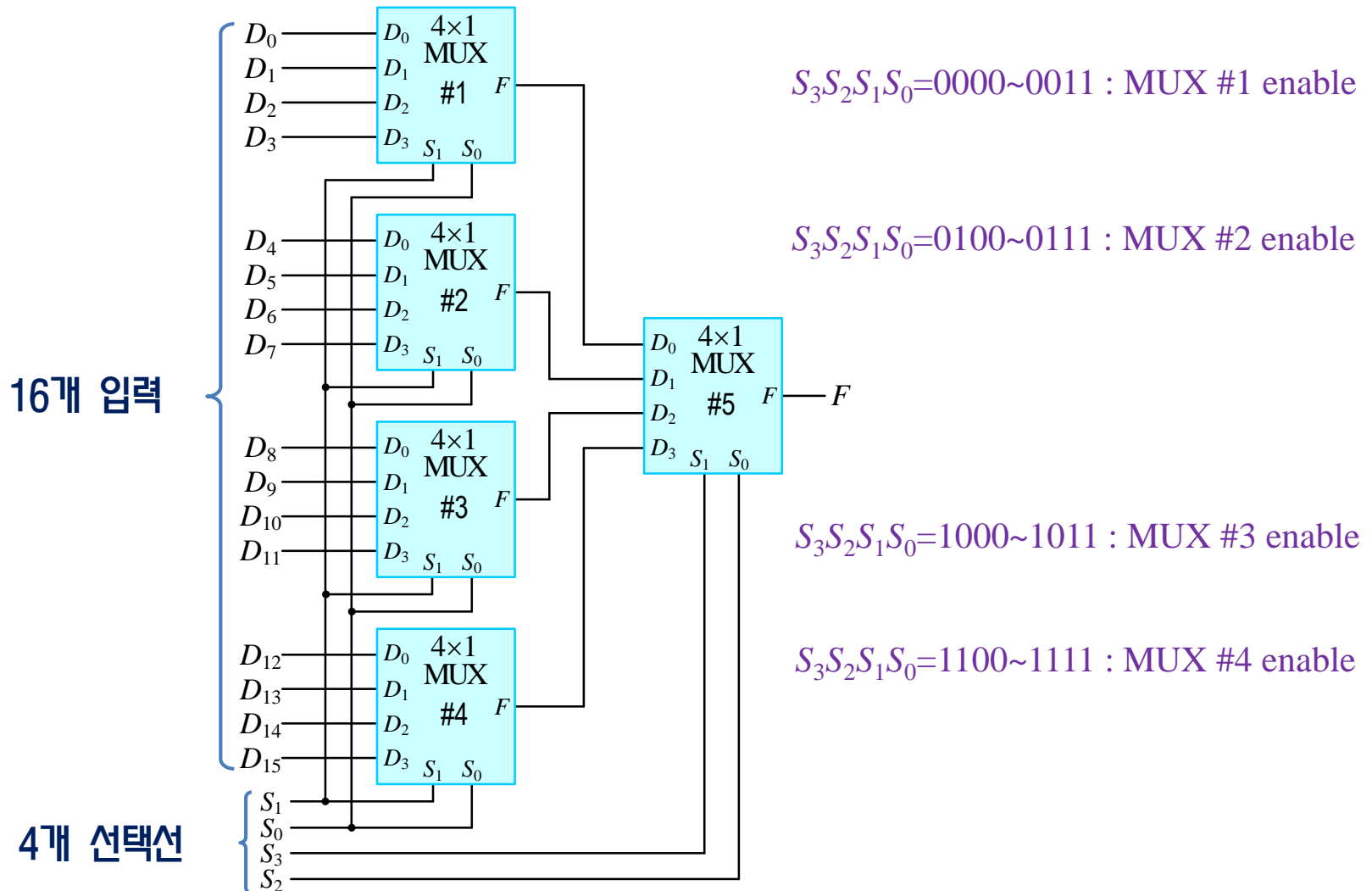


회로도

05 멀티플렉서



□ 4×1 멀티플렉서 5개를 이용한 16×1 멀티플렉서





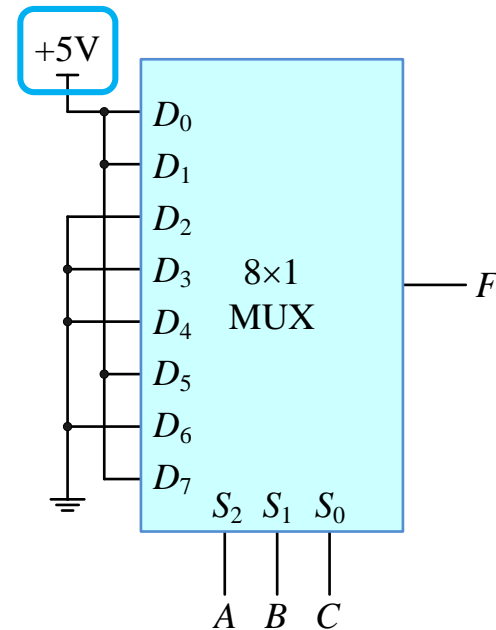
멀티플렉서를 이용한 조합회로 구현

□ $F(A, B, C) = \sum m(0, 1, 5, 7)$ 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 (생략)

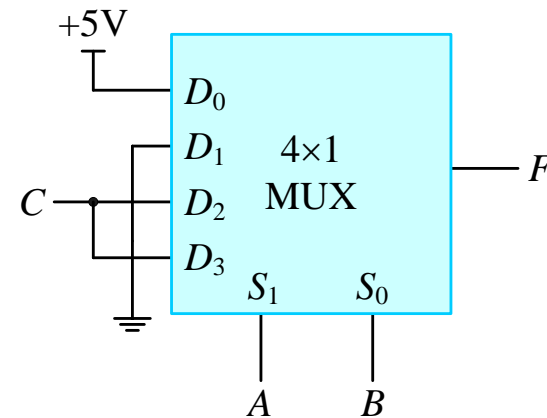


□ $F(A, B, C) = \sum m(0, 1, 5, 7)$ 를 4×1 멀티플렉서로 구현하는 경우

❖ A, B 는 선택선으로 C 는 D_0, D_1, D_2, D_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

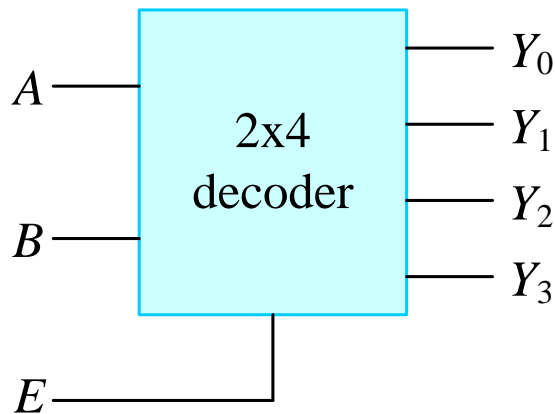


회로도

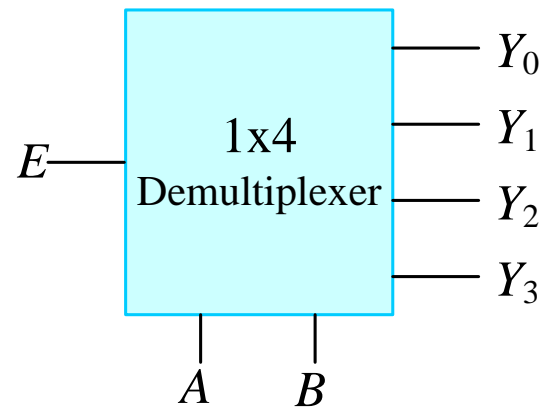
06 디멀티플렉서



- ❖ 1개의 인에이블 입력을 가지고 있는 디코더는 디멀티플렉서로서의 기능을 수행
- ❖ **디멀티플렉서**는 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.



2×4 디코더



1×4 디멀티플렉서



BCD 코드 - 3초과 코드 변환

입력				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

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

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

$$E_3 = B_3 + B_2 B_1 + B_2 B_0$$

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

$$E_2 = \overline{B_2} B_1 + \overline{B_2} B_0 + B_2 \overline{B_1} \overline{B_0}$$

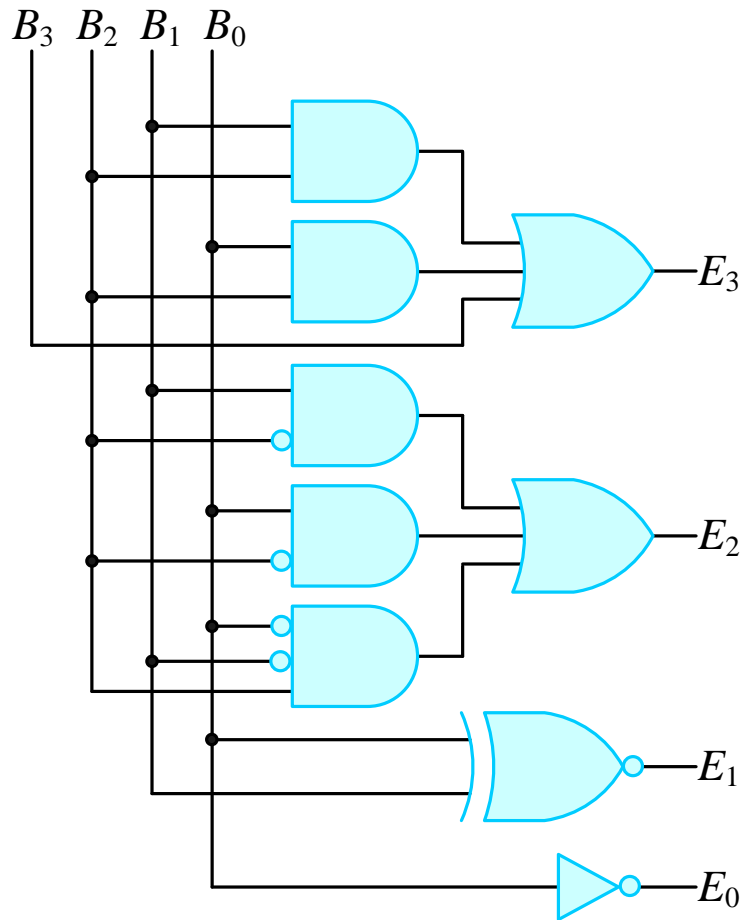
$B_3B_2 \backslash B_1B_0$	00	01	11	10
00	1		1	
01	1		1	
11	x	x	x	x
10	1		x	x

$$E_1 = \overline{B_1} \overline{B_0} + B_1 B_0 = B_1 \oplus B_0$$

<카르노 맵>

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

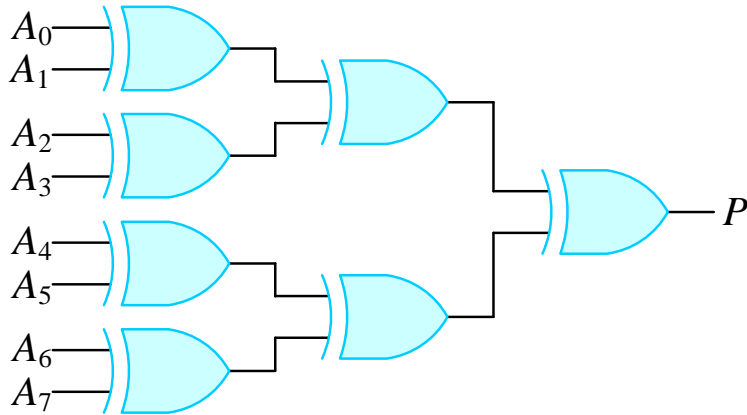
$$E_0 = \overline{B_0}$$



회로도

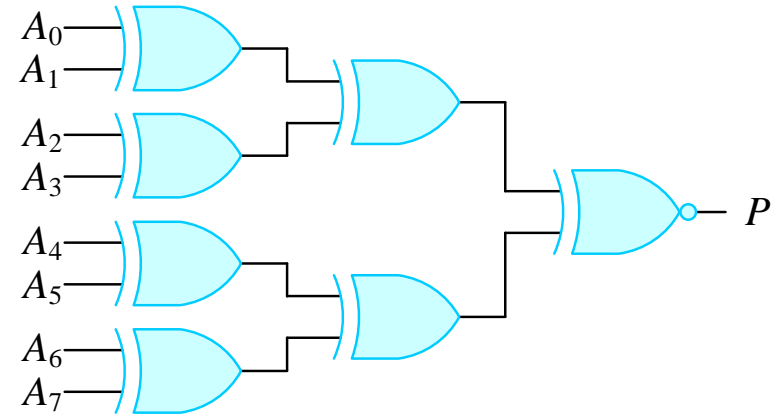


□ 8비트 짝수/홀수 패리티 발생회로



$$P = A_0 \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7$$

짝수 패리티 발생회로



$$P = A_0 \oplus A_1 \oplus A_2 \oplus A_3 \oplus A_4 \oplus A_5 \oplus A_6 \oplus A_7$$

홀수 패리티 발생회로



9. 다음 중 조합논리 회로가 아닌 것은

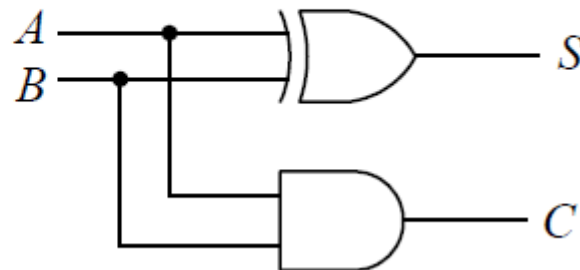
- ㉠ 반가산기 ㉡ 디코더
- ㉢ 멀티플렉서 ㉣ 플립플롭

11. 마이크로프로세서 내에서 산술 연산의 기본 연산은?

- ㉠ 덧셈 ㉡ 뺄셈 ㉢ 곱셈 ㉣ 나눗셈

16. 아래의 논리회로는?

- ㉠ 반가산기
- ㉡ 전가산기
- ㉢ 반감산기
- ㉣ 전감산기





34. 전가산기 회로(full adder)의 구성으로 옳은 것은?

- ㉠ 입력 2개, 출력 4개로 구성
- ㉡ 입력 2개, 출력 3개로 구성
- ㉢ 입력 3개, 출력 2개로 구성
- ㉣ 입력 3개, 출력 3개로 구성

35. 전가산기의 회로 구성은?

- ㉠ 2개의 반가산기와 1개의 OR 게이트로 구성
- ㉡ 2개의 반가산기와 1개의 NOR 게이트로 구성
- ㉢ 2개의 반가산기와 1개의 AND 게이트로 구성
- ㉣ 2개의 반가산기와 1개의 NAND 게이트로 구성



46. 다음은 전가산기이다. $A=1$, $B=1$, $C_i=0$ 일 때 출력 S_0 , C_0 는?
(단, S_0 는 sum, C_0 는 carry이다.)

- ㉠ $S_0 = 0, C_0 = 0$
- ㉡ $S_0 = 1, C_0 = 0$
- ㉢ $S_0 = 0, C_0 = 1$
- ㉣ $S_0 = 1, C_0 = 1$



58. 두 개의 데이터를 비교하는데 적합한 논리 연산은?

- ㉠ AND
- ㉡ OR
- ㉢ NOR
- ㉣ XOR



75. 인코더의 입력선이 8개이면, 출력선은 몇 개가 되는가?

㉠ 1

㉡ 2

㉢ 3

㉣ 4

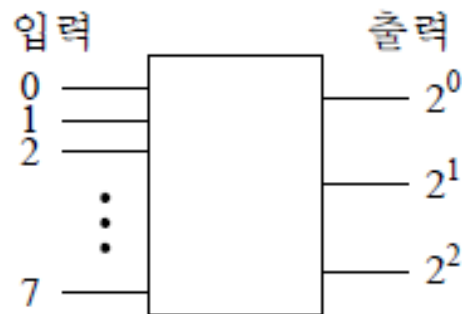
77. 그림과 같이 2^3 개(0~7)의 십진수 입력을 넣었을 때 출력이 2진수 (000~111)로 나오는 회로의 명칭은?

㉠ 디코더 회로

㉡ A/D 변환회로

㉢ D/A 변환회로

㉣ 인코더 회로



80. 부호화된 데이터를 해독하여 정보를 찾아내는 조합논리회로는?

㉠ 인코더

㉡ 디코더

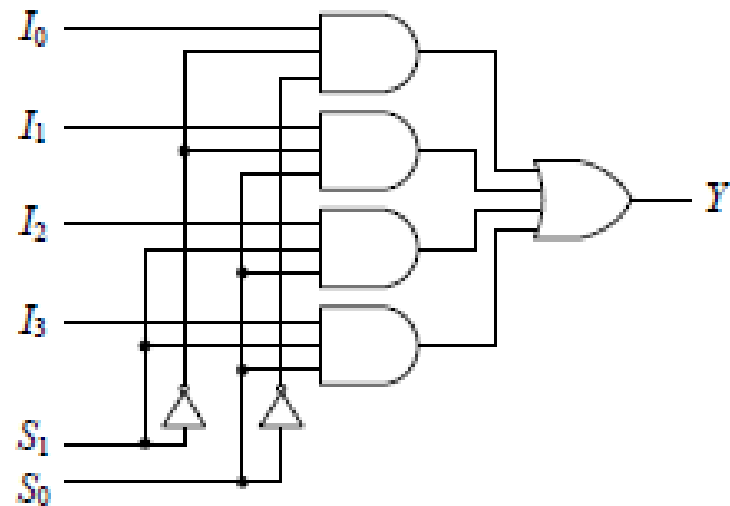
㉢ 디멀티플렉서

㉣ 멀티플렉서



115. 다음 회로의 기능은?

- ㉠ 4×1 multiplexer
- ㉡ 6×1 multiplexer
- ㉢ 4×1 decoder
- ㉣ 6×1 encoder



118. 다음 회로에서 출력 F 로 나올 수 없는 것은?

- ㉠ \overline{B}
- ㉡ $\overline{AB} + A\overline{B}$
- ㉢ AB
- ㉣ $A + B$

