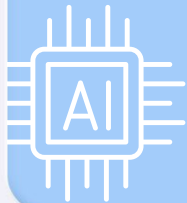




디지털논리회로 [Digital Logic Circuits]

8강.

# 조합논리회로(3)



컴퓨터과학과 강지훈 교수



# 학습 목차

## ▶ 8 강

### 01 MSI를 이용한 조합논리회로(2)

- 디코더
  - 멀티플렉서
  - 디멀티플렉서
- .....

## 8강. 조합논리회로(3)



제5장. 조합논리회로

5.5

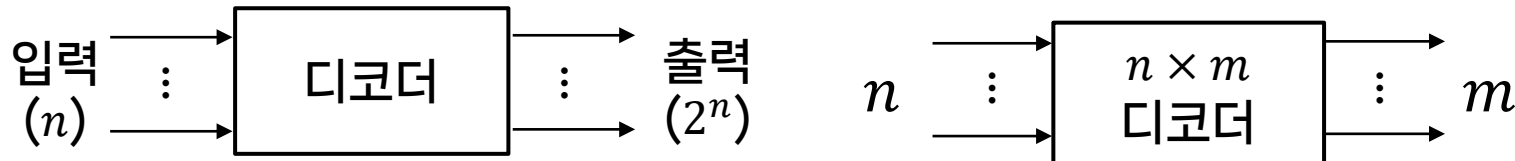
### MSI를 이용한 조합논리회로(2)



## 5.5.1 인코더/디코더

### • 디코더

- 부호화 된 입력을 받아서 부호화 되지 않은 출력을 내보내는 복호화기
- $n$ 비트의 2진 코드를 최대  $2^n$ 개의 서로 다른 정보로 바꾸어 주는 조합논리회로



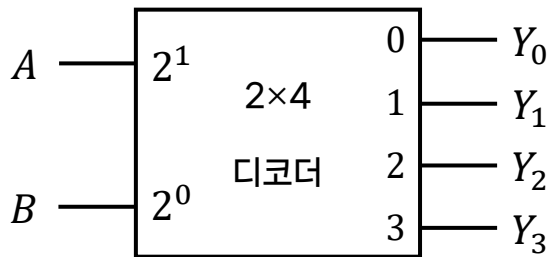


## 5.5.1 인코더/디코더



### • 2×4 디코더

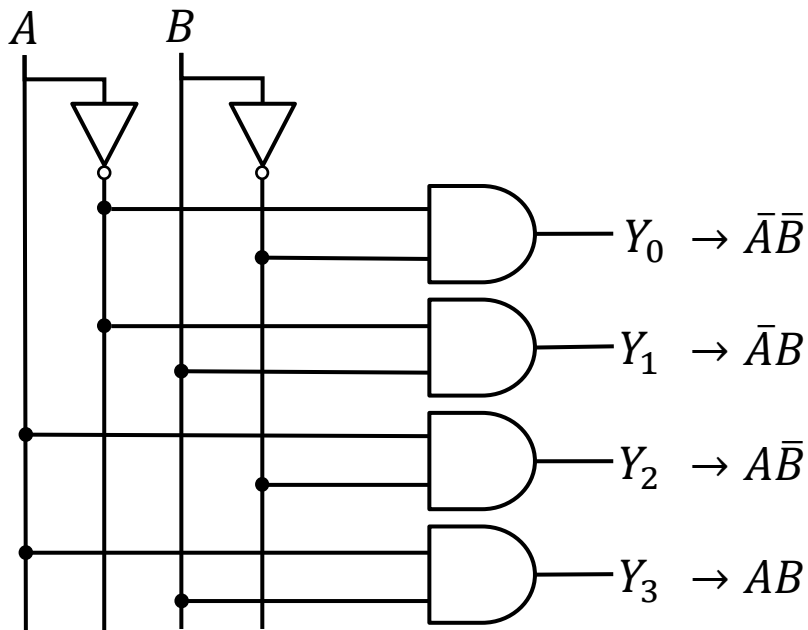
#### 1. 블록도



#### 2. 진리표

입력		출력			
A	B	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

#### 3. 내부 회로도

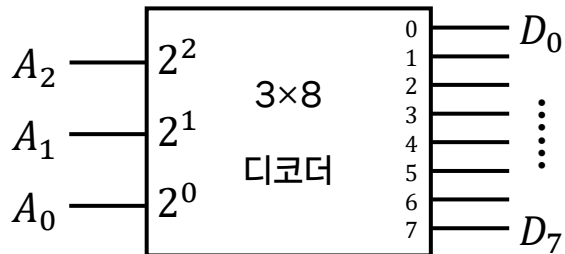




# 5.5.1 인코더/디코더

## • 3×8 디코더

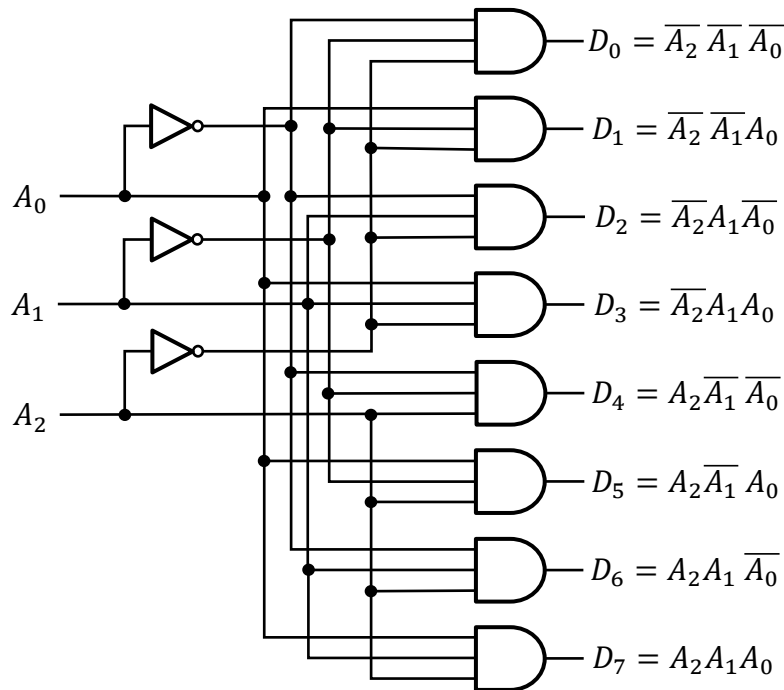
### 1. 블록도



### 2. 진리표

입력			출력							
$A_2$	$A_1$	$A_0$	$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
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

### 3. 내부 회로도



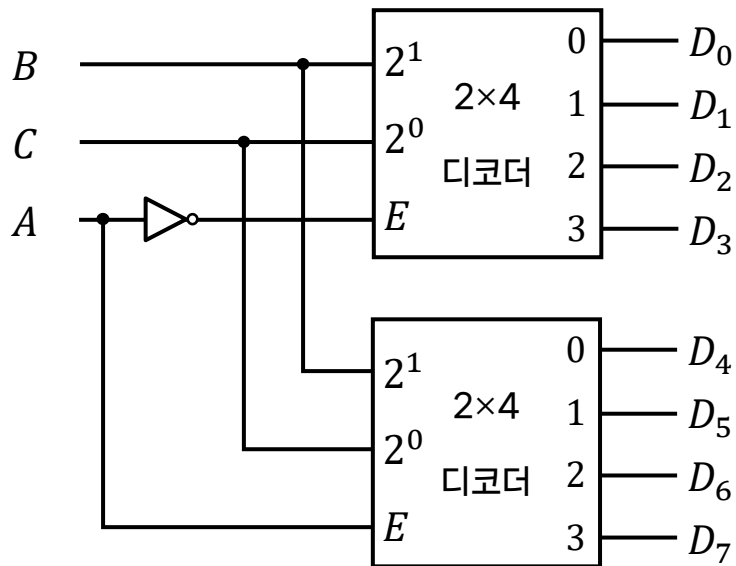


## 5.5.1 인코더/디코더



### • 디코더의 확장

- 작은 디코더 여러 개를 결합하여 필요한 크기의 디코더 구성  
2×4 디코더 2개를 이용한 3×8 디코더



입력은 3비트( $A, B, C$ )로 구성

- $B$ 와  $C$ 는 두 디코더에 모두 연결
- 최상위 비트  $A$ 의 입력에 따라 두 디코더 중 하나만 동작

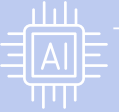
#### ◆ $A$ 가 0이라면

- 위쪽 디코더는  $B, C$ 에 따라  $D_0 \sim D_3$  중 하나의 최소항 선택

#### ◆ $A$ 가 1이라면

- 아래쪽 디코더는  $B, C$ 에 따라  $D_4 \sim D_7$  중 하나의 최소항 선택

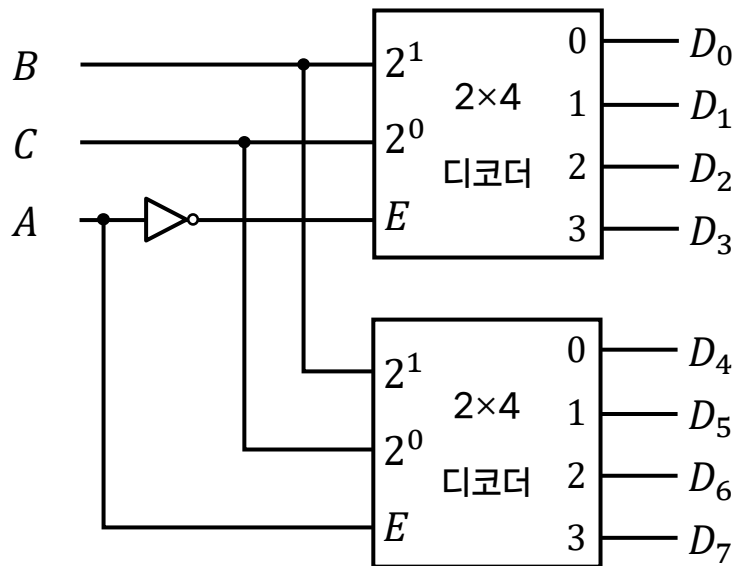




## 5.5.1 인코더/디코더

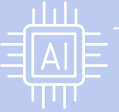


### • 3×8 디코더



입력			출력							
A	B	C	$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
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



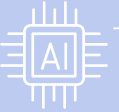


## 5.5.1 인코더/디코더



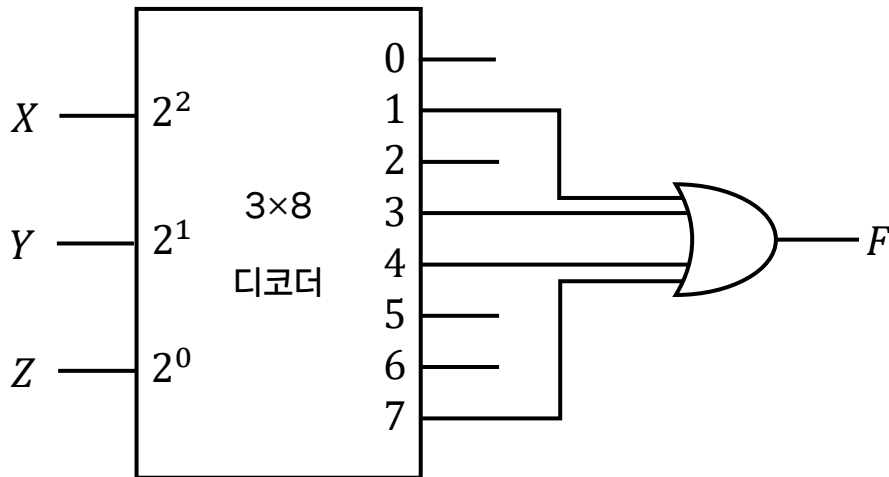
### • 디코더를 이용한 부울함수의 구현

- 디코더는  $n$ 개의 입력변수에 대한  $2^n$ 개의 최소항을 만듦
- 따라서, 부울함수는 최소항의 합으로 표현되며, 디코더를 이용하면 부울함수를 구현할 수 있음
- $n$ 개의 입력과  $m$ 개의 출력을 가진 조합논리회로는  $n \times 2^n$  디코더와  $m$ 개의 OR 게이트로 만들 수 있음



## 5.5.1 인코더/디코더

- $F(X, Y, Z) = \Sigma m(1, 3, 4, 7)$ 를 디코더를 이용하여 구현
  - 해당 부울함수는 3개의 입력과 8개의 최소항을 나타냄. 따라서,  $3 \times 8$  디코더가 필요함
  - 주어진 부울함수에서 최소항은 4개(1, 3, 4, 7)이며, 이를 OR 게이트로 연결





## 5.5.1 인코더/디코더



### • 디코더를 활용한 BCD-3초과 코드 변환기 구현(1)

- 디코더와 OR 게이트로 구현할 때에는 부울함수를 최소항의 합 형태로 표현
  - 최소항의 합 형태는 진리표나 카르노 도표를 이용해서 구함

입력 BCD 코드					출력 3-초과 코드			
10진숫자	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	1	0	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	0	0
6	0	1	1	0	1	0	0	1
7	0	1	1	1	1	0	1	0
8	1	0	0	0	1	0	1	1
9	1	0	0	1	1	1	0	0

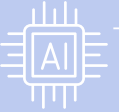
$$W(A, B, C, D) = \Sigma m(5, 6, 7, 8, 9)$$

$$X(A, B, C, D) = \Sigma m(1, 2, 3, 4, 9)$$

$$Y(A, B, C, D) = \Sigma m(0, 3, 4, 7, 8)$$

$$Z(A, B, C, D) = \Sigma m(0, 2, 4, 6, 8)$$





## 5.5.1 인코더/디코더



### • 디코더를 활용한 BCD-3초과 코드 변환기 구현(2)

#### • 디코더와 OR 게이트를 이용한 논리회로도 작성

##### ▪ 디코더 선택

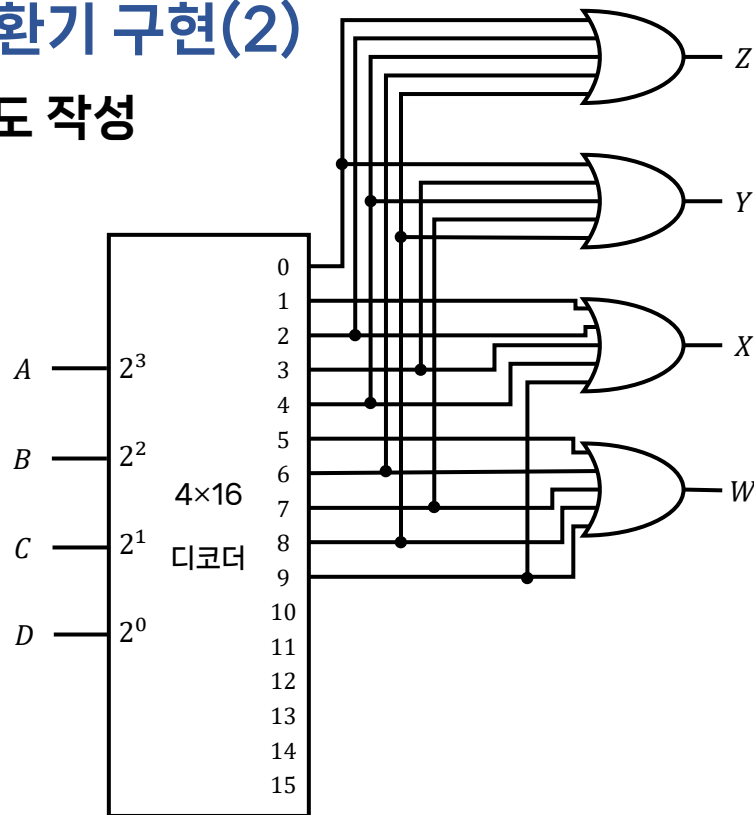
입력변수에 대한 최소항을 모두 발생시킬 수 있는  
4×16 디코더가 필요함

$$W(A, B, C, D) = \Sigma m(5, 6, 7, 8, 9)$$

$$X(A, B, C, D) = \Sigma m(1, 2, 3, 4, 9)$$

$$Y(A, B, C, D) = \Sigma m(0, 3, 4, 7, 8)$$

$$Z(A, B, C, D) = \Sigma m(0, 2, 4, 6, 8)$$





## 5.5.2 멀티플렉서/디멀티플렉서

### • 멀티플렉서(Multiplexer, MUX)

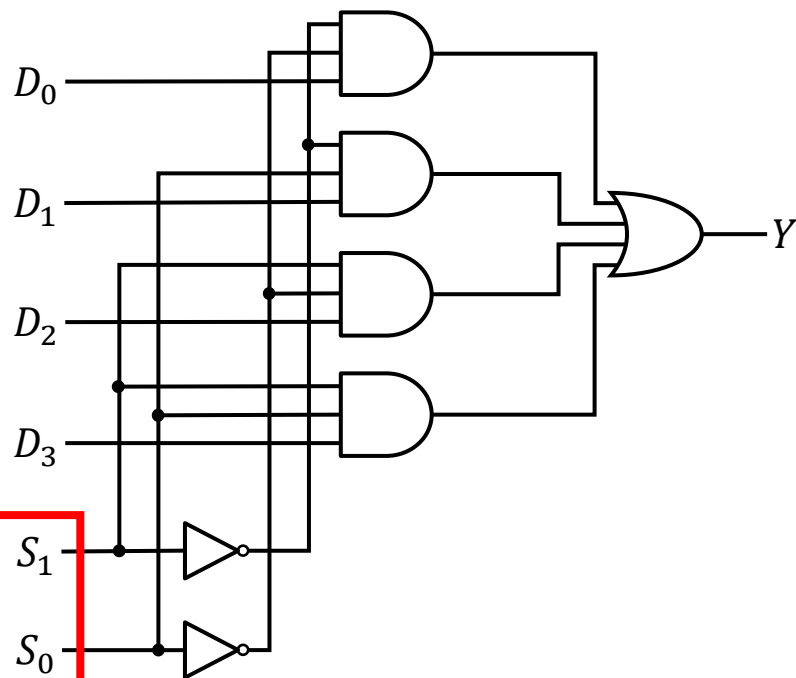
- 여러 개의 입력 신호 중에 하나를 선택하여 단일 출력으로 보내는 조합논리회로
- 특정 입력 신호를 선택하기 위해 선택변수 사용
- 만일  $2^n$ 개의 입력 신호 중에서 특정 입력을 선택하기 위해서는  $n$ 개의 선택변수가 있어야 함
- $n$ 개의 선택변수의 조합에 따라 특정 입력 신호가 선택됨
- 데이터 선택기(Data Selector) 라고도 하며, 약어로 MUX로 표시



## 5.5.2 멀티플렉서/디멀티플렉서



### • 4×1 멀티플렉서(1)



선택변수

$S_1, S_0$  이 1, 0 이라면

- $D_2$ 가 입력되는 AND 게이트의 두 입력은 모두 1이 되어 출력이 1이 됨
- 나머지 3개의 AND 게이트의 출력은 0이 됨
- 즉, OR 게이트로 입력되는 4개의 입력 중  $D_2$ 가 입력된 AND 게이트의 출력만 1이 됨
- 따라서 OR 게이트의 출력은  $D_2$ 의 값이 됨





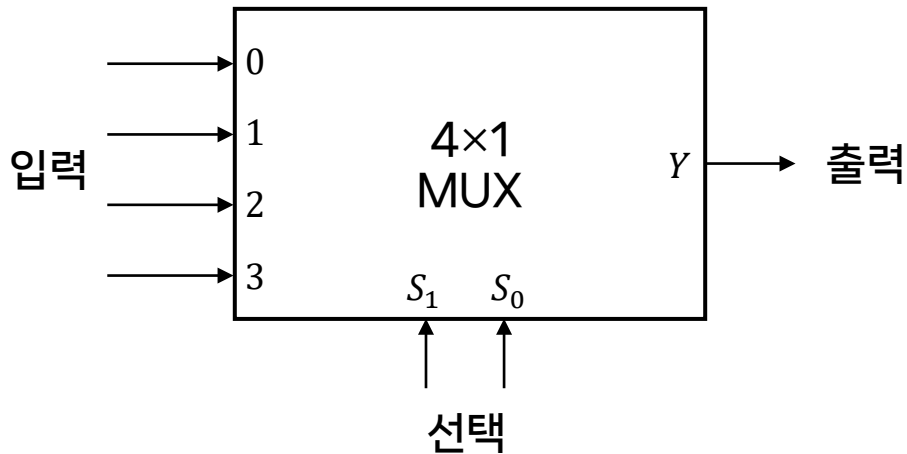
## 5.5.2 멀티플렉서/디멀티플렉서

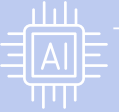
### • 4×1 멀티플렉서(2)

함수표

$S_1$	$S_0$	$Y$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

블럭도

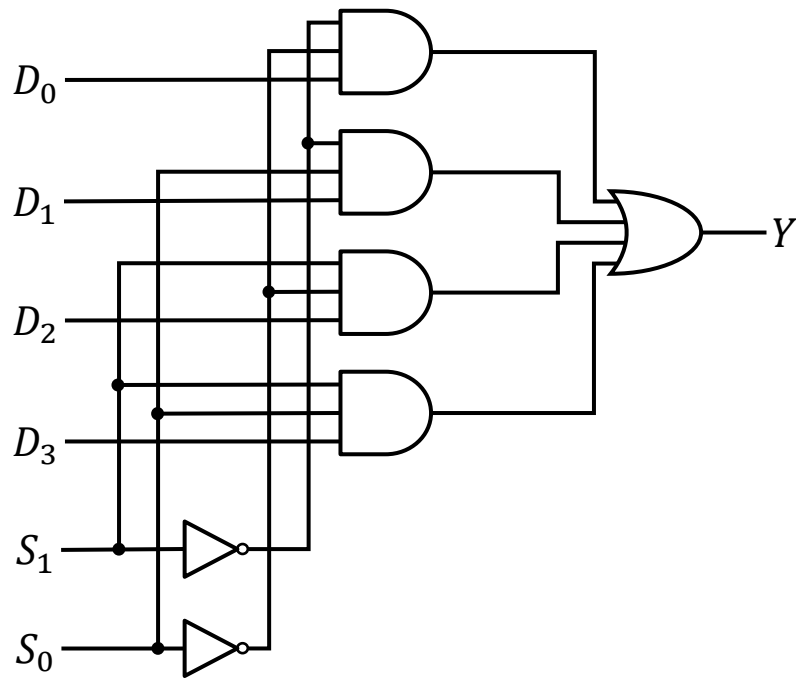
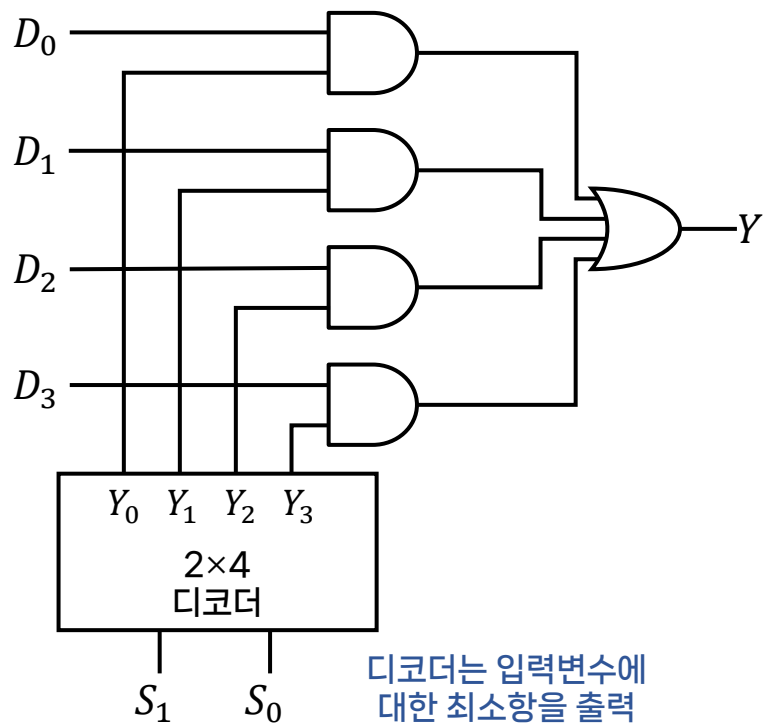




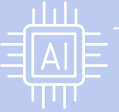
## 5.5.2 멀티플렉서/디멀티플렉서



### • 멀티플렉서에서 디코더 기능







## 5.5.2 멀티플렉서/디멀티플렉서

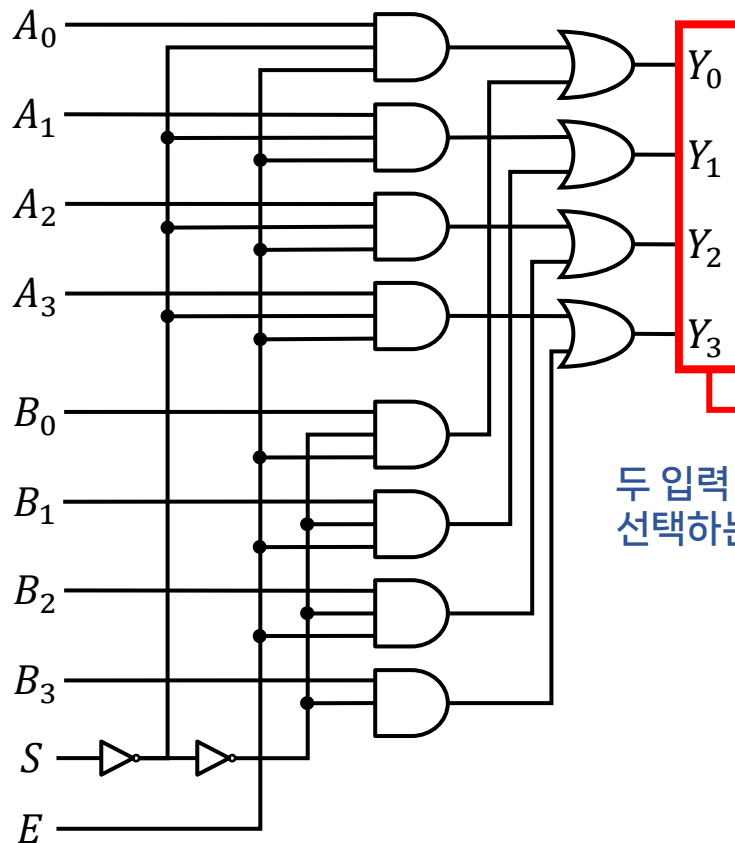


### • 구동입력을 가진 멀티플렉서

4개의  $2 \times 1$  MUX의  
논리회로도

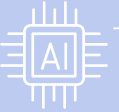
함수표

$E$	$S$	출력	$Y$
0	X	모두	0
1	0	선택	A
1	1	선택	B



두 입력  $A, B$  중 하나를  
선택하는 동작 수행





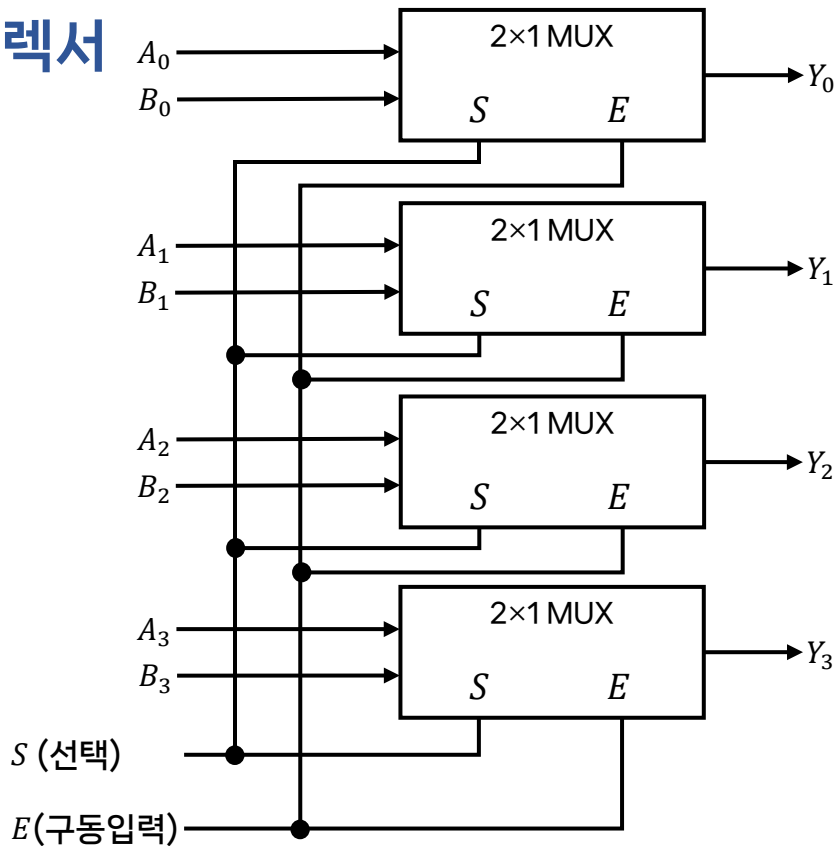
## 5.5.2 멀티플렉서/디멀티플렉서



### • 구동입력을 가진 멀티플렉서

4개의  $2 \times 1$  MUX 블록도

- 4비트로 구성된 두 수  $A, B$  중 하나를 출력하는 경우

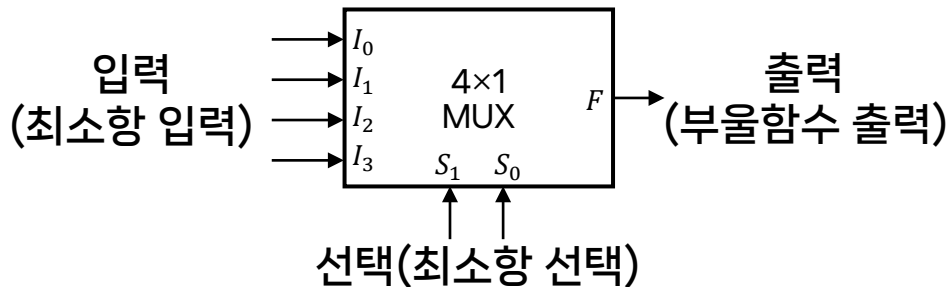


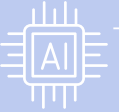


## 5.5.2 멀티플렉서/디멀티플렉서

### • 멀티플렉서를 이용한 부울함수 구현

- 멀티플렉서는 OR 게이트를 가진 디코더와 같은 기능을 수행함
- 따라서, 멀티플렉서의 선택선으로 입력변수의 최소항을 선택할 수 있으며, 입력선을 이용해 해당 최소항을 입력할 수 있음
- $n$ 개의 선택 입력과  $2^n$ 개의 데이터 입력을 가진 멀티플렉서를 이용하면  $n + 1$ 개의 변수를 가진 부울함수를 구현할 수 있음



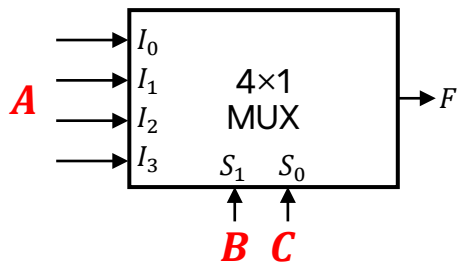


## 5.5.2 멀티플렉서/디멀티플렉서

### • 멀티플렉서를 이용한 부울함수 구현과정

#### 1. 최소항의 변수를 MUX의 입력선과 선택선에 연결

- 변수가  $A, B, C, \dots$  순서라면,  $A$ 는 입력선에 그 다음 변수인  $B, C, \dots$ 를 선택선의 높은 순서부터 차례대로 연결



- $n$ 개의 선택 입력과  $2^n$ 개의 데이터 입력을 가진 멀티플렉서를 이용하면  $n + 1$ 개의 변수를 가진 부울함수를 구현할 수 있음  
→ 3개의 변수를 가진 부울함수를 구현하려면,  
2개의 선택입력과  
 $2^2$ 개의 데이터 입력을 가진 멀티플렉서를 사용

#### 2. 입력선에 연결되는 변수 A에 대한 최소항을 결정

- 이 단계의 작업에서  $I_0, I_1, I_2, I_3, \dots$ 를 결정. 이는 구현표에 의해 결정됨

#### 3. 완성된 구현표에 의해 MUX에 입력단을 연결하면 부울함수를 구현할 수 있음

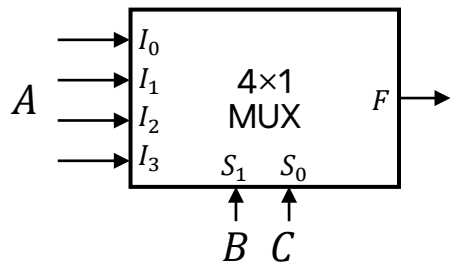




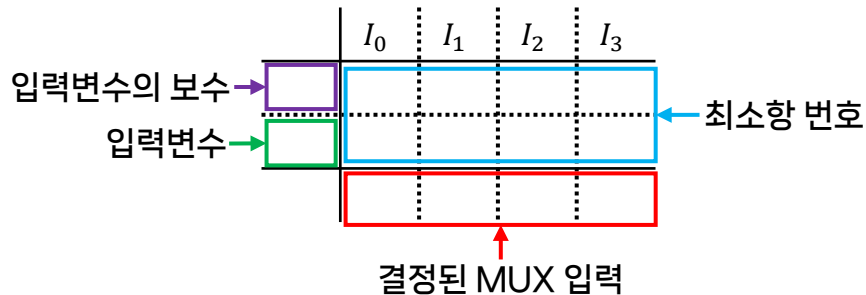
## 5.5.2 멀티플렉서/디멀티플렉서

### • $F(A, B, C) = \Sigma m(1, 2, 6, 7)$ 의 구현(1)

#### 1. 최소항의 변수를 MUX에 할당



#### 2. 구현표 작성 ( $I_0, I_1, I_2, I_3$ 를 결정하는 단계)



#### 3. 구현표 완성을 위해 진리표 이용

최소항	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

X	Y	Z	최소항	
			항	표시
0	0	0	$\bar{X}\bar{Y}\bar{Z}$	$m_0$
0	0	1	$\bar{X}\bar{Y}Z$	$m_1$
0	1	0	$\bar{X}YZ$	$m_2$
0	1	1	$\bar{X}YZ$	$m_3$
1	0	0	$X\bar{Y}\bar{Z}$	$m_4$
1	0	1	$X\bar{Y}Z$	$m_5$
1	1	0	$XY\bar{Z}$	$m_6$
1	1	1	$XYZ$	$m_7$

	$I_0$	$I_1$	$I_2$	$I_3$
$\bar{A}$	0	1	2	3
A	4	5	6	7

A가 가장 왼쪽 변수이기 때문에  
보수형과 정상형이 반반씩 존재함



## 5.5.2 멀티플렉서/디멀티플렉서

### • $F(A, B, C) = \Sigma m(1, 2, 6, 7)$ 의 구현(2)

#### 4. MUX의 입력을 결정(1)

	$I_0$	$I_1$	$I_2$	$I_3$
$\bar{A}$	0	1	2	3
$A$	4	5	6	7

- 구현표에 작성된 최소항의 목록에 진리표에서 주어진 함수의 최소항을 표시한 뒤 각 열을 규칙에 따라 MUX의 입력으로 결정

#### ※ MUX의 입력 결정 규칙

- 동일한 열에 속한 두 최소항이 표시되어 있지 않다면, 해당 열의 MUX 입력은 0.
- 동일한 열에 속한 두 최소항이 모두 표시되어 있다면, 해당 열의 MUX 입력은 1.
- 아래쪽 최소항만 표시되어 있다면, 해당 열의 MUX 입력은  $A$
- 위쪽 최소항만 표시되어 있다면, 해당 열의 MUX 입력은  $\bar{A}$

#### 5. MUX의 입력을 결정(2)

	$I_0$	$I_1$	$I_2$	$I_3$
$\bar{A}$	0	1	2	3
$A$	4	5	6	7
	0	$\bar{A}$	1	$A$





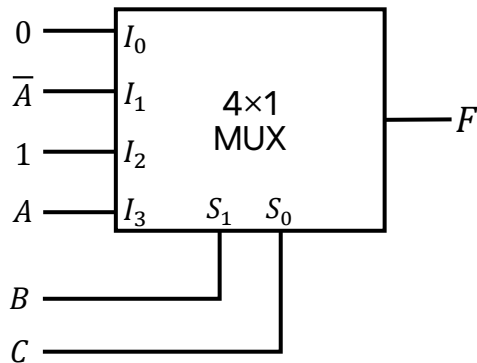
## 5.5.2 멀티플렉서/디멀티플렉서

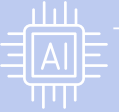


### • $F(A, B, C) = \Sigma m(1, 2, 6, 7)$ 의 구현(3)

6. 구현표를 이용하여 부울함수를 구현

	$I_0$	$I_1$	$I_2$	$I_3$
$\bar{A}$	0	1	2	3
$A$	4	5	6	7
	0	$\bar{A}$	1	$A$



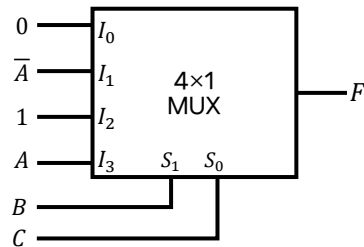


## 5.5.2 멀티플렉서/디멀티플렉서



### • 멀티플렉서를 이용하여 구현된 부울함수의 검증

#### • 선택선에 따라 출력값을 확인



최소항	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	0
6	1	1	0	1
7	1	1	1	1

$BC = 00$ : MUX의  $I_0$ 이 선택됨.  $I_0 = 0$  이기 때문에  $F = 0$ ,  $A$ 값과 무관하게  $BC = 00$ 일 때  $m_0 = \bar{A}\bar{B}\bar{C}$ ,  $m_4 = \bar{A}\bar{B}\bar{C}$ 는 모두 0을 출력

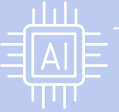
$BC = 01$ : MUX의  $I_1$ 이 선택됨.  $I_1 = \bar{A}$  이기 때문에  $m_1 = \bar{A}\bar{B}C$  면  $F = 1$ ,  $m_5 = A\bar{B}C$  면  $F = 0$  을 출력

$BC = 10$ : MUX의  $I_2$ 이 선택됨.  $I_2 = 1$  이기 때문에  $F = 1$ 이 되며,  $A$ 값과 무관하게 출력이 1이므로  $m_2 = \bar{A}B\bar{C}$ ,  $m_6 = AB\bar{C}$ 는 모두 1 출력

$BC = 11$ : MUX의  $I_3$ 이 선택됨.  $I_3 = A$  이기 때문에  $m_3 = \bar{A}BC$  면  $A = 0$ 이기 때문에  $F = 0$  출력,  $m_7 = ABC$  면  $A = 1$ 이기 때문에 출력  $F = 1$







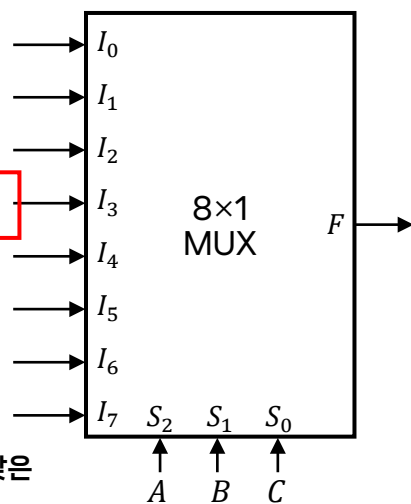
## 5.5.2 멀티플렉서/디멀티플렉서



### • $F(A, B, C, D) = \sum m(1, 3, 6, 9, 11, 12, 14, 15)$ 의 구현(1)

1. 최소항의 변수를 MUX에 할당

2. 구현표 작성

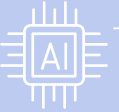


이번에는 가장 낮은  
자리 변수를  
입력으로 사용

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$\bar{D}$	0	2	4	6	8	10	12	14
$D$	1	3	5	7	9	11	13	15

최소항	입력				출력
	$A$	$B$	$C$	$D$	$F$
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1





## 5.5.2 멀티플렉서/디멀티플렉서



•  $F(A, B, C, D) = \Sigma m(1, 3, 6, 9, 11, 12, 14, 15)$ 의 구현(2)

3. 구현표 완성을 위해 진리표 이용

최소항	입력				출력
	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	1
15	1	1	1	1	1

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$\overline{D}$	0	2	4	6	8	10	12	14
D	1	3	5	7	9	11	13	15
	D	D	0	$\overline{D}$	D	D	$\overline{D}$	1





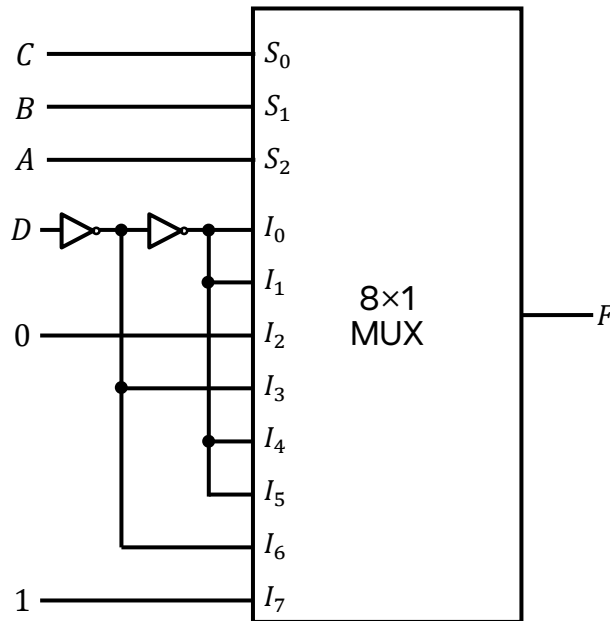
## 5.5.2 멀티플렉서/디멀티플렉서

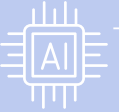


•  $F(A, B, C, D) = \Sigma m(1, 3, 6, 9, 11, 12, 14, 15)$ 의 구현(3)

4. 구현표를 이용하여 부울함수를 구현

	$I_0$	$I_1$	$I_2$	$I_3$	$I_4$	$I_5$	$I_6$	$I_7$
$\bar{D}$	0	2	4	6	8	10	12	14
$D$	1	3	5	7	9	11	13	15
	$D$	$D$	0	$\bar{D}$	$D$	$D$	$\bar{D}$	1





## 5.5.2 멀티플렉서/디멀티플렉서

### • 디멀티플렉서(demultiplexer)

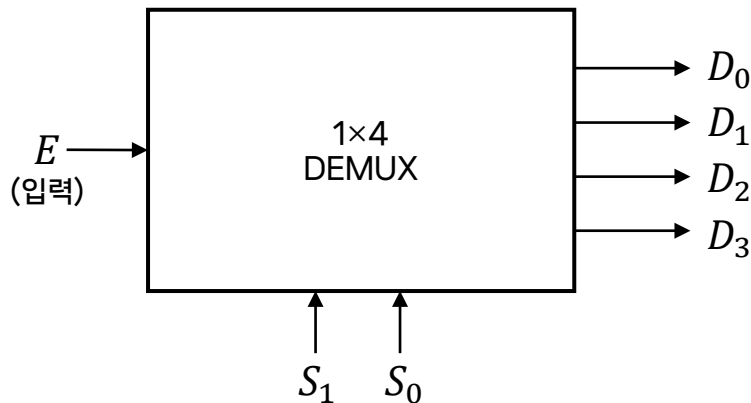
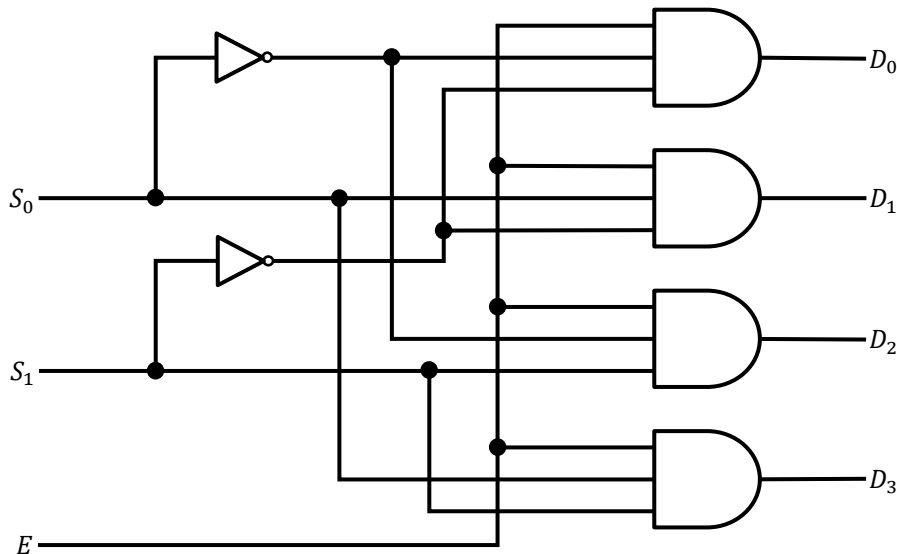
- 멀티플렉서와 반대되는 연산을 수행하는 조합논리회로
- 데이터 분배기(data distributor)라고도 불림
- 한 개의 입력선으로부터 정보를 받아 이를  $2^n$ 개의 출력선 중 하나로 내보냄
  - 출력선의 제어는  $n$ 개의 선택입력 조합으로 제어됨
- 디멀티플렉서는 약어로 DEMUX로 표시함



## 5.5.2 멀티플렉서/디멀티플렉서



### • 논리회로도와 블록도



- 데이터 입력  $E$ 는 두 선택신호  $S_1, S_0$ 에 의해 출력선을 선택함

$S_1, S_0$ 가 10이라면, 출력  $D_2$ 만 입력  $E$ 를 출력함.

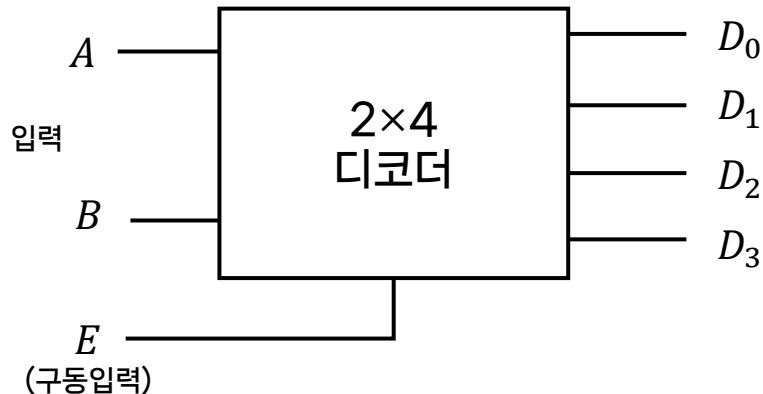
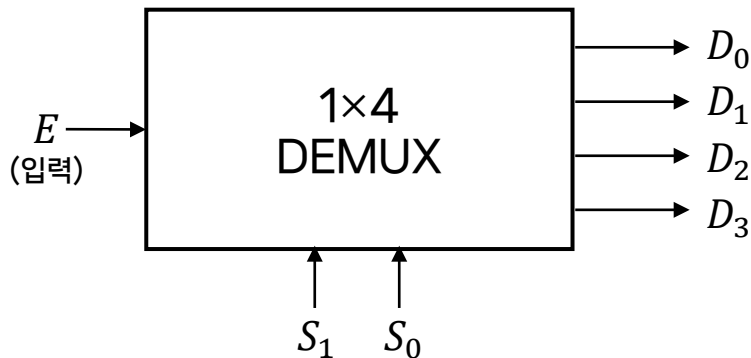
즉, 동작 특성이 디코더와 유사하게 작동함 (입력 신호에 따라 1이 출력될 출력선을 결정)



## 5.5.2 멀티플렉서/디멀티플렉서



### • 디멀티플렉서와 디코더와의 유사관계



입력			출력			
$E$	$A$	$B$	$D_0$	$D_1$	$D_2$	$D_3$
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

- DEMUX와 디코더는 동일한 진리표를 가짐
- DEMUX는  $E$ 가 데이터 입력,  $A, B$ 는 선택 입력
- 디코더는  $A, B$ 가 데이터 입력,  $E$ 가 구동 입력





## • MSI 장치 활용의 특성

- 회로 설계 시 MSI 장치는 논리회로의 게이트나 배선을 자동으로 최적화 해주는 마법 같은 도구는 아님
- 반도체 설계 단계에서 최적화되어 최적화 수준과 기능이 검증되어 미리 만들어진 장치를 사용하는 것임
  - 이를 통해 직접 테스트하고 검증을 하기 위한 노력과 시간을 감소시킬 수 있음
- MSI 장치를 활용할 때 비용적인 장점도 얻을 수 있음
  - 직접 게이트를 배치할 때의 개발비용과 MSI 소자를 구매하는 비용 사이에서 적절한 타협점을 찾아야 함



# 내용 정리

Summary

Contents



## 8강 | 조합논리회로(3)



디지털 +  
논리회로



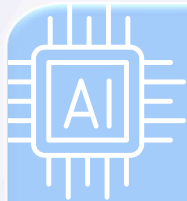
### 01 MSI를 이용한 조합논리회로

- 디코더
- 멀티플렉서
- 디멀티플렉서

Digital Logic Circuits







다음시간에는

9강.

# 순서논리회로(1)