

# 컴퓨터 구조

## 조합 논리 회로

고려대학교 세종캠퍼스 인공지능사이버보안학과  
구 자 훈

# 목차

---

## ❖ 학습 목표

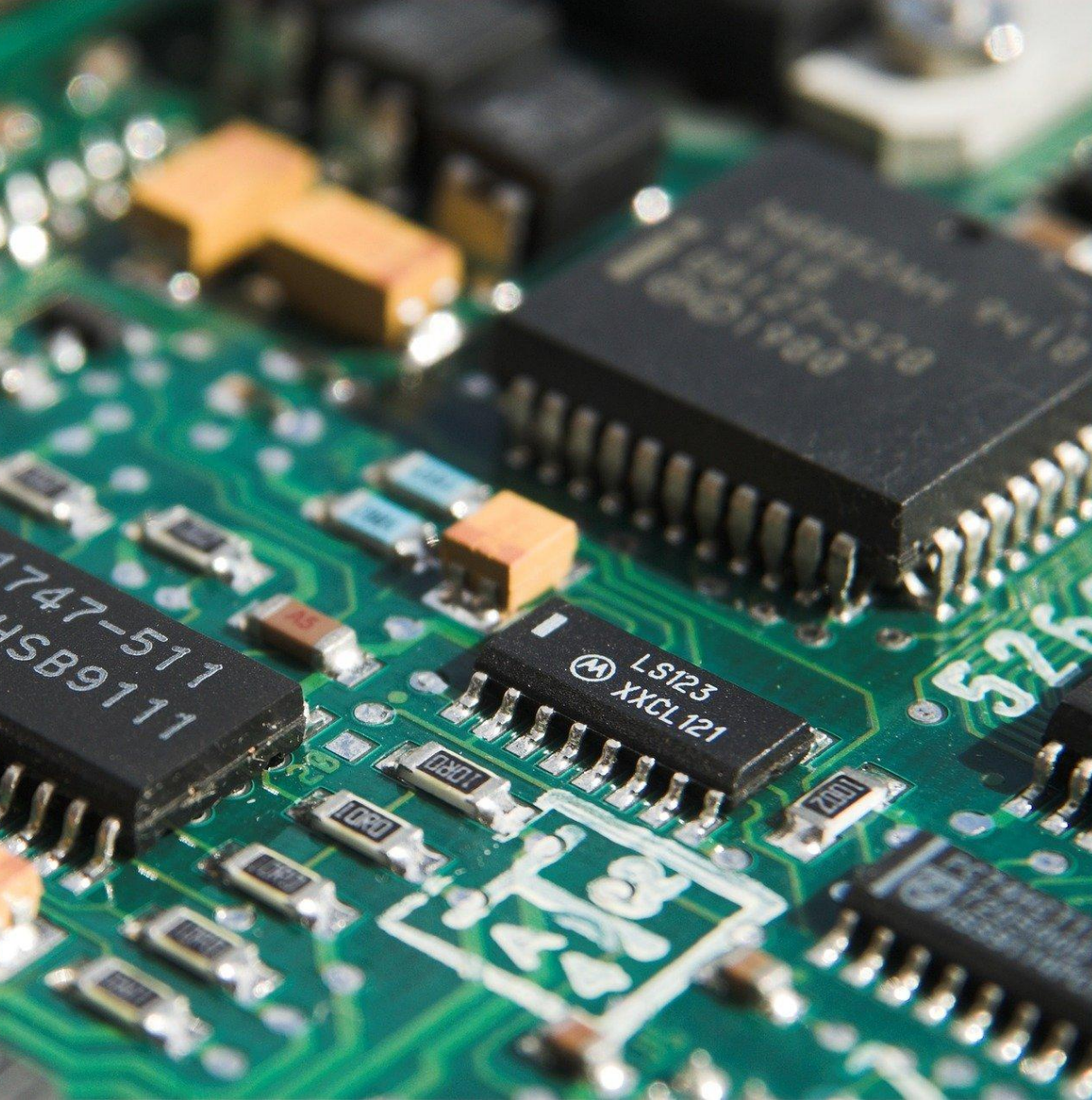
- 조합 논리회로의 동작 표현 방법인 논리식, 진리표, 논리회로도 해석할 수 있다.
- 가산기, 비교기, 패리티 발생/검사기, 인코더/디코더, 멀티플렉서/디멀티플렉서의 기능을 설명할 수 있다.

## ❖ Part 1.

- 조합 논리회로 소개
- 산술 조합 논리회로
- 조합 논리회로 빌딩블록

## ❖ Part 2.

- 퀴즈
- 요약



# Part 1

- 4.1 조합 논리회로 소개
- 4.2 산술 조합 논리회로
- 4.3 조합 논리회로 빌딩블록

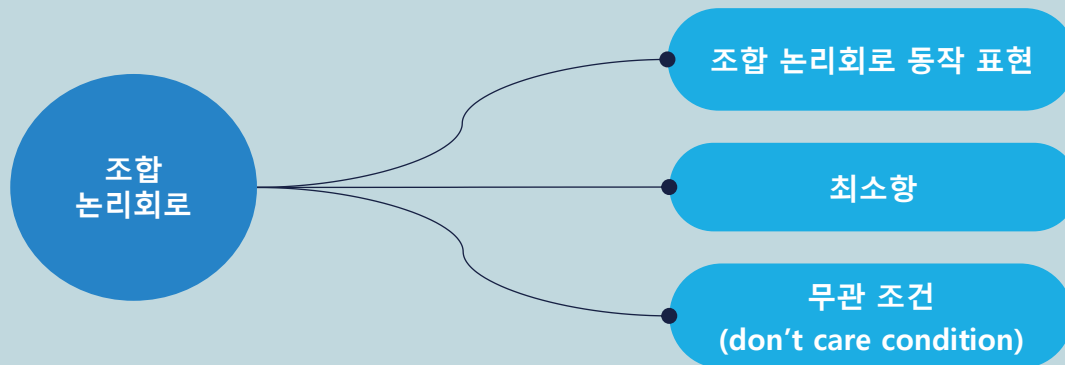
# 4.1 조합 논리회로 소개

## 학습 목표

- 조합 논리 회로의 동작을 표현하는 방법 이해
- 진리표, 논리식, 논리회로도 표현 방법 숙달

## 논리 회로 (logic circuit)

- 논리게이트로 만든 논리적 회로
- 조합 논리회로 (combinational logic circuit) : 입력 조합에 따라 출력 결정
- 순차 논리회로 (sequential logic circuit) : 기억 소자를 포함한 회로



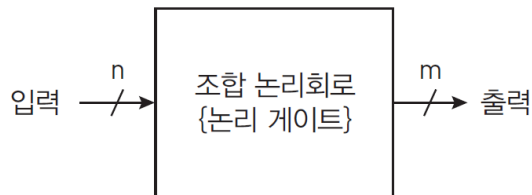
# 01 조합 논리회로의 동작 표현

## 조합 논리회로 (combinational circuit)

- 입력신호의 조합에 따라 출력을 결정하는 회로

## 동작 표현 방법 (입력/출력은 0 또는 1의 값만 가짐)

- 진리표 = 동작 특성표 = 특성표 (characteristic table)
  - 진리표 : 입력신호 조합에 따라 출력신호를 표로 표현
- 논리식(logic equation) = 부울식(Bool equation) = 논리함수(logic function)
  - 논리식 : 논리 변수를 부울 대수의 AND, OR, NOT과 같은 논리 연산자로 조합한 공식
- 논리 회로도(logic diagram)
  - 논리 회로도 : 논리식에 해당하는 논리 게이트 기호를 연결해서 표현한 그림



< n x m 조합 논리회로 >

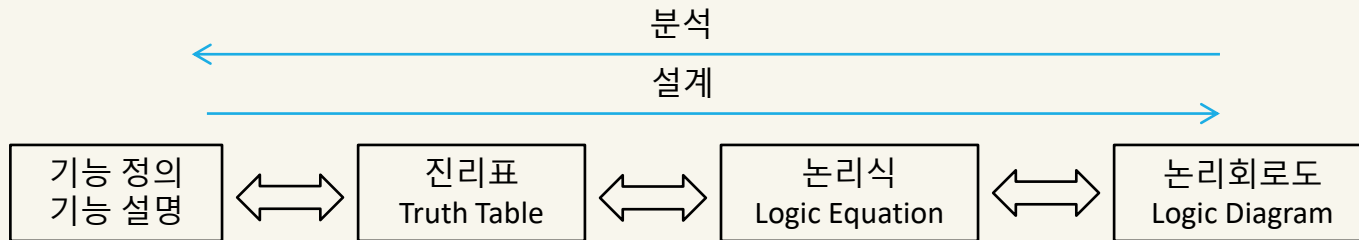
일련 번호	입력신호				출력신호			
	$X_n$	$X_{n-1}$	...	$X_1$	$Y_m$	$Y_{m-1}$	...	$Y_1$
0	0	0	0	0				
1	0	0	0	1				
2	0	0	1	0				
...								
$2^n-1$	1	1	1	1				

< m x n 조합 논리회로의 진리표 >

# 01 조합 논리회로

## 조합 논리회로의 동작 표현 방법

### 논리 회로의 설계 과정



### 수업에서의 설명 순서

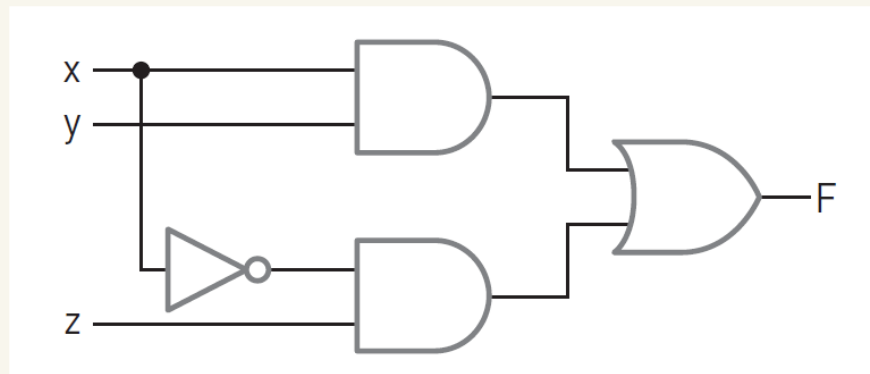
- (1) 논리식 → 논리 회로도
- (2) 논리 회로도 → 논리식
- (3) 논리식 → 진리표
- (4) 진리표 → 논리식 : 최소항 (또는 최대항)에 의한 방법

# 01 조합 논리회로 : 논리식 → 논리회로도

## 논리식의 연산 우선 순위

- 괄호 ( ) > NOT ( ' ) > AND > OR

## 논리 회로도 그리는 방법



<  $F = x \cdot y + x' \cdot z$  의 논리회로도 >

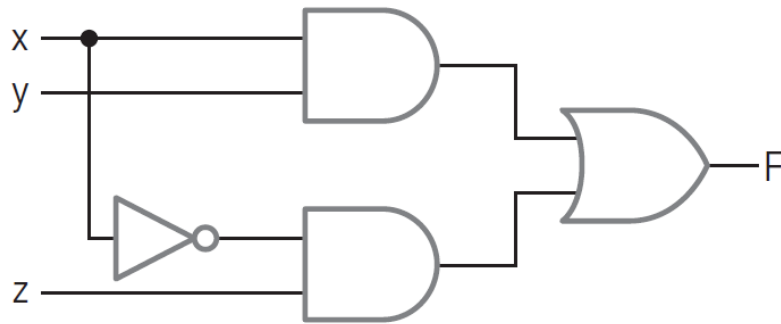
1. 입력 신호를 왼편에, 출력 신호를 오른편에 배치한다.
2. 연산 우선순위에 따라 입력에서 출력 방향으로 게이트를 배치하고 입력과 출력을 연결한다.

# 01 조합 논리회로 : 논리 회로도 → 논리식

## 논리식 구하는 방법

1. 입력, 출력, 게이트의 출력에 논리 변수 할당
2. 입력단부터 출력단 방향으로 게이트의 논리식 적기
3. 출력단에 포함된 게이트의 출력을 입력 신호의 논리식으로 대체

## 논리식 구하기 예제



○ 1, 2 단계

$$g1 = x'$$

$$g3 = g1 \cdot z$$

$$g2 = x \cdot y$$

$$F = g2 + g3$$

○ 3 단계

$$F = g2 + g3$$

$$= x \cdot y + g1 \cdot z$$

$$= x \cdot y + x' \cdot z$$



# 01 조합 논리회로 : 논리식 → 진리표

## 논리식으로 진리표를 구하는 방법

1. 모든 입력신호의 조합을 2진수 순서로 표에 배치한다.
2. 논리식의 연산 우선순위에 따라 세부 항에 대한 출력을 구한다.

$$F = x \cdot y + x' \cdot z$$

x	y	z	x'	$x \cdot y$	$x' \cdot z$	F
0	0	0	1	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	1	0	1
1	1	1	0	1	0	1

## 02 조합 논리회로 최소항 (minterm)

### 진리표로부터 논리식을 구하는 두 가지 방법

- 논리곱의 합(sum of products)으로 표현하는 방법
  - 진리표에서 출력이 1인 논리항을 찾아 논리식을 구함
- 논리합의 곱(product of sums)으로 표현하는 방법
  - 진리표에서 출력이 0인 논리항을 찾아 논리식을 구함

서로 쌍대(Dual)

### 논리곱과 최소항

- 논리곱: 논리변수들이 AND 연산으로 묶인 항.
- 최소항(minterm): 논리변수들이 모두 참여하는 논리곱.
- 예) 입력 변수가  $x, y, z$ 일 때
  - 논리곱의 예:  $yz, x'y'z', xy'z'$  등
  - 최소항의 예:  $x'y'z', x'yz, xy'z$  등

### 논리곱의 합 (sum of products)

- $F1(x,y,z) = x + y \cdot z$
- $F2(x,y,z) = x \cdot y + x' \cdot z$
- $F3(x,y,z) = x' \cdot y' \cdot z' + x \cdot y \cdot z$

## 02 조합 논리회로 최소항 (minterm)

입력 신호의 조합 중에서 출력이 하나만 1인 논리함수

- 출력이 1개만 1이라면 논리곱으로 모든 항을 사용한 최소항을 유추할 수 있다.

입력 {x, y, z}에 대한 최소항

입력			최소항 (기호/논리식)							
x	y	z	$m_0$ $x'y'z'$	$m_1$ $x'y'z$	$m_2$ $x'yz'$	$m_3$ $x'yz$	$m_4$ $xy'z'$	$m_5$ $xy'z$	$m_6$ $xyz'$	$m_7$ $xyz$
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

## 02 조합 논리회로 최소항의 합

### 논리식 구하기

- 1단계: 진리표에 대한 논리식을 최소항의 합으로 표현
  - $F = 1$ 이어야 논리곱의 합으로 결과를 유추할 수 있음
- 2단계: 간소화 (부울대수, 인접항 찾기)

### 진리표에 대한 논리식

- 최소항의 합

$$F(x,y,z) = x'y'z + x'yz + xyz' + xyz$$

- 간소화

$$F(x,y,z) = x'z + xy$$

x	y	z	F	$m_1$ $x'y'z$	$m_3$ $x'yz$	$m_6$ $xyz'$	$m_7$ $xyz$
0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0
0	1	0	0	0	0	0	0
0	1	1	1	0	1	0	0
1	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0
1	1	0	1	0	0	1	0
1	1	1	1	0	0	0	1

## 03 조합 논리회로 무관 조건

### 무관 조건(don't care condition)

- 특정한 입력 신호 조합이 절대로 발생하지 않거나  
입력 신호에 대한 출력이 회로의 동작에 영향을 주지 않는 조건
- 진리표에 x 또는 d로 표현
- 무관조건이 많을 수록 인접항도 많아질 가능성이 커지므로
  - 논리식이 간단해짐
  - 논리회로가 간단해짐

# 03 조합 논리회로

## 무관조건

### 인에이블(enable)기능이 있는 디코더의 진리표

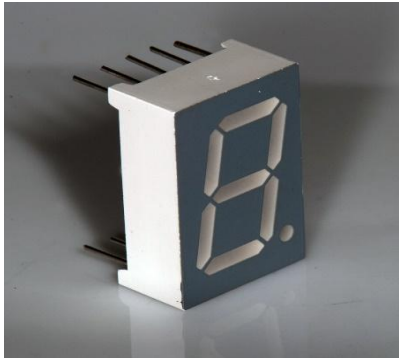
- 이 장치의 입력은 3비트 / 출력은 4비트
- 입력신호 enable은 장치의 동작 여부를 결정
- Enable = 0이면 장치는 동작하지 않음 / Enable = 1이면 장치가 동작함
- Enable = 0이면 입력신호  $D_1, D_2$ 는 회로 출력에 영향을 주지 않음  
(=이 두 비트의 입력은 입력 무관조건에 해당됨)

입력			출력				
Enable	$D_1$	$D_0$	$Y_3$	$Y_2$	$Y_1$	$Y_0$	Valid
0	x	x	x	x	x	x	0
1	0	0	0	0	0	1	1
1	0	1	0	0	1	0	1
1	1	0	0	1	0	0	1
1	1	1	1	0	0	0	1

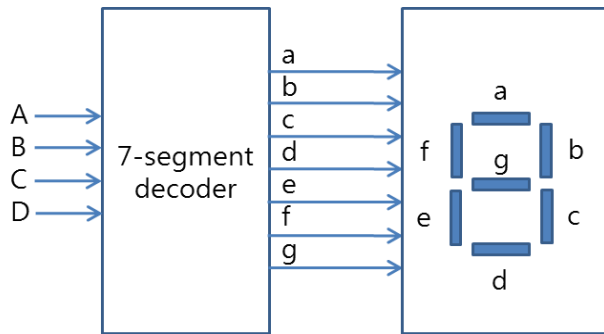
< 무관 조건을 포함한 디코더의 진리표 >

# 03 조합 논리회로

## 7-Segment 표시장치



<7-세그먼트 장비>  
출처: 위키백과



<7-세그먼트 표시>

### 8421 BCD 7-세그먼트 표시장치 디코더

- LED 7개가 8자 모양으로 배치
- LED를 켜고 끄면서 숫자 0~9까지 출력하는 장치

### 입력 신호

- 4개의 입력 신호가 있음 A,B,C,D (A는 MSB)
- 입력 신호는 BCD숫자이므로 0000 부터 1001 까지 10개의 조합 사용
- 1010 ~ 1111까지 6개의 조합은 사용되지 않음

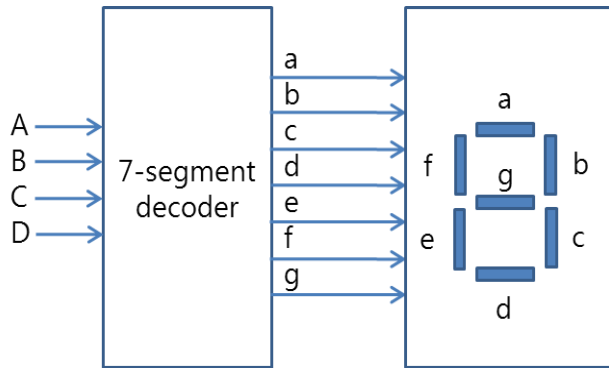
### 출력 신호

- 7개의 출력 신호 a,b,c,d,e,f,g가 존재
- 각 LED는 1이면 불이 켜지고 0이면 LED가 꺼짐
- 입력 1010 ~ 1111은 출력 무관 조건

# 03 조합 논리회로

## 7-Segment 표시장치

### 8421 BCD 7-세그먼트 표시장치 디코더



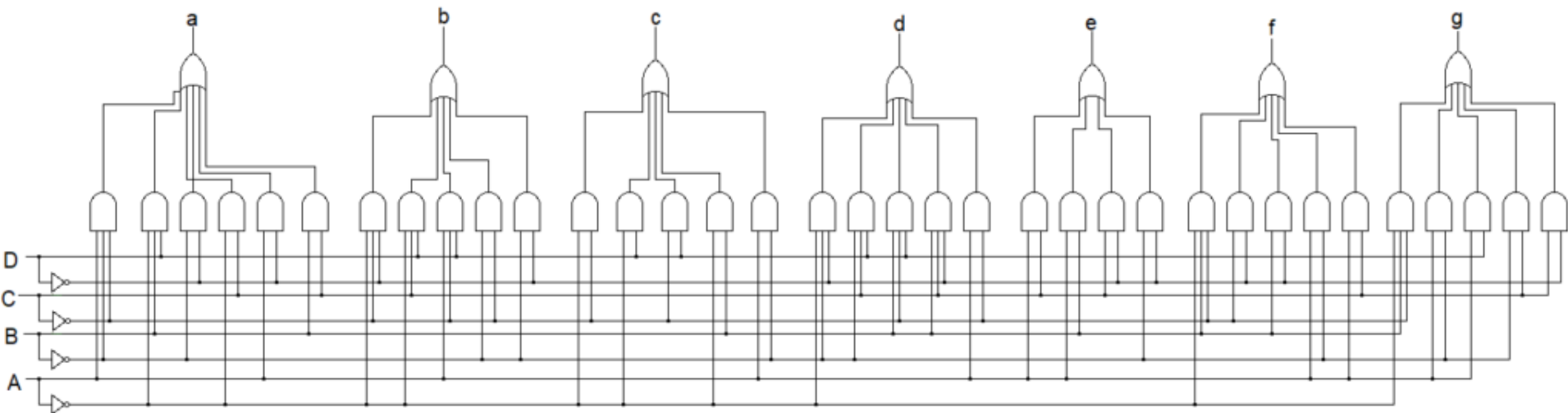
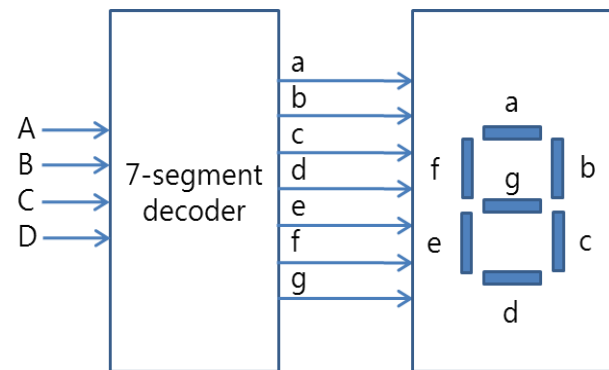
16진수	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	1	0
8	1	0	0	0	1	0	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
A	1	0	1	0	x	x	x	x	x	x	x
B	1	0	1	1	x	x	x	x	x	x	x
C	1	1	0	0	x	x	x	x	x	x	x
D	1	1	0	1	x	x	x	x	x	x	x
E	1	1	1	0	x	x	x	x	x	x	x
F	1	1	1	1	x	x	x	x	x	x	x



# 03 조합 논리회로

## 7-Segment 표시장치

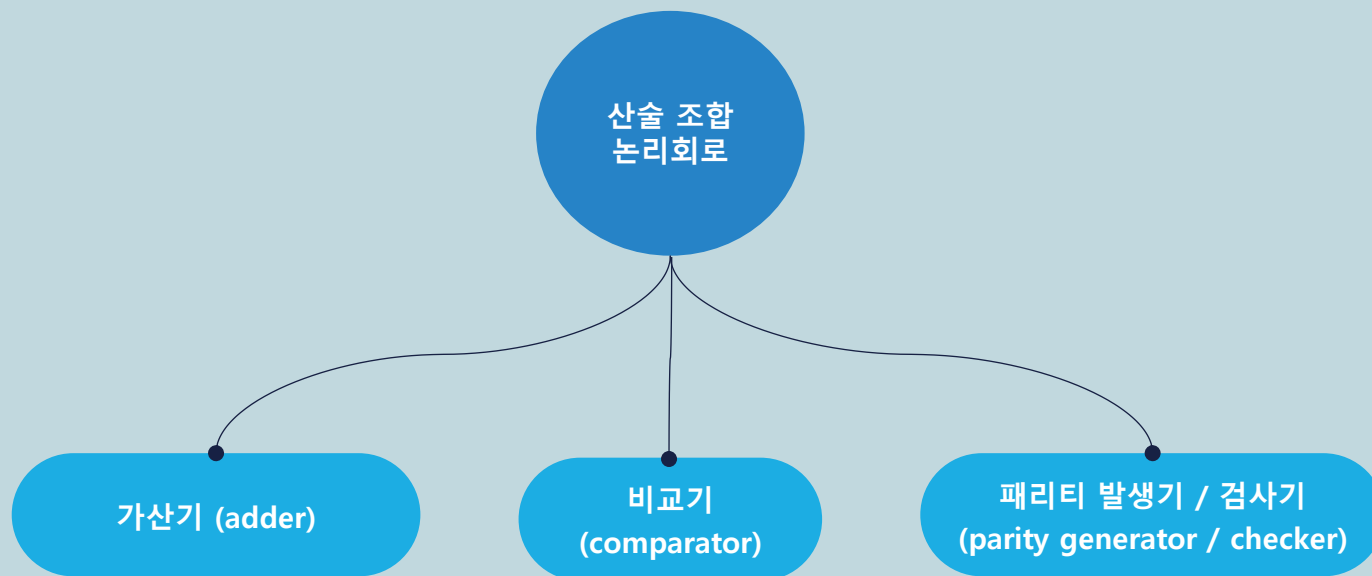
8421 BCD 7-세그먼트  
표시장치 회로도



## 4.2 산술 조합 논리회로

### 학습 목표

- 조합 논리 회로로 만든 산술 회로의 동작 이해



# 01 산술 조합 논리회로 가산기

---

## 산술 연산(arithmetic operation)회로

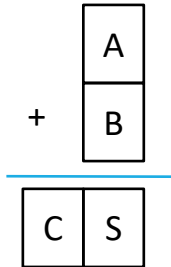
- 2진수 사칙연산과 논리 연산을 수행하는 조합 논리회로

## 가산기(adder)

- 두개의 이진수를 더하는 조합 논리
- 입력은 두개의 이진 수 / 출력은 합(sum)과 자리 올림(carry)

# 01 산술 조합 논리회로

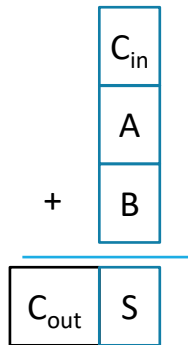
## 가산기 : 가산기의 종류



(a) 반가산기

**반가산기**  
(HA, half adder)

1비트 2진수 A, B를 더해 S와 자리올림수 C를 구한다.



(b) 전가산기

**전가산기**  
(FA, full adder)

1비트 2진수 A, B 그리고 한자리 아래 자리올림수  $C_{in}$ 를 포함한 3개의 비트를 더해 자리올림수  $C_{out}$ 을 구한다.

# 산술 조합 논리회로

## 01 가산기 : 가산기의 종류

	$C_3$	$C_2$	$C_1$	$C_0$
	$A_3$	$A_2$	$A_1$	$A_0$
+	$B_3$	$B_2$	$B_1$	$B_0$

$C_{out}$	$S_3$	$S_2$	$S_1$	$S_0$
-----------	-------	-------	-------	-------

(c) 4비트 병렬가산기

병렬 가산기  
(PA, parallel adder)

여러 비트 2진수 2개를 더해 합과 자리올림수를 구함

# 01 산술 조합 논리회로 반가산기

## 반가산기 (half adder)

- 두 비트(A, B)를 더하여 합(S)과 자리올림수(C)를 계산하는 회로

### 입출력 변수

- 입력  
: A, B
- 출력  
: S(sum), C(carry)

### 진리표

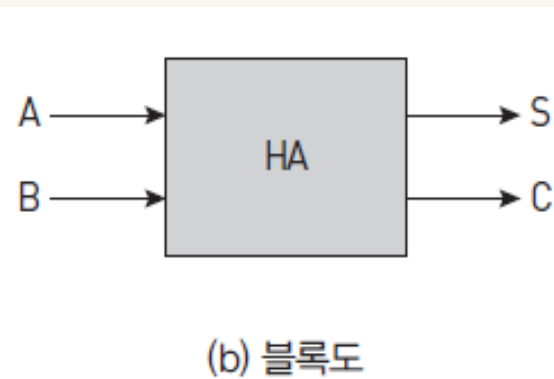
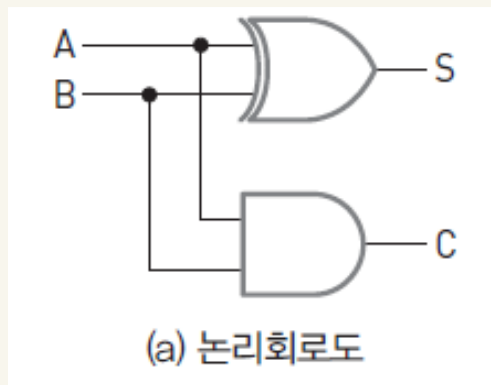
입력		출력		설명
A	B	C	S	
0	0	0	0	$0 + 0 = 00$
0	1	0	1	$0 + 1 = 01$
1	0	0	1	$1 + 0 = 01$
1	1	1	0	$1 + 1 = 10$

# 01 산술 조합 논리회로 반가산기

## 논리식

- $S = A \oplus B$
- $C = A \cdot B$

## 논리회로도



# 01 산술 조합 논리회로 전가산기

## 전가산기 (full adder)

- 한 비트 2진수 두 개(A, B)와 아랫단에서 발생한 자리올림수( $C_{in}$ )까지 세 비트를 더하여 합(S)과 자리올림수( $C_{out}$ )를 계산하는 회로

### 입출력 변수

- 입력  
: A, B,  $C_{in}$ (carry in)
- 출력  
: S(sum),  $C_{out}$ (carry out)

### 진리표

입력			출력		설명
A	B	$C_{in}$	$C_{out}$	S	
0	0	0	0	0	$0 + 0 + 0 = 00$
0	0	1	0	1	$0 + 0 + 1 = 01$
0	1	0	0	1	$0 + 1 + 0 = 01$
0	1	1	1	0	$0 + 1 + 1 = 10$
1	0	0	0	1	$1 + 0 + 0 = 01$
1	0	1	1	0	$1 + 0 + 1 = 10$
1	1	0	1	0	$1 + 1 + 0 = 10$
1	1	1	1	1	$1 + 1 + 1 = 11$

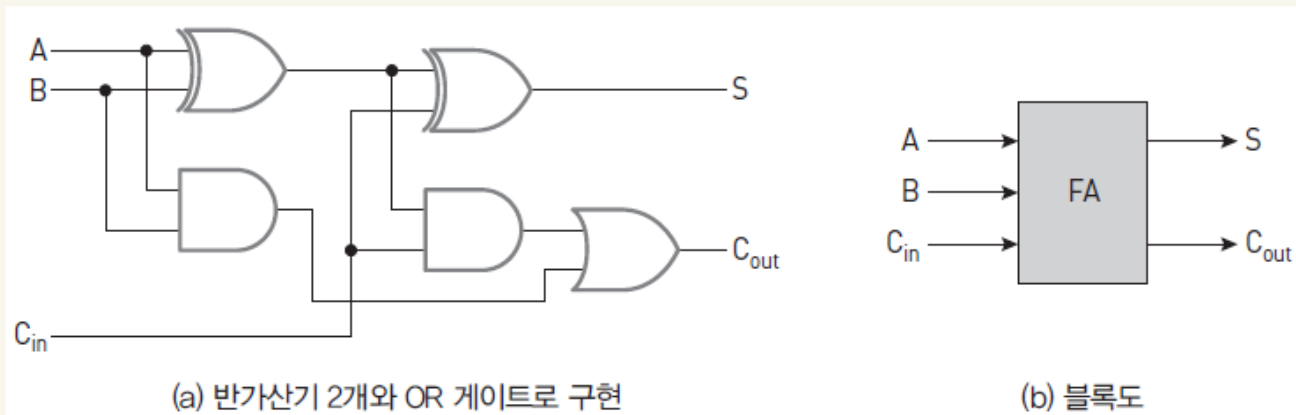


# 01 산술 조합 논리회로 전가산기

## 논리식

- $S = A \oplus B \oplus C_{in} = (A \oplus B) \oplus C_{in}$
- $C_{out}(\text{carry out}) = A \cdot B + A \cdot C_{in} + B \cdot C_{in} + A \cdot B \cdot C_{in} = A \cdot B + (A \oplus B) \cdot C_{in}$

## 논리회로도

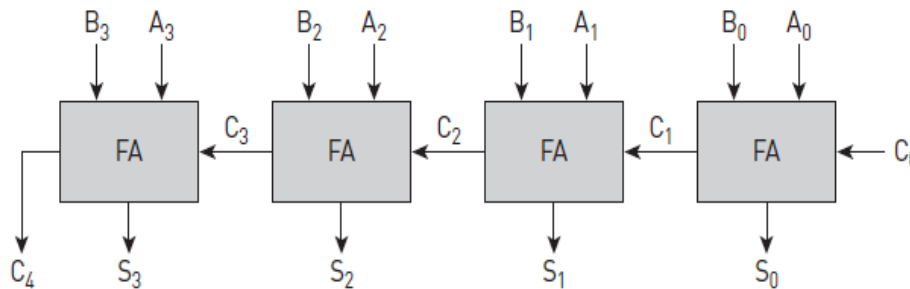


# 01 산술 조합 논리회로 병렬 가산기

## 병렬 가산기 (PA, parallel adder)

- 전 가산기를 여러 개 사용해 여러 비트를 더하는 가산기

### Ex ) 4 비트를 더하는 가산기



자리올림수:	1 1 1 0
입력 X	0 0 1 1
입력 Y	1 1 1 0
합:	0 0 0 1

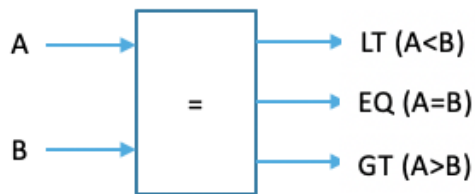
$C_0=0$ ,  $X=0011$ ,  $Y=1110$ 일 때, 합(S) = 0001, 자리올림수(C)=1110

## 02 산술 조합 논리회로 비교기

### 비트 비교기 (comparator)

- 두 비트를 비교하여 '크다, 같다, 작다'를 출력하는 회로

#### 입출력 변수



#### 진리표

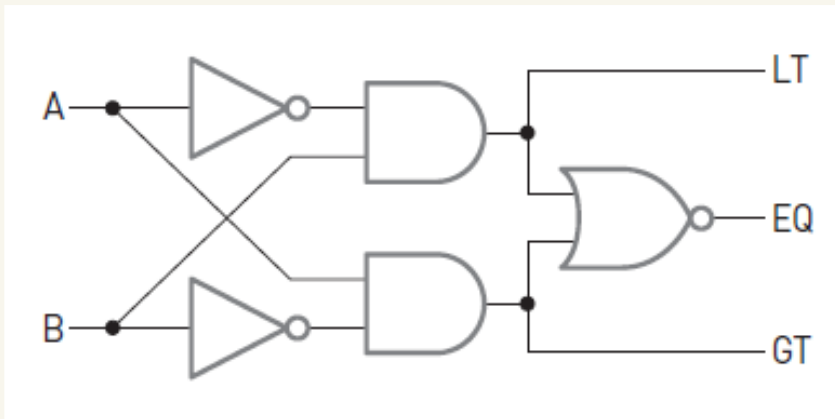
입력		출력			비고
A	B	LT (A'B)	EQ (A⊕B)'	GT (AB')	
0	0	0	1	0	0 = 0
0	1	1	0	0	0 < 1
1	0	0	0	1	1 > 0
1	1	0	1	0	1 = 1

# 02 산술 조합 논리회로 비교기

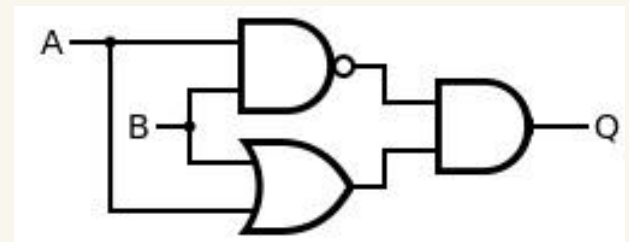
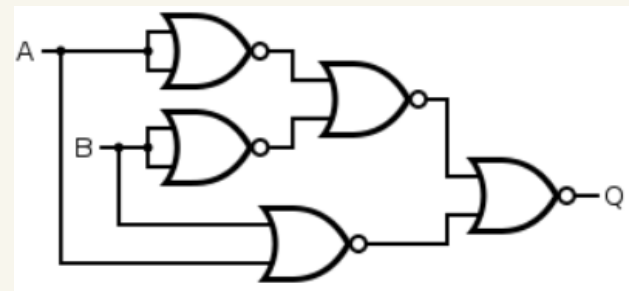
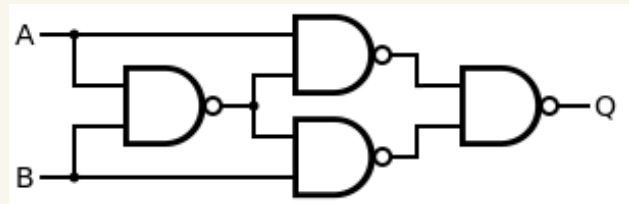
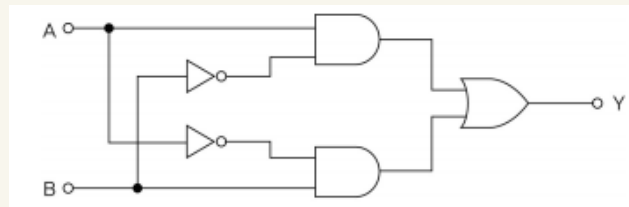
## 논리식

- $LT = (A'B)$  // A=0, B=1
- $GT = (AB')$  // A=1, B=0
- $EQ = A'B' + AB = (A \oplus B)'$  // AB = 00 or 11

## 논리 회로도



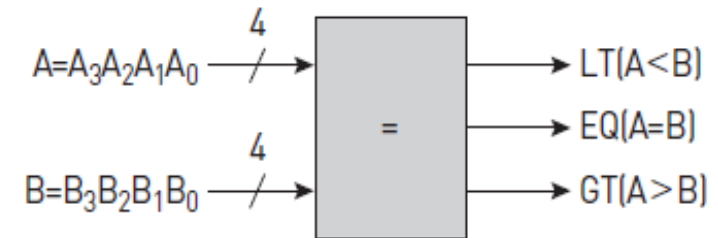
## ※ 참고(XOR)회로도



# 02 산술 조합 논리회로 4비트 비교기

두 개의 4비트 2진수를 비교하여 크다(GT), 같다(EQ), 작다(LT) 출력

입력		출력			비교
$A_3A_2A_1A_0$	$B_3B_2B_1B_0$	LT	EQ	GT	
$X_3X_2X_10$	$X_3X_2X_11$	1	0	0	A < B
$X_3X_20\text{ d}$	$X_3X_21\text{ d}$	1	0	0	
$X_30\text{ d d}$	$X_31\text{ d d}$	1	0	0	
$0\text{ d d d}$	$1\text{ d d d}$	1	0	0	
$X_3X_2X_1X_0$	$X_3X_2X_1X_0$	0	1	0	A = B
$X_3X_2X_11$	$X_3X_2X_10$	0	0	1	A > B
$X_3X_21\text{ d}$	$X_3X_20\text{ d}$	0	0	1	
$X_31\text{ d d}$	$X_30\text{ d d}$	0	0	1	
$1\text{ d d d}$	$0\text{ d d d}$	0	0	1	



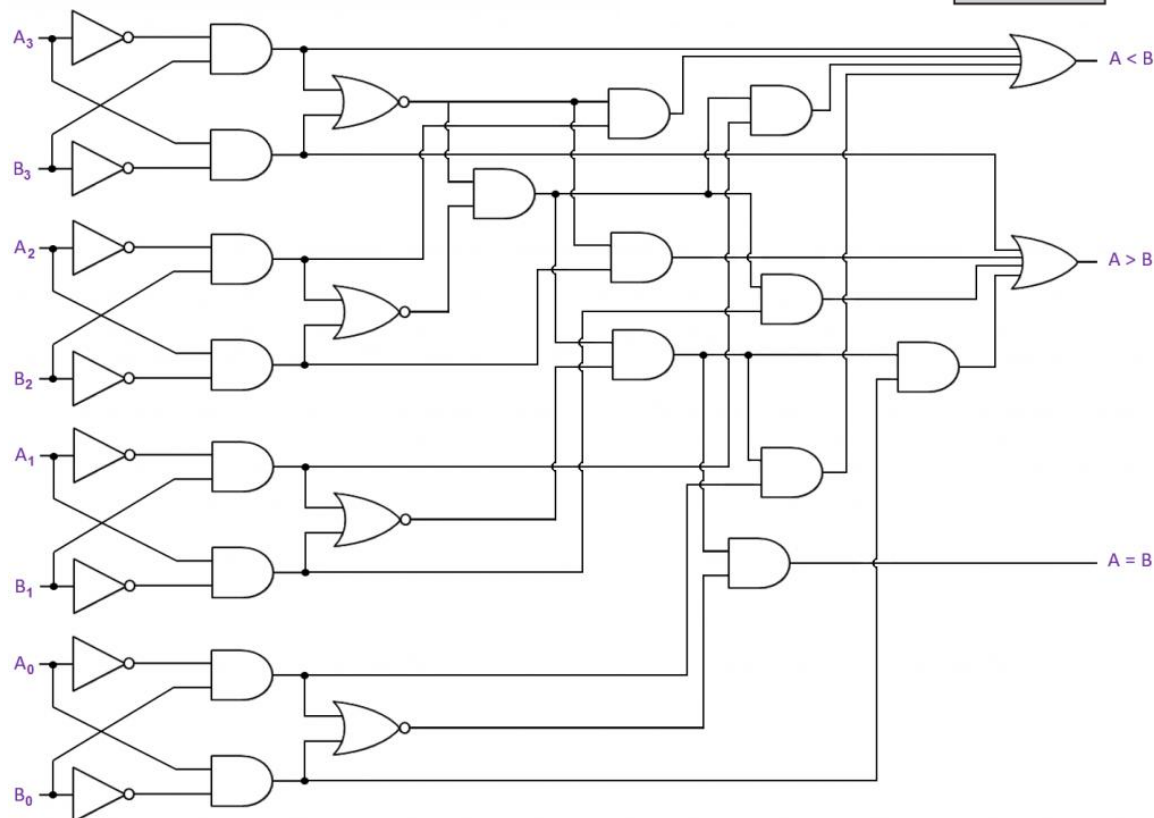
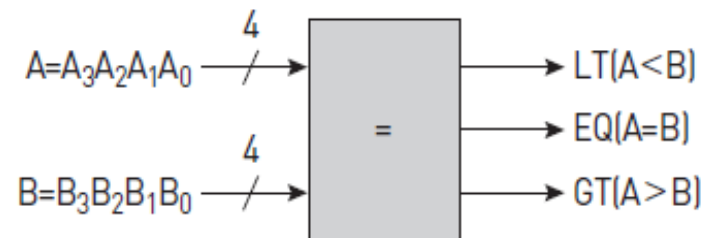
아래 입력에 대한 출력은?

- 1)  $A = 1010, B = 1001$   
LT=\_\_, EQ=\_\_, GT=\_\_
- 2)  $A = 0101, B = 0101$   
LT=\_\_, EQ=\_\_, GT=\_\_
- 3)  $A = 0001, B = 0010$   
LT=\_\_, EQ=\_\_, GT=\_\_

# 02 산술 조합 논리회로

## 4비트 비교기

두 개의 4비트 2진수를 비교하여  
크다(GT), 같다(EQ), 작다(LT) 출력



# 03 산술 조합 논리회로 패리티 발생기 / 검사기

## 패리티 비트(parity bit)

- 2진 데이터에 포함된 1의 수를 짝수(또는 홀수)로 맞추도록 추가하는 비트
- 가장 간단한 오류 검출: 홀수 비트 오류 검출, 오류 수정 불가
- 통신: 송신부에서 패리티 생성, 수신부에서 패리티 검사
- **XOR연산을 통해 패리티 검출기로 사용 가능**

## 아스키코드에 패리티를 추가한 예

아스키코드	7비트 코드	짝수 패리티	홀수 패리티
'A'	100_0001	0_100_0001	1_100_0001
'T'	101_0100	1_101_0100	0_101_0100

# 03 산술 조합 논리회로 XOR 게이트

## XOR: 홀수 함수

- 1의 수가 홀수일 때, 출력 1

A	B	C	XOR
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

## 짝수 패리티 생성기(generator)

- 데이터에 1이 홀수면 1을 추가
- $P = X_6 \oplus X_5 \oplus X_4 \oplus X_3 \oplus X_2 \oplus X_1 \oplus X_0$

## 짝수 패리티 검사기(checker)

- 데이터에 1이 홀수면 출력 1, 즉 오류 검출
- $C = P \oplus X_6 \oplus X_5 \oplus X_4 \oplus X_3 \oplus X_2 \oplus X_1 \oplus X_0$

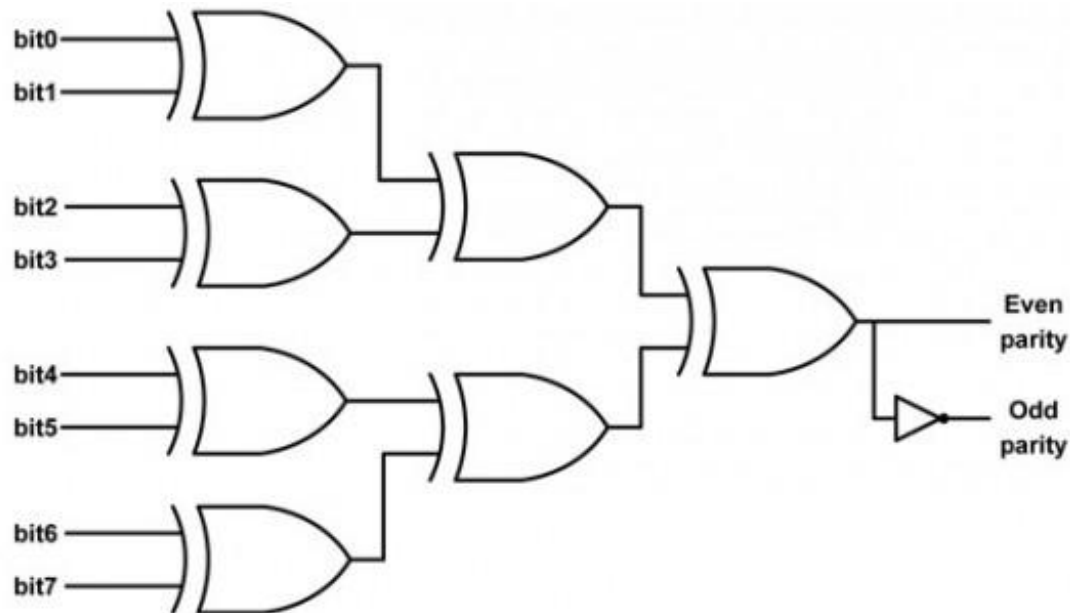


# 03 산술 조합 논리회로

## 조합 논리회로 블록빌딩 : XOR 게이트

### <8비트 패리티 생성기 >

출처: <https://gateoverflow.in/>

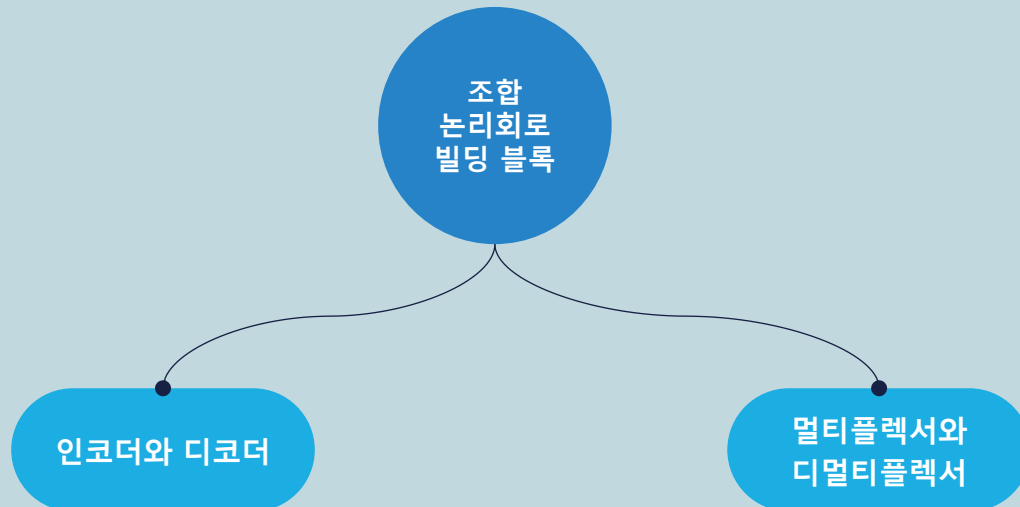


INPUT	EVEN PARITY	ODD PARITY
01010101	0	1
11010101	1	0

## 4.3 조합 논리회로 빌딩 블록

### 학습 목표

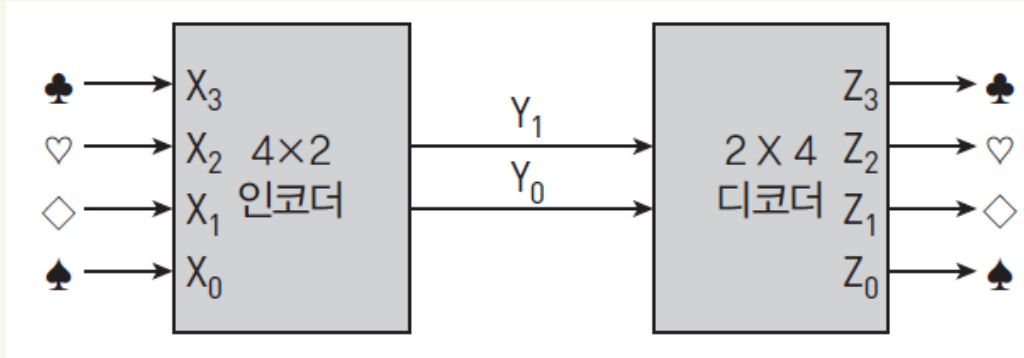
- 자주 사용되는 조합 논리회로 빌딩 블록의 동작 이해
- 빌딩 블록(building block) : 규모가 큰 논리회로의 설계를 용이하게 하기 위해 미리 만들어진 부품



# 01 조합 논리회로 빌딩 블록

## 인코더와 디코더

### 인코더와 디코더



### 4비트 인코더

기호	입력				출력		코드
	$X_3$	$X_2$	$X_1$	$X_0$	$Y_1$	$Y_0$	
♠	0	0	0	1	0	0	00
◇	0	0	1	0	0	1	01
♥	0	1	0	0	1	0	10
♣	1	0	0	0	1	1	11

### 4비트 디코더

코드	입력		출력				기호
	$Y_1$	$Y_0$	$Z_3$	$Z_2$	$Z_1$	$Z_0$	
00	0	0	0	0	0	1	♠
01	0	1	0	0	1	0	◇
10	1	0	0	1	0	0	♥
11	1	1	1	0	0	0	♣

# 01 조합 논리회로 빌딩 블록

## 인코더

### 인코더 (encoder)

- 집합의 원소(n개)에 대한 코드  $\lceil \log_2 n \rceil$  생성

### 우선순위 인코더(priority encoder)

- 인코더의 입력에 우선순위를 부여한 인코더
- 예) 4비트 인코더
  - 입력의 모든 조합을 포함하지 않음
  - 입력 신호 중 반드시 1이 하나라도 있을 때 출력이 유효 (V)
  - 입력 신호 중 하나만 1이면 나머지를 살펴볼 필요가 없음

# 01 조합 논리회로 빌딩 블록

## 인코더

Ex ) 4비트 우선순위 인코더

입력				출력		
$X_3$	$X_2$	$X_1$	$X_0$	$Y_1$	$Y_0$	$V$
0	0	0	0	x	x	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

# 01 조합 논리회로 빌딩 블록

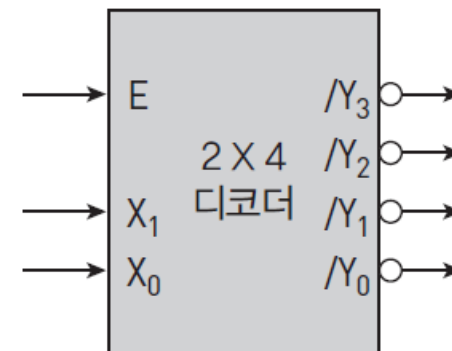
## 디코더

### 디코더(decoder)

- $n$ 비트 코드에 대한  $2^n$ 비트 2진수(one-hot) 출력
- One-hot: 비트 중 하나만 값이 다른 2진수
- 출력은 One-hot 형태로 0값을 갖는 값이 최종 결과로 출력

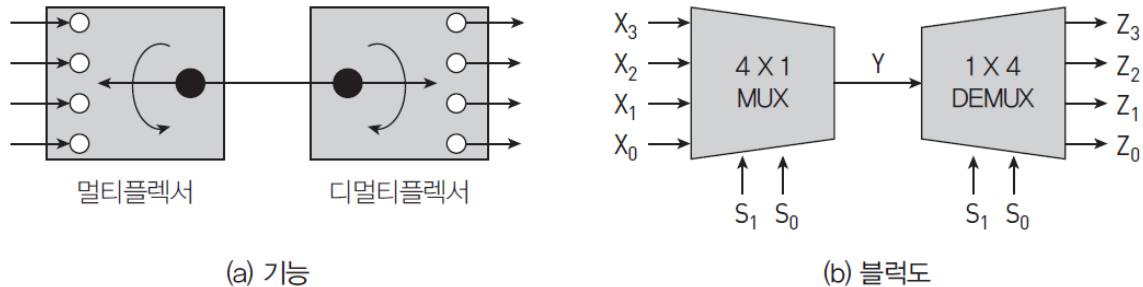
### 인에이블 제어선이 있는 디코더

E	$X_1$	$X_0$	$/Y_3$	$/Y_2$	$/Y_1$	$/Y_0$
0	x	x	1	1	1	1
1	0	0	1	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	1
1	1	1	0	1	1	1



# 02 조합 논리회로 빌딩 블록

## 멀티플렉서와 디멀티플렉서



### 멀티플렉서 & 디멀티플렉서

#### 1. 멀티플렉서 (multiplexer)

- 여러 개의 입력선 중에 하나를 선택하여 출력으로 전달
- 입력  $2^n$  (스위치  $n$ ) → 출력 1

#### 2. 디멀티플렉서 (demultiplexer)

- 하나의 입력선을 여러 개의 출력선 중 하나로 전달
- 입력 1 (스위치  $n$ ) → 출력  $2^n$

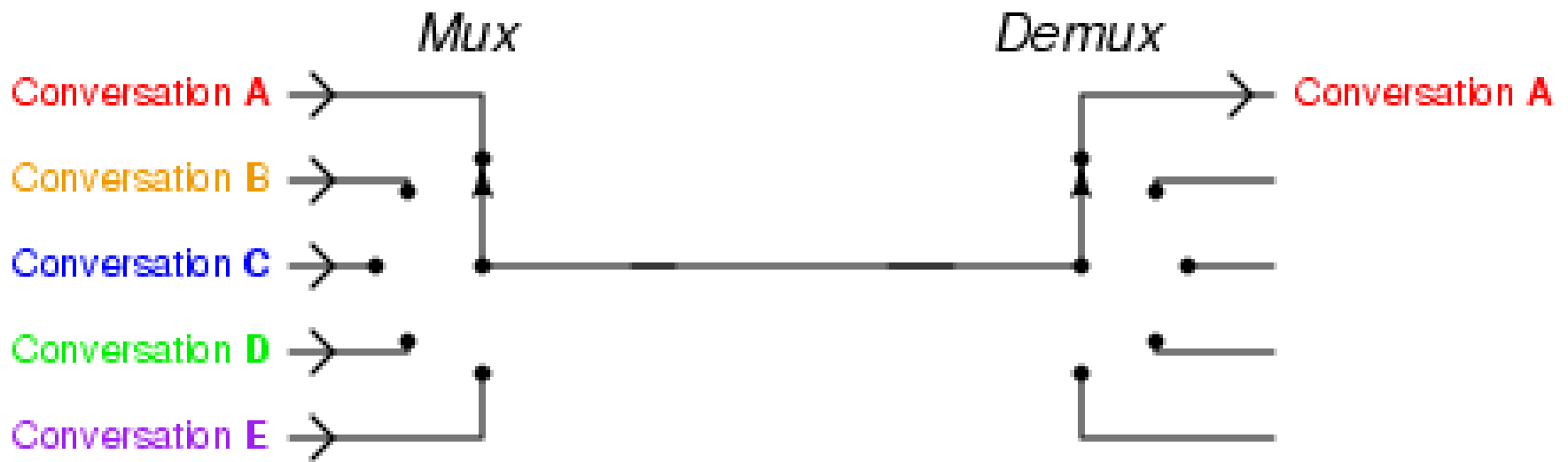
### Ex ) 신호 전달

- 멀티플렉서 선택선  $S_1S_0=01$
- 디멀티플렉서 선택선  $S_1S_0=11$

$X_1$  → Y →  $Z_3$

## 02 조합 논리회로 빌딩 블록 멀티플렉서와 디멀티플렉서

### 멀티플렉서/디멀티플렉서 동작의 예



출처: 위키백과



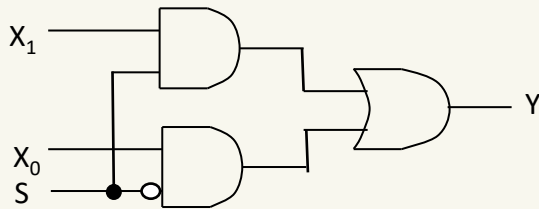
# 02 조합 논리회로 빌딩 블록

## 멀티플렉서

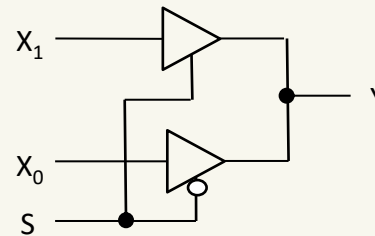
### EX ) 멀티플렉서 구현 (2X1)

S	Y
0	$X_0$
1	$X_1$

$$Y = S'X_0 + SX_1$$



<게이트로 구성>

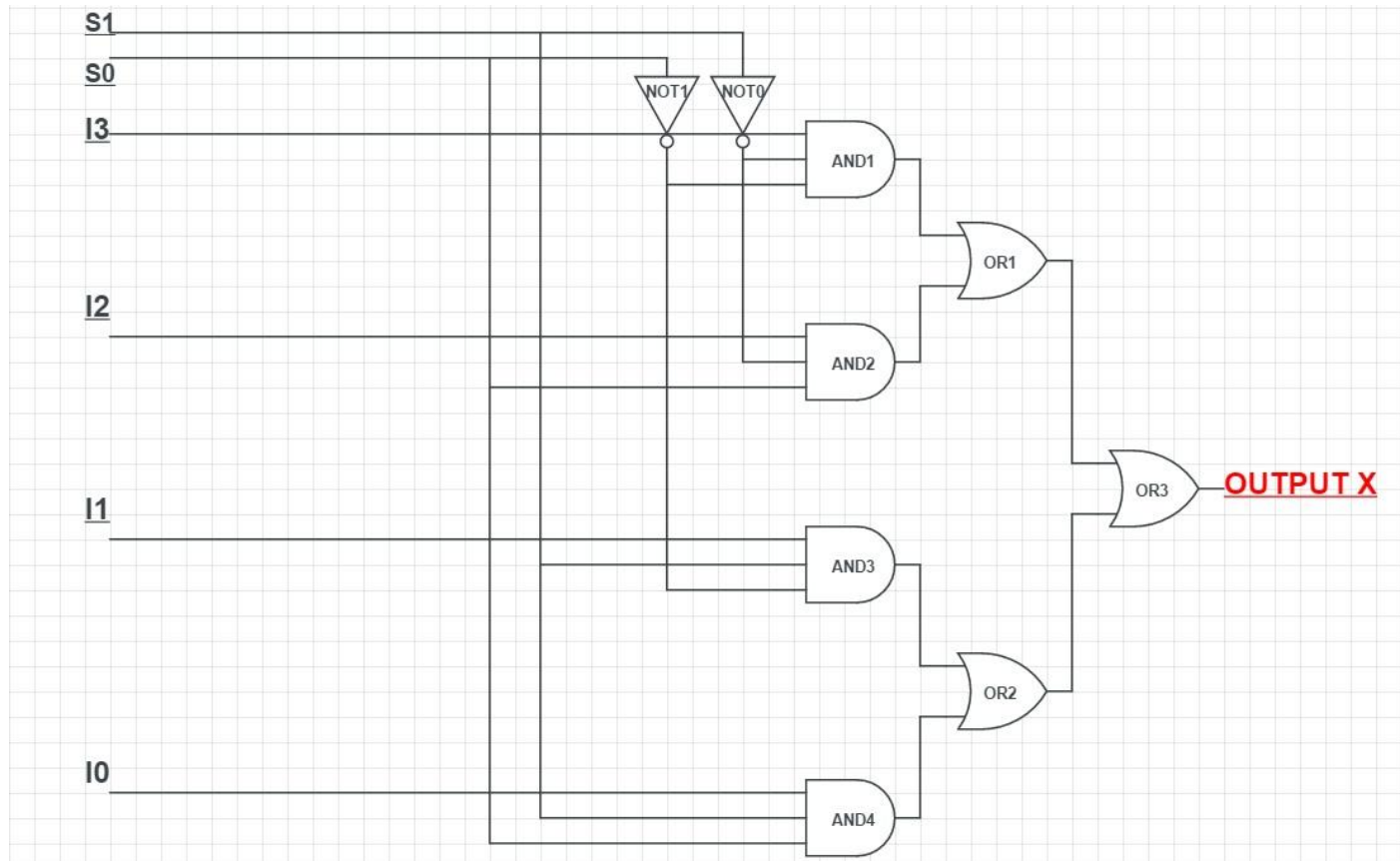


<3상태 버퍼 구현 2x1 MUX>

# 02 조합 논리회로 빌딩 블록

## 멀티플렉서

### EX ) 멀티플렉서 구현 (4X1)

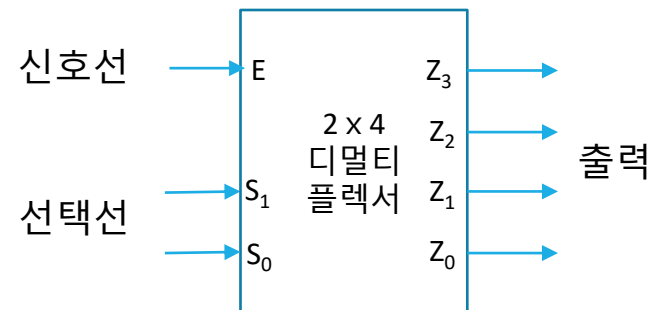
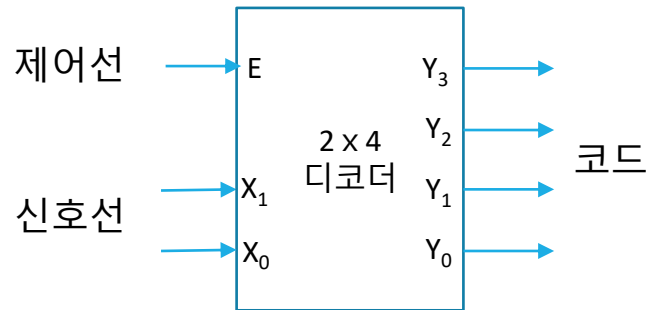


출처: 위키백과

# 02 조합 논리회로 빌딩 블록

## 디멀티플렉서

EX ) 디멀티플렉서 = 인에이블이 있는 디코더



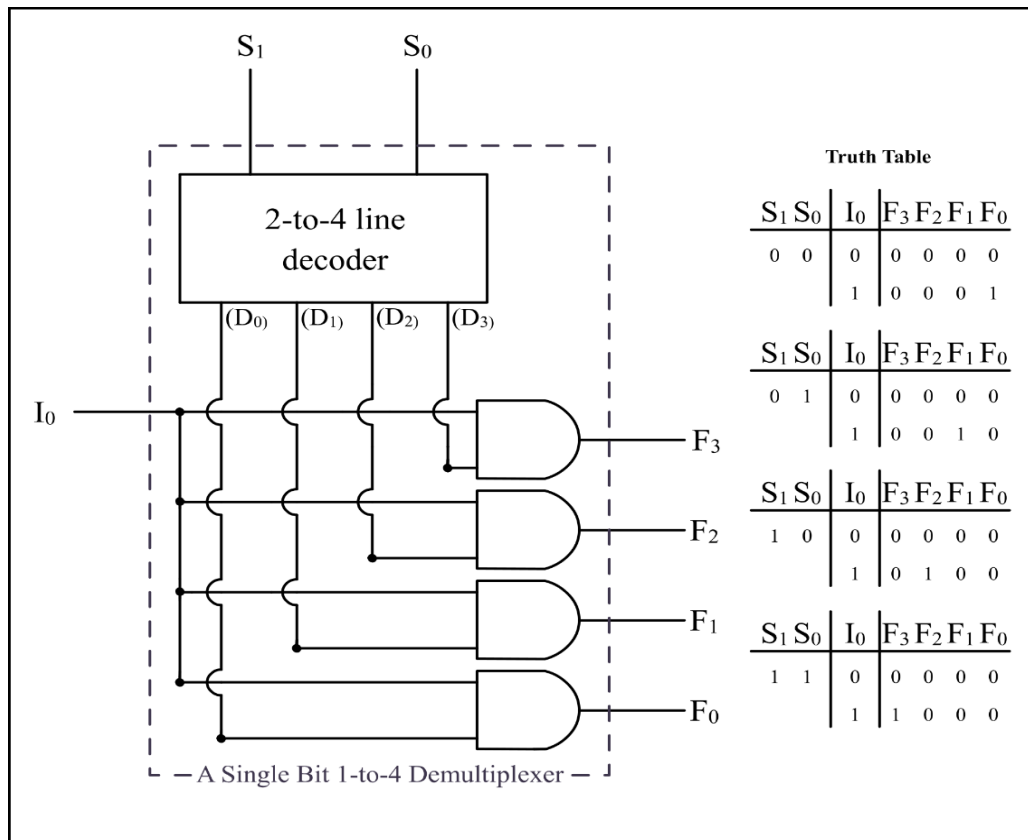
인에이블 제어선이 있는 디코더 (출력 정논리)						
E	X <sub>1</sub>	X <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

디멀티플렉서					
S <sub>1</sub>	S <sub>0</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

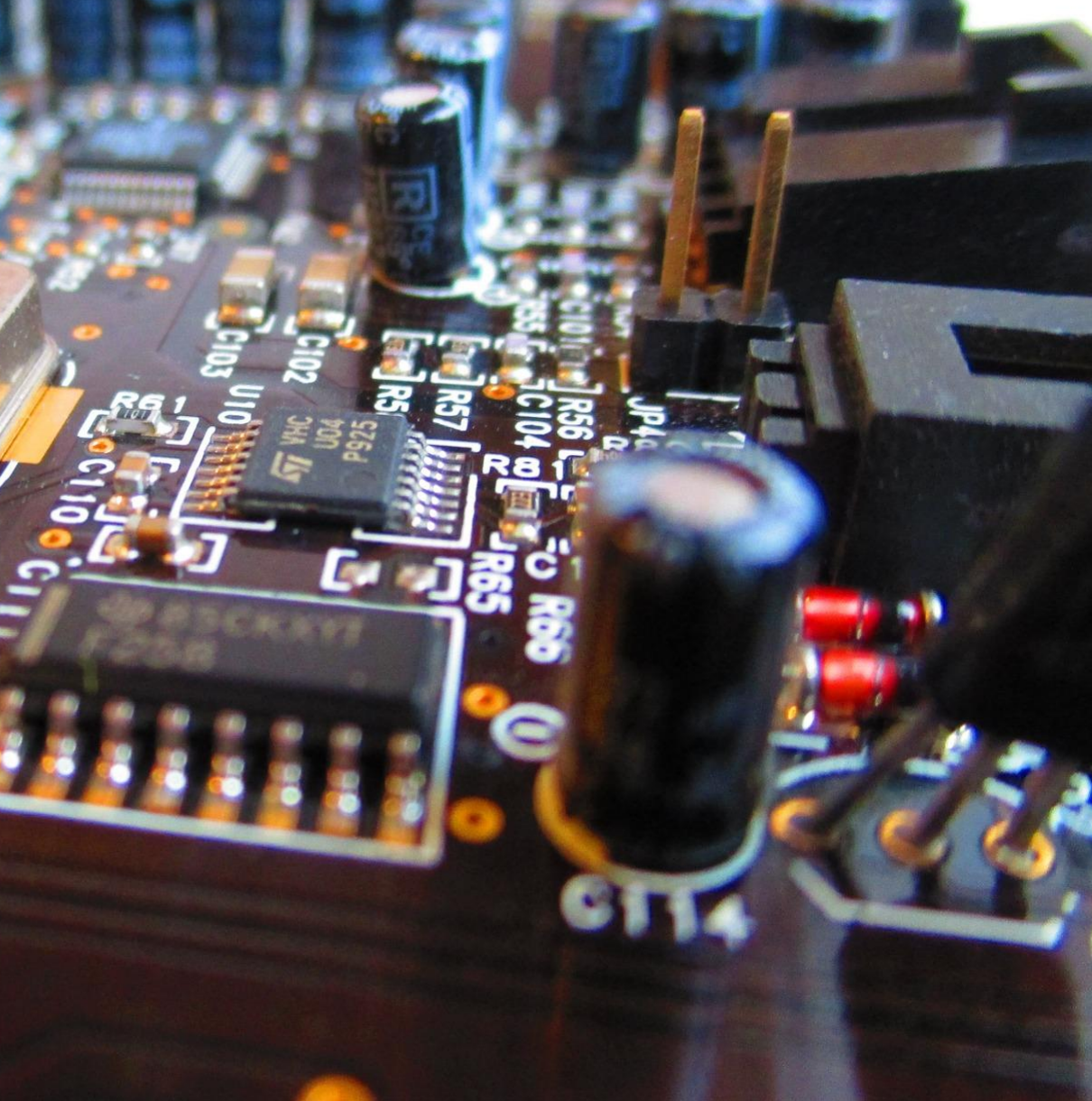
# 02 조합 논리회로 빌딩 블록

## 다멀티플렉서

### EX ) 2X4 다멀티플렉서



출처: 위키백과



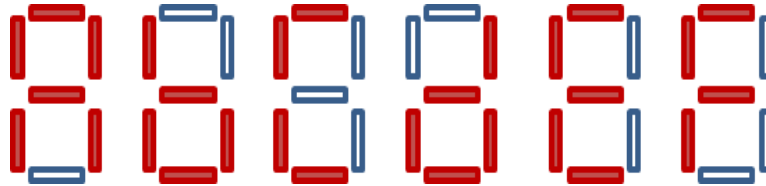
## Part 2

4.4 퀴즈

4.5 요약

# 01 조합 논리회로 문제

알파벳이 표현 가능한 7-세그먼트의 진리표를 작성하시오.



16진수	A	B	C	D	a	b	c	d	e	f	g
A	1	0	1	0							
B	1	0	1	1							
C	1	1	0	0							
D	1	1	0	1							
E	1	1	1	0							
F	1	1	1	1							

1	1	1	0	0	0	1	1	1	1	1	f
1	1	1	1	0	0	1	0	1	1	1	e
1	0	1	1	1	1	0	1	0	1	1	d
0	1	1	1	0	0	1	0	0	1	1	c
1	1	1	1	1	0	0	1	1	0	1	b
1	1	1	0	1	1	1	0	1	0	1	a
g	j	e	p	c	q	e	d	c	g	a	16진수

## 02 조합 논리회로 요약

### 4.1 조합 논리회로 소개

- 조합 논리회로의 동작 표현 : 진리표, 논리식, 논리회로도
- 무관 조건 : 출력에 영향을 주지 않는 입력 조건

### 4.2 산술 조합 논리회로 소개

- 가산기 & 비교기
- 패리티 발생기 / 검사기

### 4.3 조합 논리회로 빌딩 블록

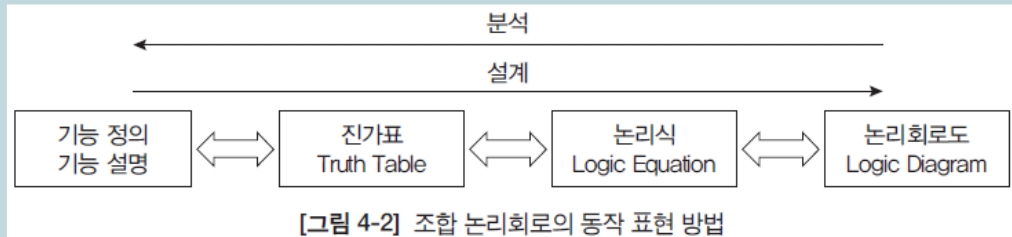
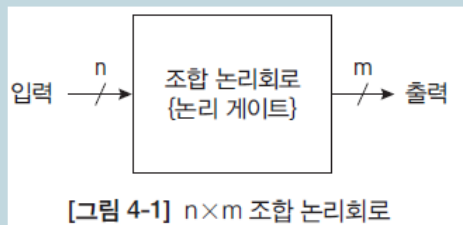
- 인코더 & 디코더
- 멀티플렉서 & 디멀티플렉서

### 제 5장 순차 논리소자

- 래치, 플립플롭 : 1비트 기억 소자
- 레지스터 : n비트 기억 소자

# 02 조합 논리회로 소개

## 조합 논리회로와 동작 표현 방법



## 진리표에서 논리식 구하는 방법

1. 논리식을 최소항의 합으로 표현한다.
2. 부울대수 공식으로 간소화 한다.

## 무관조건 (don't care condition)

- 회로의 동작에 영향을 주지 않는 입력 조합이나 출력



## 02 조합 논리회로 산술 논리회로

### 산술 회로

- 입력에 대한 출력이 항상 같다. 따라서 조합 논리회로.

### 가산기

- 반가산기

$$(\text{Carry}, \text{Sum}) \leq A + B$$

- 전가산기

$$(\text{Carry}, \text{Sum}) \leq A + B + C_{\text{in}}$$

- 병렬 가산기

$$(\text{Carry}, S_3S_2S_1S_0) \leq A_3A_2A_1A_0 + B_3B_2B_1B_0$$

### 비교기

- 1비트 비교기

$$(\text{LT}, \text{EQ}, \text{GT}) \leq (A == B)$$

- 4비트 비교기

$$(\text{LT}, \text{EQ}, \text{GT}) \leq (A_3A_2A_1A_0 == B_3B_2B_1B_0)$$

### 패리티 발생기 / 검사기

- XOR 게이트

홀수 발생기

## 02 조합 논리회로 빌딩블록

### 인코더

- 코드를 만드는 회로
- 여러 장치가 프로세서로 전달하는 신호를 받아 코드를 전달

### 디코더

- 코드를 푸는 회로
- 기억장치 주소를 입출력장치 선택선으로 변환

### 멀티플렉서

- 여러 개의 입력 중 하나를 선택
- 신호를 시스템 버스로 연결

### 디멀티플렉서

- 하나의 신호를 여러 곳 중 하나로 전달
- 시스템 버스의 신호를 여러 레지스터 중 하나로 연결