

조합논리회로

Jo, Heeseung

4.1.1 조합 논리회로의 동작 표현

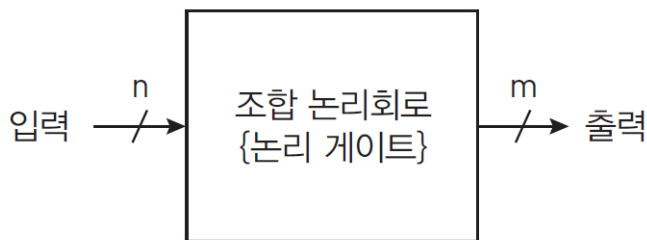
조합 논리회로(combination logic circuit)

- 입력 조합에 따라 출력을 결정하는 논리회로

동작 표현 방법

- 진가표 = 동작 특성표 = 특성표 (characteristic table)
- 논리식(logic equation) = 부울식(Bool equation) = 논리함수(logic function)
- 논리 회로도(logic diagram)

[표 4-1] $m \times n$ 조합 논리회로의 진가표



[그림 4-1] $n \times m$ 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				

동작 표현 방법



설명 순서

- (1) 논리식 → 논리회로도
- (2) 논리회로도 → 논리식
- (3) 논리식 → 진가표
- (4) 진가표 → 논리식: 최소항(또는 최대항)에 의한 방법

논리식 → 논리회로도

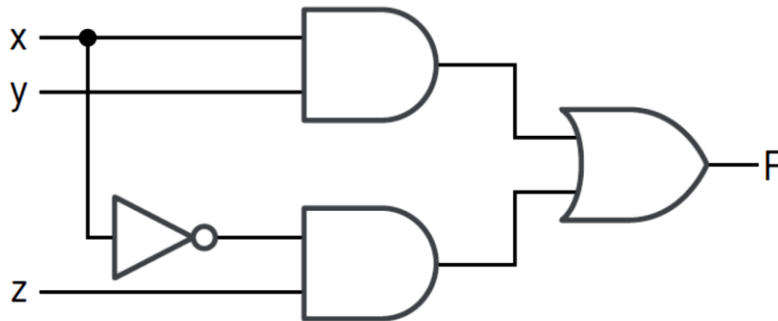
논리식의 연산 우선 순위

- 괄호() > NOT(') > AND > OR

논리회로도 그리는 방법

- 입력신호를 왼편에, 출력신호를 오른편에 배치
- 연산 우선순위에 따라 입력에서 출력 방향으로 게이트를 배치하고 입력과 출력을 연결

[예제 4-1] $F = x \cdot y + x' \cdot z$



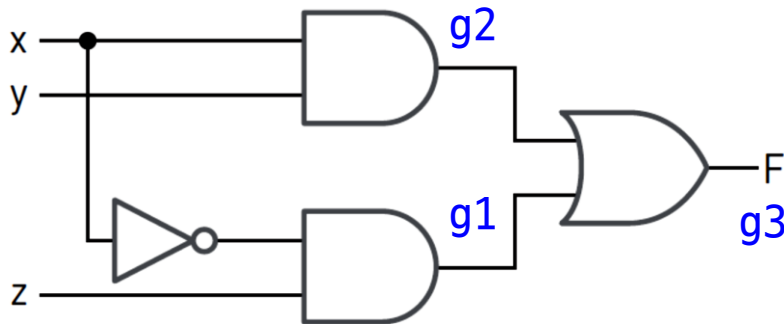
[그림 4-3] 논리회로도 그리기

논리회로도 → 논리식

논리식 구하는 방법

- 입력, 출력, 게이트의 출력에 논리 변수 할당
- 입력단부터 출력단 방향으로 게이트의 논리식 적기
- 출력단에 포함된 게이트의 출력을 입력 신호의 논리식으로 대치

[예제 4-2] 논리식은?



논리식 → 진가표

방법

- 모든 입력신호의 조합을 2진수 순서로 표에 배치
- 논리식의 연산 우선순위에 따라 세부 항에 대한 출력을 구함

[예제 4-3] $F = x \cdot y + x' \cdot z$

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

4.1.2 최소항

진가표로부터 논리식을 구하는 두 가지 방법

- 논리곱의 합(sum of products)으로 표현하는 방법
- 논리합의 곱(product of sums)으로 표현하는 방법



서로 쌍대

논리곱과 최소항

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

곱항의 합(sum of products)

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

최소항 (minterm)

논리변수들이 모두 참여하는 논리곱

= 입력신호의 조합 중에서 출력이 하나만 1인 논리함수

[표 4-3] 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								
0	0	1								
0	1	0								
0	1	1								
1	0	0								
1	0	1								
1	1	0								
1	1	1								

최소항의 합

논리식 구하기

- 1단계: 진가표에 대한 논리식을 최소항의 합으로 표현
- 2단계: 간소화 (부울대수, 인접항 찾기)

[예제 4-4] 진가표에 대한 논리식

x	y	z	F	m_1	m_3	m_6	m_7
0	0	0	0				
0	0	1	1				
0	1	0	0				
0	1	1	1				
1	0	0	0				
1	0	1	0				
1	1	0	1				
1	1	1	1				

(1) 최소항의 합

$$F(x,y,z) = \text{-----}$$

(2) 간소화

$$F(x,y,z) = \text{-----}$$

Exercise

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

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

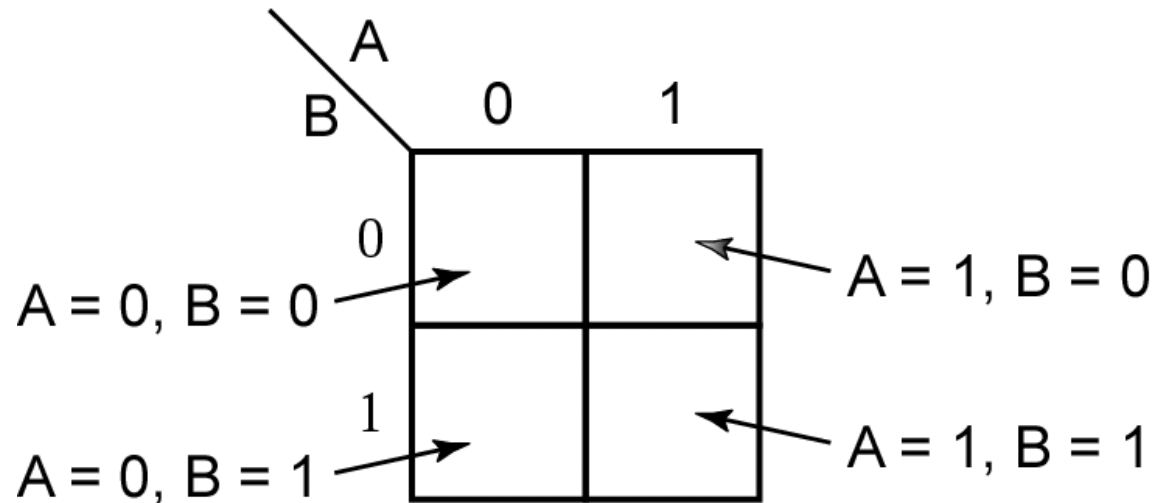
$$F(x,y,z) = xz + y'z$$

Two- and Three-Variable Karnaugh Maps

Karnaugh Map

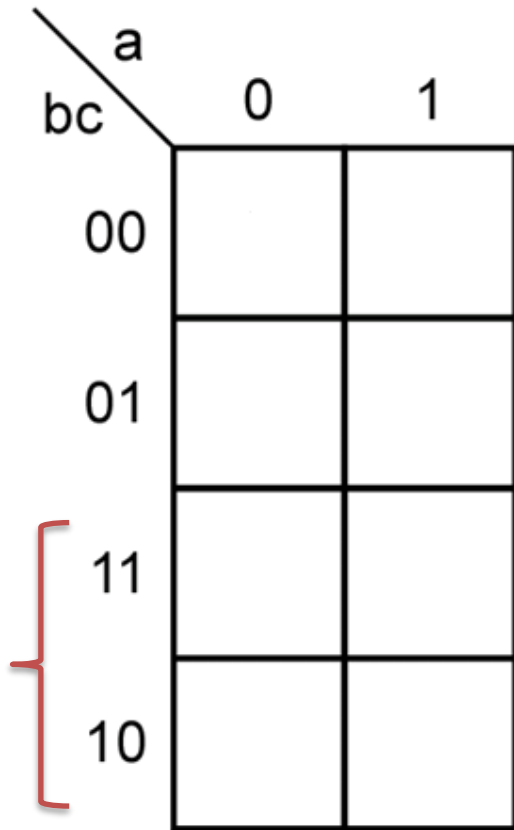
- 최소항의 합으로 표현하고 간소화하기 위한 방법

2-variable Karnaugh Map

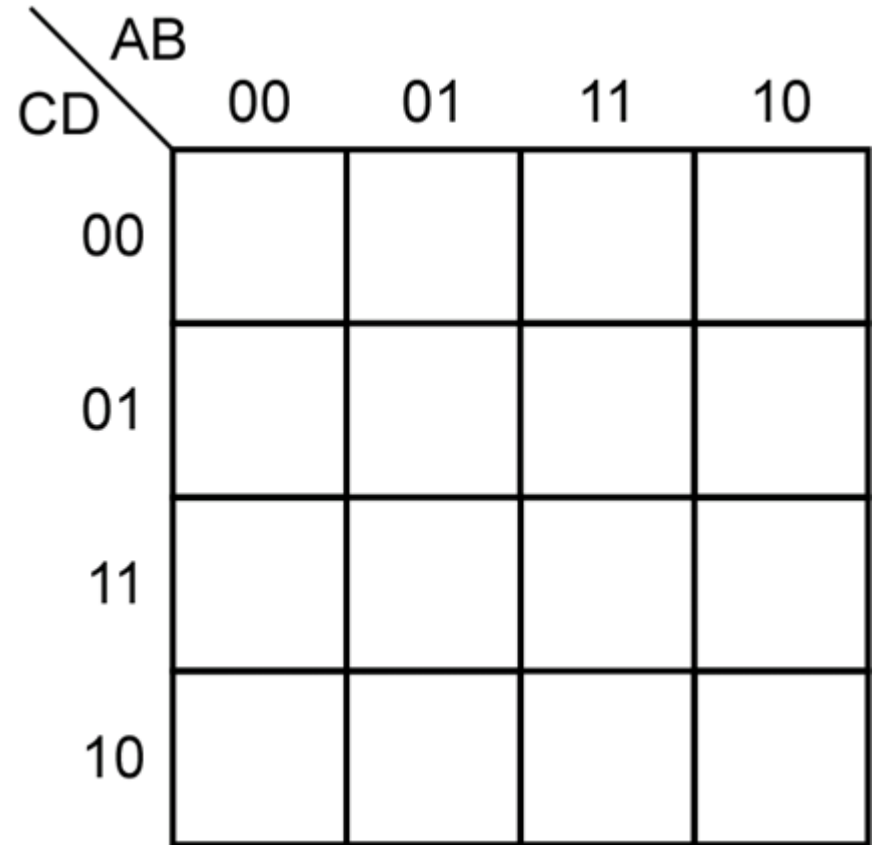


Two- and Three-Variable Karnaugh Maps

3-variable Karnaugh Map



4-variable Karnaugh Map

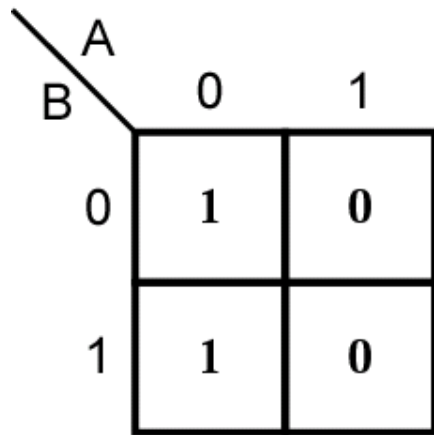


Two- and Three-Variable Karnaugh Maps

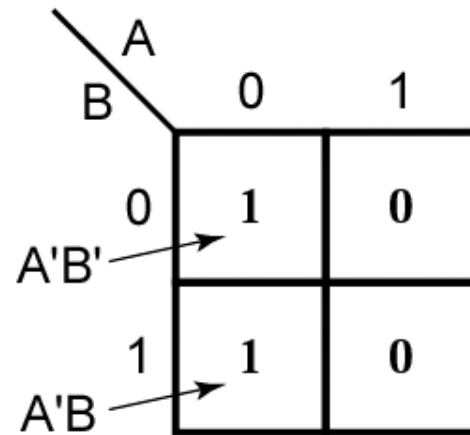
Truth Table for a function F

(a)

A	B	F
0	0	1
0	1	1
1	0	0
1	1	0

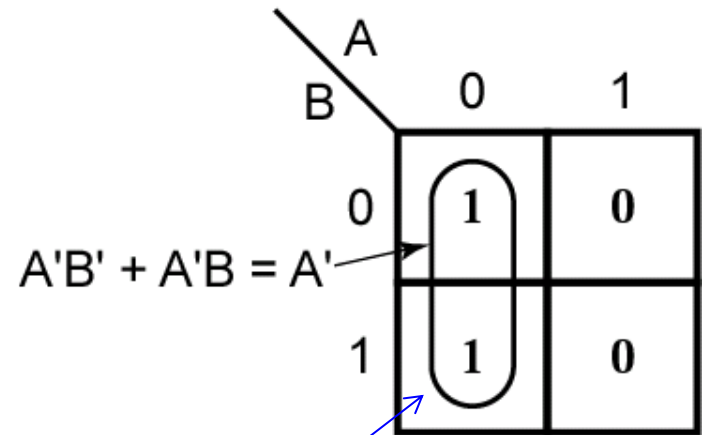


(b)



$$F = A'B' + A'B$$

(c)



$$F = A'$$

(d)

Looping adjacent 1s as large as possible
The number of 1 must be 2^k

Two- and Three-Variable Karnaugh Maps

Simplification of a Three-Variable Function

a \ bc	0	1
00		
01	1	1
11	1	
10		

$$F = \sum m(1, 3, 5)$$

(a) Plot of minterms

$$T_1 = a'b'c + a'bc = a'c$$

a \ bc	0	1
00		
01	1	1
11	1	
10		

$$F = a'c + b'c$$

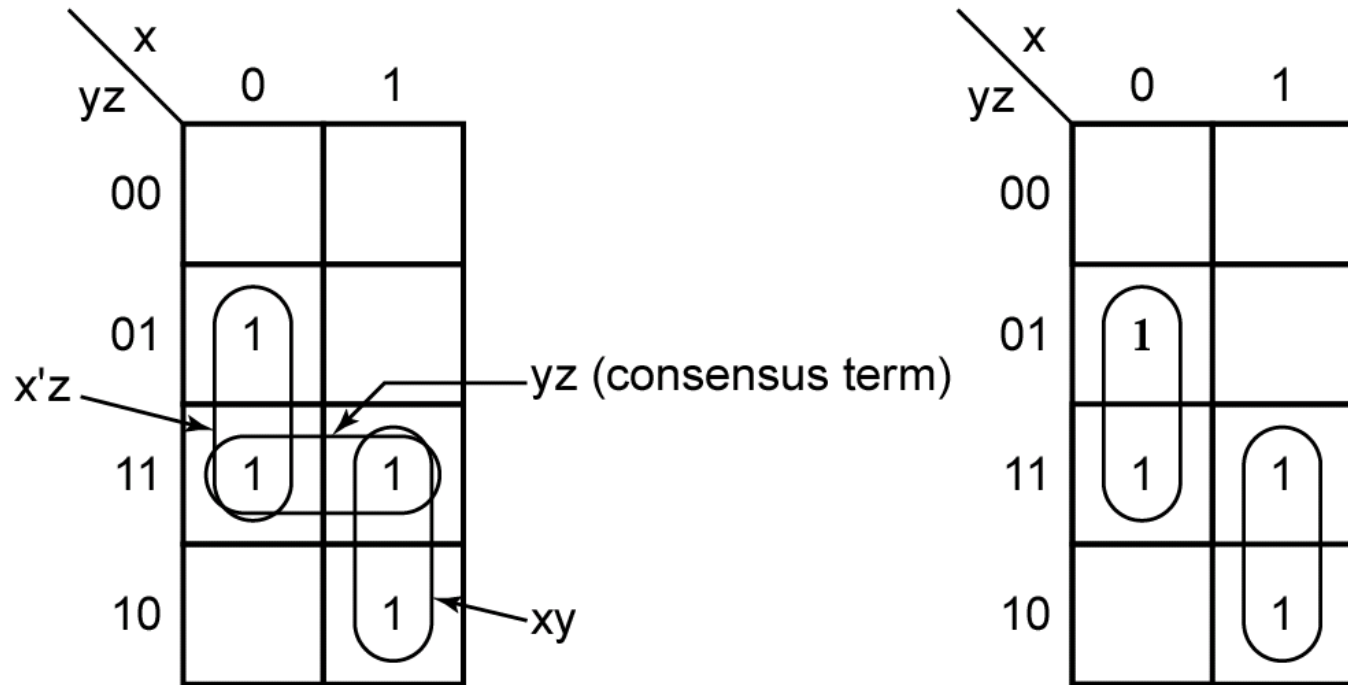
(b) Simplified form of F

$$T_2 = a'b'c + ab'c = b'c$$

$$F = T_1 + T_2 = a'c + b'c$$

Two- and Three-Variable Karnaugh Maps

Karnaugh maps which illustrate the Consensus theorem



$$xy + x'z + yz = xy + x'z$$

Consensus term is redundant

x	y	z	F	m_1	m_3	m_6	m_7
0	0	0	0				
0	0	1	1				
0	1	0	0				
0	1	1	1				
1	0	0	0				
1	0	1	0				
1	1	0	1				
1	1	1	1				

Consensus theorem

Consensus theorem (합의 항 규칙)

$$XY + X'Z + YZ = XY + X'Z$$

Proof :

$$\begin{aligned} XY + X'Z + YZ &= XY + X'Z + (X + X')YZ \\ &= (XY + XYZ) + (X'Z + X'YZ) \\ &= XY(1 + Z) + X'Z(1 + Y) = XY + X'Z \end{aligned}$$

Example:

$$a'b' + ac + bc' + b'c + ab = a'b' + ac + bc'$$

Dual form of consensus theorem

$$(X + Y)(X' + Z)(Y + Z) = (X + Y)(X' + Z)$$

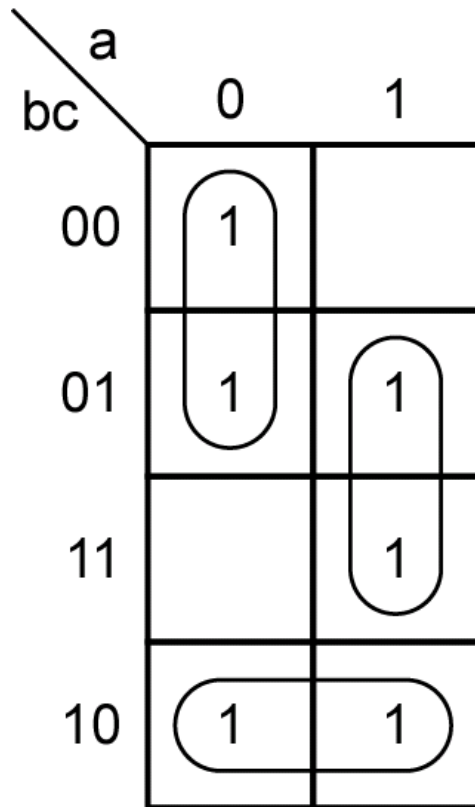
Example:

$$(a + b + c')(a + b + d')(b + c + d') = (a + b + c')(b + c + d')$$

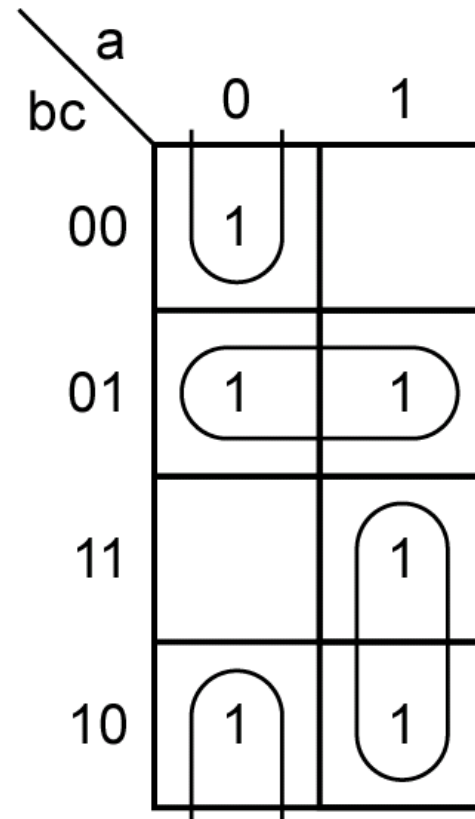
Two- and Three-Variable Karnaugh Maps

Function with Two Minimal Forms

$$F = \sum m(0,1,2,5,6,7)$$



$$F = a'b' + bc' + ac$$



$$F = a'c' + b'c + ab$$

Karnaugh Maps Ex.

$$F(a,b,c,d) = \sum m(1,3,5,7,9)$$

Karnaugh Maps Ex.

<i>A</i>	<i>B</i>	<i>C</i>	<i>f</i>
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Karnaugh Maps Ex.

$$F(a,b)=\Sigma m(1,2,3)$$

$$F(a,b,c)=\Sigma m(1,3,5,7)$$

$$F(a,b,c,d)=\Sigma m(0,2,5,7,8,10,13,15)$$

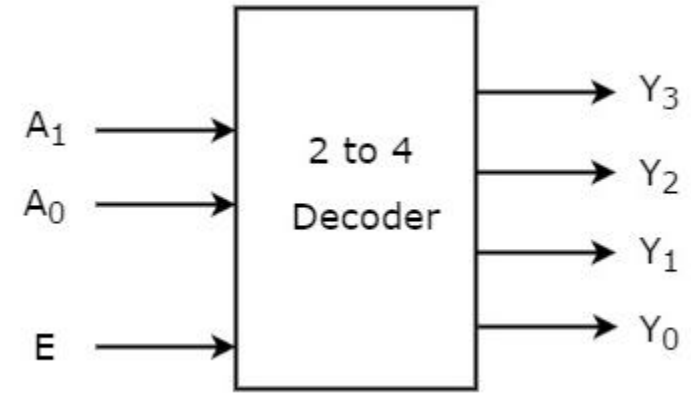
$$F(a,b,c,d)=\Sigma m(0,1,2,3,8,9,10,11)$$

$$F(a,b,c)=\Sigma m(0,2,4,6)$$

4.1.3 무관조건

무관조건(don't care condition)

- 특정한 입력신호 조합이 절대로 발생하지 않거나 입력신호에 대한 출력이 회로의 동작에 영향을 주지 않는 조건
- 진가표에 x 또는 d로 표현



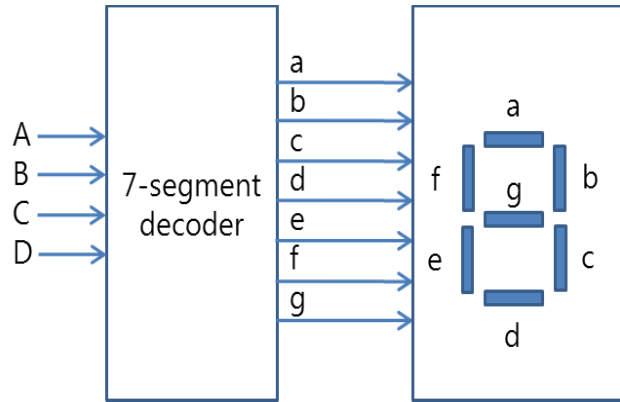
예) 디코더 진가표

입력			출력				
Enable	A ₁	A 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

7-Segment 표시장치

8421 BCD 7-세그먼트 표시장치 디코더

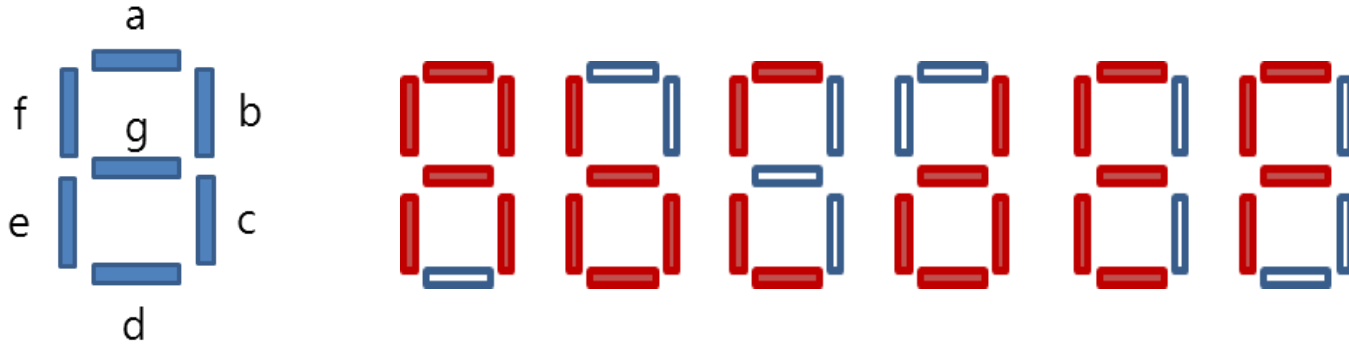
- 0-9를 표현, A-F 입력은 없음



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							
2	0	0	1	0							
3	0	0	1	1							
4	0	1	0	0							
5	0	1	0	1							
6	0	1	1	0							
7	0	1	1	1							
8	1	0	0	0							
9	1	0	0	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

[예제 4-5] 7-세그먼트 표시장치

다음과 같이 A-F를 표현하는 진리표를 작성해보시오



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							

Incompletely Specified Functions (Ex.)

Finding function:

Case 1: assign '0' on X's

$$F = A'B'C' + A'BC + ABC = A'B'C' + BC$$

Case 2: assign '1' to the first X and '0' to the second 'X'

$$F = A'B'C' + A'B'C + A'BC + ABC = A'B' + BC$$

Case 3: assign '1' on X's

$$F = A'B'C' + A'B'C + A'BC + ABC' + ABC = A'B' + BC + AB$$

A	B	C	F
0	0	0	1
0	0	1	X
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	X
1	1	1	1

→ The case 2 leads to the simplest function

That's all?

Karnaugh Maps Ex.

Truth Table with Don't Cares

<i>A B C</i>	<i>F</i>
0 0 0	1
0 0 1	X
0 1 0	0
0 1 1	1
1 0 0	0
1 0 1	0
1 1 0	X
1 1 1	1

4.2 산술 조합 논리회로

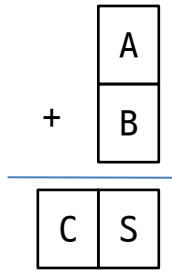
산술 회로

- 입력에 대한 출력이 항상 같음
- 따라서, 조합 논리회로
- 조합논리회로로 산술회로를 구성 가능

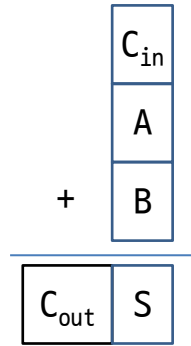
예)

- 가산기(adder)
- 비교기(comparator)
- 패리티 발생기/검사기(parity generator/checker)

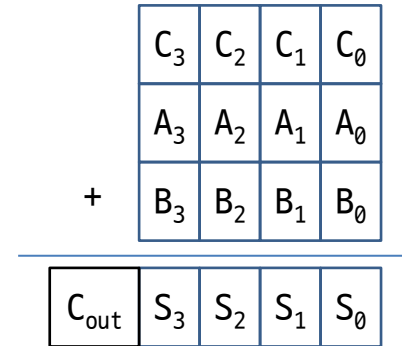
4.2.1 가산기



(a) 반가산기



(b) 전가산기

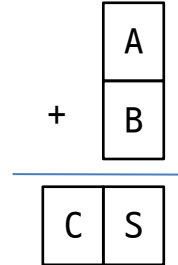


(c) 4비트 병렬가산기

반가산기

반가산기(half adder)

- 두 비트를 더하여
- 합과 자리올림수를 계산하는 회로



입출력 변수

- 입력: A, B
- 출력: S(sum), C(carry)

논리식

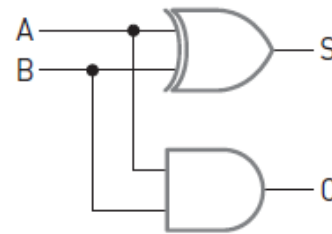
S = _____

C = _____

진가표

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

논리회로도




(a) 논리회로도



(b) 블록도


논리게이트 종류 (XOR)

〈표 3-11〉 2-입력 XOR 게이트

입력		출력	XOR 게이트 기호
X	y	$F=x\oplus y$	
0	0	0	
0	1	1	
1	0	1	
1	1	0	

$$A'B + AB' = A\oplus B$$

〈표 3-12〉 2-입력 XNOR 게이트

입력		출력	XNOR 게이트 기호
x	y	$F=(x\oplus y)'$	
0	0	1	
0	1	0	
1	0	0	
1	1	1	

$$A'B' + AB = (A\oplus B)'$$

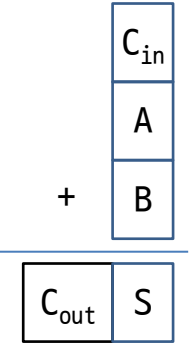
전가산기(1)

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)



진가표

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

전가산기(2)

$$\begin{aligned} S &= a'b'c + a'bc' + ab'c' + abc \\ &= (a'b'+ab)c + (a\oplus b)c' \\ &= (a\oplus b)'c + (a\oplus b)c' \\ &= (a\oplus b)\oplus c \end{aligned}$$

$$\begin{aligned} C_{out} &= a'bc_{in} + ab'c_{in} + abc_{in}' + abc_{in} \\ &= ab(c_{in}' + c_{in}) + (a'b + ab')c_{in} \\ &= ab + (a\oplus b)c_{in} \end{aligned}$$

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

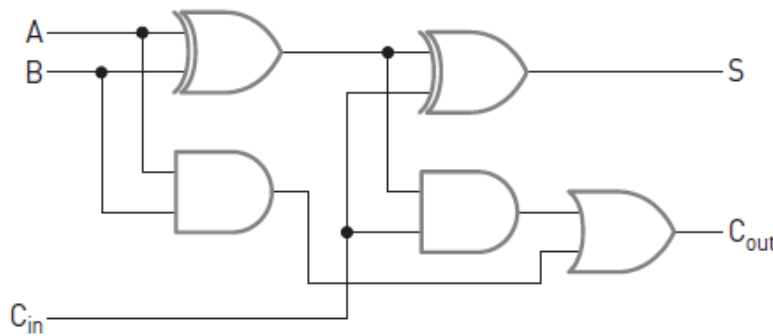
전가산기(3)

논리식

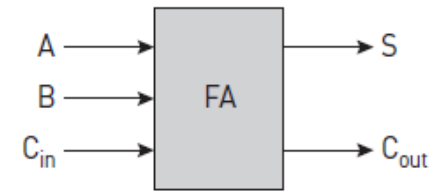
- $S = A \oplus B \oplus C_{in} = (A \oplus B) \oplus C_{in}$
- $C_{out}(\text{carry out}) = A \cdot B + (A \oplus B) \cdot C_{in}$
 - $A \cdot B$: A와 B를 더했을 때 올림수가 발생한 경우
 - $(A \oplus B) \cdot C_{in}$: A와 B를 더한 합이 1이고, 이전 단에서 올림수가 발생한 경우

반가산기의 S

논리회로도



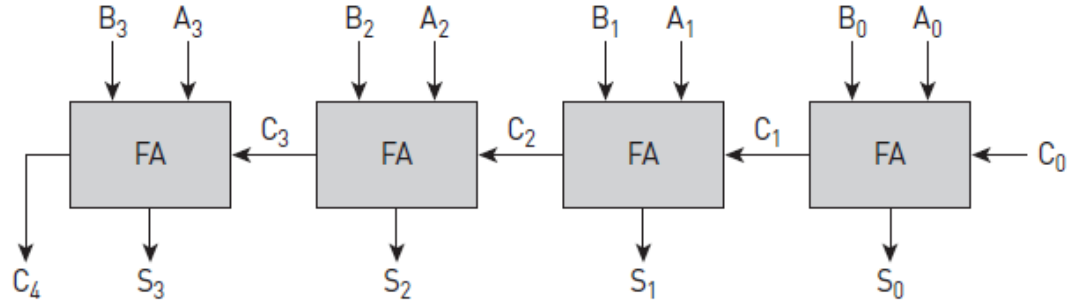
(a) 반가산기 2개와 OR 게이트로 구현



(b) 블록도

병렬 가산기

4 비트를 더하는 가산기



[예제 4-6] $C_0=0$, $B=0011$, $A=1110$ 일 때, 합과 자리올림수는?

자리올림수:

입력 B

0 0 1 1

입력 A

1 1 1 0

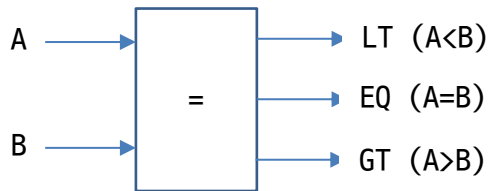
합:

4.2.2 비교기

문제 정의: 1 비트 비교기

- 두 비트를 비교하여
- '크다, 같다, 작다'를 출력하는 회로

입출력 변수



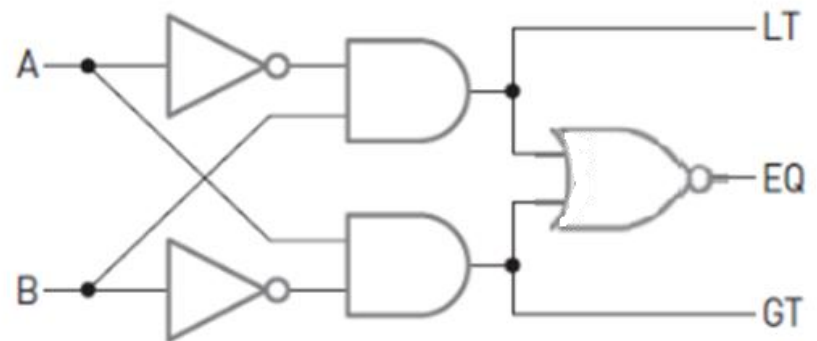
진가표

입력		출력			비고
A	B	LT	EQ	GT	
0	0				0 = 0
0	1				0 < 1
1	0				1 > 0
1	1				1 = 1

논리식

$$\begin{aligned}
 LT &= A'B && // A=0, B=1 \\
 EQ &= A'B'+AB = (A\oplus B)' && // AB=00 \text{ or } 11 \\
 &= (A'B + AB')' \\
 GT &= AB' && // A=1, B=0
 \end{aligned}$$

논리회로도



4.2.3 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

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

[예제 4-8] 짝수 패리티 문자 'A' 전송. 수신 데이터가 올바른가?

1) 수신 데이터 0_100_0001 ? _____

2) 수신 데이터 0_100_1001 ? _____

3) 수신 데이터 1_100_1001 ? _____

XOR 게이트

XOR: 홀수 함수

- 1의 수가 홀수일 때, 출력 1

짝수 패리티 생성기(generator)

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

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

짝수 패리티 검사기(checker)

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

산술 논리회로 요약

산술 회로

- 입력에 대한 출력이 항상 같음
- 따라서, 조합 논리회로

가산기

- 반가산기 (Carry, Sum) $\leftarrow A + B$
- 전가산기 (Carry, Sum) $\leftarrow A + B + C_{in}$
- 병렬 가산기 (Carry, $S_3S_2S_1S_0$) $\leftarrow A_3A_2A_1A_0 + B_3B_2B_1B_0$

비교기

- 1비트 비교기 (LT, EQ, GT) $\leftarrow (A==B)$
- 4비트 비교기 (LT, EQ, GT) $\leftarrow (A_3A_2A_1A_0 == B_3B_2B_1B_0)$

패리티 발생기/검사기

- XOR 게이트: 홀수 발생기

4.3 조합 논리회로 빌딩블록

빌딩블록

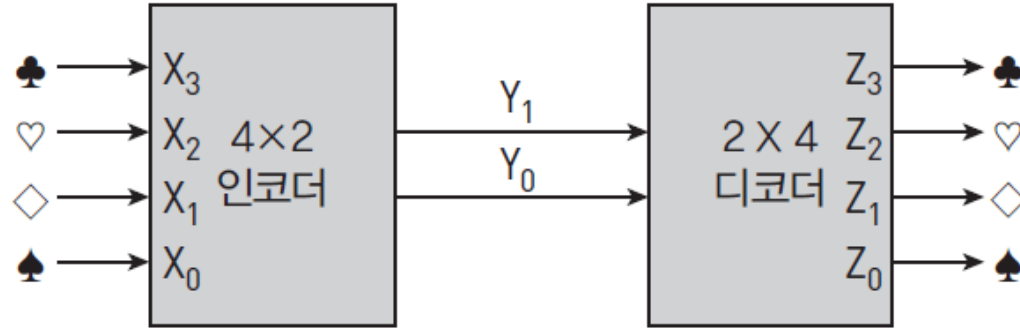
- 자주 사용되는 조합 논리회로

예)

- 인코더와 디코더
- 멀티플렉서와 디멀티플렉서

4.3.1 인코더와 디코더

집합의 원소	코드체계 3
♠	00
◇	01
♡	10
♣	11



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	♣

인코더

인코더(encoder): 집합의 원소(n개)에 대한 코드 ($\log_2 n$ 비트) 생성

4비트 인코더

- 입력의 모든 조합을 포함하지 않음
- 입력 신호 중 반드시 1이 하나
- 아니라면?

우선순위 인코더: 입력 신호에 우선순위 부여

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

[예제 4-9] 우선순위 인코더 출력은?

1) $X = 0110 \rightarrow Y = ______, V = ______$

2) $X = 0000 \rightarrow Y = ______, V = ______$

3) $X = 0001 \rightarrow Y = ______, V = ______$

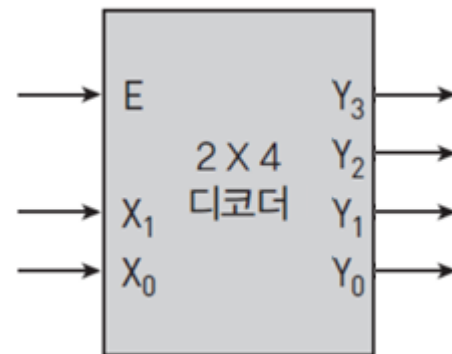
디코더

디코더(decoder)

- n비트 코드에 대한 2^n 비트 2진수(one-hot) 출력
- One-hot: 비트 중 하나만 값이 다른 2진수

Enable 제어선이 있는 디코더

E	X_1	X_0	Y_3	Y_2	Y_1	Y_0
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



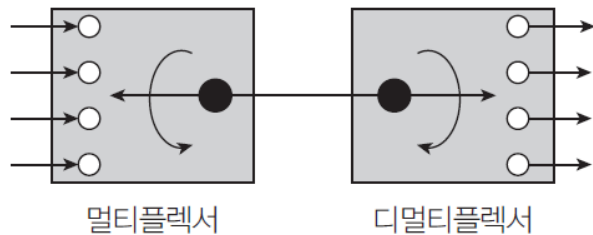
[예제 4-10] 출력은?

1) $E = 0, X = 01 \rightarrow Y = \underline{\hspace{2cm}}$

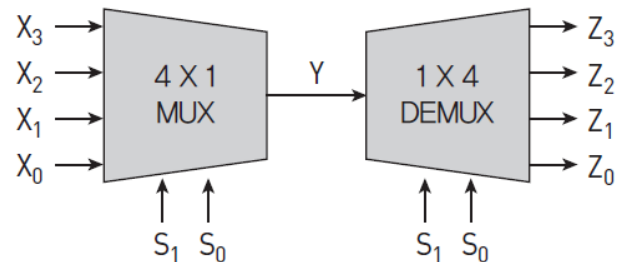
2) $E = 1, X = 01 \rightarrow Y = \underline{\hspace{2cm}}$

3) $E = 1, X = 11 \rightarrow Y = \underline{\hspace{2cm}}$

4.3.2 멀티플렉서/디멀티플렉서



(a) 가능



(b) 블럭도

멀티플렉서(multiplexer, MUX)

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

디멀티플렉서(demultiplexer)

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

[예제 4-11] 신호 전달

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

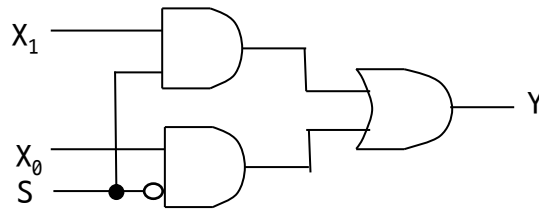
$$X_1 \rightarrow Y \rightarrow Z_3$$

멀티플렉서

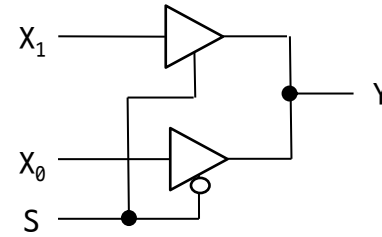
2×1 멀티플렉서 구현

S	Y
0	X_0
1	X_1

$$Y = S'X_0 + SX_1$$



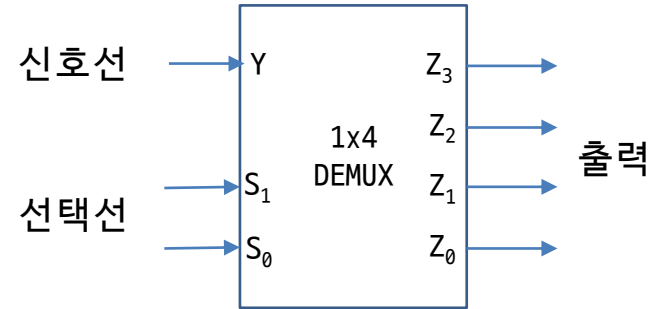
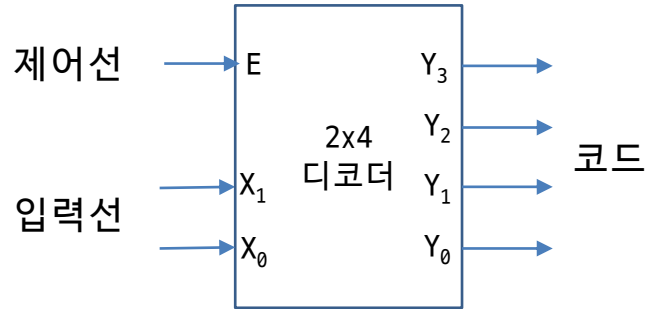
게이트 구현 2×1 MUX



세 상태 버퍼 구현 2×1 MUX

디멀티플렉서

디멀티플렉서 = 제어선이 있는 디코더



제어선이 있는 디코더						
E	X ₁	X ₀	Y ₃	Y ₂	Y ₁	Y ₀
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 ₁	S ₀	Z ₃	Z ₂	Z ₁	Z ₀
0	0	0	0	0	Y
0	1	0	0	Y	0
1	0	0	Y	0	0
1	1	Y	0	0	0

4.3 조합 논리회로 빌딩블록 요약

인코더

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

디코더

- 코드를 푸는 회로
- 기억장치 주소를 입출력장치 선택선으로 변환

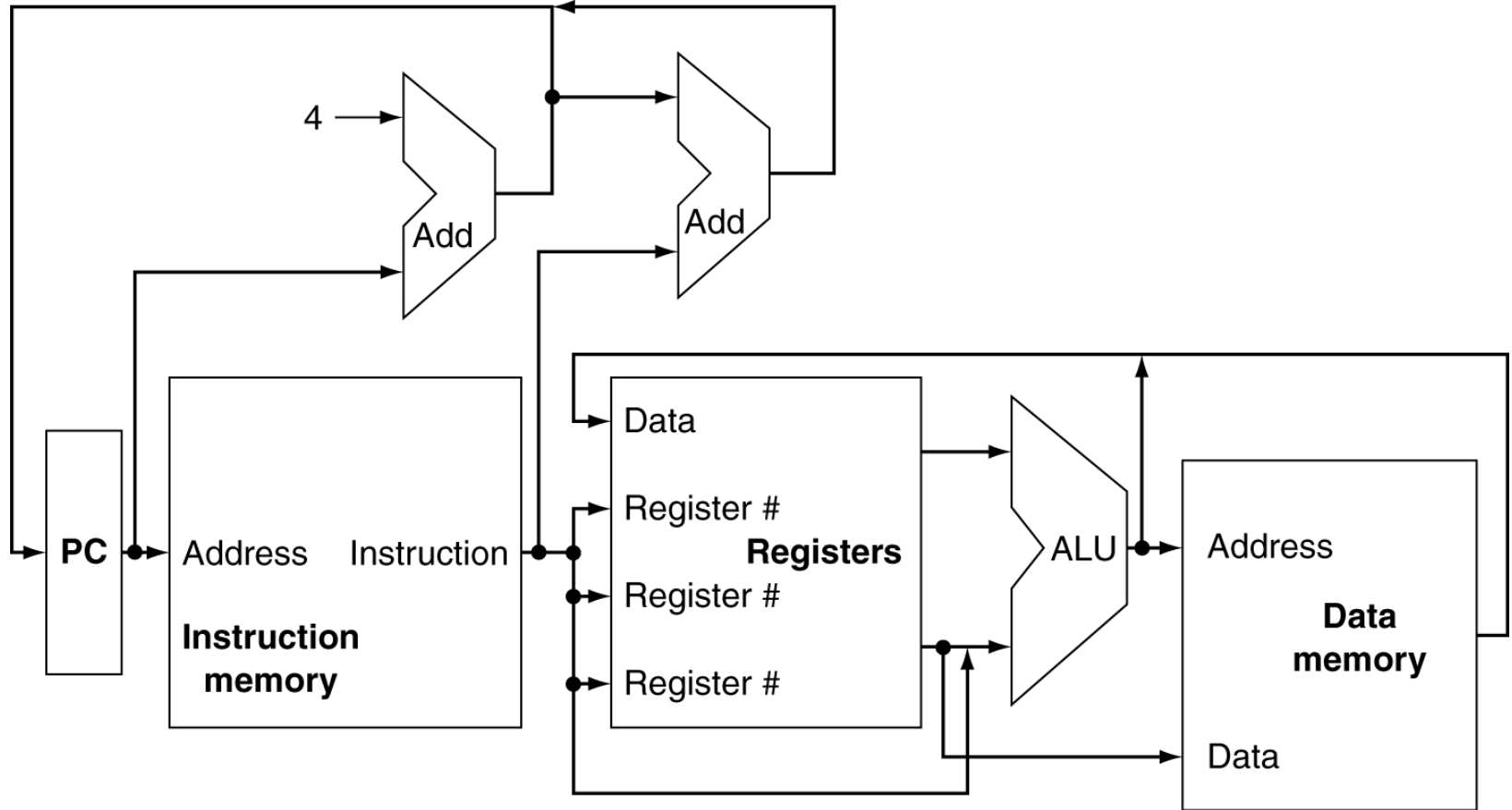
멀티플렉서

- 여러 개의 입력 중 하나를 선택
- 신호를 시스템 버스로 연결

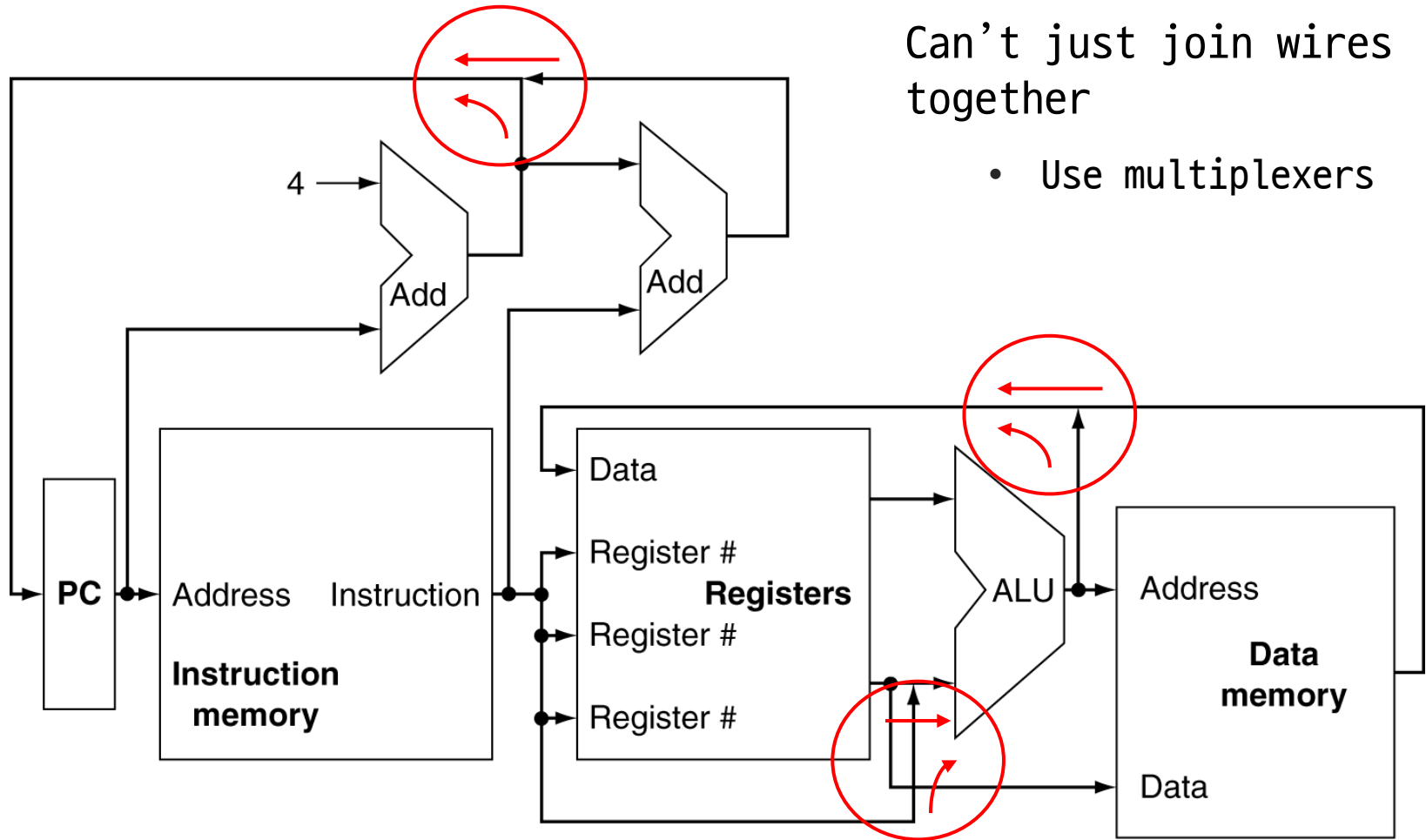
디멀티플렉서

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

CPU Overview



Multiplexers



Can't just join wires together

- Use multiplexers

Control

