

디지털 시스템

(제3장 논리회로의 묘사)-1/2

3

우 송 대 학 교
철도전기시스템학과
최 상 성

디지털 회로의 입/출력 표현

➤ 논리 레벨의 2진법 표현

- 0 [V] → 2진수 '0' 이라고 정의
- 5 [V] → 2진수 '1' 이라고 정의

➤ 입력 (Input) & 출력(Output)

- 입력 시스템에 '0' 또는 '1' 중의 하나가 입력 되는 것으로 간주
- 출력 시스템에 '0' 또는 '1' 중의 하나가 출력 되는 것으로 간주



디지털 시스템은 논리회로-논리 게이트로 구성
복잡한 회로의 동작은 부울 대수로 묘사하고 해석

3-1 부울 상수와 변수

➤ 부울 대수(Boolean Algebra)

- 논리회로의 **입력과 출력 사이의 관계**를 나타내는 방법 중 하나
- 부울 대수의 **상수와 변수**는 두 가지 값(“0” 과 “1”) 만 가짐
 - 상수(Constant): 값을 지정 받으면 변할 수 없음
 - 변수(Variable): 값을 지정 받아도 언제나 바꿀 수 있음

표 3-1 공통 논리 용어

논리 0	논리 1
거짓(False)	참(True)
꺼짐(Off)	켜짐(On)
낮음(Low)	높음(High)
부정(No)	긍정(Yes)
열린 스위치(Open switch)	닫힌 스위치(Closed switch)

➤ 부울 대수의 3가지 기본 논리 연산(Logic Operation)

- **OR, AND, NOT**

3-1 부울 상수와 변수

<학습성과 평가문제>

1. 주어진 응용 문제에서 필요한 입력 수 보다 더 많은 입력 개수가 있다. 기본으로 주어진 입력들이 LOW로 된다면 회로에는 아무런 영향을 주지 않는다. 주어진 입력들을 변수 혹은 상수 중 어느 것으로 처리해야 하는가? 이들 입력에 대한 변수 이름은 무엇인가?

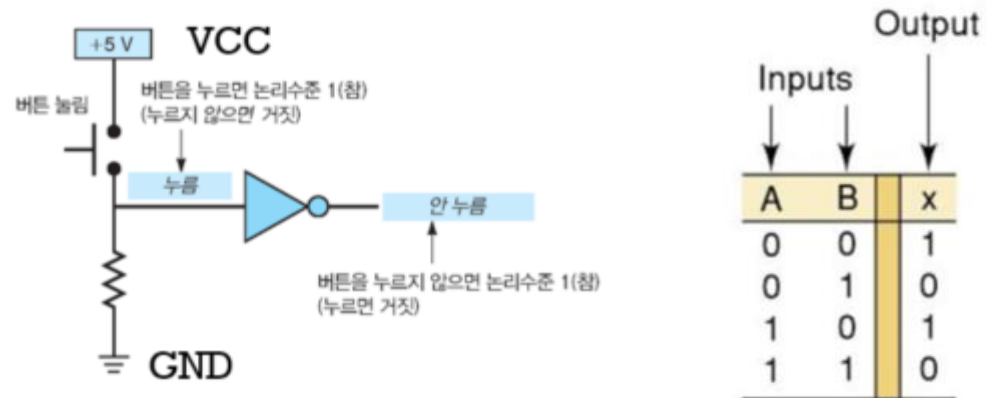
<풀이> 논리적으로 0이라는 상수를 표현할 때 보통 LOW 또는 GND로 표시 한다

2. 2장에서 설명한 2비트 입출력 인코더(비트 그레이 코드)는 논리회로 입력에 연결된 2개의 A와 B의 채널을 갖는다. 이들 입력이 변수 혹은 상수 중 어느 것으로 처리해야 하는가? 이들 입력에 대한 적당한 변수 이름은 무엇 인가?

<풀이> 변수, A, B

3. 부울 함수를 정의하라

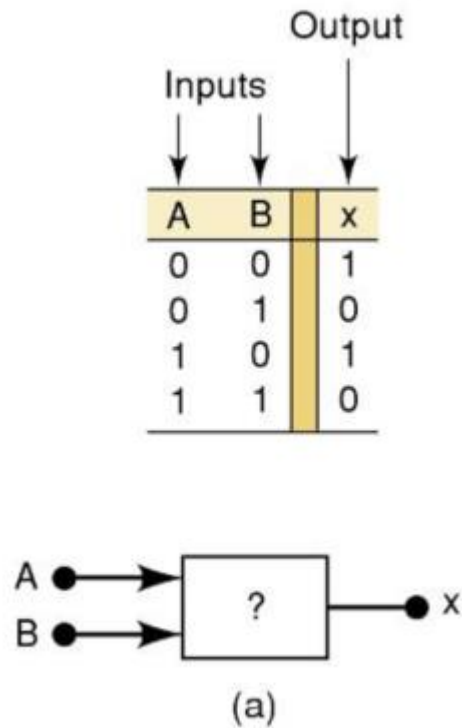
<풀이> 0 또는 1의 입력값들에 대하여
0 또는 1의 출력값을 갖는 함수이다.



3-2 진리표(Truth Table)

➤ 진리표

- 논리회로의 입력 단자에 가해진 논리값에 의해 회로의 출력이 어떻게 결정되는지 나타내는 수단



Inputs		Output
A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

3-2 진리표(Truth Table)

<학습성과 평가문제>

1. 그림 3-1(c)에 주어진 4입력 회로에서 B를 제외한 모든 입력이 1 일 때 출력 값은?

<풀이> $x=1$

2. $A=1, B=0, C=1, D=0$ 일 경우에 1번 문제의 출력 값은?

<풀이> $x=0$

3. 5 입력 진리표는 모두 몇 줄이 필요한가?

<풀이> $2^5 = 32$ 줄.

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)

3-3 OR 연산과 OR 게이트

➤ OR 연산의 부울 기호 표현

$X = A + B$ — Read as “X equals A OR B”

이 표현에서 + 기호는 일반 덧셈을 의미하는 것이 아니라 OR 연산을 나타내고 있다

➤ OR 연산

• A = 1, B = 1 이라 가정하면 OR 연산은

$$1 + 1 = 1 \quad (1 + 1 = 2 \text{ 아님})$$

부울기호 표현 $x = 1 + 1 + 1 = 1$
A=참 OR B=참 OR C=참 이면 x=참

3-3 OR 연산과 OR 게이트

➤ OR Gate

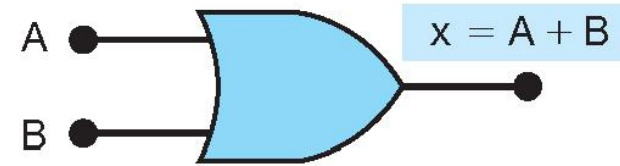
- 2개 이상의 입력을 가지고 있으며, **출력이 입력의 OR 조합과 같이 동작하게 만든 회로**

그림 3-2 (a) OR 연산의 진리표. (b) 2 입력 OR 게이트의 회로 기호.



OR		
A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(a)



OR 게이트

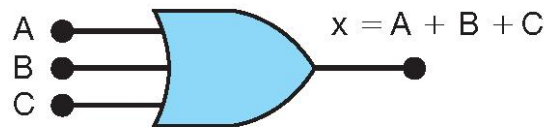
(b)

3-3 OR 연산과 OR 게이트

➤ OR Gate 연산 요약

- **OR 연산**은 입력 중 어느 하나라도 “1” 이 되면 결과(출력)는 “1” 이 된다
그렇지 않으면 출력은 “0” 이 된다
- **OR 게이트**는 회로의 입력에 대해 OR 연산을 수행하는 논리회로 이다
- $x = A + B$ 라는 표현은 “**x는 A OR B와 같다**” 라고 읽는다

그림 3-3 3 입력 OR 게이트
의 기호와 진리표.

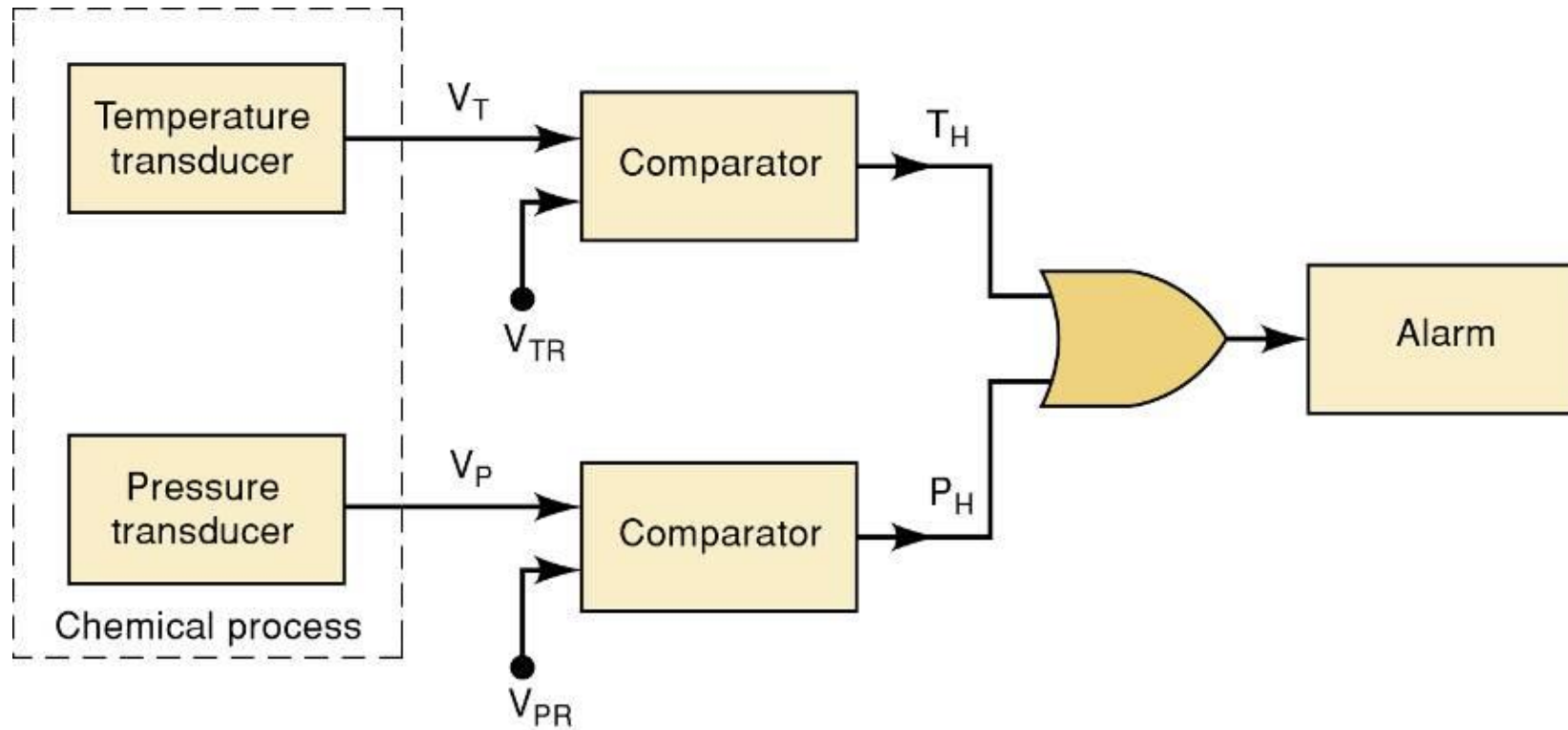


A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

* 게이트 항의 허용/금지 연산에 대한 자세한 내용은 4장에서 설명한다.

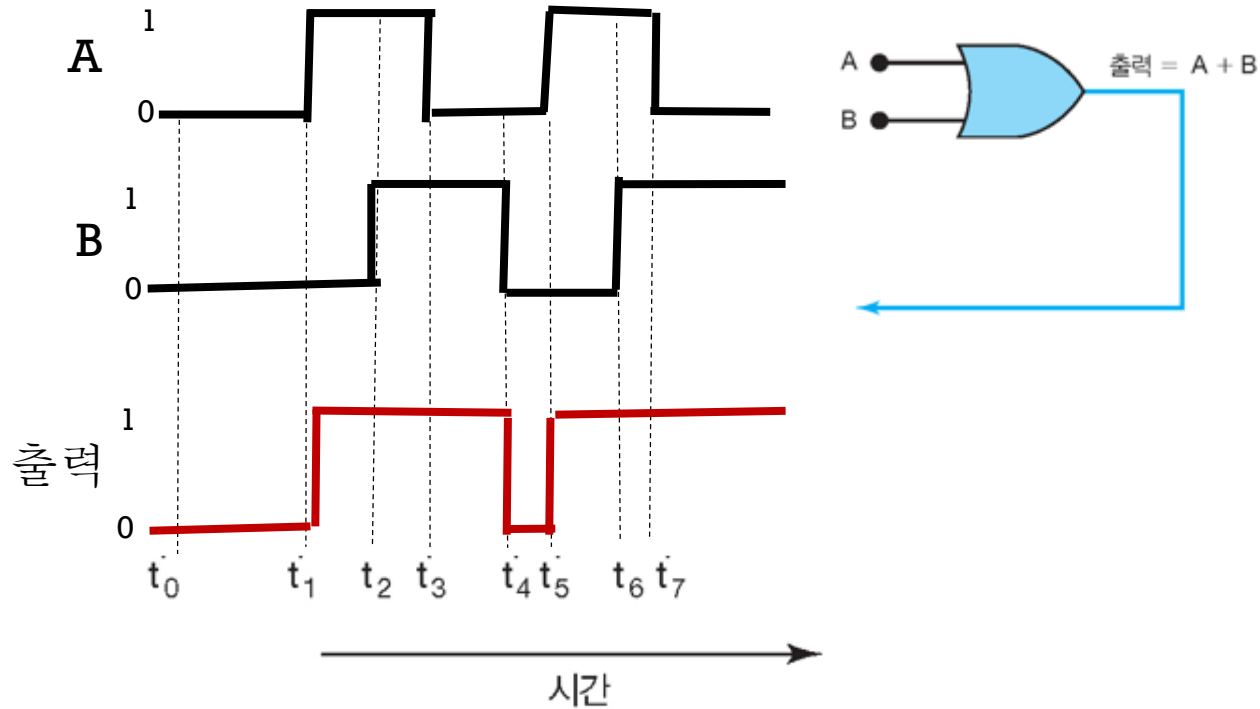
3-3 OR 연산과 OR 게이트

〈예제 3-1〉 Alarm System에서 OR Gate 사용 예



3-3 OR 연산과 OR 게이트

〈예제 3-2〉 그림 3-5에서 OR 게이트의 출력을 구하라

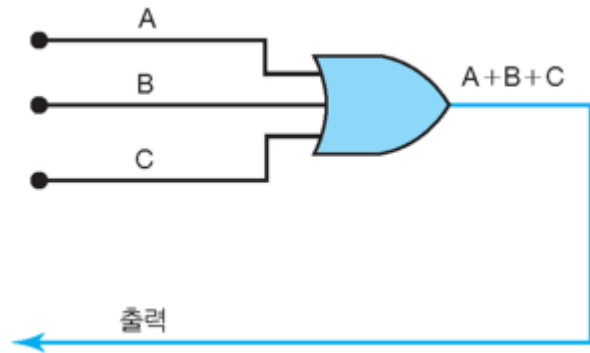
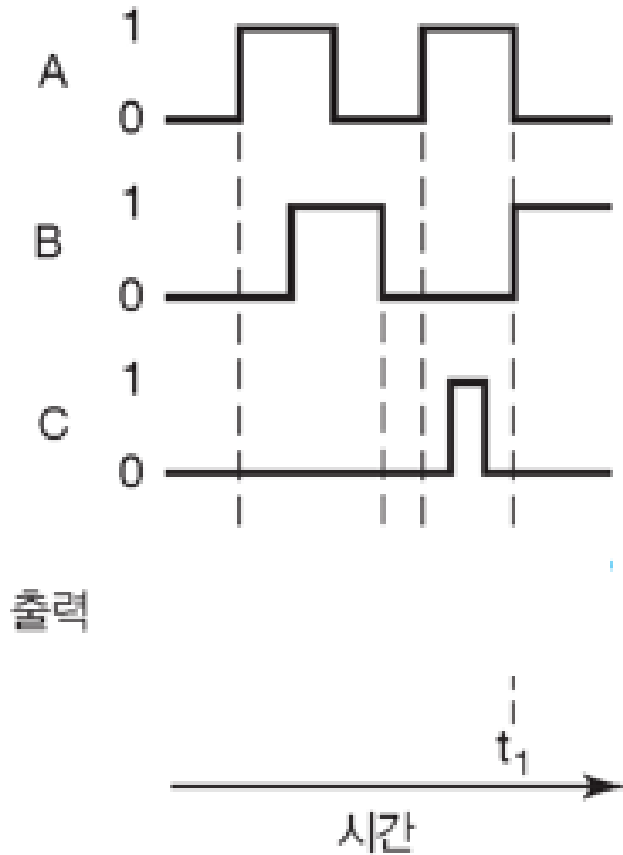


시간	입력		출력
t	A	B	A+B
$t_0 \sim t_1$	0	0	0
$t_1 \sim t_2$	1	0	1
$t_2 \sim t_3$	1	1	1
$t_3 \sim t_4$	0	1	1
$t_4 \sim t_5$	0	0	0
$t_5 \sim t_6$	1	0	1
$t_6 \sim t_7$	1	1	1

시간	입력		출력
t	A	B	A+B
$t_0 \sim t_1$			
$t_1 \sim t_2$			
$t_2 \sim t_3$			
$t_3 \sim t_4$			
$t_4 \sim t_5$			
$t_5 \sim t_6$			
$t_6 \sim t_7$			

3-3 OR 연산과 OR 게이트

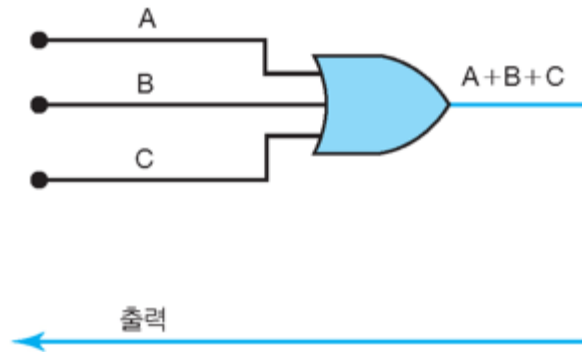
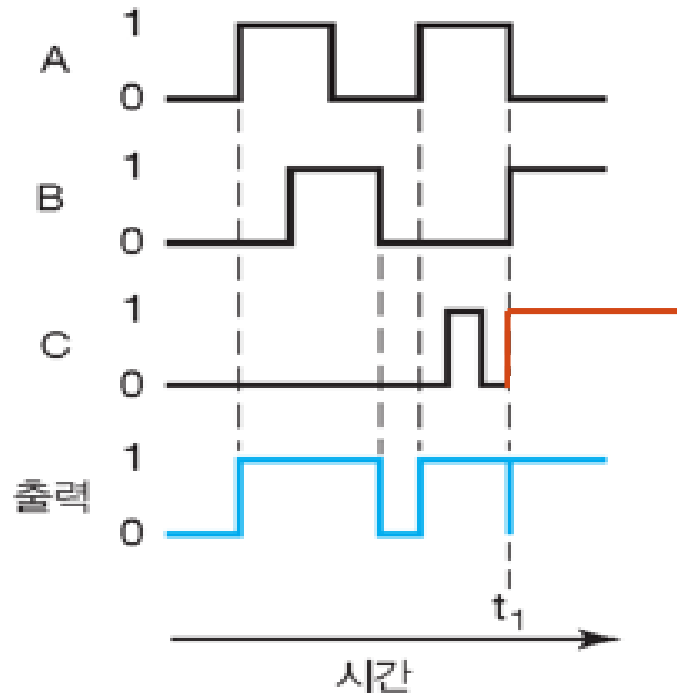
<예제 3-3A> 그림 3-6에서 OR 게이트의 출력을 구하라



입력			출력
A	B	C	A+B+C
0	0	0	0
1	0	0	1
1	1	0	1
0	1	0	1
0	0	0	0
1	0	0	1
1	0	1	1
1	0	0	1
0	1	0	1

3-3 OR 연산과 OR 게이트

〈예제 3-3B〉 그림 3-6에서 t_1 시간에 입력 A와 B가 전환 되면서 입력 C가 HIGH 상태를 유지 한다면 글리치는 어떻게 되겠는가?



입력			출력
A	B	C	A+B+C
0	0	0	0
1	0	0	1
1	1	0	1
0	1	0	1
0	0	0	0
1	0	0	1
1	0	1	1
1	0	0	1
0	1	1	1

3-3 OR 연산과 OR 게이트

<학습성과 평가문제>

1. OR 게이트 출력이 LOW를 유지하기 위한 단 한가지 입력 조건은 무엇인가?

<풀이> 입력이 모두 LOW를 유지하는 경우

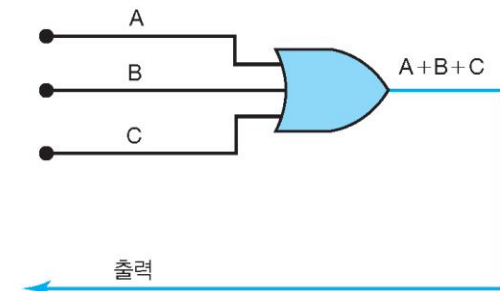
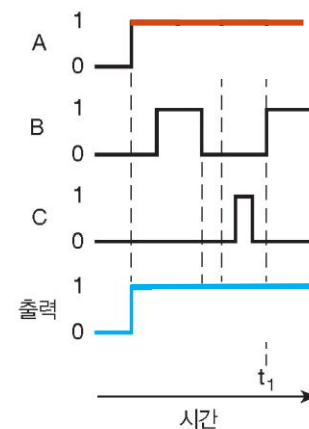
2. 6입 OR 게이트 부울 표현식을 써라

<풀이> $x = A+B+C+D+E+F$

3. 그림 3-6에서 입력 A가 계속 1 수준을 유지 한다면 출력 파형은 어떻게 되겠는가?

<풀이> 출력은 항상 HIGH

그림 3-6 예제 3-3A와 예제 3-3B.



3-4 AND 연산과 AND 게이트

➤ AND 연산의 부울 기호 표현

$$X = A \cdot B \cdot C \text{ — Read as “X equals A AND B AND C”}$$

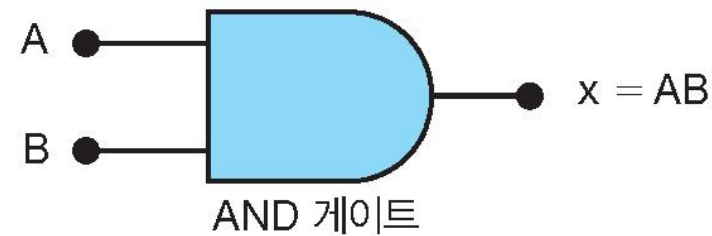
이 표현에서 \cdot 기호는 일반 곱셈을 의미하는 것이 아니라 AND 연산을 나타내고 있다
x is true (1) when A AND B are true (1)

➤ AND Gate

그림 3-7 (a) AND 연산의 진리표 (b) AND 게이트 기호.

AND		
A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

(a)



(b)

3-4 AND 연산과 AND 게이트

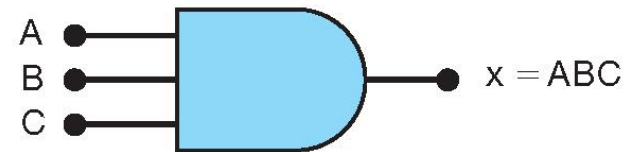
➤ AND Gate 연산 요약

- **AND 연산**은 “1” 과 “0” 사이의 일반 곱셈과 같이 수행한다
- **AND 게이트**는 회로의 입력에 대해 AND 연산을 수행하는 논리회로 이다
- AND 게이트의 **모든 입력이 “1” 일 때에만 출력이 “1”** 이 되며, 다른 경우에는 출력이 “0” 이다
- $x = A \cdot B$ 표현식은 “**x는 A AND B와 같다** “ 라고 읽는다

그림 3-8 3 입력 AND 게이트의 진리표와 기호.

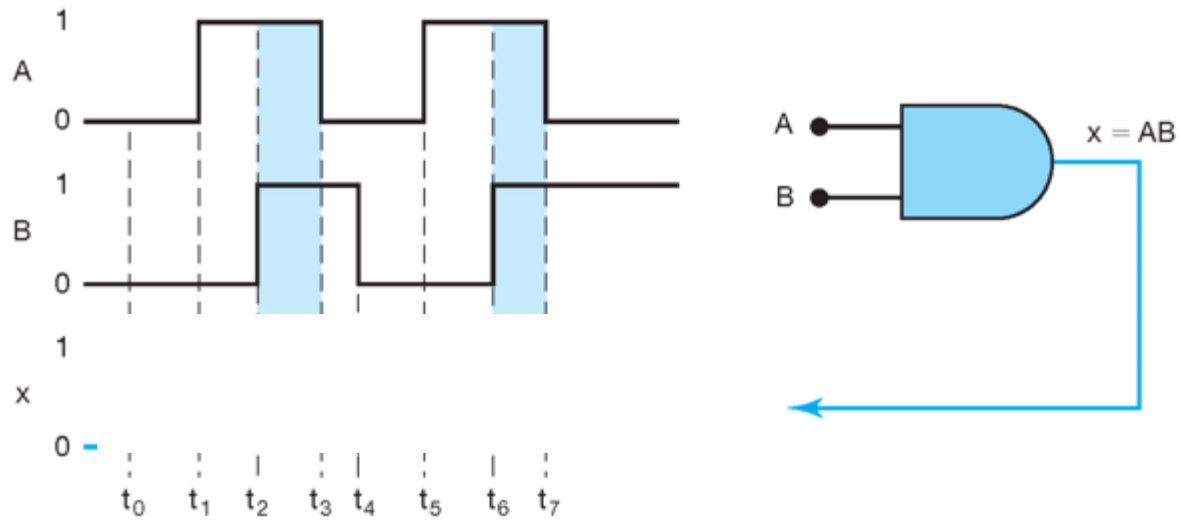


A	B	C	x = ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



3-4 AND 연산과 AND 게이트

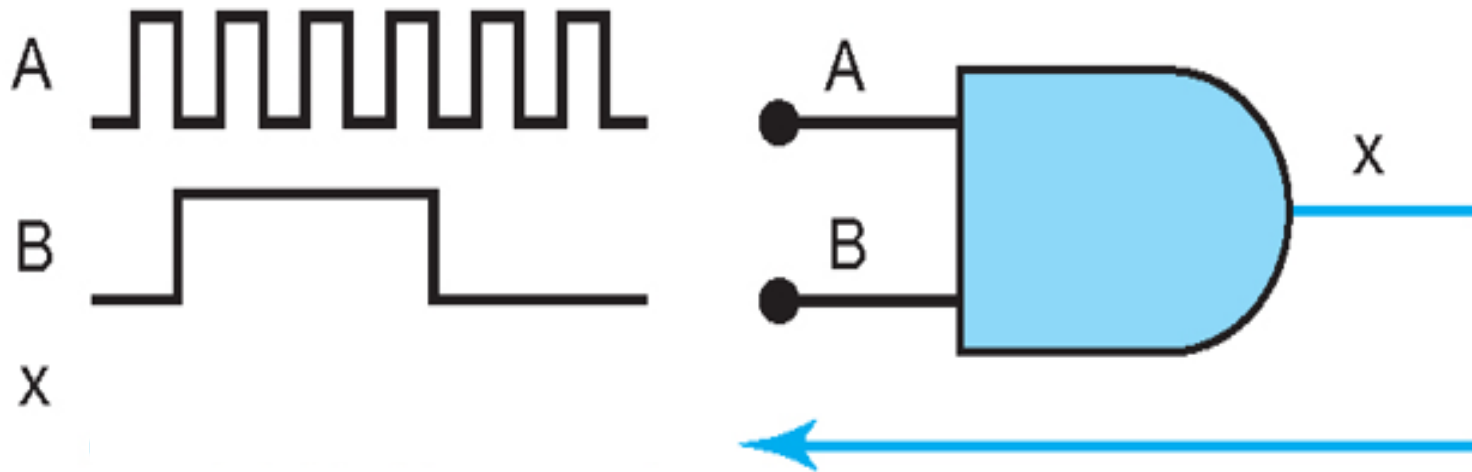
〈예제 3-4〉 그림 3-9의 AND 게이트에서 주어진 입력 파형에 대해 출력 x 의 파형을 구하라



시간	입력		출력
t	A	B	A·B
$t_0 \sim t_1$	0	0	0
$t_1 \sim t_2$	1	0	0
$t_2 \sim t_3$	1	1	1
$t_3 \sim t_4$	0	1	0
$t_4 \sim t_5$	0	0	0
$t_5 \sim t_6$	1	0	0
$t_6 \sim t_7$	1	1	1

3-4 AND 연산과 AND 게이트

〈예제 3-5A〉 그림 3-10과 같은 AND 게이트에 대한 출력 파형을 구하라

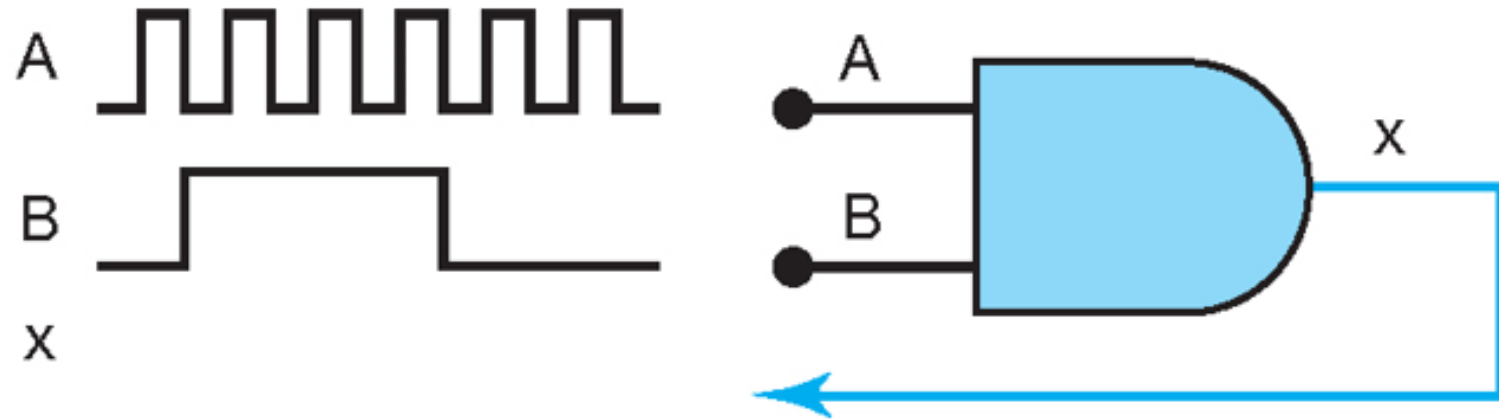


입력		출력
A	B	A·B
0	0	0
1	0	0
0	1	0
1	1	1
0	1	0
1	1	1
0	1	0
1	1	1
0	0	0
1	0	0
0	0	0

3-4 AND 연산과 AND 게이트

〈예제 3-5B〉 그림 3-10에서 입력 B가 계속 0 이라면 출력 파형 x는 어떻게 되겠는가?

입력		출력
A	B	A·B
0	0	0
1	0	0
0	0	0
1	0	0
0	0	0
1	0	0
0	0	0
1	0	0
0	0	0
1	0	0
0	0	0



입력 B가 계속 LOW 상태라면
출력 x는 계속 LOW에 머무른다

3-4 AND 연산과 AND 게이트

<학습성과 평가문제>

1. 5 입력 AND 게이트에서 출력 HIGH가 되는 입력의 조합은 무엇인가?

<풀이> 입력 모두가 HIGH(1)일 경우

2. 2 입력 AND 게이트에서 만일 첫번째 입력이 출력에 도달 하는 것을 금지 시키려면(방지 하려면) 두번째 입력에는 어떤 논리 값을 적용 해야 하는가?

<풀이> LOW 입력

3. 다음 문장의 참, 거짓을 구분 하라.

“AND 게이트의 출력은 같은 입력 조건이라면 OR 게이트의 출력과 항상 다르다”

<풀이> 거짓

AND		
A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

OR		
A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

3-5 NOT 연산과 NOT 게이트

➤ NOT 연산의 부울 기호 표현

$X = \bar{A}$ — Read as: “X equals NOT A”

“X equals the *inverse* of A”

$A' = \bar{A}$

“X equals the *complement* of A”

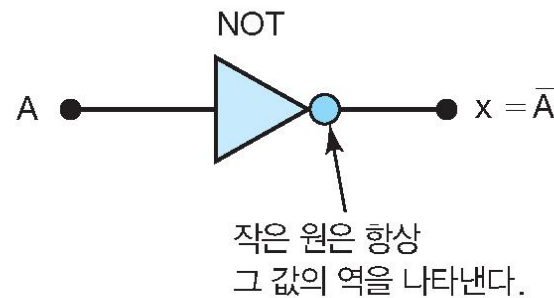
➤ NOT Gate

그림 3-11 인버터(NOT 회로)의 (a) 진리표, (b) 논리 기호, (c) 파형 예.

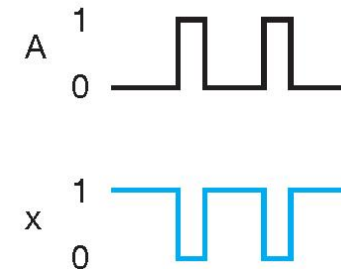


NOT	
A	$x = \bar{A}$
0	1
1	0

(a)



(b)

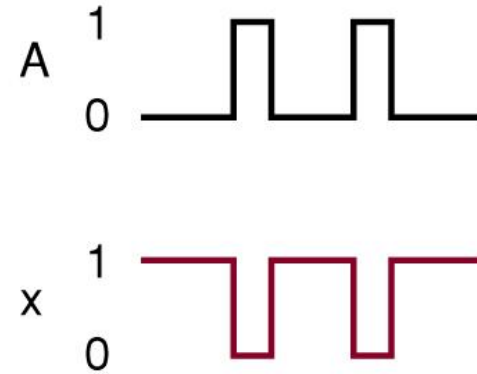
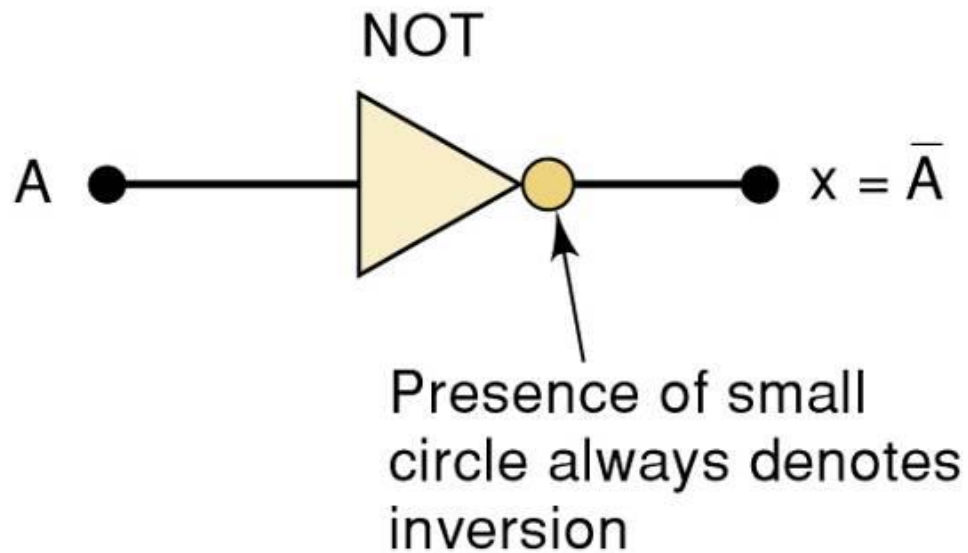


(c)

3-5 NOT 연산과 NOT 게이트

➤ NOT 게이트 연산 요약

- 일반적으로 **INVERTER**라고 부른다



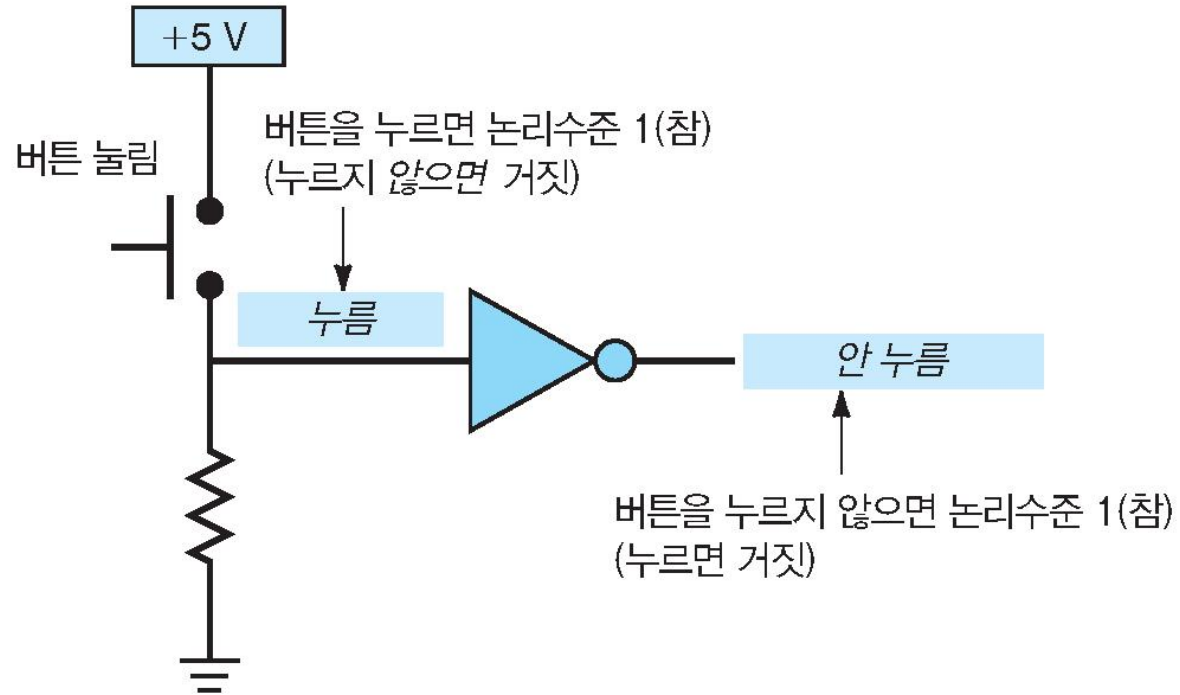
입력 $A = 1$ 이면 출력 $x = 0$
입력 $A = 0$ 이면 출력 $x = 1$

이 회로는 항상 하나의 입력을 가지며,
출력 논리 레벨은 항상 이 입력 논리 레벨의 반대이다

3-5 NOT 연산과 NOT 게이트

<응용 3-1> NOT Gate 응용 예

그림 3-12 출력이 참일 때 단
추가 눌러지지 않았다는 것을
의미하는 NOT 게이트.

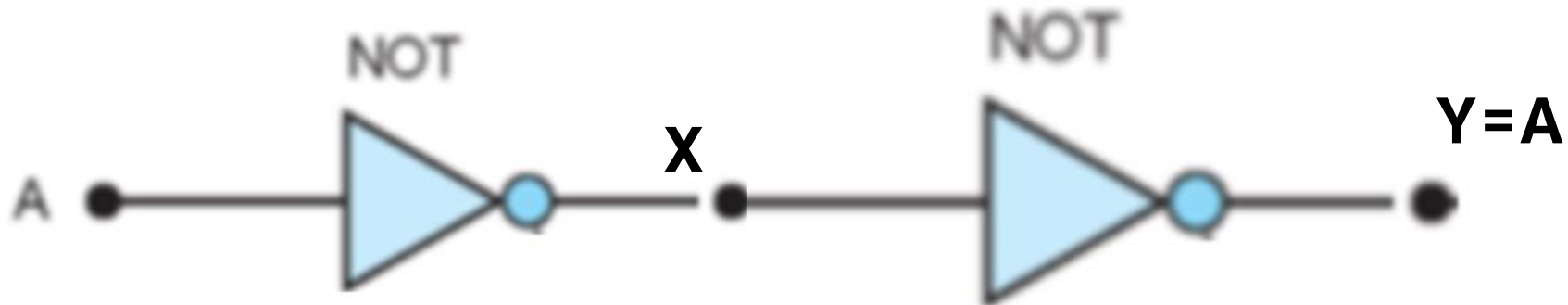


3-5 NOT 연산과 NOT 게이트

<학습성과 평가문제>

1. 그림 3-11에 있는 인버터의 출력은 두 번째 인버터의 입력에 연결되어 있다면 입력 a의 각 레벨에 대해 두 번째 인버터의 출력 값을 구하라.

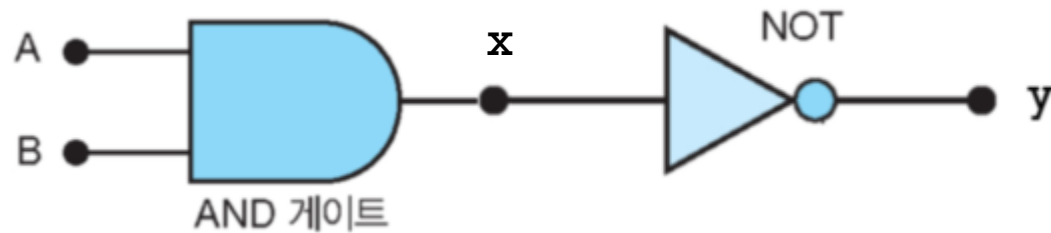
<풀이> 입력 A와 같다



3-5 NOT 연산과 NOT 게이트

<학습성과 평가문제>

2. 그림 3-7의 AND 게이트 출력은 인버터의 입력에 연결되어 있다고 가정한다
입력 A와 B의 모든 조합에 대해 인버터 출력 y값을 보여 주는 진리표를 작성하라



A	B	$X=A \cdot B$	$Y = \bar{X}$
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

A	B	$X=A \cdot B$	$Y = \bar{X}$

3-6 논리회로의 부울 대수 표현

➤ OR, AND, NOT 규칙 요약

<i>OR</i>	<i>AND</i>	<i>NOT</i>
$0 + 0 = 0$	$0 \cdot 0 = 0$	$\overline{0} = 1$
$0 + 1 = 1$	$0 \cdot 1 = 0$	$\overline{1} = 0$
$1 + 0 = 1$	$1 \cdot 0 = 0$	
$1 + 1 = 1$	$1 \cdot 1 = 1$	

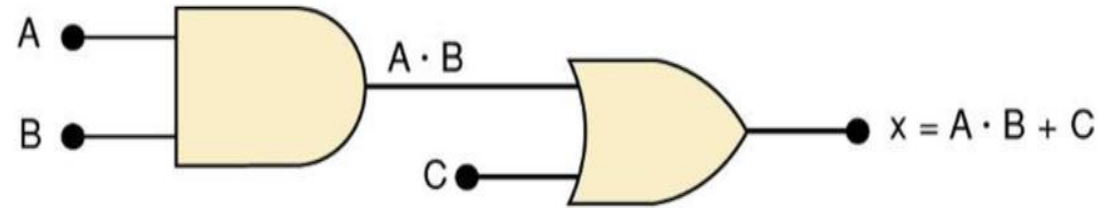
이 3가지 기초 부울 연산을 이용하여 어떠한 연산 회로도 표현할 수 있다

3-6 논리회로의 부울 대수 표현

➤ 연산 우선 순위

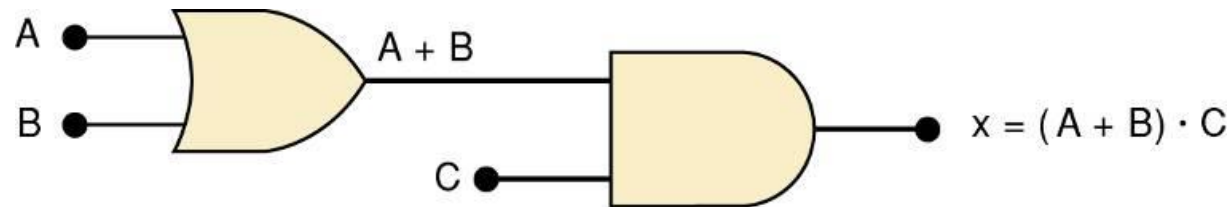
- AND 와 OR 게이트 모두 포함하고 있을 경우 AND 게이트를 먼저 수행 한다

$$x = A \cdot B + C$$



- ()가 있는 경우 ()를 먼저 수행 한다

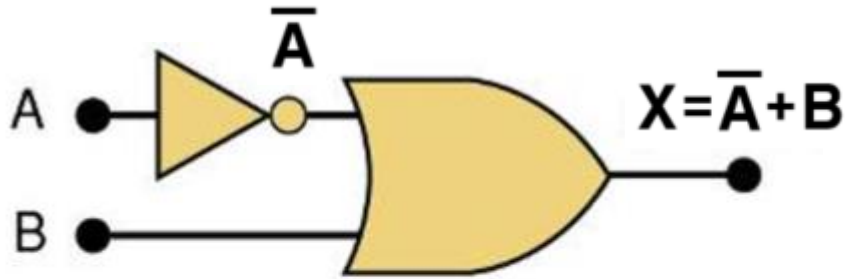
$$x = (A + B) \cdot C$$



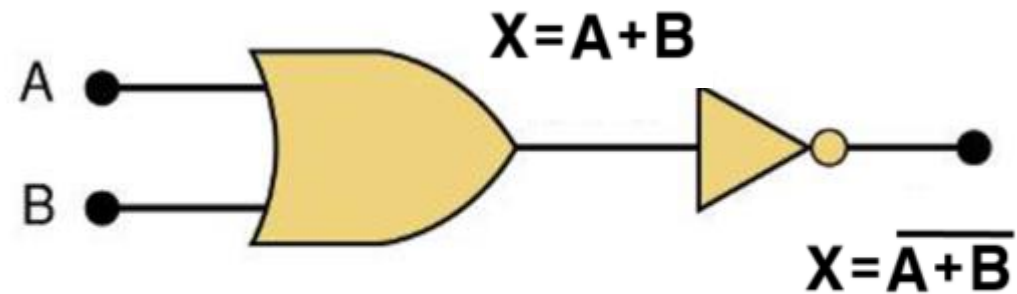
3-6 논리회로의 부울 대수 표현

➤ 논리회로에 인버터가 있는 경우

- 출력은 입력의 표현 위에 빗줄을 그은 것과 같다



(a)

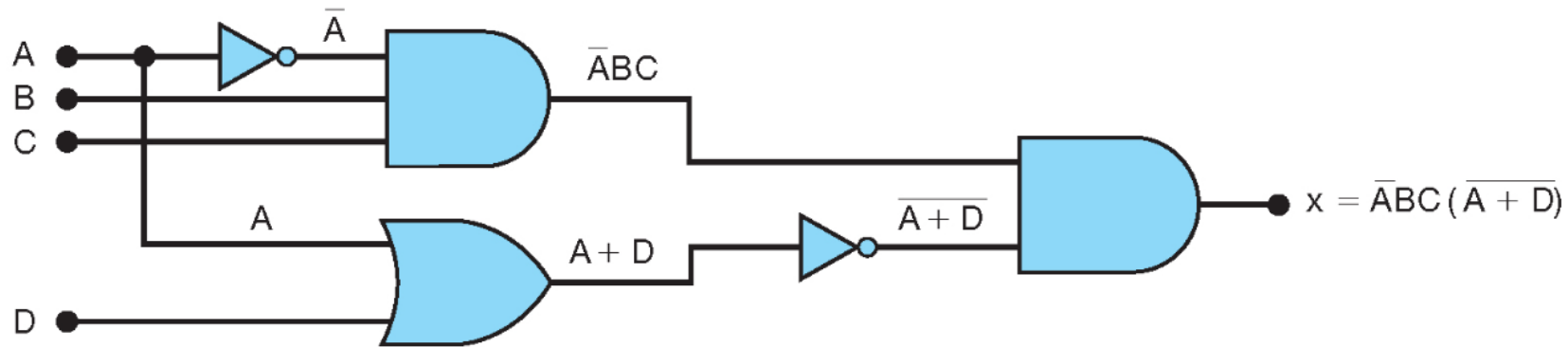


(b)

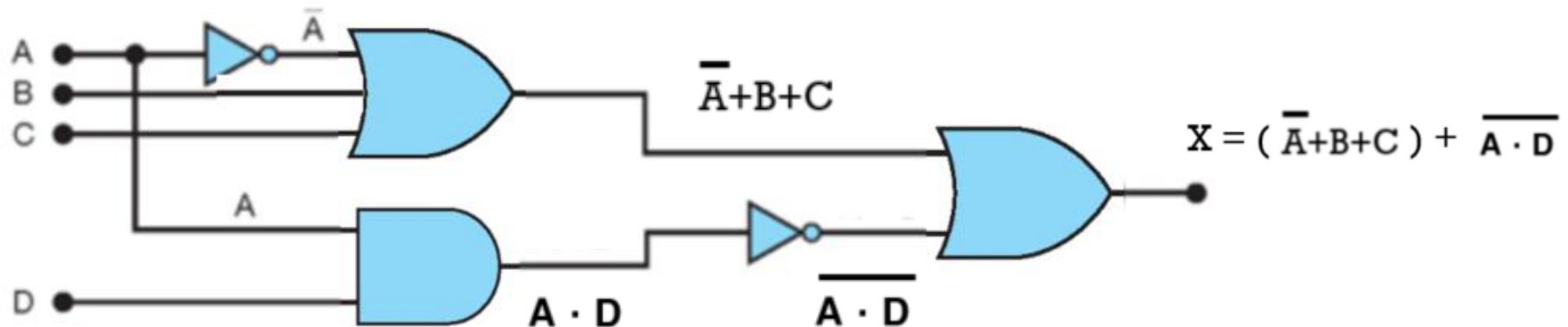
3-6 논리회로의 부울 대수 표현

<학습성과 평가문제>

- 그림 3-15의 (a)에서 AND 게이트를 OR 게이트로 바꾸고, OR 게이트를 AND 게이트로 바꾼 뒤, 출력 x의 부울 표현식을 써라



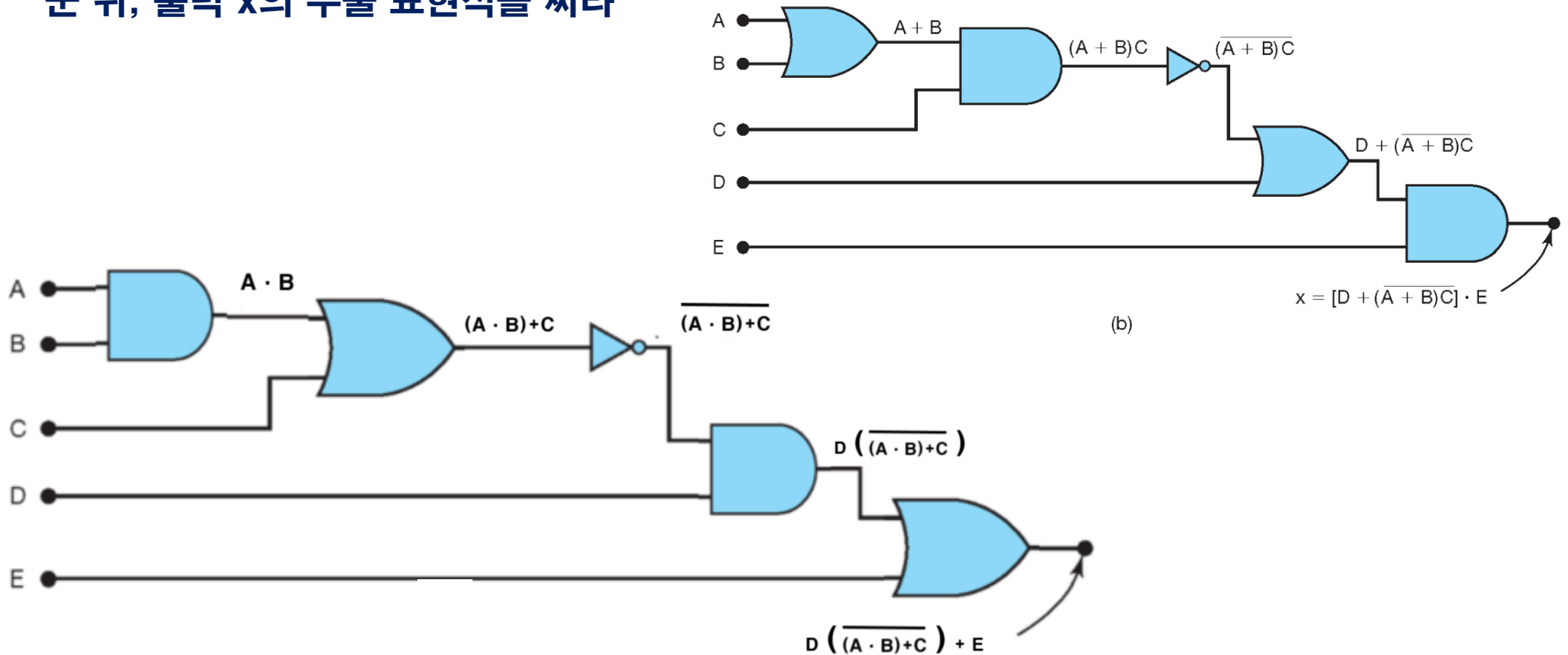
(a)



3-6 논리회로의 부울 대수 표현

<학습성과 평가문제>

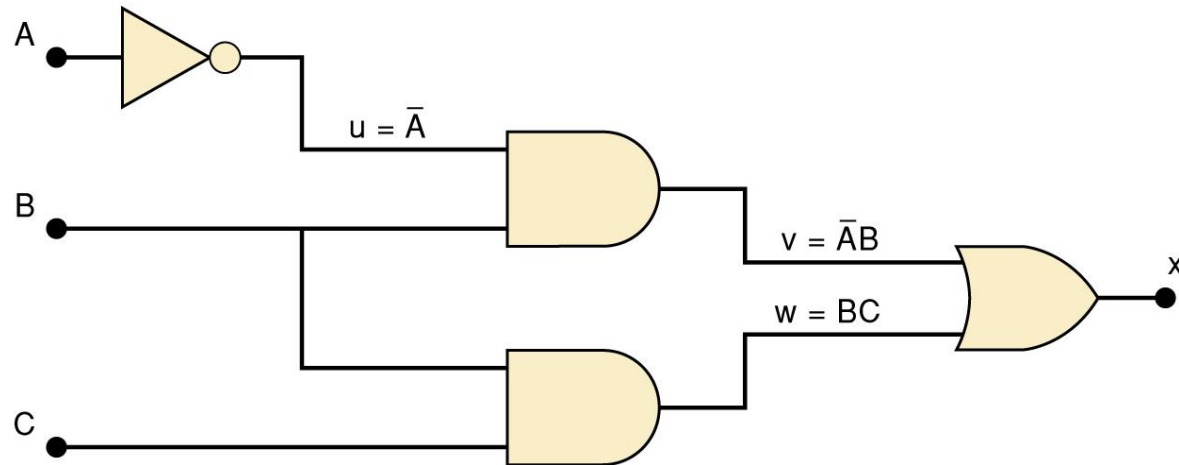
2. 그림 3-15의 (b)에서 AND 게이트를 OR 게이트로 바꾸고, OR 게이트를 AND 게이트로 바꾼 뒤, 출력 x의 부울 표현식을 써라



3-7 논리회로 출력값 구하기

표를 이용한 분석

- 우선 변수 한 개에 대한 역을 구한다($U=\bar{A}$)
- $v=\bar{A}B$ (\bar{A} 와 B가 모두 1 일 경우만 출력 $v=1$)
- $w = BC$ (B와 C 모두 1 일 경우만 출력 $w=1$)
- $x = v + w$ (입력 $v = 1$ OR $w = 1$ 이면 출력 $x=1$)



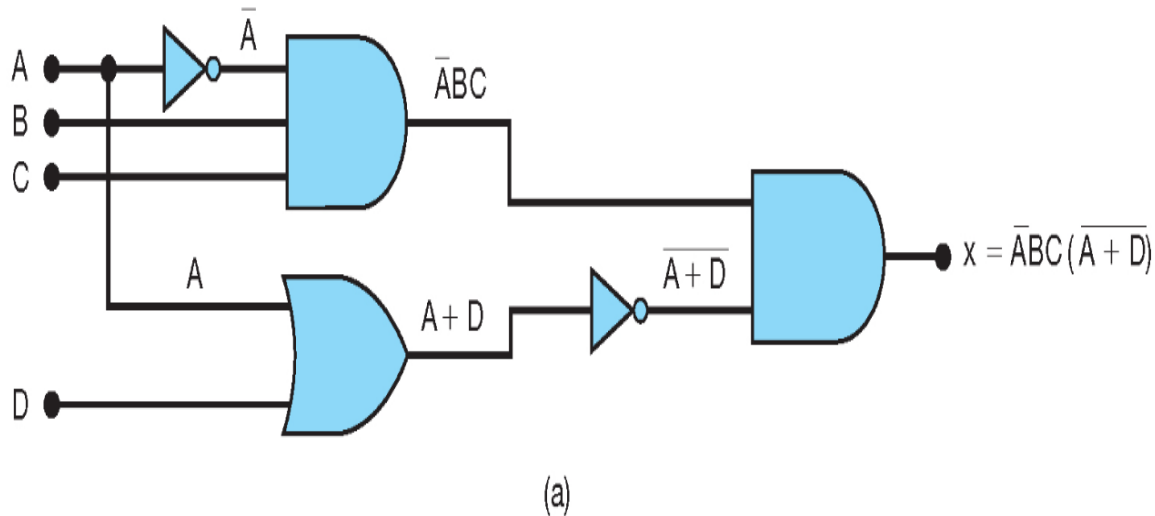
A	B	C	$\bar{u} = \bar{A}$	$\bar{v} = \bar{A}B$	$w = BC$	$x = v+w$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

A	B	C	U	V	W	X
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

3-7 논리회로 출력값 구하기

<예제 3-6>

그림 3-15(a) 에서 각각의 노드에 대하여 논리 상태를 나타내는 표를 만들어서 회로의 동작을 분석하라.



A	B	C	D	\bar{A}	$\bar{A}BC$	A+D	$\overline{A+D}$	$\bar{A}BC(\overline{A+D})$
0	0	0	0					
0	0	0	1					
0	0	1	0					
0	0	1	1					
0	1	0	0					
0	1	0	1					
0	1	1	0					
0	1	1	1					
1	0	0	0					
1	0	0	1					
1	0	1	0					
1	0	1	1					
1	1	0	0					
1	1	0	1					
1	1	1	0					
1	1	1	0					

3-7 논리회로 출력값 구하기

<예제 3-6>

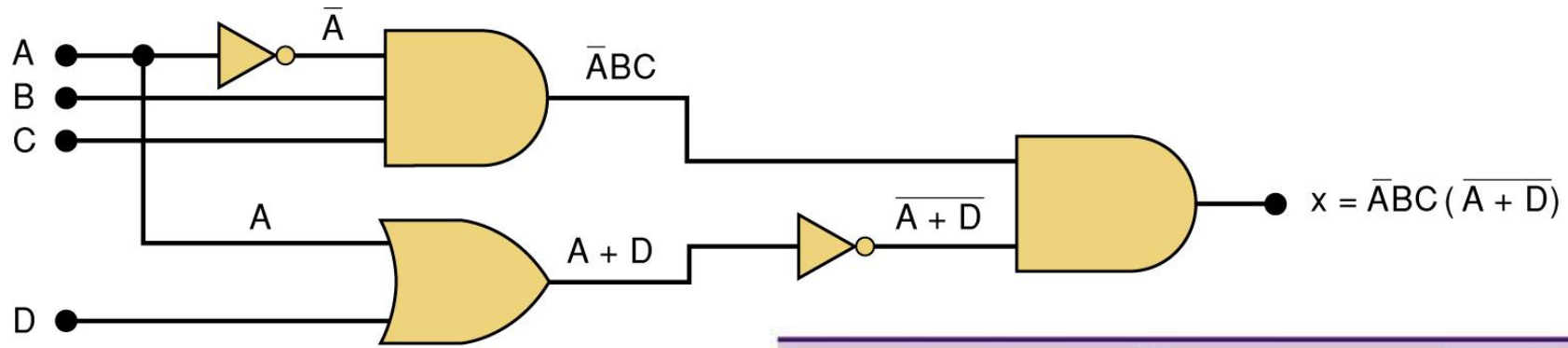


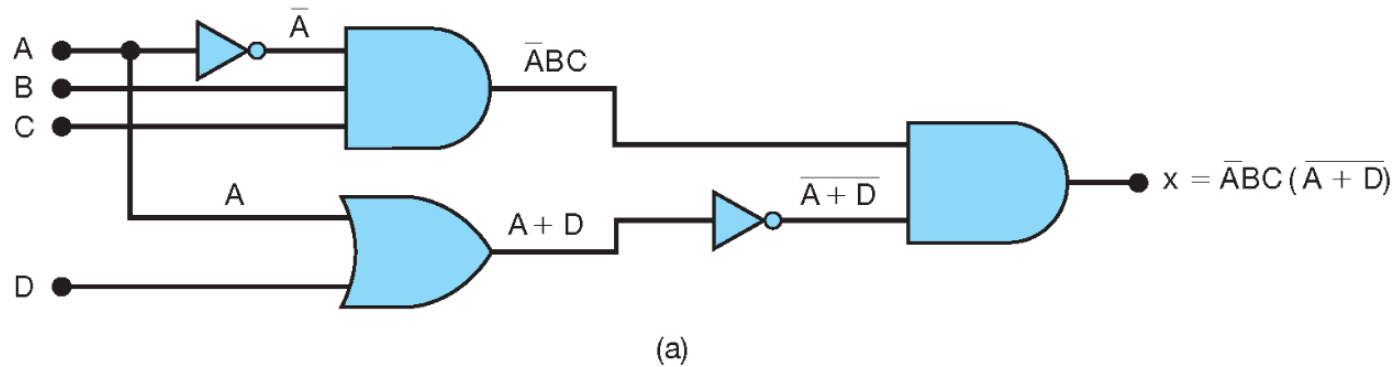
Table of logic state at each node of the circuit shown.

A	B	C	D	$t = \bar{A}BC$	$u = A + D$	$v = \overline{A + D}$	$x = tv$
0	0	0	0	0	0	1	0
0	0	0	1	0	1	0	0
0	0	1	0	0	0	1	0
0	0	1	1	0	1	0	0
0	1	0	0	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	0	1	0	1	1
0	1	1	1	1	1	0	0
1	0	0	0	0	1	0	0
1	0	0	1	0	1	0	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	0	0
1	1	0	0	0	1	0	0
1	1	0	1	0	1	0	0
1	1	1	0	0	1	0	0
1	1	1	1	0	1	0	0

3-7 논리회로 출력값 구하기

<학습성과 평가문제>

- 그림 3-15(a)에서 입력 $A=0, B=1, C=1, D=0$ 조건일 때 부울 표현식을 사용하여 출력 x 값을 구하라



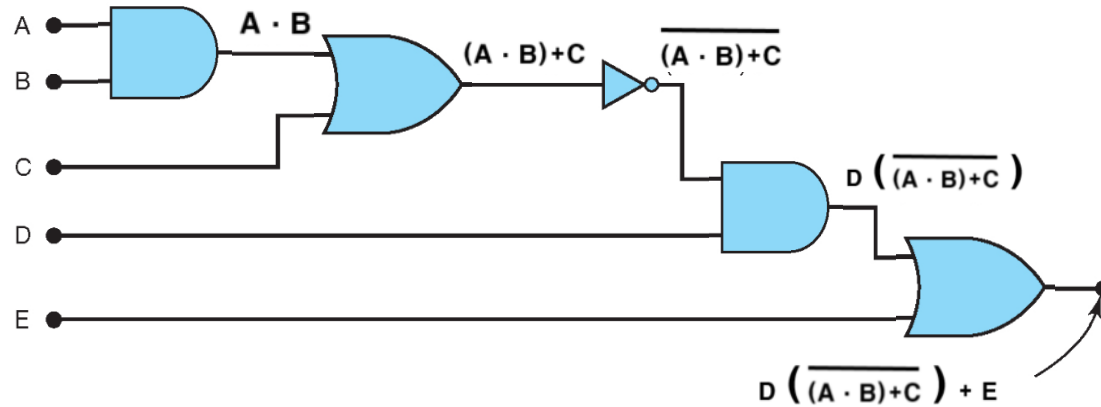
A	B	C	D	\bar{A}	$\bar{A}BC$	$A+D$	$\overline{A+D}$	$\bar{A}BC(\overline{A+D})$
0	1	1	0	1	1	0	1	1

A	B	C	D	\bar{A}	$\bar{A}BC$	$A+D$	$\overline{A+D}$	$\bar{A}BC(\overline{A+D})$
0	1	1	0					

3-7 논리회로 출력값 구하기

<학습성과 평가문제>

2. 그림 3-15(b)에서 $A=B=E=1, C=D=0$ 조건일 때 부울 표현식을 사용하여 출력 x 값을 구하라



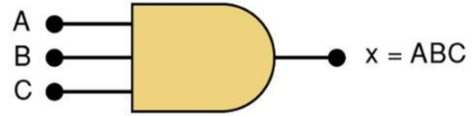
A	B	C	D	E	$A \cdot B$	$(A \cdot B) + C$	$\overline{(A \cdot B) + C}$	$D \cdot \overline{(A \cdot B) + C}$	$D \cdot \overline{(A \cdot B) + C} + E$
1	1	0	0	1	1	1	0	0	1

A	B	C	D	E	$A \cdot B$	$(A \cdot B) + C$	$\overline{(A \cdot B) + C}$	$D \cdot \overline{(A \cdot B) + C}$	$D \cdot \overline{(A \cdot B) + C} + E$
1	1	0	0	1			0	0	1

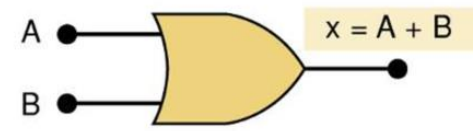
3-8 부울 표현식으로 부터 회로 구현

➤ AND, OR, NOT Gate 그리기

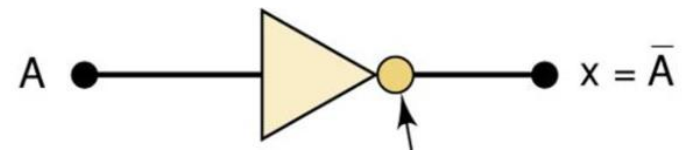
- $x = A \cdot B \cdot C$ 표현식은 3개 입력 AND 게이트를 그릴 수 있다



- $x = A + B$ 표현식은 2개 입력의 OR 게이트를 그릴 수 있다



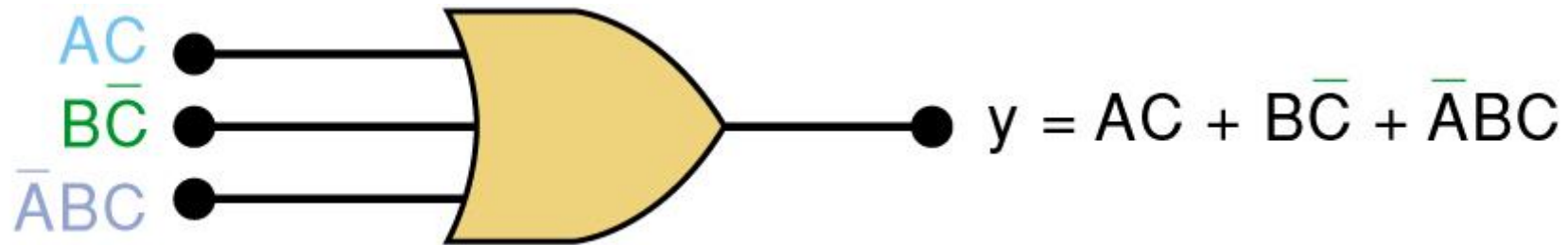
- $x = \bar{A}$ 표현식은 1개 입력의 NOT 게이트를 그릴 수 있다



3-8 부울 표현식으로 부터 회로 구현

- 아래 부울 표현식은 3개의 입력(AC , $B\bar{C}$, $\bar{A}BC$)을 가진 OR Gate로 그릴 수 있다

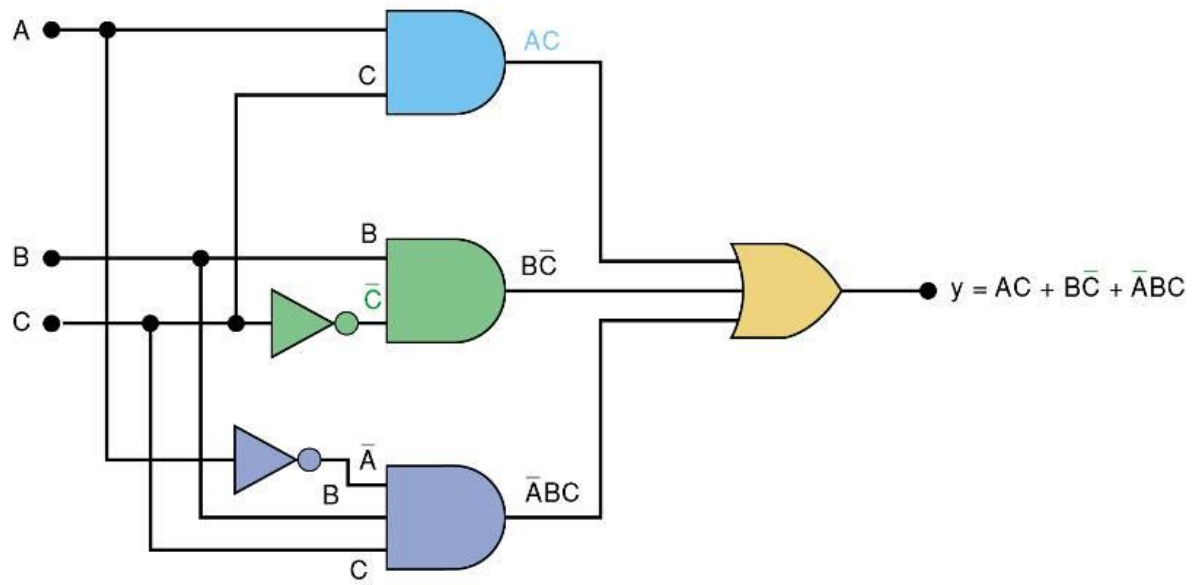
$$y = AC + B\bar{C} + \bar{A}BC$$



3-8 부울 표현식으로 부터 회로 구현

- 아래 부울 표현식은 OR Gate 입력으로 AND Gate 형태의 3개 입력을 가진 형태로 그릴 수 있다

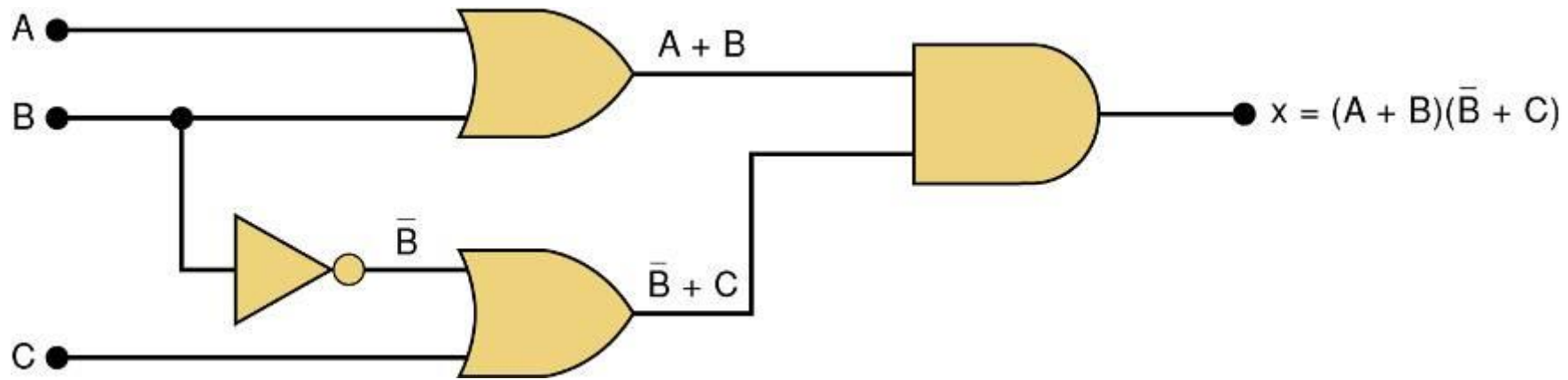
$$y = AC + B\bar{C} + \bar{A}BC$$



3-8 부울 표현식으로 부터 회로 구현

< 예제 3-7 > 아래 표현식을 수행하는 회로도를 그려라

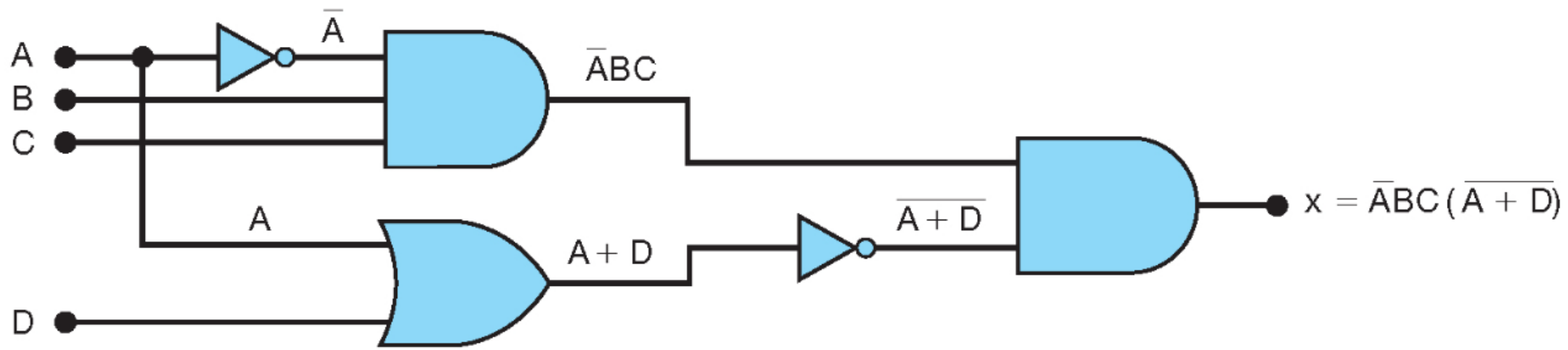
$$x = (A + B) (\bar{B} + C)$$



3-8 부울 표현식으로 부터 회로 구현

<학습성과 평가문제>

1. $x = \bar{A}BC(\overline{A+D})$ 표현식을 수행하는 회로를 3 입력 이하의 게이트를 사용하여 회로도를 그려라

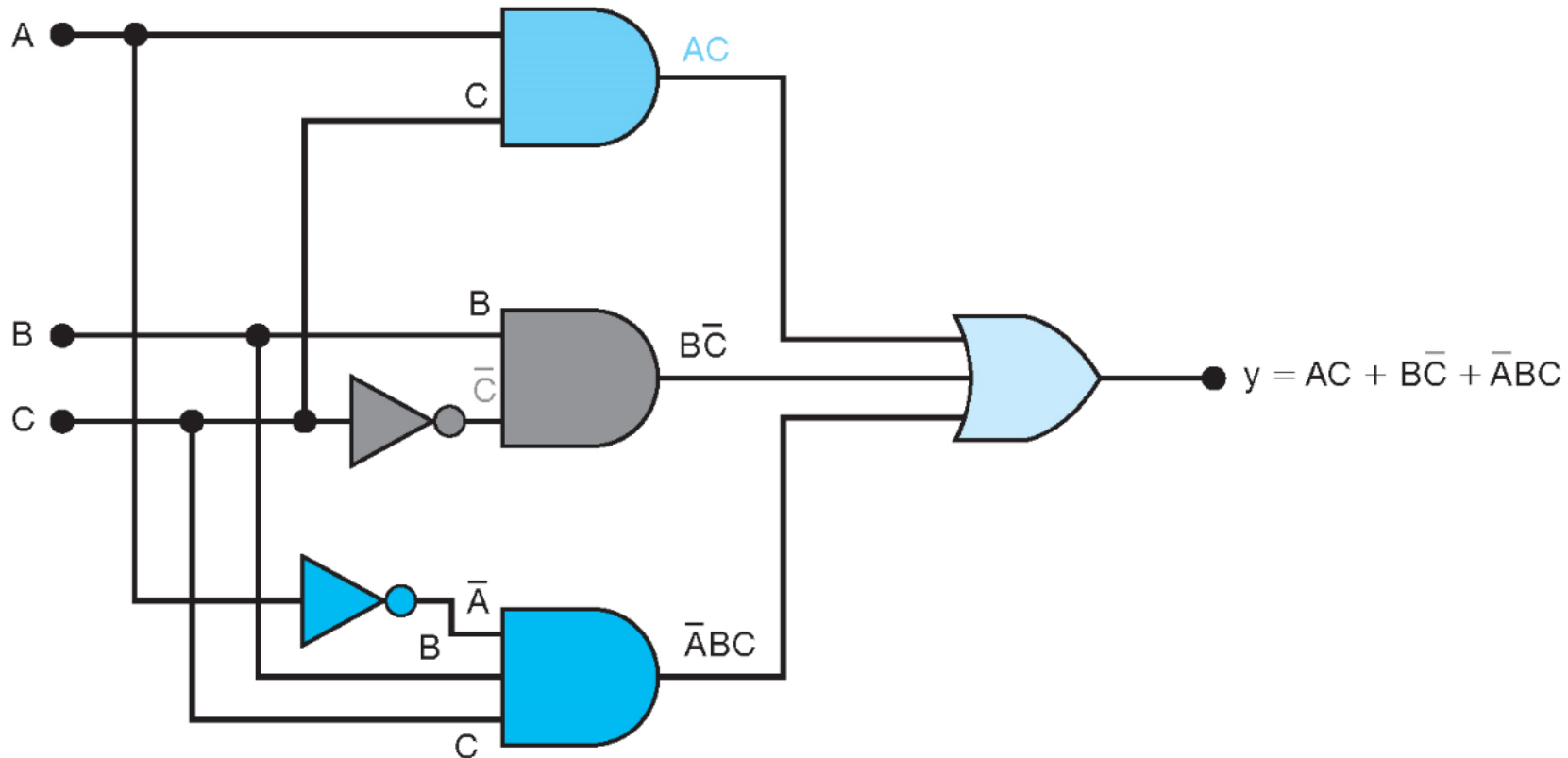


(a)

3-8 부울 표현식으로 부터 회로 구현

<학습성과 평가문제>

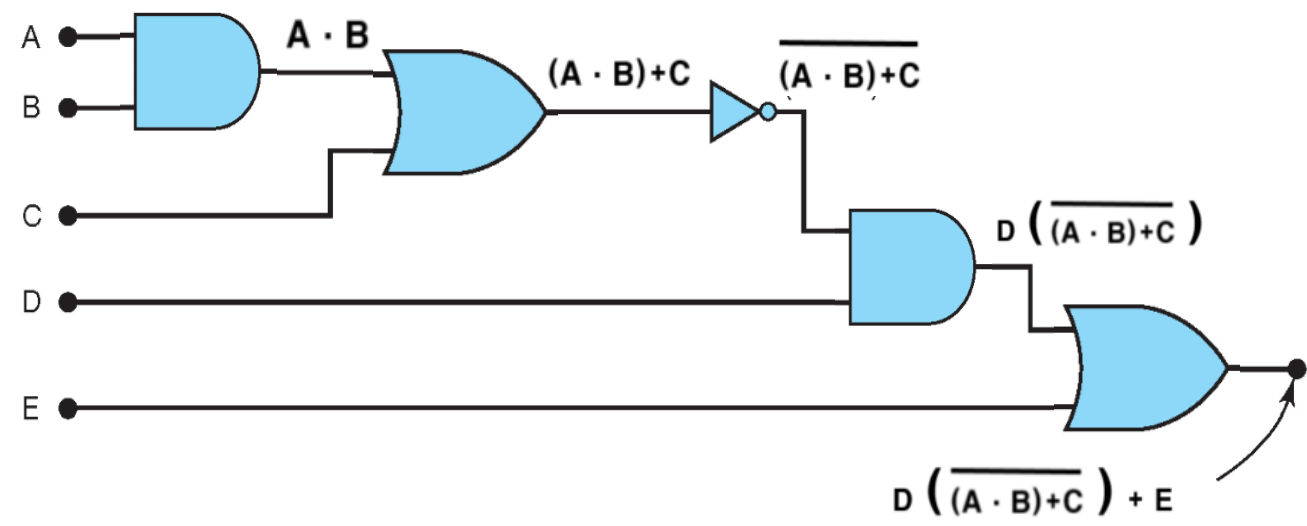
2. $y = AC + B\bar{C} + \bar{A}BC$ 표현식을 위한 회로도를 그려라



3-8 부울 표현식으로 부터 회로 구현

<학습성과 평가문제>

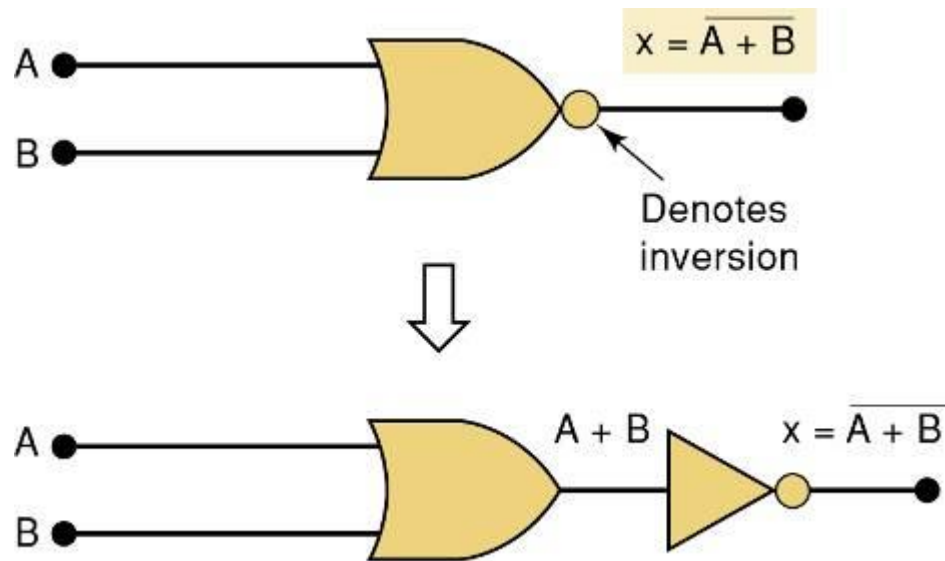
3. $x = D(\overline{(A \cdot B) + C}) + E$ 표현식을 위한 회로도를 그려라



3-9 NOR 게이트와 NAND 게이트

➤ NOR 게이트

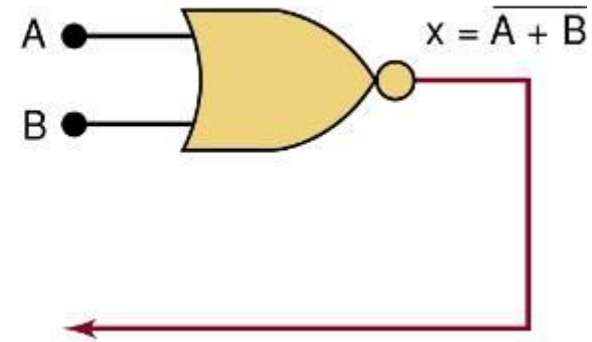
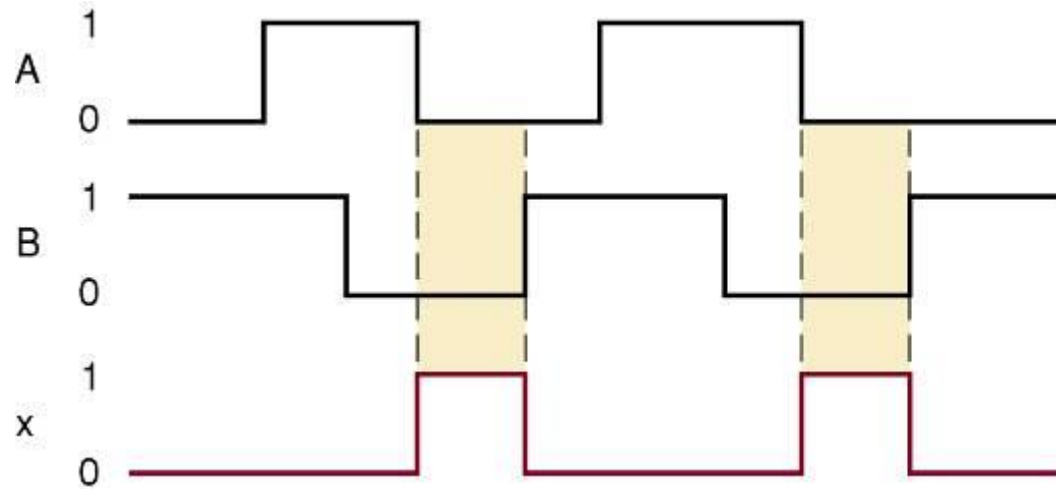
- NOR 게이트는 OR 게이트와 NOT 게이트가 혼합된 형태



A	B	OR		NOR	
		A + B	$\overline{A + B}$		
0	0	0	1		
0	1	1	0		
1	0	1	0		
1	1	1	0		

3-9 NOR 게이트와 NAND 게이트

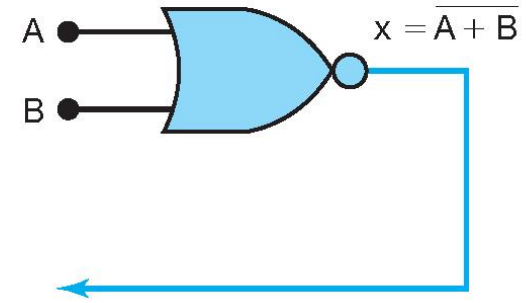
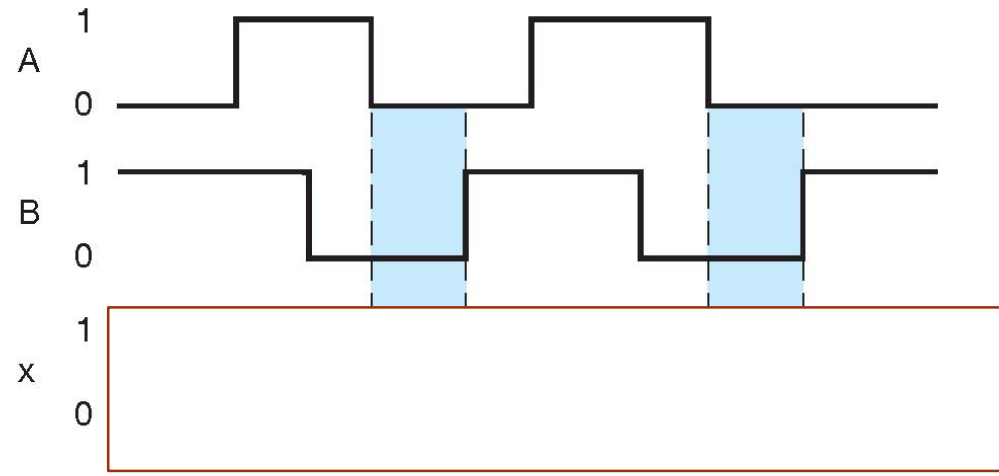
➤ NOR 게이트 입/출력 파형



3-9 NOR 게이트와 NAND 게이트

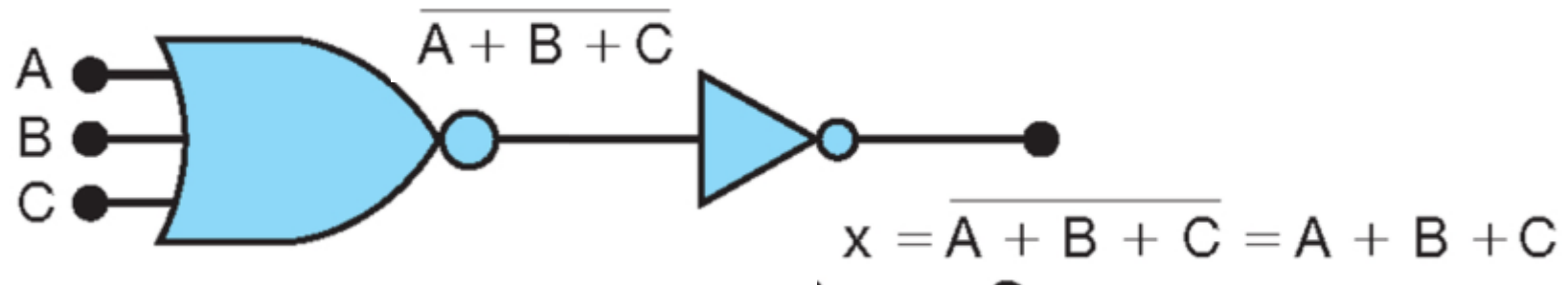
< 예제 3-8 > 그림 3-20에서 보인 파형에 대하여 NOR 게이트의 출력 파형을 구하라

그림 3-20 예제 3-8.



3-9 NOR 게이트와 NAND 게이트

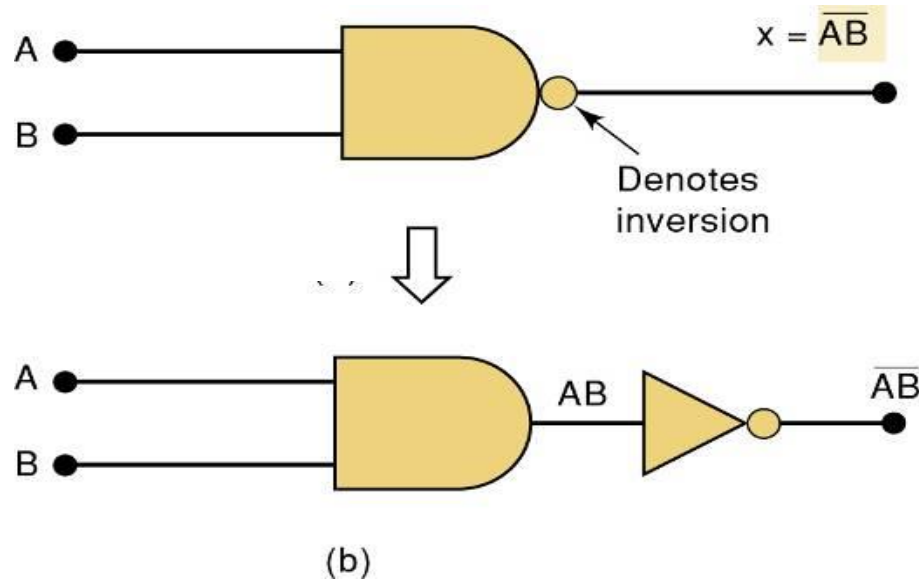
< 예제 3-9 > 3 입력 NOR 게이트 뒤에 인버터가 있는 경우에 대한 부울 표현식을 구하라



3-9 NOR 게이트와 NAND 게이트

➤ NAND 게이트

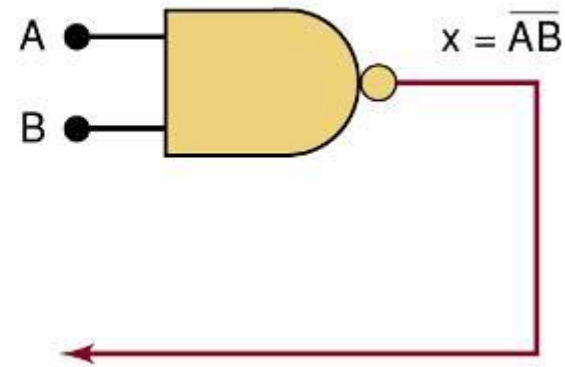
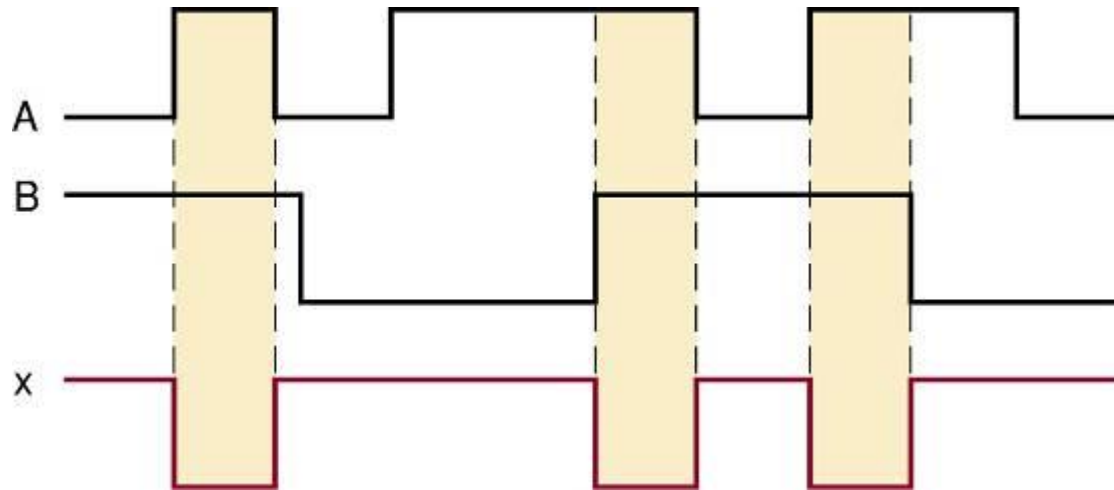
- NAND 게이트는 AND 게이트와 NOT 게이트가 혼합된 형태



		AND	NAND
A	B	AB	\overline{AB}
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

3-9 NOR 게이트와 NAND 게이트

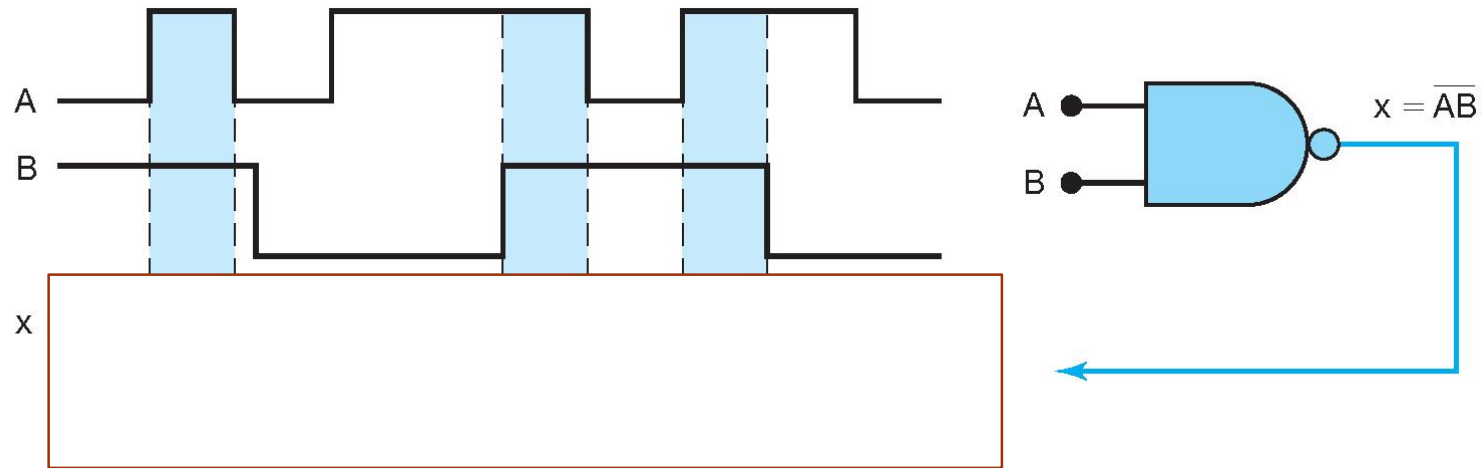
➤ NAND 게이트 입/출력 파형



3-9 NOR 게이트와 NAND 게이트

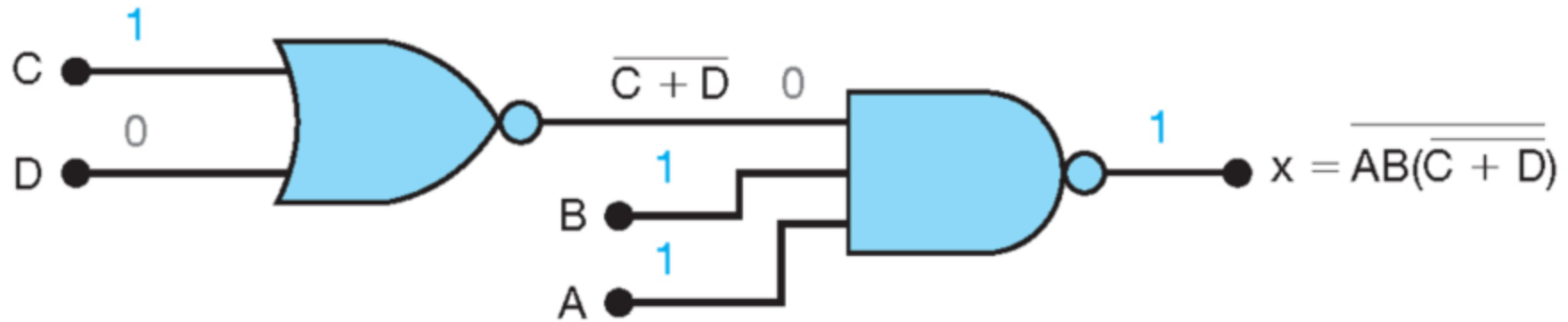
< 예제 3-10 > 그림 3-23의 입력 파형에 대하여 NAND 게이트의 출력 파형을 구하라

그림 3-23 예제 3-10.



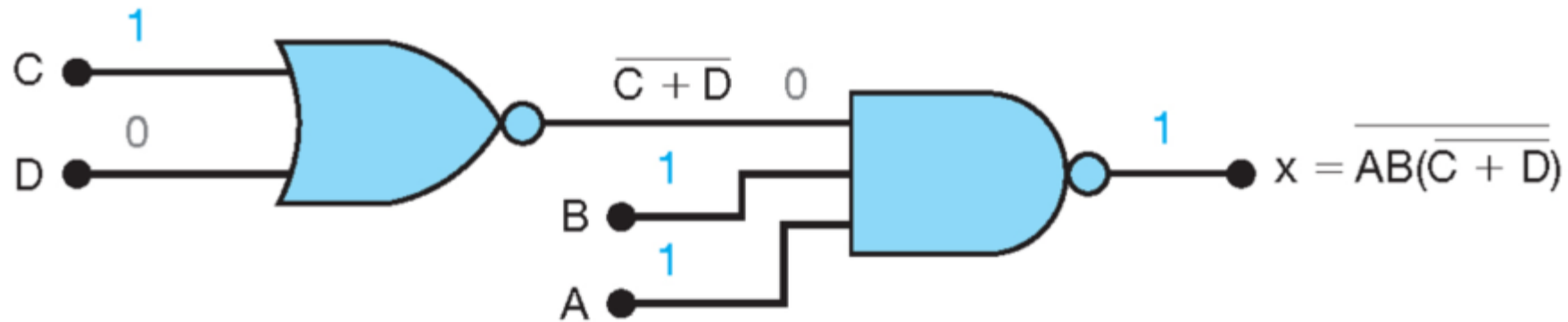
3-9 NOR 게이트와 NAND 게이트

< 예제 3-11 > NOR와 NAND 게이트를 사용하여 표현식 $x = \overline{AB(C+D)}$ 에 대한 논리회로를 구현하라



3-9 NOR 게이트와 NAND 게이트

< 예제 3-12 > 그림 3-24에서 $A+B+C=1$ 이고 $D=0$ 일때 출력 파형을 그려라

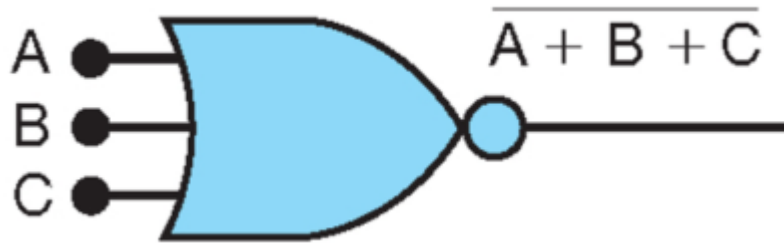


3-9 NOR 게이트와 NAND 게이트

<학습성과 평가문제>

1. 3 입력 NOR 게이트에서 출력이 HIGH가 되는 입력 조건은 무엇인가?

<풀이> 모든 입력이 LOW 일때(입력 $A=B=C=0$ 인 경우)



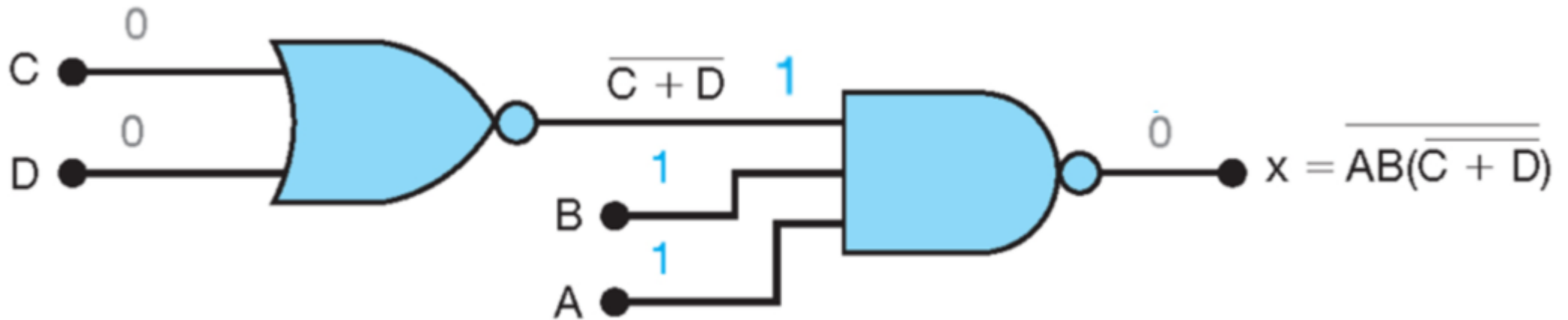
A	B	C	$A+B+C$
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

3-9 NOR 게이트와 NAND 게이트

<학습성과 평가문제>

2. 그림 3-24에서 $A=B=1$ 이고 $C=D=0$ 일때 출력 값을 구하라?

<풀이> $x=0$

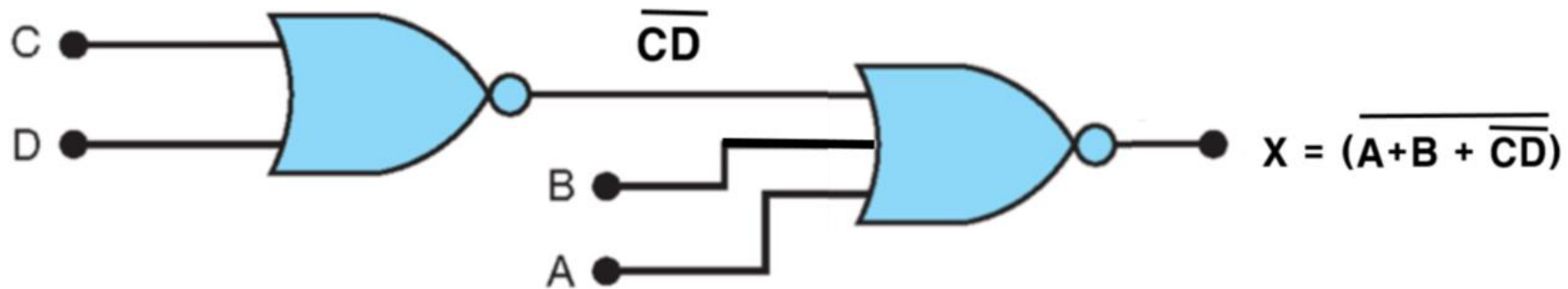


3-9 NOR 게이트와 NAND 게이트

<학습성과 평가문제>

3. 그림 3-24에서 NOR 게이트를 NAND 게이트로 바꾸고 NAND 게이트를 NOR 게이트로 바꾸면 x에 대한 새로운 표현식은?

<풀이> $X = \overline{(A+B + \overline{CD})}$



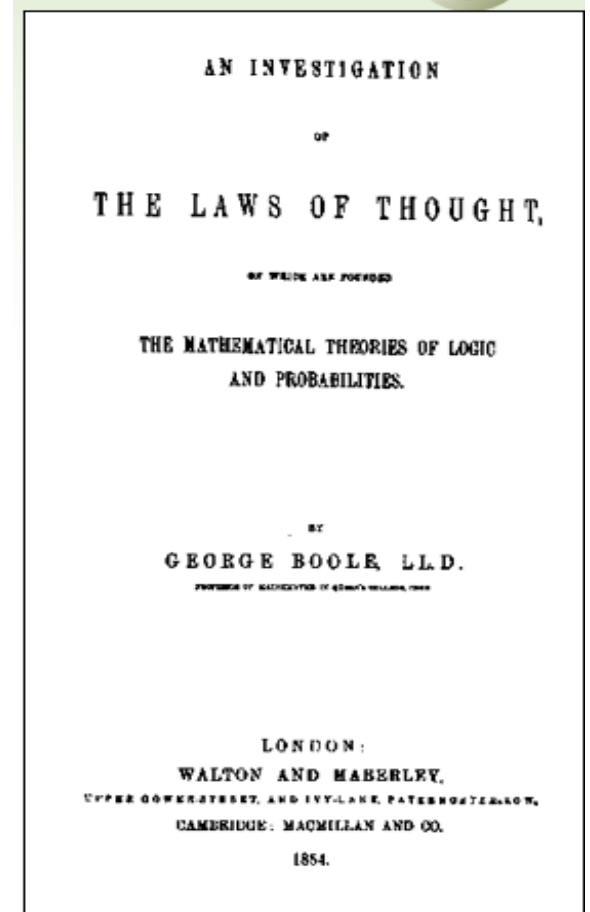
George Boole은 누구인가?



1854 수학자 Gorge Boole이라는 사람은 “사고의 조사” 라는 책에서 참과 거짓이라는 2개의 환경 속에서 논리적인 판단을 만들어 내는 법을 기술하였다

현재, 부울 논리로 알려져 있으며 기호와 연산자를 사용하여 판단을 해 나가는 시스템을 부울 대수라고 부른다.

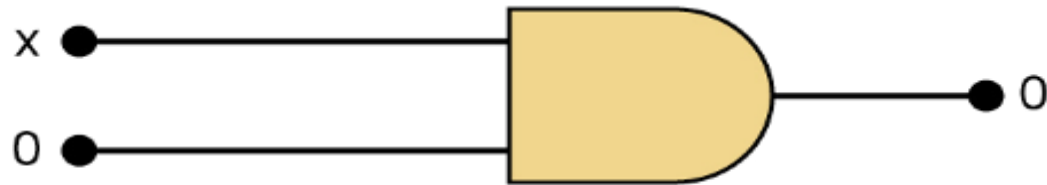
이와 같은 방법으로 정규 대수학에서 미지수를 표현하는 방법으로 x , y 같은 기호를 사요이며, 부울 대수에서는 논리적 표현으로 두개의 값, 참 혹은 거짓으로 표현한다.



3-10 부울 정리(Boolean Theorems)

➤ 단변수 정리(1)

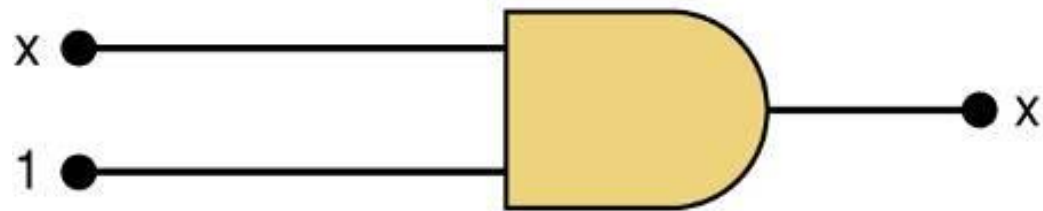
- 부울 정리는 복잡한 논리 표현과 논리회로를 간략히 하는데 사용



(1) $x \cdot 0 = 0$

어떤 변수든 “0” 과 AND 되면 결과는 “0” 이 된다

If $x = 0$, then $0 \cdot 0 = 0$
If $x = 1$, then $1 \cdot 0 = 0$
Thus, $x \cdot 0 = 0$

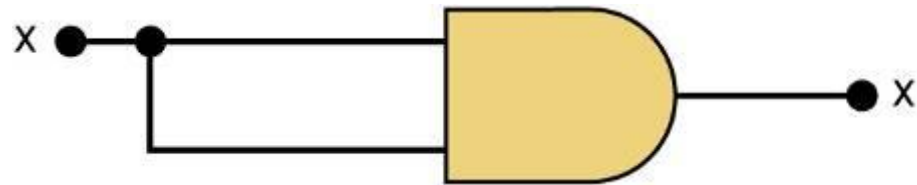


(2) $x \cdot 1 = x$

If $x = 0$, then $0 \cdot 1 = 0$
If $x = 1$, then $1 \cdot 1 = 1$
Thus, $x \cdot 1 = x$

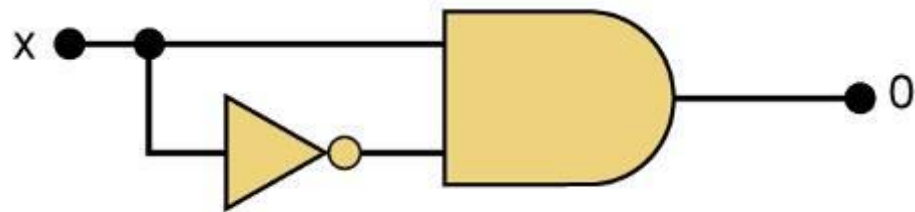
3-10 부울 정리(Boolean Theorems)

➤ 단변수 정리(2)



(3) $x \cdot x = x$

If $x = 0$, then $0 \cdot 0 = 0$
If $x = 1$, then $1 \cdot 1 = 1$
Thus, $x \cdot x = x$

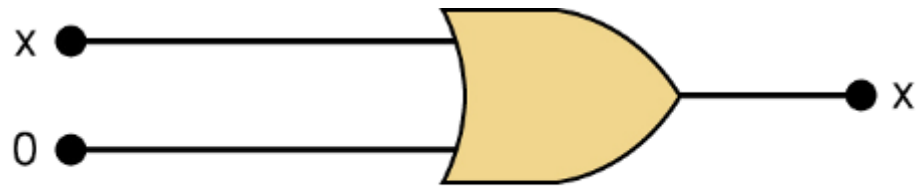


(4) $x \cdot \bar{x} = 0$

If $x = 0$, then $0 \cdot 1 = 0$
If $x = 1$, then $1 \cdot 0 = 0$
Thus, $x \cdot x' = 0$

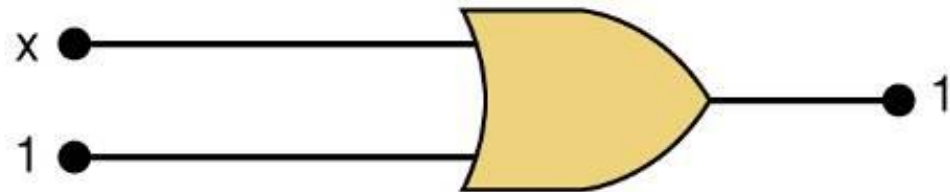
3-10 부울 정리(Booleen Theorems)

➤ 단변수 정리(3)



(5) $x + 0 = x$

If $x = 0$, then $0 + 0 = 0$
If $x = 1$, then $1 + 0 = 1$
Thus, $x + 0 = x$

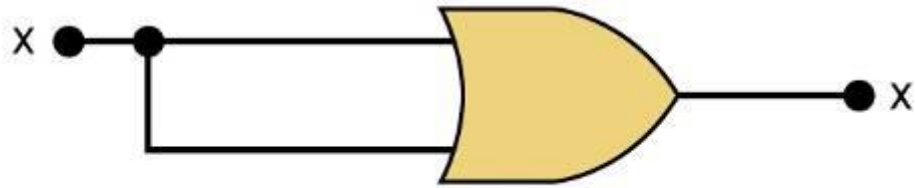


(6) $x + 1 = 1$

If $x = 0$, then $0 + 1 = 1$
If $x = 1$, then $1 + 1 = 1$
Thus, $x + 1 = 1$

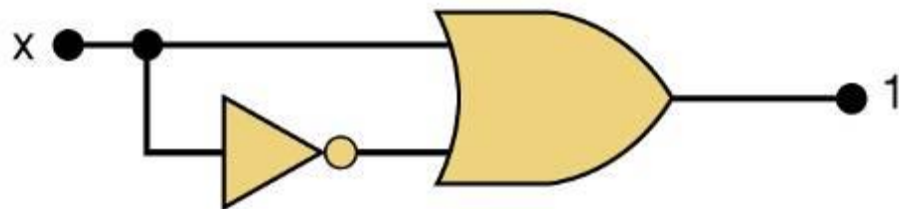
3-10 부울 정리(Booleen Theorems)

➤ 단변수 정리(4)



(7) $x + x = x$

If $x = 0$, then $0 + 0 = 0$
If $x = 1$, then $1 + 0 = 1$
Thus, $x + x = x$



(8) $x + \bar{x} = 1$

If $x = 0$, then $0 + 1 = 1$
If $x = 1$, then $1 + 0 = 1$
Thus, $x + \bar{x} = 1$

3-10 부울 정리(Boolean Theorems)

➤ 다변수 정리(1)

▪ 교환 법칙(Commutative laws)

$$- x + y = y + x \quad (9)$$

$$- x y = yx \quad (10)$$

▪ 결합 법칙(Associative laws)

$$- x + (y + z) = (x + y) + z = x + y + z \quad (11)$$

$$- x(yz) = (xy)z = xyz \quad (12)$$

▪ 분배 법칙(Distributive law)

$$- x(y + z) = xy + xz \quad (13a)$$

$$- (w + x)(y + z) = wy + xy + wz + xz \quad (13b)$$

일반 대수와 마찬가지로
기억하면 사용하기 쉽다

3-10 부울 정리(Boolean Theorems)

➤ 다변수 정리(2)

- 일반 대수에 상응하는 규칙이 없다

$$-x + xy = x \quad (14)$$

$$-x + x' y = x + y \quad (15a)$$

$$-x' + xy = x' + y \quad (15b)$$

$$\begin{aligned} x + xy &= x(1 + y) \\ &= x \cdot 1 && \text{[using theorem (6)]} \\ &= x && \text{[using theorem (2)]} \end{aligned}$$

x	y	xy	x + xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

3-10 부울 정리(Booleen Theorems)

< 예제 3-13 > 다음 표현식을 부울 정리를 이용하여 간략히 나타내어라

$$y = \overline{A}BD + A\overline{B}D$$

< 풀이 > 정리 (13-a)을 이용하여 공통 변수 $\overline{A}B$ 로 묶고, 정리(8)을 이용하면

$$\begin{aligned} y &= \overline{A}BD + A\overline{B}D \\ &= \overline{A}B (D + \overline{D}) \\ &= \overline{A}B \cdot 1 \\ &= \overline{A}B \end{aligned}$$

3-10 부울 정리(Boolean Theorems)

< 예제 3-14 > 다음 표현식을 부울 정리를 이용하여 간략히 나타내어라

$$Z = (\bar{A} + B)(A + B)$$

<풀이> 정리 (13-b)을 이용하여 각 항을 곱하여 전개

정리 (4)에 따라 $A \cdot \bar{A} = 0$ 대입

정리 (3)에 따라 $B \cdot B = B$ 대입

정리 (13)을 이용하여 공통 인수 B로 묶는다

정리 (2)와 (5) 대입

$$\begin{aligned} Z &= (\bar{A} + B)(A + B) \\ &= (\bar{A}A + \bar{A}B + BA + BB) \\ &= 0 + \bar{A}B + BA + B \\ &= B(\bar{A} + A) \\ &= B \cdot 1 \\ &= B \end{aligned}$$

3-10 부울 정리(Booleen Theorems)

〈예제 3-15〉 다음 표현식을 부울 정리를 이용하여 간략히 나타내어라

$$x = ACD + \bar{A}BCD$$

〈풀이〉 공통 인수 CD로 묶는다

정리 (15-a)를 이용하여 $A + \bar{A}B$ 를 $A+B$ 로 대체

정리 (13- π)를 이용 각항을 전개

$$\begin{aligned}x &= ACD + \bar{A}BCD \\ &= CD(A + \bar{A}B) \\ &= CD(A+B) \\ &= ACD + BCD\end{aligned}$$

3-10 부울 정리(Boolean Theorems)

<학습성과 평가문제>

1. 정리 (13)과 (14)를 이용하여 아래 식을 간략히 나타내어라

$$Y = A\bar{C} + ABC\bar{C}$$

<풀이> 정리 (13)을 이용하여 공통 인수 C 로 묶고
정리 (14)를 이용하여 A+AB를 A로 대체

$$\begin{aligned} Y &= A\bar{C} + ABC\bar{C} \\ &= \bar{C}(A + AB) \\ &= A\bar{C} \end{aligned}$$

3-10 부울 정리(Boolean Theorems)

<학습성과 평가문제>

3. 정리 (13)과 (8)를 이용하여 아래식을 간략히 나타내어라

$$Y = \overline{A}BCD + \overline{A}BC\overline{D}$$

<풀이> 정리 (13)을 이용하여 공통 인수 $\overline{A}BD$ 로 묶는다

정리 (8)를 이용하여 $C + \overline{C} = 1$ 로 대체

$$\begin{aligned} Y &= \overline{A}BCD + \overline{A}BC\overline{D} \\ &= \overline{A}BD(C + \overline{C}) \\ &= \overline{A}BD \end{aligned}$$

3-10 부울 정리(Boolean Theorems)

<학습성과 평가문제>

3. 정리 (13)과 (15-b)를 이용하여 아래식을 간략히 나타내어라

$$Y = \bar{A}D + ABD$$

<풀이> 정리 (13)을 이용하여 공통 인수 D 로 묶는다

정리 (15-b)를 이용하여 $\bar{A}+AB$ 를 $\bar{A}+B$ 로 대체

$$\begin{aligned} Y &= \bar{A}D + ABD \\ &= D(\bar{A}+AB) \\ &= D(\bar{A}+B) \\ &= \bar{A}D + BD \end{aligned}$$

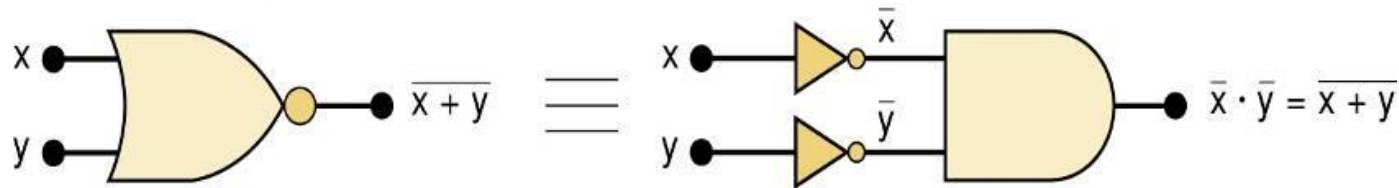
3-11 드모르간 정리(DeMorgan's Theorems)

➤ 드모르간 정리(1)

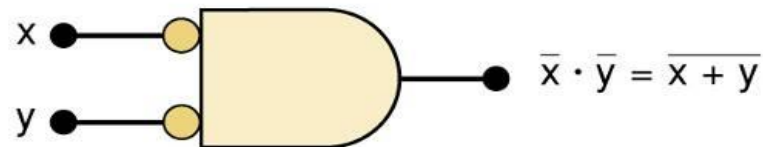
- 변수의 합 또는 곱의 역을 간략히 표현할 때 매우 유용

Equivalent circuits implied by Theorem (16)

$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$



The alternative symbol for the NOR function.

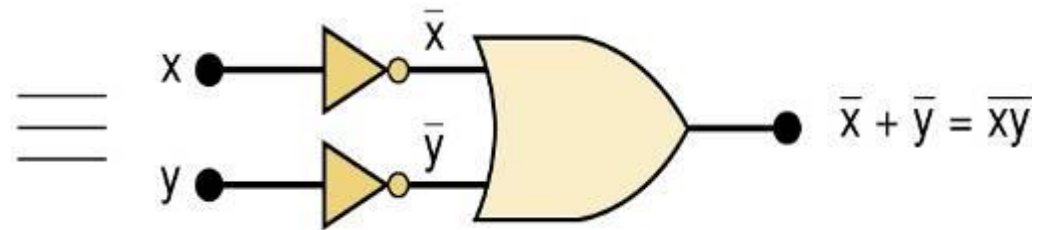
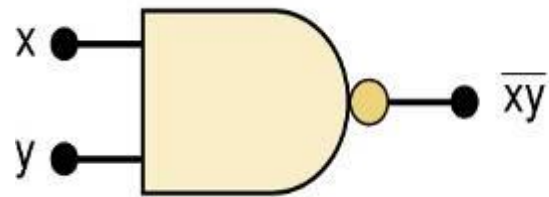


3-11 드모르간 정리(DeMorgan's Theorems)

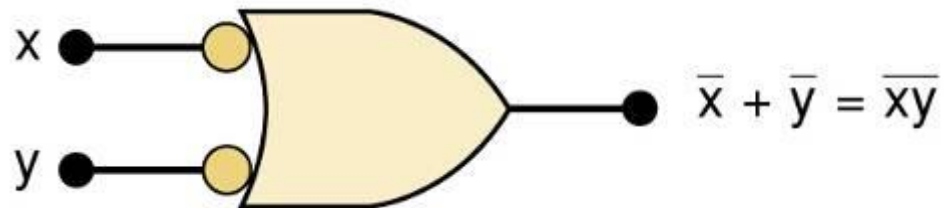
➤ 드모르간 정리(2)

Equivalent circuits implied by Theorem (17)

$$(17) \quad \overline{(x \cdot y)} = \bar{x} + \bar{y}$$

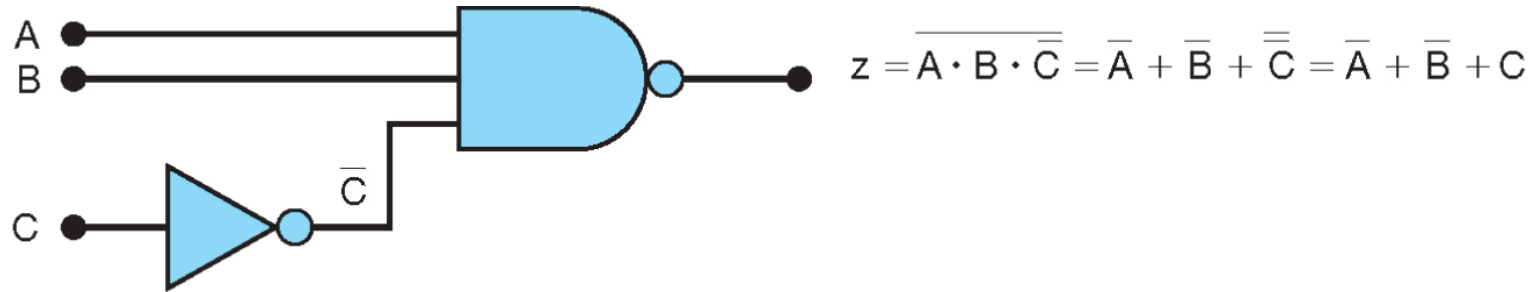


The alternative symbol for the NAND function.



3-11 드모르간 정리(DeMorgan's Theorems)

< 예제 3-17 > 그림 3-28에 있는 DeMorg 출력 표현식을 구하고 드모르간 정리를 사용하여 표현식을 간략히 나타내어라



< 풀이 > 드모르간 정리 (17)을 이용 하면

$$z = \overline{A \cdot B \cdot \bar{C}} = \bar{A} + \bar{B} + \bar{\bar{C}} = \bar{A} + \bar{B} + C$$

3-11 드모르간 정리(DeMorgan's Theorems)

<학습성과 평가문제>

1. 드모르간 정리를 사용하여 $Z = \overline{A + B + C}$ 를 단일 변수의 역으로만 표현되는 식으로 변환하라

<풀이>
$$\begin{aligned} Z = \overline{A + B + C} &= (\overline{AB}) \cdot \overline{C} \\ &= \overline{AB} + C \end{aligned}$$

2. 표현식 $y = \overline{RST+Q}$ 를 문제 1번과 같이 변환하라

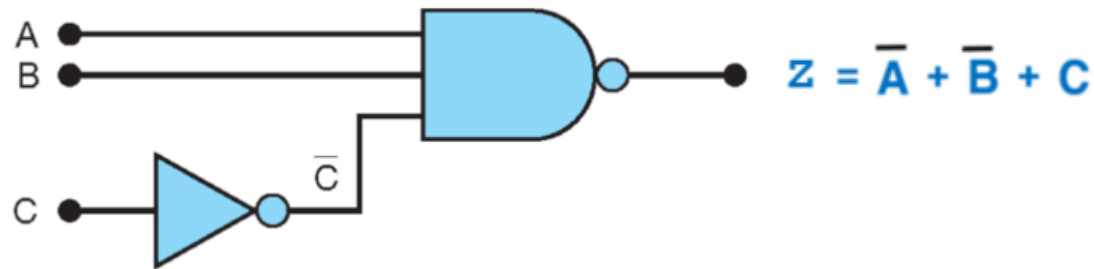
<풀이>
$$\begin{aligned} Y &= \overline{RST + Q} \\ &= \overline{RST} + \overline{Q} \\ &= (\overline{R} + \overline{S} + \overline{T}) + Q \\ &= (\overline{R} + \overline{S} + \overline{T})Q \end{aligned}$$

3-11 드모르간 정리(DeMorgan's Theorems)

<학습성과 평가문제>

3. 출력 표현식이 $z = \bar{A} \bar{B} C$ 인 회로를 NOR 게이트와 인버터만 사용하여 구현 하라

<풀이> $z = \bar{A} \bar{B} C = (\bar{A} \bar{B}) + C = \bar{A} + \bar{B} + C$



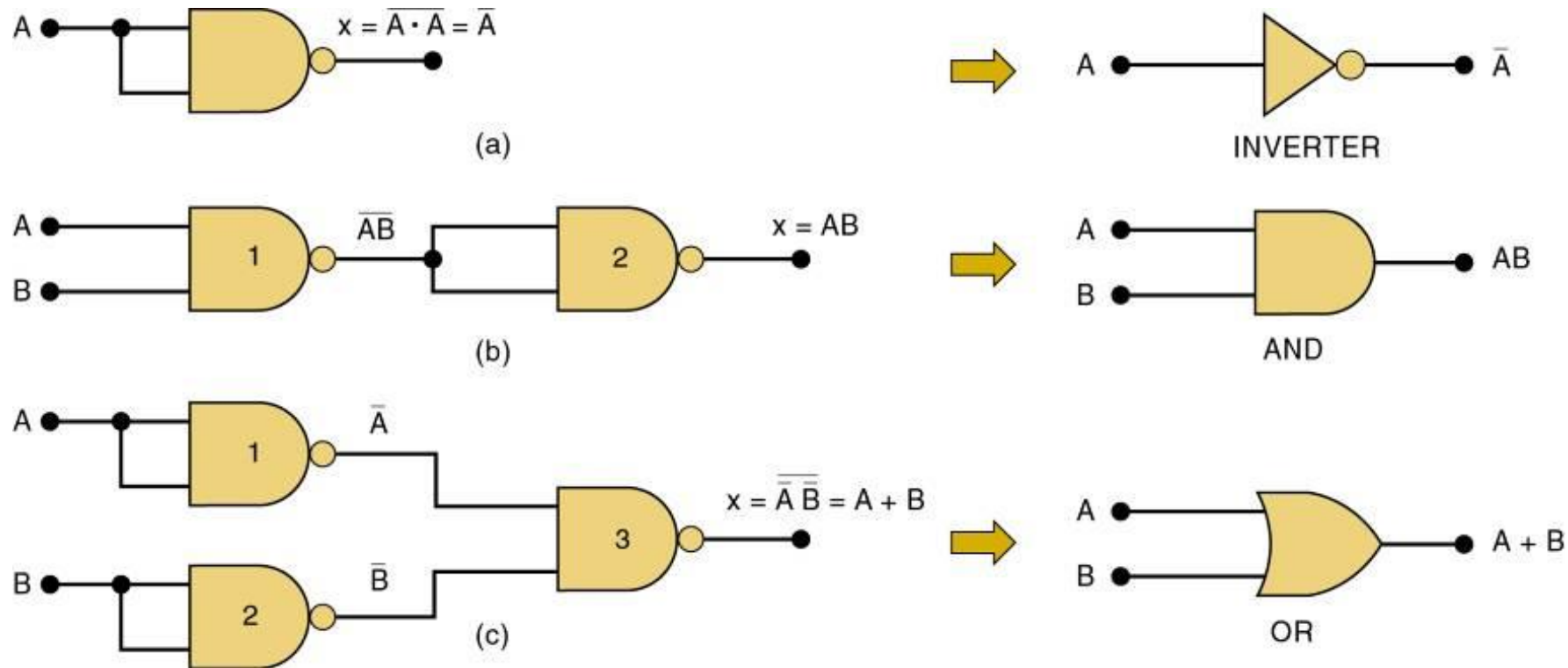
4. 드모르간 정리를 사용하여 $y = \overline{A + \bar{B} + \bar{C}D}$ 를 단일 변수의 역으로만 표현 되는 식으로 변환 하라

<풀이> $y = \overline{A + \bar{B} + \bar{C}D} = (\bar{A} + \bar{\bar{B}}) + \bar{\bar{C}D}$
 $= \bar{A} B + (C + \bar{D})$

3-12 NAND 게이트와 NOR 게이트의 범용성

➤ AND, OR, NOT 게이트 기능을 NAND 게이트로 구현

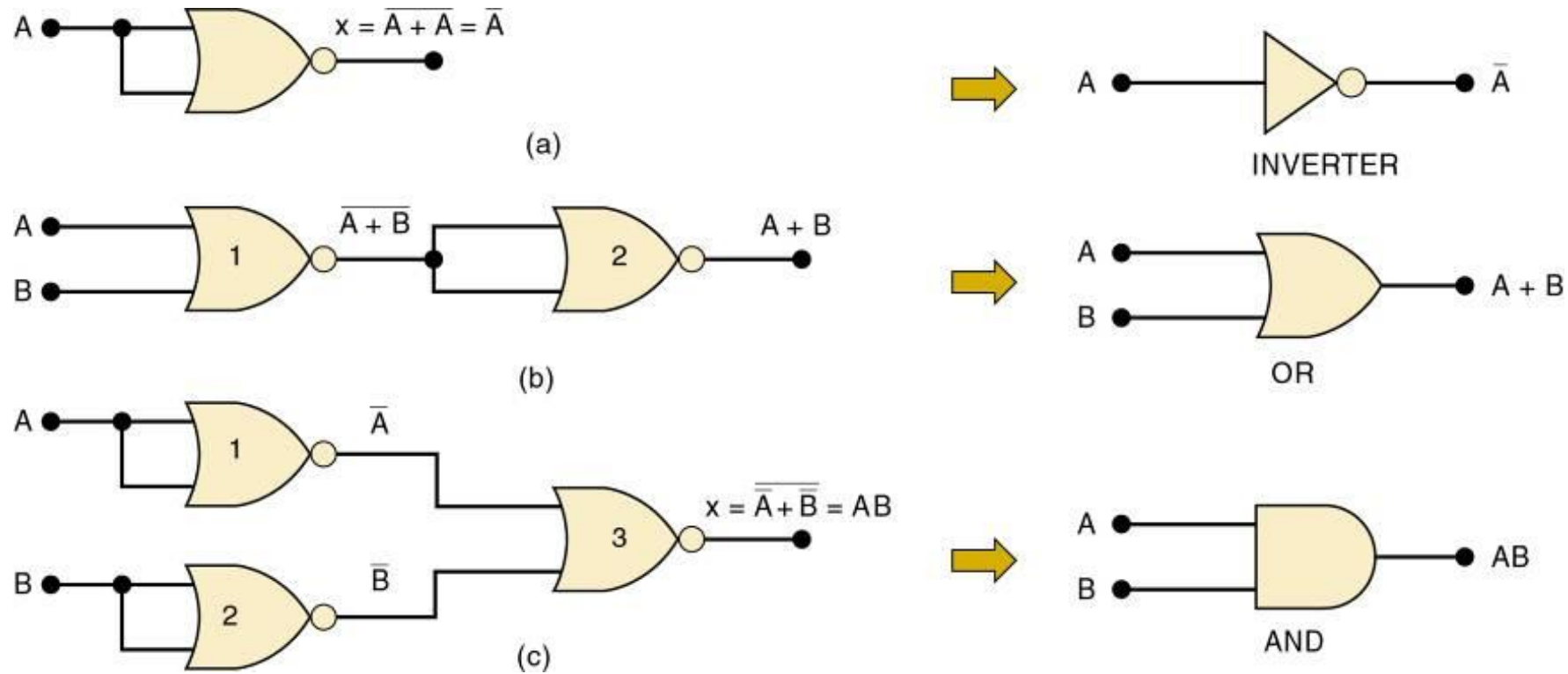
- NAND 게이트를 연결하면 어떤 부울 연산도 수행할 수 있다



3-12 NAND 게이트와 NOR 게이트의 범용성

➤ AND, OR, NOT 게이트 기능을 NOR 게이트로 구현

- NOR 게이트를 연결하면 어떤 부울 연산도 수행할 수 있다



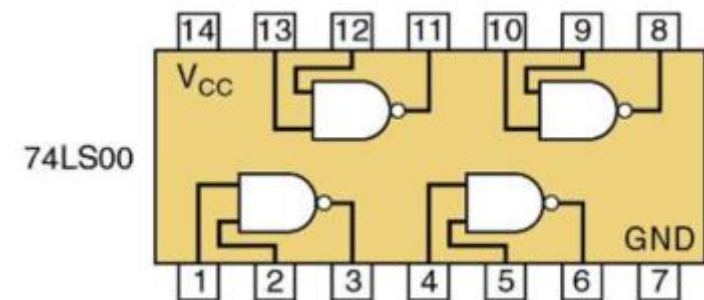
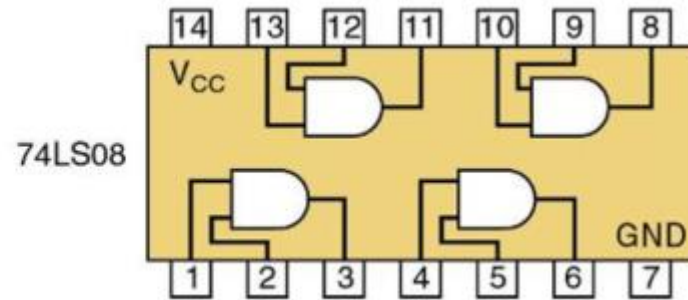
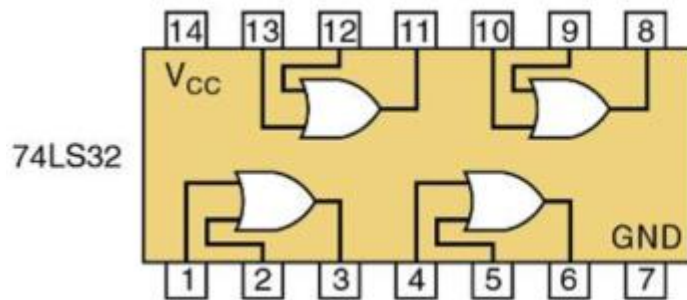
3-12 NAND 게이트와 NOR 게이트의 범용성

<예제 3-18>

A : 컨베어 벨트 속도가 지나치게 빠르면 HIGH, B: 벨트 끝에 있는 부품 수거통이 꽉 차면 HIGH

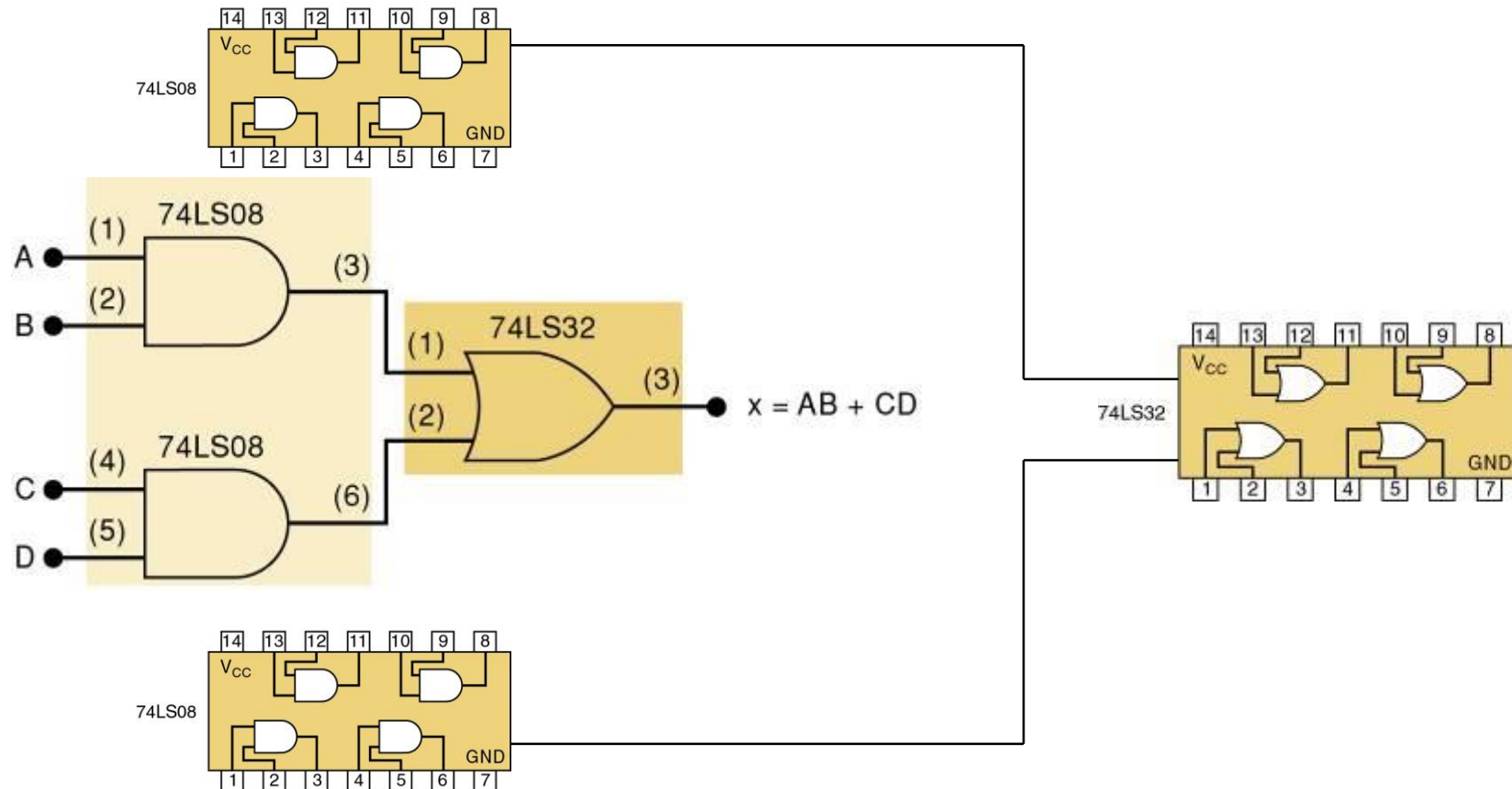
C: 벨트 장력이 너무 높으면 HIGH, D: 수동 정지 신호가 켜지면 HIGH

A와 B의 조건이 동시에 HIGH 가 되거나 C와 D 조건이 동시에 HIGH 가 되면
신호 x 가 HIGH 되는 논리회로 필요(논리 표현식 $x = (AB + CD)$)



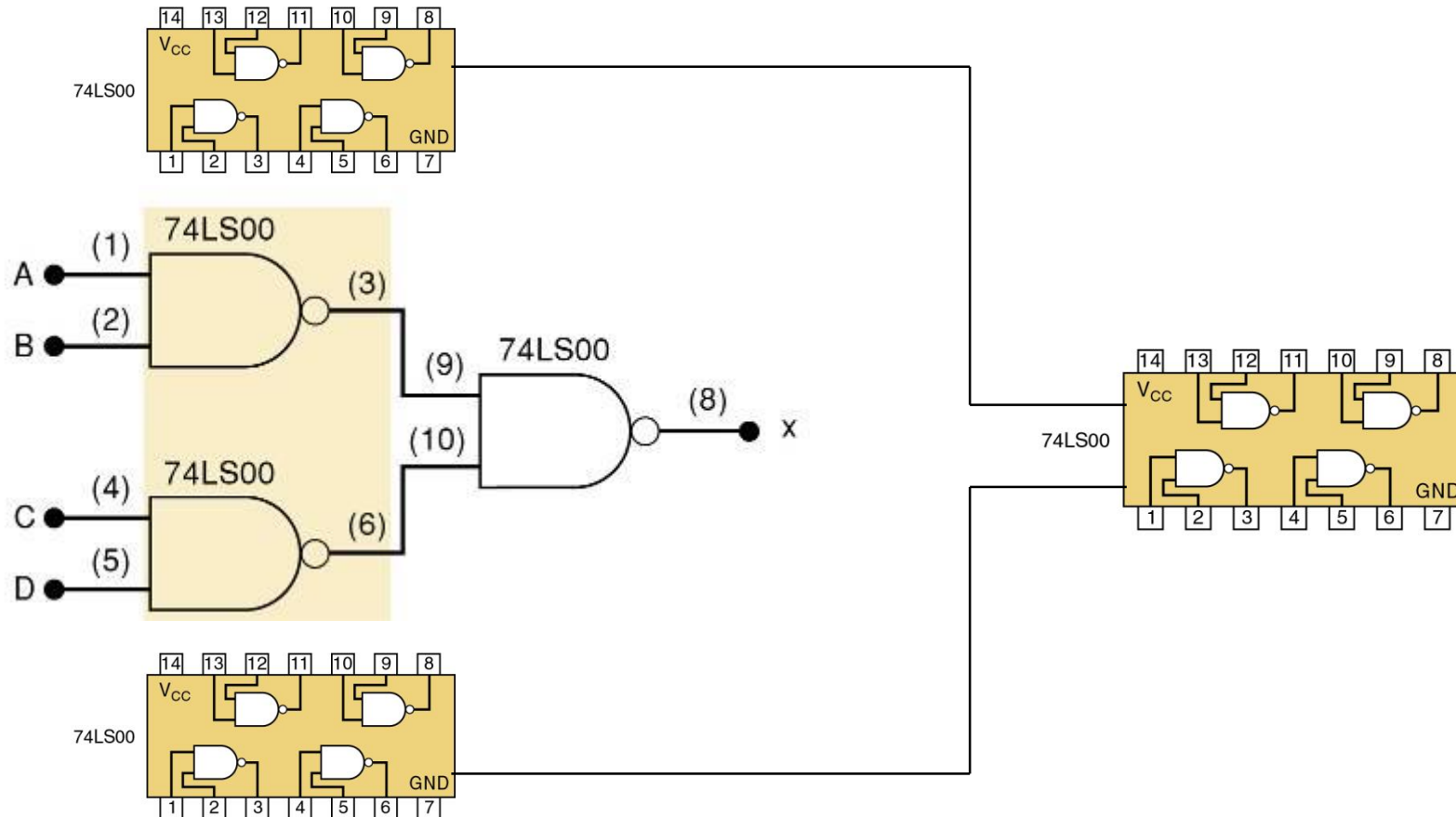
3-12 NAND 게이트와 NOR 게이트의 범용성

<예제 3-18> 구현 방법 # 1



3-12 NAND 게이트와 NOR 게이트의 범용성

<예제 3-18> 구현 방법 # 2



3-12 NAND 게이트와 NOR 게이트의 범용성

<학습성과 평가문제>

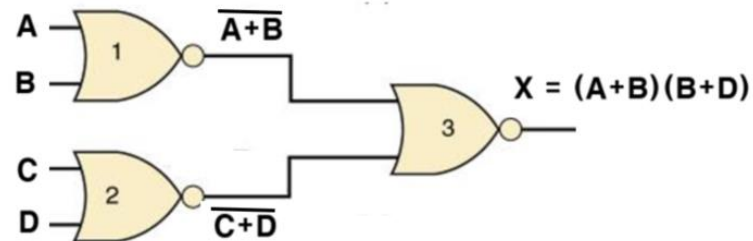
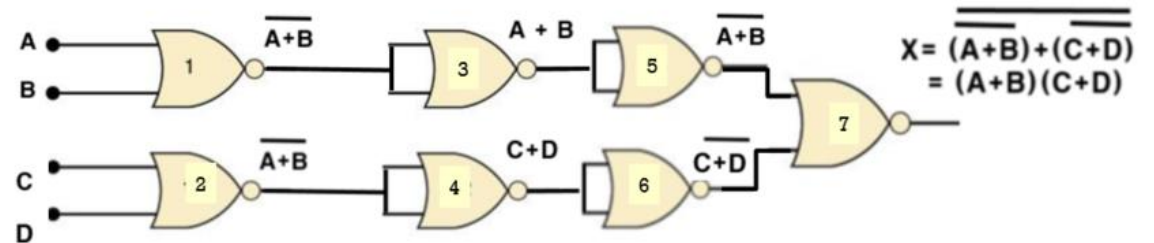
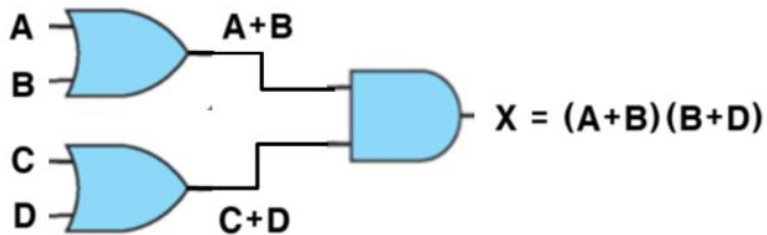
1. 하나의 논리회로에서 역 연산을 구현하는 방법이 몇 가지나 되는가?

<풀이> 3가지

2. 표현식 $x=(A+B)(C+D)$ 를 OR와 AND 게이트로 구현하라. 그리고 그림 3-30에 보인 NOR로 구성하는 방법을 사용하여 각 OR 게이트와 AND 게이트를 NOR 게이트로 변환하여 구현하라. 어떤 회로가 더 효율적인가?

<풀이> $x=(A+B)(C+D)$

NOR 게이트를 3개만으로 가능

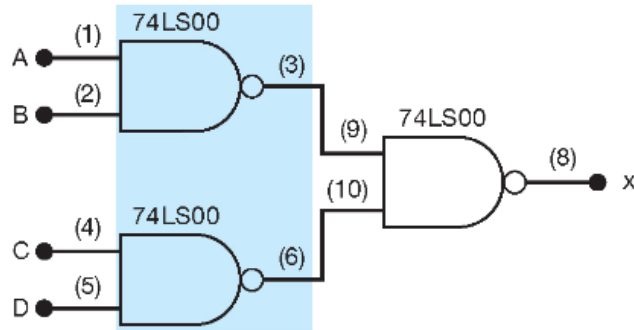


3-12 NAND 게이트와 NOR 게이트의 범용성

<학습성과 평가문제>

3. 그림 3-32의 (C)에 있는 회로의 출력 표현식을 쓰고, 드모르간 정리를 사용하여 그림 3-32(a)에 있는 회로에 대한 표현식과 등가임을 보여라

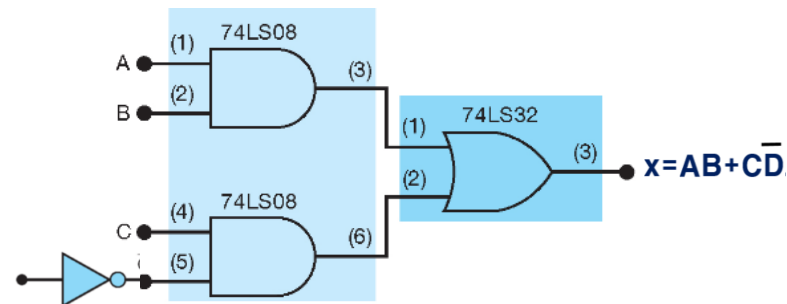
<풀이>



$$X = \overline{\overline{AB} \overline{CD}} = \overline{\overline{AB} + \overline{CD}} = AB + CD$$

4. 그림 3-32(a)에서 만약 출력 $x = AB + \overline{CD}$ 로 변경 시키면 D의 반전된 입력이 필요로 한다. 그럴 경우에 몇 개의 IC 칩이 필요 한가?

<풀이> 3개

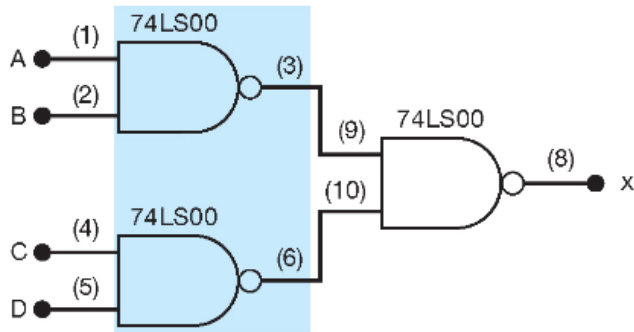


3-12 NAND 게이트와 NOR 게이트의 범용성

<학습성과 평가문제>

5. 위의 문제와 같이 출력이 변경 되었을 때 그림 3-32(c)의 NAND gate를 사용할 경우 얼마나 많은 IC 칩이 필요한가?

<풀이> 1개

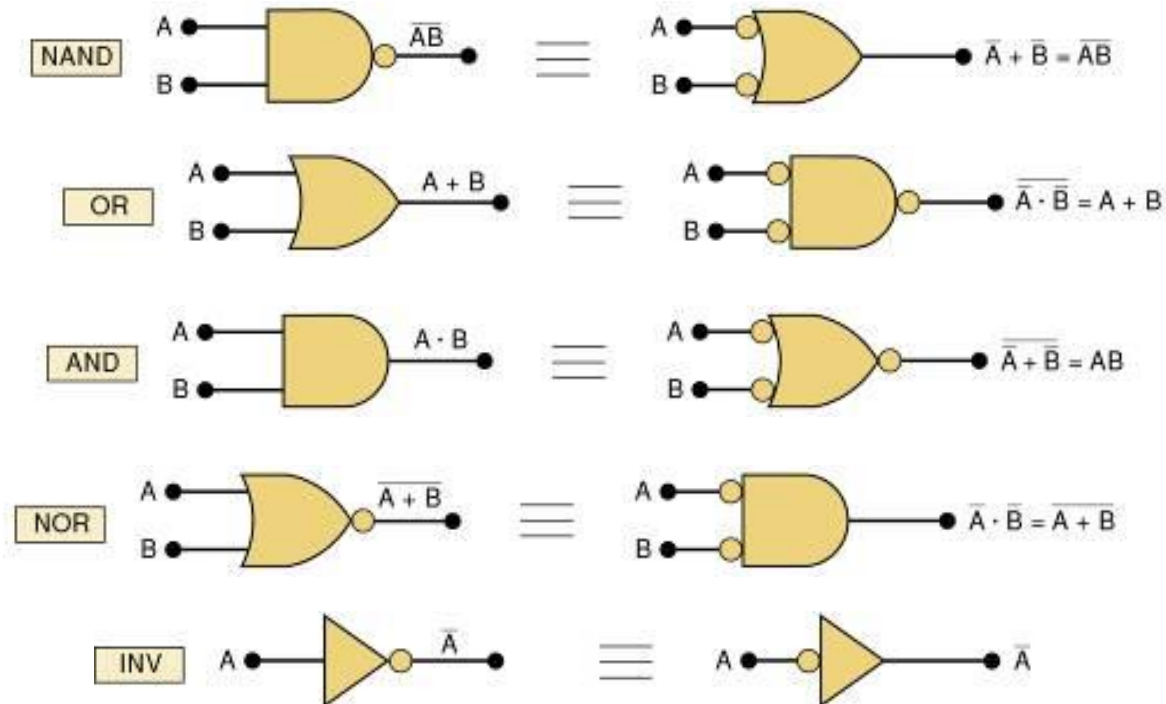


$$X = \overline{\overline{AB}} \overline{\overline{CD}} = \overline{AB} + \overline{CD} \\ = AB + CD$$

3-13 논리 게이트의 대안 표시

➤ 표준 논리 기호를 대안 논리 기호로 변환

- 표준 기호의 모든 입출력을 역변환 시킨다
(입출력 단자에 방울(작은 원)이 없다면 방울을 더하고 있다면 제거하면 된다)
- 연산 기호를 AND에서 OR로, 또는 OR에서 AND로 바꾼다
(인버터의 경우에는 연산 기호를 바꾸지 않는다)



3-13 논리 게이트의 대안 표시

논리 기호의 해석

논리회로 기호의 입력 또는 출력 단자에

- 방울이 붙어 있지 않을 때 그 단자를 HIGH 활성화(Active HIGH)
- 방울이 붙어 있을 때 그 단자를 LOW 활성화(Active LOW)

그림 3-34 NAND 게이트의 두 가지 기호 해석.

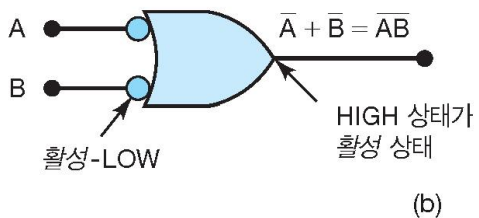
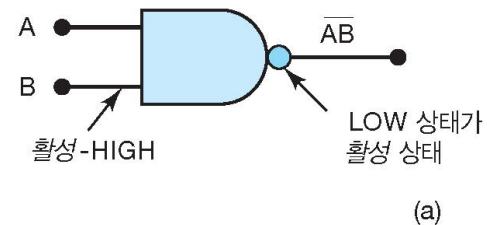
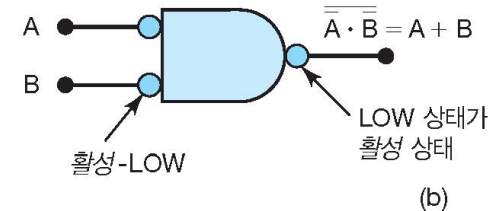
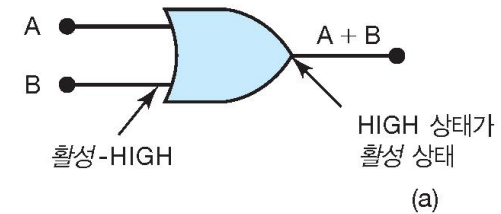


그림 3-35 OR 게이트의 두 가지 기호 해석.



입력 중 어느 하나라도 HIGH 일 때 출력은 HIGH가 된다.

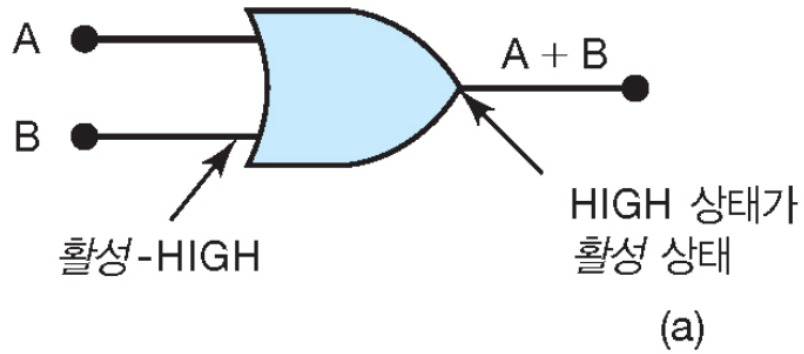
모든 입력이 HIGH일 때 출력은 LOW가 된다.

입력 중 어느 하나라도 LOW 일 때 출력은 HIGH가 된다.

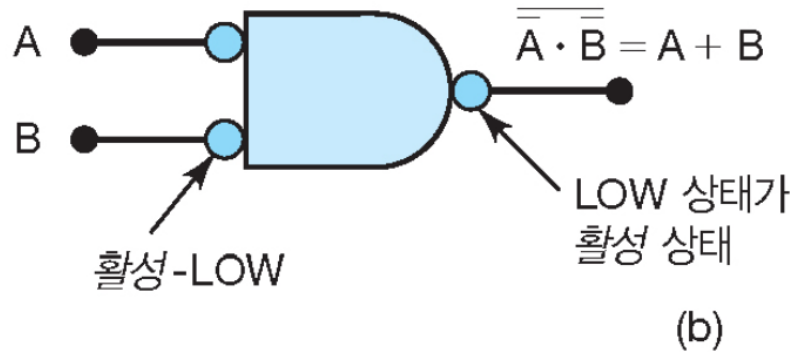
모든 입력이 LOW일 때 출력은 LOW가 된다.

3-13 논리 게이트의 대안 표시

〈예제 3-19〉 두 OR 게이트 기호를 해석하는 방법을 보여라



입력 중 어느 하나라도 HIGH 일 때 출력은 HIGH가 된다.



모든 입력이 LOW일 때 출력은 LOW가 된다.

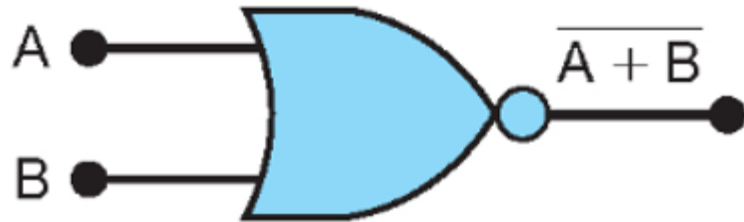


3-13 논리 게이트의 대안 표시

<학습성과 평가문제>

1. 그림 3-33에서 표준 NOR 게이트 기호의 동작을 해석하는 방법을 적어라

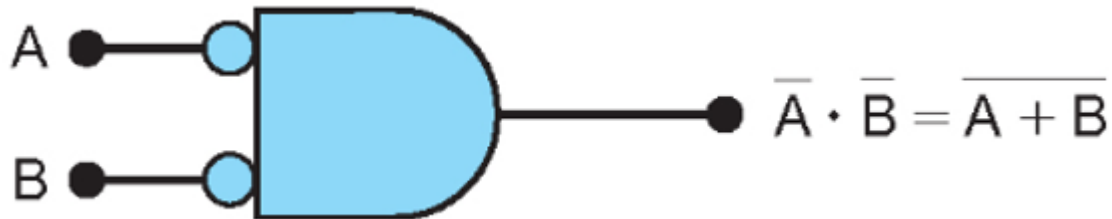
<풀이>



입력 중 어느 하나라도 HIGH일 때
출력은 LOW가 된다

2. NOR 게이트 대안 표시에 대해 1번 문재와 같이 적어라

<풀이>



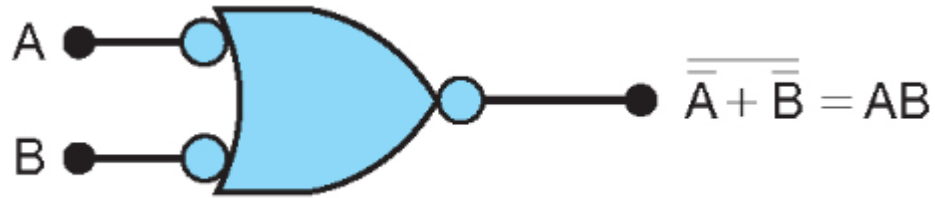
모든 입력이 LOW일 때
출력은 HIGH가 된다

3-13 논리 게이트의 대안 표시

<학습성과 평가문제>

3. AND 게이트 대안 표시에 대해 1번 문재와 같이 적어라

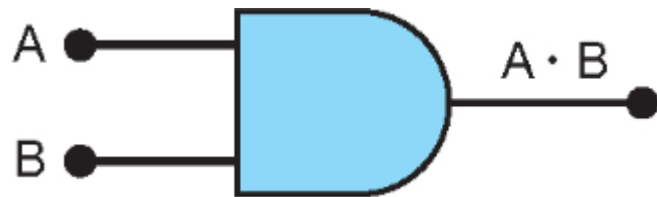
<풀이>



입력 중 어느 하나라도 LOW 일때
출력은 LOW가 된다

4. AND 게이트 표준 표시에 대해 1번 문재와 같이 적어라

<풀이>

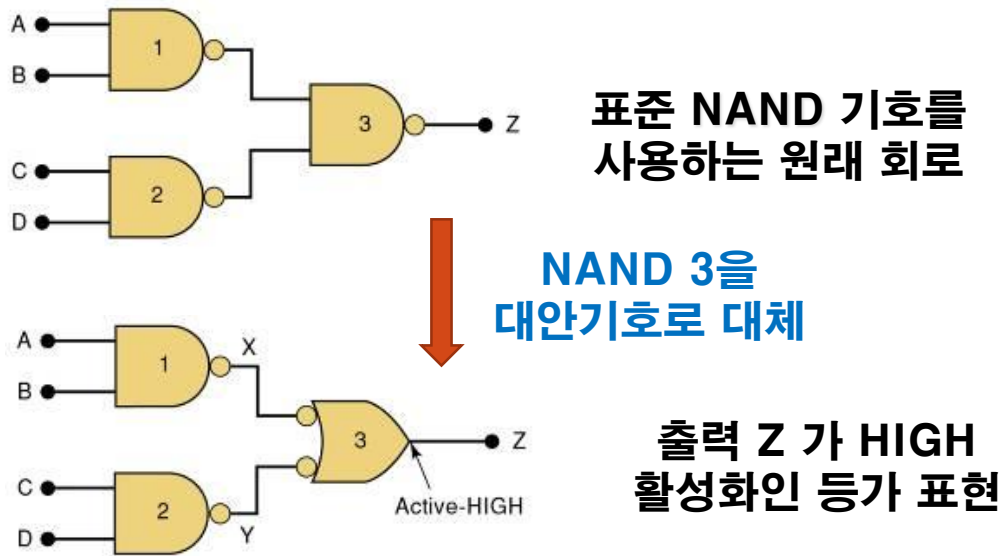


모든 입력이 HIGH 일 때
출력은 HIGH가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

➤ 대안 게이트 기호(1)

- 대안 게이트 기호를 적절히 사용하면 회로의 동작을 보다 분명하게 나타낼 수 있다



A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

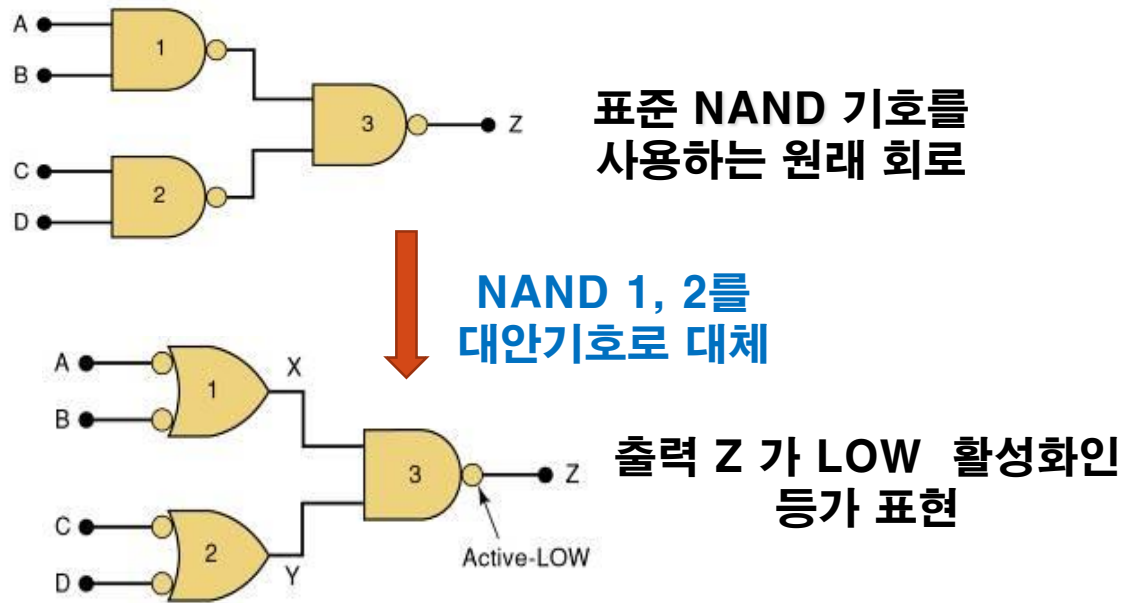
출력 z는 HIGH 활성 출력을 가진 NAND 게이트 신호에서 얻어 진다

➔ 출력 z는 A=B=1 이거나 C=D=1 또는 두 경우 모두 1일 때 HIGH 가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

➤ 대안 게이트 기호(2)

- 대안 게이트 기호를 적절히 사용하면 회로의 동작을 보다 분명하게 나타낼 수 있다



출력 z는 LOW 활성화 출력을 가진 NAND 게이트 신호에서 얻어 진다
 → 출력 z는 A 또는 B가 LOW 이고 C 또는 D가 LOW일 때만 LOW가 된다

A	B	C	D	Z
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

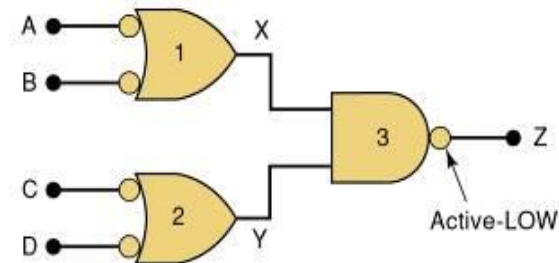
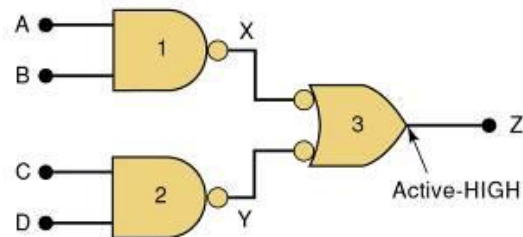
3-14 어떤 게이트 표현을 사용할 것인가?

➤ 어떤 회로를 사용할 것인가?

- 회로 출력이 수행하는 기능에 달려 있다
 - 출력 z가 1 상태일 때 어떤 동작을 수행 하도록 되어 있다면 출력 Z가 HIGH 활성화인 등가 회로 사용
 - 출력 z가 0 상태일 때 어떤 동작을 수행 하도록 되어 있다면 출력 Z가 LOW 활성화인 등가 회로 사용

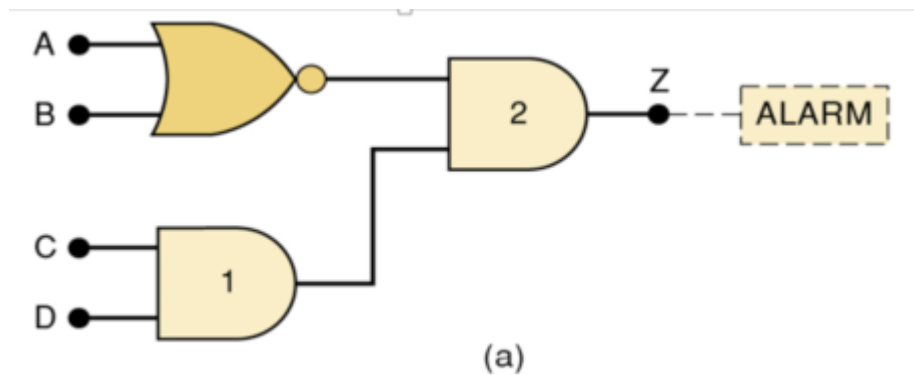
➤ 방울의 위치

- 가능하다면 방울이 달린 출력은 항상 방울이 달린 입력에 연결
- 방울이 없는 출력은 방울이 없는 입력에 연결



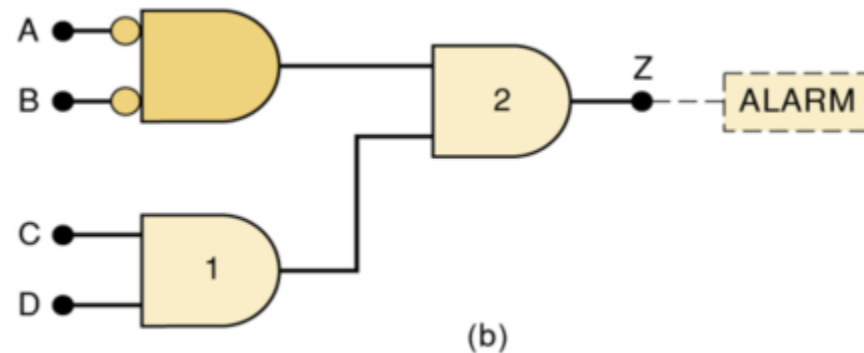
3-14 어떤 게이트 표현을 사용할 것인가?

<예제 3-20> 아래 논리회로를 출력 z가 HIGH 일 때 Alarm이 동작하도록 설계



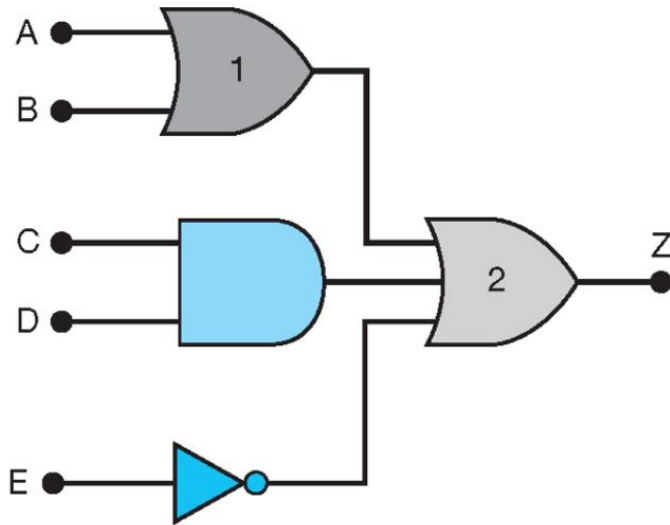
회로도를 수정하면 논리 회로를 더 효과적으로 표현할 수 있다

NOR게이트 기호를 AND Gate 2의 방울 없는 입력과 매치되도록 방울 없는 출력(HIGH 활성화)의 대안 기호로 변환



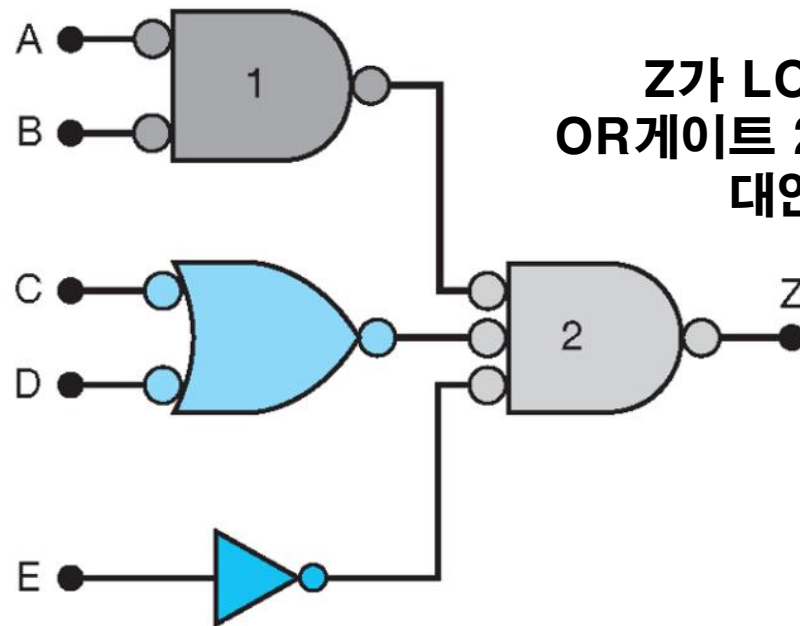
3-14 어떤 게이트 표현을 사용할 것인가?

<예제 3-21> 아래 논리회로를 출력이 LOW 가 되면 다른 논리회로를 활성화 시킨다. 이 회로도를 좀 더 효율적으로 동작하게 그려라



AND 게이트와 OR 게이트 출력을 방울 있는 대안기호로 변환하여 모든 방울 출력이 게이트 2의 입력이 되게 한다

회로도를 수정하면 논리 회로를 더 효과적으로 표현할 수 있다



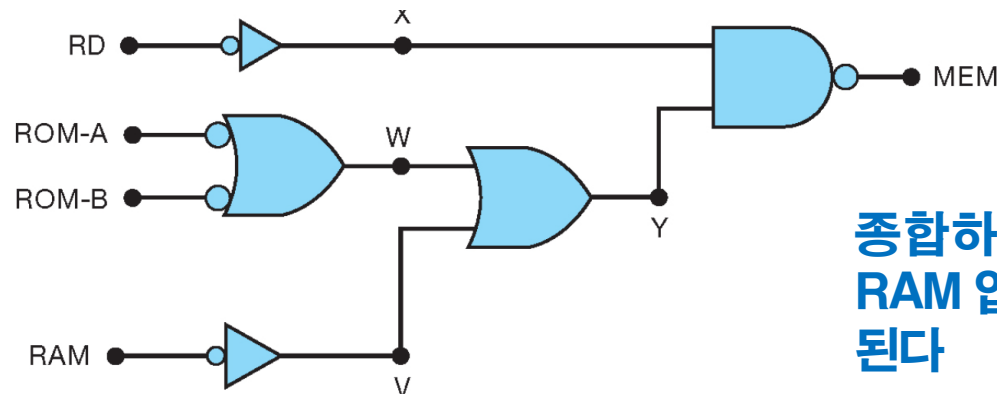
Z가 LOW 활성화이므로 OR게이트 2의 출력을 방울 달린 대안기호로 변환

3-14 어떤 게이트 표현을 사용할 것인가?

➤ 회로의 분석

〈예제 3-22〉 아래 회로도에서 MEM을 활성화 시키는 입력 조건을 구하라

- MEM이 LOW 활성화이므로 X와 Y가 모두 HIGH 일때만 LOW가 된다
- X는 RD=0일때만 HIGH가 된다
- Y는 W 또는 V가 HIGH 일 때 HIGH가 된다
- V는 RAM=0일 때 HIGH가 된다
- W는 ROM-A 또는 ROM-B=0일때 HIGH가 된다

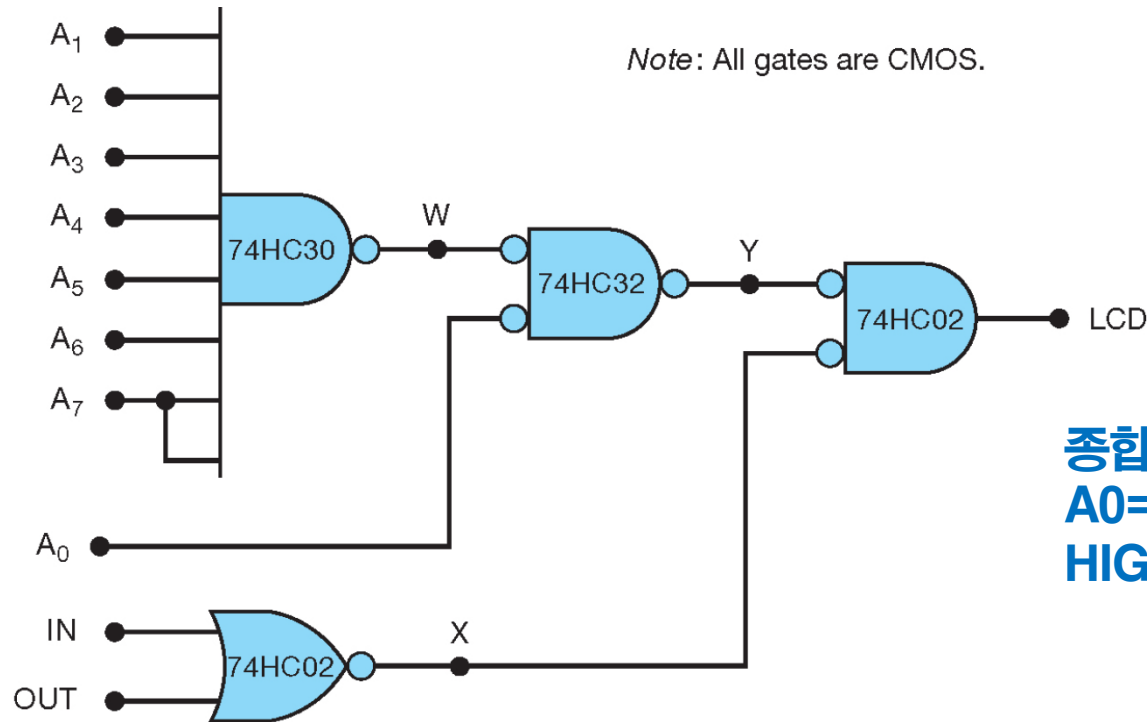


종합하면 MEM은 RD=0이고 ROM-A, ROM-B, RAM 입력중 최소한 개의 입력이 LOW일 때 LOW가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

회로의 분석

〈예제 3-23〉 회로는 LCD=1일 때 동작 한다. LCD 표시기가 동작하기 위한 입력 조건을 구하라



- LCD는 HIGH 활성화이고 X=Y=0일 때 만 HIGH
- X는 IN 또는 OUT 이 HIGH일 때 LOW
- W=0이고 A₀=0 일 때 Y가 LOW
- A1부터 A7 까지 모두 HIGH일때 W는 LOW

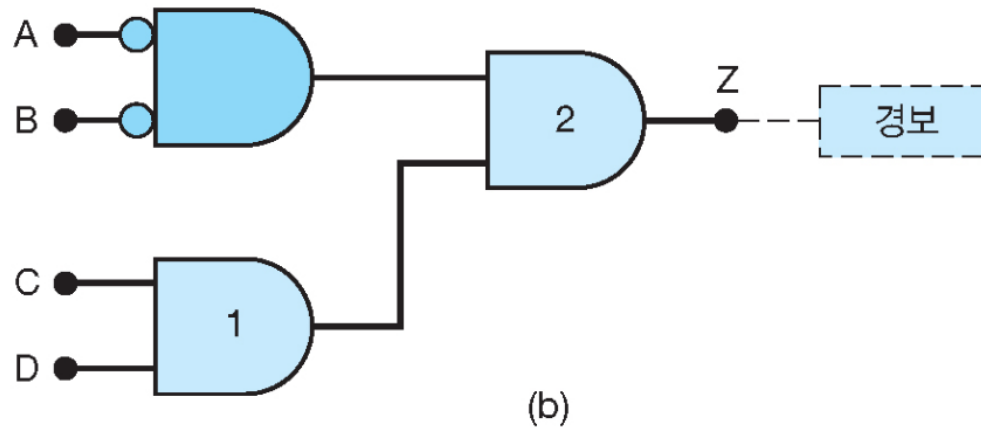
종합하면 A1=A2=A3A4=A5=A6=A7=1 이고 A0=0, IN 또는 OUT이 1이거나 모두 1일 때 HIGH가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

<학습성과 평가문제>

1. 예제 3-22, 3-23에 있는 방법을 사용하여 그림 3-37(b) 회로의 출력을 활성화 시키는데 필요한 조건을 구하라

<풀이>



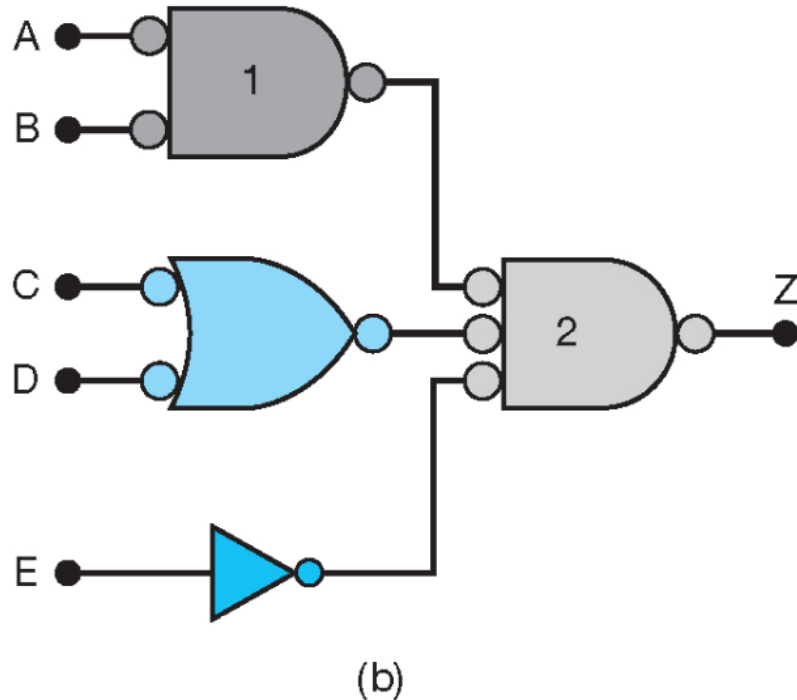
출력 Z는 $A=B=0$ 이고 $C=D=1$ 일 때만 HIGH가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

<학습성과 평가문제>

2. 그림 3-38(b) 회로에 대해 문제 1을 반복하라

<풀이>



출력 Z는 $A=B=0$, $E=1$ 이고 C와 D 중 하나 또는 둘 모두 0일 경우만 LOW가 된다

3-14 어떤 게이트 표현을 사용할 것인가?

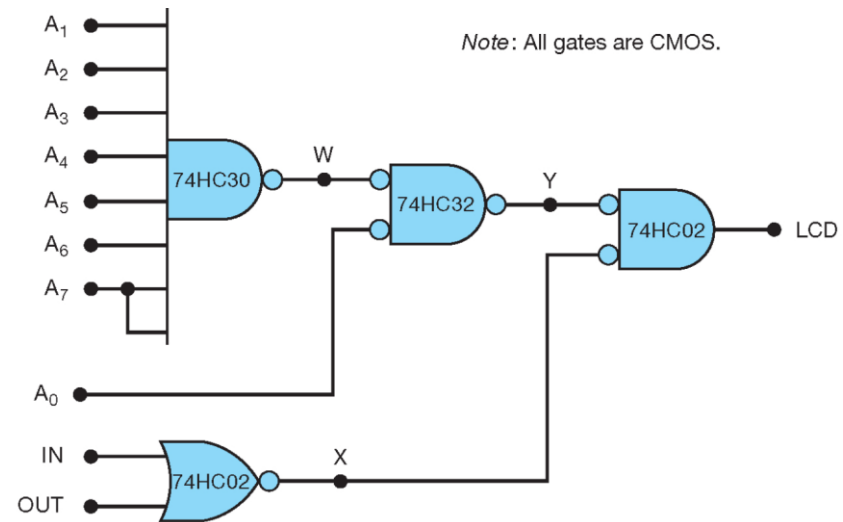
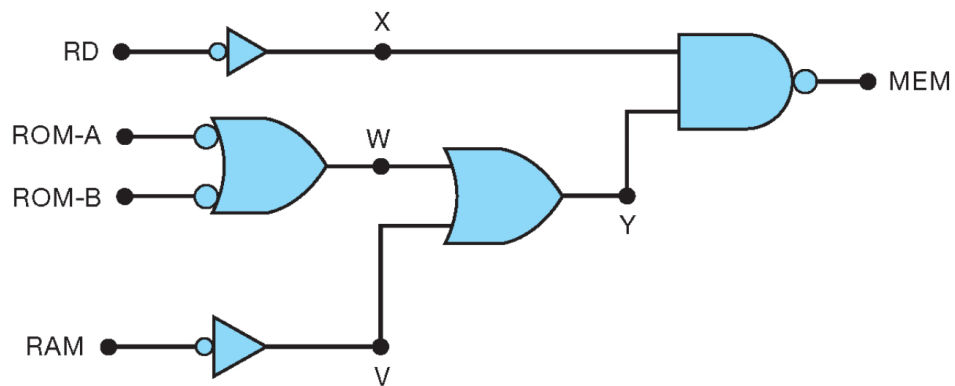
<학습성과 평가문제>

3. 그림 3-39 에는 몇 개의 NAND 게이트가 있는가?

<풀이> 2개

4. 그림 3-40 에는 몇 개의 NOR 게이트가 있는가?

<풀이> 2개



3-14 어떤 게이트 표현을 사용할 것인가?

<학습성과 평가문제>

5. 모든 입력이 활성화 상태일 때 그림 3-38(b)의 출력값은 무엇인가?

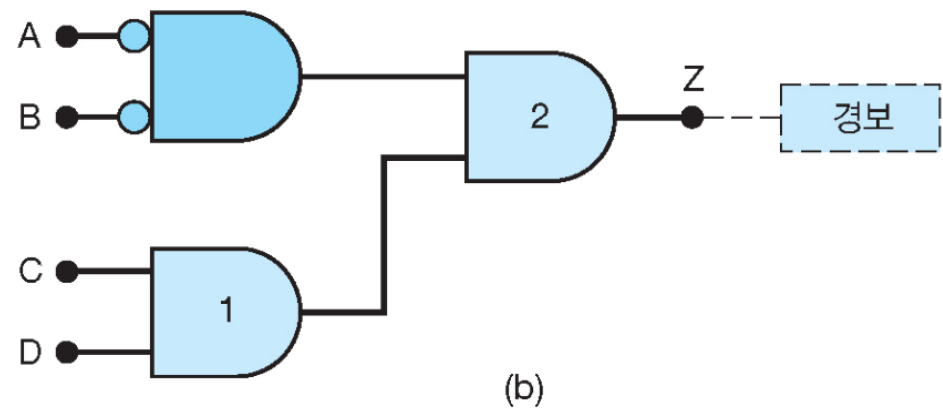
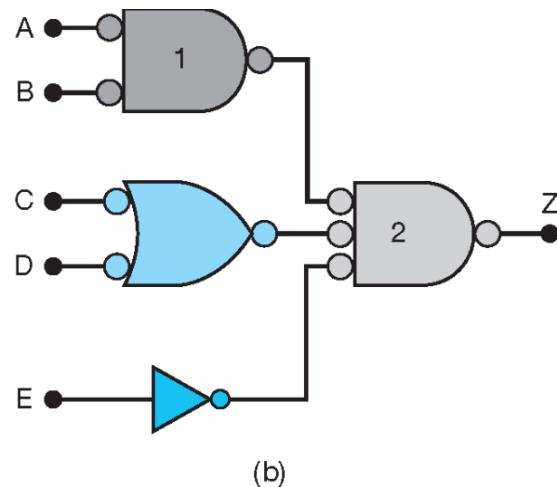
<풀이> LOW

6. 그림 3-37(b)에서 경보 출력이 활성화 되기 위한 조건은 무엇인가?

<풀이> $A=B=0, C=D=1$

7. 다음중에서 LOW 활성화 되는 신호는 어느 것인가? RD, W' , R/W'

<풀이> W'



3-15 전파 지연

➤ Propagation Delay 정의

- 시스템에서 입력이 들어온 후 적절한 출력이 만들어질 때 까지의 지연되는 시간
 - 논리 회로 속도는 전파 지연과 밀접한 관계
- 논리 회로에 적용되는 부품들은 전파 지연 값을 언급된 데이터 시트를 가지고 있다
 - 제품 응용을 위해서는 회로가 충분히 빠르게 동작함이 확인되어야 한다

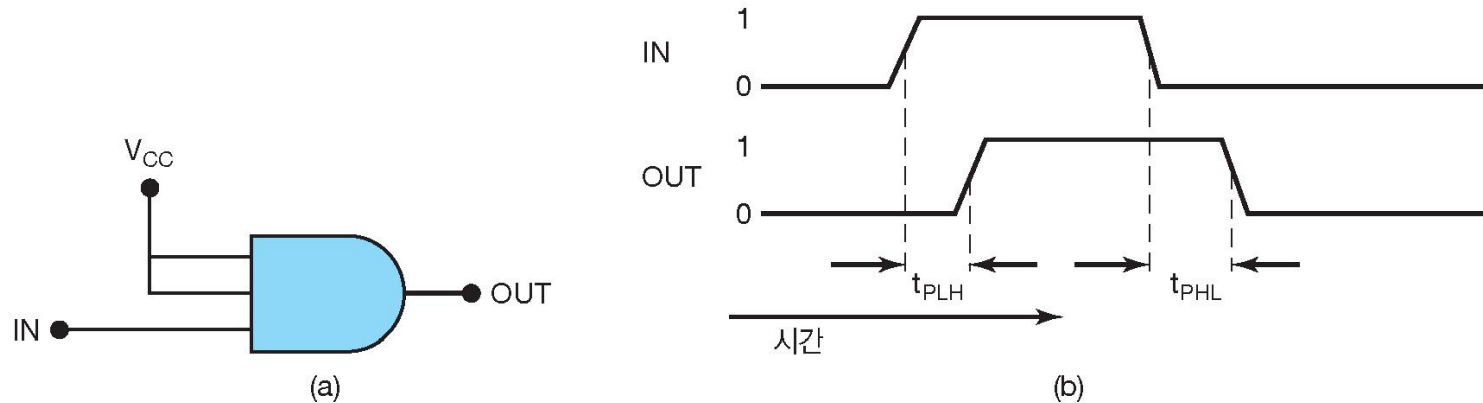


그림 3-41 논리 게이트의 전파 지연 시간 측정.

3-15 전파 지연

<학습성과 평가문제>

1. 전파지연 시간을 측정할 때 신호의 천이가 왜 수직으로 되지 않는가?

<풀이> 시간축은 나노세컨드이고 상태를 바꾸는데에는 상대적으로 시간이 많이 걸린다

2. 신호가 수직으로 천이 되지 않을 때 시간 측정을 어느 부분에서 해야 하는가?

<풀이> 입력의 50% 지점에서 출력의 50% 지점 까지

3. 입력이 들어 온 후 출력이 HIGH에서 LOW로 될 때 까지의 시간 측정 변수는 무엇인가?

<풀이> t_{PHL}

4. 입력이 들어 온 후 출력이 LOW에서 HIGH로 될 때 까지의 시간 측정 변수는 무엇인가?

<풀이> t_{PLH}

3-16 논리회로 기술 방법 요약

- **지금까지 논리회로 동작을 공부한 이유**
 - 이 논리 결정을 표현할 수 있어야 한다
 - 이 논리 기능을 결합시켜 의사 결정 시스템을 구성할 수 있어야 한다
- **지금까지 공부한 기본 논리 기능을 표현하는 방법 요약**
 - 일상 용어로 논리 표현
 - 진리표
 - 전통적인 논리 기호
 - 부울 대수 표현식
 - 타이밍 다이어그램

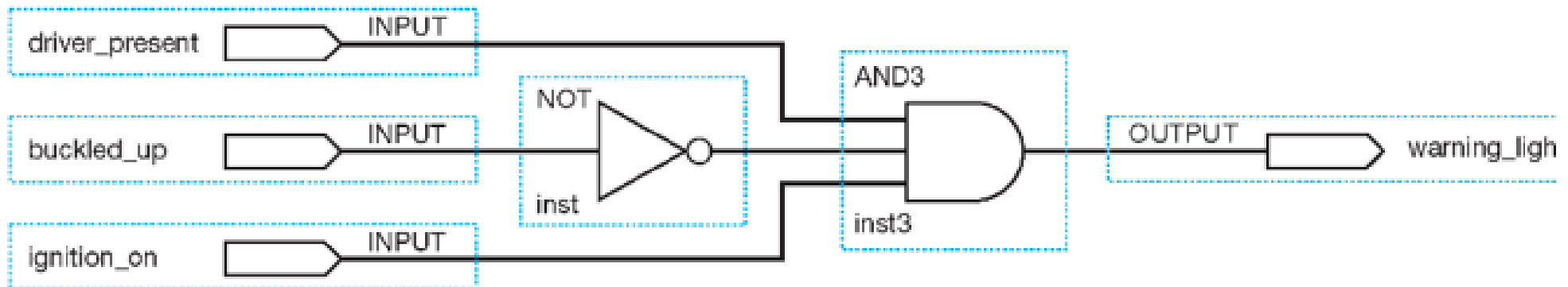
3-16 논리회로 기술 방법 요약

〈예제 3-24〉 아래 회로를 부울 대수 표현식, 논리기호를 사용한 회로도, 진리표와 타이밍 다이어그램으로 나타내어라
운전자가 승차 상태이고 AND 운전자가 안전벨트를 매지 않았고 (NOT) AND 시동을 걸었으며 THEN 정보등을 켜라

$$\text{warning_light} = \text{driver_present} \cdot \text{buckled_up} \cdot \text{ignition_on}$$

(a)

회로도



(b)

3-16 논리회로 기술 방법 요약

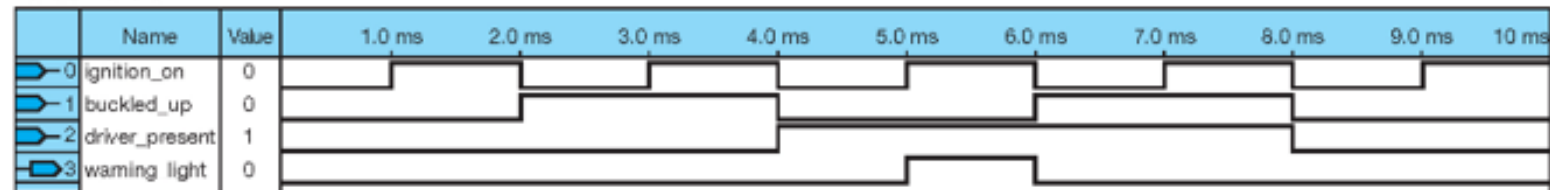
〈예제 3-24〉 운전자가 승차 상태이고 AND 운전자가 안전벨트를 매지 않았고 (NOT) AND 시동을 걸었으며 THEN 정보등을 켜라

진리표

driver_present	buckled_up	ignition_on	warning_light
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

(c)

타이밍 다이어그램



(d)

그림 3-42 논리회로의 묘사 방법. (a) 부울 표현식. (b) 회로도. (c) 진리표. (d) 타이밍 다이어그램.

3-17 하드웨어 기술 언어와 프로그램 언어 비교

➤ HDL – Hardware Description Languages

- 논리회로를 표현하기 위한 정형화된 컴퓨터 언어
- VHDL vs AHDL
 - VHDL(Very High Speed Integrated Circuit Hardware Description Language)
 - ✓ 1980년대 초반 미국 국방성(DoD)에서 개발
 - ✓ IEEE에서 표준화
 - ✓ 회로 설계를 실제 소자로 프로그래밍하는 비트 패턴으로 변환하는 데 널리 사용
 - AHDL(Altera Hardware Description Language)
 - ✓ Altera사에서 개발하여 소유권을 보유하고 있는 언어
 - ✓ Altera사가 프로그래밍 가능한 로직 디바이스(PLD)를 구성하기 위해 개발
 - ✓ 어떤 논리회로라도 표현할 수 있는 범용 언어가 아니라 Altera 개발환경에 사용
 - ➔ 디지털 시스템을 Altera PLD(Programmable Logic Device)로 프로그램

3-17 하드웨어 기술 언어와 프로그램 언어 비교

➤ 컴퓨터 프로그램 언어

- 컴퓨터는 작업 목록을 따라 작동하며 각 작업은 순차적으로 수행
- 동작 속도는 컴퓨터가 각 명령을 얼마나 빨리 실행할 수 있는지에 따라 결정
- 대표적 컴퓨터 프로그램 언어: BASIC, C, JAVA ...

➤ 하드웨어 기술 언어와 프로그래밍 언어를 구별하는 것은 중요

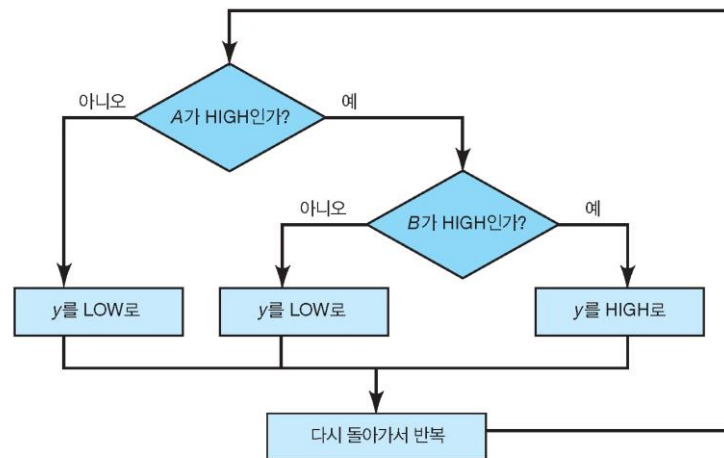
- 하드웨어 기술 언어는 하드웨어 구성을 나타내기 위한 것이라면 프로그램 언어는 컴퓨터가 필요한 과제를 수행할 수 있도록 명령어의 순서를 나타낸 것
 - 두 경우 모두 언어는 소자를 프로그래밍하는 데 사용
- 컴퓨터의 동작과 논리 소자의 동작은 상이
 - 디지털 논리 회로의 동작 속도는 입력 변화에 대해 출력이 얼마나 빨리 반응할 수 있는지에 따라 결정
 - 컴퓨터의 동작 속도는 컴퓨터가 각 명령어를 얼마나 빨리 수행할 수 있는지에 따라 결정

3-17 하드웨어 기술 언어와 프로그램 언어 비교

<예제 3-25> $y = AB$ 연산 수행에서 논리회로와 컴퓨터 연산 비교

- 논리회로는 간단히 AND 게이트로 구성
 - 출력 y 는 A와 B의 입력 이 동시에 HIGH 가 된 후 10ns 이내에 HIGH 가 된다
- 컴퓨터는 결정을 하는 명령어들로 이루어진 프로그램을 실행해야 한다
 - 각 명령어가 20ns 정도 소모한다면 입력의 변화에 대응하기 위해서는 2-3개(40~60ns) 명령어가 필요
(그림에서 순서도를 이루는 각 모양은 명령어 하나에 해당)

그림 3-43 컴퓨터 프로그램의 의사 결정 과정.



3-17 하드웨어 기술 언어와 프로그램 언어 비교

<학습성과 평가문제>

1. HDL은 무엇의 약자인가?

<풀이> 하드웨어 기술 언어(Hardware Description Language)

2. HDL을 사용하는 목적은 무엇인가?

<풀이> 디지털 회로와 그것의 동작을 기술하기 위함이다

3. 컴퓨터 프로그램 언어를 사용하는 목적은?

<풀이> 컴퓨터에게 처리할 업무의 순차적인 목록을 주기 위함이다

4. HDL과 컴퓨터 프로그램 언어의 핵심적인 차이점은 무엇인가?

<풀이> > HDL은 동시에 수행하는 하드웨어 회로를 기술하나 컴퓨터 프로그램은 한번에 실행한다

5. 어디서 AHDL을 만들었는가?

<풀이> > 알테라 회사

6. 어디서 VHDL을 만들었는가?

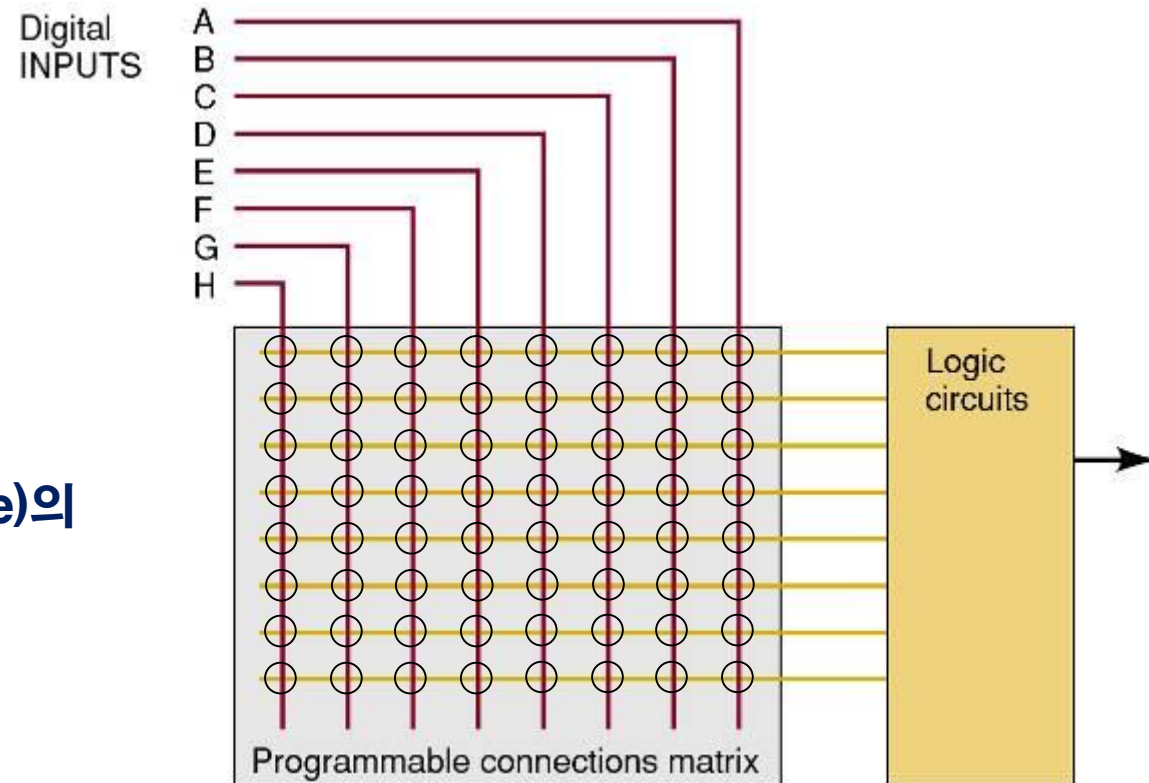
<풀이> > 미 국방성

3-18 논리회로를 PLD로 구현하는 법

PLD는 프로그래밍이 가능한 논리 소자로
내부에는 수 천 개의 논리 게이트가 그물 형식으로 들어 있다

이 프로그램이 가능한 배선은 수
천개의 Connected(1)이나 Not
Connected(0)의 연결 중의 하나

행 (horizontal wire)과 열 (vertical wire)의
각 교차점은 프로그램 가능한 연결점



3-18 논리회로를 PLD로 구현하는 법

➤ 하드웨어 기술 언어

- 개인용 컴퓨터에서 쉽게 다루고 저장할 수 있는 형식
- 설계자가 회로의 동작을 간단하면서도 편리한 방법으로 구현
 - 컴퓨터는 컴파일러(Compiler)라고 부르는 특수한 소프트웨어로 하드웨어 기술 언어를 PLD에 실을 수 있는 0과 1의 패턴으로 변환한다
 - 하드웨어 기술 언어를 잘 사용할 수 있으면 부울대수, 회로도, 또는 진리표 를 다루는 것 보다 PLD 프로그램 하기가 훨씬 쉬워 진다

3-18 논리회로를 PLD로 구현하는 법

<학습성과 평가문제>

1. PLD는 무엇의 약자인가?

<풀이> 프로그램 가능한 논리 소자(Programmable Logic Device)

2. PLD에서 회로는 전자회로적으로 어떻게 구성 되는가?

<풀이> 스위칭 매트릭스에서 신호를 연결 하거나 끊는 것에 의해 재구성 한다

4. 컴파일러가 하는 일은 무엇인가?

<풀이> > 스위칭 매트릭스 구성을 위해 HDL 코드를 비트 유형으로 변환

3-19 HDL 의 형식과 구문 표현

➤ 다이어그램의 왼쪽에는 입력, 오른쪽에는 출력, 중간 기호는 내부 동작을 정의

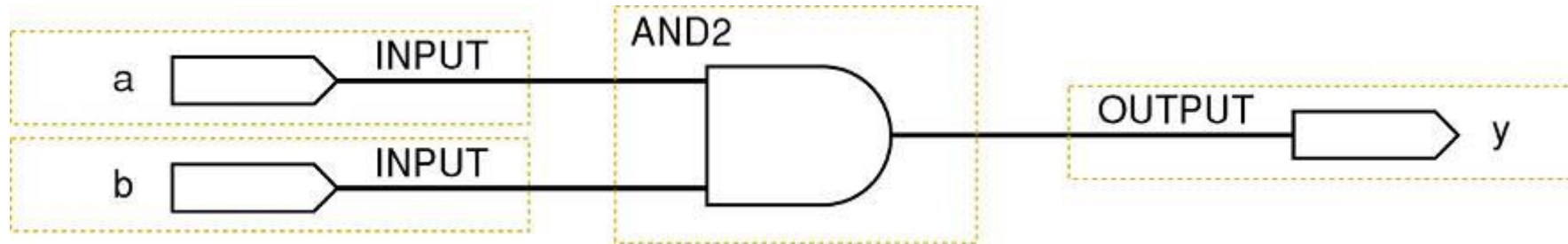


그림 3-46 HDL 파일의 구조.

▪ HDL 파일 구조

- 컴퓨터에 의해 해석되는 언어는 엄격한 구문(Syntax) 규칙을 따라야 합니다
- 형식(Format)은 입력, 출력 및 입력의 정의, 입력에 대한 출력 응답의 정의를 나타냅니다.



3-19 HDL 의 형식과 구문 표현

- **텍스트 기반 언어에서는 설명 된 회로에 이름을 지정해야 합니다.**
 - 동작 정의는 입출력 (I / O) 정의 다음에 오는 명령문 세트에 포함됩니다.
- **입력 및 출력 (포트)은 포트의 특성에 따라 이름을 지정하고 정의해야 합니다**
 - 모드는 입력인지 출력인지 또는 둘 다인지를 정의
- **유형은 비트 수와 그 비트가 그룹화되고 해석되는 방법을 나타냅니다**
 - 단일 비트 입력은 0과 1의 두 가지 값만 가질 수 있습니다
 - 4 비트 2 진수는 16 개의 다른 값 (00002 - 11112) 중 하나를 가질 수 있습니다.

3-19 HDL 의 형식과 구문 표현

➤ AHDL을 사용한 부울 연산 표현

- SUBDESIGN 이라는 키워드는 회로 블록의 이름을 부여
 - 이 경우 회로 이름은 and_gate
(파일 이름은 반드시 and_gate.tdf)
- 입력/출력을 위한 변수는 쉼표로 구분되고 뒤에 INPUT/OUTPUT 가 따라옴
 - 입 / 출력 정의는 괄호로 묶음
 - 변수가 다중 비트로 지정되지 않으면 단일 비트 유형으로 가정
 - 단일 출력 비트는 OUTPUT 모드로 선언
- AHDL 회로의 동작을 설명하는 Statement는 키워드 BEGIN과 END 사이의 논리 섹션에 포함
 - END 뒤에 반드시 세미콜론을 붙임
 - BEGIN과 END 사이의 Statement는 항상 동시에 평가

```
SUBDESIGN and_gate
(
    a, b      :INPUT;
    y        :OUTPUT;
)
BEGIN
    y = a & b;
END;
```

&	AND
#	OR
!	NOT
S	XOR

3-19 HDL 의 형식과 구문 표현

➤ VHDL을 사용한 부울 연산 표현

- ENTITY라는 키워드는 회로 블록에 이름을 부여
 - 이 경우 회로 이름은 and_gate
 - 컴파일러에서 명명 한 변수는 소문자
- 키워드 PORT는 컴파일러에게 입력을 정의하고 이 회로 블록으로 출력한다는 것을 알려줌
 - BIT 표시는 목록의 각 변수는 단일 비트임을 컴파일러에게 알려줌
- ARCHITECTURE 선언은 블록 내부의 모든 작업을 설명하는 데 사용
 - 모든 ENTITY에는 하나 이상의 ARCHITECTURE가 연결되어 있어야 함
- 본문 (BEGIN과 END 사이)에는 블록 작동에 대한 설명이 있습니다.

```
ENTITY and_gate IS
PORT ( a, b :IN BIT;
      y :OUT BIT);
END and_gate;
ARCHITECTURE ckt OF and_gate IS
BEGIN
    y <= a AND b;
END ckt;
```

3-19 HDL 의 형식과 구문 표현

<학습성과 평가문제>

1. ENTITY 선언의 역할은 무엇인가?

<풀이> 입력과 출력을 정의 한다

2. 어느 부분이 회로의 정의 부분을 정의하는 핵심 부분인가?

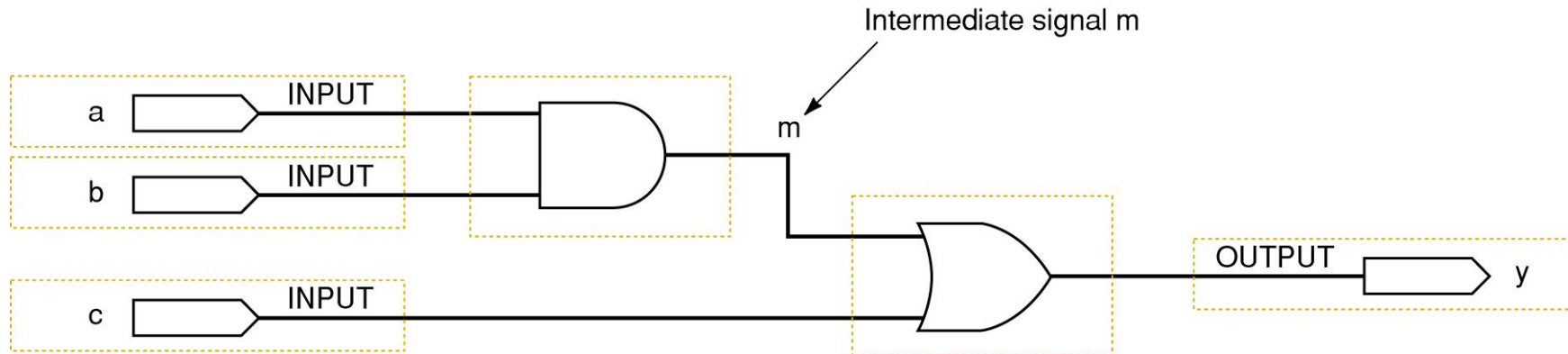
<풀이> Architecture 선언부

4. 논리 신호에 값을 할당하는 할당(Assignment) 연산자는 무엇인가?

<풀이> > <=

3-20 중간 신호

- 많은 설계에서 회로 블록 내부의 신호 점 (예 : 물린 노드 또는 로컬 신호)을 정의해야 할 필요가 있습니다.



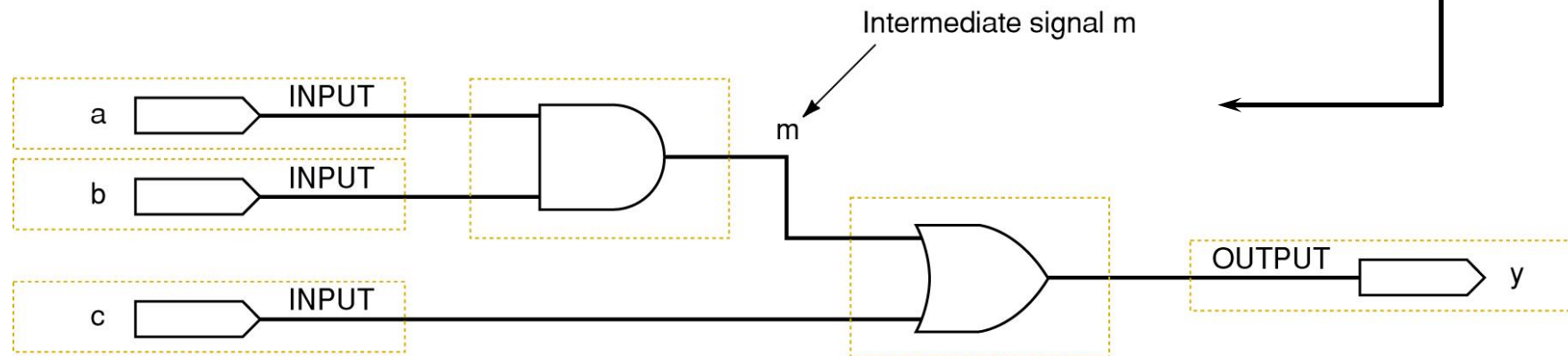
3-20 중간 신호

➤ ADHL의 숨겨진 마디

▪ AHDL Local Signals

- 주석은 % 문자로 묶임.
- 두 개의 대시 뒤의 텍스트는 문서 전용
- 키워드 VARIABLE은 중간 신호를 정의
- 키워드 NODE는 변수의 특성을 지정

```
1  % Intermediate variables in AHDL (Figure 3-49)
2  Digital Systems 11th ed
3  NS Widmer
4  MAY 24, 2010      %
5  SUBDESIGN fig3_50
6  (
7  a,b,c      :INPUT;  -- define inputs to block
8  y          :OUTPUT; -- define block output
9  )
10 VARIABLE
11 m          :NODE;   -- name an intermediate signal
12 BEGIN
13 m = a & b;      -- generate buried product term
14 y = m # c;     -- generate sum on output
15 END;
```



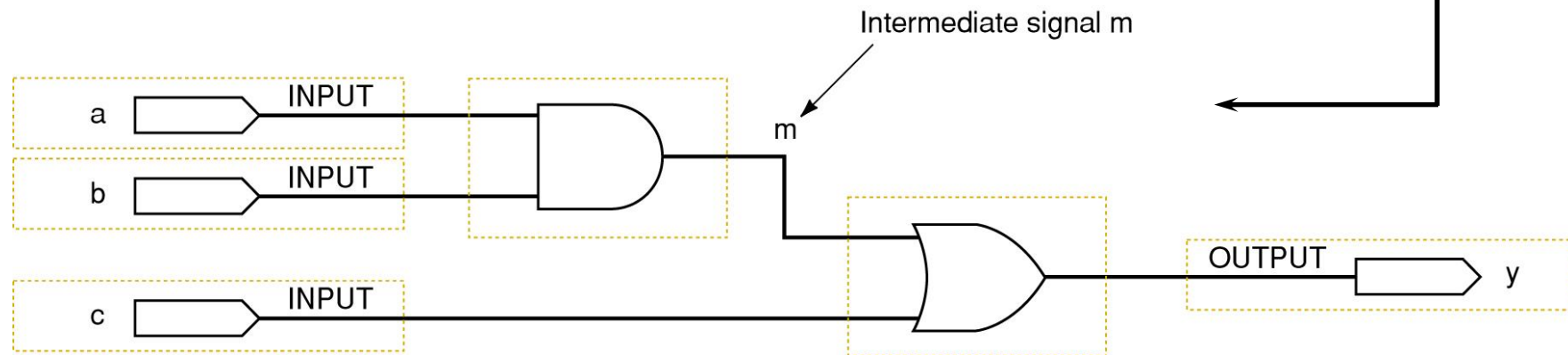
3-20 중간 신호

➤ VHDL의 숨겨진 마디

▪ VHDL Local Signals

- 두 개의 대시 뒤의 텍스트는 문서 전용
- 키워드 SIGNAL은 중간 신호를 정의
- 키워드 BIT는 신호 유형을 지정

```
1  -- Intermediate variables in VHDL (Figure 3-49)
2  -- Digital Systems 11th ed
3  -- NS Widmer
4  -- MAY 24, 2010
5
6  ENTITY fig3_51 IS
7  PORT( a, b, c  :IN BIT;    -- define inputs to block
8        y        :OUT BIT); -- define block output
9  END fig3_51;
10
11 ARCHITECTURE ckt OF fig3_51 IS
12     SIGNAL m      :BIT;    -- name an intermediate signal
13
14 BEGIN
15     m <= a AND b;         -- generate buried product term
16     y <= m OR c;         -- generate sum on output
17
18 END ckt;
```



3-19 HDL 의 형식과 구문 표현

<학습성과 평가문제>

1. 중간 변수를 위한 지정된 명칭은 무엇인가?

<풀이> NODE

2. 중간 변수는 어디에서 선언 하는가?

<풀이> I/O를 정의한 다음과 BEGIN 앞에서 선언한다

3. M식과 y식에서 어느 것이 먼저 오는지에 따라 차이가 발생 하는가?

<풀이> > 차이 없다

4. 한줄의 주석을 위해 사용되는 문자는 무엇인가?

<풀이> > %

5.

<풀이> > ...

감사합니다 !



for details about this presentation, please email to sschoi@wsu.ac.kr