



컴퓨터 구조

논리회로기초

고려대학교 세종캠퍼스 인공지능사이버보안학과
구 자 훈

목차

❖ 학습 목표

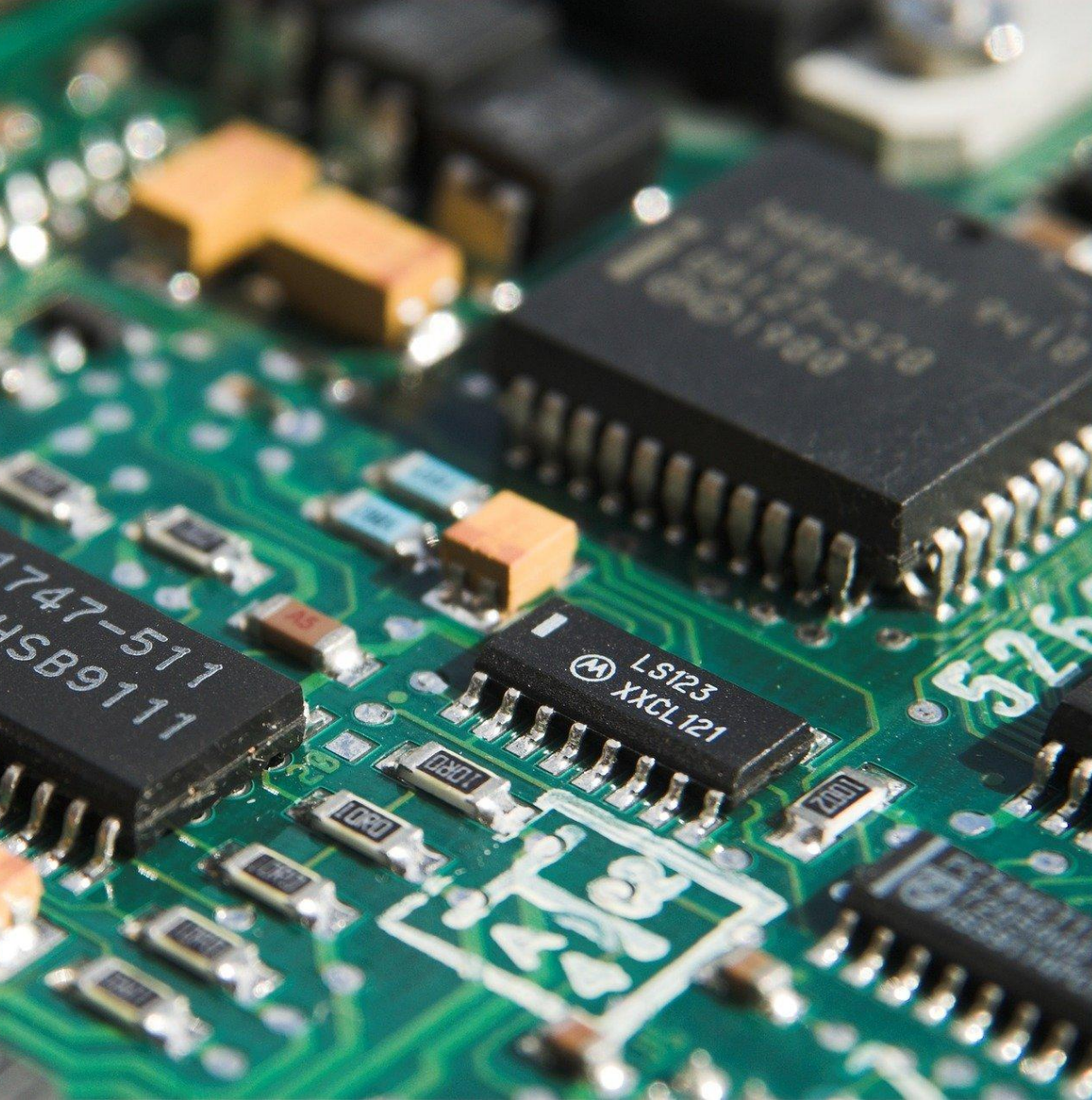
- 부울 대수
: 논리 회로에 대한 수학적 배경과 논리 연산 규칙을 이해한다.
- 논리 게이트(logic gates)의 종류와 동작을 이해한다.

❖ Part 1.

- 논리
- 논리 연산 규칙
- 논리게이트

❖ Part 2.

- 퀴즈
- 요약



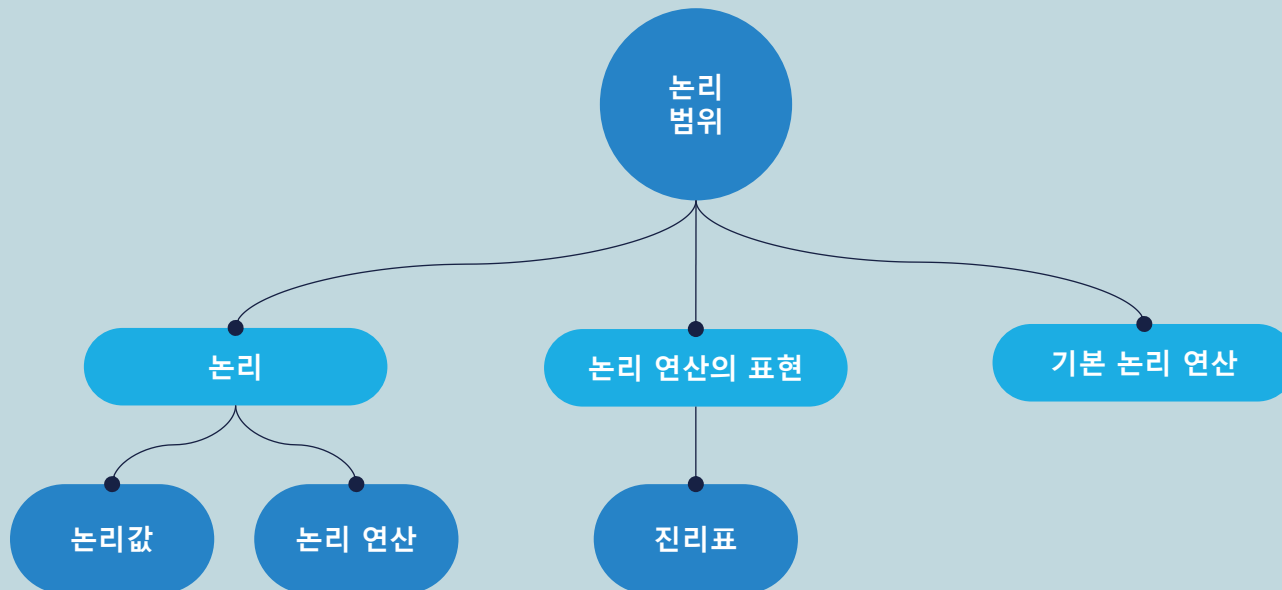
Part 1

3.1 논리

3.2 논리 연산 규칙

3.3 논리 게이트

3.1 논리 범위



01 논리 범위

논리

논리

- 참과 거짓을 다루는 학문
- 명제(proposition)과 참(true)/거짓(false)인지를 결정
- 단순 명제로 복잡명제의 참/거짓을 결정
- 컴퓨터구조의 논리는 2진수를 참/거짓에 대응시키는 시스템

논리값 (logic value)

- 논리가 다루는 값 → 참/거짓 대응 자리에 따라 무게가 다름
- 컴퓨터의 데이터는 전기 흐름 과 같은 안정된 물리 신호로 표현되며 이 신호를 1, 0으로 대응시켜 처리 결과를 얻을 수 있음

논리값	거짓(false)	참(true)
2진수	0	1
스위치	닫힘(off)	열림(on)
전기 신호	끊김(Low)	흐름(High)

01 논리 범위

논리

논리상수 (logic constant) / 논리 변수(logic variable)

- 논리값을 표현
- 논리 상수는 고정된 값 = {false, true} = {0, 1}
- 논리 변수는 논리 상수 중 하나의 값을 갖는 변수

논리 변수(logic variable)

- 문자열로 표시
- 예) x, y, z, a1, a2, alarm, bell
- 입력 변수(입력 신호) / 출력 변수(출력 신호)로 구성

논리 연산(logic operation)

- 입력 변수와 출력 변수의 관계
- 논리 연산 표현 =
- {논리식(logical equation), 진리표(truth table), 논리 회로도(logic gate)} → 오늘 배울 내용!

02 논리 연산의 표현

진리표(truth table)

입력변수 목록	출력변수 목록
입력 변수 값의 조합	해당 조합에 대한 출력 값

Ex) 짝수 패리티 진리표

- 문자열 내 1비트의 모든 숫자가 짝수 또는 홀수인지를 보증하기 위해 전송하고자 하는 데이터의 각 문자에 1 비트를 더하여 전송하는 방법.

- 2가지 종류의 패리티 비트가 있다.
(홀수 패리티, 짝수 패리티)

x	y	z	F
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

03 논리 범위 기본 논리 연산

논리곱 (AND)

입력 X, Y
모두 1일 경우 1값 반환

논리곱(AND) $Z = X \cdot Y = XY = X \text{ AND } Y$		
X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

논리합 (OR)

입력 X, Y 중에서
하나만 1일 경우 1값 반환

논리합(OR) $Z = X + Y = X \text{ OR } Y$		
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

논리 부정 (NOT)

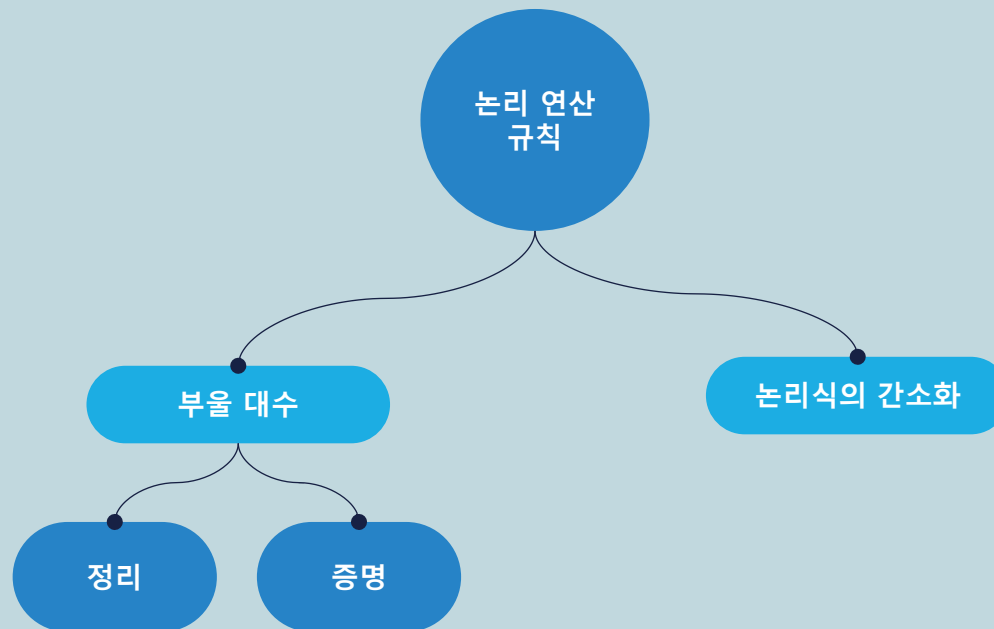
입력 x와 반대 값

논리부정(NOT) $Z = \bar{X}$ 또는 $Z = X'$	
X	Z
0	1
1	0

3.2 논리 연산 규칙

학습 목표

- 부울대수의 기본 법칙 숙지
- 부울대수를 이용한 논리식 간소화



01 논리 연산 규칙 부울대수

대수(algebra)

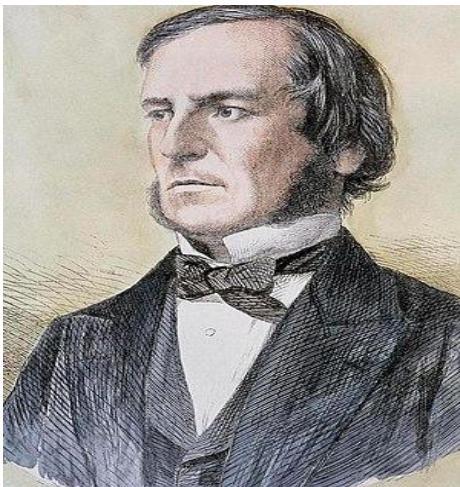
- 일련의 공리(axioms)을 만족하는 수학적 구조
- 공리: 수학이나 논리학 따위에서 증명이 없이 자명한 진리로 인정되며, 다른 명제를 증명하는 데 전제가 되는 원리. (네이버 사전)
- 원소의 집합(set of elements) & 연산자(operator)의 관계

연산자

연산에 참여하는 항의 수에 따라서 달라짐

- 이항 연산자(binary operator)
 - 실수 연산: $+$, $-$, \times , \div
 - 논리 연산: AND, OR
- 단항 연산자(unary operator)
 - 실수 연산: 음수 $-$, 제곱근(square root), 로그(log)
 - 논리 연산: NOT

01 논리 연산 규칙 부울 대수



부울 (George Boole, 1815-1864)

- 일반 대수 이론을 논리값에 적용하는 대수 창안

부울 대수 (Boolean algebra)

- 원소의 집합 = $\{0, 1\}$
- 연산자 = $\{AND(\cdot), OR(+), NOT(')\}$

쌍대식(dual equation)

- 논리식에 대하여 $\{0 \Leftrightarrow 1, AND \Leftrightarrow OR\}$ 로 교체하여 만든 논리식
- 쌍대식의 정리: 원래의 논리식이 참이면 쌍대식도 항상 참이다.

- $x \cdot x = x \quad \Leftrightarrow \quad x + x = x$
- $x \cdot 0 = 0 \quad \Leftrightarrow \quad x + 1 = 1$
- $(x')' = x$

01 논리 연산 규칙 부울 대수 정리

1

닫힘 (closure)

2

결합 법칙
(associative law)

3

교환 법칙
(commutative law)

4

분배 법칙
(distributive law)

5

항등원
(identity element)

6

역원(inverse) 또는
보수(complement)

01 논리 연산 규칙 부울 대수 정리

닫힘(closure)

- 연산의 결과값이 다시 그 집합의 원소에 속한다.
- 부울 대수는 AND(\cdot), OR($+$), NOT($'$) 연산에 대하여 닫혀 있다.

결합법칙(associative law)

$$(x * y) * z = x * (y * z) \text{ for all } x, y, z \in S$$

- \cdot 와 $+$ 연산에 대하여 결합 법칙 성립

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \quad \Leftrightarrow \quad (x + y) + z = x + (y + z)$$

교환법칙(commutative law)

$$x * y = y * x \text{ for all } x, y \in S$$

- \cdot 와 $+$ 연산에 대하여 교환 법칙 성립

$$(x \cdot y) = (y \cdot x) \quad \Leftrightarrow \quad (x + y) = (y + x)$$

01 논리 연산 규칙 부울 대수 정리

분배 법칙(distributive law)

$$x * (y \cdot z) = (x * y) \cdot (x * z)$$

- $(\cdot, +), (+, \cdot)$ 연산에 대하여 분배 법칙 성립

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z) \quad \Leftrightarrow \quad x + (y \cdot z) = (x + y) \cdot (x + z)$$

항등원(identity element)

- 항등원 : 자기 자신이 반환되게 해주는 값
- \cdot 에 대한 항등원은 1, $+$ 에 대한 항등원은 0

$$e * x = x * e = x \text{ for every } x \in S$$

$$x \cdot 1 = x \quad \Leftrightarrow \quad x + 0 = x$$

역원(inverse) 또는 보수(complement)

$$x * y = e$$

- 논리 변수 x 에 대한 역원(보수)은 x'

$$x \cdot x' = 0 \quad \Leftrightarrow \quad x + x' = 1$$

01 논리 연산 규칙 부울 대수 증명

진가표에 의한 분배 법칙 증명

$$x \bullet (y + z) = (x \bullet y) + (x \bullet z)$$

입력변수			분배 법칙의 왼쪽 식		분배 법칙의 오른쪽 식		
x	y	z	y+z	$x \bullet (y+z)$	$(x \bullet y)$	$(x \bullet z)$	$(x \bullet y) + (x \bullet z)$
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

01 논리 연산 규칙 드모르간의 법칙 (De Morgan's law)

드모르간의 법칙 (De Morgan's law)

$$(x + y)' = x' \bullet y' \Leftrightarrow (x \bullet y)' = x' + y'$$

$$(x_1 + x_2 + \cdots + x_n)' = x_1' \bullet x_2' \cdots \bullet x_n' \Leftrightarrow (x_1 \bullet x_2 \cdots \bullet x_n)' = x_1' + x_2' \cdots + x_n'$$

드모르간의 법칙 증명



입력변수		왼쪽 논리식		오른쪽 논리식		
x	y	x+y	(x+y)'	x'	y'	x'·y'
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0

02 논리 연산 규칙 논리식의 간소화

간소화(simplification)

- 더 간단한 논리회로로 구현 (논리 게이트 수 감소)

예제

- 풀어봅시다.

$$x + x \bullet y =$$

$$x \bullet (x + y) =$$

$$x + x' \bullet y =$$

$$x \bullet (x' + y) =$$

02 논리 연산 규칙 논리식의 간소화

간소화(simplification)

- 더 간단한 논리회로로 구현 (논리 게이트 수 감소)

예제

- 풀어봅시다.

$$x + x \bullet y = x$$

$$x \bullet (x + y) = x$$

$$x + x' \bullet y = x + y$$

$$x \bullet (x' + y) = x \bullet y$$

02 논리 연산 규칙 복잡한 간소화

문제 : $f(x, y, z) = x'yz + xyz' + xyz$

$$\begin{aligned} f(x, y, z) &= x'yz + xyz' + xyz \\ &= xyz' + x'yz + xyz + xyz && // \ x = x + x \\ &= (xyz' + xyz) + (x'yz + xyz) && // \ + \text{ 연산 순서 변경} \\ &= xy(z' + z) + (x' + x)yz && // \ x'y + xy = (x' + x)y \\ &= xy \cdot 1 + 1 \cdot yz && // \ x' + x = 1 \\ &= xy + yz \end{aligned}$$

02 논리 연산 규칙 간소화 원리

인접항(adjacent terms)을 찾아 하나로 합친다.

인접항

논리식에 포함된 AND항(OR 항) 중에서
논리 변수 하나의 표현이 다른 항

문제 : 논리항에 대한 인접항은?

- (1) $x \cdot y \rightarrow x' \cdot y, x \cdot y'$
- (2) $(x+y)$
- (3) $x \cdot y \cdot z'$
- (4) $(x' + y' + z)$

카르노 맵(Karnaugh map)

- 인접항을 인접한 곳에 배치하여 간소화 하는 방법
- 궁금한 학생은 논리회로를 자세히 공부해 볼 것!

02 논리 연산 규칙 간소화 원리

문제 : 2변수 논리항 xy 에 대한 인접항 결합

- x 가 다른 항: $xy + x'y =$ _____
- y 가 다른 항: $xy + xy' =$ _____

문제 : 3변수 논리항 xyz 에 대한 인접항 결합

- x 가 다른 항: $xyz + x'yz =$ _____
- y 가 다른 항: $xyz + xy'z =$ _____
- z 가 다른 항: $xyz + xyz' =$ _____

02 논리 연산 규칙 간소화 예제

문제 : 인접항 풀이: $f(x, y, z) = x'yz + xyz' + xyz$ (힌트 : 같은 항을 복사)

[풀이] $f(x, y, z) = x'yz + xyz' + xyz = xy + yz$

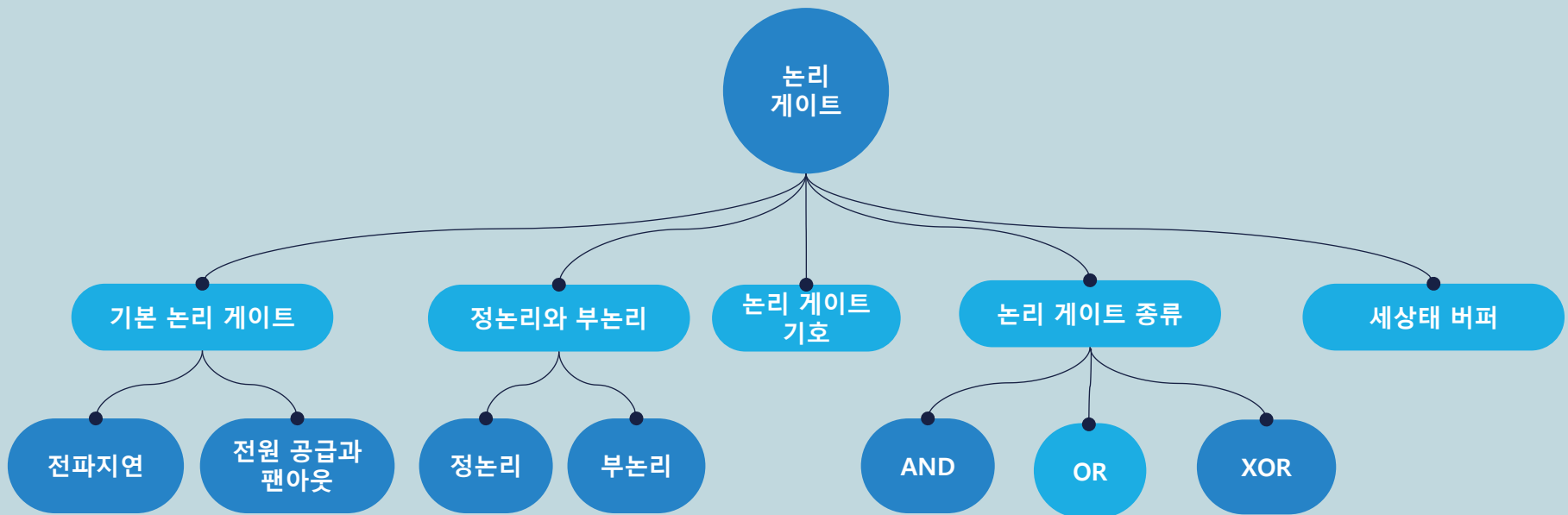
문제 : $f(x, y, z) = x'y'z' + x'y'z + xy'z' + xy'z$

[풀이] $f(x, y, z) = x'y'z' + x'y'z + xy'z' + xy'z$
 $= x'y' + xy'$
 $= y'$

3.3 논리 게이트

학습 목표

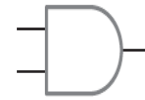
- 논리게이트의 동작 표현에 쓰이는 관례(convention) 이해
- 여러 가지 논리게이트에 대한 기호와 동작 이해



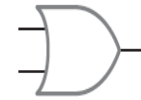
01 논리 게이트 기본 논리 게이트

논리 게이트 : 신호가 통과하는 문

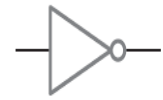
- 입력 신호에 따라서 출력 신호가 결정된다.
- 논리 게이트는 반도체 소자인 트랜지스터로 만들어진다.
- 0/1, 거짓과 참, 전압 레벨의 높고 낮음



2입력 AND 게이트



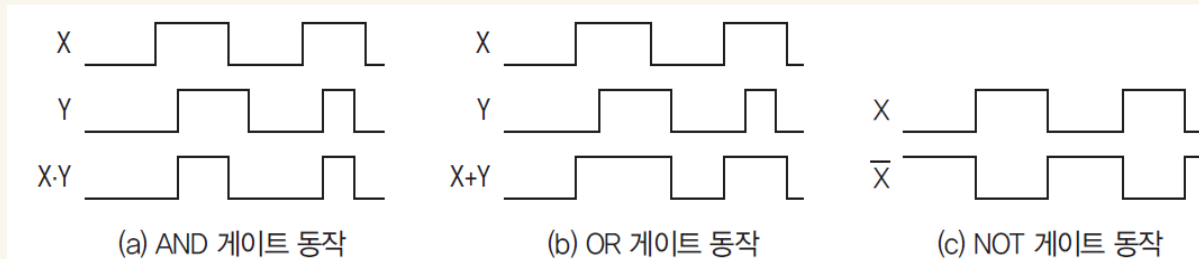
2입력 OR 게이트



NOT 게이트

타이밍 다이어그램 (timing diagram)

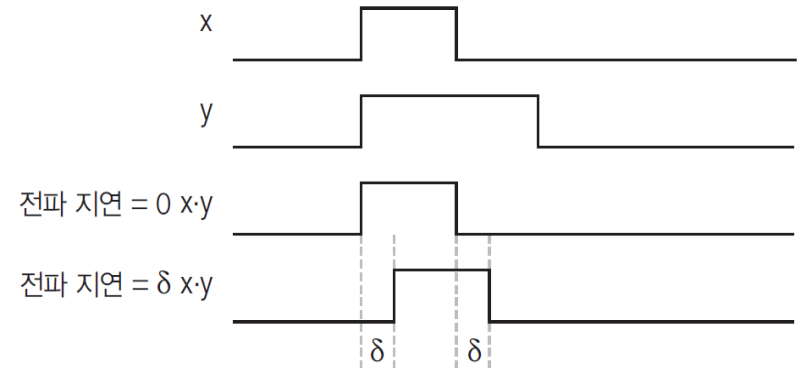
- 이론적으로는 0,1로 디지털시스템 동작이 표현되지만,
하드웨어 논리게이트는 전압 레벨이 변하는 전기신호로 동작



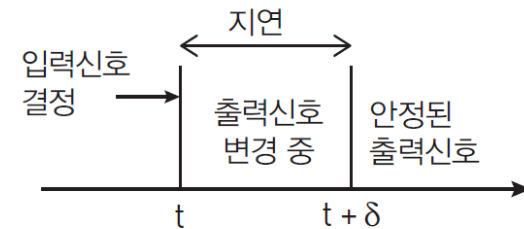
01 논리 게이트 전파 지연

전파 지연(propagation delay)

- 입력 신호에 따라 출력 신호가 변하는 시간 간격
- 신호가 게이트를 통과하는 시간
- 게이트 통과 때마다 누적
- 일반적으로 수 나노 초
- 전파지연시간만큼 지나야 안정된 출력
 - 입력신호가 통과하는 게이트가 많을 수록 전파 지연도 누적됨
 - 입력신호가 적게 만드는 것이 바람직하다



(a) AND 게이트의 전파 시간



(b) 논리회로의 출력 지연

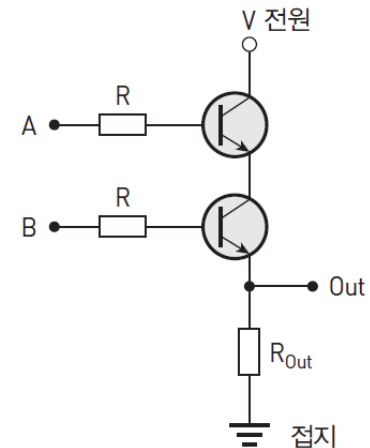
01 논리 게이트 전원 공급과 팬아웃

공급 전원 (supply voltage)

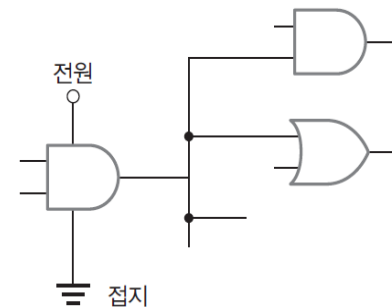
- 실제 디지털소자는 전원(+), 접지(-) 로 연결됨
- 단 논리게이트를 그림으로 그릴 때는 전원/접지를 표시하지 않음

팬아웃(fan-out)

- 이론적으로 게이트에는 무한의 입력이 가능
- 하나의 출력에 연결할 수 있는 입력의 수의 제한을 팬아웃 이라고 함
 - 공급전원으로 생성하는 출력 신호의 세기(power)로 인한 문제
 - 팬아웃이 많을수록 전력이 약해짐
 - 일반적으로 8~10정도 값을 가짐



(a) 트랜지스터 AND 게이트



(b) 팬아웃(fan-out)

02 논리 게이트 정논리와 부논리

상태
(state)

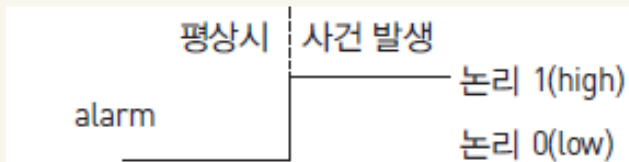
- 디지털 시스템의 논리 소자 출력이 변하지 않고 일정시간 유지되는 그 순간의 값
- 논리 회로의 출력은 0 또는 1이기 때문에 둘 중 하나는 중요한 사건(event)가 발생한 것을 의미함
- 이진 시스템에서는 사건의 발생을 0 또는 1로 표현할 수 있으며, 이를 정논리 / 부논리로 표현 할 수 있음

02 논리 게이트

정논리와 부논리

정논리

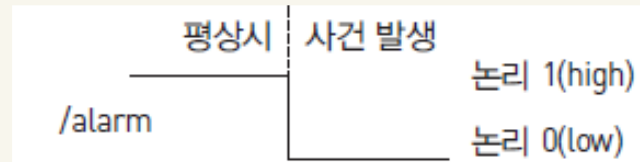
- 평상시 상태 0
- 사건 발생 1



(a) 정논리

부논리

- 평상시 상태 1
- 사건 발생 0
- 신호 이름에 '/' 또는 '-'



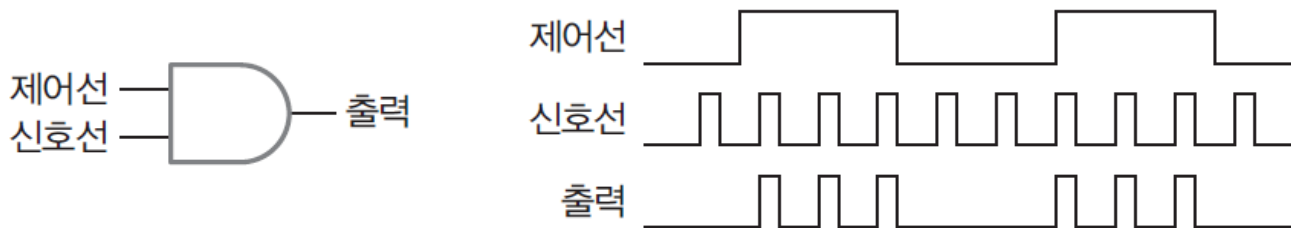
(b) 부논리

입력 / 출력과 정논리 / 부논리 조합

	정논리	부논리
입력	입력 정논리-출력 정논리	입력 정논리-출력 부논리
출력	입력 부논리-출력 정논리	입력 부논리-출력 부논리

03 논리 게이트 기호



논리게이트 기호





신호 전달	신호 부정(NOT) 입력/출력에 추가	AND 연산 입력선 2개 이상	OR 연산 입력선 2개 이상
(a) 삼각형	(b) 작은 원	(c) 직선형 입력	(d) 곡선형 입력

04 논리 게이트 종류 : AND

<2-입력 AND게이트>

입력		출력	AND 게이트 기호	
X	y	$F = x \cdot y$	입력 정논리 - 출력 정논리	입력 부논리 - 출력 부논리
0	0	0		
0	1	0		
1	0	0		
1	1	1		



<2-입력 NAND게이트>

입력		출력	NAND 게이트 기호	
X	y	$F = (x \cdot y)'$	입력 정논리 - 출력 부논리	입력 부논리 - 출력 정논리
0	0	1		
0	1	1		
1	0	1		
1	1	0		



04 논리 게이트

논리 게이트 종류 : OR

<2-입력 OR 게이트>

입력		출력	OR 게이트 기호	
X	y	$F = x+y$	입력 정논리 - 출력 정논리	입력 부논리 - 출력 부논리
0	0	0		
0	1	1		
1	0	1		
1	1	1		


<2-입력 NOR 게이트>

입력		출력	NOR 게이트 기호	
x	y	$F = (x+y)'$	입력 정논리 - 출력 부논리	입력 부논리 - 출력 정논리
0	0	1		
0	1	0		
1	0	0		
1	1	0		


04 논리 게이트

논리 게이트 종류 : XOR

<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	

<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	

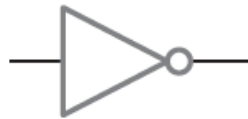
04 논리 게이트 버퍼, NOT 게이트, 3입력 게이트

버퍼 (buffer)

- 입력신호의 논리값이 변하지 않고 그대로 출력으로 전달하는 논리소자
- NOT게이트는 출력이 부논리인 버퍼라고 할 수 있음
- 정논리 NOT 게이트와 부논리 NOT게이트와 일치함
 - 정논리 NOT게이트는 입력신호 1을 0으로 반전하는 것이 중요한 이벤트
 - 부논리 NOT게이트는 입력신호를 0을 1로 반전하는게 중요한 이벤트



(a) 버퍼



(b) 정논리 NOT 게이트



(c) 부논리 NOT 게이트

< 버퍼와 NOT 게이트 >

04 논리 게이트

버퍼, NOT 게이트, 3입력 게이트

3입력 게이트

- 입력신호가 하나인 NOT 게이트와 버퍼를 제외한 게이트는 입력이 3개 이상일 수 있음
 - 3입력 AND의 경우 입력이 모두 1일 때 출력이 1임
 - 3입력 OR게이트의 경우 입력 중 적어도 하나가 1일 경우 출력이 1
 - 3입력 XOR의 경우 입력 중 1의 수가 홀수일 때 출력이 1임



(a) $F = X \cdot Y \cdot Z$



(b) $F = X + Y + Z$



(c) $F = X \oplus Y \oplus Z$



(d) $F = (X \cdot Y \cdot Z)'$



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



(f) $F = (X \oplus Y \oplus Z)'$

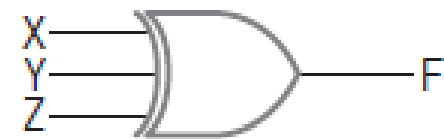
< 3-입력 게이트 >

04 논리 게이트

3입력 XOR 게이트

XOR 게이트: 입력 중 1이 홀수 개 있을 때 출력 1

X	Y	Z	$F = X \oplus Y \oplus Z$
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



(c) $F = X \oplus Y \oplus Z$

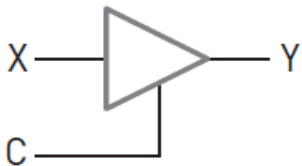
04 논리 게이트

상태 버퍼 : 3상태 버퍼

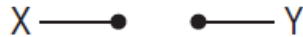
3상태 버퍼(Tri-state buffer)

- 버퍼에 제어신호를 추가해 출력을 전달하거나 전달하지 않도록 만든 버퍼
- X는 입력, Y는 출력, C는 제어 신호
- (b)에서의 출력은 입력이 끊어진 상태($C=0$)로 이 상태를 하이 임피던스 상태라고 함 (high impedance) / $Z = \text{High Impedance}$
- (c)에서의 출력은 입력과 연결되어 있는 상태 ($C=1$) 입력신호를 그대로 출력

< 정논리 3상태 버퍼 >



(a) 정논리 3상태 버퍼



(b) $C=0$ 일 때 등가회로



(c) $C=1$ 일 때 등가회로

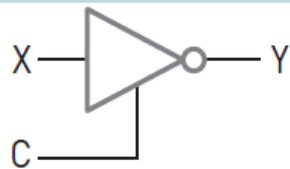
04 논리 게이트

상태 버퍼 : 3상태 버퍼

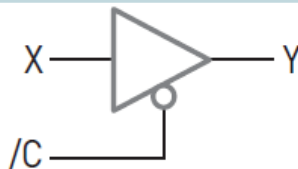
3상태 버퍼 종류

- 3상태 버퍼에는 정논리 3상태 버퍼, 부논리 3상태 버퍼 등 다양한 종류가 있다.

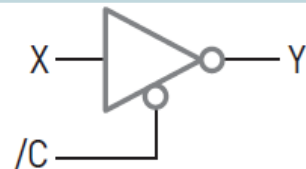
< 여러가지 형태의 3상태 버퍼 >



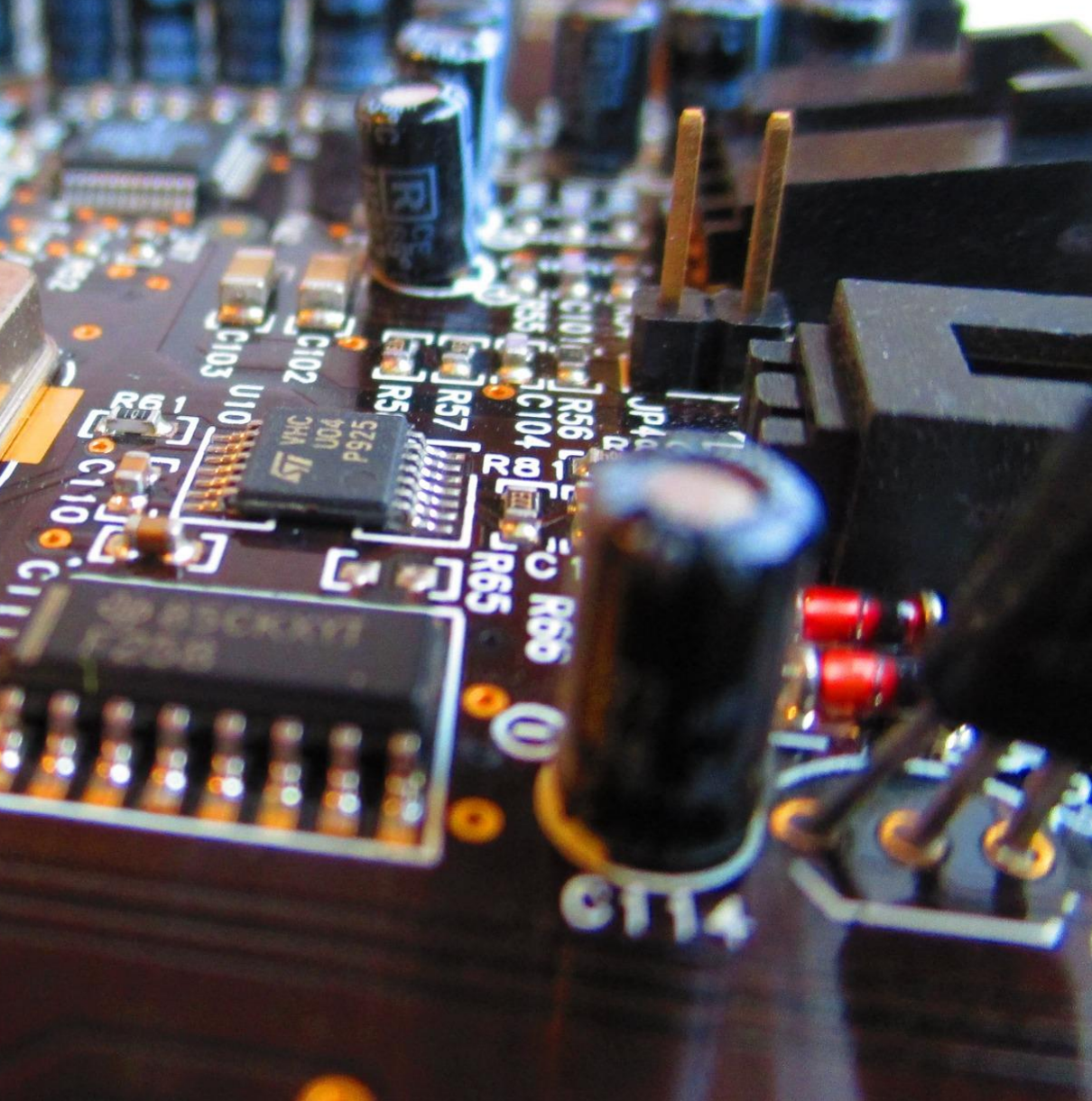
(a) 정논리 3상태 NOT 게이트



(b) 부논리 3상태 버퍼



(c) 부논리 3상태 NOT 게이트



Part 2

3.4 퀴즈

3.5 요약

01 논리회로 기초 문제

논리식 간소화

(1) $xy + xy'$

(2) $f(x,y,z) = x'y'z' + x'y'z + xy'z' + xy'z$

쌍대식 구하기

(3) $x \cdot x = x \quad \Leftrightarrow$

(4) $x \cdot 0 = 0 \quad \Leftrightarrow$

논리게이트

2입력 AND, OR, XOR 논리게이트를 그리시오.

01 논리회로 기초 문제

논리식 간소화

(1) $xy + xy'$

(2) $f(x,y,z) = x'y'z' + x'y'z + xy'z' + xy'z$

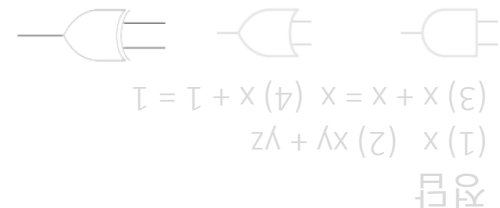
쌍대식 구하기

(3) $x \cdot x = x \quad \Leftrightarrow$

(4) $x \cdot 0 = 0 \quad \Leftrightarrow$

논리게이트

2입력 AND, OR, XOR 논리게이트를 그리시오.



02 논리회로 기초

02 논리회로 기초 요약

3.1 논리

- 논리값(0/1, false/true), 논리 상수, 논리 변수 개념 소개

3.2 논리 연산 규칙

- 부울 대수: 논리값에 대한 연산 규칙
- 논리식의 간소화: 부울 대수 활용, 인접항 통합

3.3 논리 게이트

- 정논리, 부논리
- 논리게이트 종류와 기호
- 3상태 버퍼

02 논리회로 기초 논리 요약

논리

- 참과 거짓을 다루는 학문
- 물리적 개념을 논리적 개념으로 변환
- 논리 상수: False/True, 0/1 (디지털 논리)
- 논리 변수: 논리 상수의 값을 갖는 변수

기본 논리 함수

- 기본 논리 함수: AND, OR, NOT

02 논리 연산 규칙 요약

논리회로 기초

부울 대수

단항	AND, OR, NOT 연산의 결과는 {0, 1}
결합법칙	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$
교환 법칙	$(x \cdot y) = (y \cdot x)$
분배 법칙	$x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
항등원	$x \cdot 1 = x$
역원(보수)	$x \cdot x' = 0$

드모르간의 법칙

$$(x + y)' = x' \cdot y'$$

논리식 간소화

- 부울 대수 정리 활용
- 인접항을 찾아 서로 다른 항 제거

02 논리회로 기초 논리 게이트 요약

논리 게이트

- 논리연산을 수행하는 하드웨어 소자

타이밍 다이아그램

- 논리게이트의 입출력 신호의 변화를 표현하는 그림

정논리 / 부논리

- 정논리: 평상시 0(low), 사건 발생 1(high)
- 부논리: 평상시 1(high), 사건 발생 0(low)

논리게이트 종류

- AND, OR, NOT, XOR, NAND, NOR, XNOR, NOT, BUFFER

3상태 버퍼

- {0, 1, high impedance}

Thank you

Jahoon Koo
(sigmao@korea.ac.kr)