

# 논리회로기초

Heeseung Jo

## 3.1 논리

논리: 참과 거짓을 다루는 학문

논리값(logic value) 대응

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

논리상수 = {false, true} = {0, 1}

논리변수: 문자열로 표시 (x, y, z, a1, a2, alarm, bell, ...)

- 입력변수(입력 신호)
- 출력변수(출력 신호)

논리연산: 입력변수와 출력변수의 관계

# 논리연산의 표현

논리연산 표현 = {논리식, 진가표, 논리회로도}

진가표(truth table)

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

(예) 짝수 패리티 진가표

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

# 기본 논리 연산

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

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

## 3.1 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: AND, OR, NOT

## 3.2 논리연산 규칙

---

### 학습 목표

- 부울대수의 기본 법칙 숙지
- 부울대수를 이용한 논리식 간소화

### 내용

- 3.2.1 부울대수
- 3.2.2 논리식의 간소화

## 3.2.1 부울대수

### 대수(algebra)

- 일련의 **공리(axioms)**을 만족하는 수학적 구조
- 원소의 집합(set of elements) & 연산자(operator)의 관계

### 연산자

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

# Definitions

## 공리(公理, axiom)

- 논리학이나 수학 등의 이론체계에서 가장 기초적인 근거가 되는 명제(命題)
- 증명할 필요가 없이 자명한 진리
- 질문이나 의문의 여지없이 참이라고 받아들여야 하는 것

## 정리(定理, theorem)

- 증명이 필요한 명제 중 증명이 완료된 명제를 정리
- 정리를 증명하기 위해 사용되는 보조적인 명제 : 보조정리(lemma)
- 정리로부터 쉽게 도출되는 부가적인 명제 : 따름정리(corollary)

## 명제(命題, proposition)

- 논리학적으로 뜻이 분명한 문장
- 참/거짓을 따질 수 있는가?
- 예) 노인들은 공경을 해야 할 대상이다



# 부울 대수

부울 (George Boole, 1815-1864)

- 논리값에 적용하는 대수 창안

부울대수

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

쌍대식(dual equation)

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

•  $x \cdot x = x \quad \Leftrightarrow$  \_\_\_\_\_

•  $x \cdot 0 = 0 \quad \Leftrightarrow$  \_\_\_\_\_

•  $(x')' = x$

# 부울대수 정리

## 1. 닫힘(closure)

연산의 결과값이 다시 그 집합의 원소에 속한다

부울대수는 AND( $\cdot$ ), OR(+), NOT(') 연산에 대하여 닫혀 있다

## 2. 결합법칙(associative law)

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

• 와 + 연산에 대하여 결합법칙 성립

$$(x \cdot y) \cdot z = x \cdot (y \cdot z) \iff (x + y) + z = x + (y + z)$$

## 3. 교환법칙(commutative law)

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

• 와 + 연산에 대하여 교환법칙 성립

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

# 부울대수 정리

## 4. 분배법칙(distributive law)

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

$(\cdot, +)$ ,  $(+, \cdot)$  연산에 대하여 분배법칙 성립

$$x \cdot (y + z) = (x \cdot y) + (x \cdot z) \iff x + (y \cdot z) = (x + y) \cdot (x + z)$$

$x \cdot y + z \neq (x \cdot y) + (x \cdot z) \implies$  연산자 우선순위에 의해 성립하지 않음

## 5. 항등원(identity element, e)

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

$\cdot$  에 대한 항등원은 1,  $+$  에 대한 항등원은 0

$$x \cdot 1 = x \iff x + 0 = x$$

## 6. 역원(inverse) 또는 보수(complement)

$$x * y = e$$

논리변수  $x$ 에 대한 역원(보수)은  $x'$

$$x \cdot x' = 0 \iff x + x' = 1$$

# 부울대수 정리 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·(y+z)	(x·y)	(x·z)	(x·y)+(x·z)
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

# 드모르간의 법칙 (De Morgan's Law)

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

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

[표 3-5] 드모르간의 법칙 증명

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

## 3.2.2 논리식의 간소화

간소화(simplification)

- 더 간단한 논리회로로 구현 (논리 게이트 수 감소)

[예제 3-1]

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

[예제 3-2]

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

[예제 3-3]

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

[예제 3-4]

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

# 복잡한 간소화 예제

[예제 3-5]  $f(x, y, z) = x'yz + xyz' + xyz$

[풀이]

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

# 간소화 원리

인접항(adjacent terms)을 찾아 하나로 합친다

인접항

- 논리식에 포함된 AND항(OR항)중에서 논리 변수 하나의 표현이 다른 항
- 변수의 수 만큼 인접항이 존재

[예제 3-6] 논리항에 대한 인접항?

(1)  $x \cdot y$                      $\rightarrow x' y, x y'$

(2)  $(x+y)$

(3)  $x \cdot y \cdot z'$

(4)  $(x' + y' + z)$

카르노 맵(Karnaugh map)

- 인접항을 인접한 곳에 배치하여 간소화 하는 방법



# 간소화 원리

예) 2변수 논리항  $xy$ 에 대한 인접항 결합

- $x$ 가 다른 항:  $xy + x'y =$

- $y$ 가 다른 항:  $xy + xy' =$

예) 3변수 논리항  $xyz$ 에 대한 인접항

- $x$ 가 다른 항:  $xyz + x'yz =$

- $y$ 가 다른 항:  $xyz + xy'z =$

- $z$ 가 다른 항:  $xyz + xyz' =$

# 간소화 예제

[예제 3-5] 인접항 풀이:  $f(x, y, z) = x'yz + xyz' + xyz$

[풀이]  $f(x, y, z) =$

[예제 3-7]  $f(x, y, z) = x'y'z' + x'y'z + xy'z' + xy'z$

[풀이]  $f(x, y, z) =$   
=  
=

## 3.2 논리연산 규칙 요약

### 부울 대수

- 닫힘                      AND, OR, NOT 연산의 결과는  $\{0, 1\}$
- 결합법칙                 $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
- 교환법칙                 $(x \cdot y) = (y \cdot x)$
- 분배법칙                 $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$
- 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

## 3.3 논리게이트

---

### 학습 목표

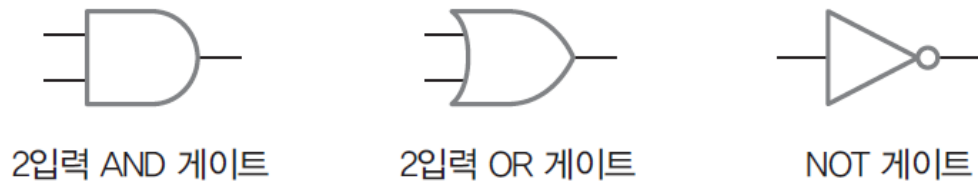
- 논리게이트의 동작 표현에 쓰이는 관례(convention) 이해
- 여러 가지 논리게이트에 대한 기호와 동작 이해

### 내용

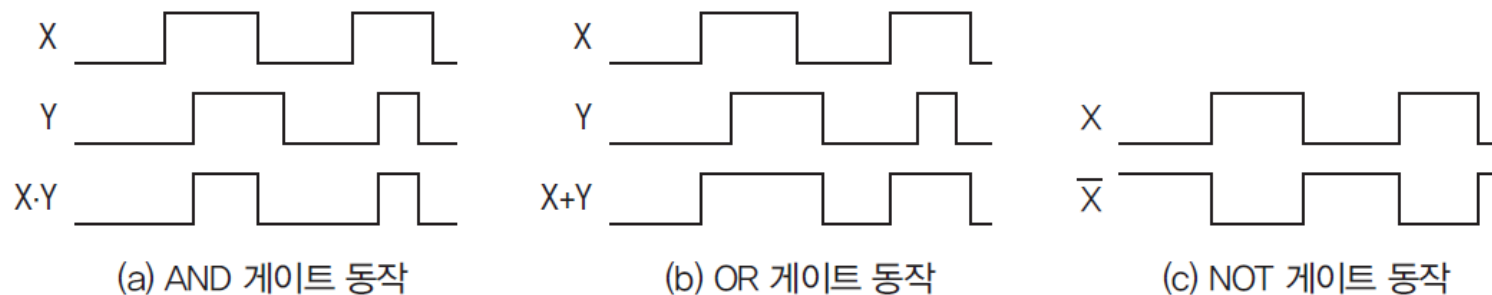
- 3.3.1 기본 논리게이트
- 3.3.2 정논리와 부논리
- 3.3.3 논리게이트 기호
- 3.3.4 논리게이트 종류
- 3.3.5 세상태 버퍼

## 3.3.1 기본 논리게이트

논리 게이트: 신호가 통과하는 문



[그림 3-1] 논리게이트 기호

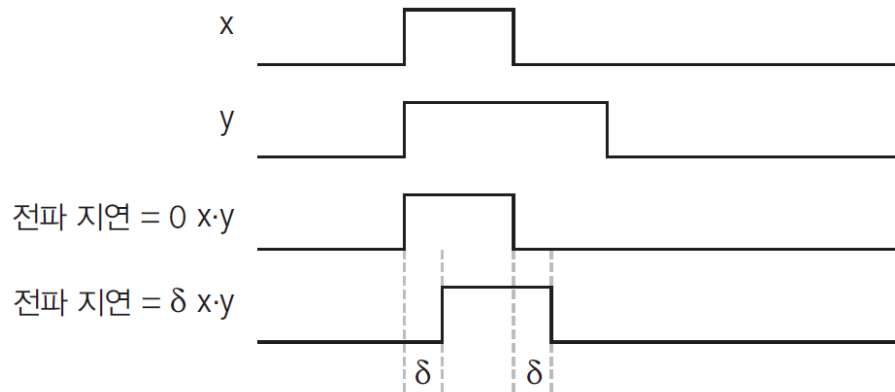


[그림 3-2] 기본 게이트의 출력에 대한 타이밍 다이어그램

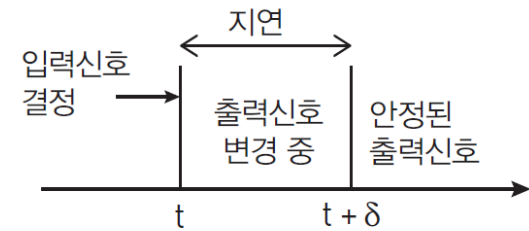
# 전파 지연

## 전파 지연(propagation delay)

- 입력신호에 따라 출력신호가 변하는 시간 간격
- 신호가 게이트를 통과하는 시간
- 게이트 통과 때마다 누적
- 일반적으로 수 나노초



(a) AND 게이트의 전파 시간



(b) 논리회로의 출력 지연

[그림 3-3] 전파 지연

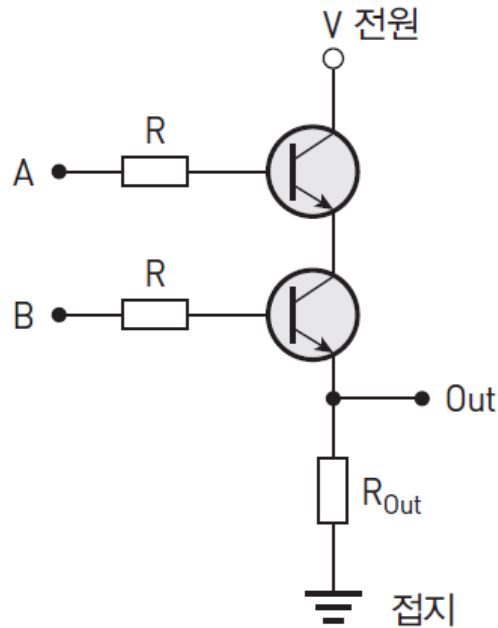
# 전원 공급과 팬아웃

## Supply Voltage

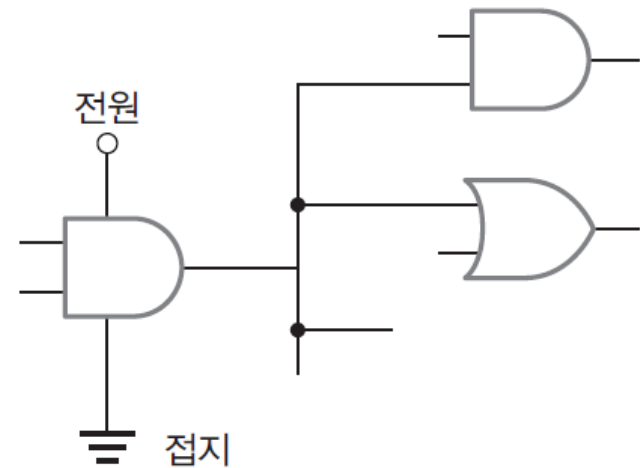
- 전원은 high level, 접지는 low level을 의미 (일반적으로 그림 생략)

## Fan-out

- 게이트를 통과한 아웃이 다른 게이트로 연결 가능
- 그 수에는 제한이 있음



(a) 트랜지스터 AND 게이트



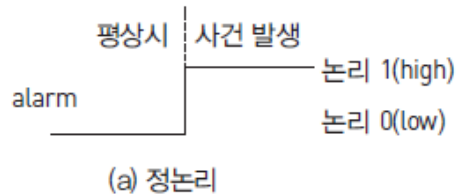
(b) 팬아웃(fan-out)

[그림 3-4] 전원 공급과 팬아웃

## 3.3.2 정논리와 부논리

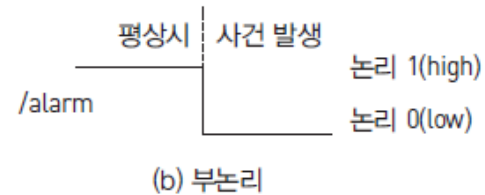
### 정논리

- 평상시 상태 0
- 사건 발생 1



### 부논리

- 평상시 상태 1
- 사건 발생 0
- 신호 이름에 / 또는 -



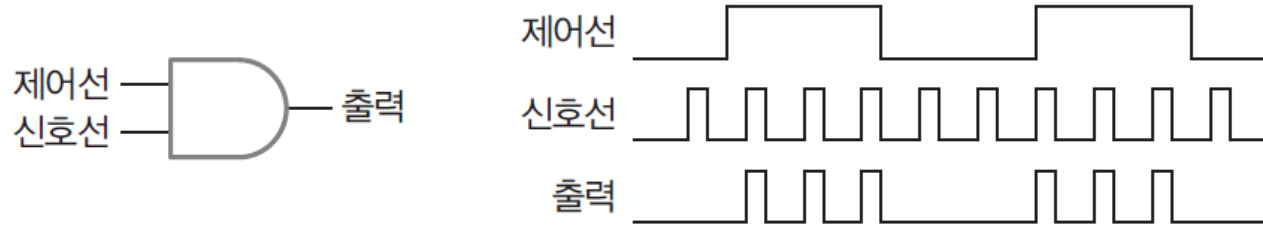
[그림 3-5] 정논리와 부논리

### 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



### 3.3.3 논리게이트 기호



[그림 3-6] 제어선과 신호선



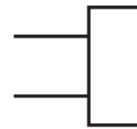
(a) 삼각형

신호 전달



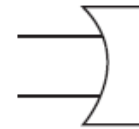
(b) 작은 원

신호 부정 (NOT)  
입력/출력에 추가



(c) 직선형 입력

AND 연산  
입력선 2개 이상





(d) 곡선형 입력



OR 연산  
입력선 2개 이상

### 3.3.4 논리게이트 종류

〈표 3-7〉 2-입력 AND 게이트



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		

〈표 3-8〉 2-입력 NAND 게이트



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		

# 논리게이트 종류 (OR)

〈표 3-9〉 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		

〈표 3-10〉 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		


# 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	

1이 홀수 개 있을 때 1

〈표 3-12〉 2-입력 XNOR 게이트

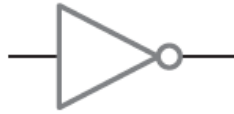
입력		출력	XNOR 게이트 기호
x	y	$F=(x\oplus y)'$	
0	0	1	
0	1	0	
1	0	0	
1	1	1	

두 입력이 같으면 1

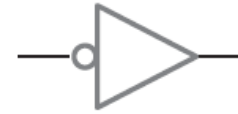
# 버퍼, NOT 게이트, 3입력 게이트



(a) 버퍼



(b) 정논리 NOT 게이트



(c) 부논리 NOT 게이트

[그림 3-8] 버퍼와 NOT 게이트



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

[그림 3-9] 3-입력 게이트

# [예제 3-8] 3입력 XOR 게이트

XOR 게이트: 입력 중 1이 홀수 있다 = 출력 1

X	Y	Z	F = $X \oplus Y \oplus Z$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

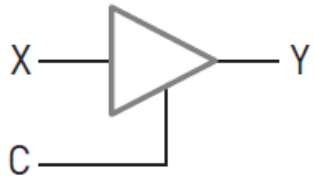


(c)  $F = X \oplus Y \oplus Z$

### 3.3.5 3상태 버퍼

#### 3상태 버퍼

- 버퍼에 제어신호를 추가하여 전달하거나 전달하지 않도록 만든 것



(a) 정논리 3상태 버퍼

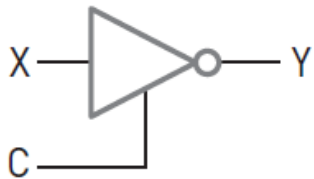


(b) C=0일 때 등가회로

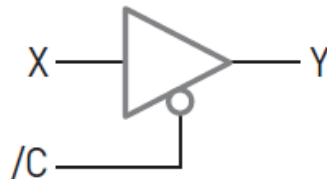


(c) C=1일 때 등가회로

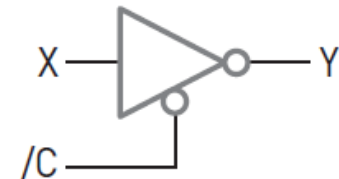
[그림 3-10] 정논리 3상태 버퍼



(a) 정논리 3상태 NOT 게이트



(b) 부논리 3상태 버퍼



(c) 부논리 3상태 NOT 게이트

[그림 3-11] 여러 가지 형태의 3상태 버퍼

## 3.3 논리게이트 요약

### 논리게이트

- 논리연산을 수행하는 하드웨어 소자

### 타이밍 다이어그램

- 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



## 3.4 논리회로 기초 요약

### 3.1 논리

- 논리값(0/1, false/true), 논리상수, 논리변수 개념 소개

### 3.2 논리연산 규칙

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

### 3.3 논리게이트

- 정논리, 부논리
- 논리게이트 종류와 기호
- 3상태 버퍼

## 제4장 조합 논리회로

- 조합 논리회로의 동작
- 빌딩 블록: 가산기, 비교기, 인코더/디코더, 멀티플렉서/디멀티플렉서