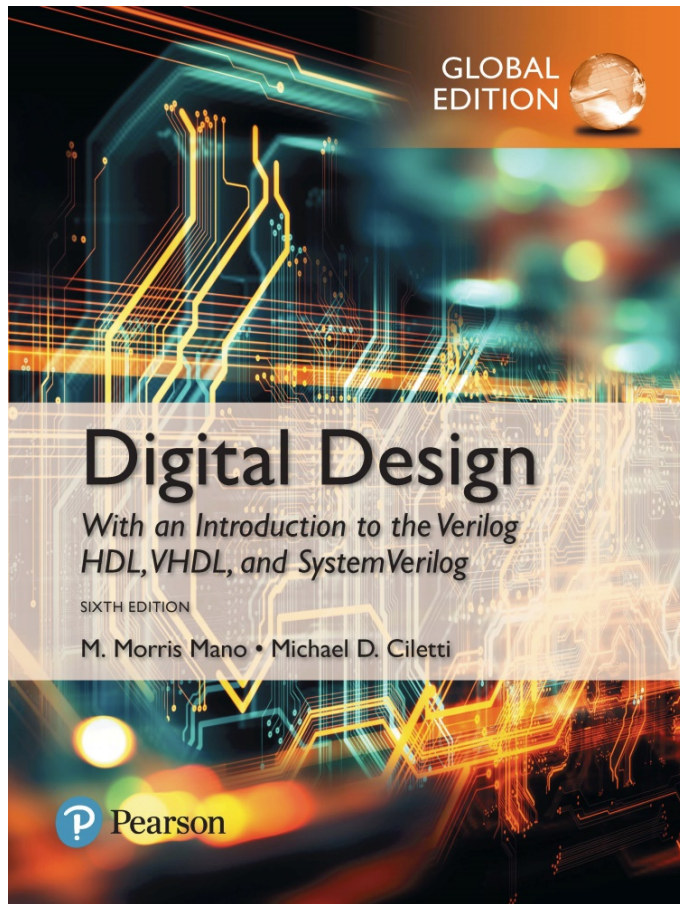


Digital Design (디지털 디자인)

With an Introduction to the Verilog HDL, VHDL, and SystemVerilog

5th or 6th Edition, Global Edition



Chapter 01

Digital Systems and Binary Numbers

디지털 시스템과 2진수 체계

전자공학과 김동한 교수

1.1 디지털 시스템

- Human (decimal: 0,1,...,9) ↔ Most computers (binary: 0,1)
- Digital system (using binary number) manipulates discrete element (2진 형태로 나타나는 정보의 이산적 요소를 처리하는 시스템)
: communication, business transactions, traffic control, space guidance, medical treatment, weather monitoring, the Internet, many other commercial, industrial, and scientific enterprises
- Electronic system
 - i) Analog (analogous: 유사) system: it treats physical amount directly
→ weight, height, voltage, current etc.
 - ii) Digital (digitus: 손가락) system : it treats physical amount by discrete elements
→ 1) naturally discrete, 2) continuous → discrete
→ In general, base-2 binary number system
 - iii) Hybrid system: Combination of analog and digital systems

1.1 디지털 시스템

Ex)

- An automobile ignition system: digital system (on/off)
- An automobile gas gauge system: analog system (The position of the indicator is an analogy of the level of gasoline in the tank)
- 현대의 모든 디지털 시스템에서의 신호는 2개의 이산된(discrete) 값만을 가지며, 이를 2진수(binary)라고 함
- 비트(bit, binary digit)라고 불리는 2진수는 0과 1을 가짐. 이러한 비트의 집합을 2진 코드(binary code)라고 함
- 컴퓨터: 디지털 시스템의 대표적인 예. CPU, 메모리, 입/출력장치, 통신
 - CPU: 프로그램에 따라 산술 및 기타 데이터 처리
 - 메모리: 프로그램과 데이터 저장

1.2 2진수

- 10진수 시스템 (decimal system): 기수(밑, base, radix)가 10이고, 계수에 10의 멍승(거듭제곱, power)을 곱함
- 예) $7392 = 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 + 2 \times 10^0$
- 2진수 시스템 (binary system): 밑이 2이고, 계수에 2의 거듭제곱을 곱함
- 예) $11010.11 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}$
- 5진수? 16진수?
- 컴퓨터에서는 2^{10} 의 값이 K, 2^{20} 의 값이 M, 2^{30} 의 값이 G, 2^{40} 의 값이 T

Table 1.1
Powers of Two.

<i>n</i>	<i>2ⁿ</i>	<i>n</i>	<i>2ⁿ</i>	<i>n</i>	<i>2ⁿ</i>
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024 (1K)	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096 (4K)	20	1,048,576 (1M)
5	32	13	8,192	21	2,097,152
6	64	14	16,384	22	4,194,304
7	128	15	32,768	23	8,388,608

1.3 기수(밑)의 변환 (10진수 \leftrightarrow 아무 진수)

- 예제 1.1) 10진수 41을 2진수로 변환
- 예제 1.2) 10진수 153을 8진수로 변환

1.3 기수(밑)의 변환 (10 진수 \leftrightarrow 아무 진수)

- 기수(밑)의 변환은 정수부분과 소수부분을 나눠서 진행
 - 10 진수 소수를 2 진수로 변환하는 것은 나눗셈 대신 곱셈을 사용하고 나머지 대신 정수를 선택함
 - 예제 1.3) $(0.6875)_{10}$ 을 2 진수로 변환
-
- 예제 1.4) $(0.513)_{10}$ 진수 8 진수로 변환

1.4 8진수와 16진수

- 2진수에서 8진수, 16진수로의 변환은 2진수의 숫자를 소수점 기준으로 숫자 3개(8진수의 경우), 숫자 4개(16진수의 경우)로 묶어서 변환함.
반대도 마찬가지임.
- 컴퓨터에서는 2진수의 특성을 가지면서 표시하는 숫자 갯수를 줄이기 위해서 8진수나 16진수를 사용함 (16진수 2개는 8bit=1byte)
- $(10\ 110\ 001\ 101\ 011\ .\ 111\ 100\ 000\ 110)_2 = (26153.7406)_8$
2 6 1 5 3 . 7 4 0 6
- $(10\ 1100\ 0110\ 1011\ .\ 1111\ 0010)_2 = (2C6B.F2)_{16}$
2 C 6 B . F 2
- $(673.124)_8 = (110\ 111\ 011\ .\ 001\ 010\ 100)_2$
6 7 3 . 1 2 4
- $(306.D)_{16} = (0011\ 0000\ 0110\ .\ 1101)_2$
3 0 6 . D

Table 1.2
Numbers with Different Bases.

Decimal (base 10)	Binary (base 2)	Octal (base 8)	Hexadecimal (base 16)
00	0000	00	0
01	0001	01	1
02	0010	02	2
03	0011	03	3
04	0100	04	4
05	0101	05	5
06	0110	06	6
07	0111	07	7
08	1000	10	8
09	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1.5 보수

- 컴퓨터에서 뺄셈 연산과 논리적 조작을 쉽게 하기 위해서 사용
- r 의 보수와 $(r-1)$ 의 보수로 나뉨. 여기서는 2진수 연산에만 사용할 것이므로 $r=2$ 임. 즉, 2의 보수와 1의 보수 사용
- $r-1$ 의 보수 (감소된 기저 보수)
 - 10진수의 9의 보수는 9에서 각 숫자를 뺀
 - 2진수의 1의 보수는 $0 \rightarrow 1, 1 \rightarrow 0$ 으로 바꿈
 - 8진수의 7의 보수는 7에서 각 숫자를 뺀
 - 546700에 대한 9의 보수는 $999999 - 546700 = 453299$
 - 012398에 대한 9의 보수는 $999999 - 012398 = 987601$
 - 1011000에 대한 1의 보수는 0100111
 - 0101101에 대한 1의 보수는 1010010

1.5 보수

- r 의 보수 (기저 보수)
 - 쉬운 방법: $r-1$ 의 보수에 1을 더함. 단 $N=0$ 일 때는 0임
 - 다른 방법: 10진수의 경우, 모든 최하위 유효숫자 0을 변화시키지 않고, 최하위 유효숫자 중 첫번째 0이 아닌 숫자를 10에서 빼고, 그 보다 높은 유효숫자들은 9에서 뺌. (246700에 대한 10의 보수는 753300)
 - 다른 방법: 2진수의 경우, 모든 최하위 유효숫자 1과 첫번째 1을 변화시키지 않고, 그 외의 높은 자리의 유효숫자는 1을 0으로 0을 1로 바꿈 (1101100에 대한 2의 보수는 0010100, 0110111에 대한 2의 보수는 1001001)
- 보수의 보스는 원래 수

1.5 보수

- 보수를 이용한 뺄셈

1. 부호가 없는 두 수의 뺄셈 $M - N$: 기본적으로 M 에 N 의 보수를 더함

1. $M \geq N$ 일 때

- M 에 N 의 보수를 더하고 최종자리올림(end carry)를 버림

2. $M < N$ 일 때

- M 에 N 의 보수를 더하고 합에 보수를 취한 후 음의 부호 추가

- 예제 1.5)

— $72532 - 3250$

- $72532 + 96570(3250 \text{에 대한 } 10 \text{의 보수}) = 169282$
- 169282 에서 최종자리올림(end carry)인 1을 버림 = 69282

- 예제 1.6)

— $3250 - 72532$

- $03250 + 27468(72532 \text{에 대한 } 10 \text{의 보수}) = 30718$

■ 32718 에 대한 10의 보수인 69282 에 -부호를 붙인 -69282



Pearson

Copyright © 2019 Pearson Education Ltd.

1.5 보수

- 예제 1.7) $X = 1010100$, $Y = 1000011$

(a) $X - Y$

- $1010100 + 0111101$ (Y에 대한 2의 보수) = 10010001
- 최종자리올림을 버림 = 0010001

(a) $Y - X$

- $1000011 + 0101100$ (X에 대한 2의 보수) = 1101111
- 1101111 의 2에 대한 보수에 -부호를 붙인 -0010001

1.5 보수

- $r-1$ 보수를 이용한 뺄셈

1. 부호가 없는 두 수의 뺄셈 $M - N$: 기본적으로 M 에 N 의 보수를 더함

1. $M \geq N$ 일 때

- M 에 N 의 $r-1$ 의 보수를 더하고 최종자리올림(end carry)를 버린 후 순환자리올림(end-around carry)를 더함

2. $M < N$ 일 때

- M 에 N 의 $r-1$ 의 보수를 더하고 합에 $r-1$ 의 보수를 취한 후 음의 부호 추가

1.5 보수

- 예제 1.8) $X = 1010100$, $Y = 1000011$

(a) $X - Y$

- $1010100 + 0111100$ (Y에 대한 1의 보수) = 10010000
- 최종자리올림을 버림 = 0010000
- 순환자리올림을 더함 = 0010001

(a) $Y - X$

- $1000011 + 0101011$ (X에 대한 1의 보수) = 1101110
- 1101110 의 1에 대한 보수에 -부호를 붙인 -0010001

1.6 부호 있는 2진수

- 숫자의 가장 왼쪽 비트를 부호, 나머지 비트를 숫자로 나타냄. 양수는 0, 음수는 1로 함 (부호-크기 방식, **signed-magnitude convention**)
 - 11001은 부호가 없는 수로 간주하면 25이고, 부호가 있는 수로 간주하면 -9임
- 부호-보수 방식(**signed-complement system**): 음수를 보수로 표현. 양수는 가장 왼쪽 비트가 0이고, 그 보수는 음수를 나타내는 1로 시작함. 2진수의 경우, 1의 보수와 2의 보수 중에 2의 보수가 많이 사용됨. 컴퓨터에서 많이 사용하는 방법
- 예) -9 표현방법

부호-크기 방식: 10001001 (+9 = 00001001)

부호화 1의 보수: 11110110 (+9 = 00001001)

부호화 2의 보수: 11110111

Table 1.3
Signed Binary Numbers.

Decimal	Signed-2's Complement	Signed-1's Complement	Signed Magnitude
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	—	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
-8	1000	—	—

1.6 부호 있는 2진수

- 산술덧셈: 부호-크기 방식에서는 부호가 다르면 큰 수에서 작은 수를 빼고 큰 수의 부호를 붙임. 즉, 부호와 크기를 비교하는 과정이 필요함. 하지만 부호-보수 방식에서는 덧셈만 하면 됨.

+6 0000110

+13 00001101

+19 00010011

+6 0000110

-13 11110011

-7 11111001

-6 11111010

+13 00001101

+7 00000111

-6 11111010

-13 11110011

-19 111011011

- 음수의 경우, 2의 보수로 바뀌서 해당하는 양수가 원지 파악하는 것이

1.6 부호 있는 2진수

- 산술뺄셈: 더하는 수(감수)를 2의 보수로 바꿔서 더함

$$-6 - (-13) = -6 + 13 \Rightarrow 11111010 - 11110011$$

$$\Rightarrow 11111010 + 00001101 = 100000111$$

$$\Rightarrow \text{끝자리올림을 제거} \Rightarrow 00000111 = +7$$

- 덧셈과 뺄셈을 수행하는데 공통된 하드웨어를 사용할 수 있으므로 컴퓨터에서 많이 사용됨

1.7 2진 코드

- 2진 코드에서 n 비트로 2^n 개에 해당하는 값을 0과 1로 표현할 수 있음 (00, 01, 10, 11)
- 2진수 시스템이 컴퓨터에 가장 자연스럽지만, 사람들은 10진수 시스템에 더 익숙함. 따라서 사람에게 익숙한 10진수를 입력 받아서 컴퓨터에 맞도록 2진수로 변환하고 연산한 다음, 결과값을 다시 10진수로 바꿔서 사람에게 출력함
- 따라서 10진수를 2진 코드(2진수가 아님)로 나타낼 필요가 있음
- 대표적인 예는 BCD 코드, 3초과 코드, 그레이코드, 2421, 8 4 -2 -1, ASCII

1.7 2진 코드

- BCD 코드 (4개의 비트 사용, 16개 가능한 조합 중에 6개는 사용 안함)
 - 1010 에서 1111은 아무 의미 없음

Table 1.4

Binary-Coded Decimal (BCD).

Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

1.7 2진 코드

- 10진수 396을 BCD 코드로 변환하면, 0011 1001 0110
- 10진수 185를 BCD 코드로 변환하면, 0001 1000 0101
하지만 이걸 2진수로 나타내면, 10111001 임. (차이를 이해해야 함)
- BCD 덧셈
 - 각 자리를 2진수 합처럼 더하고 1010과 같거나 크다면 0110을 더해서 BCD로 바꿈
- 예) $184 + 576 = 760$ 를 BCD 덧셈으로 표현

— BCD	1	1(캐리)		
	0001	1000	0100	184
+	0101	0111	0110	576
2진수 합	0111	10000	1010	
0110 더함		0110	0110	
BCD 합	0111	0110	0000	760

1.7 2진 코드

- BCD 뺄셈: 10의 보수로 변환 후 더함(9에서 각 비트를 빼고 1을 더함). 부호를 나타내는 가장 왼쪽의 비트는 양수면 0, 음수면 9임

- 예) $375 + (-240) = 135$

$$\begin{array}{r} 0\ 375 \\ +\ 9\ 760\ (\text{240의 10의 보수}) \\ \hline 0\ 135\ (\text{캐리 없음}) \end{array}$$

- 다른 10진식 코드
 - 10개의 숫자에 대한 2진 코드는 최소 4비트가 필요
 - BCD와 2421 코드는 가중코드
 - 2421 코드에서는 4를 0100이나 1010으로 나타낼 수 있음
 - 2421 코드와 3초과 코드는 자기보완적인 코드(10진수의 9의 보수가 0과 1을 서로 바꿈으로써 얻어짐. 예) 10진수 395는 3초과코드로 0110 1100 1000, 395의 9의 보수인 604는 3초과코드로 1001 0011 0111)

Table 1.5
Four Different Binary Codes for the Decimal Digits.

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, – 2, – 1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused bit combi- nations	1011	0110	0001	0010
	1100	0111	0010	0011
	1101	1000	1101	1100
	1110	1001	1110	1101
	1111	1010	1111	1110

1.7 2진 코드

- 그레이코드: 한 숫자에서 다른 숫자로 갈 때 한 비트만 변함. 예를 들면, 7에서 8로 갈 때, 0100에서 1100으로 변함. 숫자가 연속적으로 변화할 때 오류를 줄여줌(2진수에서는 7에서 8로 갈 때 0111에서 1000으로 변함)

Table 1.6
Gray Code.

Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

1.7 2진 코드

- ASCII 문자코드(American Standard Code for Information Interchange):
7비트(128개)를 사용해서 26개의 대문자, 26개의 소문자, 10개의 숫자, 32개의 특수문자, 34개의 제어문자 표현
- 대부분의 컴퓨터는 8비트가 기본 단위임.
따라서 남은 1비트는 그리스문자나 이탤릭 문자 등으로 사용되거나 (확장 ASCII코드), 패리티 비트로 사용(오류 검출)

- | | | |
|---------------|----------|----------|
| | 짝수패리티 | 홀수패리티 |
| • A = 1000001 | 01000001 | 11000001 |
| • T = 1010100 | 11010100 | 01010100 |

$b_4b_3b_2b_1$	$b_7b_6b_5$							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	'	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L		l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

Control Characters			
NUL	Null	DLE	Data-link escape
SOH	Start of heading	DC1	Device control 1
STX	Start of text	DC2	Device control 2
ETX	End of text	DC3	Device control 3
EOT	End of transmission	DC4	Device control 4
ENQ	Enquiry	NAK	Negative acknowledge
ACK	Acknowledge	SYN	Synchronous idle
BEL	Bell	ETB	End-of-transmission block
BS	Backspace	CAN	Cancel
HT	Horizontal tab	EM	End of medium
LF	Line feed	SUB	Substitute
VT	Vertical tab	ESC	Escape
FF	Form feed	FS	File separator
CR	Carriage return	GS	Group separator
SO	Shift out	RS	Record separator
SI	Shift in	US	Unit separator
SP	Space	DEL	Delete

1.8 2진 기억 장치와 레지스터

- 레지스터(register): 2진 셀의 집합. 2진 셀이란 2개의 상태(여기서는 0, 1)를 기억하는 단위
- 예) 1100 0011 1100 1001 이 16셀의 레지스터에 저장되어 있다면, 무엇을 뜻하는가?
 - 2진수 숫자라면 50121이고, 8비트의 문자(ASCII code) 2개 라면 CI
 - 4개의 4비트 코드라면, 9096₃초과코드

Figure 1.1
Transfer of information among registers.

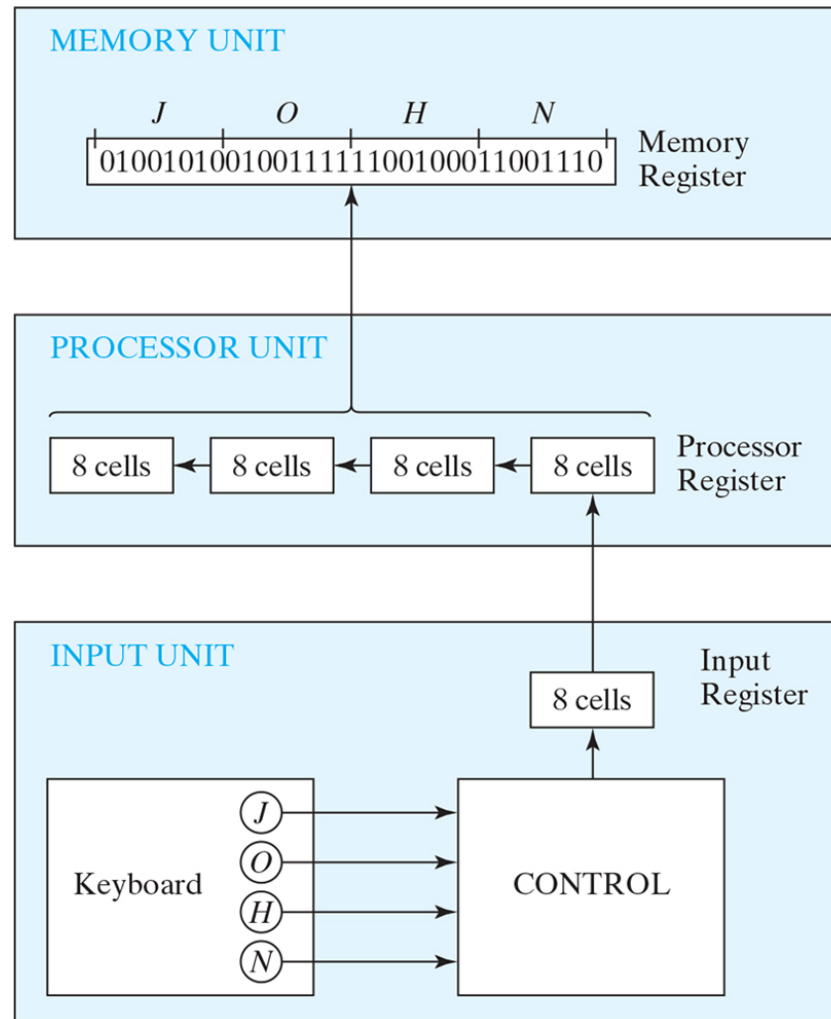
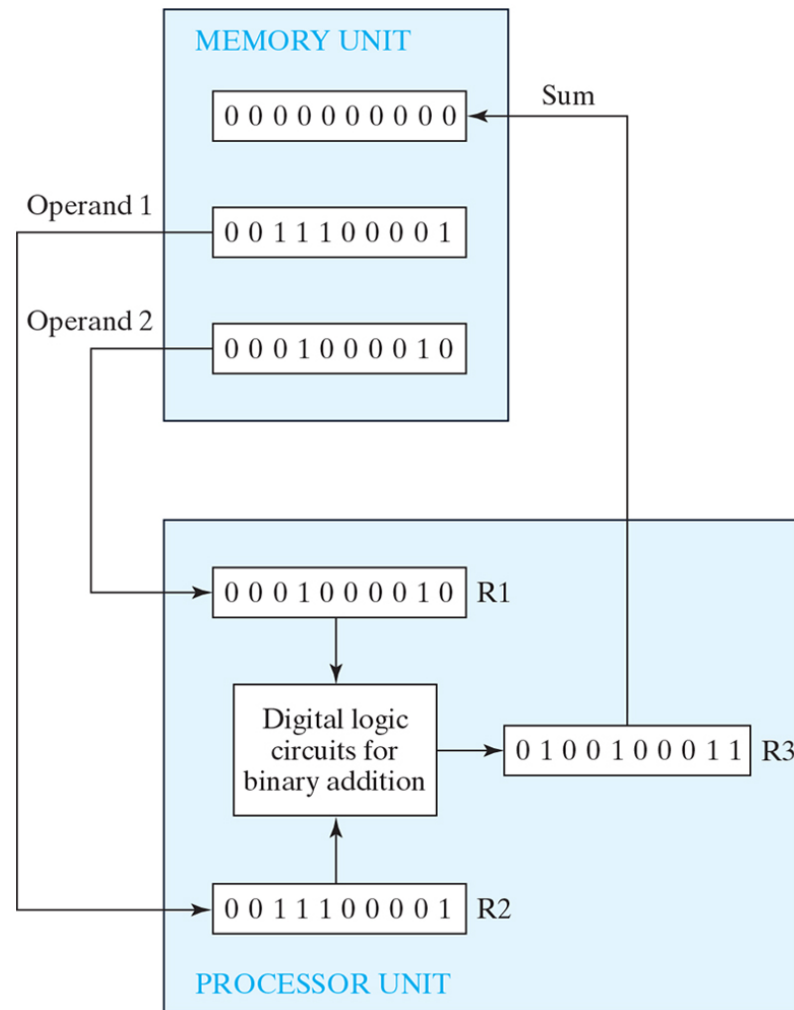


Figure 1.2
Example of registers in binary information processing.



1.9 2진 논리

- 2진 논리: 2진 변수와 논리 연산으로 구성
 - 변수는 A, B, C, x, y, z 로 표기하고 각 변수는 0 혹은 1을 가짐
 - 3개의 기초 논리 연산은 AND, OR, NOT
 - 주의: 2진수 연산과 다름
 - (2진수 연산) $1 + 1 = 10$ 이지만, (2진 논리) $1 + 1 = 1$ (+는 OR)

Table 1.8
Truth Tables of Logical Operations

AND			OR			NOT	
x	y	$x \cdot y$	x	y	$x + y$	x	x'
0	0	0	0	0	0	0	1
0	1	0	0	1	1	1	0
1	0	0	1	0	1		
1	1	1	1	1	1		

1.9 2진 논리

- 논리 게이트: 전압과 같은 전기적 신호로 0과 1을 표현하고, 이를 논리 연산(AND, OR, NOT 등) 한 후에 다시 전기적 신호로 출력하는 회로.
- 0과 1을 나타내는 전압값이 있어야 함. 보통 높은 전압을 1로 낮은 전압을 0으로 표현함.

Figure 1.3
Signal levels for binary logic values.

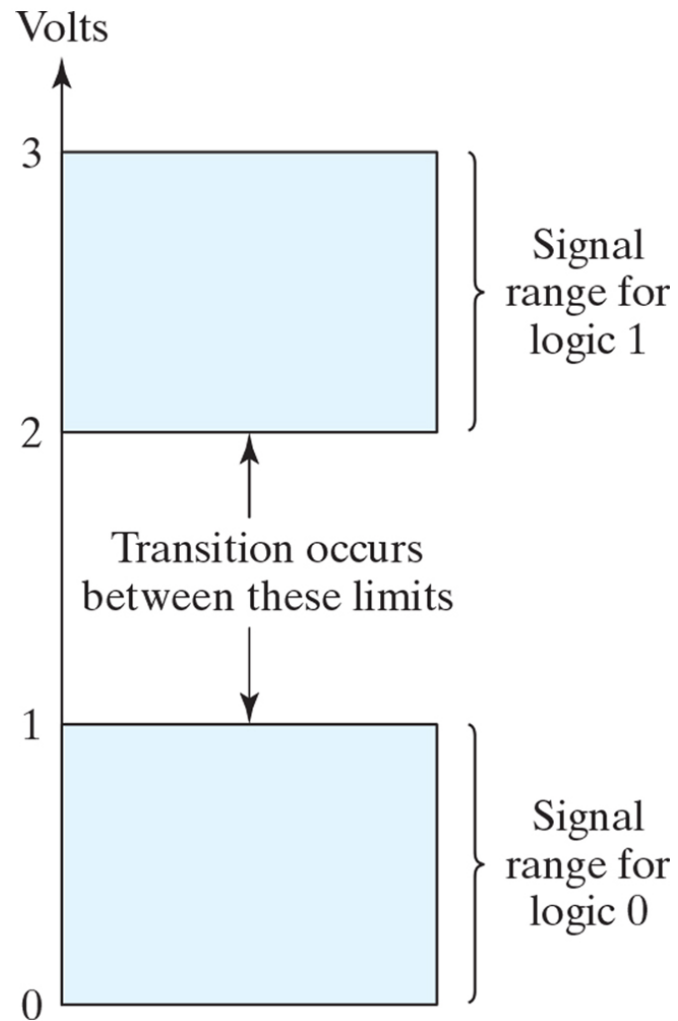
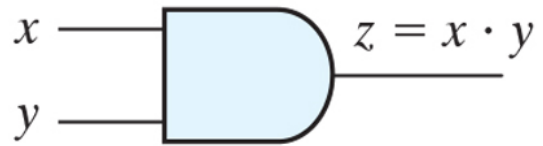
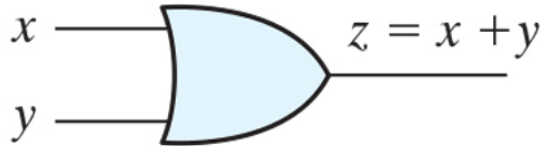


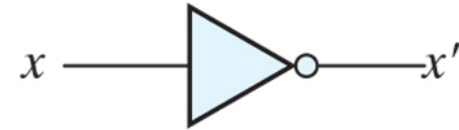
Figure 1.4
Symbols for digital logic circuits.



(a) Two-input AND gate



(b) Two-input OR gate



(c) NOT gate or inverter

Figure 1.5
Input–output signals for gates.

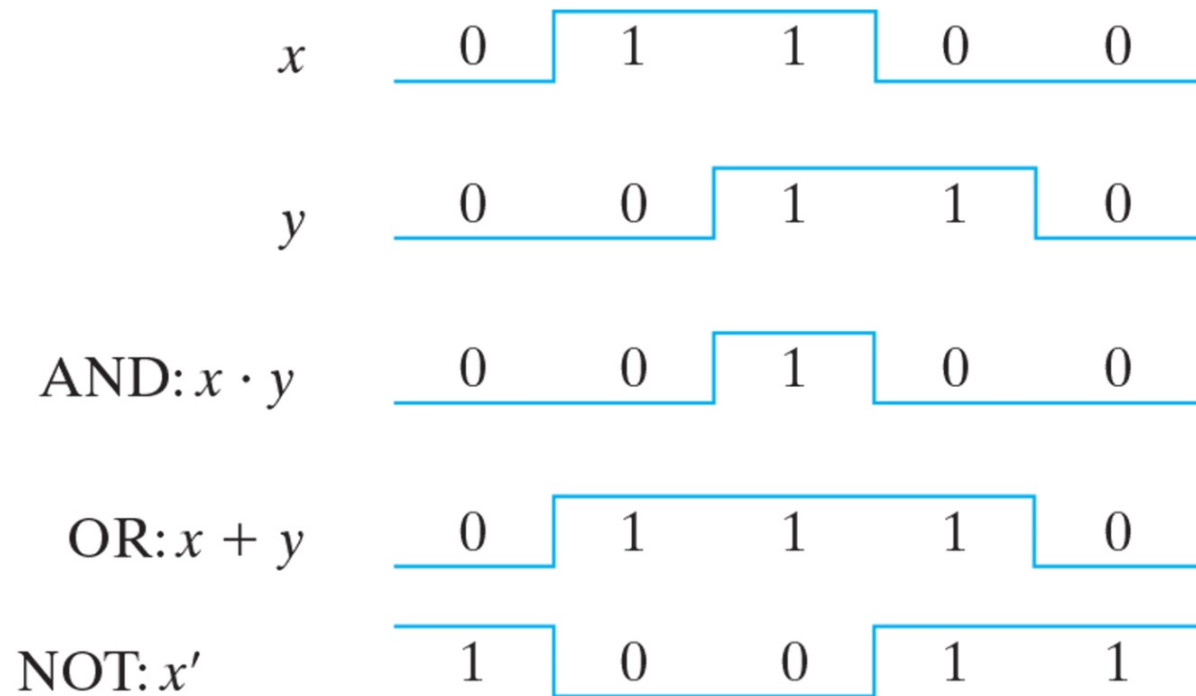
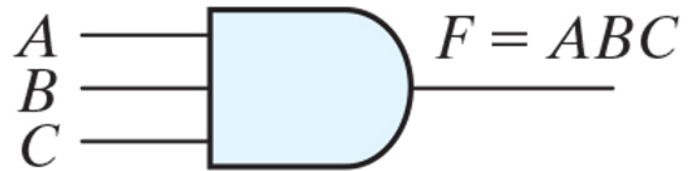
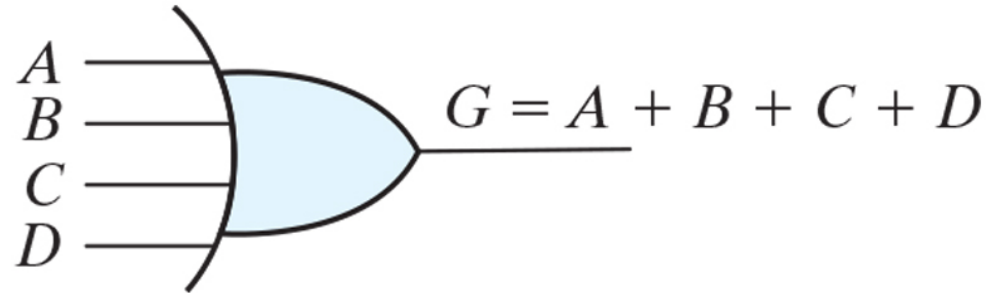


Figure 1.6
Gates with multiple inputs.



(a) Three-input AND gate



(b) Four-input OR gate