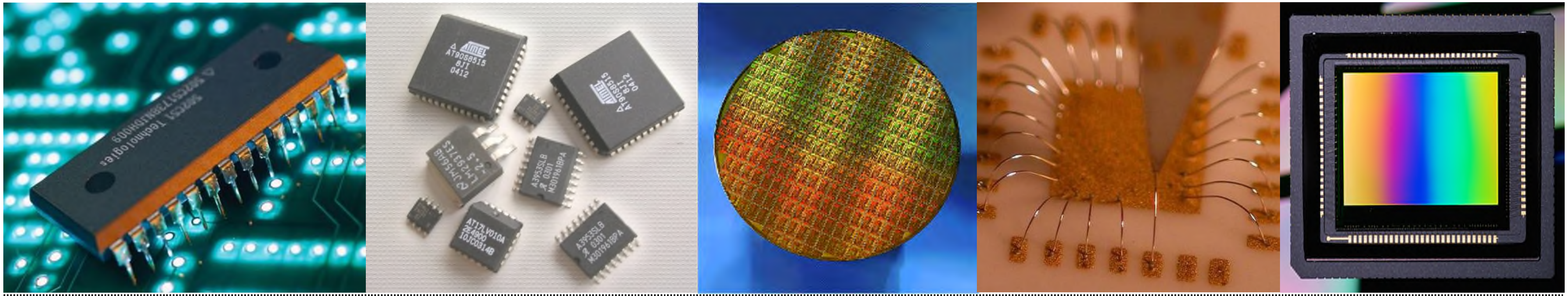


1. Digital System &Binary Numbers

1.1 Digital System

- Digital System
  - Communication, Business transaction, Traffic control, other commercial, industrial, scientific enterprises etc.
  - Discrete elements of information
- Signal
  - Discrete elements of information are represented in a digital system by physical quantities
- Binary Code
  - Discrete elements of information are represented with groups of bits
  - bit : binary + digit

1.2 Binary Number

- $a_5a_4a_3a_2a_1a_0.a_{-1}a_{-2}a_{-3}$
  $= a_nr^n+a_{n-1}r^{n-1}+...+a_2r^2+a_1r+a_0+a_{-1}r^{-1}+a_{-2}r^{-2}+...+a_{-m}r^{-m}$

$7392 = 7 \times 10^3 + 3 \times 10^2 + 9 \times 10^1 +2 \times 10^0$

$(11010.11)_2 =1\times2^4+ 1\times2^3+ 0\times2^2+ 1\times2^1+ 0\times2_0+ 1\times2_{-1}+ 1\times2_{-2}$

$= (26.75)_{10}$

$2^{10} = 1$Kilo

$2^{20} = 1$Mega

$2^{30} = 1$Giga

Table 1-1
Powers of Two

n	2^n	n	2^n	n	2^n
0	1	8	256	16	65,536
1	2	9	512	17	131,072
2	4	10	1,024	18	262,144
3	8	11	2,048	19	524,288
4	16	12	4,096	20	1,048,576
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

1.2 Binary Number

					오타, 1011
Augend	101101	Minuend:	101101	Multiplicand:	~~101~~
Addend	+100111	Subtrahend:	-100111	Multiplier:	*101
sum	1010100	Difference:	000110		1011
					0000
					1011
				Product:	10111

1.3 Number Base Conversions

- Ex 1-1) Convert decimal 41 to binary.

	Integer Quotient		Remainder	Coefficient
41/2 =	20	+	½	$a_0 = 1$
20/2 =	10	+	0	$a_1 = 0$
10/2 =	5	+	0	$a_2 = 0$
5/2 =	2	+	½	$a_3 = 1$
2/2 =	1	+	0	$a_4 = 0$
1/2 =	0	+	½	$a_5 = 1$

Integer	Remainder
41	
20	1
10	0
5	0
2	1
1	0
0	1

Answer

=101001

answer : $(41)_{10} = (a_5a_4a_3a_2a_1a_0)_2 = (101001)_2$

1.3 Number Base Conversions

- Ex 1-2) Convert decimal 153 to octal.

153	
19	1
2	3
0	2

$= (231)_8$

- Ex 1-3) Convert $(0.6875)_{10}$ to binary. **소수부부이 0이 될 때 까지 계속 2를 곱해주면 됨**

	Integer		Fraction	Coefficient
0.6875*2 =	1	+	0.3750	$a_{-1} = 1$
0.3750*2 =	0	+	0.7500	$a_{-2} = 0$
0.7500*2 =	1	+	0.5000	$a_{-3} = 1$
0.5000*2 =	1	+	0.0000	$a_{-4} = 1$

Answer: $(0.6875)_{10} = (0.a_{-1}a_{-2}a_{-3}a_{-4})_2 = (0.1011)_2$

1.4 Octal and Hexadecimal Numbers

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

$$(\underset{2}{\underline{10}}\ \underset{6}{\underline{110}}\ \underset{1}{\underline{001}}\ \underset{5}{\underline{101}}\ \underset{3}{\underline{011}}\ .\ \underset{7}{\underline{111}}\ \underset{4}{\underline{100}}\ \underset{0}{\underline{000}}\ \underset{6}{\underline{110}})_2 = (26153.7460)_8$$

$$(\underset{2}{\underline{10}}\ \underset{C}{\underline{1100}}\ \underset{6}{\underline{0110}}\ \underset{B}{\underline{1011}}\ .\ \underset{F}{\underline{1111}}\ \underset{2}{\underline{0010}})_2 = (2C6B.F2)_{16}$$

1.5 Complements – Diminished Radix Complement

- (r-1)'s complements of N is $(r^n-1)-N$

- r=10, r-1=9, 9'complements of N is $(10^n-1)-N$

  Ex) the 9's complements of 546700 is 999999-546700 = 453299

  the 9's complements of 012398 is 999999-012398 = 987601

- For binary number, r=2, r-1=1

  1'complements of N is $(2^n-1)-N$

  Ex) the 1's complements of 1011000 is 0100111

  the 1's complements of 0101101 is 1010010

1.5 Complements – Radix Complement

- The r's complements of an n-digit number N is r^n-N, N≠0

  0, N=0

- $r^n - N = [(r^n - 1) - N] + 1$

  → The r's complements is obtained by adding 1 to the (r-1)'s complements

- Ex) The 10's complements of 012398 is 987602

  The 10's complements of 246700 is 753300

  The 2's complements of 1101100 is 0010100

  The 2's complements of 0110111 is 1001001

PEARSON Education Canada

- Ex1-5) using 10's complement, subtract 72532-3250.

M =	72532
10's complement of N =	+ 96750
Sum =	169282
Discard end carry 10^5 =	-100000
Answer =	69282

- Ex1-6) Using 10's complement, subtract 3250-72532.

M =	03250
10's complement of N =	+27468
Sum =	30718

There is no end carry

Therefore, the answer is –(10's complement of 30718)=-69282

- Ex1-7) X=1010100, Y=1000011, (a) X-Y, (b) Y-X

(a) X−Y

X =	1010100
2's complement of Y =	+0111101
Sum =	10010001
Discard end carry 2^7 =	-10000000
Answer: X-Y =	0010001

(b) Y−X

Y =	1000011
2's complement of X =	+0101100
Sum =	1101111

There is no carry.

The answer is Y−X = −(2's complement of 1101111)=−0010001

- Ex1-8) Repeat Example 1-7 using 1'complement.

(a) X−Y = 1010100−10000011

X =	1010100
1's complement of Y =	+0111100
Sum =	10010000
End-around carry =	+ 1
Answer: X-Y =	0010001

(b) Y−X = 10000011−1010100

Y =	1000011
1's complement of X =	+0101011
Sum =	1101110

There is no carry.

The answer is Y−X = −(1's complement of 1101110)=−0010001

1.6 Signed Binary Numbers

- Ex) The number 9 represented in binary with eight bit

  +9 : 00001001

  -9 : 10001001 (signed-magnitude representation)

  11110110 (signed-1's-complement representation)

  11110111 (signed-2's-complement representation)

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 Signed Binary Numbers

- Arithmetic Addition

  - signed-magnitude system follows the rules of ordinary arithmetic.
  - signed-complement system requires only addition.

+6	00000110	-6	11111010
+13	00001101	+13	00001101
+19	00010011	+7	00000111

+6	00000110	-6	11111010
-13	11110011	-13	11110011
-7	11111001	-19	11101101

- Arithmetic Subtraction

$$(\pm A)-(+B) = (\pm A)+(-B)$$
$$(\pm A)-(-B) = (\pm A)+(+B)$$

1.7 Binary Code-BCD code

2진수를 10진수처럼 사용하기 위해서

- the 4-bit code for one decimal

$$(185)_{10} = (0001\ 1000\ 0101)_{BCD} = (10111001)_2$$

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

- BCD Addition

4	0100	4	0100	8	1000
+5	+0101	+8	+1000	+9	+1001
9	1001	12	1100	17	10001
			+0110		+0110
			10010		10111

- if the binary sum is greater or equal to 1010, we add 0110 to obtain the correct BCD

1.7 Binary Code-Other Decimal Codes

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	0 0 0 0
1	0001	0001	0100	0 1 1 1
2	0010	0010	0101	0 1 1 0
3	0011	0011	0110	0 1 0 1
4	0100	0100	0111	0 1 0 0
5	0101	1011	1000	1 0 1 1
6	0110	1100	1001	1 0 1 0
7	0111	1101	1010	1 0 0 1
8	1000	1110	1011	1 0 0 0
9	1001	1111	1100	1 1 1 1
	1010	0101	0000	0 0 0 1
Unused	1011	0110	0001	0 0 1 0
bit	1100	0111	0010	0 0 1 1
combi-	1101	1000	1101	1 1 0 0
nations	1110	1001	1110	1 1 0 1
	1111	1010	1111	1 1 1 0

1.7 Binary Code-Gray Code

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

그레이코드는 연속한 숫자 사이에 비트가 단 하나만 다르다.
-> 오류를 더 쉽게 발견하거나 최적화를 위해서 사용

1.7 Binary Code- ASCII Character Code

Table 1-7
American Standard Code for Information Interchange (ASCII)

	$b_7b_6b_5$							
$b_4b_3b_2b_1$	**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

- Error-Detecting Code

데이터 전송에 있어서 비트를 하나를 더 붙여서 1의 갯수를 체크하여 데이터 전송에 오류가 있었는지 확인하는(Error-Detecting) 방식

	With even parity **1의 갯수가 짝수(even)**	With odd parity **1의 갯수가 홀수(odd)**
ASCII A = 1000001	01000001	11000001
ASCII T = 1010100	11010100	01010100

1.8 Binary Storage and Registers

- Registers – A register with n cells can store any discrete quantity of information that contains n bits.
- Register Transfer

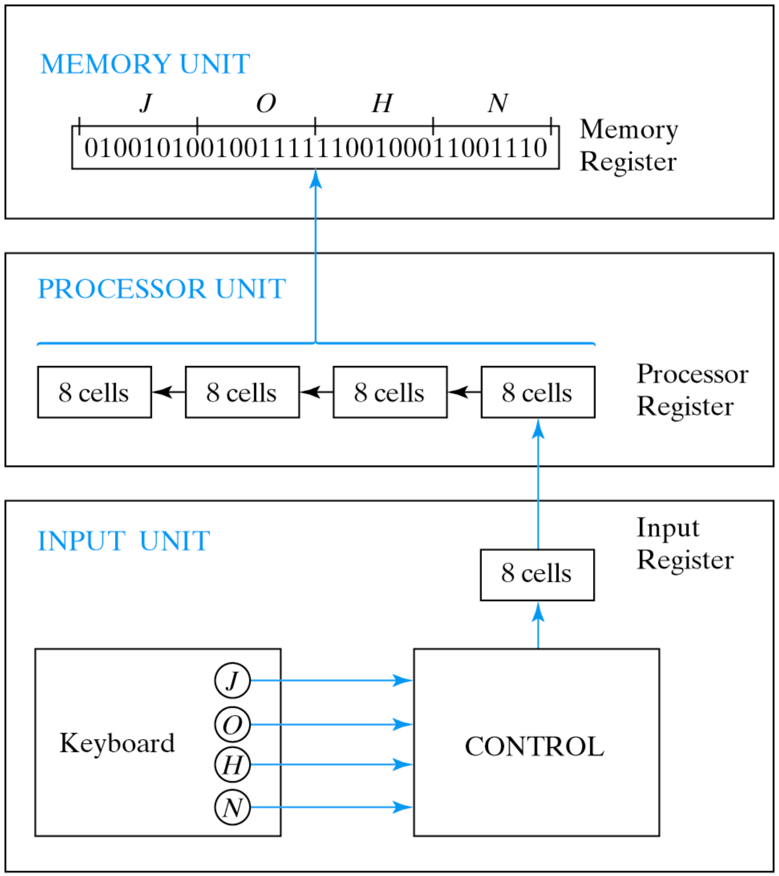


Fig. 1-1 Transfer of information with registers

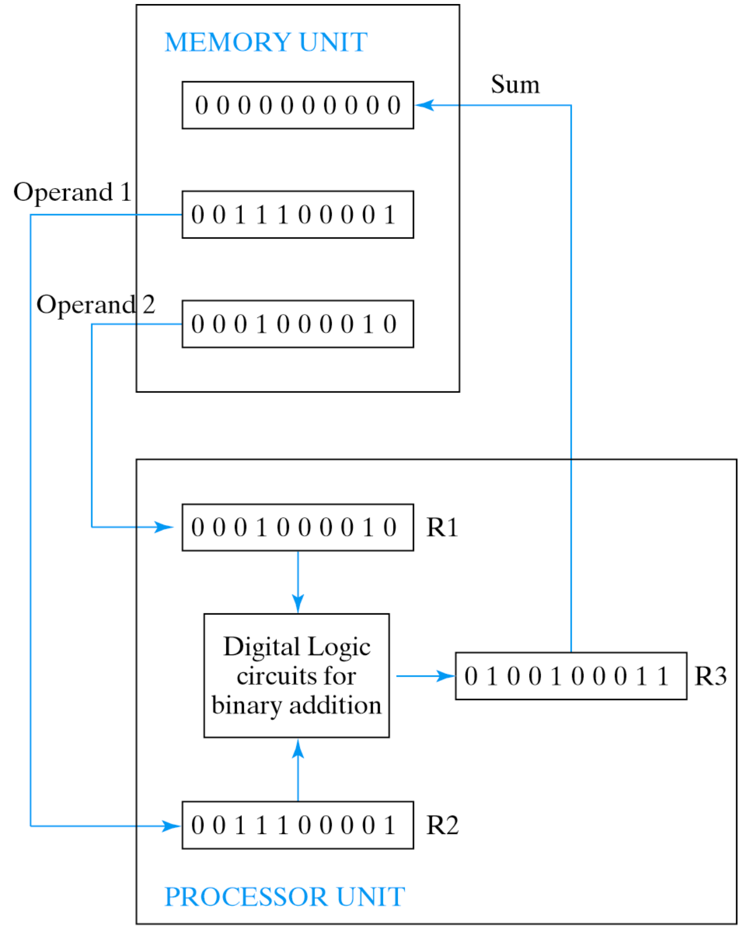


Fig. 1-2 Example of binary information processing

1.9 Binary Logic

- Definition of Binary Logic

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		

- Logic Gates

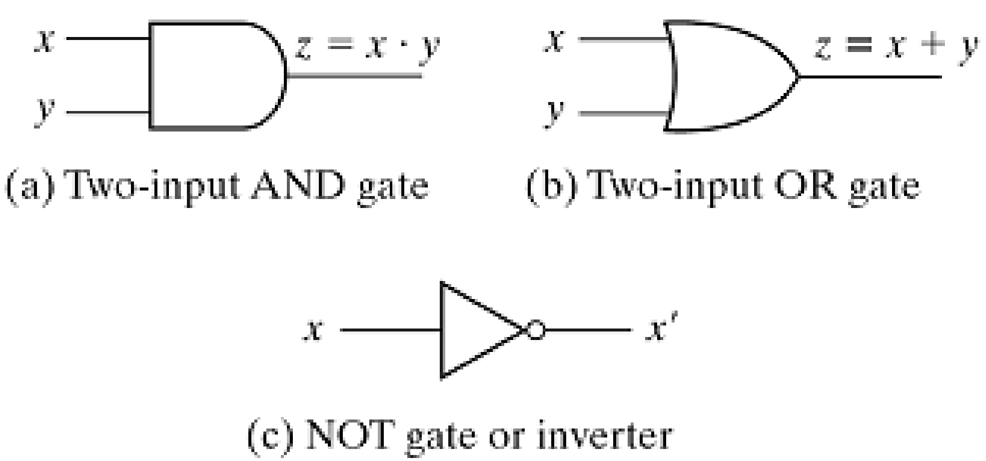


Fig. 1-4 Symbols for digital logic circuits

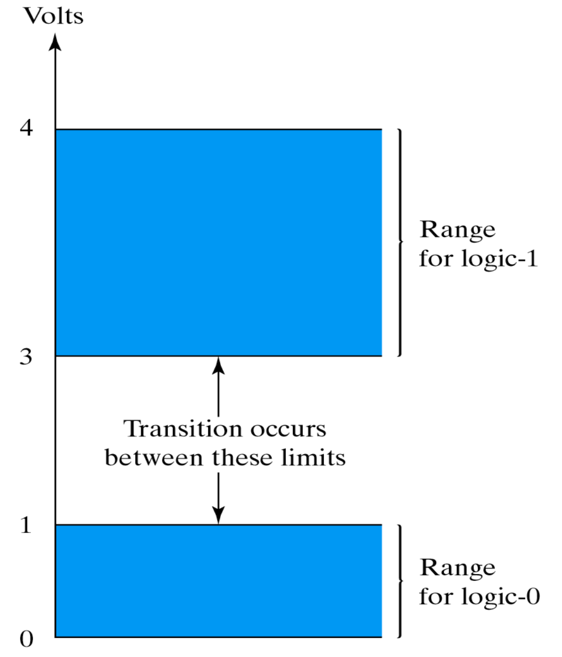


Fig. 1-3 Example of binary signals

1.9 Binary Logic

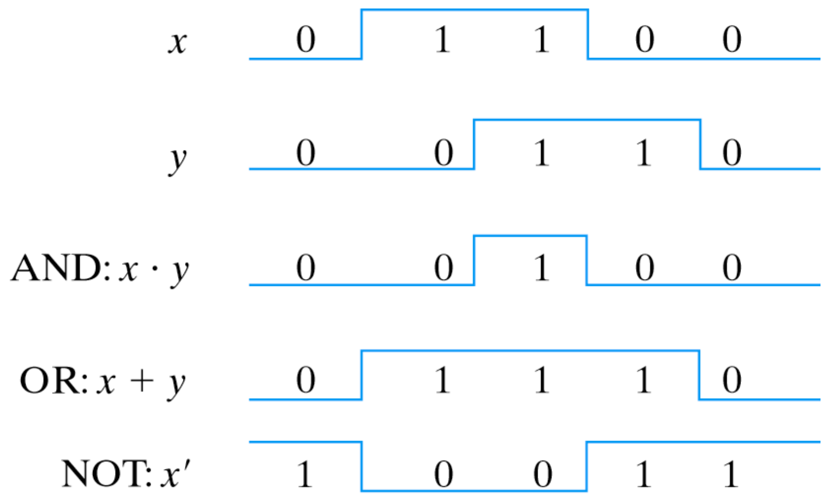
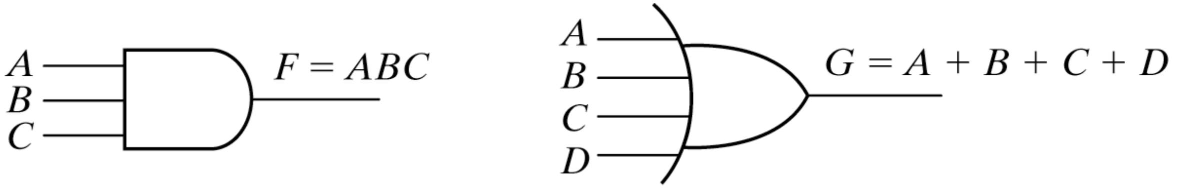


Fig. 1-5 Input-output signals for gates



(a) Three-input AND gate

(b) Four-input OR gate

Fig. 1-6 Gates with multiple inputs