🐱

Ch.1 Digital Systems and Binary Numbers

Digital systems

Digital vs Analog

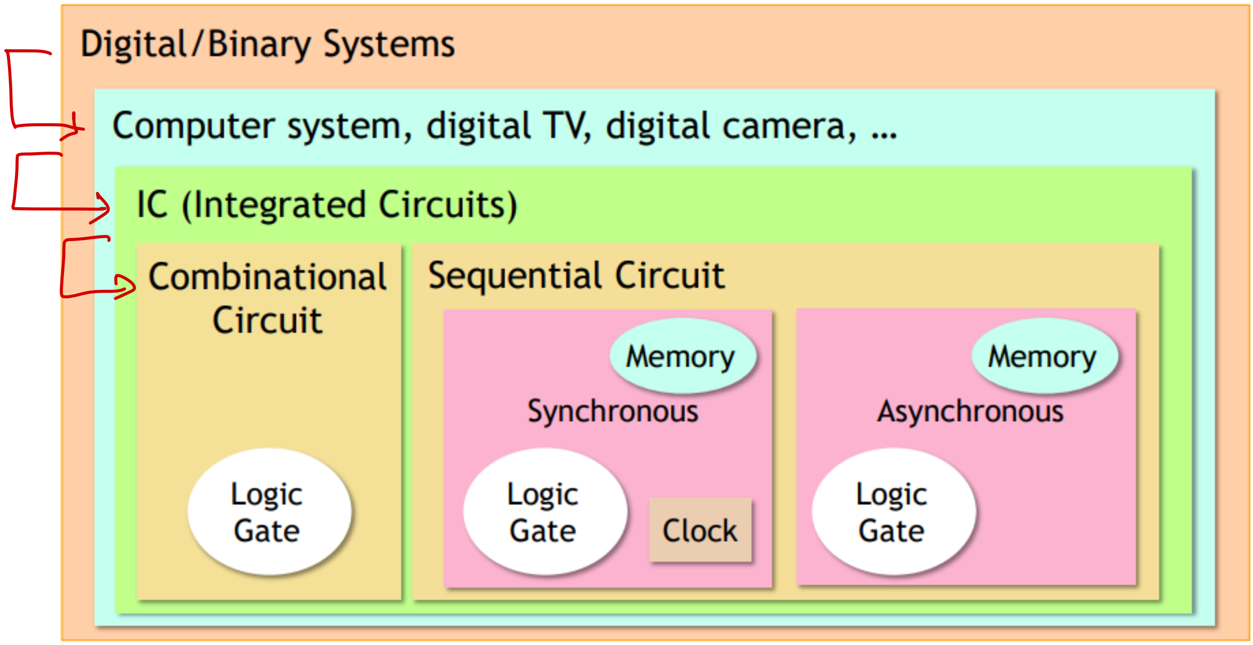
- **Digital** : 한정된 숫자로 이산적인 값을 나타냄
- **Analog** : 무한적인 값들을 셀 수 없는 숫자로 나타냄

Digits vs Bits

- **Digits** → **discreate**
    - Bits → binary digits
- **Digital Computer** → **digital system** → **bits!!**

↳ Binary System을 이해하는데 필요

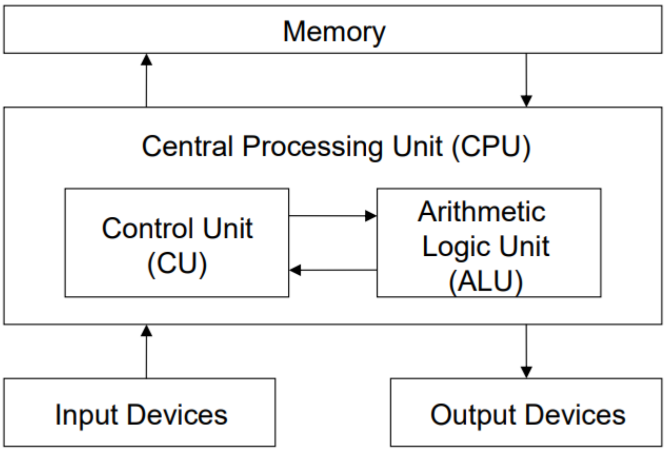
H/W in Digital Systems



Digital Computers

- **H/W** : electronic components & electromechanical devices
- **S/W** : programs composed of sequential instructions
  - System S/W → OS, compliers

- Basic Structure



Number Systems

- **Base(Radix value)** : 기수 ① / coefiicients ② : 계수
- Binary, Octal, Hexadecimal
- **Base-r**
  - Base : r
  - Set of coefficients = {0, 1, 2, … , r-1}

Ex Base : 10

Set of n : 0, 1 ... 9

$\rightarrow a_2a_1a_0 . a_{-1}a_{-2}$

$= r^2a_2 + r^1a_1 + r^0a_0 + r^{-1}a_{-1} + r^{-2}a_{-2}$

Binary Number

- **Bit** : a digit in a binary number
- **Byte**: 8bits

1kilo = 2^{10}

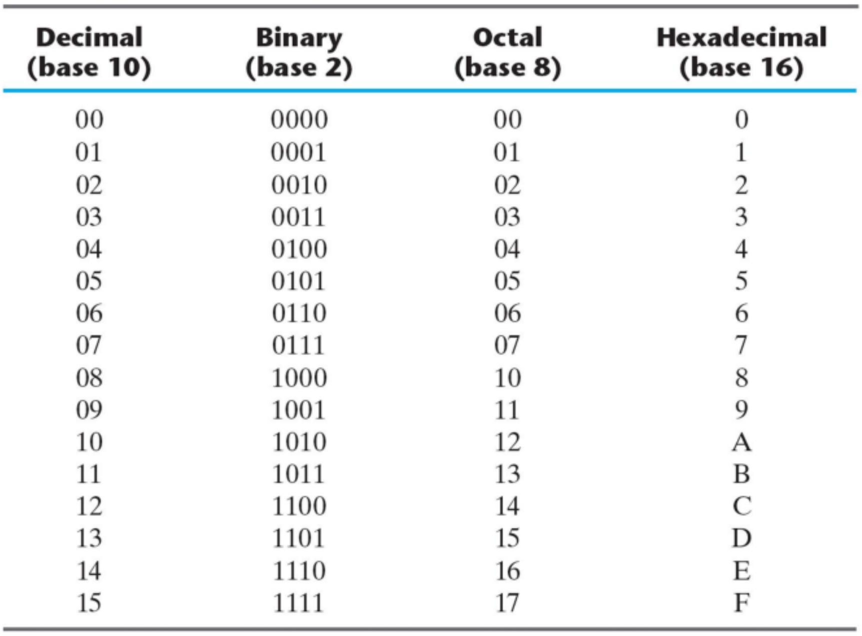
1Mega = 2^{20}

1Giga = 2^{30}

1Tera = 2^{40}

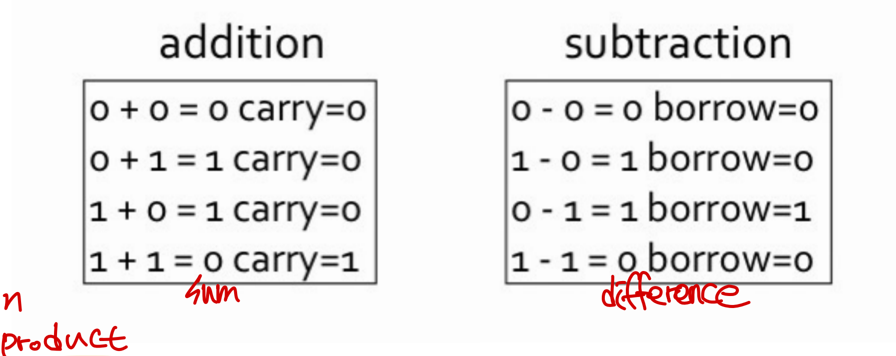
- **N bits** → from 0 to 2의 n승 -1

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

- **Arithmetic Operations(연산)** ← 10진수와 똑같음



addition	subtraction
0 + 0 = 0 carry=0	0 - 0 = 0 borrow=0
0 + 1 = 1 carry=0	1 - 0 = 1 borrow=0
1 + 0 = 1 carry=0	0 - 1 = 1 borrow=1
1 + 1 = 0 carry=1	1 - 1 = 0 borrow=0

sum

difference

① addition

② subtraction

③ multiplication
↳ product

- Number base conversion(소수부, 정수부)
  - intger ⇒ divisions 반복
  - fracion ⇒ multiplications 반복

12345 . 678
Integer Fraction

- MSB, LSB

2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0
1 0 0 0 1 0 0 1
MSB LSB

- Complements
  - (r-1)'s complement = (r의 n승-1) - N(Base)
  - r's complement = (r의 n승) - N(Base) = (r-1)'s complement + 1
    - 2의 보수 : 첫 1이 나오고 그 뒤로는 반대로 바꿀 것

X = 1010100
Y = 1000011
⇒ ① X + (Y의 2의 보수)
② ① − 2^n
③ Sum에 음수!

  - Subtraction(작은 거 - 큰 거) ex. 0 − 1 ?
    - M - N(M < N) = M + r의 n승 - N = M + 2의 보수 - r의 n승

ex. +7 : 0111

① ※ Signed magnitude: changing only the sign bit → 1111
② ※ Signed-1's complement: complementing all the bits including sign bit → 1000
③ ※ Signed-2's complement: taking 2's complement of all the bits including sign bit → 1001

  - Signed Binary Numbers
    - positive : 0
    - negative : 1
    - Arithmetic operation → carry out은 버림

→ H/W : leftmost position of the number ⇒ sign bit

↳ carry out이 sign-bit position일 때

Binary Codes(숫자로 된 기호를 할당한다고 생각할 것)

- Digital System
  - binary numbers 말고 다른 이산적인 요소의 정보를 가지고 있음
    - 이를 binary code로 표현

- **code assgining rules**
  - n-bit binary code → 2의 n승 distinct combinations(0과 1로)
  - 각각의 요소가 binary bit combination으로 나타내짐
  - 같은 값을 가지는 요소는 x → 모호성을 방지

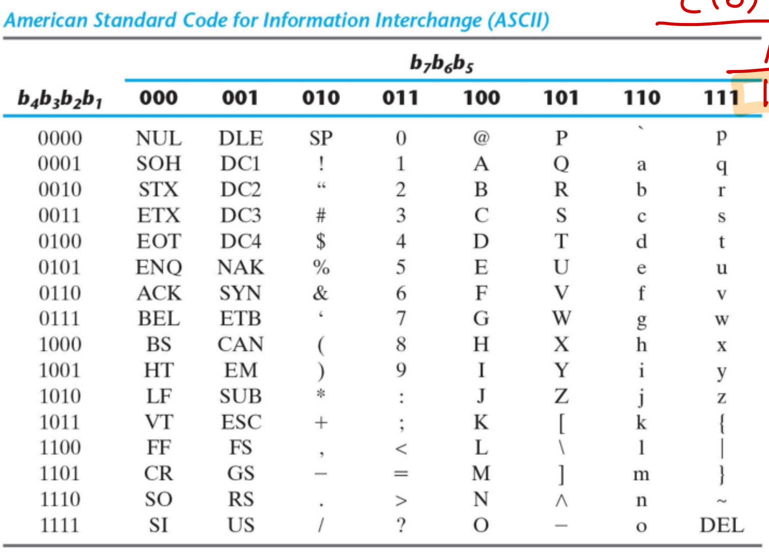
- **Binary Coded Decimal(BCD)**
  - 인간과 컴퓨터 간의 정보 교환을 위해선 전환이 필요 → 자연어를 BCD code로 처리
  - addition, decimal arithmetic → binary operation과 같음
  - 9를 초과하는 것은 meaning이 없다. ⇝ "6을 더해서 고쳐줌"

ex).

$$\begin{array}{rr} 4 & 0100 \\ +8 & 1000 \\ \hline +12 & 1100 \Rightarrow \text{mean x} \\ (+6) & 0110 \\ \hline 1 & 0010 \\ \boxed{1} & \boxed{2} \end{array}$$

EX

- **ASCII code**



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

① detect any odd combination errors.
② Cannot detect any even combination errors

- **Error-Detecting Code** : to detect the error in data communication or processing.
  - parity bit : the total number of 1's either even or odd ⇒ even/odd parity

    ↳ extra bit

Binary Storage

- **Binary cell** : a device that possesses two stable states(cpu)

- cell input / cell output

  receives data and control signals

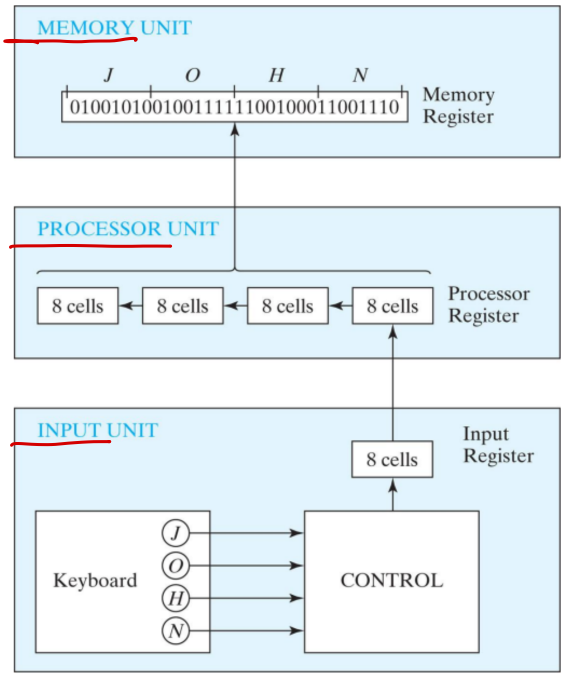
  physical quantity indicating which state the cell is in

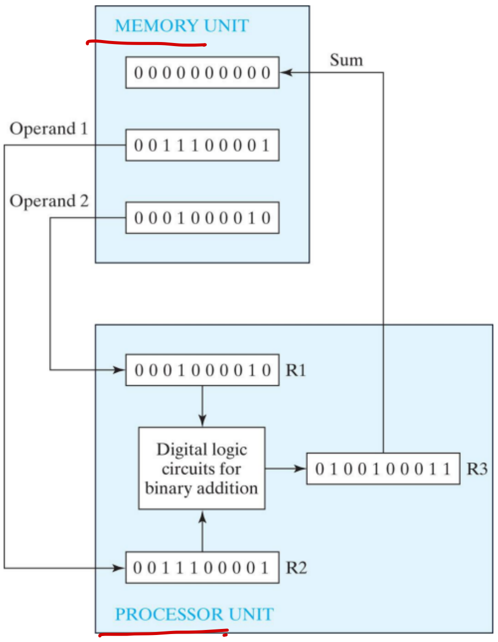
- **Registers**
  - groupt of n binary cells == an n-bit register
  - n-bit register has 2의 n승 states ⇒ all possible n-bit strings
  - register state ⇒ value, ASCII 해석될 수 있음
  - 직렬(serial), 병렬(parallel) I/O 가능

- **Register Transfer**
  - registers ar interconnected
    - 하나의 register의 content는 다른 것으로 transfer
    - content는 바뀔 수 있음
    - transforming circuit ⇒ data processing or data path element
  - 따라서 컴퓨터는 register와 digital logic circuit이 필요

    holding data

    manipulating binary variables





Binary logic

- deals with variables that have two values ⇒ {0,1}
  Ⓣ Ⓕ
- logic is based on Boolean Algebra operators
- variables denoted by letters of alphabet.

- **operations**
  - ① and → 논리곱 ⇒ $X \cdot Y = Z$ / $XY = Z$
  - ② or → 논리합 ⇒ $X + Y = Z$
  - ③ not → $X' = Z$
    $\overline{X} = Z$

NOT

x	z
0	1
1	0

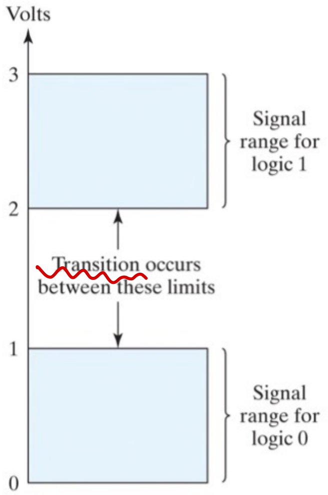
AND

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

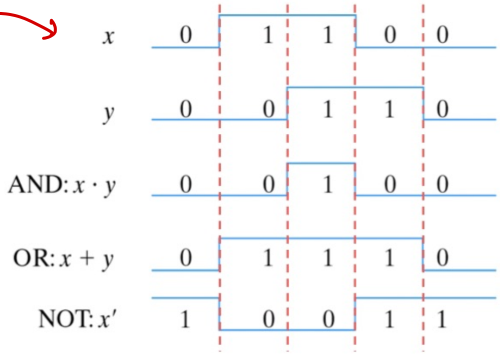
OR

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

- **logic gates**
  - eletronic circuits operate on 1 or more input signals to produce an output signal ⇒ 가장 기본적인 단위의 전자회로
  - ① signals ⇒ voltages(전압) or currents
    - as 0 or 1
    - defined by a range of voltage or current values



  - ② circuit diagrams use graphical symbols for logic gates
  - register간의 logic gate들의 연결이 data를 변형할 수 있음
  - Logic gates represent interconnections of transistors and other electronic components
  - ③ timing diagrams ⇒ input, ouput signal 표현
  - ④ multi-input logic gates ⇒ more than one input


x 0 1 1 0 0
y 0 0 1 1 0
AND: x · y 0 0 1 0 0
OR: x + y 0 1 1 1 0
NOT: x′ 1 0 0 1 1