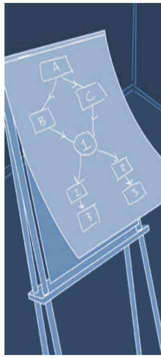
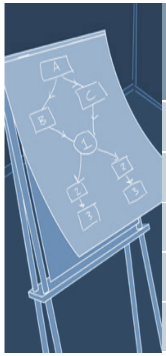
Chapter 1.
Digital Systems and Information

Mar., 2008



1. Information Representation
2. Number Systems
3. Arithmetic Operations
4. Decimal Codes
5. Alphanumeric Codes
6. Gray Codes

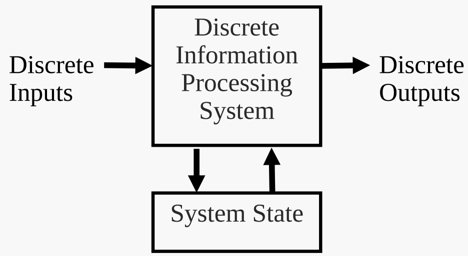


1. 정보 표현

2. Nuumber Systems
3. Arithmetic Operations
4. Decimal Codes
5. Alphanumeric Codes
6. Gray Codes

DIGITAL & COMPUTER SYSTEMS - Digital System

- Takes a set of discrete information inputs and discrete internal information (system state) and generates a set of discrete information outputs.



◈ Type of Digital System

- No state present
  - *Combinational Logic System*
  - Output = Function(Input)
- State present
  - State updated at discrete times → Synchronous *Sequential System*
  - State updated at any time → Asynchronous Sequential System
  - State = Function (State, Input)
  - Output = Function (State) or Function (State, Input)

Digital System Example:

A Digital Counter (e. g., odometer):

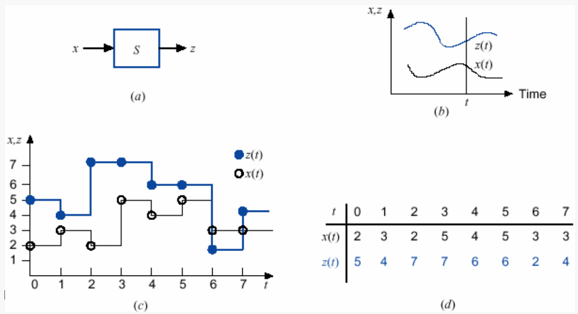
Count Up →
Reset →

Inputs: Count Up, Reset
Outputs: Visual Display
State: "Value" of stored digits

DIGITAL & COMPUTER SYSTEMS - Digital System

◈ What is Digital System ?

- **Digital system** is a system in which signals have finite number of discrete values (electric impulses, decimal digits, arithmetic operations etc.)
- **Analog system** is a system in which signals have infinite number of values (electoric voltage that vary with time)
- **Synchronous**; systems where signals may change only at discrete instants
- **Asynchronous**; systems where signals may change at any instant



[Wikipedia]

- A **digital** system uses discrete (that is, discontinuous) values to represent information for input, processing, transmission, storage, etc. By contrast, non-digital (or analog) systems use a continuous range of values to represent information. Although digital representations are discrete, the information represented can be either discrete, such as numbers, letters or icons, or continuous, such as sounds, images, and other measurements of continuous systems.
- 디지털은 자료를 연속적인 실수가 아닌, 특정한 최소 단위를 갖는 이산적인 수치를 이용하여 처리하는 방법을 의미
- Analog signal, a variable signal continuous in both time and amplitude.

Why are Digital Systems important ?

- It is well suited for numerical and non-numerical information processing
- Information processing can use a general-purpose system (computer)
- The finite number of values in a digital signals is represented by a vector of signas with just 2 values (binary signals)
- Digital signals are quite insensitive to variations of component variable values

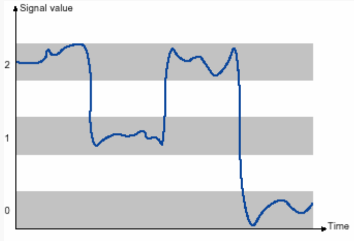


Figure 1.2: Separation of digital signal values.

- Numerical digital systems can be made more accurate by increasing the number of digits used in the representation.
- Complex digital systems are built as integrated circuits composed of a large number of very simple devices.
- It is possible to select among different implementations of systems that trade off speed and amount of hardware.

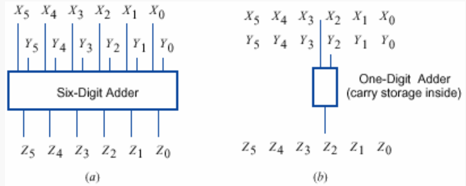


Figure 1.3: Six-digit adder: a) Parallel implementation. b) Serial implementation.

When are Digital Systems used ?

- Digital representation and processing methods widely used
- Extraordinary progress in digital technology and use Indispensable in modern society
- New applications fueled by the development of computer technology
- Knowledge about the design and use of digital systems required in a large variety of human activities

INFORMATION REPRESENTATION - Signals

Digital System;

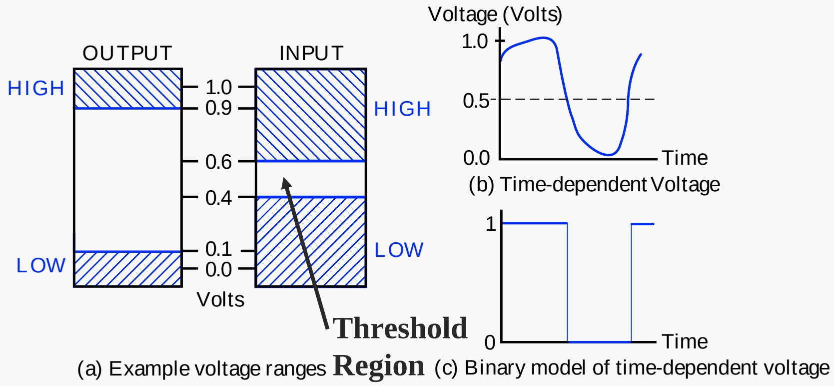
- Stores, moves and processes **information**

Information;

- Represents broad range of **phenomena**
- From physical world : characterized by parameters
  - *Analog Signal* : 시간상에서 연속적으로 변하는 물리량 신호
  - Such as weight(중력,무게), temperature(온도), pressure(압력), velocity(속도), frequency(빈도)
  - **Continuous**
- From man-made world : parameters
  - *Digital Signal* : 이산적(discrete)인 정수 체계로 표현되는 시간적으로 불연속적인 신호
  - Such as business records using words
    - Quantities and currencies taking on values from alphabet
    - Integers or units of currency
  - **Discrete** in nature

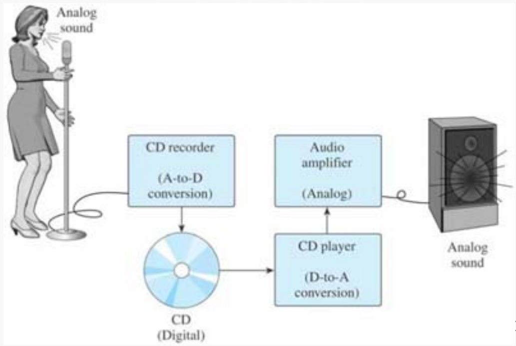
- For digital systems, the information variables take on discrete values.
- Two level, or *binary values* are the most prevalent values in digital systems.
- Binary values are represented abstractly by:
  - digits 0 and 1
  - words (symbols) False (F) and True (T)
  - words (symbols) Low (L) and High (H)
  - words ON and OFF.

Signal Example – Physical Quantity: Voltage

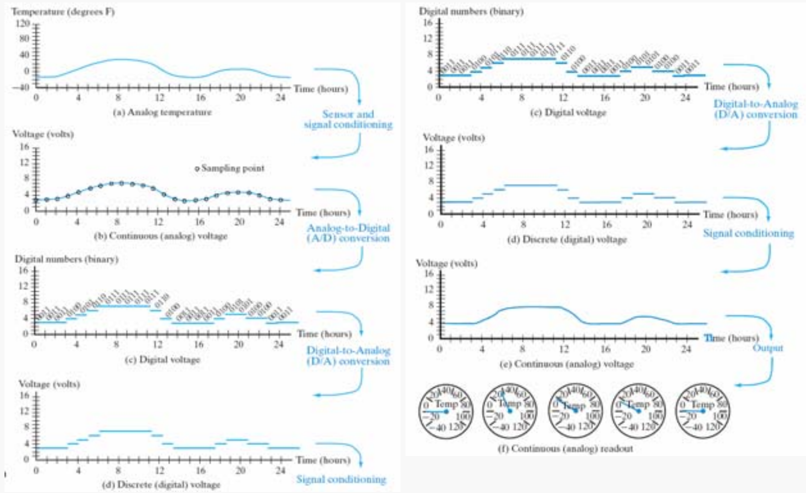


- The reason why binary is used
  - Assume a system with 10 value
    - If overall voltage range is from 0 to 5V
    - each stage will be less than 0.5V
    - consider noise margin, it will be less than 0.25V
- This would require complex and costly electronic circuit
  - As a consequence, use of the multivalued circuit is limited
  - Instead, **binary circuit is simple, easy to design, reliable**

- A binary digit is called a *bit*
  - Information is represented in digital computers by groups of bits
- By using various techniques, groups of bits can be made to represent not only binary number, but also other groups of discrete symbols
- Groups of bits can even specify to the computer the instruction to be executed and data to be processed
- Natural signal → analog-to-digital converter → discrete quantities of information



- Example 1-1. Temperature Measurement and Display



1. temperature values over one 24-hour : plotted in Fig. 1-4(a)
   - measured by sensor consisting of a thermistor(temp→resistance)
     - provided analog voltage that is proportional to temperature

2. using signal conditioning this voltage :
   - changed continuous voltage ranging 0~15V : Fig. 1-4(b)

3. analog voltage : sampled at a rate of once per hour
   - a very slow sampling rate for illustration : dotted in Fig. 1-4(b)
   - each value sampled : applied to an analog-to-digital(A/D) converter
     - replaced with digital no. written in binary(4 bit) : Fig. 1-4(c)
       - equivalent : decimal values 0~15
   - value of temperature : in process of conversion
     - quantized from an finite no. of values to just 16 values
   - analog temperature -25 ~ -15 : represented by digital temp. -20
   - quantization error :
     - this discrepancy between actual temp. and digital temp.
     - to obtain greater precision : to increase no. of bits beyond four

4. digital value : pass through microcomputer
   - → wireless transmitter (digital output device) : outdoor site

5. digital value : transmitted
   - → wireless receiver (digital input device) : indoor base station
   - → microcomputer : calculation performed
     - to adjust its value based on thermistor properties
     - → displayed with an analog meter

6. digital value : converted to analog value
   - by digital-to-analog(D/A) converter
     - giving the quantized, discrete voltage level : Fig. 1-4(d)

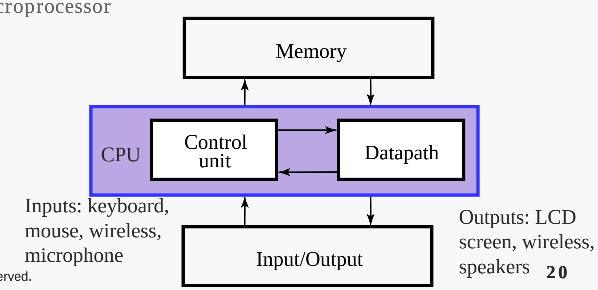
7. continuous signal : using signal conditioning (Fig. 1-4e)
   - such as processing of the output by a low pass filter

8. applied to analog voltage display : Fig. 1-4(f)
   - labeled with the corresponding temperature values

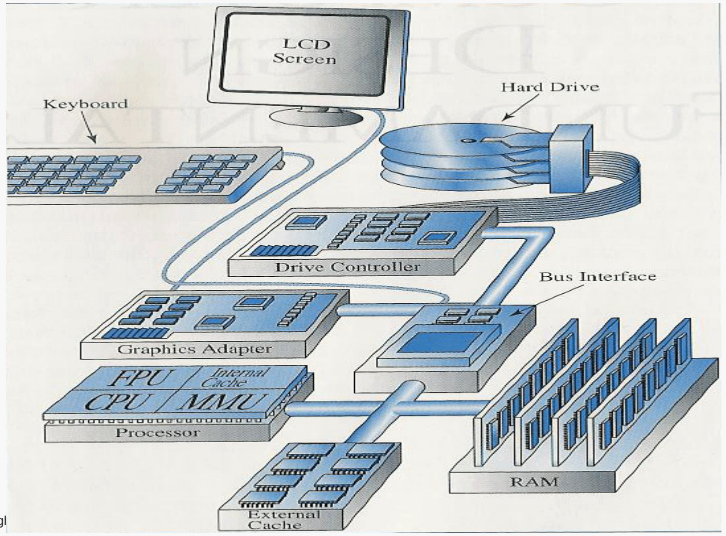
Digital Computer Example

Block Diagram of a Digital Computer

- **Memory :** stored program and data (input, output, intermediate data)
- **Datapath :** performed arithmetic & other data-processing operations (as specified by program)
- **Control unit :**
  - supervised flow of information
  - retrieves instructions
- **CPU (Central Processing Unit) :** 메모리에 저장된 프로그램으로부터 명령어를 하나씩 읽어서, 명령어에 대응하는 동작을 실행하는 Datapath를 제어
  - processor microprocessor



⟐ More on the Generic Computer



⟐ Processor(IC) : heart of the computer

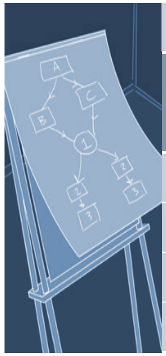
- four functional modules :
  - CPU, FPU, MMU, internal cache
- CPU(central processing unit) :
  - contained datapath, control unit
  - performed fixed-point operations
    - fixed-point number : integer number
- FPU(floating-point unit) :
  - contained datapath, control unit like CPU
  - performed floating-point operations
    - floating-point number : real number
  - handled very large and very small numbers

- MMU(memory management unit) :
  - memory hierarchy : cache - RAM - hard disk
  - cache : allow CPU, FPU get at data to be processed
    - much faster than with RAM alone
  - cause the memory that appears to be available to CPU
    - to be much, much larger than actual size of RAM

⟐ Processor bus, I/O bus :

- attached to the bus interface to carry data having
  - different numbers of bits, different speeds,
  - different ways of controlling the movement of data
- bus interface hardware : handles these differences
  - so that data can be communicated between two buses

- ⟐ **Ch. 1~6 : Logic Design of Digital Circuits in general**
- ⟐ **Ch. 5~7 : Primary Components of Digital System,**
  - their operation, and their design
- ⟐ **Ch. 8 : Operational Characteristics of RAM**
- ⟐ **Ch. 9 : Datapath and Control for Simple Computer**
- ⟐ **Ch. 10~13 : Basics of Computer Design**
  - Ch. 10 : typical instructions
    - employed in computer instruction set architecture
  - Ch. 11 : architecture and design of CPUs
  - Ch. 12 : I/O devices and communication with them
  - Ch. 13 : memory hierarchy concepts related to cache & MMU



1. Information Representation

2. Number Systems

3. Arithmetic Operations
4. Decimal Codes
5. Alphanumeric Codes
6. Gray Codes

Number System

- Decimal
- Binary
- Octal
- Hexadecimal

- Arithmetic Operations
- Conversion
  - Decimal-to-Binary, Decimal-to-Octal
  - Binary-to-Octal, Binary-to-Hexadecimal

Decimal Numbering System

- 10개의 digit 사용 (r digit = 0~9)
- 예) 1473

$$1473 = 1\times10^3 + 4\times10^2 + 7\times10^1 + 3\times10^0$$
$$N_{10} = d_3\times10^3 + d_2\times10^2 + d_1\times10^1 + d_0\times10^0$$

- Weighting (가중치)
- Radix or Base (진수의 기수)
- MSD (Most Significant Digit) : 가장 큰 자리 값
- LSD (Least Significant Digit) : 가장 작은 자리 값

Binary Numbering System

- 2개 (0,1)의 digit 사용
- 예) 110101_2

$$\begin{aligned}110101_2 &= 1\times2^5 + 1\times2^4 + 0\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0\\ &= 32+16+0+4+0+1\\ &= 53_{10}\end{aligned}$$

- Radix or Base = 2
- Bit (Binary Digit) : 2진수의 각 자리 수
- 2진수의 자리 값 (뒤 테이블 참조)

$$\begin{aligned}1010.01_2 &= 1\times2^3 + 0\times2^2 + 1\times2^1 + 0\times2^0 + 0\times2^{-1} + 1\times2^{-2}\\ &= 1\times8 + 0\times4 + 1\times2 + 0\times1 + 0\times0.5 + 1\times0.25\\ &= 10.25_2\end{aligned}$$

The conversion of a decimal number to binary : 625

$625 - 512 = 113 = N_1 \qquad 512 = 2^9$

$113 - 64 = 49 = N_2 \qquad 64 = 2^6$

$49 - 32 = 17 = N_3 \qquad 32 = 2^5$

$17 - 16 = 1 = N_4 \qquad 16 = 2^4$

$1 - 1 = 0 = N_5 \qquad 1 = 2^0$

$(625)_{10} = 2^9 + 2^6 + 2^5 + 2^4 + 2^0 = (1001110001)_2$

- K (kilo) : $2^{10} = 1{,}024$
- M (Mega) : $2^{20} = 1{,}048{,}576$
- G (Giga) : $2^{30} = 1{,}073{,}741{,}824$

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

Table 1-1 Powers of Two

Octal Numbering System

- 8개 (0~7) digit 사용
- 예) 1367.4_8

$$\begin{aligned} 1367.4_8 &= 1\times 8^3 + 3\times 8^2 + 6\times 8^1 + 7\times 8^0 + 4\times 8^{-1} \\ &= 512 + 192 + 48 + 7 + 0.5 \\ &= 759.5_{10} \end{aligned}$$

Hexadecimal Numbering System

- 16개 (0~9, A~F) digit 사용
- 예) $9FB.5_{16}$

$$\begin{aligned} 9FB.5_{16} &= 9\times 16^2 + 15\times 16^1 + 11\times 16^0 + 5\times 16^{-1} \\ &= 2304 + 240 + 11 + 0.3125 \\ &= 2555.3125_{10} \end{aligned}$$

- Base-8 and base-16 are useful for representing binary quantities indirectly because they posses the property that their bases are powers of two
  - Each octal digit : three binary digits
  - Each hexadecimal digit : four binary digits
  - The length is shorter than corresponding binary number
- Conversion from binary to octal(hexadecimal) :
  - divided into groups of three(four) bits, starting from binary point
  - $(010\ 110\ 001\ 101\ 011.111\ 100\ 000\ 110)_2 = (26153.7406)_8$
  - $(0010\ 1100\ 0110\ 1011.1111\ 0000\ 0110)_2 = (2C6B.F06)_{16}$
- Conversion from octal(hexadecimal) to binary : reverse
  - $(673.12)_8 = (110\ 111\ 011.001\ 010)_2$
  - $(3A6.C)_{16} = (0011\ 1010\ 0110.1100)_2$

A number in base *r* contains *r* digits and is expressed as a power series in *r*

$$\begin{aligned} &(A_n A_{n-1} \cdots A_1 A_0 . A_{-1} A_{-2} \cdots A_{-m+1} A_{-m})_r \\ &= A_n \times r^n + A_{n-1} \times r^{n-1} + \cdots + A_1 \times r^1 + A_0 \times r^0 \\ &\quad + A_{-1} \times r^{-1} + A_{-2} \times r^{-2} + \cdots + A_{-m+1} \times r^{-m+1} + A_{-m} \times r^{-m} \end{aligned}$$

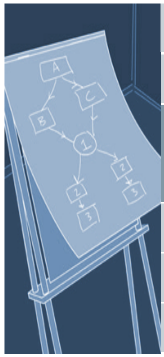
Numbering System

TABLE 1-3
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

Number Range

- The range of numbers that can be represented is based on the number of bits available in the hardware structures that store and process information
  - The number of bits is most frequently a power of two
  - For 16-bit unsigned integers, the number 537 is represented as 0000001000011001
  - The range of integer is from 0 to $2^{16} - 1$ (0 to 65,535)
  - The number 0.375 is .0110000000000000 if binary point is left of the most significant bit
  - The range of fraction is from 0 to $(2^{16} - 1) / 2^{16}$ or from 0.0 to 0.9999847412



1. Information Representation
2. Number Systems
3. Arithmetic Operation
4. Decimal Codes
5. Alphanumeric Codes
6. Gray Codes

1-3 Arithmetic Operations

Examples of the addition of two binary number

Carries(올림) :	00000	101100
Augend(피가산수) :	01100	10110
Addend(가산수):	10001	10111
Sum(합계) :	11101	101101

Examples of the subtraction of two binary number

Borrows(빌림) :	00000	00110		00110
Minuend(피감수) :	10110	10110	10011	11110
Subtrahend(감수) :	10010	10011	11110	10011
Difference(차이) :	00100	00011		-01011

Examples of the multiplication of two binary number

Multiplicand(피승수숫자) : 1011
Multiplier(승수숫자) : 101

```
   1011
    101
   ----
   1011
  0000
 1011
 ------
 110111
```


Hexadecimal Addition

Perform the addition $(59F)_{16} + (E46)_{16}$.

Hexadecimal	Equivalent Decimal Calculation				
	1 ←	Carry	1 ←		Carry
5 9 F	5		9	15	
E 4 6	14		4	6	
1 3 E 5	1 19 = 16 + 3		14 = E	21 = 16 + 5	

Octal Multiplication

Perform the multiplication $(762)_8 \times (45)_8$.

Octal	Octal		Decimal		Octal
7 6 2	5×2	=	$10 = 8 + 2$	=	12
4 5	$5 \times 6 + 1$	=	$31 = 24 + 7$	=	37
4 6 7 2	$5 \times 7 + 3$	=	$38 = 32 + 6$	=	46
3 7 1 0	4×2	=	$8 = 8 + 0$	=	10
4 3 7 7 2	$4 \times 6 + 1$	=	$25 = 24 + 1$	=	31
	$4 \times 7 + 3$	=	$31 = 24 + 7$	=	37

Conversion from Decimal to other Bases

Conversion of Decimal Integer to Octal

$153/8 = 19 + 1/8$	Remainder = 1 ↑	Least significant digit
$19/8 = 2 + 3/8$	= 3	
$2/8 = 0 + 2/8$	= 2	Most significant digit

$(153)_{10} = (231)_8$

Conversion of Decimal Integer to Binary

Convert decimal 41 to binary.

$41/2 = 20 + 1/2$	Remainder = 1 ↑	Least significant digit
$20/2 = 10$	= 0	
$10/2 = 5$	= 0	
$5/2 = 2 + 1/2$	= 1	
$2/2 = 1$	= 0	
$1/2 = 0 + 1/2$	= 1	Most significant digit

$(41)_{10} = (101001)_2$

Of course, the decimal number could be converted by the sum of powers of two:

$(41)_{10} = 32 + 8 + 1 = (101001)_2$

Conversion of Decimal Fraction to Binary

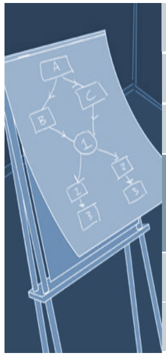
$0.6875 \times 2 = 1.3750$	Integer = 1 ↓	Most significant digit
$0.3750 \times 2 = 0.7500$	= 0	
$0.7500 \times 2 = 1.5000$	= 1	
$0.5000 \times 2 = 1.0000$	= 1	Least significant digit

$(0.6875)_{10} = (0.1011)_2$

Conversion Decimal Fraction to Octal

Convert decimal 0.513 to a three-digit octal fraction.

$0.513 \times 8 = 4.104$	Integer = 4 ↓	Most significant digit
$0.104 \times 8 = 0.832$	= 0	
$0.832 \times 8 = 6.656$	= 6	
$0.656 \times 8 = 5.248$	= 5	Least significant digit



What is Code ?

- In communications, a **code** is a rule for converting a piece of information (for example, a letter, word or phrase) into another form or representation, not necessarily of the same type.
- In communications and information processing, *encoding* is the process by which information from a source is converted into symbols to be communicated. *Decoding* is the reverse process, converting these code symbols back into information understandable by a receiver.
- 숫자, 문자, 기호 등을 다른 진수나 기호로 변환하여 나타내는 것
  - Inputs for Digital Systems = Number, Letters, Symbols
  - Digital Systems deal only 0 and 1.

1-4 Decimal Codes

Binary-Coded Decimal (BCD)

- Binary number system is the most natural system for a computer
- People are accustomed to the decimal system
- (human – decimal) : (computer – binary)
  - → Decimal number should be stored in computer with specific form
  - → **BCD**
- User enters number in decimal using keyboard
  → Decimal number is converted in BCD
  → BCD is converted in binary number for operation
  → Binary result is converted in BCD
  → BCD is converted in decimal number and display on CRT or LCD

TABLE 1-3
Binary-Coded Decimal (BCD)

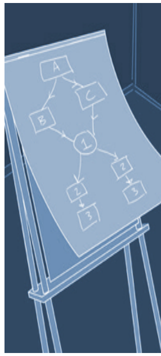
Decimal Symbol	BCD Digit
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

Table 1-3 Binary-Coded Decimal (BCD)

- $(396)_{10} = (\ 0011\ 1001\ 0110\ )_{BCD}$
- $(\ 0100\ 0011\ 1001\ 0111\ )_{BCD} = (\ 4397\ )_{10}$

- **BCD Addition**

110	BCD carry	1	1	
448		0100	0100	1000
+489		+0100	+1000	+1001
937	Binary sum	1001	1101	1 0001
	Add 6		+0110	+0110
	BCD sum		1 0011	1 0111
	BCD result	1001	0011	0111



1-5. Alphanumeric Codes

- **Any alphanumeric character set for English :**
  - at least 7-bit = 128 code,   6-bit = 64 code
  - 10 decimal digits
  - 26 letters of alphabet(uppercase, lowercase : 52)
  - several special characters

- **ASCII character code : original 7-bit code (Table 1-5)**
  - American Standard Code for Information Interchange

- **ASCII**

TABLE 1-5
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	NULL	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

Parity Bit

- to detect error in data communication and processing
- extra bit : included
  - to make the total no. of 1's in the resulting code word
    - either even(more common) or odd
- may be used with binary number
  - as well as with code, including ASCII for characters
- may be placed in any fixed position
  - ex. most significant position

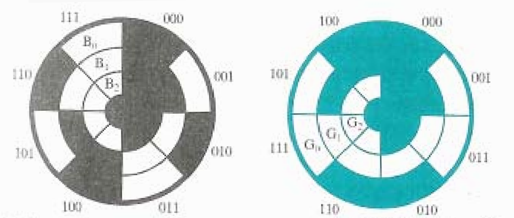
	With Even Parity	With Odd Parity
1000001	01000001	11000001
1010100	11010100	01010100

UNICODE

- UNICODE extends ASCII to 65,536 unversal characters codes
- For encoding characters in world languages
- Available in many modern applications
- 2 bytes (16-bit) code words
- For more detail, See Reading Supplement – Unicode on the Companion Website (http://www.prenhall.com/mano)

Gray Code

Binary Code	Bit Changes	Gray Code	Bit Changes
000		000	
	1		1
001		001	
	2		1
010		011	
	1		1
011		010	
	3		1
100		110	
	1		1
101		111	
	2		1
110		101	
	1		1
111		100	
	3		1
000		000	



(a) Binary Code for Positions 0 through 7 (b) Gray Code for Positions 0 through 7

Any Questions ??