Logical Organization of Computer
CHAPTER 1

DIGITAL SYSTEMS AND INFORMATION

Developed By:
Dr. Vivek Vyas
Department of MCA
DDU

1. INFORMATION REPRESENTATION

- The information represents a broad range of phenomena from the physical and man-made world.
- The physical world is characterized by parameters such as weight, temperature, pressure, velocity, flow, and sound intensity and frequency.
- Most physical parameters are continuous, typically capable of taking on all possible values over a defined range.
- In contrast, in the man-made world, parameters can be discrete in nature, such as business records using words, quantities, and currencies.

1. INFORMATION REPRESENTATION

- In general, information systems must be able to represent both continuous and discrete information.
- We refer to such a continuous voltage as an analog signal and discrete voltage as digital signal.
- The signals in most present-day electronic digital systems use just two discrete values and are therefore said to be binary.
- The two discrete values used are often called 0 and 1, the digits for the binary number system.

1. INFORMATION REPRESENTATION

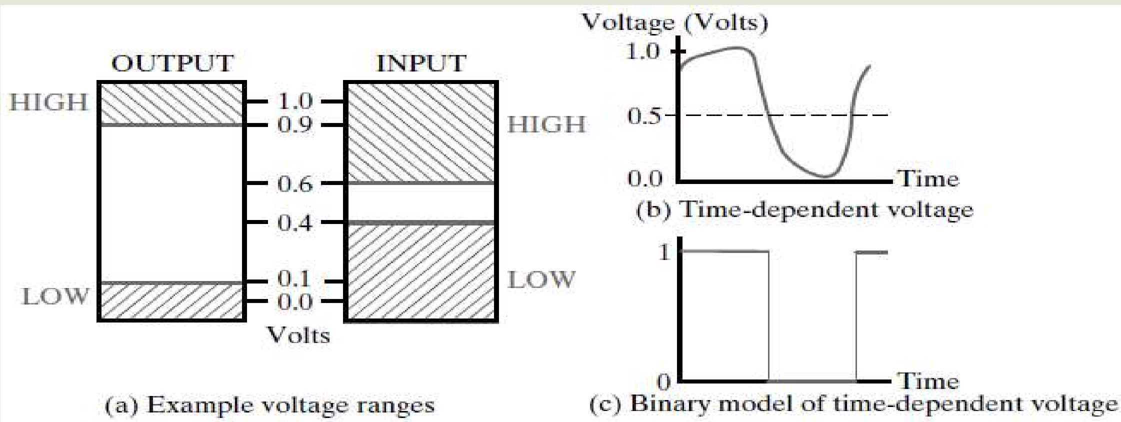
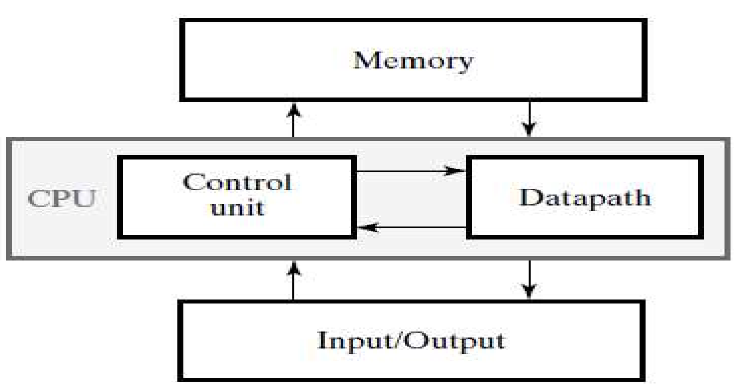


FIGURE 1
Examples of Voltage Ranges and Waveforms for Binary Signals

1. INFORMATION REPRESENTATION

- Since 0 and 1 are associated with the binary number system, they are the preferred names for the signal ranges. A binary digit is called a bit. Information is represented in digital computers by groups of bits.
- Why is binary used? In contrast to the situation in Diagram 1, consider a system with 10 values representing the decimal digits.
- In such a system, the voltages available—say, 0 to 1.0 volts—could be divided into 10 ranges, each of length 0.1 volt. A circuit would provide an output voltage within each of these 10 ranges. An input of a circuit would need to determine in which of the 10 ranges an applied voltage lies.

❑ The Digital Computer



❑ **FIGURE 2**
Block Diagram of a Digital Computer

❑ The Digital Computer

- The memory stores programs as well as input, output, and intermediate data.
- The datapath performs arithmetic and other data-processing operations as specified by the program.
- The control unit supervises the flow of information between the various units.
- A datapath, when combined with the control unit, forms a component referred to as a *central processing unit, or CPU.*

❑ The Digital Computer

- The program and data prepared by the user are transferred into memory by means of an input device such as a keyboard.
- An output device, such as an LCD (liquid crystal display), displays the results of the computations and presents them to the user.

2. NUMBER SYSTEMS

- The decimal number system is employed in everyday arithmetic to represent numbers by strings of digits. Depending on its position in the string, each digit has an associated value of an integer raised to the power of 10.
- For example, the decimal number 724.5 is interpreted to represent 7 hundreds plus 2 tens plus 4 units plus 5 tenths.
- The hundreds, tens, units, and tenths are powers of 10 implied by the position of the digits. The value of the number is computed as follows:

$$724.5 = 7 \times 10^2 + 2 \times 10^1 + 4 \times 10^0 + 5 \times 10^{-1}$$

2. NUMBER SYSTEMS

- The convention is to write only the digits and infer the corresponding powers of 10 from their positions.
- In general, a decimal number with *n* digits to the left of the decimal point and *m* digits to the right of the decimal point is represented by a string of coefficients:

$$A_{n-1}A_{n-2}\dots A_1A_0.A_{-1}A_{-2}\dots A_{-m+1}A_{-m}$$

2. NUMBER SYSTEMS

- In general, the "." is called the *radix point*. *An – 1* is referred to as the *most significant digit* (*msd*) and *Am* as the *least significant digit* (*lsd*) of the number.
- The following illustrates a base 5 number with *n* = 3 and *m* = 1 and its conversion to decimal:

$$(312.4)_5 = 3 \times 5^2 + 1 \times 5^1 + 2 \times 5^0 + 4 \times 5^{-1}$$
$$= 75 + 5 + 2 + 0.8 = (82.8)_{10}$$

❑ Binary Numbers

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

❑ Binary Numbers

- In digital systems, we refer to 2^{10} as K (kilo), 2^{20} as M (mega), 2^{30} as G (giga), and 2^{40} as T (tera). Thus,

$$4K = 2^2 \times 2^{10} = 2^{12} = 4096$$

$$16M = 2^4 \times 2^{20} = 2^{24} = 16,777,216$$

❑ Binary Numbers

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

❑ Binary Numbers

- **Binary to Decimal Conversion**

$$(11010)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = (26)_{10}$$

$$(110101.11)_2 = 32 + 16 + 4 + 1 + 0.5 + 0.25 = (53.75)_{10}$$

❑ Binary Numbers

- **Binary to Octal Conversion**

$$(010\ 110\ 001\ 101\ 011.\ 111\ 100\ 000\ 110)_2 = (26153.7406)_8$$

❑ Binary Numbers

- **Binary to Hexadecimal Conversion**

$$(0010\ 1100\ 0110\ 1011.\ 1111\ 0000\ 0110)_2 = (2C6B.F06)_{16}$$

❑ Decimal Numbers

- **Decimal to Binary conversion**
- $(625)_{10}$ = ?

$$\begin{aligned} 625 - 512 &= 113 = N_1 & 512 &= 2^9 \\ 113 - 64 &= 49 = N_2 & 64 &= 2^6 \\ 49 - 32 &= 17 = N_3 & 32 &= 2^5 \\ 17 - 16 &= 1 = N_4 & 16 &= 2^4 \\ 1 - 1 &= 0 = N_5 & 1 &= 2^0 \end{aligned}$$

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

❑ Binary Numbers

- **Conversion of Decimal Integers 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

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

❑ Binary Numbers

- **Conversion of Decimal Fractions 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 from Decimal to Other Bases

- **Conversion of Decimal Integers 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 from Decimal to Other Bases

- **Conversion of Decimal Fractions to Octal**

$0.513 \times 8 = 4.104$	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

$(0.513)_{10} = (0.407)_8$

❑ Octal and Hexadecimal Numbers

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

❑ Octal and Hexadecimal Numbers

- **Octal to Decimal Conversion**

$$(127.4)_8 = 1 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 4 \times 8^{-1} = (87.5)_{10}$$

❑ Octal and Hexadecimal Numbers

- **Octal to Binary Conversion**

$$(673.12)_8 \quad = \quad 110 \quad 111 \quad 011. \quad 001 \quad 010 = (110111011.00101)_2$$

❑ Octal and Hexadecimal Numbers

- **Hexadecimal to Decimal Conversion**

$$(B65F)_{16} = 11 \times 16^3 + 6 \times 16^2 + 5 \times 16^1 + 15 \times 16^0 = (46687)_{10}$$

❑ Octal and Hexadecimal Numbers

- **Hexadecimal to Binary Conversion**

$$(3A6.C)_{16} = 0011\ 1010\ 0110.\ 1100 \quad = (1110100110.11)_2$$

3. ARITHMETIC OPERATIONS

- Binary Addition

$$\begin{array}{r} 00000 \\ 01100 \\ +10001 \\ \hline 11101 \end{array} \qquad \begin{array}{r} 101100 \\ 10110 \\ +10111 \\ \hline 101101 \end{array}$$

3. ARITHMETIC OPERATIONS

- Binary Subtraction

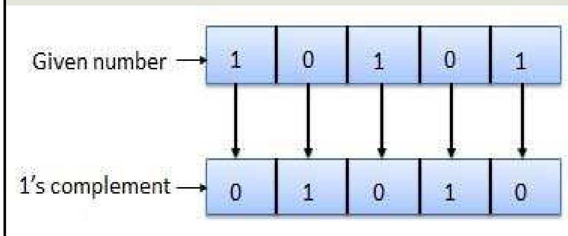
$$
\begin{array}{rr}
00000 & \quad 00110 \\
10110 & \quad 10110 \\
-10010 & \quad -10011 \\
\hline
00100 & \quad 00011
\end{array}
$$

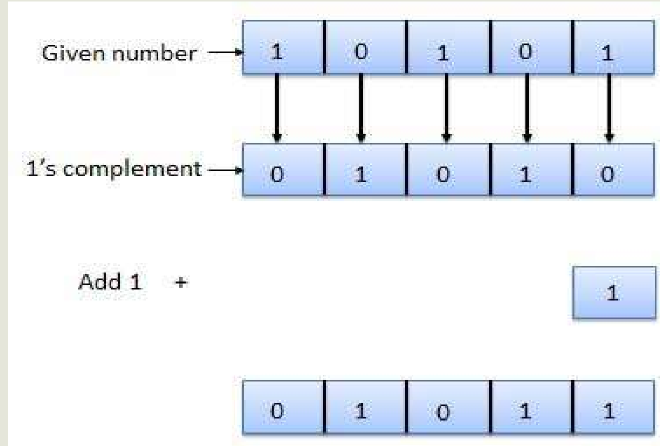
3. ARITHMETIC OPERATIONS

- Binary Multiplication

$$
\begin{array}{r}
1011 \\
\times \; 101 \\
\hline
1011 \\
0000 \\
1011 \\
\hline
110111
\end{array}
$$

❑ 1's & 2's complement





❑ r's & (r-1)'s complement

- Example: Find 3's complement and 2's complement of 121_3?
- 2's complement of 121_3
- = 222 – 121
- = 101_3
- And 3's complement of 121_3
- = 2's complement + 1
- = $101_3 + 1$
- = 102_3

❑ r's & (r-1)'s complement

- Example: Find the 3's complement and 4's complement of 130_4?
- 3's complement of 130_4
- = 333 – 130
- = 203_4
- 4's complement of 130_4
- = 3's complement of 130_4 + 1
- = 203_4+ 1
- = 210_4

❑ r's & (r-1)'s complement

- Example: Find the 4's complement and 5's complement of 224_5?
- 4's complement of 224_5
- = 444 – 224
- = 220_5
- 5's complement of 224_5
- = 4's complement of 224_5 + 1
- = 220_5+ 1
- = 221_5

❑ r's & (r-1)'s complement

- Example: Find 10's complement and 11's complement of $1A1_{11}$?
- 10's complement of $1A1_{11}$
- = AAA – 1A1
- = 909_{11}
- 11's complement of $1A1_{11}$
- = 10's complement of $1A1_{11}$ + 1
- = 909_{11}+ 1
- = $90A_{11}$

❑ r's & (r-1)'s complement

- Example: Find 11's complement and 12's complement of $AB0_{12}$?
- 11's complement of $AB0_{12}$
- = BBB – AB0
- = $10B_{12}$
- 12's complement of $AB0_{12}$
- = 11's complement of $AB0_{12}$ + 1
- = $10B_{12}$+ 1
- = 110_{12}

4. DECIMAL CODES

❑ **TABLE 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

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

5. ALPHANUMERIC CODES

- Many applications of digital computers require the handling of data consisting not only of numbers, but also of letters.
- For instance, an insurance company with thousands of policyholders uses a computer to process its files.
- To represent the names and other pertinent information, it is necessary to formulate a binary code for the letters of the alphabet.
- In addition, the same binary code must represent numerals and special characters such as $.

❑ ASCII Character Code

- The standard binary code for the alphanumeric characters is called ASCII (American Standard Code for Information Interchange). It uses seven bits to code 128 characters.
- The ASCII code contains 94 characters that can be printed and 34 nonprinting characters used for various control functions.
- The printing characters consist of the 26 uppercase letters, the 26 lowercase letters, the 10 numerals, and 32 special printable characters such as %, @, and $.

❑ ASCII Character 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	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

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

❑ Hamming Code

- 1001000
- $M+r+1<2^r$

R1	R2	1	R3	0	0	1	R4	0	0	0
1	2	3	4	5	6	7	8	9	10	11

❑ Hamming Code

- R1 = BIT NO. 1+3+5+7+9+11
- R2 = BIT NO. 2+3+6+7+10+11
- R3 = BIT NO. 4+5+6+7
- R4 = BIT NO. 8+9+10+11

❑ EBCDIC

- *Extended Binary Coded Decimal Interchange Code* (EBCDIC) is an 8-bit character encoding used on IBM mainframe and minicomputer operating systems.
- All IBM mainframe peripherals and operating systems still support EBCDIC, although the operating systems also provide ASCII and Unicode modes to allow translation between different encodings.

❑ EBCDIC

Character	EBDIC Bit Configuration	Character	EBDIC Bit Configuration
A	1100 0001	S	1110 0010
B	1100 0010	T	1110 0011
C	1100 0011	U	1110 0100
D	1100 0100	V	1110 0101
E	1100 0101	W	1110 0110
F	1100 0110	X	1110 0111
G	1100 0111	Y	1110 1000
H	1100 1000	Z	1110 1001
I	1100 1001	0	1111 0000
J	1101 0001	1	1111 0001
K	1101 0010	2	1111 0010
L	1101 0011	3	1111 0011
M	1101 0100	4	1111 0100
N	1101 0101	5	1111 0101
O	1101 0110	6	1111 0110
P	1101 0111	7	1111 0111
Q	1101 1000	8	1111 1000
R	1101 1001	9	1111 1001

❑ UNICODE

- The Unicode character set has the capacity to support over one million characters, and is being developed with an aim to have a single character set that supports all characters from all scripts, as well as many symbols, that are in common use around the world today or in the past.
- Currently, the Standard supports over 96,000 characters representing a large number of scripts. UTF-8, UTF-16 and UTF-32 are part of Unicode.