UNIVERSITY OF BABYLON
COLLEGE OF SCIENCE FOR WOMEN
COMPUTER DEPARTMENT

2024/ 2025
FIRST CLASS

Computer Organization

LECTURES

PREPARED BY:
LECTURER: ***Dr. Ahmed Mohammed Hussein***

2024-2025

❖ LEARNING OBJECTIVES

After completion of this lecture, you should be able to:

- Describe how the computer processes textual information.
- Describe the data representation

1. CHARACTERS AND STRING OF TEXT

A personal computer processes textual information. Individual key-strokes generate letters, numbers, or other symbols that are called **characters**. Groups of characters treated as units are called **strings**. Since characters and strings can't be processed directly by machine that "understand" only binary numbers, they must be encoded in some kind of binary code. The computer uses coding system called ***ASCII*** (**American Standard Code for Information Interchange**). ASCII associates a unique **8-bits** binary with each character symbol.

The standard ASCII uses 7 bits to encode a character. Thus, 2^7=128 different characters can be represented. This number is sufficiently large to represent uppercase and lowercase characters, digits, special characters such as !,^, and control characters such as CR (carriage return), LF (line feed), and so on.

Since we store the bits in units of a power of 2, we end up storing 8 bits for each character, even though ASCII requires only 7 bits. The eighth bit is put to use for two purposes:

1. ***To Parity Encode for Error Detection:*** The eighth bit can be used to represent the parity bit. This bit is made 0 or 1 such that the total number of 1s in a byte is even (for even parity) or odd (for odd parity). This can be used to detect simple errors in data transmission.
2. ***To Represent an Additional 128 Characters:*** By using all 8 bits we can represent a total of 2^8=256 different characters. This is referred to as extended ASCII.

Thus, an additional 128 encodings have been added, mostly to take care of the Latin letters, accents, and diacritical marks. . The biggest decimal equivalent we can express in 8-bits is **11111111,** which is the sum of all powers of two from zero to seven.

11111111 binary = 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

= 255 decimal.

In ASCII the letter **A** is associated with **01000001** binary. The numeral (not its value) **1** is associated with **00110001** binary. People usually convert binary codes to their decimal equivalents when they refer to them. Hence, **A** is represented by the decimal number **65** and numeral **1** is represented by decimal number **49**.

Suppose you want to store a line in computer memory. You enter the string of characters into the computer through the keyboard. The keyboard converts ***each*** key-stroke into ASCII binary code.

Everything is exactingly stored as binary numbers: letters of alphabet, numerals, punctuation marks, and special characters **($, #, %, and so on**). Each character has its own 8-bits code. The memory of computer can record binary digits and nothing else, so all kinds of information must be encoded. ***The power of a computer to manipulate symbols is hidden in the simplicity of coding scheme*.**

Table 1: illustrates Hexa and decimal codes for ASCII control codes

ASCII Control Codes

Hex	Decimal	Character	Meaning
00	0	NUL	NULL
01	1	SOH	Start of heading
02	2	STX	Start of text
03	3	ETX	End of text
04	4	EOT	End of transmission
05	5	ENQ	Enquiry
06	6	ACK	Acknowledgment
07	7	BEL	Bell
08	8	BS	Backspace
09	9	HT	Horizontal tab
0A	10	LF	Line feed
0B	11	VT	Vertical tab
0C	12	FF	Form feed
0D	13	CR	Carriage return
0E	14	SO	Shift out
0F	15	SI	Shift in
10	16	DLE	Data link escape
11	17	DC1	Device control 1
12	18	DC2	Device control 2
13	19	DC3	Device control 3
14	20	DC4	Device control 4
15	21	NAK	Negative acknowledgment
16	22	SYN	Synchronous idle
17	23	ETB	End of transmission block
18	24	CAN	Cancel
19	25	EM	End of medium
1A	26	SUB	Substitute
1B	27	ESC	Escape
1C	28	FS	File separator
1D	29	GS	Group separator
1E	30	RS	Record separator
1F	31	US	Unit separator
7F	127	DEL	Delete

Table 2: illustrates Hexa and decimal codes for ASCII printable character codes.

ASCII Printable Character Codes†

Hex	Decimal	Character	Hex	Decimal	Character	Hex	Decimal	Character
20	32	Space	40	64	@	60	96	‘
21	33	!	41	65	A	61	97	a
22	34	”	42	66	B	62	98	b
23	35	#	43	67	C	63	99	c
24	36	$	44	68	D	64	100	d
25	37	%	45	69	E	65	101	e
26	38	&	46	70	F	66	102	f
27	39	’	47	71	G	67	103	g
28	40	(	48	72	H	68	104	h
29	41	)	49	73	I	69	105	i
2A	42	*	4A	74	J	6A	106	j
2B	43	+	4B	75	K	6B	107	k
2C	44	,	4C	76	L	6C	108	l
2D	45	–	4D	77	M	6D	109	m
2E	46	.	4E	78	N	6E	110	n
2F	47	/	4F	79	O	6F	111	o
30	48	0	50	80	P	70	112	p
31	49	1	51	81	Q	71	113	q
32	50	2	52	82	R	72	114	r
33	51	3	53	83	S	73	115	s
34	52	4	54	84	T	74	116	t
35	53	5	55	85	U	75	117	u
36	54	6	56	86	V	76	118	v
37	55	7	57	87	W	77	119	w
38	56	8	58	88	X	78	120	x
39	57	9	59	89	Y	79	121	y
3A	58	:	5A	90	Z	7A	122	z
3B	59	;	5B	91	[	7B	123	{
3C	60	<	5C	92	\	7C	124	|
3D	61	=	5D	93	]	7D	125	}
3E	62	>	5E	94	^	7E	126	~
3F	63	?	5F	95	_			

†Note that 7FH (127 in decimal) is a control character listed on the previous page.

STUDENT-ACTIVITY

1. Explain how the computer processes textual information.
2. ASCII is standing for what?
3. What are the three widely used techniques for representing both positive and negative numbers
4. Convert the following binary numbers to its decimal codes:

(00110011), (00000111), (10000001), (10101010), (11110000)