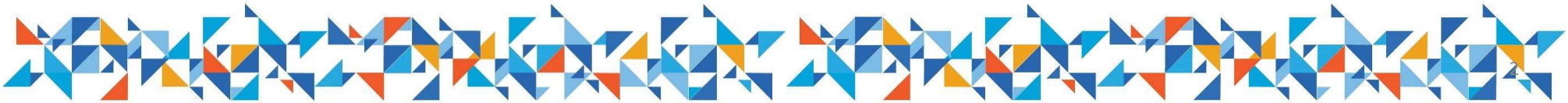
CONTENTS

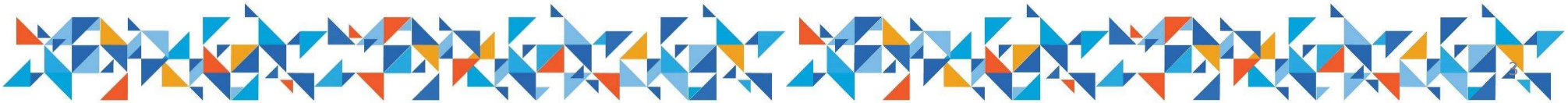
- BCD encoding scheme and Addition of BCD
- BCD-6 or 6-bit BCD
- Alphanumeric Code
- ASCII Code
- EBCDIC (Extended Binary Coded Decimal Interchange Code)
- Unicode encoding schemes (UTF-8, UTF-16 & UTF-32)



OUTCOMES

At the end of this lecture, students will know;

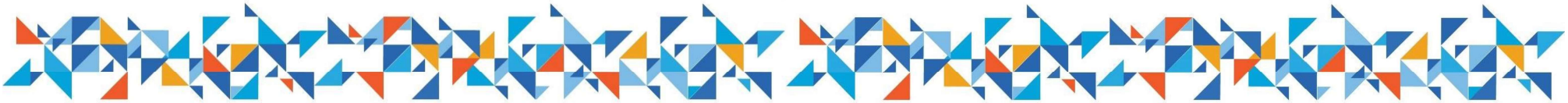
- Basic concept of BCD and addition of BCD.
- Ascii code.



BCD

Definition:

- BCD (Binary-Coded Decimal) is a way of encoding each digit of a decimal number separately using a fixed number of binary bits.
- Binary-Coded Decimal (BCD) is a representation method used to encode decimal digits using binary digits (0s and 1s).
- One of the earlier method of encoding decimal numbers.
- A 4-bit BCD (Binary Coded Decimal) can represent a total of 16 different characters. Each of the 4 bits can have two possible values (0 or 1), so there are a total of 2^4 = 16 possible combinations. However, in BCD, only the values from 0 to 9 are used to represent decimal digits.

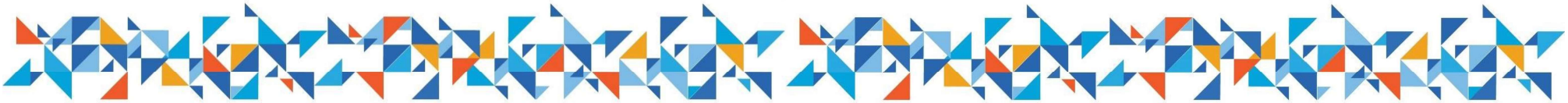


BCD….

Definition:

- Binary coded decimal (BCD) is a system of writing numbers that assigns a four-digit binary code to each digit 0 through 9 in a decimal (base-10) numeral.
- The four-bit BCD code for any particular single base-10 digit is represented as follows:

• 0 = 0000	5 = 0101
• 1 = 0001	6 = 0110
• 2 = 0010	7 = 0111
• 3 = 0011	8 = 1000
• 4 = 0100	9 = 1001



BCD Exercise….

Question: write 4-bit and 6-bit BCD codes for the following decimal numbers?

1) 5
2) 56 = 0101 0110 (one is done for you) (000101 000110 6-bit BCD)
3) 90
4) 23
5) 11
6) 34

BCD Exercise….

Question: find the decimal numbers represented by following BCD codes?

1) 0000 0111 1001 (4 bit BCD) **Answer: 079**
2) 001001 000000 000100 (6 bit BCD) Answer : 904

BCD….

How BCD works::

In BCD, each decimal digit (0-9) is represented by its 4-bit binary equivalent. This means each decimal digit is encoded using exactly four binary bits.

Example:

Let's take the decimal number 362. In BCD, each digit is encoded separately:

3 is 0011 in BCD (binary representation of 3).

6 is 0110 in BCD (binary representation of 6).

2 is 0010 in BCD (binary representation of 2).

To represent a multi-digit number 362 in BCD, you group the BCD codes of each digit together. So, for 362, the BCD representation would be 0011 0110 0010.

Why BCD is used ?

1.Advantages:

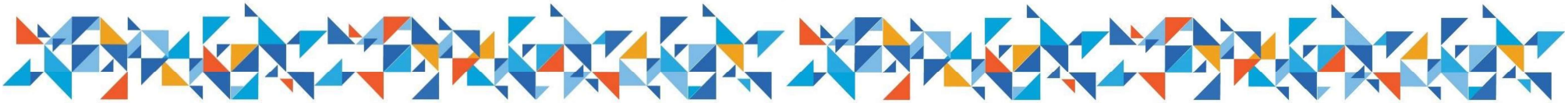
1. BCD is simple to understand and work with because it directly represents decimal digits.
2. It's easy to convert BCD to decimal and vice versa.

2.Disadvantages:

1. The major drawback of 4-bit BCD representation is that it can represent the decimal numbers only and cannot represent the various characters.

3.Usage:

1. BCD is often used in applications where decimal calculations are needed, like calculators and financial systems, because it makes it easier to perform arithmetic operations with decimal numbers.



BCD to decimal and decimal to BCD conversions

Conversions:

1) BCD to Decimal: To convert BCD to decimal, you group the BCD bits in sets of four and convert each set to its decimal equivalent.

i.e.

BCD: 0101 0010

Decimal: 5 2

2) Decimal to BCD: To convert decimal to BCD, you break down each decimal digit into its 4-bit binary equivalent and combine them.

i.e.

Decimal: 7 6

BCD: 0111 0110

Standard BCD versus 6-bit BCD

- A 6-bit BCD (Binary Coded Decimal) is a binary encoding scheme used to represent decimal numbers using 6 bits of binary data for each decimal digit.
- In standard BCD, each decimal digit is represented by a 4-bit binary code. However, in some cases, 6-bit BCD is used to represent decimal digits, allowing for a larger range of values to be encoded.
- 6-bit BCD can encode in binary the 10 digits from 0 to 9, all 26 alphabets and 28 special characters.
- With 6 bits, you have a total of 2^6 = 64 possible combinations.

Alphanumeric Code

What is Alphanumeric Code ?

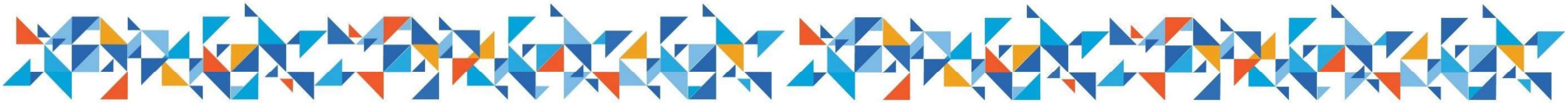
- Alphanumeric codes are a way for computers to represent a wide range of characters using a set of symbols or numbers. The term "alphanumeric" combines "alpha" (for letters of the alphabet) and "numeric" (for numbers), indicating that these codes can handle both letters and numbers, as well as other special characters.

Representation:

- Alphanumeric codes provide a mapping between characters (letters, numbers, symbols) and binary numbers (0s and 1s) that computers can understand. Each character is assigned a unique binary-code, which the computer uses to identify and process the character.
- The most common alphanumeric codes used these days are ASCII code, EBCDIC code, and UNICODE.

Alphanumeric Code uses

- **Communication and Storage:** Alphanumeric codes are essential for communication between computers and devices. When you type on a keyboard, the letters you press are translated into alphanumeric codes that the computer processes. Similarly, when you save a document or send a message, the text is stored or transmitted using these codes.
- **Display:** Computers also use alphanumeric codes to display characters on screens. The codes are converted back into recognizable characters so that you can read what's being shown on your device.
- alphanumeric codes serve as a bridge between human-readable characters and computer-understandable binary data. They enable computers to handle and interpret text, numbers, and symbols, allowing us to interact with technology in meaningful ways, from typing and messaging to creating documents and browsing the internet.



ASCII Codes

ASCII Code:

- One of the most well-known alphanumeric encoding systems is ASCII (American Standard Code for Information Interchange).
- In ASCII, each character is represented by a 7-bit code, allowing for a total of 128 different characters. This includes uppercase and lowercase letters, numbers, punctuation marks, and control characters (like Enter or Tab).
- ASCII (American Standard Code for Information Interchange) is the most common character encoding format for text data in computers and on the internet.

ASCII Codes

ASCII control characters		
00	NULL	(Null character)
01	SOH	(Start of Header)
02	STX	(Start of Text)
03	ETX	(End of Text)
04	EOT	(End of Trans.)
05	ENQ	(Enquiry)
06	ACK	(Acknowledgement)
07	BEL	(Bell)
08	BS	(Backspace)
09	HT	(Horizontal Tab)
10	LF	(Line feed)
11	VT	(Vertical Tab)
12	FF	(Form feed)
13	CR	(Carriage return)
14	SO	(Shift Out)
15	SI	(Shift In)
16	DLE	(Data link escape)
17	DC1	(Device control 1)
18	DC2	(Device control 2)
19	DC3	(Device control 3)
20	DC4	(Device control 4)
21	NAK	(Negative acknowl.)
22	SYN	(Synchronous idle)
23	ETB	(End of trans. block)
24	CAN	(Cancel)
25	EM	(End of medium)
26	SUB	(Substitute)
27	ESC	(Escape)
28	FS	(File separator)
29	GS	(Group separator)
30	RS	(Record separator)
31	US	(Unit separator)
127	DEL	(Delete)

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

ASCII encoding can encode?

The ASCII (American Standard Code for Information Interchange) encoding method encodes a set of 128 characters using 7-bit binary code. These characters include:

Control characters (codes 0-31): These are non-printable characters used for control and communication purposes, such as carriage return (CR), line feed (LF), tab (TAB), and others.

Printable characters (codes 32-127): These include letters (both uppercase and lowercase), numbers, punctuation marks, and some special characters like the space character.

Uppercase letters: A-Z (Codes 65-90)

Lowercase letters: a-z (Codes 97-122)

Digits: 0-9 (Codes 48-57)

Space: (Code 32)

Punctuation marks: . , ; : ! ? ' " (and more)

Special characters: @ # $ % & * ( ) _ + - = / \ | [ ] { } < > and more

ASCII is a 7-bit encoding system, so each character is represented using a 7-bit binary code. It was widely used in early computer systems and is still used as a basis for character encoding in modern systems, especially for the first 128 characters.

EBCDIC

- EBCDIC, which stands for Extended Binary Coded Decimal Interchange Code, is a character encoding scheme .
- Encoding Format: EBCDIC is an 8-bit encoding scheme, which means it uses 8 bits (or one byte) to represent each character. This allows for a total of 256 possible characters, including letters, numbers, special symbols, and control codes.
- EBCDIC is a character encoding scheme that was developed by IBM for its mainframe and midrange computer systems. While it played a crucial role in early computing, its usage has significantly declined in favor of ASCII and Unicode on modern computing platforms.

The character that can be encoded by EBCDIC

❖EBCDIC (Extended Binary Coded Decimal Interchange Code) is another character encoding method, primarily used in early IBM mainframe computers.

the characters encoded by EBCDIC:

✓Uppercase letters: A-Z

✓Lowercase letters: a-z (in some variations)

✓Digits: 0-9

✓Punctuation marks: . , ; : ! ? ' " (and more)

✓Special characters: @ # $ % & * ( ) _ + - = / \ | [ ] { } < > and more

NOTE: EBCDIC encoding varies slightly between different EBCDIC code pages or versions, and there were several variations used in different regions and for different purposes. For example, EBCDIC-US, EBCDIC-UK, and EBCDIC-FR were used in the United States, the United Kingdom, and France, respectively.

EBCDIC is less common today than ASCII or UTF-8 encoding. Modern computer systems and programming languages typically use ASCII or Unicode-based encodings like UTF-8, which can represent a much wider range of characters from various languages and scripts, making them more suitable for internationalization and modern computing needs.

Extended Alphanumeric Codes(Unicode encoding system)

- Extended alphanumeric encoding systems are expansions of traditional character encoding schemes, like BCD, ASCII (American Standard Code for Information Interchange), that allow for a broader range of characters to be represented, including special characters, symbols, and characters from various languages and writing systems.
- Extended alphanumeric encoding systems, particularly those based on Unicode like UTF-8, UTF-16, and UTF-32, allow computers to handle a wide range of characters from different languages, writing systems, and symbols. These encoding systems play a crucial role in modern communication, data storage, software development, and global information exchange.

Unicode encoding system

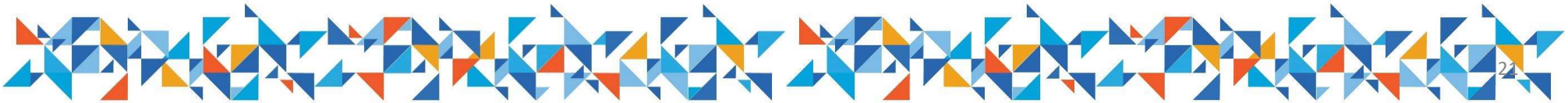
- Unicode came into existence to express a huge variety of characters that include letters, numbers, symbols, and even special characters from different languages, like smiley faces and hearts. It's a way to make sure that computers can handle text in lots of languages
- Unicode uses a variable number of bits (usually 8, 16, or 32 bits) to represent characters from different writing systems worldwide.
- In Unicode Each character is assigned a unique pattern of 0s and 1s that the computer can easily recognize.

Unicode uses

➢ **Uses of Unicode**

➢ Language Support: Unicode makes it possible for you to type, read, and display text in different languages. So, whether you're writing in English, Chinese, Arabic, or any other language, Unicode helps your computer understand and show the right characters.

- International Communication: When you send an email or a text message to someone from another part of the world, Unicode ensures that the characters in your message are correctly displayed on their end.
- **Operating Systems**: Windows, macOS, Linux.
- **Programming Languages**: Python, Java, JavaScript, C++, etc.
- **Web Development**: HTML, CSS, JSON, XML.
- **Databases**: Supports multilingual data storage and retrieval.



Unicode uses

- Digital Content: Websites, apps, and social media platforms use Unicode to show text and symbols correctly. This means you can post and view content in various languages and scripts without any issues.
- In summary, Unicode is like a language translator for computers. It lets them understand and work with text from all corners of the world, making communication, content creation, and digital interaction possible across languages and cultures.
- Unicode enabled multilingual computing, enabled the coexistence of various languages within a single system. In the context of Digital Logic Design (DLD), Unicode serves as a cornerstone for character representation and processing.

Unicode variant

- UTF-8 (Unicode Transformation Format - 8-bit) It can represent all Unicode characters using one to four bytes. "UTF-8 encodes a character into a binary string of one, two, three, or four bytes.“UTF-8 can represent over 1 million different characters.
- UTF-8 encoding is its backward compatibility with ASCII. The first 128 characters in the Unicode library match those in the ASCII library, and UTF-8 translates these 128 Unicode characters into the same binary strings as ASCII.
- UTF-8 is widely used on the internet due to its efficiency in encoding common characters while accommodating a broad range of characters.
- **UTF-16 (Unicode Transformation Format - 16-bit):** UTF-16 uses 16-bit units and can represent all Unicode characters using one or two 16-bit units (2 or 4 bytes). "UTF-16 encodes a Unicode character into a string of either two or four bytes.“
- UTF-16 can theoretically represent over 65,000 different characters using two 16-bit units (2 bytes) per character.

Unicode variants….

UTF-32 (Unicode Transformation Format - 32-bit)

- UTF-32: Like the other variants, it can represent all Unicode characters, using a fixed 32-bit unit (4 bytes) for each character. UTF-32 is less commonly used due to its higher memory requirements.

Unicode variants differences

- In terms of their ability to represent characters, all these Unicode encoding variants are equal. The choice between them usually depends on factors like memory efficiency, platform compatibility, and performance requirements.
- **Practical Usage:**
- UTF-8 is dominant for web content and data interchange due to its space efficiency and compatibility.
- UTF-16 is used in Windows systems and certain programming contexts.
- UTF-32 is rarely used outside of specialized applications requiring consistent memory usage.

Unicode variants differences

- UTF-8 is backward-compatible with ASCII, making it suitable for systems and applications that rely heavily on ASCII characters.
- UTF-16 is used extensively in Windows systems and some programming languages, where two-byte units can represent a substantial portion of characters.
- UTF-32 is less commonly used due to its memory requirements

Unicode character set

- The Unicode character set refers to a standardized collection of characters, symbols, and special characters that have been assigned unique code points (numeric identifiers) for representation in digital systems.
- This character set is designed to include characters from various languages, writing systems, symbols, emoji's, and more, with the aim of enabling multilingual computation.
- As of Unicode 14.0, which was released in September 2021, the Unicode standard defines over 143,000 characters.

Unicode character set……

- It's important to note that the Unicode character set is continually evolving. With each new version of the Unicode standard, new characters may be added to accommodate the needs of global communication, technical requirements, and cultural representation.
- All major Unicode encoding variants (UTF-8, UTF-16, and UTF-32) are capable of representing the entire range of characters defined by the Unicode standard.
- The differences between these variants lie in their encoding mechanisms and memory usage, not in the number of characters they can represent.

Space taken by UTF encoding schemes

UTF-8:

- ✓Uses 1 byte for most common characters, including ASCII characters (0-127).
- ✓Can use 2, 3, or 4 bytes for less common characters, allowing it to represent a wide range of characters from different scripts and symbols.

UTF-16:

- ✓Uses 2 bytes (16 bits) for most characters, including many common international characters.
- ✓Uses 4 bytes (32 bits) for less common characters and certain special characters represented by surrogate pairs.

UTF-32:

- ✓Uses a fixed 4 bytes (32 bits) for every character, regardless of its rarity or uniqueness.
- ✓Provides a consistent and simple encoding scheme but may be less memory-efficient than UTF-8 or UTF-16 for text that primarily consists of common characters.

Space taken by UTF……..

- In summary, UTF-8 and UTF-16 are variable-length encodings, with UTF-8 being more space-efficient for common characters, while UTF-32 is a fixed-length encoding where every character occupies 4 bytes.
- The choice of encoding depends on the specific requirements and the range of characters that need to be represented.

پوهنتون کاردان
KARDAN UNIVERSITY

Thank You…!