Digital Logic Syllabus

- **Number Systems:-** Binary, Octal and Hexa decimal representation and their conversions; BCD,ASCII, EBCDIC, Gray codes and their conversions;
- **Binary number representation:-** Signed binary number representation with 1's and 2's complement methods, Binary arithmetic.
- **Boolean Expression:-** Various Logic gates-their truth tables and circuits; Representation in SOP and POS forms; Minimization of logic expressions by algebraic method, Kmap method,
- **Combinational Circuits:** Adder and Subtractor circuits; Applications and circuits of Encoder ,Decoder, Comparator ,Multiplexer, De-Multiplexer and Parity Generator.
- **Sequential Circuits:** Basic memory element-S-R, J-K,D and T Flip Flops, various types of Registers and counters and their design.
- **Memory circuits:** RAM ,ROM, EPROM, EEPROM, Design of combinational circuits-using ROM, PLAs and PLDs. Logic familiesTTL,ECL,MOS and CMOS, their operation and specifications.

DIGITAL SYSTEM

- A Digital system is an interconnection of digital modules and it is a system that manipulates discrete elements of information that is represented internally in the binary form.
- Digital systems are used in wide variety of industrial and consumer products such as automated industrial machinery, pocket calculators, microprocessors, digital computers, digital watches, TV games and signal processing and so on.

Characteristics of Digital systems

• Digital systems manipulate discrete elements of information.

• Discrete elements are nothing but the digits such as 10 decimal digits or 26 letters of alphabets and so on.

• Digital systems use physical quantities called signals to represent discrete elements.

• In digital systems, the signals have two discrete values and are therefore said to be binary.

• A signal in digital system represents one binary digit called a bit. The bit has a value either 0 or 1.

Advantages of Digital system

1. **Ease of programmability:-**
   The digital systems can be used for different applications by simply changing the program without additional changes in hardware.
2. **Reduction in cost of hardware:-**
   The cost of hardware gets reduced by use of digital components and this has been possible due to advances in IC technology. With ICs the number of components that can be placed in a given area of Silicon are increased which helps in cost reduction.
3. **High speed:-**
   Digital processing of data ensures high speed of operation which is possible due to advances in Digital Signal Processing.
4. **High Reliability:-**
   Digital systems are highly reliable one of the reasons for that is use of error correction codes.

5. **Design is easy:-**
   The design of digital systems which require use of Boolean algebra and other digital techniques is easier compared to analog designing.

6. **Result can be reproduced easily:-**
   Since the output of digital systems unlike analog systems is independent of temperature, noise, humidity and other characteristics of components the reproducibility of results is higher in digital systems than in analog systems.

Disadvantages of Digital Systems

- Use more energy than analog circuits to accomplish the same tasks, thus producing more heat as well.
- Digital circuits are often fragile, in that if a single piece of digital data is lost or misinterpreted the meaning of large blocks of related data can completely change.
- Digital computer manipulates discrete elements of information by means of a binary code.
- Quantization error during analog signal sampling.
- Number system is a basis for counting varies items. Modern computers communicate and operate with binary numbers which use only the digits 0 &1.

Character Encoding

- Character encoding is the process of assigning numbers to graphical characters
- Character encoding tells computers how to interpret digital data into letters, numbers and symbols.
- **ASCII (American Standard Code for Information Interchange (1963):** ASCII is a 7-bit character encoding standard, representing 128 characters, including control characters, English letters, digits, and punctuation.
- **Extended ASCII:** Later, 8-bit versions of ASCII were developed, allowing for 256 characters, including additional symbols and characters for other languages.
- **EBCDIC (Extended Binary Coded Decimal Interchange Code):**
- IBM developed EBCDIC (Extended Binary-Coded Decimal Interchange Code), which is an 8-bit character encoding based on BCD.
- An 8-bit character encoding used primarily on IBM mainframe (IBM System/360) and midrange systems.
- **Unicode:** The most widely used encoding system today, capable of representing characters from nearly every writing system in the world, including emojis.

UTF-8: A variable-length encoding that uses 1 to 4 bytes per character. It is backward-compatible with ASCII and is the most common encoding on the web.
UTF-16: Uses 2 or 4 bytes per character and is commonly used in Windows and Java.
UTF-32: Uses 4 bytes per character, providing direct access to all Unicode code points, but is less common due to its inefficiency in terms of space.

Number System ($)_x$

1. **Decimal Number: ($)_{10}$**
   The decimal system has ten symbols: 0,1,2,3,4,5,6,7,8,9.
   It has a radix/ base of 10.
   In binary system weight is expressed as power of 10.

2. **Binary number: ($)_2$**
   It has only 2 symbols: 0,1.
   The binary number has a radix of 2.
   In binary system weight is expressed as power of 2.

3. **Octal Number: ($)_8$**
   Octal systems use a base or radix of 8. It uses first eight digits of decimal number system. Thus it has digits from 0,1,2,3,4,5,6,7.

4. **Hexadecimal Number: ($)_{16}$**
   It has a base of 16. There are 16 symbols. The decimal digits 0 to 9 are used as the first ten digits as in the decimal system, followed by the letters A, B, C, D, E and F, which represent the values 10, 11,12,13,14 and 15 respectively.

Number Base conversions

1. **Decimal to Binary:** To convert decimal number in any other number system (base) divide the given number with the desired number system (base) and write down the reminder from bottom to up.
   Here we have to convert in binary so divide by base 2.

   Example: $(36)_{10} = (100100)_2$

		Reminder
2	36	0
2	18	0
2	9	1
2	4	0
2	2	0
	1	1

2. **Decimal to octal:** To convert decimal number in any other number system(base) divide the given number with the desired number system(base) and write down the reminder from bottom to up.
Here we have to convert in octal so divide by base 8.

Example: $(136)_{10} = (210)_8$

		Reminder
8	136	0
8	17	1
	2 →	2

3. **Decimal to Hexadecimal:** To convert decimal number in any other number system(base) divide the given number with the desired number system(base) and write down the reminder from bottom to up.
Here we have to convert in Hexadecimal so divide by base 16.

Example: $(156)_{10} = (9C)_{16}$

		Reminder
16	156	12 -> C
	9 →	9

4. **Binary to Decimal:** To convert any number system (base) in Decimal number system multiply the given number with given base weight(place value) then add all the digits to get the decimal number.
Here we are converting binary number to decimal so multiply each bit with the weight of each bits weight (place value)

Example: $(110101)_2 = (53)_{10}$

$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
$= 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 53$

5. **Octal to Decimal:** To convert any number system (base) in Decimal number system multiply the given number with given base weight(place value) then add all the digits to get the decimal number.
Here we are converting Octal number to decimal so multiply each bit with the weight of each bits weight (place value).

Example: $(145)_8 = (101)_{10}$

$1 \times 8^2 + 4 \times 8^1 + 5 \times 8^0$
$= 1 \times 64 + 4 \times 8 + 5 \times 1 = 101$

6. **Hexadecimal to Decimal:** To convert any number system (base) in Decimal number system multiply the given number with given base weight(place value) then add all the digits to get the decimal number.

   Here we are converting Hexadecimal number to decimal so multiply each bit with the weight of each bits weight (place value).

   Example: $(136)_{16} = (310)_{10}$

   $1 \times 16^2 + 3 \times 16^1 + 6 \times 16^0$

   $= 1 \times 256 + 3 \times 16 + 6 \times 1 = 310$

7. **Binary to octal:** (make pair of 3 bits from left to right and then convert each pair in decimal)

   Example: $(1100110101)_2 = (1465)_8$

   0 0 1 | 1 0 0 | 1 1 0 | 1 0 1
   ↓ ↓ ↓ ↓
   1 4 6 5

8. **Binary to Hexadecimal:** (make pair of 4 bits from left to right and then convert each pair in decimal)

   Example: $(1100101110101)_2 = (1465)_{16}$

   1 1 0 1 | 1 0 0 1 | 0 1 1 1 | 0 1 0 1
   ↓ ↓ ↓ ↓
   13 9 7 5
   ↓
   D

9. **Octal to Binary: (convert every individual decimal digit in 3bit binary number and then write together)**

   Example: $(453)_8 = (100101011)_2$

   4 5 3
   ↓ ↓ ↓
   100 101 011

10. Hexadecimal to Binary: (convert every individual decimal digit in 4 bit binary number and then write together)

Example: $(638)_{16} = (011000111000)_2$

$$\begin{array}{ccc} 6 & 3 & 8 \\ \downarrow & \downarrow & \downarrow \\ 0110 & 0011 & 1000 \end{array}$$

11. Octal to Hexadecimal: (Convert the given octal in decimal or binary and then convert that Decimal or binary number to hexadecimal)

Here I have converted it in binary number for ease.

Example: $(2341)_8 = (010011100001)_2 = (4E1)_{16}$

12. Hexadecimal to Octal: (Convert the given hexadecimal in Decimal or Binary and then convert that Decimal or binary number to octal)

Here I have converted it in binary number for ease.

Example: $(3B69)_{16} = (0011101101101001)_2 = (35551)_8$

Number Base conversions of fraction number

1. Fraction Decimal to Binary:

Example:

$(36.65)_{10} = (100100.\ 1010)_2$

Intregal Part

		Reminder
2	36	0
2	18	0
2	9	1
2	4	0
2	2	0
	1	1

To solve fraction part we multiply with given base

0.65 x 2= 1.30
0.30 x 2 = 0.60
0.60 x 2 = 1.20
0.20 x 2 = 0.40

2. **Fraction Decimal to octal:** To solve fraction part we multiply with given base. Either encounter 0 or get repeated pattern or options given in the question matched.

Example: $(136.123)_{10} = (210.0767)_8$

		Reminder
8	136	0
8	17	1
	2 →	2

0.123 x 8 = 0.984
0.984 x 8 = 7.872
0.872 x 8 = 6.976
0.976 x 8 =7.808

3. **Fraction Decimal to Hexadecimal:**

Example: $(156.25)_{10} = (9C.4)_{16}$

		Reminder
16	156	12 -> C
	9 →	9

0.25 x 16 =4.00

4. **Fraction Binary to Decimal:**

Example: $(110101.101)_2 = (53.625)_{10}$

$1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \; . \; 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$
$= 1 \times 32 + 1 \times 16 + 0 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \; . \; 1 \times 1/2 + 1 \times 1/8 = 53.625$

5. **Fraction Octal to Decimal:**

Example: $(145.24)_8 = (101.3125)_{10}$

$1 \times 8^2 + 4 \times 8^1 + 5 \times 8^0 \; . \; 2 \times 8^{-1} + 4 \times 8^{-2}$
$= 1 \times 64 + 4 \times 8 + 5 \times 1 \; . \; 2 \times 1/8 + 4 \times 1/64 = 101.3125$

6. **Fraction Hexadecimal to Decimal:**

Example: $(136.29)_{16} = (310.1601)_{10}$

$1 \times 16^2 + 3 \times 16^1 + 6 \times 16^0 \; . \; 2 \times 16^{-1} + 9 \times 16^{-2}$
$= 1 \times 256 + 3 \times 16 + 6 \times 1 \; . \; 2 \times 1/16 + 9 \times 1/256 = 310.1601$

7. Fraction Binary to octal:

Example: $(1100110101.1011)_2 = (1465.54)_8$

001 100 110 101 . 101 100

1 4 6 5 . 5 4

8. Fraction Binary to Hexadecimal:

Example: $(1100101110101.11001)_2 = (1465.C\ 8)_{16}$

1101 1001 0111 0101 . 1100 1000

13 9 7 5 . 12 8

D C

9. Fraction Octal to Binary:

Example: $(453.21)_8 = (100101011.010001)_2$

4 5 3 . 2 1

100 101 011 . 010 001

10. Fraction Hexadecimal to Binary: (convert every individual decimal digit in 4 bit binary number and then write together)

Example: $(638.92)_{16} = (011000111000.10010010)_2$

6 3 8 . 9 2

0110 0011 1000 . 1001 0010

11. Fraction Octal to Hexadecimal:

Example: $(2341.11)_8 = (010011100001.001001)_2 = (4E1.24)_{16}$

12. Fraction Hexadecimal to Octal:

Example: $(3B69.4C)_{16} = (0011101101101001.010011)_2 = (35551.23)_8$

Addition and subtraction of binary number

Carry 1 11
1101011
+ 1001011
10110110

Carry 111
100111
+ 100101
1001100

Borrow 1 1 2
1100011
- 1001111
0010100

Borrow
1101011
- 1001011
0100000

Addition of any number system

Carry 111
$(4576)_8$
+ $(3245)_8$
$(10043)_8$

Carry 11
$(3215)_6$
+ $(5243)_6$
$(12502)_6$

Carry 1 1
$(X7Y6)_8$
+ $(4X33)_8$
$(605Y)_8$
X=1, y=1

Carry 11
$(X1Y2)_8$
+ $(41XY)_8$
$(7X20)_8$
X=3, Y=6