COURSE CODE & NAME :- DCA1205, Digital Logic

SET – I

1.)

Question :- What is number system? Explain how to convert decimal number to binary number with example?

Answer :- **Number systems** are the technique to represent numbers in the computer system architecture, every value that you are saving or getting into/from computer memory has a defined number system.

Computer architecture supports following number systems.

- **Binary number system**
- **Octal number system**
- **Decimal number system**
- **Hexadecimal (hex) number system**

1) Binary Number System

A Binary number system has only two digits that are **0 and 1**. Every number (value) represents with 0 and 1 in this number system. The base of binary number system is 2, because it has only two digits.

2) Octal number system

Octal number system has only eight (8) digits from **0 to 7**. Every number (value) represents with 0,1,2,3,4,5,6 and 7 in this number system. The base of octal number system is 8, because it has only 8 digits.

3) Decimal number system

Decimal number system has only ten (10) digits from **0 to 9**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8 and 9 in this number system. The base of decimal number system is 10, because it has only 10 digits.

4) Hexadecimal number system

A Hexadecimal number system has sixteen (16) alphanumeric values from **0 to 9** and **A to F**. Every number (value) represents with 0,1,2,3,4,5,6, 7,8,9,A,B,C,D,E and F in this number system. The base of hexadecimal number system is 16, because it has 16 alphanumeric values. Here **A is 10, B is 11, C is 12**, **D is 13**, **E is 14** and **F is 15**.

Number system	Base	Used digits	Example	C language assignments
Binary	2	0,1	$(11110000)_2$	int val=0b11110000;
Octal	8	0,1,2,3,4,5,6,7	$(360)_8$	int val=0360;
Decimal	10	0,1,2,3,4,5,6,7,8,9,	$(240)_{10}$	int val=240;
Hexadecimal	16Used digits	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	$(F0)_{16}$	int val=0xF0;

Number System Conversions

There are three types of conversion:

- **Decimal Number System to Other Base**
  [for example: Decimal Number System to Binary Number System]
- **Other Base to Decimal Number System**
  [for example: Binary Number System to Decimal Number System]
- **Other Base to Other Base**
  [for example: Binary Number System to Hexadecimal Number System]

Decimal Number System to Other Base

To convert Number system from **Decimal Number System** to **Any Other Base** is quite easy; you have to follow just two steps:
A) Divide the Number (Decimal Number) by the base of target base system (in which you want to convert the number: Binary (2), octal (8) and Hexadecimal (16)).
B) Write the remainder from step 1 as a Least Signification Bit (LSB) to Step last as a Most Significant

Decimal to Binary Conversion

Decimal Number is : $(12345)_{10}$

2	12345	1	LSB
2	6172	0	
2	3086	0	
2	1543	1	
2	771	1	
2	385	1	
2	192	0	
2	96	0	
2	48	0	
2	24	0	
2	12	0	
2	6	0	
2	3	1	
	1	1	MSB

Result

Binary Number is :- $(11000000111001)_2$

Decimal to Octal Conversion

Decimal Number is : **(12345)**$_{10}$

Example 1

Decimal Number is : **(12345)**$_{10}$

Result

Hexadecimal Number is:-**(3039)**$_{16}$

16	12345
16	771
16	48
8	3

9	LSB
3	
0	
3	MSB

Example 2

Decimal Number is : **(725)**$_{10}$

Result

Hexadecimal Number is **(2D5)**$_{16}$

Convert
10, 11, 12, 13, 14, 15
to its equivalent...
A, B, C, D, E, F

16	725
16	45
	2

5	5	LSB
13	D	
2	2	MSB

Other Base System to Decimal Number Base

To convert Number System from **Any Other Base System** to **Decimal Number System,** you have to follow just three steps:

A) Determine the base value of source Number System (that you want to convert), and also determine the position of digits from LSB (first digit's position – 0, second digit's position – 1 and so on).

B) Multiply each digit with its corresponding multiplication of position value and Base of Source Number System's Base.

C) Add the resulted value in step-B.

Explanation regarding examples:

Below given exams contains the following rows:

A) Row 1 contains the **DIGITs** of number (that is going to be converted).

B) Row 2 contains the **POSITION** of each digit in the number system.

C) Row 3 contains the multiplication: **DIGIT* BASE^POSITION.**

D) Row 4 contains the calculated result of **step C.**

E) And then add each value of **step D,** resulted value is the Decimal Number.

Binary to Decimal Conversion

Binary Number is : **(11000000111001)**$_2$

1	1	0	0	0	0	0	0	1	1	1	0	0	1
13	12	11	10	9	8	7	6	5	4	3	2	1	0
$1x2^{13}$	$1x2^{12}$	$0x2^{11}$	$0x2^{10}$	$0x2^9$	$0x2^8$	$0x2^7$	$0x2^6$	$1x2^5$	$1x2^4$	$1x2^3$	$0x2^2$	$0x2^1$	$1x2^0$
8192	4096	0	0	0	0	0	0	32	16	8	0	0	1

=8192+4096+32+16+8+1

=12345

Octal to Decimal Conversion

Octal Number is : $(30071)_8$

3	0	0	7	1
4	3	2	1	0
$3*8^4$	$0*8^3$	$0*8^2$	$7*8^1$	$1*8^0$
12288	0	0	56	1

Result

=12288+0+0+56+1
=12345
Decimal Number is: $(12345)_{10}$

Hexadecimal to Decimal Conversion

Hexadecimal Number is : $(2D5)_{16}$

2	D (13)	5
2	1	0
$2*16^2$	$13*16^1$	$5*16^0$
512	208	5

Result

=512+208+5
=725
Decimal Number is: $(725)_{10}$

2.)

Question :- Define Boolean Algebra. Simplify the following expression using Boolean algebra?

a.) A+AB b.) AB +AB' c.) A'BC+AC d.) A'B+ABC'+ABC

Answer :- Boolean algebra is a type of mathematical operation that, unlike regular algebra, works with binary digits (bits): 0 and 1. While 1 represents true, 0 represents false. Computers can perform simple to extremely complex operations with the use of Boolean algebra. Boolean algebra and Boolean operations are the basis for computer logic. Unlike conventional mathematical operations – addition, subtraction, division and multiplication – the operations in Boolean algebra are different and limited in number. There are three operations: NOT, AND and OR. The NOT operation returns the opposite of the value that is provided to it. For example, 1 is the opposite of 0 and vice versa. So there are just two outcomes of the operation. Both the AND and OR operations take two digits and return 0 or 1 depending on the inputs. The AND operation returns 1 in case both the inputs are equal to 1. Else, it returns 0. The OR operation returns 1 only if either of the values given to it is 1. Else, it returns a value of 0. Boolean algebra is named for George Boole, a mathematician who first described it in 1847.

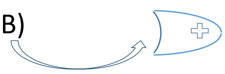
A.) **A+AB :-**

F = A+AB
Taking a common in the above expression, we get :
F = A (1+ B)
1+ Any variable is always 1 in Boolean Algebra
So , F= A

b.) **AB +AB' :-**

A +AB =A

A(1+B)



A(1) = A

c.) A'BC+AC :-

y = A'BC+AC

Y=C(A'B+A)

Y=C(A+A')(A+B)

Y=C(1)(A+B)

Y=C(A+B)

Y = CA+BC

d.) A'B+ABC'+ABC :-

Let X=(A'BC+ABC)=BC

Y=(AB'C+ABC)=AC

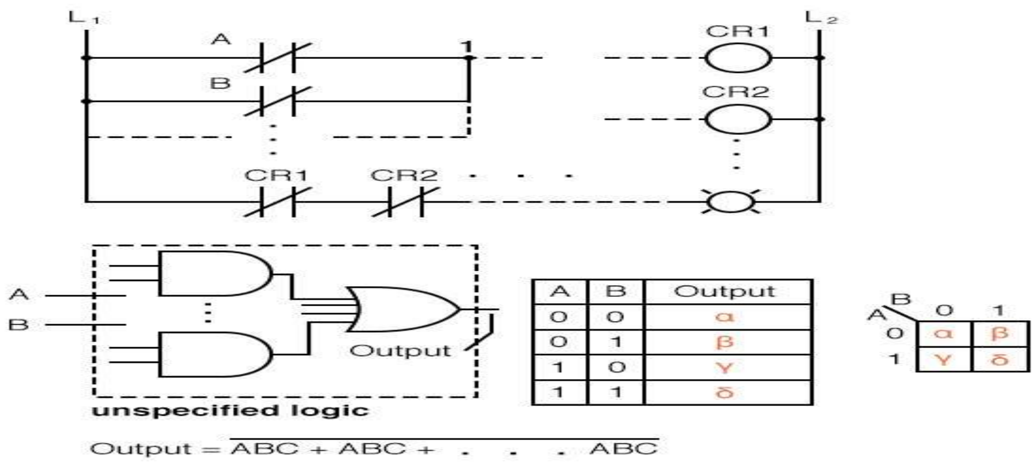
And Z=(ABC'+ABC)=AB

The given expression = X+Y+Z=AB+BC+CA

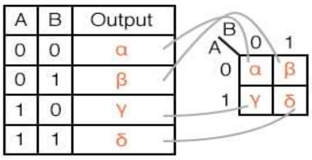
3.)

Question :- Define K-map? Simplify f (a, b, c, d) =∑m(0, 2, 4, 6, 7, 8, 9, 11, 12, 14).

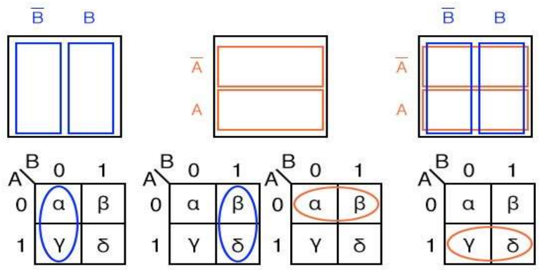
Answer :- Karnaugh map with the aid of Venn diagrams, let's put it to use. Karnaugh maps reduce logic functions more quickly and easily compared to Boolean algebra. By reduce we mean simplify, reducing the number of gates and inputs .We like to simplify logic to a lowest cost form to save costs by elimination of components. We define lowest cost as being the lowest number of gates with the lowest number of inputs per gate .Given a choice, most students do logic simplification with Karnaugh maps rather than Boolean algebra once they learn this tool.



We show five individual items above, which are just different ways of representing the same thing: an arbitrary 2-input digital logic function. First is relay ladder logic, then logic gates, a truth table, a Karnaugh map, and a Boolean equation. The point is that any of these are equivalent. Two inputs A and B can take on values of either 0 or 1, high or low, open or closed, True or False, as the case may be. There are 22 = 4 combinations of inputs producing an output. This is applicable to all five examples. These four outputs may be observed on a lamp in the relay ladder logic, on a logic probe on the gate diagram. These outputs may be recorded in the truth table, or in the Karnaugh map. Look at the Karnaugh map as being a rearranged truth table. The Output of the Boolean equation may be computed by the laws of Boolean algebra and transferred to the truth table or Karnaugh map. Which of the five equivalent logic descriptions should we use? The one which is most useful for the task to be accomplished.



The outputs of a truth table correspond on a one-to-one basis to Karnaugh map entries. Starting at the top of the truth table, the A=0, B=0 inputs produce an output α.Note that this same output α is found in the Karnaugh map at the A=0, B=0 cell address, upper left corner of K-map where the A=0 row and B=0 column intersect. The other truth table outputs β, χ, δ from inputs AB=01, 10, 11 are found at corresponding K-map locations. Below, we show the adjacent 2-cell regions in the 2-variable K-map with the aid of previous rectangular Venn diagram like Boolean regions.



Cells α and χ are adjacent in the K-map as ellipses in the left most K-map below. Referring to the previous truth table, this is not the case. There is another truth table entry (β) between them. Which brings us to the whole point of the organizing the K-map into a square array, cells with any Boolean variables in common need to be close to one another so as to present a pattern that jumps out at us. For cells α and χ they have the Boolean variable B' in common. We know this because B=0 (same as B') for the column above cells α and χ. Compare this to the square Venn diagram above the K-map. A similar line of reasoning shows that β and δ have Boolean B (B=1) in common. Then, α and β have Boolean A' (A=0) in common. Finally, χ and δ have Boolean A (A=1) in common. Compare the last two maps to the middle square Venn diagram. To summarize, we are looking for commonality of Boolean variables among cells. The Karnaugh map is organized so that we may see that commonality.

Simplify f (a, b, c, d) =Σm(0, 2, 4, 6, 7, 8, 9, 11, 12, 14).

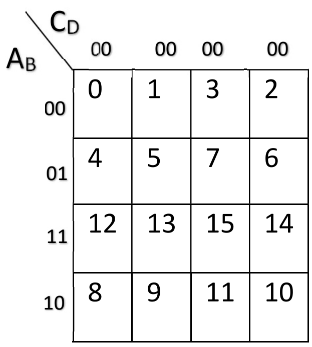
:- The number indicate K-map cell address locations. For maxterms this is location of **0**s, as show below . Product-OF-Sums solution is completed in the usual manner.

$Out = (A+B+\overline{C}+D)(A+\overline{B}+\overline{C}+D)+(\overline{A}+B+\overline{C}+D)(\overline{A}+\overline{B}+\overline{C}+D)$

$Out = (A+B+C+D)(A+B+C+D)(A+B+C+D)(A+B+C+D)$

$(\overline{A}+\overline{B}+\overline{C}+\overline{D})(\overline{A}+\overline{B}+\overline{C}+\overline{D})(\overline{A}+B+\overline{C}+D)$

F(A,B,C,D)= Πm(2,6,8,9,12,11,14)



AB \ CD	00	01	11	10
00	1	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	0	0	0

AB \ CD	00	00	00	00
00	1	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	0	0	0

$f(A,B,C,D)=(\overline{A}+B)(\overline{C}+D)$

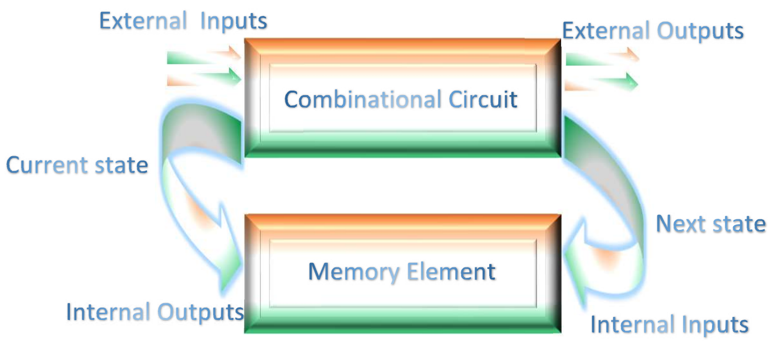
SET – I I

4.)

Question :- Define Sequential Circuits. Draw and explain the working of JK, S-R, D Flip-Flops?

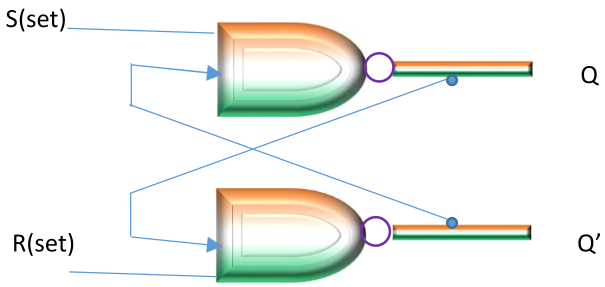
Answer :- A sequential circuit is the assimilation of a combinational logic circuit and a storage element. With the applied inputs to the combinational logic, the circuit outputs are derived. These sequential circuits deliver the output based on both the current and previously stored input variables. The derived output is passed on to the next clock cycle. Sequential circuits consist of memory devices to store binary data. This binary information describes the current state of the sequential circuit. As sequential circuits work along with the combinational circuit, there are two types of combinational logic inputs where those are

- External inputs where these are not monitored by the circuit
- Internal inputs are the function of previous state output.



As every digital and memory circuit is built based on the finite state machines, sequential circuits are implemented for the construction of these machines. So, these circuits hold more prominence in digital and electronics technology.

1/0 Ʊ



1/0 Ʊ

S | R	Q | Q’
1 0	0 1
1 1	0 1
0 1	1 0
1 1	1 0
0 0	1 1

(AFTER S=1 , R=0) (AFTER S=0, R=1)

Like the NOR Gate S-R flip flop, this one also has four states. They are

S=1, R=0—Q=0, Q'=1
This state is also called the SET state.

S=0, R=1—Q=1, Q'=0
This state is known as the RESET state.

In both the states you can see that the outputs are just compliments of each other and that the value of Q follows the compliment value of S

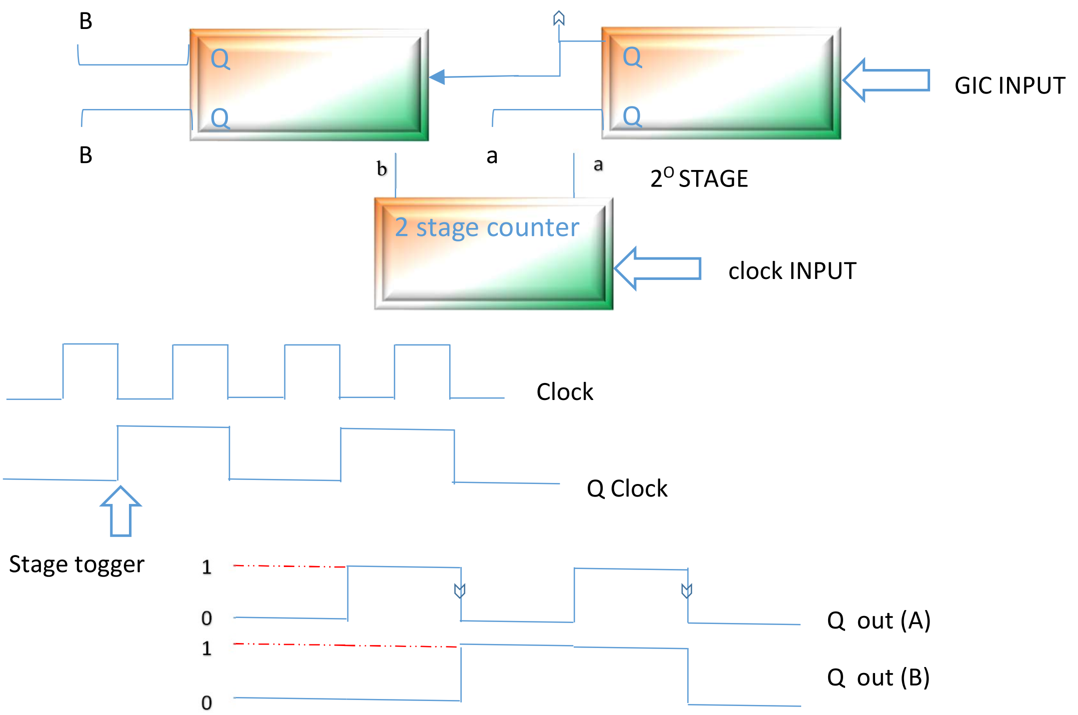
S=0, R=0—Q=1, & Q' =1 [Invalid]
If both the values of S and R are switched to 0 it is an invalid state because the values of both Q and Q' are 1. They are supposed to be compliments of each other. Normally, this state must be avoided.

S=1, R=1—Q & Q'= Remember
If both the values of S and R are switched to 1, then the circuit remembers the value of S and R in their previous state.

5.)

Question :- What is a Digital counter? Explain types of counters in digital circuit?

Answer :- Counter is a digital device and the output of the counter includes a predefined state based on the clock pulse applications. The output of the counter can be used to count the number of pulses. Generally, counters consist of a flip-flop arrangement which can be synchronous counter or asynchronous counter. In synchronous counter, only one clock i/p is given to all flip-flops, whereas in asynchronous counter, the o/p of the flip flop is the clock signal from the nearby one. The applications of the microcontroller need counting of exterior events such as exact internal time delay generation and the frequency of the pulse trains. These events are frequently used in digital systems & computers. Both these events can be executed by software techniques, but software loops for counting will not give the exact result slightly more important functions are not done. These problems can be rectified by timers and counters in the microcontrollers which are used as interrupts.



Types of Counters

Counters can be categorized into different types according to the way they are clocked. They are

- Asynchronous Counters
- Synchronous Counters
- Asynchronous Decade Counters
- Synchronous Decade Counters
- Asynchronous Up-Down Counters
- Synchronous Up-Down Counters

For better understanding of this type of counters, here we are discussing some of the counters.

Asynchronous Counters

The diagram of a 2-bit asynchronous counter is shown below. The exterior clock is connected to the clock i/p of the FF0 (first flip-flop) only. So, this FF changes the state at the decreasing edge of every clock pulse, but FF1 changes only when activated by the decreasing edge of the Q o/p of FF0. Because of the integral propagation delay through a FF, the change of the i/p clock pulse and a change of the Q o/p of FF0 can never occur at precisely the same time. So, the FF's cannot be activated concurrently, generating an asynchronous operation. Note that for ease, the changes of Q0,Q1 & CLK in the above diagram are shown as concurrent, even though this is an asynchronous counter. Actually, there is a small delay b/n the Q0, Q1 and CLK changes. Generally, all the CLEAR i/ps are connected together, so before counting starts then that a single pulse can clear all the FFs. The clock pulse fed into FF0 is rippled through the new counters after propagation delays, such as a ripple on the water, hence the term Ripple Counter. The circuit diagram of the two bit ripple counter includes four different states ,each one consisting with a count value. Likewise, a counter with n FFs can have 2N states. The number of states in a counter is called as its mod number. Therefore, a two-bit counter is a mod-4 counter.

5.)

Question :- Explain the design of an electronic tennis scoring system?

Answer :- tennis scoring system is a standard widespread method for scoring tennis matches, including pick-up games. Some tennis matches are played as part of a tournament, which may have various categories, such as singles and doubles. The great majority are organised as a single-elimination tournament, with competitors being eliminated after a single loss, and the overall winner being the last competitor without a loss. Optimally, such tournaments have a number of competitors equal to a power of two in order to fully fill out a single elimination bracket. In many professional and top-level amateur events, the brackets are seeded according to a recognised ranking system, in order to keep the best players in the field from facing each other until as late in the tournament as possible; additionally, if byes are necessary because of a less-than-full bracket, those byes in the first round are usually given to the highest-seeded competitors.

The score for the 2005 Mens Final of the SAP Open, San Jose. The winner was Andy Roddick and the runner-up was Cyril Saulnier.

A tennis match is composed of points, games, and sets. A set consists of a number of games (a minimum of six), which in turn each consist of points. A set is won by the first side to win 6 games, with a margin of at least 2 games over the other side (e.g. 6–3 or 7–5). If the set is tied at six games each, a tie-break is usually played to decide the set. A match is won when a player or a doubles team has won the majority of the prescribed number of sets. Matches employ either a best-of-three (first to two sets wins) or best-of-five (first to three sets wins) set format. The best-of-five set format is usually only used in the men's singles or doubles matches at Grand Slam and Davis Cup matches.