Name = Shivam Kumar Singh

Reg No = 22BSA 10166

(Q1)

Ans

* step (1) : convert the binary number to decimal.

$A = 10_{10} = 1010_2$

$B = 12_{10} = 1100_2$

* step 2 = Ensure Both Number Have the same Bit width.

* $A = 01010_2$
  $B = 01100_2$

* (3) Perform operation

(1) Addition : A + B

$$\begin{array}{r} 01010 \\ +\,01100 \\ \hline \boxed{10110} \end{array}$$ (result In binary)

Result $10110_2 = 22_{10}$

(ii) sub : A - B

(B)

using 2's complement for substraction:-

7) Find the 2's complement of B

* B:: 01100 → 10011.

* Add1 -:. 10011+1 = 10100.

* 2's complement of B = 10100

Add A and 2's complement of B

$$\begin{array}{r} 01010 \\ +\ 10100 \\ \hline \boxed{11110} \end{array}$$

* take 2's complement and take its ~~magnitude~~ magnitude.

* Invert 11110: 00001
  Add 1: 0001+1 = 10100

~~2's complement of B = 10100.~~

magnitude : (2_{10})

* Final Result $(A - B) = -(2_{10})$ Ans

(c)

III Ans

substraction B-A

using 2's complement for substraction.

1) Invert A: 01010 → 10101.

→ Add 1 : 10101 +1 = 10110.

→ 2's complement of A = 10110.

Add B and 2's complement of A

```
  01100
  10110
 100010
```

Dropping the overflow bit Result Is $00010_2 = 2_{10}$

IV multiplication A×B

A ⇒ 01010 B ⇒ 01100.

```
    01010
  × 01100
  -------
    00000   (0 from rightmost bit)
 +  01010   (shift left once)
 +  10100   (shift left twice)
   ------
   111100   result
```

= Result. $(111100_2 = 60_{10})$ Ans

Q2 Ans

* characteristics equation of D Flip Flop:

clk	D	Qn	Qn+1
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

→ Truth table

D \ Q	0	1
0	0	0
1	1	1

→ k- map

$$\boxed{Q_{n+1} = Q^+ = D}$$

→ equation

* characteristics equation of J K Flip Flop.

clk	J	K	Qn	Qn+1
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	1	0
↑	1	0	0	1
↑	1	0	1	1
↑	1	1	0	1
↑	1	1	1	0

J \ KQ	00	01	11	10
0	0	1	0	0
1	1	1	0	1

$$\boxed{Q_{n+1} = Q^+ = J\bar{Q} + \bar{K}Q}$$

Q2 Ans

* characteristics equation of D Flip Flop :-

clk	D	Qn	Qn+1
↑	0	0	0
↑	0	1	0
↑	1	0	1
↑	1	1	1

→ Truth table

D \ Q	0	1
0	0	0
1	1	1

→ k- map

$$\boxed{Q_{n+1} = Q^{+} = D}$$

→ equation

* charecteristics equation of J K Flip Flop.

clk	J	K	Qn	Qn+1
↑	0	0	0	0
↑	0	0	1	1
↑	0	1	0	0
↑	0	1	1	0
↑	1	0	0	1
↑	1	0	1	1
↑	1	1	0	1
↑	1	1	1	0

J \ KQ	00	01	11	10
0	0	1	0	0
1	1	1	0	1

$$\boxed{Q_{n+1} = Q^{+} = J\bar{Q} + \bar{K}Q}$$

* characteristic equation of T Flip-Flop :-

CLK	T	Q_n	Q_{n+1}
↑	0	0	0
↑	0	1	1
↑	1	0	1
↑	1	1	0

→ truth table

$\bar{T}Q$

T \ Q	0	1
0	0	1
1	1	0

→ k-map

$T\bar{Q}$

$$\boxed{Q_{n+1} = Q^+ = T\bar{Q} + \bar{T}Q = T \oplus Q}$$

equation.

3 b Ans

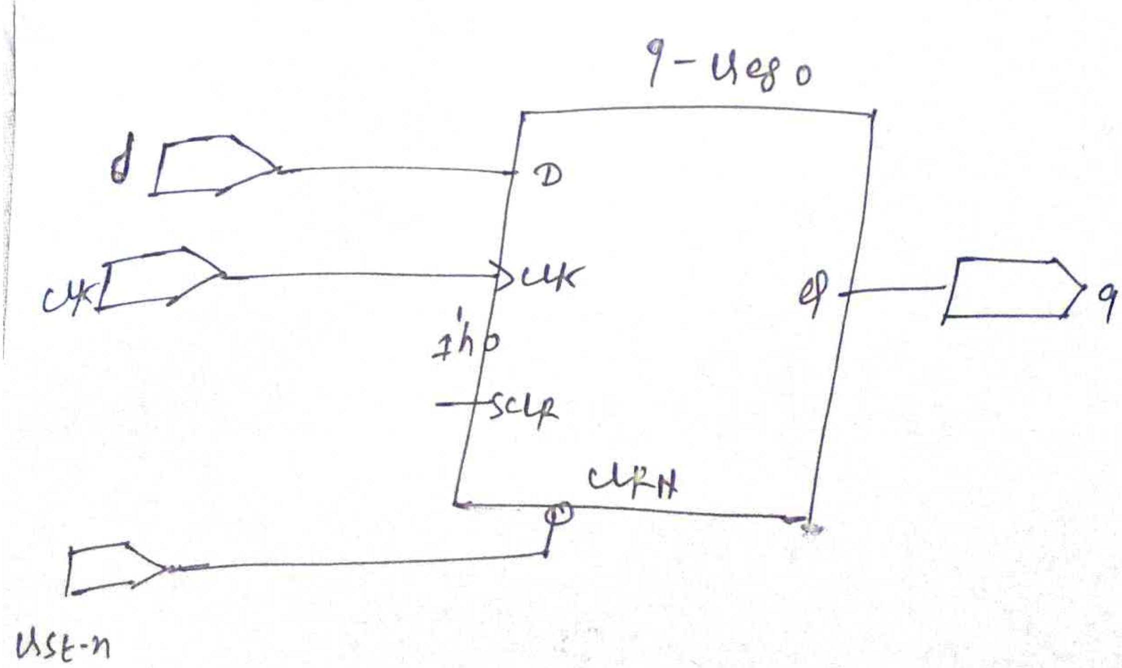
```
// D flip-flop

Module d-ff (
Out put reg q.
Input d,
Input rst-n.
Input clk
);
always @ (posedge clk or negedge rst-n)

begin
    if (.! rst_n)
     begin
        q <=0;

    end
    else
    begin
     q <=d;
   end
end
end module
```

* Explanation

—*—*—*—*

* The module is named d-ff.
* it has four ports.
* the module is named d-ff
* it has four ports:

* q (output): Respresent the out of of D flip-flop
* d (Input): Represent the Input data
* rst-n (Input): Represent the Active low reset signal
* clk (Input): Represent the clk signal



(Q3)

```
module d_ff_tb;
// Inputs
reg d;
reg rst_n;
reg clk;

// outputs

wire q ;

// Instantiate the d flip flop module

d_ff uut (
  .q (q),
  .d (d),
  .rst_n(rst_n),
  .clk (clk)
);

// clock generation

always #5 clk = ~clk;

// Test bench Initialize Inputs
  d =0;
  rst_n = 0;
  clk = 0;
```

// reset de-assertion after some time

#50 rst_n = 1;

// Apply Input d changes

#10 d = 1;
#20 d = 0;
#30 d = 1;

End Simulation

#100 $finish;

end

endmodule.

4 Ans

1. Binary conversion

—*—*—*—*

Dividend	Remainder
$100/2 = 50$	0
$50/2 = 25$	0
$25/2 = 12$	1
$12/2 = 6$	0
$6/2 = 3$	0
$3/2 = 1$	1
$1/2 = 0$	1

$$\boxed{100_{10} = 1100100_2}$$

2. Octal conversion

—*—*—*—*

8	100	4
8	12	4
8	1	1

144

Read the Reverse

$(100)_{10} = (144)_8$ Ans

(3) Hexa conversion

—*—*—*—*

$(100)_{10} = (64)_{16}$

Divide $(100)_{10}$ successively by 16 until the quotient is 0

100 / 16 = 6, remainder is 4

6 / 16 = 0, remainder is 6

show Hexa decimal of $(100)_{10} = (64)_{16}$ Ans

(4) BCD Conversion

—*—*—*

(*) write each decimal digit 100 as a 4 bit Binary

1 = 0001,

0 = 0000,

0 = 0000,

Bcd = 0001 0000 0000

Q5) Ans

Full Adder In Data flow model:

```
module fulladder(
 Input a,
 Input b,
 Input cIn.
 output s,

 output cout
                                   & → and
);

assign s = a^b^cin
assign cout = (a & b) | cin&(a^b);

end module
```

(II) Full Adder In Gate level mode

```
module fa(
Input a,
Input b,
Input cin,
out put cout
);
wire x1, x2, x3;
xor (x1, a, b);
and (x3, a, b);
xor (s, x, cin);
and (x2, x1, cin);
or (cout, x2, x3);
end module
```