ASSIGNMENT3

Name:T.B.P.Hariharan RollNo:2019102019

1.The Netlist for the circuit is shown below

```
.include TSMC_180nm.txt


.param LAMBDA = 0.09u

.param W = {2*LAMBDA}

.global gnd vdd


VDD vdd gnd 1.8


**INPUTS**

Vin1 A  gnd pulse(1.8 0 0ns 100ps 100ps 10ns 20ns 0)

Vin2 B  gnd pulse(1.8 0 0ns 100ps 100ps 20ns 40ns 0)

Vin3 S  gnd 1.8

Vin4 S_bar gnd 0


**INPUT INVERTER**

.subckt cmosinv out in hi lo


.param width_N = {W}

.param width_P = {2*W}


MN out in lo lo CMOSN W={width_N} L={2*LAMBDA} AS={5*width_N*LAMBDA}
PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}

MP out in hi hi CMOSP W={width_P} L={2*LAMBDA} AS={5*width_P*LAMBDA}
PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}

.ends cmosinv


**OUTPUT INVERTER**

.subckt varinv out in hi lo pmtr = 3


.param width_N = {pmtr*W}

.param width_P = {(6-pmtr)*W}


MN out in lo lo CMOSN W={width_N} L={2*LAMBDA} AS={5*width_N*LAMBDA}
PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}

MP out in hi hi CMOSP W={width_P} L={2*LAMBDA} AS={5*width_P*LAMBDA}
PS={10*LAMBDA+2*width_P} AD={5*width_P*LAMBDA} PD={10*LAMBDA+2*width_P}


.ends varinv


**INPUT INVERTER**

xI1 in A     vdd gnd cmosinv

xI2 in1 B     vdd gnd cmosinv


**OUTPUT INVERTER**

xI3 Y  out vdd gnd varinv pmtr = 3
```

```
**PASS TRANSISTORS**

M1 in  S out gnd CMOSN W={W} L={2*LAMBDA} AS={5*W*LAMBDA}
PS={10*LAMBDA+2*W} AD={5*W*LAMBDA} PD={10*LAMBDA+2*W}

M2 in1 S_bar out gnd CMOSN W={W} L={2*LAMBDA} AS={5*W*LAMBDA}
PS={10*LAMBDA+2*W} AD={5*W*LAMBDA} PD={10*LAMBDA+2*W}



**TRANSIENT ANALYSIS**
.tran 10ps 100ns



*Calculating delay from A to Y
.measure tran fall
+TRIG v(A) VAL = '0.5*1.8' FALL = 1
+TARG v(Y) VAL = '0.5*1.8' FALL = 1
.measure tran rise
+TRIG v(A) VAL = '0.5*1.8' RISE = 1
+TARG v(Y) VAL = '0.5*1.8' RISE = 1
.measure tran delay_AtoY param = (fall + rise)/2 goal = 0
```

```
.control
   run
   set hcopypscolor = 1
   set color0 = white
   set color1 = black

   set curplottitle = "tbphariharan 2019102019(a)"
   plot A
   hardcopy 1_A.eps A

   set curplottitle = "tbphariharan 2019102019(b)"
   plot B
   hardcopy 1_B.eps B

   set curplottitle = "tbphariharan 2019102019 output(S = 1)"
   plot Y
   hardcopy 1_Y.eps Y
```

```
   hardcopy 1_Y.eps Y


.endc
```

Wp = (6 – k)W , Wn = kw.

For Width(W) = 4 * LAMBDA(least possible value as per design rule checker) , LAMBDA = 0.09 micron .

The minimum average delay is observed to be $1.21463 * 10^{-10}$ seconds when K = 4,

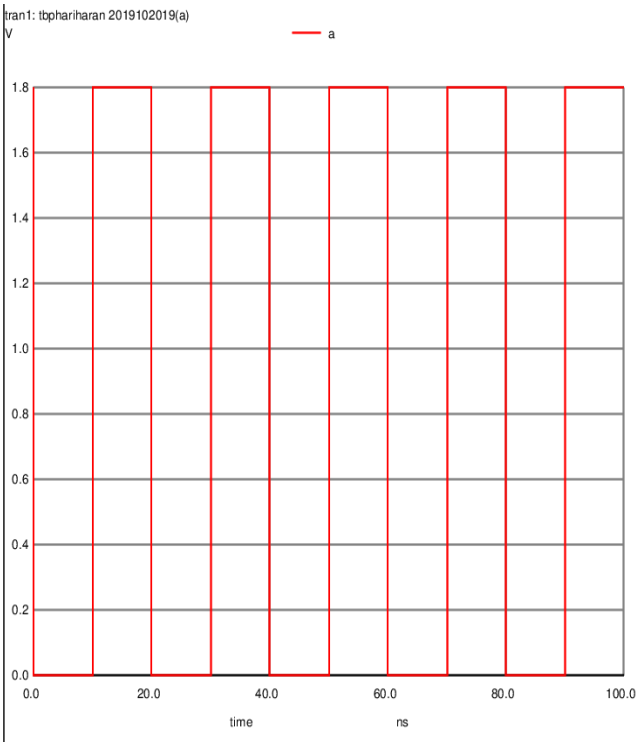
Wn = 4W,Wp = 2W.

Here We are finding delay for S = 1 we can find it for S = 0 but both give same result, it is sufficient to calculate for S = 1 or S = 0.

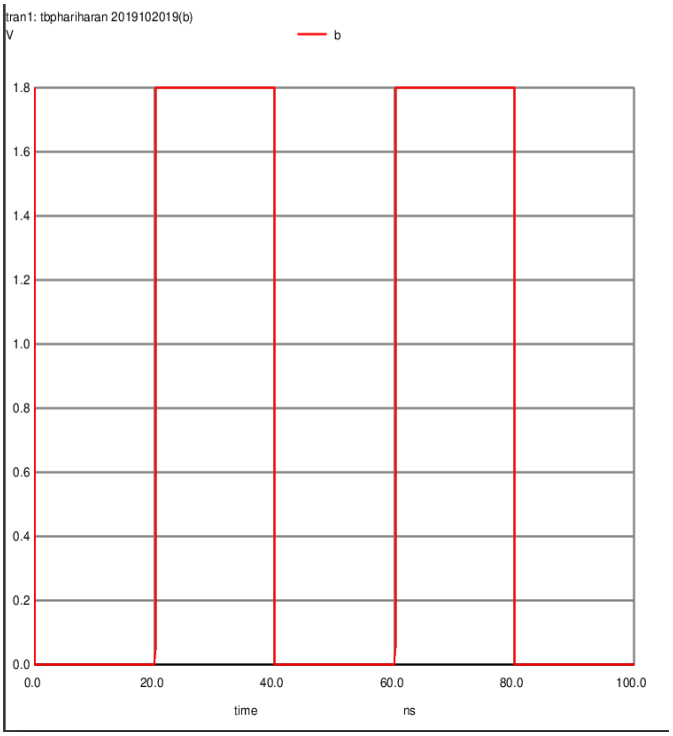
Simulation Results From NGSPICE.

```
Initial Transient Solution
--------------------------

Node                                   Voltage
----                                   -------
vdd                                        1.8
a                                          1.8
b                                          1.8
s                                          1.8
s_bar                                        0
in                                   1.128e-08
in1                                  1.128e-08
y                                          1.8
out                                  1.12794e-08
vin4#branch                                  0
vin3#branch                                  0
vin2#branch                                  0
vin1#branch                                  0
vdd#branch                          -3.07636e-11

 Reference value :  6.33425e-08
No. of Data Rows : 20071

  Measurements for Transient Analysis

fall               =  1.310454e-10 targ=  1.810454e-10 trig=  5.000000e-11
rise               =  1.118804e-10 targ=  1.026188e-08 trig=  1.015000e-08
delay_atoy         =  1.21463e-10
```

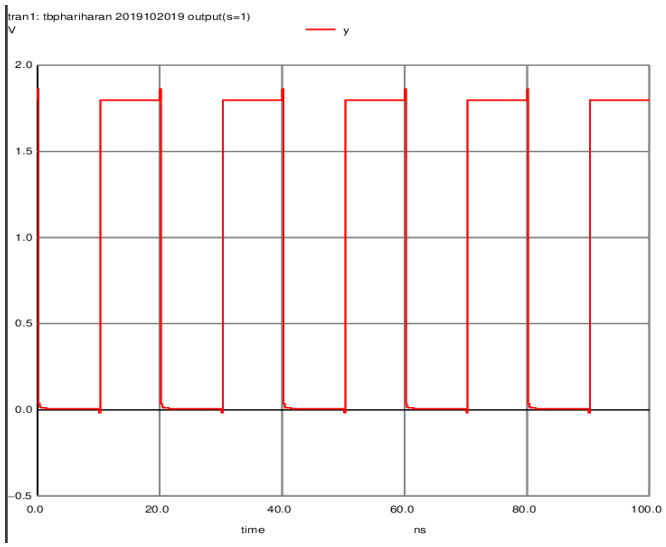
Plots:



Input a



Input b



Output Y

2(a).The Shannon Expansion and Circuit diagram are shown

$f(x_1,x_2,x_3,x_4) = x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4$

$f(x_1,x_2,x_3,x_4) = \bar{x}_1 f(1,x_2,x_3,x_4) + \bar{x}_1 f(0,x_2,x_3,x_4)$

$f(1,x_2,x_3,x_4) = x_2 f(1,1,x_3,x_4) + \bar{x}_2 f(1,0,x_3,x_4)$

$f(0,x_2,x_3,x_4) = x_2 f(0,1,x_3,x_4) + \bar{x}_2 f(0,0,x_3,x_4)$

$f(1,1,x_3,x_4) = x_3 f(1,1,1,x_4) + \bar{x}_3 f(1,1,0,x_4)$

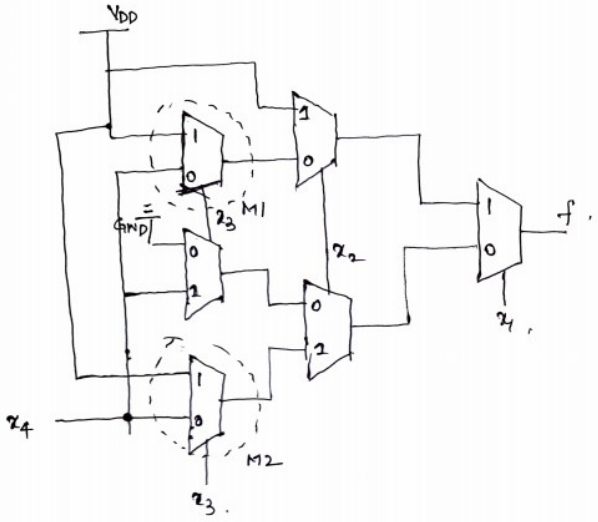
$f(1,0,x_3,x_4) = x_3 f(1,0,1,x_4) + \bar{x}_3 f(1,0,0,x_4)$

$f(0,1,x_3,x_4) = x_3 f(0,1,1,x_4) + \bar{x}_3 f(0,1,0,x_4)$

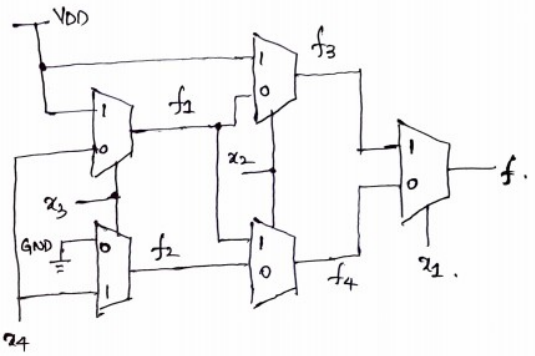
$f(0,0,x_3,x_4) = x_3 f(0,0,1,x_4) + \bar{x}_3 f(0,0,0,x_4)$

but, $f(0,0,0,x_4) = 0$, $f(0,0,1,x_4) = x_4$, $f(0,1,0,x_4) = x_4$, $f(0,1,1,x_4) = 1$, $f(1,0,0,x_4) = x_4$,

$f(1,0,1,x_4) = 1$, $f(1,1,0,x_4) = 1$, $f(1,1,1,x_4) = 1$.



FINAL CIRCUIT DIAGRAM:



observation: The mux M1 & M2 have same input and select lines.
one of the mux can be removed.

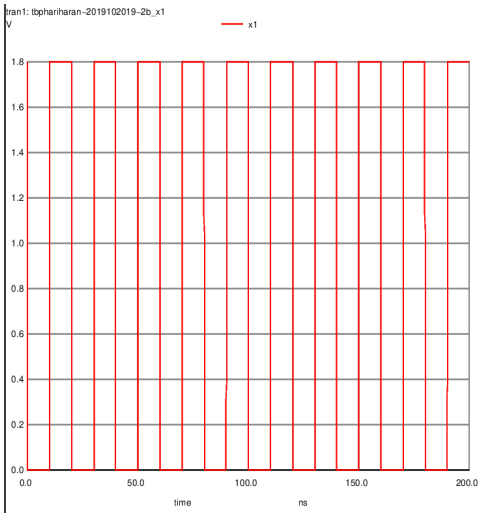
2(b).The netlist For the above circuit is shown

```
.include TSMC_180nm.txt


.param LAMBDA = 0.09u
.param Width_N = {4*LAMBDA}
.global gnd vdd


VDD vdd gnd 1.8



Vin1 x1 gnd pulse(1.8 0 0ns 100ps 100ps 10ns 20ns 0)
Vin1_bar x1_bar gnd pulse(0 1.8 0ns 100ps 100ps 10ns 20ns 0)
Vin2 x2 gnd pulse(1.8 0 0ns 100ps 100ps 20ns 40ns 0)
Vin2_bar x2_bar gnd pulse(0 1.8 0ns 100ps 100ps 20ns 40ns 0)
Vin3 x3 gnd pulse(1.8 0 0ns 100ps 100ps 40ns 80ns 0)
Vin3_bar x3_bar gnd pulse(0 1.8 0ns 100ps 100ps 40ns 80ns 0)
Vin4 x4 gnd pulse(1.8 0 0ns 100ps 100ps 80ns 200ns 0)
Vin4_bar x4_bar gnd pulse(0 1.8 0ns 100ps 100ps 80ns 160ns 0)



.subckt mux A B sel sel_bar out


M1 A sel out gnd CMOSN W={width_N} L={2*LAMBDA} AS={5*width_N*LAMBDA}
PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}

M2 B sel_bar out gnd CMOSN W={width_N} L={2*LAMBDA} AS={5*width_N*LAMBDA}
PS={10*LAMBDA+2*width_N} AD={5*width_N*LAMBDA} PD={10*LAMBDA+2*width_N}


.ends mux



xM1 vdd x4  x3 x3_bar f1 mux
xM2 x4  gnd x3 x3_bar f2 mux
xM3 vdd f1  x2 x2_bar f3 mux
xM4 f1  f2  x2 x2_bar f4 mux
xM5 f3  f4  x1 x1_bar f  mux


.tran 5ps 200ns


.control


   set hcopypscolor = 1
   set color0 = white
   set color1 = black


   run


   set curplottitle = "tbphariharan-2019102019-2b_x1"
   plot x1
```

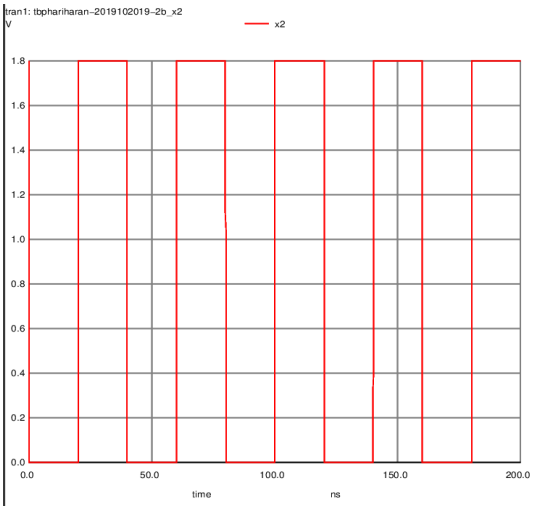
```
hardcopy x1.eps x1

set curplottitle = "tbphariharan-2019102019-2b_x2"
plot x2
hardcopy x2.eps x2

set curplottitle = "tbphariharan-2019102019-2b_x3"
plot x3
hardcopy x3.eps x3

set curplottitle = "tbphariharan-2019102019-2b_x4"
plot x4
hardcopy x4.eps x4

set curplottitle = "tbphariharan-2019102019-2b_f"
plot f
hardcopy f.eps f

.endc
```

Plots :

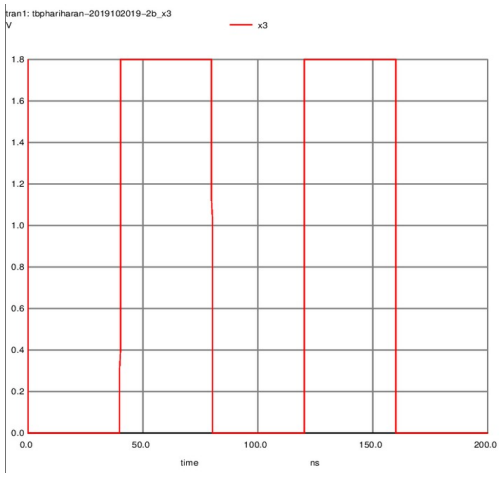
Inputs :



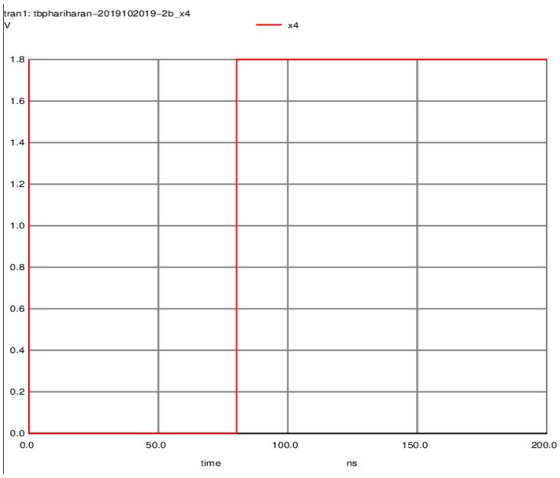
x1



x2

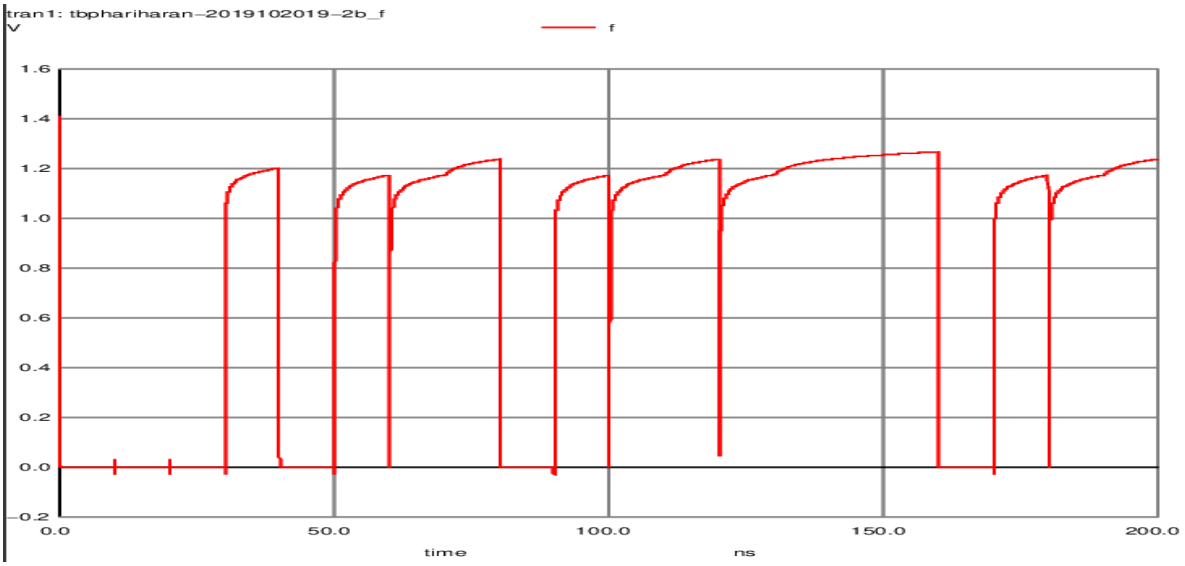


x3



x4

Outputs:



$$f = x_1x_2 + x_1x_3+x_1x_4+x_2x_3+x_2x_4+x_3x_4$$

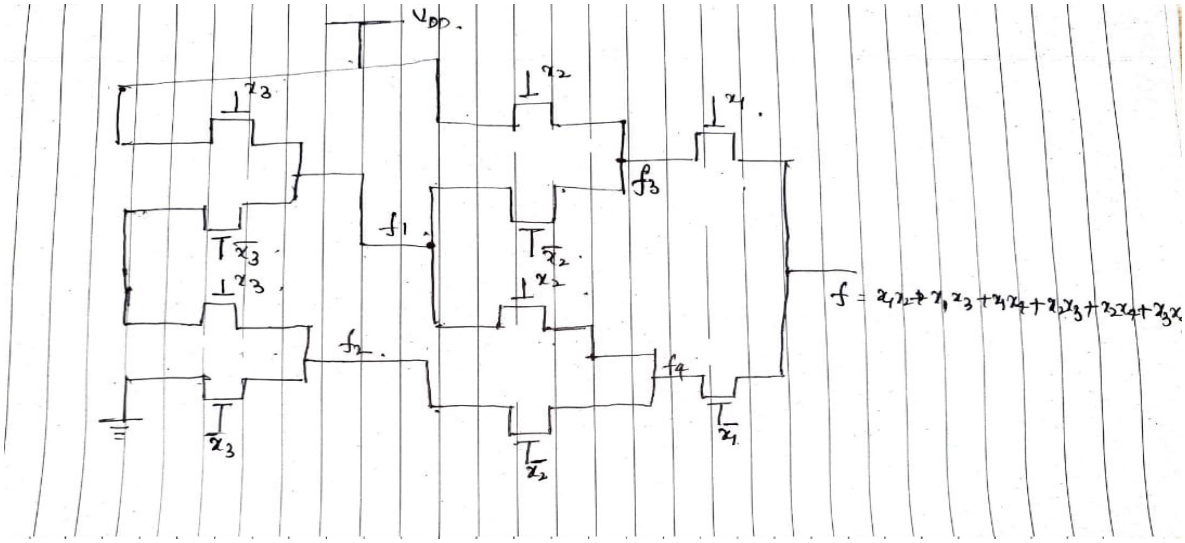
2(c).

$f(x_1 , x_2 , x_3 , x_4) = x_1x_2 + x_1x_3+x_1x_4+x_2x_3+x_2x_4+x_3x_4$

Observation : If any two of the inputs are **one** the output will be 1.So to get output **one** all the inputs should not be necessarily **one.**

If any three of the inputs are **zero** the output will be 0.So to get output as zero all the inputs should not be necessarily **zero.**

Circuit Diagram Shown Below:



The Paths will be ,

Charging Path:

The Transistors having x2 and x1 as gates if x2 = 1 , x1 = 1 , then f will be **1**.

Discharging Path:

The transistors having $\overline{x_3}, \overline{x_2}, \overline{x_1}$ as gates if x3 = 0 , x2 = 0 , x1 = 0 , then f will be **0.**

```
.measure tran rise
+TRIG V(f) VAL = '0.1*1.2' RISE = 1
+TARG V(f) VAL = '0.9*1.2' RISE = 1

.measure tran fall
+TRIG V(f) VAL = '0.9*1.2' FALL = 1
+TARG V(f) VAL = '0.1*1.2' FALL = 1
.control
```

By adding the above snippet to the netlist in 2(a) we can find the rise and fall time.

From Graph the we get the maximum voltage to be 1.2(approximately).

The rise and fall time are shown below.

```
  Measurements for Transient Analysis

rise                =  4.032076e-10 targ=  3.055795e-08 trig=  3.015474e-08
fall                =  3.845555e-11 targ=  1.422150e-10 trig=  1.037594e-10
```

Initial State Final State

The input combinations for charging (0 , 0 , 0 , 0) = (1 , 1 , 0 , 0)

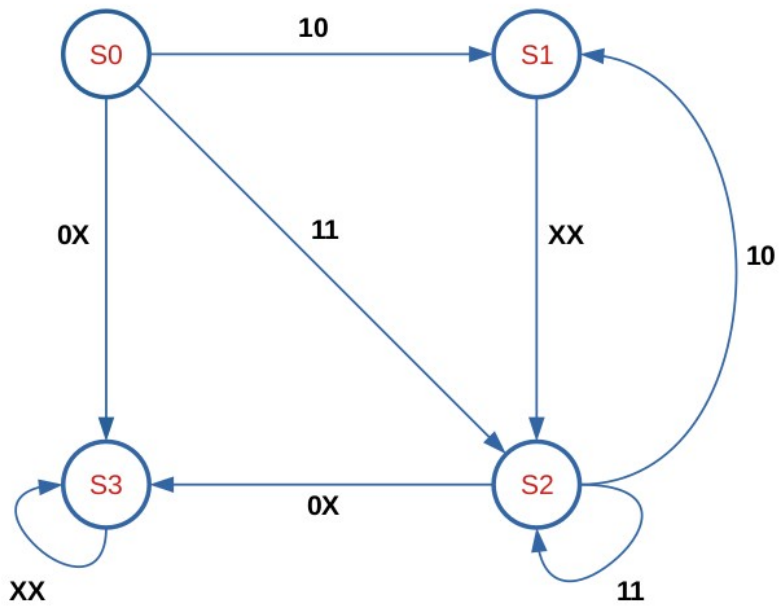
The input combinations for discharging (1 ,1 ,0 , 0) = (1 , 0 , 0 , 0)

2(d).The buffers have lower charging resistance and increased Drive capacity using this advantage this can decrease the rise time the buffer can be connected after the transistors having gates x1 and x2 , thus it decreases the rise time.

3.(a) state table is shown below

	Inputs ($XR > 0$) ($XR \geq YR$)			Outputs				
State	0–	10	11	*Subtract*	*Swap*	*Select XY*	*Load XR*	*Load YR*
S_0 (Begin)	S_3	S_1	S_2	0	0	1	1	1
S_1 (Swap)	S_2	S_2	S_2	0	1	0	1	1
S_2 (Subtract)	S_3	S_1	S_2	1	0	0	1	0
S_3 (End)	S_3	S_3	S_3	0	0	0	0	0

The state diagram for the above state table is shown below.



X represents 0 or 1

S0 represents begin state.
S1 represents swap state.
S2 represents subract state.
S3 represents end state(The state where GCD has been computed or it has exited the loop).

3(b).Clearly there are 4 states we require $[\log_2 4] = 2$, D-FlipFlops are required.

We assign binary codes to the states,

$S_0 = 00$, $S_1 = 01$, $S_2 = 10$, $S_3 = 11$.

Let the output of First and Second D-FlipFlop be D_1 , D_0 respectively.

D_1, D_0 be the present state of the two FlipFlops.

D_1^+, D_0^+ be the next state of the two FlipFlops.

The Control Unit's Combinational Logic is derived from the excitation table.

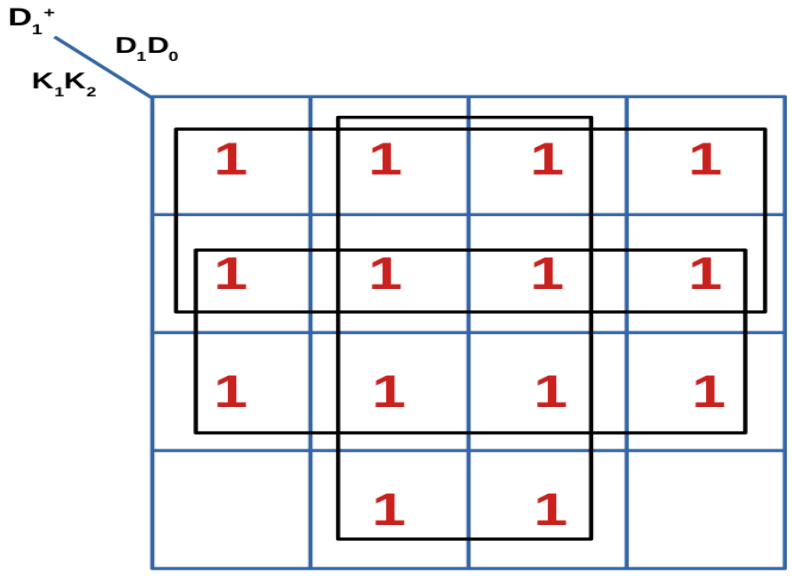
The excitation table is shown below

INPUTS		PRESENT STATE		NEXT STATE		OUTPUTS				
XR > 0	XR ≥ YR	D_1	D_0	D_1^+	D_0^+	SUBRACT	SWAP	SELECT XY	LOAD XR	LOAD YR
0	X	0	0	1	1	0	0	1	1	1
0	X	0	1	1	0	0	1	0	1	1
0	X	1	0	1	1	1	0	0	1	0
0	X	1	1	1	1	0	0	0	0	0
1	0	0	0	0	1	0	0	1	1	1
1	0	0	1	1	0	0	1	0	1	1
1	0	1	0	0	1	1	0	0	1	0
1	0	1	1	1	1	0	0	0	0	0
1	1	0	0	1	0	0	0	1	1	1
1	1	0	1	1	0	0	1	0	1	1
1	1	1	0	1	0	1	0	0	1	0
1	1	1	1	1	1	0	0	0	0	0

X represents 0 or 1

From the excitation table the control unit's logic can be derived from KMAP

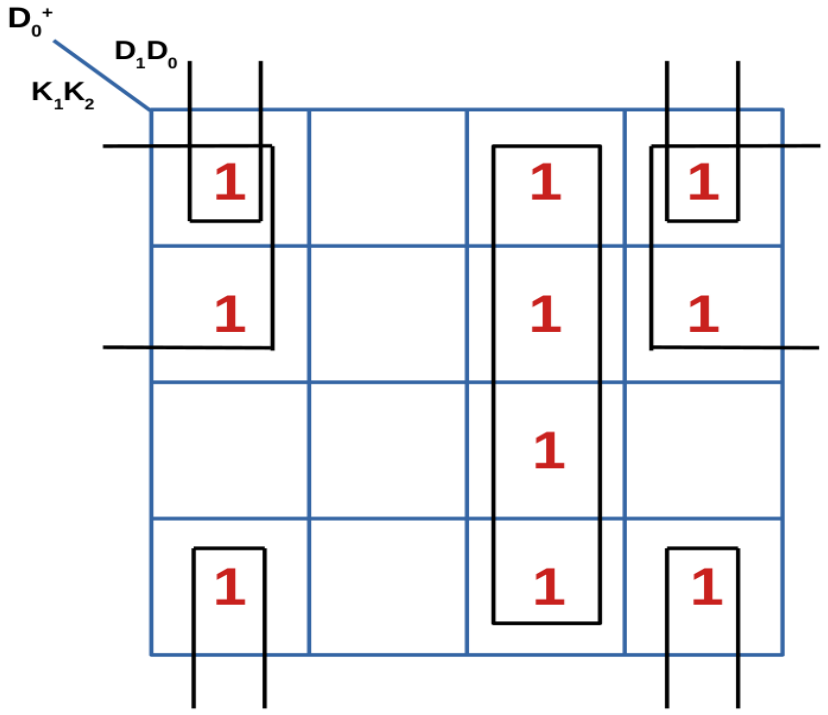
The KMAP for D_1^+ is shown below



From the above KMAP,

$$D_1^+ = \overline{(XR>0)} + (XR \geq YR) + D_0$$

The KMAP for D_0^+ is shown below



From the above KMAP,

$$D_0^+ = D_1 D_0 + \overline{(XR \geq YR)} \cdot \overline{(D_0)} + \overline{(XR>0)} \cdot \overline{(D_0)}$$

From the excitation table,

Subract = $D_1 \cdot \overline{D_0}$

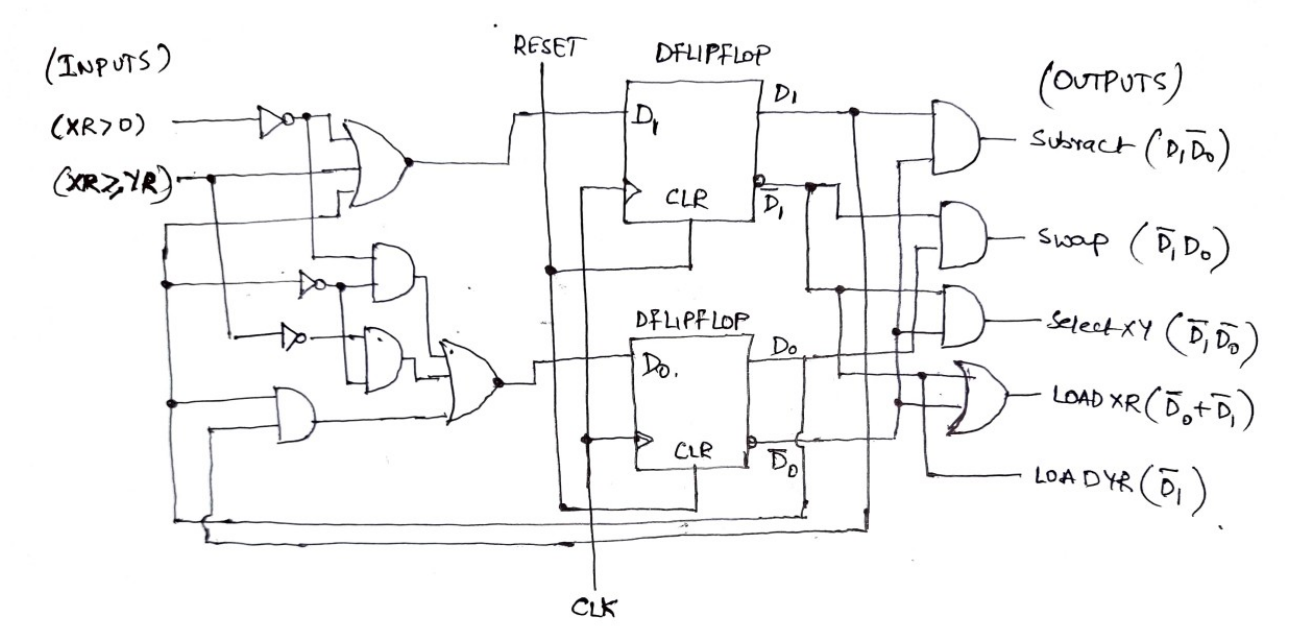
Swap = $\overline{D_1} \cdot D_0$

SelectXY(SXY) = $\overline{D_1} \cdot \overline{D_0}$

Load XR(LXR) = $\overline{D_1} + \overline{D_0}$

Load YR(LYR) = $\overline{D_1}$

The circuit diagram for the **CONTROL UNIT** is shown below



CONTROL UNIT OF GCD PROCESSOR.

3(c).The Above Designed Control Unit is implemented in Verilog HDL.

In this implementation the Four states are encoded with binary codes

S0(begin) = “00”,S1(swap) = “01”,S2(Subract) = “10”, S3(End) = “11”.

In this implementation we are not using One Hot Encoding.

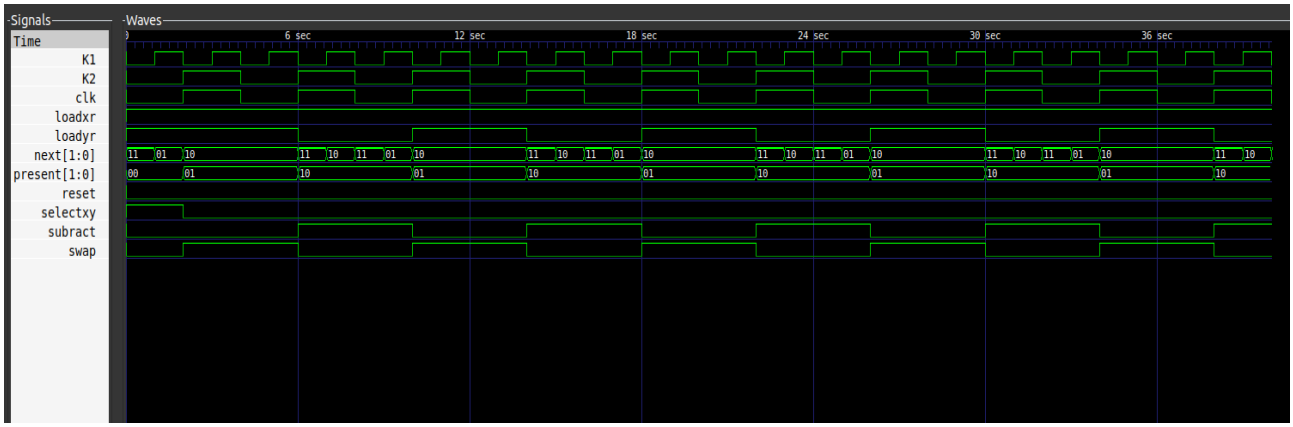
Implementation In verilog :

```
module ControlUnit(
    input K1 , K2 , clk , reset,
    output reg subract = 0 ,
    output reg swap = 0 ,
    output reg selectxy = 0 ,
    output loadxr = 0 ,
    output loadyr = 0,
    output reg [1:0] present = 2'b00 ,
    output reg [1:0] next = 2'b00
);

    parameter S0 = 2'b00,
              S1 = 2'b01,
              S2 = 2'b10,
              S3 = 2'b11;

    always @(posedge clk) begin
        if(reset == 1) begin
            present <= S0;
        end
```

```
    always @(present , K1 , K2) begin
        next = S0; //default state
        case(present)
            S0 : begin
                if(K1 == 0)
                    next = S3;
                else if((K1 == 1) & (K2 == 1))
                    next = S2;
                else
                    next = S1;
            end
            S1 : begin
                next = S2;
            end
            S2 : begin
                if(K1 == 0)
                    next = S3;
                else if((K1 == 1) & (K2 == 1))
                    next = S2;
                else
                    next = S1;
            end
            S3 : begin
```

```
            end
        endcase
    end
    always @(*) begin
        subract = (present == S2) ? 1 : 0;
        swap = (present == S1) ? 1 : 0;
        selectxy = (present == S0) ? 1 : 0;
        loadxr = ((present == S0) | (present == S1) | (present == S2)) ? 1
: 0;
        loadyr = ((present == S0) | (present == S1)) ? 1 : 0;
    end


endmodule
```

Simulation Results From GTKWave:

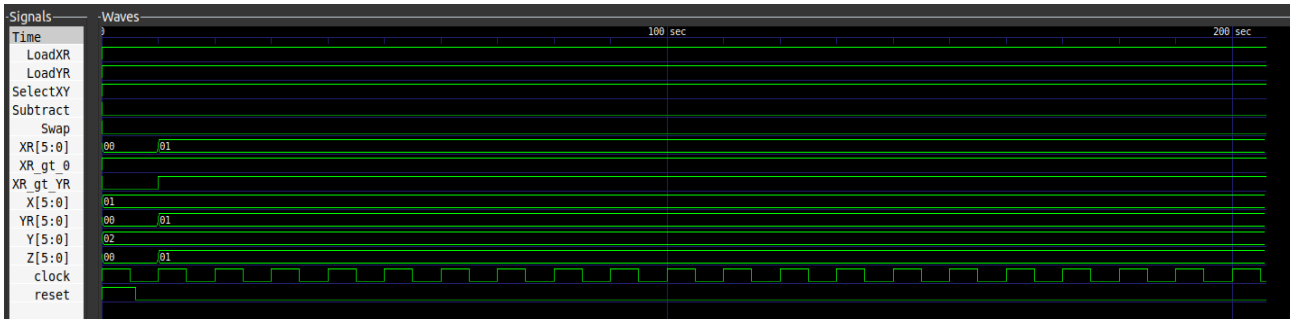


Where K1 = (XR > 0) , K2 = (XR ≥ YR) , clk is Clock Signal , loadxr is loading the register xr , loadyr is loading the register yr , next[1:0] represents next state of the Control Unit , present[1:0] represents present state of the control unit , reset is reset signal , selectxy is selecting the values x and y. Subract signal, Swap is Swap signal.

3(d).The behavioral description is shown below

```
module DataPath (
   output wire [5:0] Out,
   output wire K1 , K2, //K1 = (XR > 0) , K2 = (XR >= YR)
   output reg[5:0] AR , BR,
   input wire [5:0] A , B,
   input wire reset , subract , swap , selectxy , loadxr,
   loadyr , clk
);
   wire [5:0] reg_AR_in , reg_BR_in , reg_BR_out , reg_AR_out , sub_out;
   assign reg_AR_in = subract ? sub_out : (swap ? reg_BR_out :
          selectxy ? A : reg_AR_out);
   assign reg_BR_in = swap ? reg_AR_out : (selectxy ? B : reg_BR_out);
   always @(posedge clk) begin
       if(reset & loadxr)
           AR <= 0;
       else if (loadxr)
           AR <= reg_AR_in;
   end
   always @(posedge clk) begin
       if(reset & loadyr)
           BR <= 0;
       else if (loadxr)
           BR <= reg_AR_in;
   end
   assign reg_AR_out = AR,
          reg_BR_out = BR,
          Out = reg_BR_out;
   assign sub_out = reg_AR_out - reg_BR_out;
   assign K1 = (reg_AR_out > 0) ? 1 : 0;
   assign K2 = (reg_AR_out >= reg_BR_out) ? 1 : 0;
endmodule
```

Simulation Results From GTKWave



The behavioral description of the DataPath Unit is Shown Above.

It is simulated against X = 1 , Y = 2.

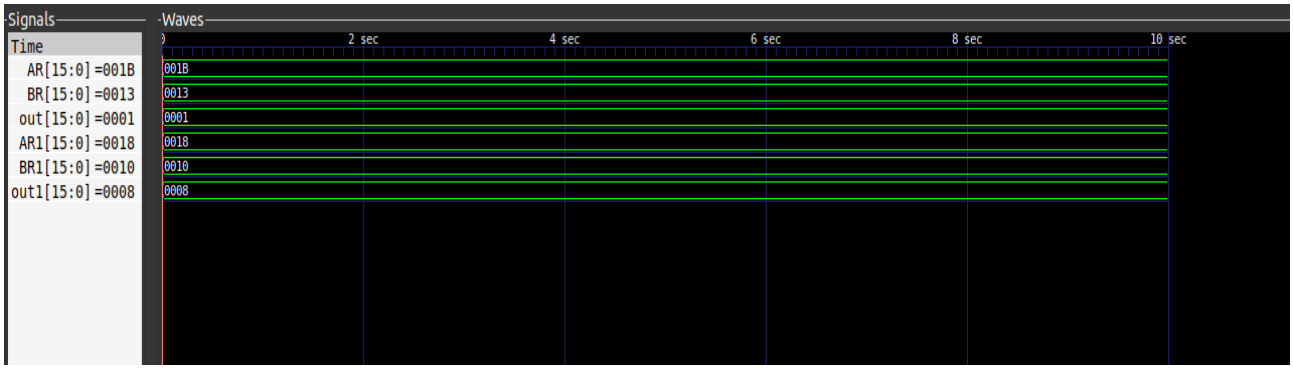
“XR_gt_0” represents XR > 0 , XR_gt_YR represents XR > YR. Z is the output of the GCD processor.

3(e).The below description computes GCD and the test bench is shown below.

```
module GCD_behavioral #(parameter W = 16)
(
   input [W - 1:0] AR , BR,
   output[W - 1:0] out
);
   reg [W - 1:0] A , B , out , swap;
   always @(*) begin
       A = AR; B = BR;
       while (B != 0) begin
           if(A < B) begin
               swap = A;
               A = B;
               B = swap;
           end
              A = A - B;
       end
       out = A;
   end

endmodule
```

```
`include "behavioralgcd.v"
module gcd_behavioraltb();
    reg [15:0] AR , BR , AR1 , BR1;
    wire [15:0] out , out1;

    GCD_behavioral#(16) gcd_unit(AR , BR , out);
    initial begin
        $dumpvars(0 , gcd_behavioraltb);
        $dumpfile("dump.vcd");
    end
    initial begin
        AR = 27; BR = 19;
        #10;
        $display("gcd(27 , 19) = %x",out);
        $finish;
    end
    GCD_behavioral#(16) gcd_unit1(AR1 , BR1 , out1);
    initial begin
        AR1 = 24; BR1 = 16;
        #10;
        $display("gcd(24 , 16) = %x",out1);
        $finish;
    end
```

Simulation Results From GTKWave



‘AR’ , ‘BR’ , ‘AR1’ , ‘BR1’ , ‘out’ , ‘out1’ are all 16 bit registers(It is a default value but it can be changed to either 32 or 64-bit with the help of parameter W).

‘AR’ represents the number 27 , ‘BR’ represents the number 19 , ‘out’ represents gcd(27 , 19) = 1, clearly the output of ‘out’ signal is 1.

‘AR1’ represents the number 24 , ‘BR1’ represents the number 16 , ‘out1’ represents gcd(24,16) = 8 , clearly the output of ‘out1’ is 8.