

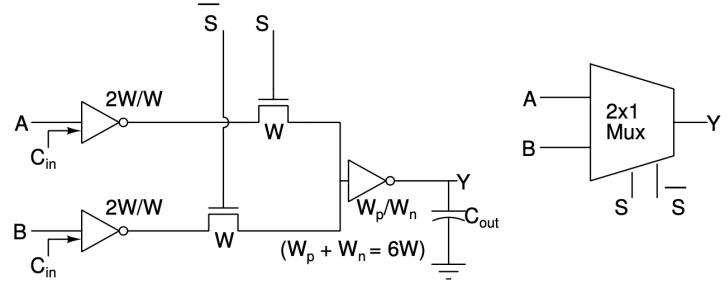
# VLSI Assignment #3 - Tanmay Pathak

2018102023

## Question 1)

Values, constraints and assumptions for the question

- Electrical Effort = 2 ( $C_{OUT}/C_{IN} = 2$ )
- Length of each transistor  $L = 0.18\mu\text{m}$
- $C_{IN} = 100\text{fF}$ ;  $C_{OUT} = 200\text{fF}$
- $W = 6\lambda$  where  $\lambda=0.09\mu\text{m}$
- $W_n + W_p = 6W$
- $W_n = k\lambda$
- $W_p = (36 - K)\lambda$



**AIM: To find an ideal value such that the overall delay is minimised.**

The netlist used for the question is given below

```
NETLIST FOR Q1 - TANMAY PATHAK - 2018102023

.include TSMC_180nm.txt
.param SUPPLY=1.8
.param LAMBDA=0.09u
.param K=30
.param width_N={6*LAMBDA}
.param width_P={12*LAMBDA}
.param width_N2={K*LAMBDA}
.param width_P2={(36-K)*LAMBDA}

.global gnd vdd
.global gnd vdd

Vdd      vdd      gnd      'SUPPLY'
* vin_a a 0 pulse 0 1.8 0ns 100ps 100ps 4.9ns 10ns
vin_a a 0 pulse 0 1.8 40ns 100ps 100ps 40ns 80ns
* vin_b b 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 10ns
vin_b b 0 pulse 0 1.8 0ns 100ps 100ps 4.9ns 10ns
vin_s s 0 pulse 0 1.8 50ns 100ps 100ps 50ns 100ns
* vin_b b 0 pulse 0 1.8 40ns 100ps 100ps 40ns 80ns

.subckt inv yi xi vdd gnd
M1 yi xi gnd 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 yi xi vdd vdd 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 inv

.subckt inv2 yi xi vdd gnd
M1 yi xi gnd gnd CMOSN W={width_N2} L={2*LAMBDA} AS={5*width_N2*LAMBDA} PS={10*LAMBDA+2*width_N2} AD={5*width_N2*LAMBDA} PD={10*LAMBDA+2*width_N2}
M2 yi xi vdd vdd CMOSP W={width_P2} L={2*LAMBDA} AS={5*width_P2*LAMBDA} PS={10*LAMBDA+2*width_P2} AD={5*width_P2*LAMBDA} PD={10*LAMBDA+2*width_P2}
.ends inv

X5 s_bar s vdd gnd inv

* Main circuit starts
X0 y1 a vdd gnd inv
X1 y2 b vdd gnd inv

M3 y1 s y3 gnd CMOSN W={6*LAMBDA} L={2*LAMBDA} AS={5*6*LAMBDA*LAMBDA} PS={10*LAMBDA+2*6*LAMBDA} AD={5*6*LAMBDA*LAMBDA} PD={10*LAMBDA+2*6*LAMBDA}
M4 y2 s_bar y3 gnd CMOSN W={6*LAMBDA} L={2*LAMBDA} AS={5*6*LAMBDA*LAMBDA} PS={10*LAMBDA+2*6*LAMBDA} AD={5*6*LAMBDA*LAMBDA} PD={10*LAMBDA+2*6*LAMBDA}

X2 y4 y3 vdd gnd inv2

C1 a 0 100f
C2 b 0 100f
Cout y4 0 200f

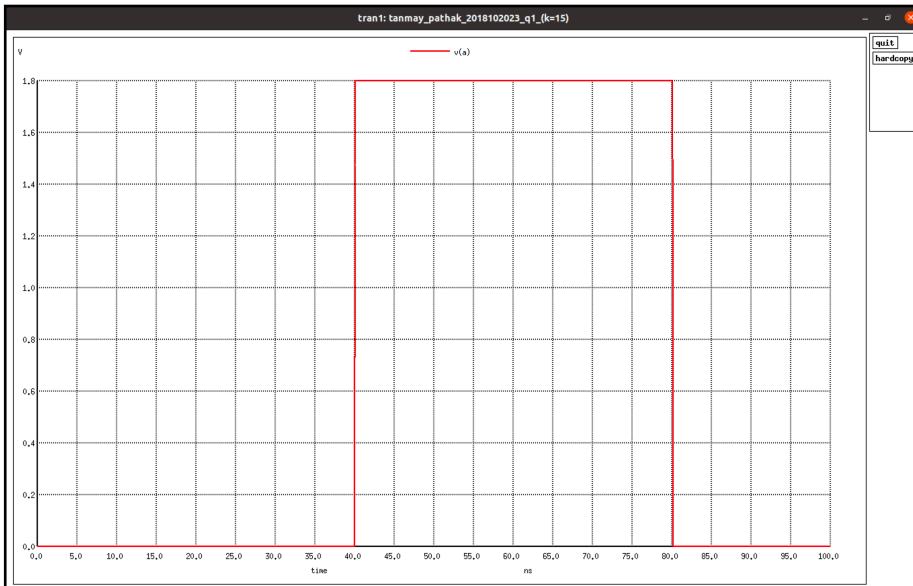
.tran 0.1n 100n

.measure tran rise trig v(y4) val=0.1 td=0 rise=2 targ v(y4) val=1.62 td=0 rise=2
.measure tran fall trig v(y4) val=1.62 td=0 fall=6 targ v(y4) val=0.1 td=0 fall=6

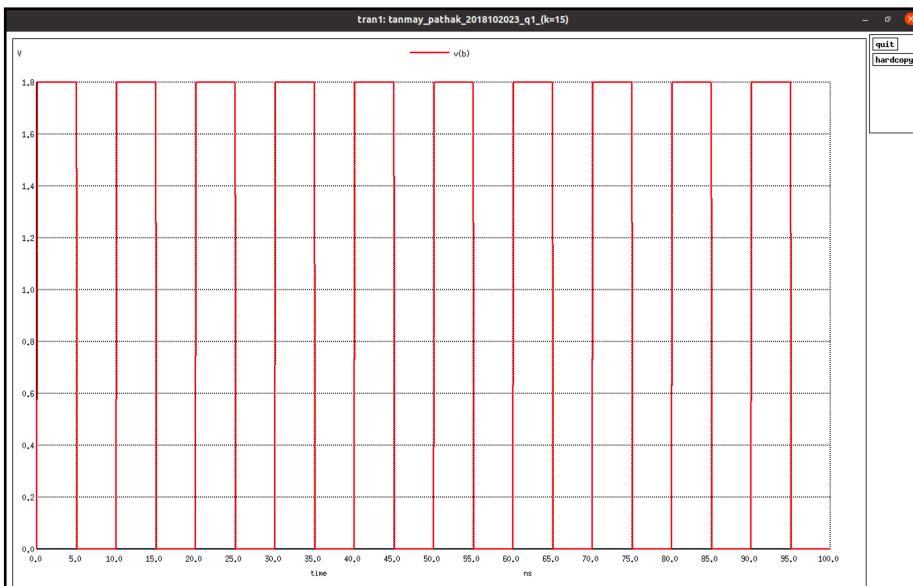
.control
set hcopypscolor = 1
set color0=white
set color1=black

run
set curplottitle= tanmay_pathak_2018102023_q1_(k=30)
plot v(a)
plot v(b)
plot v(s)
plot v(s_bar)
plot v(y4)
.endc
```

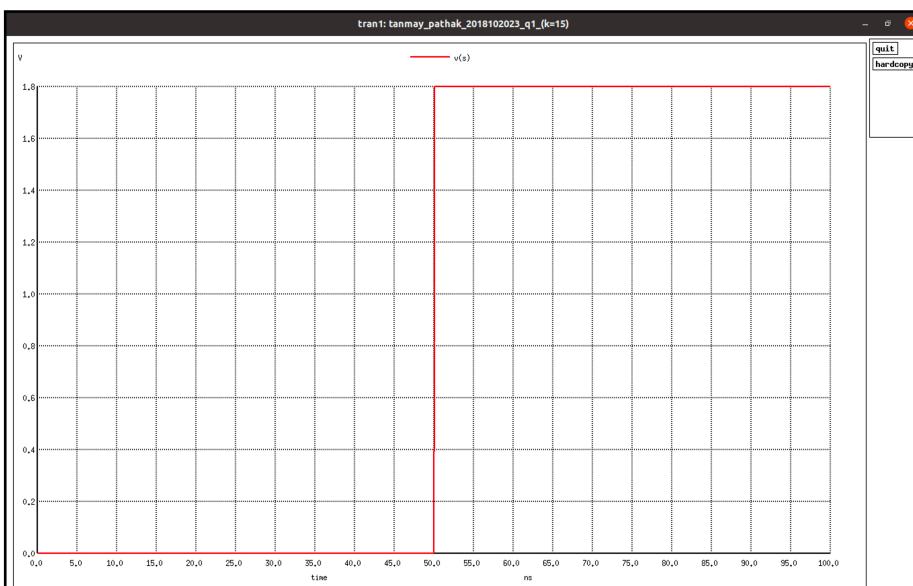
## OUTPUT: Graphs for k=15



Input waveform at A

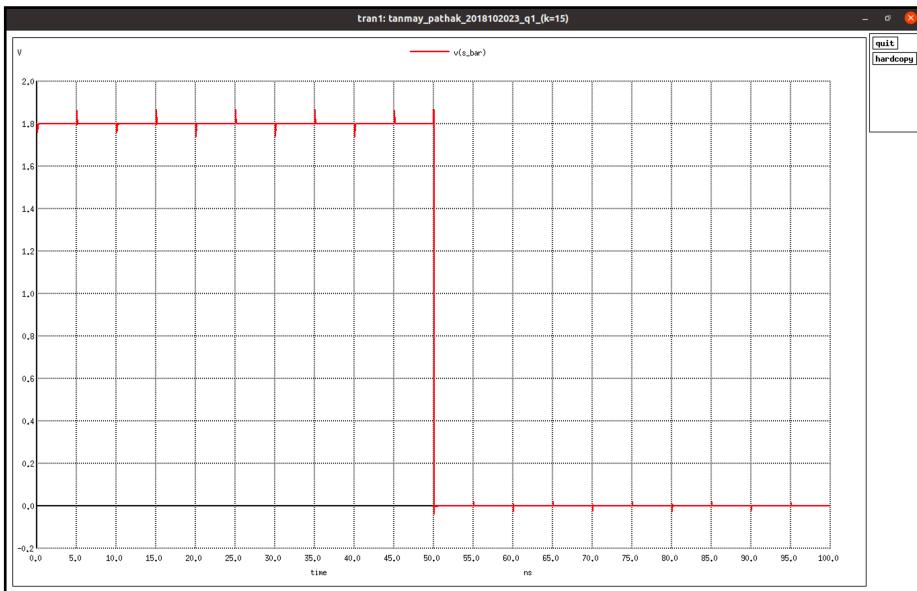


Input waveform at B

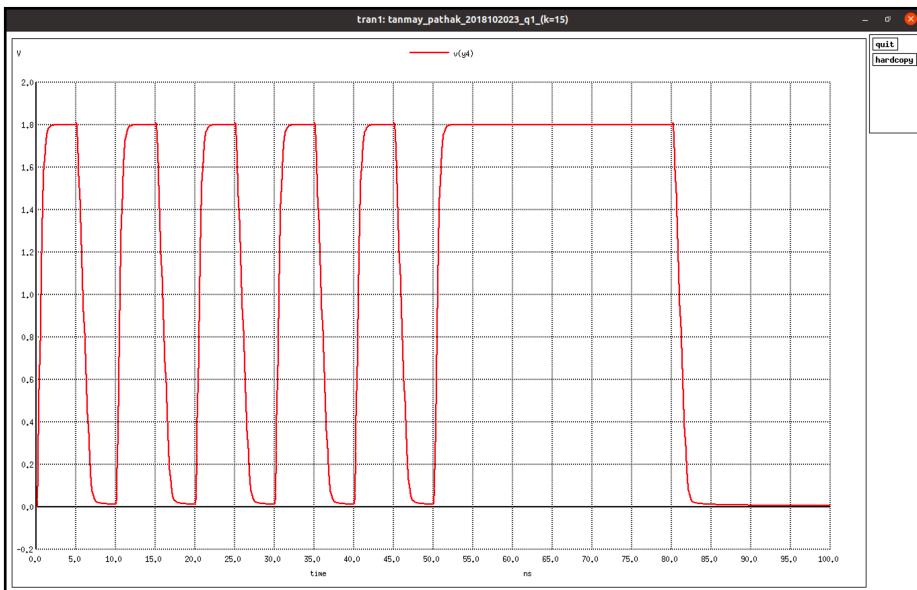


Selection line for the multiplexer

S is zero for the first 50ns. Thus the output will be B. Moreover, S is set to one for 50-100ns. Thus the output will be A

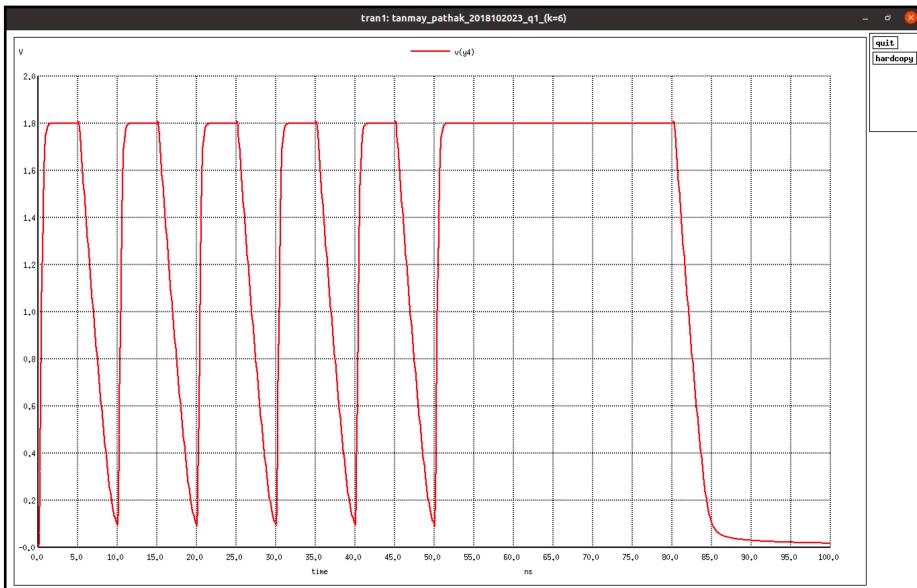


Compliment of S ( $S'$ )

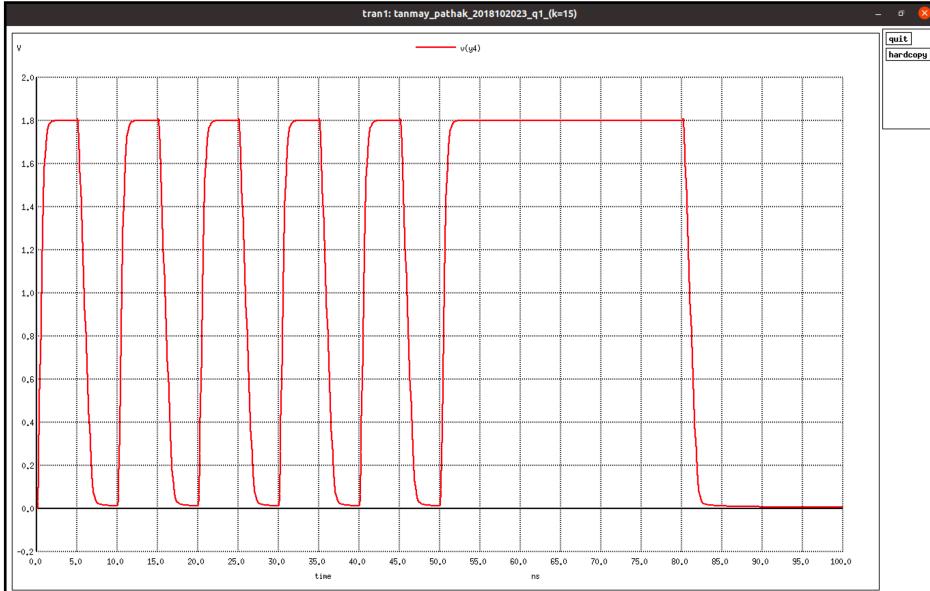


Output Waveform

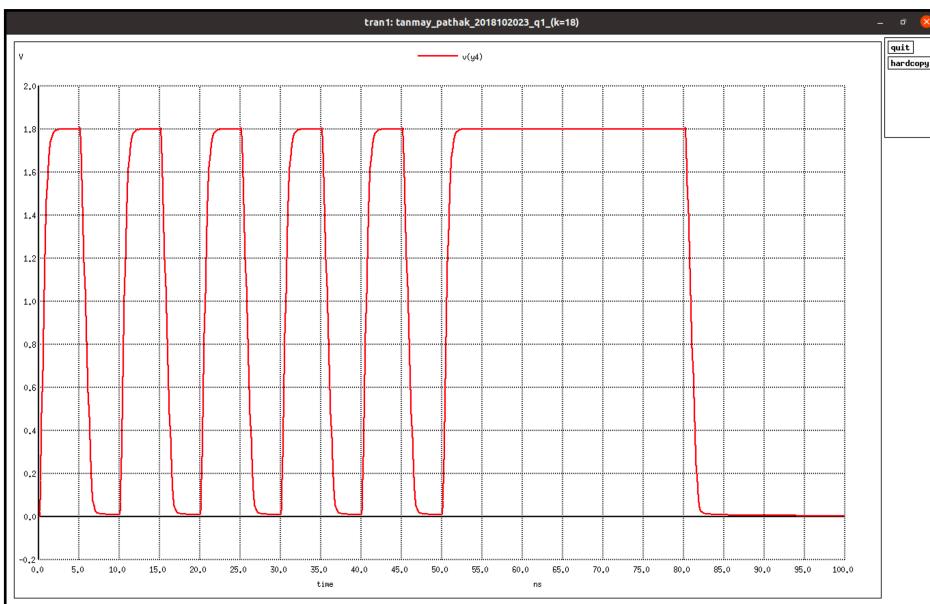
## OUTPUT: Graphs for different values of K



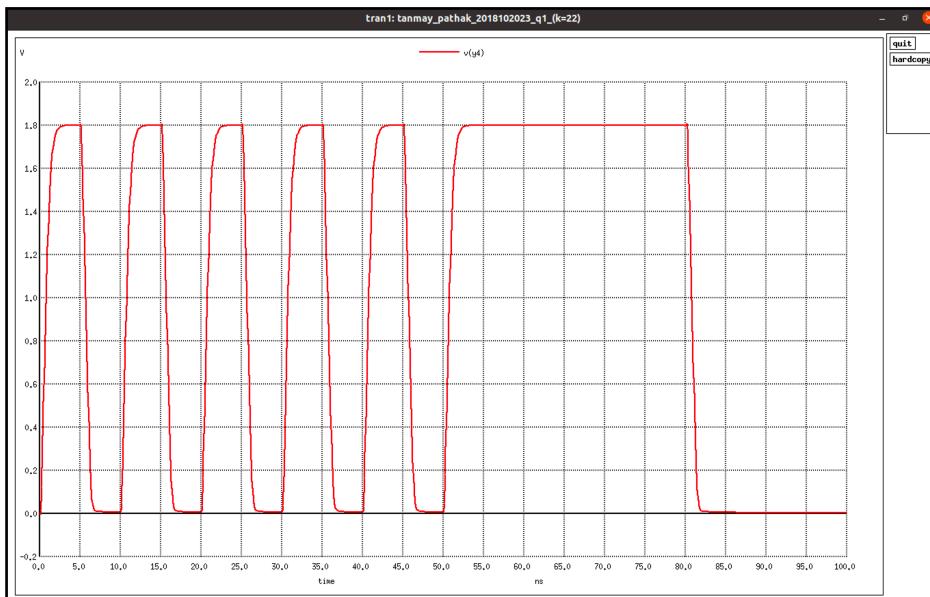
Output Waveform ( $k=6$ )



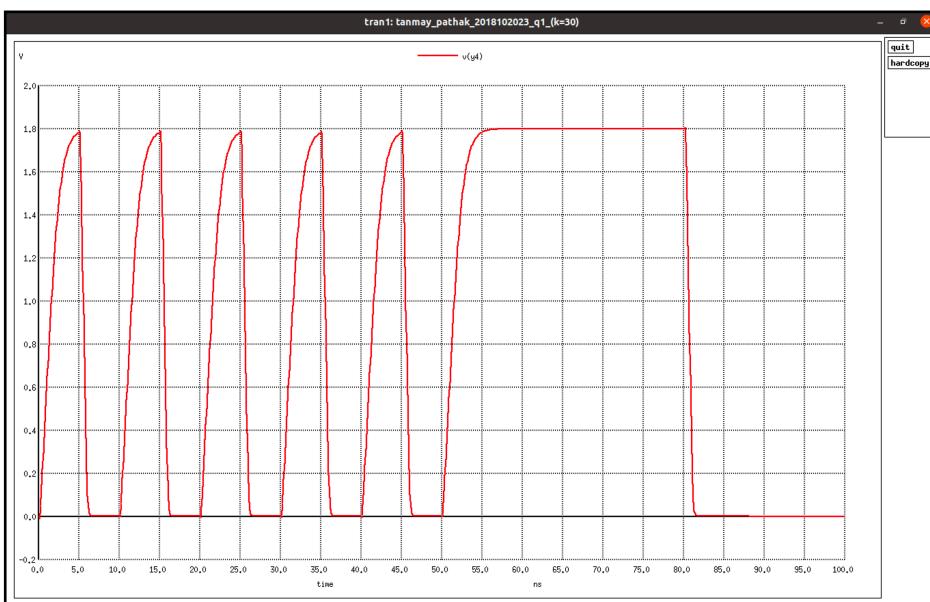
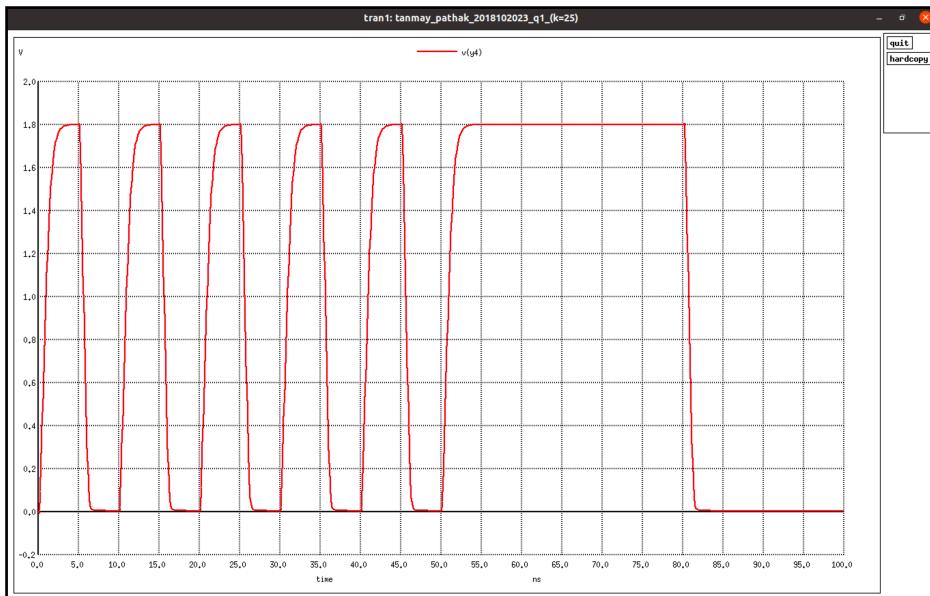
Output Waveform (k=15)



Output Waveform (k=18)



Output Waveform (k=22)



### Delay observed whilst altering transistor widths

Value of K	Average rise time (ns)	Average fall time (ns)	Average delay (ns)
6	0.63	4.4	2.56
15	0.84	1.6	1.22
18	0.97	1.38	1.175
19	1.02	1.31	1.165
22	1.24	1.13	1.185
25	1.56	1	1.28
30	2.77	0.839	1.805

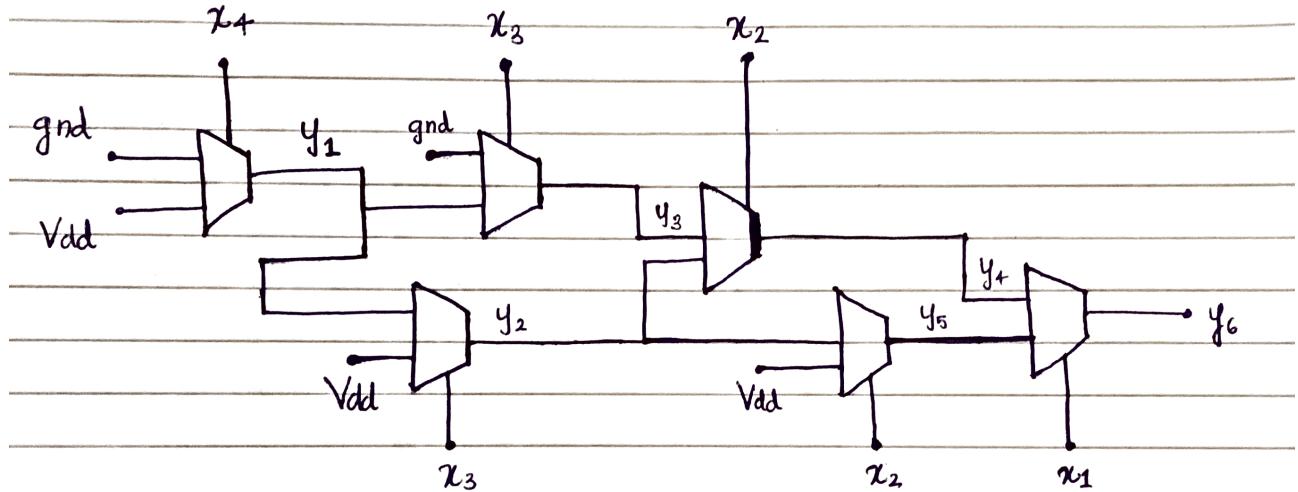
**RESULT:** We observe that the average delay is minimum for k=19. Thus the average delay observed is 1.165ns and hence the values used are  $W_n = 19\lambda$  and  $W_p = 17\lambda$ .

## Question 2 - (a)

The following equation is the logic function which is to be implemented using 2x1 multiplexor.

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

We use Shannon's theorem to simplify the function to use 2x1 MUXs. We use 6 (2x1 MUXs). Inputs  $x_1, x_2, x_3$  &  $x_4$  are given as select lines for the MUXes



## Question 2 - (b)

The netlist used for the question is given below

```
NETLIST FOR Q2 - TANMAY PATHAK - 2018102023

.include TSMC_180nm.txt
.param SUPPLY=1.8
.param LAMBDA=0.09
.param width_P={12*LAMBDA}
.param width_N={6*LAMBDA}
.global gnd vdd

Vdd vdd gnd 'SUPPLY'
vin_x1 x1 0 pulse 0 1.8 0ns 100ps 100ps 4.9ns 10ns
vin_x2 x2 0 pulse 0 1.8 0ns 100ps 100ps 9.9ns 20ns
vin_x3 x3 0 0
vin_x4 x4 0 0

.subckt mux y x x_bar one zero
M1 one x y 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 zero x_bar y 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

.subckt inv yi xi vdd gnd
M1 yi xi gnd 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 yi xi vdd gnd 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 inv

X0 x1_bar x1 vdd gnd inv
X1 x2_bar x2 vdd gnd inv
X2 x3_bar x3 vdd gnd inv
X3 x4_bar x4 vdd gnd inv

X4 y1 x4 x4_bar vdd gnd mux
X5 y2 x3 x3_bar vdd y1 mux
X6 y3 x3 x3_bar y1 gnd mux
X7 y4 x2 x2_bar y2 y3 mux
X8 y5 x2 x2_bar vdd y2 mux
X9 y6 x1 x1_bar y5 y4 mux

CL y6 0 5f

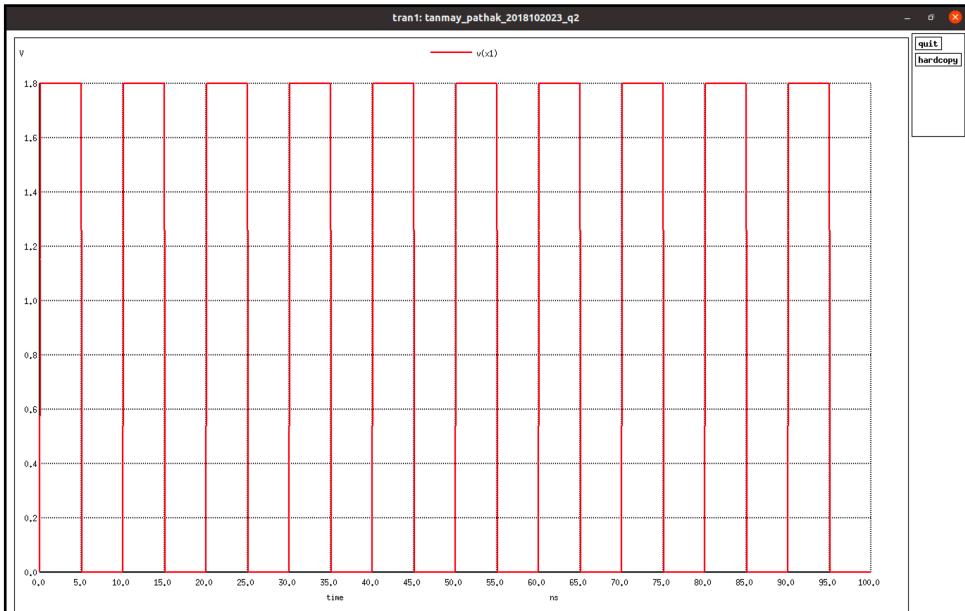
* transient analysis
.tran 0.1n 100n

* finding delay
.measure tran rise trig v(y6) val=0.01 td=0 rise=1 targ v(y6) val=1.13 td=0 rise=1
.measure tran fall trig v(y6) val=1.13 td=0 fall=1 targ v(y6) val=0.01 td=0 fall=1

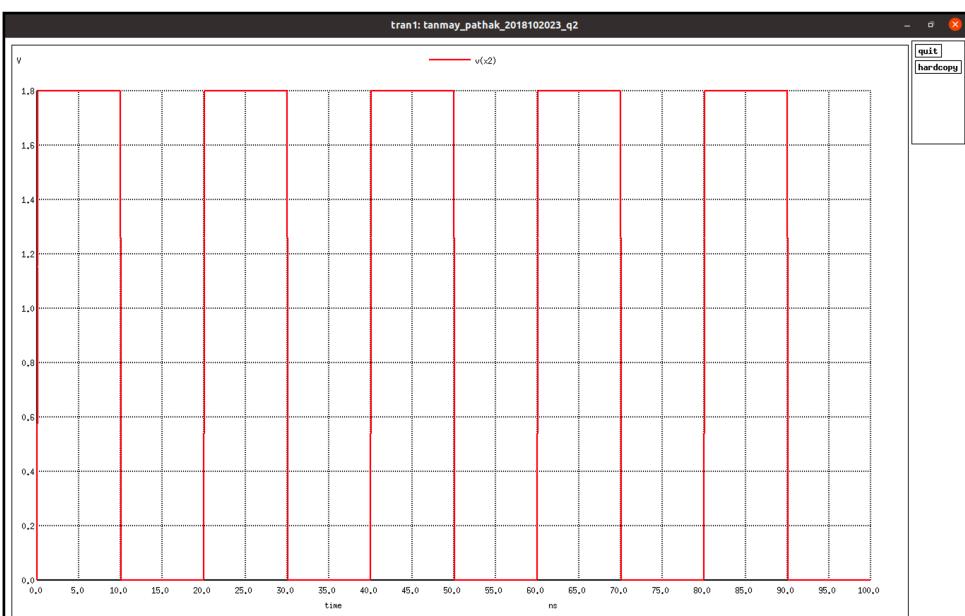
.control
set hcopyright = 1
set color0=white
set color1=black

run
set curplottitle= "Tanmay_pathak_2018102023_q2"
plot v(x1)
plot v(x2)
plot v(x3)
plot v(x4)
plot v(y6)
endc
```

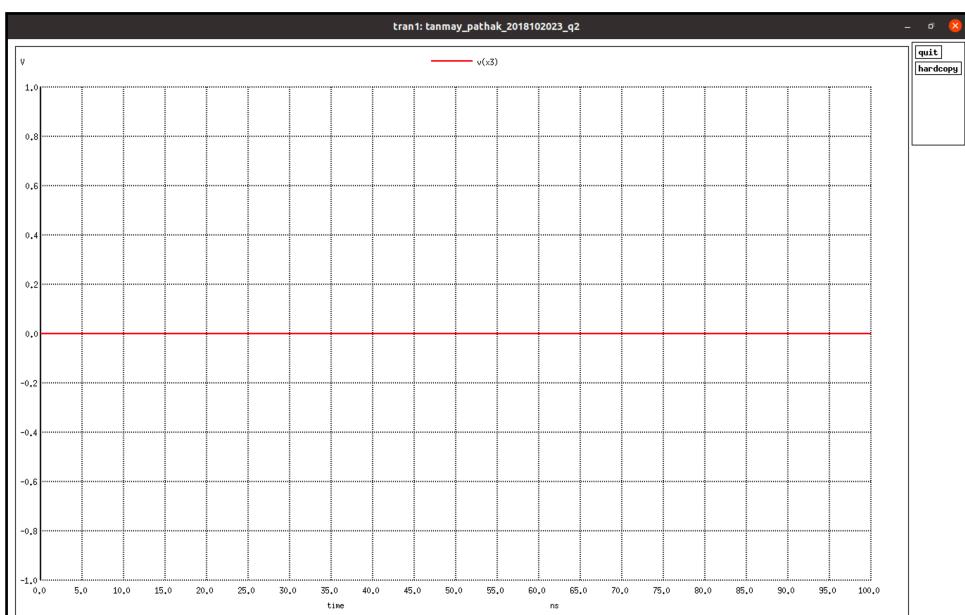
OUTPUT - The output waveforms for the circuit are given below



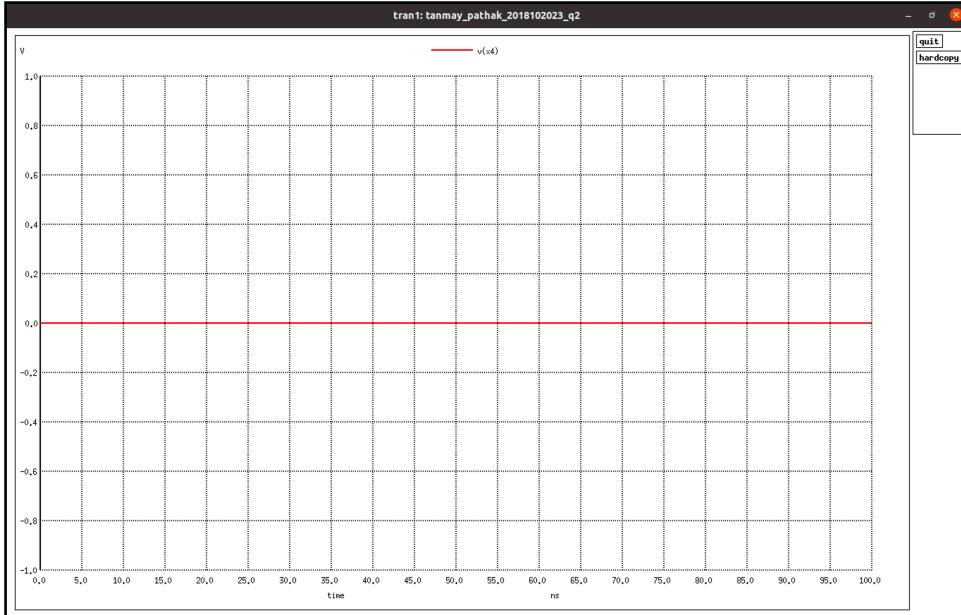
Output waveform for  $x_1$



Output waveform for  $x_2$



Output waveform for  $x_3$

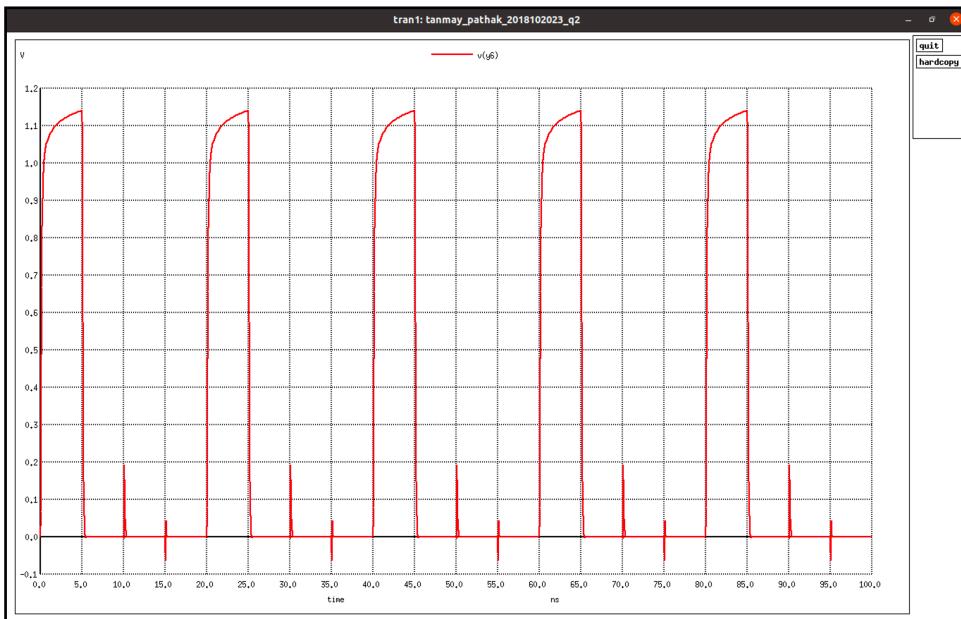


Output waveform for  $x_4$

The circuit working is given below

- 1) It gives a high output when at least 2 of the inputs are set to high
- 2) It gives a low output when at least 3 of the inputs are set to low

Thus to verify the working of the circuit, the output should be similar to the intersection of  $x_1$  &  $x_2$

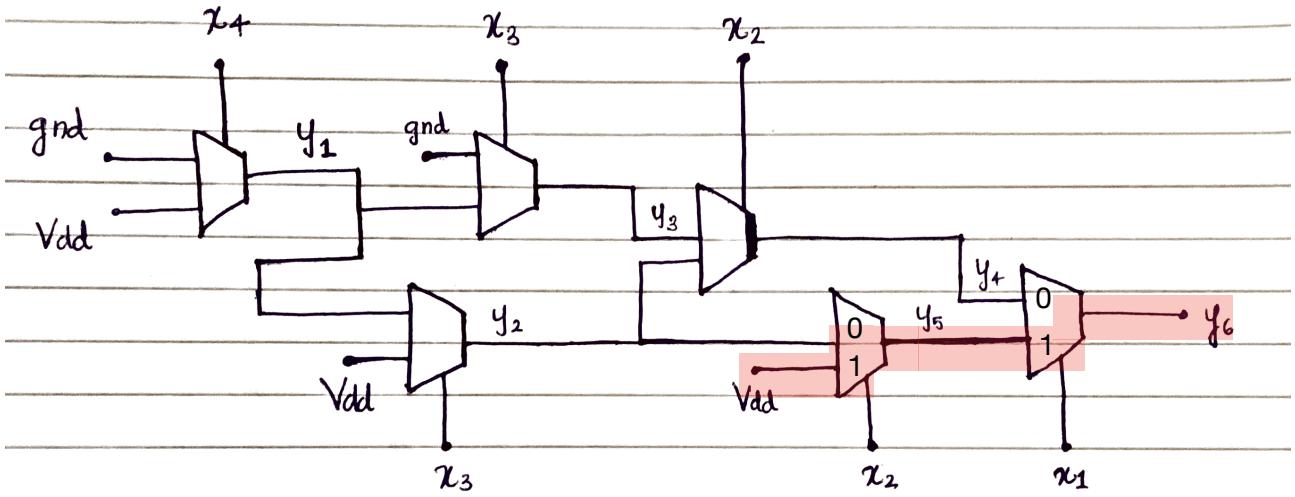


The output is similar to the intersection of  $x_1$  and  $x_2$ . Thus the functionality of the circuit is verified.

---

### Question 2 - (c)

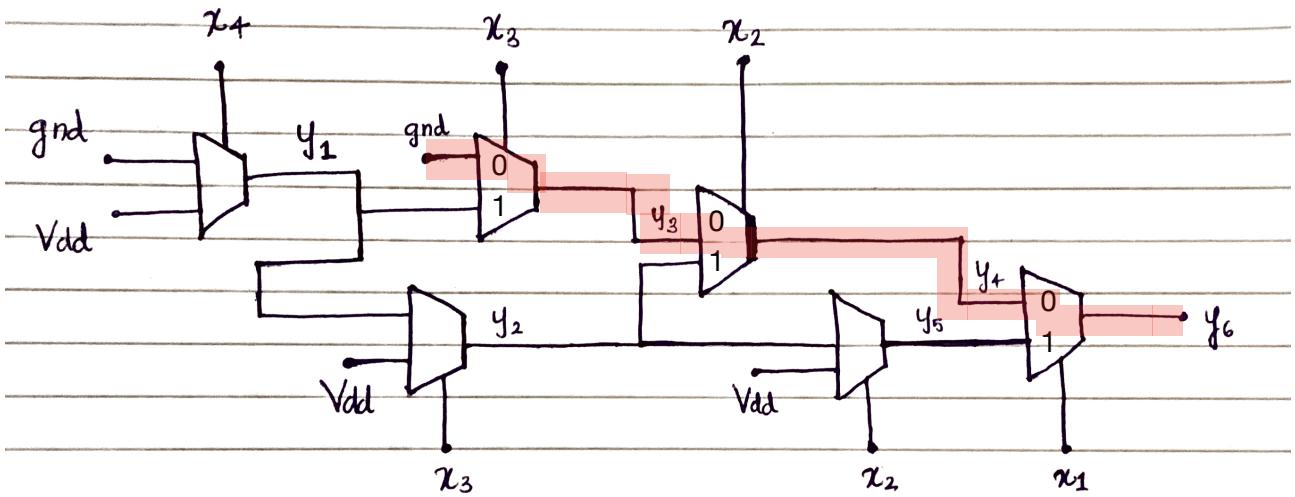
**$t_{PLH}$** : represents the minimum time taken to get a transition from 0 to 1 [low to high] in the output. This will happen when the output gets routed to the closest  $V_{dd}$  in MUX input.



So,  $x_1 = 1$  and  $x_2 = 1$  will route the closest  $V_{dd}$  to the output and hence it will face the least number of transistors and hence least is the delay. Therefore the inputs have to  $x_1=1$  and  $x_2=1$  and either of them flipping from 0 to 1 will give the high at output with least delay.

We observe that  $t_{PLH} = 0.37\text{ns}$

$t_{PHL}$ : represents the minimum time taken to get a transition from 1 to 0 [high to low] in the output. This occurs when the output gets routed to the closest Gnd and MUX input.



So,  $x_3=0$ ,  $x_1=0$  and  $x_2=0$  will route the closest Gnd to the output and hence it will face the least number of transistors and the least delay. Therefore the inputs have to  $x_3=0$ ,  $x_1=0$  and  $x_2=0$  and either of them flipping from 1 to 0 will give the low at output with least delay.

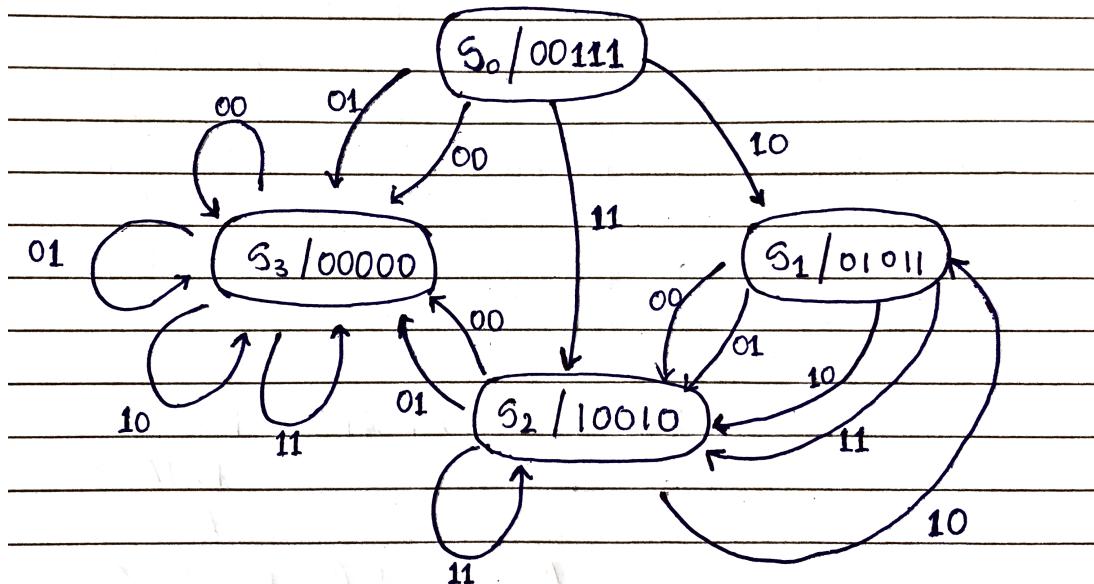
We observe  $t_{PHL} = 0.15\text{ns}$

## Question 2 - (d)

We observe that the  $t_{PHL}$  is smaller than the  $t_{PLH}$ . To negate this difference we can add even number of inverters along the discharge path so that the inverter delay aid in matching with the

high or low delay. We can have a tapered inverter chain setup to account for capacitor and delay requirements.

### Question 3 - (a) - State diagram

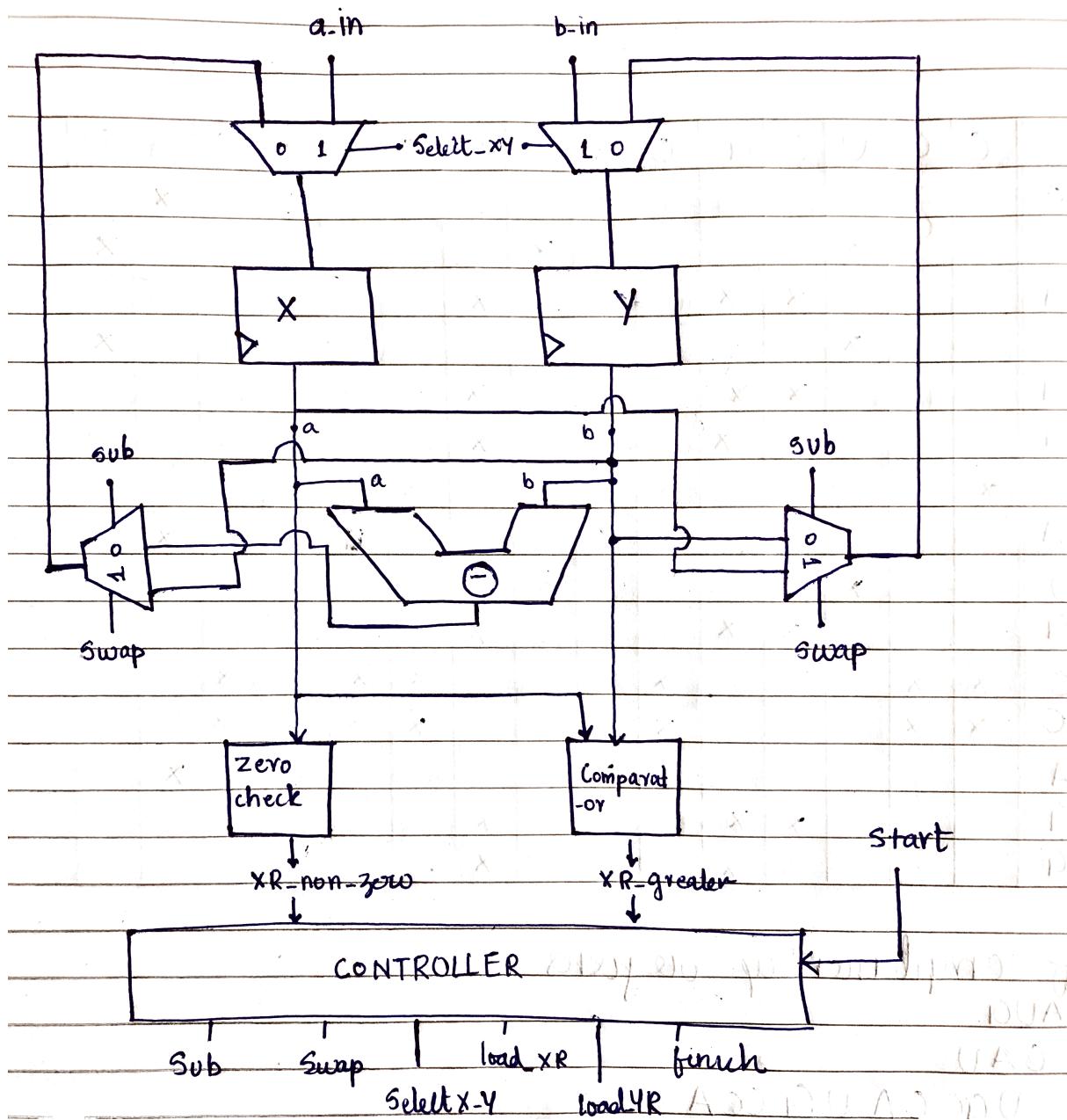


For the given functionality the FSM we construct will have 4 states with the following working.

- 1) **State 0:** This is the begin state and the output corresponding for this state is 00111 and is characterised by the following order [Subtract - Swap - Select\_XY - Load\_XY - Load\_YR]. This is also the state when values whose GCD is to be computed are loaded into respective registers. If Start (another input to the controller) is set to one, then the controller enters this state and gives out 00111 as output.
- 2) **State 1:** This is the swap state and the output corresponding to this state is 01011 and is characterised by the following order [Subtract - Swap - Select\_XY Load\_XY Load\_YR]. This is also the state when the controller understands the value in X register is smaller than that in Y register and instructs the data-path to perform a swap.
- 3) **State 2:** This is the subtract state and the output corresponding to this state is 10010 and is characterised by the following order [Subtract - Swap - Select\_XY - Load\_XY - Load\_YR]. This is the state when the controller instructs the data-path to subtract Y from X and store it back in X.
- 4) **State 3:** This is the end state and the output corresponding to this state is 00000 and is characterised by the following order [Subtract - Swap - Select\_XY - Load\_XY - Load\_YR]. In this state if the value in X has reached zero, the GCD is fully computed. Once this state is reached, the Finish register is set to 1 and the controller will remain in this state till the end and the computed GCD sits in the Y register.

Data path consists of all components [like Registers - MUXs - Comparator - Subtractor] in the circuit that operate in data. Data paths also need control signals so they know which data to operate upon which is facilitated using a controller. A controller gives appropriate control signals to the data path so that they can operate on required data and in our situation, the controller is implemented as an FSM.

### Question 3 - (b) - Circuit Diagram



The Flow of the circuit is as follows

The register X gets loaded with one of the 3 wires [a\_in or (a-b) or b] which is fixed by the control signal given by the controller

The register Y gets loaded with one of the 2 wires [b\_in or b] which is fixed by the control signal given by the controller

The values are loaded in every positive edge of the cycle.

The controller gets XR\_non\_zero and XR\_greater as inputs and are computed but the comparator.

The output of the controller goes in as control signal (select lines) for the data-path.

---

## Question 3 - (c and d) - Verilog implementation

The Verilog implementation for the question is given below

```
// Verilog HDL implementation for Q3 - TANMAY PATHAK - 2018102023
module gcd_new (
    a, b, a_in, b_in, start, finish, clk
);
    input clk, start;
    output finish;
    output [4:0] a, b;
    input [4:0] a_in, b_in;
    reg [4:0] a, b;
    wire start_wire;
    assign start_wire = start;
    reg XR_greater, XR_non_zero, swap, load_XR, load_YR,
    subtract, finish, select_XY;

    wire [4:0] a_wire, b_wire;
    assign a_wire = a;
    assign b_wire = b;

    always @(posedge clk) begin
        if (select_XY && load_XR)begin
            a <= a_in;
        end
        else if (load_XR && swap) begin
            a <= b;
        end
        else if (subtract && load_XR) begin
            a <= a - b;
        end

        if (select_XY && load_YR)begin
            b <= b_in;
        end
        else if (load_YR && swap) begin
            b <= a;
        end
    end

    wire greater, non_zero;
    assign non_zero = a ? 1 : 0;
    assign greater = (a >= b) ? 1 : 0;

    always @(!non_zero) begin
        XR_non_zero = non_zero;
    end
    always @(!greater) begin
        XR_greater = greater;
    end

    always @(*) begin
        if(start_wire) begin
            select_XY <= 1;
            load_XR <= 1;
            load_YR <= 1;
            subtract <= 0;
            swap <= 0;
            finish <= 0;
        end

        else begin
            if(~XR_non_zero) begin
                select_XY <= 0;
                load_XR <= 0;
                load_YR <= 0;
                subtract <= 0;
                swap <= 0;
                finish <= 1;
            end

            else if (~XR_greater) begin
                select_XY <= 0;
                load_XR <= 1;
                load_YR <= 1;
                subtract <= 0;
                swap <= 1;
                finish <= 0;
            end

            else if (XR_greater) begin
                select_XY <= 0;
                load_XR <= 1;
                load_YR <= 0;
                subtract <= 1;
                swap <= 0;
                finish <= 0;
            end
        end
    end
end
endmodule
```

a\_in and b\_in are the input values given to compute the GCD and (a & b) are the registers we are working on. Subtract, Swap, Select\_XY, Load\_XR and Load\_YR are control signals for the functioning of the data-path and they all come from the controller. These control signals are the outputs of the controller. Moreover, the inputs to the controller comes from the data-path and depending in the inputs, the state of the machine is fixed by the controller.

The test bench used is given below

```
//Testbench for Q3 - TANMAY PATHAK - 2018102023
module gcd_test;
    wire finish;
    reg clk, start;
    reg [4:0] a_in, b_in;
    wire [4:0] a, b;

    gcd_new gcd_test(
        .a(a),
        .b(b),
        .a_in(a_in),
        .b_in(b_in),
        .start(start),
        .finish(finish),
        .clk(clk)
    );
    initial begin
        clk = 0;
    end
    parameter PERIOD = 2;
    always # (PERIOD) clk = ~clk;
    initial begin
        $dumpfile("gcd_gtk.vcd");
        $dumpvars(0, gcd_test);
        a_in = 5'd24; //Number_1 for GCD calculation
        b_in = 5'd16; //Number_2 for GCD calculation
        start = 1;
        #4 start = 0;
    end
endmodule
```

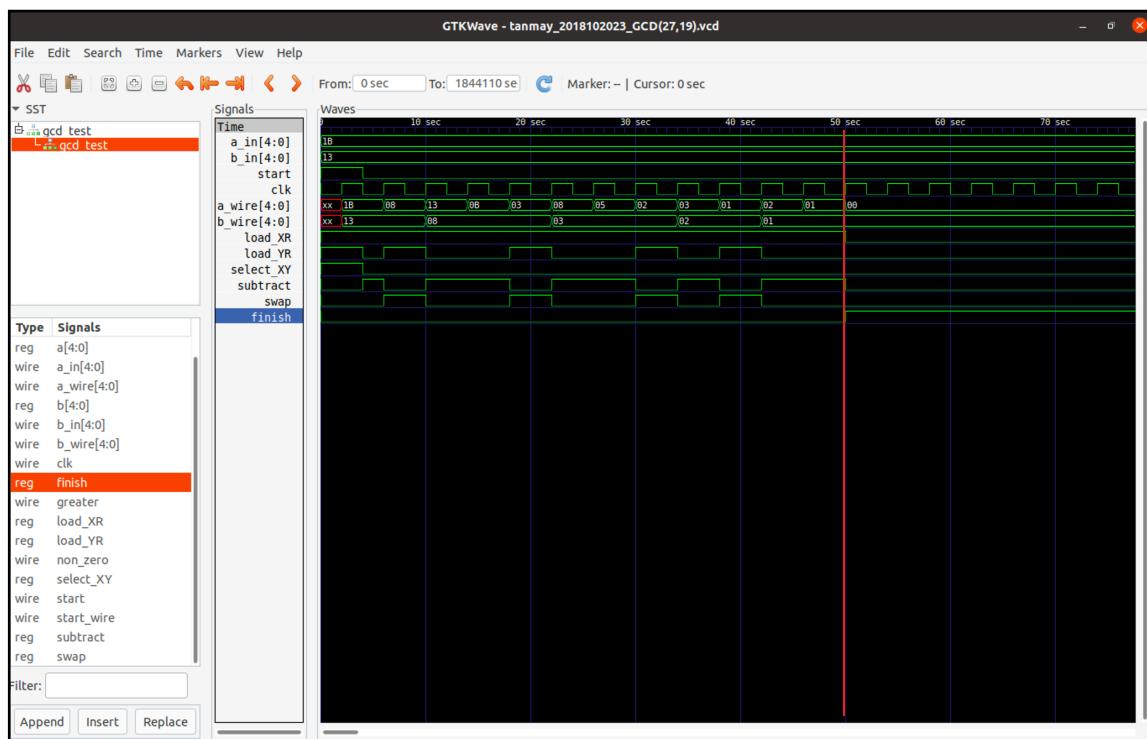
---

## Question 3 - (d) - Simulation results

### Calculate the GCD of 27 and 19

According to the simulation, when the value of finish is 1, we have the final GCD value in b\_wire as 1.

This result is in line with the mathematical calculation of GCD(27,19)



According to the simulation, when the value of finish is 1, we have the final GCD value in b\_wire as 8.

This result is in line with the mathematical calculation of  $\text{GCD}(24, 16)$

