**Q1.**

1. **- Identifying the Critical Path:**

Critical Path Delay = Clock-to-Q Delay + Gate Delays + Setup Time Clock-to-Q Delay (R1) = 60 ps (max) Gate Delays:

* G1 Propagation Delay = 100 ps (max)
* G2 Propagation Delay = 100 ps (max)
* G3 Propagation Delay = 100 ps (max)
* G4 Propagation Delay = 100 ps (max) Setup Time (R2) = 50 ps

Total Critical Path Delay = 60 ps + (4 x 100 ps) + 50 ps = 510 ps

1. **- Calculating the Maximum Clock Rate:**

Maximum Clock Frequency = 1 / Critical Path Delay Maximum Clock Frequency = 1 / 510 ps = 1.96 GHz

1. **- Checking for Hold-Time Violations: Hold Time Condition:**

Clock-to-Q Delay (min) + Minimum Path Delay ≥ Hold Time Clock-to-Q Delay (min) = 40 ps

Minimum Path Delay (G1) = 80 ps (min)

Total Minimum Delay = 40 ps + 80 ps = 120 ps

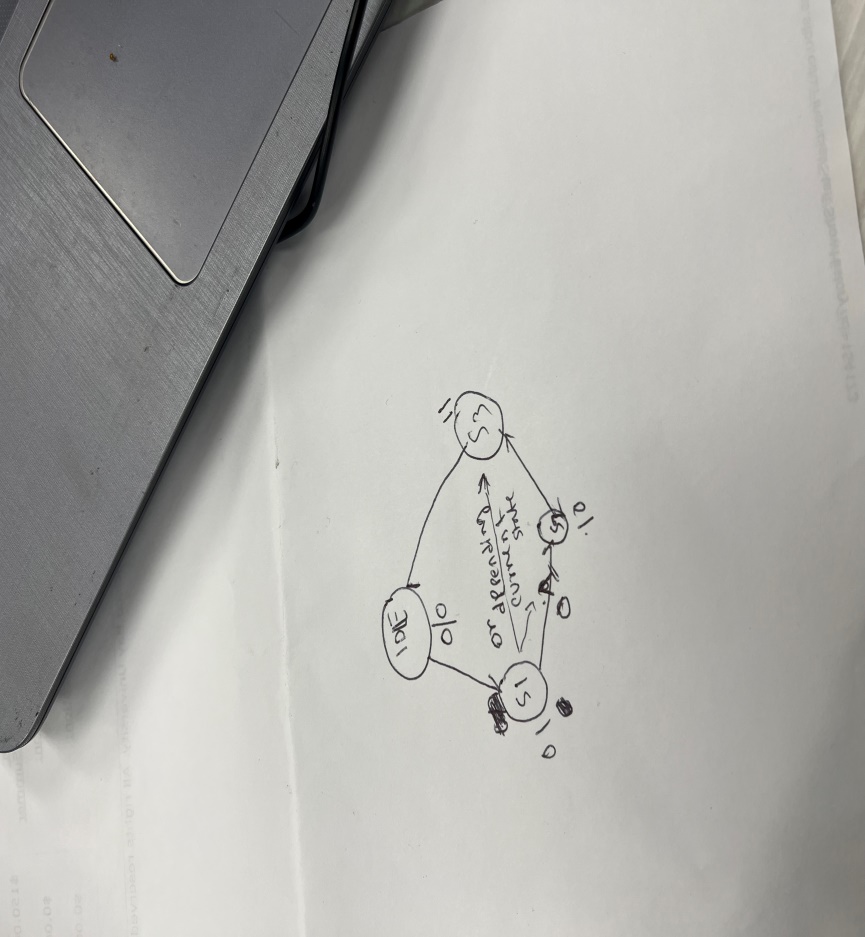
Hold Time = 45 ps

**Why Can't Hold-Time Be Violated?**

Since 120 ps ≥ 45 ps, the hold time condition is satisfied, and there are no hold-time violations.

The timing constraints of the circuit naturally prevent hold-time violations because the minimum delay is greater than the hold time.

**Q2.**

****

`timescale 1ns / 1ps

module fsm (

input clk,

input reset,

output reg [1:0] state

);

localparam IDLE = 2'b00,

S1 = 2'b01,

S2 = 2'b10,

S3 = 2'b11;

always @(posedge clk or posedge reset) begin

if (reset) begin

state <= IDLE;

end

else begin

case (state)

IDLE: state <= S1;

S1: begin

if (state == S1) state <= S2;

else state <= S3;

end

S2: state <= S3;

S3: state <= IDLE;

default: state <= IDLE;

endcase

end

end

endmodule

`timescale 1ns / 1ps

module fsm\_tb;

// Inputs

reg clk;

reg reset;

// Outputs

wire [1:0] state;

// Instantiate the Unit Under Test (UUT)

fsm uut (

.clk(clk),

.reset(reset),

.state(state)

);

initial begin

// Initialize Inputs

clk = 0;

reset = 1;

// Wait 100 ns for global reset to finish

#100;

reset = 0;

// Add stimulus here

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

#50 clk = ~clk;

end

initial begin

$monitor("Time: %0d, State: %b", $time, state);

#500 $finish;

end

endmodule

