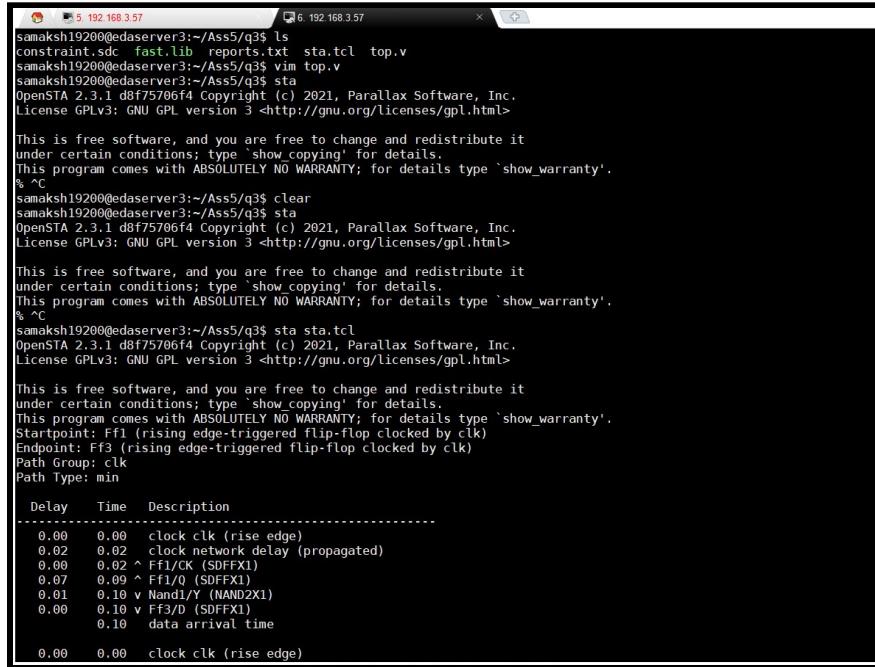


# Assignment 5

Samaksh Gupta  
2019200

Q1)



The screenshot shows a terminal window with two tabs open. The current tab displays the output of a STA (Statistical Analysis Tool) simulation. The command run was `fast.lib reports.txt sta.tcl top.v`. The output includes the OpenSTA license information, which states it is free software under certain conditions, comes with ABSOLUTELY NO WARRANTY, and provides details on how to redistribute it. It also lists the startpoint (FF1), endpoint (FF3), path group (clk), and path type (min). A detailed timing report follows:

| Delay | Time | Description                      |
|-------|------|----------------------------------|
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.02  | 0.02 | clock network delay (propagated) |
| 0.00  | 0.02 | ^ FF1/CK (SDFFX1)                |
| 0.07  | 0.09 | ^ FF1/Q (SDFFX1)                 |
| 0.01  | 0.10 | v Nand1/Y (NAND2X1)              |
| 0.00  | 0.10 | v FF3/D (SDFFX1)                 |
|       | 0.10 | data arrival time                |
| 0.00  | 0.00 | clock clk (rise edge)            |

Q2)

```
module top(
    input clk,
    input in1,
    input in2,
    output out
);
// CLKINVX1
// NAND2X1
// SDFFX1

wire clk1; wire clk2; wire clk3; wire clk4; wire clk5; wire clk6;
wire q1; wire q2; wire q3; wire q4;
wire in22;

CLKINVX1 Inv1(clk, clk1);
CLKINVX1 Inv2(clk1, clk2);
CLKINVX1 Inv3(clk1, clk3);
CLKINVX1 Inv4(clk1, clk4);
CLKINVX1 Inv5(clk4, clk5);
CLKINVX1 Inv6(clk5, clk6);

SDFFX1 Ff1(.CK(clk2), .D(in1), .Q(q1));
SDFFX1 Ff2(.CK(clk3), .D(in22), .Q(q2));
SDFFX1 Ff3(.CK(clk6), .D(q4), .Q(out));

CLKINVX1 Inv7(in2, in22);
CLKINVX1 Inv8(q2, q3);

NAND2X1 Nand1(q1, q3, q4);

endmodule
```

## Constraint Commands

create\_clock -name clk -period 10 [get\_ports clk] : Creates the clock with period 10.

set\_propagated\_clock [get\_clocks clk] : Add some delays for non idealities in the clock.

set\_input\_delay -clock clk 1 [get\_ports {in1 in2}] : Put delays on the input ports in1, in2

set\_input\_transition 0.01 [get\_ports {in1 in2}] : Add slew to the input ports (Putting rise/fall time)

set\_output\_delay -clock clk 0.2 [get\_ports {out}] : Add delays to constraint the output

set\_load 0.05 [get\_ports {out}] : Puts the final output load our 'out' port would have.

set\_multicycle\_path : This command specifies the number of cycles that the data path must-have for setup or hold, so that designated timing paths in the current design has no default setup or hold relations ~ [Source](#)

## Setup Time and Hold Time

**Setup Analysis:** The data on the input pin of the capture flip flop must arrive 'Setup Time' before the clock edge arrives on the flip flop.

**Hold Analysis:** A new data must not arrive on the input pin of the capture flip flop for 'Hold Time' after the clock edge has arrived on the flip flop.

We would see in the assignment that most of the changes in the constraint file don't affect the Hold Slack as Hold Time is calculated on the same clock edge in which the data was launched by the launching flip flop.

**Clock Skew:** The positive edge of two different flip flops (routed via the same clock) will not necessarily arrive at the same time. The flip-flop at distance would get clock after a delay. This affects both Setup and Hold time.

**Clock Jitter:** This affects a clock connected to one a single flop. This is an internal effect. It defines that the clock edge might occur after some delay (+ve or -ve) in comparison to the period. This has no effect on Hold analysis, It only affects Setup analysis.

Hold Analyses are much more problematic. Setup violations can be resolved by reducing the clock frequency and making the system slower in general. However, Hold violations are extremely difficult to remove.

Q3)

```
5. 192.168.3.57 6. 192.168.3.57
create_clock -name clk -period 10 [get_ports clk]
set_propagated_clock [get_clocks clk]

set_input_delay -clock clk 1 [get_ports {in1 in2}]
set_input_transition 0.01 [get_ports {in1 in2}]

set_output_delay -clock clk 0.2 [get_ports {out}]
set_load 0.05 [get_ports {out}]
~
```

```
This is free software, and you are free to change and redistribute it
under certain conditions; type `show_copyright' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
Startpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Endpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min

Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.02 0.02 clock network delay (propagated)
0.00 0.02 ^ Ff1/CK (SDFFX1)
0.07 0.09 ^ Ff1/Q (SDFFX1)
0.01 0.10 v Nand1/Y (NAND2X1)
0.00 0.10 v Ff3/D (SDFFX1)
0.10 data arrival time

0.00 0.00 clock clk (rise edge)
0.04 0.04 clock network delay (propagated)
0.00 0.04 clock reconvergence pessimism
0.04 ^ Ff3/CK (SDFFX1)
-0.01 0.02 library hold time
0.02 data required time
-----
0.02 data required time
-0.10 data arrival time
-----
0.08 slack (MET)

Startpoint: in2 (input port clocked by clk)
Endpoint: Ff2 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max

Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 v input external delay
0.00 1.00 v in2 (in)
0.01 1.01 ^ Inv7/Y (CLKINVX1)
0.00 1.01 ^ Ff2/D (SDFFX1)
1.01 data arrival time

10.00 10.00 clock clk (rise edge)
0.02 10.02 clock network delay (propagated)
0.00 10.02 clock reconvergence pessimism
10.02 ^ Ff2/CK (SDFFX1)
-0.05 9.97 library setup time
9.97 data required time
-----
9.97 data required time
-1.01 data arrival time
-----
8.96 slack (MET)
```

**The worst path for hold analysis:** FF1 → FF3 @ Slack of 0.08

**The worst path for Setup analysis:** In2 → FF2 @ Slack of 8.96

Constraints of Q3)

Here, I have set the clock period to 10 units and gave a propagated clock delay for non-idealities. Next, I am constraining inputs and outputs using the input, output delay command.

The load as per the constraints file is set to 0.05fF.

We can see the tool ran and the slack came out to be positive for both hold and setup analysis. The worst path was also reported by the tool (mentioned above). This path, however, depends on the clock period and the capacitance load.

Input transition time is set to 0.01, this would be significantly increased in later questions.

#### Q-4)

```
##  
#Constraints file for Q4  
##  
  
#Give the clock  
create_clock -name clk -period 10 [get_ports clk]  
  
#Set non Idealities  
set_propagated_clock [get_clocks clk]  
  
#Constraints on input  
set_input_delay -clock clk 1 [get_ports {in1 in2}]  
set_input_transition 0.01 [get_ports {in1 in2}]  
  
#Constraints on output  
set_output_delay -clock clk 0.2 [get_ports {out}]  
set_load 0.05 [get_ports {out}]  
  
#Latency  
set_clock_latency 0.8 -source [get_clocks clk]  
set_clock_latency 1.1 [get_clocks clk]  
  
#Transition  
set_clock_transition 0.75 clk  
  
# Uncertainty  
set_clock_uncertainty -setup 0.2 [get_clocks clk]  
set_clock_uncertainty -hold 0.05 [get_clocks clk]  
~  
~
```

#### Effect on Hold Slack: -

Clock uncertainty contains two parts ‘Jitter’ and ‘Skew’. Since all the flip flops get triggered on the positive edge; ‘Jitter’ will not affect the slack. However, skew in the worst case (If positive) will further constraint the hold requirements and hence reduce the slack.

#### Effect on Setup Slack: -

Setup is affected by both jitter and skew. Especially if the skew is negative then although ‘hold’ gets relaxed, but setup gets further constrained and we would see a reduction in the value of setup slack.

| Startpoint: in2 (input port clocked by clk)                    |             |                               |
|--|-------------|-------------------------------|
| Endpoint: Ff2 (rising edge-triggered flip-flop clocked by clk) |             |                               |
| Path Group: clk  |             |                               |
| Path Type: max   |             |                               |
| Delay  | Time        | Description                   |
| 0.00   | 0.00        | clock clk (rise edge)         |
| 1.90   | 1.90        | clock network delay (ideal)   |
| 1.00   | 2.90        | v input external delay        |
| 0.00   | 2.90        | v in2 (in)                    |
| 0.01   | 2.91        | ^ Inv7/Y (CLKINVX1)           |
| 0.00   | 2.91        | ^ Ff2/D (SDFFX1)              |
| 0.00   | 2.91        | data arrival time             |
| 10.00  | 10.00       | clock clk (rise edge)         |
| 1.90   | 11.90       | clock network delay (ideal)   |
| -0.20  | 11.70       | clock uncertainty             |
| 0.00   | 11.70       | clock reconvergence pessimism |
| 1.95   | 11.70       | ^ Ff2/CK (SDFFX1)             |
| -0.01  | 11.69       | library setup time            |
| 0.00   | 11.69       | data required time            |
| 2.05   | 11.69       | data required time            |
| -2.07  | 11.69       | data arrival time             |
| 0.02   | slack (MET) |                               |
| 11.69  | slack (MET) |                               |
| -2.91  | slack (MET) |                               |
| 8.79   | slack (MET) |                               |

Hold Slack: 0.02

Setup Slack: 8.79

The results are consistent with the hypothesis, that in the worst-case slacks have further reduced.

**Q5)**

The screenshot shows two terminal windows. The left window has the title '5. 192.168.3.57' and contains the Verilog code for Q5. The right window has the title '6. 192.168.3.57' and displays the analysis results.

```
# Q5
#Divinding the clock period by 3, rest is same as Q3
create_clock -name clk -period 3.33 [get_ports clk]
set_propagated_clock [get_clocks clk]

set_input_delay -clock clk 1 [get_ports {in1 in2}]
set_input_transition 0.01 [get_ports {in1 in2}]

set_output_delay -clock clk 0.2 [get_ports {out}]
set_load 0.05 [get_ports {out}]
~
```

The screenshot shows two terminal windows. The left window contains the Verilog code for Q5. The right window displays the timing analysis results.

is program comes with ABSOLUTELY NO WARRANTY; for details type 'show\_warranty'.
rst slack 2.29
rst slack 0.08
arpoint: FF1 (rising edge-triggered flip-flop clocked by clk)
dpoint: FF3 (rising edge-triggered flip-flop clocked by clk)
th group: clk
th type: min

| Delay | Time | Description                      |
|-------|------|----------------------------------|
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.02  | 0.02 | clock network delay (propagated) |
| 0.00  | 0.02 | ^ FF1/CK (SDFFX1)                |
| 0.07  | 0.09 | ^ FF1/Q (SDFFX1)                 |
| 0.01  | 0.10 | v NAND1/Y (NAND2X1)              |
| 0.00  | 0.10 | v FF3/D (SDFFX1)                 |
| 0.10  |      | data arrival time                |
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.04  | 0.04 | clock network delay (propagated) |
| 0.00  | 0.04 | clock reconvergence pessimism    |
| 0.04  | 0.04 | ^ FF3/CK (SDFFX1)                |
| -0.01 | 0.02 | library hold time                |
| 0.02  |      | data required time               |
| 0.02  |      | data required time               |
| -0.10 |      | data arrival time                |
| 0.08  |      | slack (MET)                      |

Startpoint: in2 (input port clocked by clk)
Endpoint: FF2 (rising edge-triggered flip-flop clocked by clk)
path Group: clk
path Type: max

| Delay | Time | Description                      |
|-------|------|----------------------------------|
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.00  | 0.00 | clock network delay (propagated) |
| 1.00  | 1.00 | v input external delay           |
| 0.00  | 1.00 | v in2 (in)                       |
| 0.01  | 1.01 | ^ INV3/Y (CLKINVX1)              |
| 0.00  | 1.01 | ^ FF2/O (SDFFX1)                 |
| 1.01  |      | data arrival time                |
| 3.33  | 3.33 | clock clk (rise edge)            |
| 0.02  | 3.35 | clock network delay (propagated) |
| 0.00  | 3.35 | clock reconvergence pessimism    |
| 3.35  | 3.35 | ^ FF2/CK (SDFFX1)                |
| -0.05 | 3.30 | library setup time               |
| 3.30  |      | data required time               |
| 3.30  |      | data required time               |
| -1.01 |      | data arrival time                |
| 2.29  |      | slack (MET)                      |

### **Effect on Hold Slack: -**

Changing the clock period has no effect on Hold time checks since hold time is calculated on the same clock edge for the capturing and launching flip flop. We can observe that the Hold slack is the same as that of Q-3 (0.08).

### **Effect on Setup Slack: -**

In Setup analysis, data must arrive at the input of the flip flop, ‘setup’ time before the clock arrives at the clk pin on the flip flop. Since we have reduced the clock period this would imply that the data has a tighter constraint now. It has to come even quicker as the clock signals come 3 times faster now. So, we should see a reduction in the slack ( $8.96 \rightarrow 2.29$ )

## Q6)

```
# 5. 192.168.3.57      # 6. 192.168.3.57

# Q6

#Increasing The Transition Time on input

create_clock -name clk -period 10 [get_ports clk]
set_propagated_clock [get_clocks clk]

set_input_delay -clock clk 1 [get_ports {in1 in2}]

#Setting the transition time to 0.75 (Also, checked on 0.5 and 1)
set_input_transition 0.75 [get_ports {in1 in2}]

set_output_delay -clock clk 0.2 [get_ports {out}]
set_load 0.05 [get_ports {out}]
```

```
This is free software, and you are free to change and redistribute it
under certain conditions; type 'show_copyright' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.
Startpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Endpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min
Delay  Time  Description
-----.
0.00  0.00  clock clk (rise edge)
0.02  0.02  clock network delay (propagated)
0.00  0.02 ^ Ff1/CK (SDFFX1)
0.07  0.09 ^ Ff1/Q (SDFFX1)
0.01  0.10 v Nand1/Y (NAND2X1)
0.00  0.10 v Ff3/D (SDFFX1)
0.10  0.10 data arrival time
0.00  0.00  clock clk (rise edge)
0.04  0.04  clock network delay (propagated)
0.00  0.04  clock reconvergence pessimism
0.04 ^ Ff3/CK (SDFFX1)
-0.01  0.02 library hold time
0.02  0.02 data required time
0.02  0.02 data required time
-0.10  0.10 data arrival time
0.08  slack (MET)

Startpoint: in2 (input port clocked by clk)
Endpoint: Ff2 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
Delay  Time  Description
-----.
0.00  0.00  clock clk (rise edge)
0.00  0.00  clock network delay (propagated)
1.00  1.00 v input external delay
0.00  1.00 v in2 (in)
0.09  1.09 ^ Inv7/Y (CLKINVX1)
0.00  1.09 ^ Ff2/D (SDFFX1)
1.09  data arrival time
10.00 10.00  clock clk (rise edge)
0.02 10.02  clock network delay (propagated)
0.00 10.02  clock reconvergence pessimism
10.02 ^ Ff2/CK (SDFFX1)
-0.08  9.94 library setup time
9.94  data required time
9.94  data required time
-1.09  9.94 data arrival time
8.85  slack (MET)
```

Transition:0.5

Hold Slack: 0.08

Setup Slack: 8.88

Transition:0.75

Hold Slack: 0.08

Setup Slack: 8.85

Transition:1

Hold Slack: 0.08

Setup Slack: 8.82

**Hold slack** is independent of Transition time on the input ports. We see that the hold slack does not change. Hold slack is calculated on the same clock edge and having transition time on the clock edge will not change anything.

**Setup slack** reduces as transition time is increased. Having an ideal case where transition time is almost zero would give the maximum slack and we would have a relaxed constraint. If our transition time is poor our setup would increase and hence the slack would reduce resulting in tighter constraints. We know that Setup slack is a non-linear function of slew, and we can observe it via these experiments.

Q7)

5. 192.168.3.57      6. 192.168.3.57

```
###  
# Q7  
#Effect Of Increasing The Load at The Output.  
  
create_clock -name clk -period 10 [get_ports clk]  
set_propagated_clock [get_clocks clk]  
  
set_input_delay -clock clk 1 [get_ports {in1 in2}]  
set_input_transition 0.01 [get_ports {in1 in2}]  
  
set_output_delay -clock clk 0.2 [get_ports {out}]  
  
# Increasing the output Load to 1.75  
set_load 1.75 [get_ports {out}]
```

5. 192.168.3.57      6. 192.168.3.57

Under certain conditions; type 'show\_copying' for details.  
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show\_warranty'.  
Startpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)  
Endpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)  
Path Group: clk  
Path type: min

| Delay | Time | Description                      |
|-------|------|----------------------------------|
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.02  | 0.02 | clock network delay (propagated) |
| 0.00  | 0.04 | Ff1/Q (SDFFX1)                   |
| 0.07  | 0.09 | ^ Ff1/Y (SDFFX1)                 |
| 0.01  | 0.10 | v Nand1/Y (NAND2X1)              |
| 0.00  | 0.10 | v Ff3/D (SDFFX1)                 |
| 0.10  | 0.10 | data arrival time                |
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.04  | 0.04 | clock network delay (propagated) |
| 0.00  | 0.04 | clock reconvergence pessimism    |
| 0.04  | 0.04 | ^ Ff3/CK (SDFFX1)                |
| -0.01 | 0.02 | library hold time                |
| 0.02  | 0.02 | data required time               |
| 0.00  | 0.00 | clock clk (rise edge)            |
| 0.02  | 0.02 | clock network delay (propagated) |
| 0.00  | 0.02 | clock reconvergence pessimism    |
| 0.02  | 0.02 | ^ Ff3/Y (SDFFX1)                 |
| 0.00  | 0.02 | data required time               |
| 0.00  | 0.02 | data arrival time                |
| 0.08  | 0.08 | slack (MET)                      |

Startpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)  
Endpoint: out (output port clocked by clk)  
Path Group: clk  
Path Type: max

| Delay | Time  | Description                      |
|-------|-------|----------------------------------|
| 0.00  | 0.00  | clock clk (rise edge)            |
| 0.04  | 0.04  | clock network delay (propagated) |
| 0.00  | 0.04  | ^ Ff3/CK (SDFFX1)                |
| 5.50  | 5.54  | ^ Ff3/Q (SDFFX1)                 |
| 0.00  | 5.54  | out (out)                        |
|       | 5.54  | data arrival time                |
| 10.00 | 10.00 | clock clk (rise edge)            |
| 0.00  | 10.00 | clock network delay (propagated) |
| 0.00  | 10.00 | clock reconvergence pessimism    |
| -0.20 | 9.80  | output external delay            |
|       | 9.80  | data required time               |
|       | 9.80  | data required time               |
|       | -5.54 | data arrival time                |
|       | 4.26  | slack (MET)                      |

Load: 1.75      Hold: 0.08      Setup: 4.26

Load: 1.5      Hold: 0.08      Setup: 5.04

Load: 1      Hold: 0.08      Setup: 6.59

Increasing the final output load can be analogous to having an increased combination delay and an increased clk-q delay. This increases the time data needs to propagate and hence, the data reaches late in comparison to Q3. So we see a fall in the setup slack.

Hold slack remains unchanged because the effect of increasing the output load is felt by the clocks too and hence on the same edge, a difference would not be caused.

Q8)

```
5. 192.168.3.57 6. 192.168.3.57
#Q_8
##Same File as Q3, except last three lines

#Master Clock
create_clock -name clk -period 10 [get_ports clk]

#Non_idealities in clk
set_propagated_clock [get_clocks clk]

#Constrainin Inputs and Outputs
set_input_delay -clock clk 1 [get_ports {in1 in2}]
set_input_transition 0.01 [get_ports {in1 in2}]
set_output_delay -clock clk 0.2 [get_ports {out}]

#Load on the output
set_load 0.05 [get_ports {out}]

# Multi-cycle path for setup analysis with path multiplier 8
set_multicycle_path 8 -setup -from [get_ports {in2}] -to [get_pins FF2/D]

#False path to fix hold failure for that path
#set_false_path -hold -from [get_ports {in2}] -to [get_pins FF2/D]

#
#Multi cycle path for hold analysis
#set_multicycle_path 8 -hold -from [get_ports {in2}] -to [get_pins FF2/D]
~
```

```
This is free software, and you are free to change and redistribute it
under certain conditions; type 'show_copyright' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type 'show_warranty'.
Startpoint: in2 (input port clocked by clk)
Endpoint: FF2 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
Delay Time Description
-----+
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 ^ input external delay
0.00 1.00 ^ in2 (in)
0.01 1.01 v Inv7/Y (CLKINVX1)
0.00 1.01 v FF2/D (SDFFX1)
1.01 data arrival time
70.00 70.00 clock clk (rise edge)
0.02 70.02 clock network delay (propagated)
0.00 70.02 clock reconvergence pessimism
70.02 ^ FF2/CK (SDFFX1)
-0.01 70.01 library hold time
70.01 data required time
-----+
70.01 data required time
-1.01 data arrival time
-----+
-69.00 slack (VIOLATED)

Startpoint: in1 (input port clocked by clk)
Endpoint: FF1 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
Delay Time Description
-----+
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 ^ input external delay
0.00 1.00 ^ in1 (in)
0.00 1.00 ^ FF1/D (SDFFX1)
1.00 data arrival time
10.00 10.00 clock clk (rise edge)
0.02 10.02 clock network delay (propagated)
0.00 10.02 clock reconvergence pessimism
10.02 ^ FF1/CK (SDFFX1)
-0.05 9.97 library setup time
9.97 data required time
-----+
9.97 data required time
-1.00 data arrival time
-----+
8.97 slack (MET)
```

Hold Slack: -69 (Violated)

Setup Slack: 8.97

Since we have a violation... we would continue to fix it via the two methods given in the question.

## SET FALSE PATH has been uncommented and the hold slack is positive now.

```
5. 192.168.3.57 6. 192.168.3.57
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
Startpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Endpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min
Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.02 0.02 clock network delay (propagated)
0.00 0.02 ^ Ff1/CK (SDFFX0)
0.07 0.09 ^ Ff1/Q (SDFFX1)
0.01 0.10 v Nand1/Y (NAND2X1)
0.00 0.10 v Ff3/D (SDFFX1)
0.10 0.10 data arrival time

0.00 0.00 clock clk (rise edge)
0.04 0.04 clock network delay (propagated)
0.00 0.04 clock reconvergence pessimism
0.04 ^ Ff3/CK (SDFFX1)
-0.01 0.02 library hold time
0.02 data required time

0.02 data required time
0.10 data arrival time
-----
0.08 slack (MET)

Startpoint: in1 (input port clocked by clk)
Endpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 ^ input external delay
0.00 1.00 ^ in1 (in)
0.00 1.00 ^ Ff1/D (SDFFX1)
1.00 1.00 data arrival time

10.00 10.00 clock clk (rise edge)
0.02 10.02 clock network delay (propagated)
0.00 10.02 clock reconvergence pessimism
10.02 ^ Ff1/CK (SDFFX1)
-0.05 9.97 library setup time
9.97 data required time

9.97 data required time
-1.00 data arrival time
-----
8.97 slack (MET)
```

Hold Slack: 0.08

Setup Slack: 8.97

## SET FALSE PATH has been commented and another set\_multicycle is uncommented.

```
5. 192.168.3.57 6. 192.168.3.57
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
Startpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Endpoint: Ff3 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min
Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.02 0.02 clock network delay (propagated)
0.00 0.02 ^ Ff1/CK (SDFFX1)
0.07 0.09 ^ Ff1/Q (SDFFX1)
0.01 0.10 v Nand1/Y (NAND2X1)
0.00 0.10 v Ff2/D (SDFFX1)
0.10 0.10 data arrival time

0.00 0.00 clock clk (rise edge)
0.04 0.04 clock network delay (propagated)
0.00 0.04 clock reconvergence pessimism
0.04 ^ Ff3/CK (SDFFX1)
-0.01 0.02 library hold time
0.02 data required time

0.02 data required time
-0.10 data arrival time
-----
0.08 slack (MET)

Startpoint: in1 (input port clocked by clk)
Endpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: max
Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 ^ input external delay
0.00 1.00 ^ in1 (in)
0.00 1.00 ^ Ff1/D (SDFFX1)
1.00 1.00 data arrival time

10.00 10.00 clock clk (rise edge)
0.02 10.02 clock network delay (propagated)
0.00 10.02 clock reconvergence pessimism
10.02 ^ Ff1/CK (SDFFX1)
-0.05 9.97 library setup time
9.97 data required time

9.97 data required time
-1.00 data arrival time
-----
8.97 slack (MET)
```

Hold Slack: 0.08

Setup Slack: 8.97

**8a)** Here we get a hold violation and an increased slack. The multi-cycle path command increases a relaxation on the two subsequent positive edges and as a result, the setup time improves. But we enter a hold violation as we are checking on the 8th scale. This results in the data arriving during the hold time of the positive edge of the capture flip flop. Hold analysis is performed 1 edge before the setup is done (7th) and as a result, a violation has occurred. The data has changed between the beginning and the Hold\_Window + 7th clock edge time.

**Hold Slack:** -69 (Violated)

**Setup Slack:** 8.97

**8b)** As mentioned in the question PDF, we are caring the hold time violation via the set\_false\_path command. This causes the tool to not consider the path that is causing the hold violation during hold analysis. This gives us a clean result in the hold analysis as the path itself is ignored by the tool.

**Hold Slack:** 0.08

**Setup Slack:** 8.97

**8c)** Here we fix the hold violation by method 2 as mentioned in the pdf. We use the multicycle command and we observe that the hold violation has been removed.

Setup analysis is done on the 8th positive edge of the clock, to remove violations we need the hold to be evaluated at least after the 7th clock edge. Thus, we put it as such (8). Hence, by using multi-cycle path command and the factor as 8, we are able to remove the hold violation.

**Hold Slack:** 0.08

**Setup Slack:** 8.97

## DESCRIPTION

This command specifies the number of cycles that the data path must have for setup or hold, so that that designated timing paths in the current design have no default setup or hold relations.

PrimeTime applies certain rules to determine single cycle timing relationships for paths between clocked elements; the rules are based on active edges. For flip-flops, a single active edge both launches and captures data. For latches, the open edge launches data and the close edge latches data.

## Source:

[https://www.micro-ip.com/STA/dictionary\\_547\\_17/set\\_multicycle\\_path.html](https://www.micro-ip.com/STA/dictionary_547_17/set_multicycle_path.html)

### Q-9)

```
5. 192.168.3.57 6. 192.168.3.57
#09
##
#Same as Question 3, but only the top file has changed.

create_clock -name clk -period 10 [get_ports clk]

set_propagated_clock [get_clocks clk]

set_input_delay -clock clk 1 [get_ports {in1 in2}]
set_input_transition 0.01 [get_ports {in1 in2}]

set_output_delay -clock clk 0.2 [get_ports {out}]

set_load 0.05 [get_ports {out}]
~
```

```
5. 192.168.3.57 6. 192.168.3.57
This is free software, and you are free to change and redistribute it
under certain conditions; type `show_copying' for details.
This program comes with ABSOLUTELY NO WARRANTY; for details type `show_warranty'.
Startpoint: in1 (input port clocked by clk)
Endpoint: Ff1 (rising edge-triggered flip-flop clocked by clk)
Path Group: clk
Path Type: min

Delay Time Description
-----
0.00 0.00 clock clk (rise edge)
0.00 0.00 clock network delay (propagated)
1.00 1.00 v input external delay
0.00 1.00 v in1 (in)
0.00 1.00 v Ff1/D (SDFFX1)
1.00 data arrival time

0.00 0.00 clock clk (rise edge)
0.02 0.02 clock network delay (propagated)
0.00 0.02 clock reconvergence pessimism
0.02 ^ Ff1/CK (SDFFX1)
-0.01 0.01 library hold time
0.01 data required time
-----
0.01 data required time
-1.00 data arrival time
-----
0.99 slack (MET)

Startpoint: Ff3 (rising edge-triggered flip-flop clocked by clk')
Endpoint: out (output port clocked by clk)
Path Group: clk
Path Type: max

Delay Time Description
-----
5.00 5.00 clock clk' (rise edge)
0.03 5.03 clock network delay (propagated)
0.00 5.03 ^ Ff3/CK (SDFFX1)
0.23 5.26 v Ff3/Q (SDFFX1)
0.00 5.26 v out (out)
5.26 data arrival time

10.00 10.00 clock clk (rise edge)
0.00 10.00 clock network delay (propagated)
0.00 10.00 clock reconvergence pessimism
-0.20 9.80 output external delay
9.80 data required time
-----
9.80 data required time
-5.26 data arrival time
-----
4.54 slack (MET)
```

The worst path for hold analysis: In1 → FF1 @ Slack of 0.99

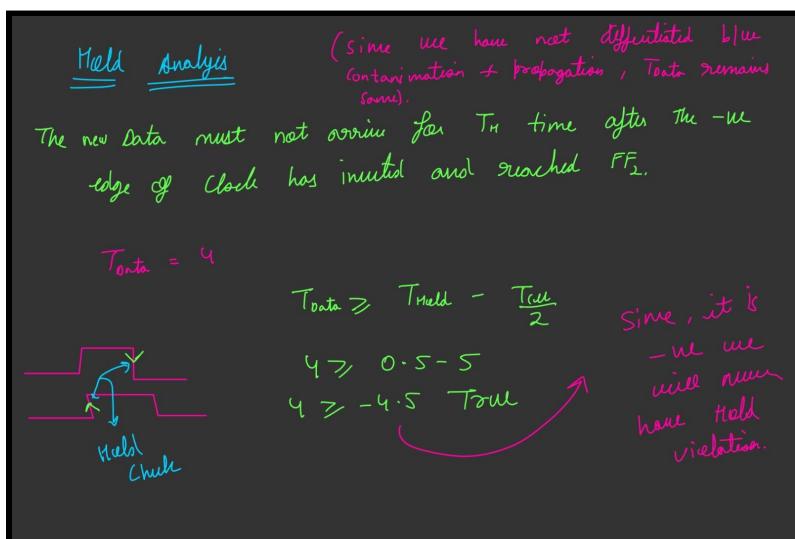
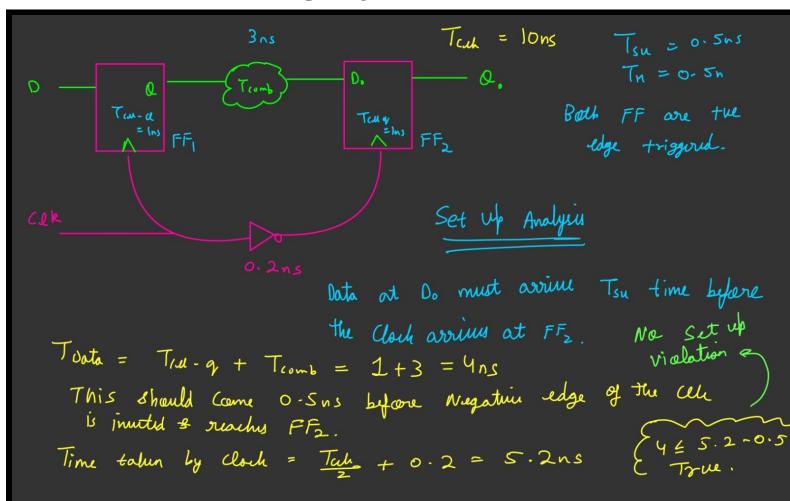
The worst path for Setup analysis: FF3 → Out @ Slack of 4.54

Hold slack has increased and Setup slack has reduced. The worst-case path has changed as well. The idea is that here, flip flop 3 is triggered on the negative edge and the other two flip flops are on the positive edge of the input 'clk'. This results in a notion that not even hold will depend on the period of the clock.

**Setup time** would reduce as not in the 'Clock edge timing': Tclk has reduced to Tclk/2 and therefore Slack will go down (Clk\_Time - Data\_Time).

**Hold time** will increase as not we need to consider Tclk/2 as well while doing a comparison between new data. This wouldn't be needed in the case where all flip flops trigger on the same edge. This additional Tclk/2 factor increases the hold time.

### Consider the following Toy Example to understand this concept: -



As a result, hold analysis has now increased in the question and the set-up time went down. The calculations have changed as per the following mathematical equations.

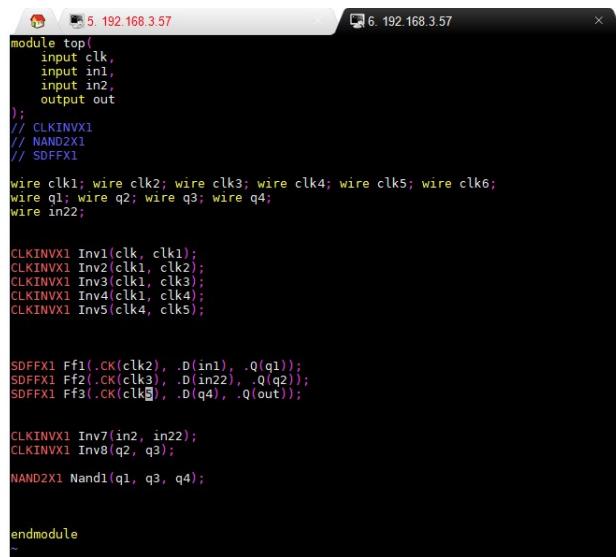
Having a capture flip flop been negative edge triggered would result in the problem being more elaborate. Since the two subsequent flip-flops are triggered at opposite edges, we now no longer deal with T\_period in setup, moreover, we need to consider T\_period/2 during hold as well.

Both these conditions result in us having to develop new sets of logic and equations to do setup and hold analysis.

As shown in the toy example, during setup checks we now no longer compare T\_data with T\_clk, we do it with T\_clk/2. Since the term from which we subtract T\_data has reduced itself (Clk\_Time has reduced) this results in Setup slack reducing.

In the hold calculation, normally we do not consider T\_period, because the hold violation is checked on the same clock edge. However, here that is not possible. Here we need to check hold violation at opposite clock edges which ideally would have T\_period/2 time gap between them. This results in an extra positive quantity being introduced and hence the slack increases.

### New Top file (Netlist)



```
5. 192.168.3.57 6. 192.168.3.57
module top(
    input clk,
    input in1,
    input in2,
    output out
);
// CLKINVX1
// NAND2XI
// SDFFX1

wire clk1; wire clk2; wire clk3; wire clk4; wire clk5; wire clk6;
wire q1; wire q2; wire q3; wire q4;
wire in22;

CLKINVX1 Inv1(.clk(clk1), .Q(q1));
CLKINVX1 Inv2(.clk1, clk2);
CLKINVX1 Inv3(.clk1, clk3);
CLKINVX1 Inv4(.clk1, clk4);
CLKINVX1 Inv5(.clk4, clk5);

SDFFX1 Ff1(.CK(clk2), .D(in1), .Q(q1));
SDFFX1 Ff2(.CK(clk3), .D(in22), .Q(q2));
SDFFX1 Ff3(.CK(clk4), .D(q4), .Q(out));

CLKINVX1 Inv7(in2, in22);
CLKINVX1 Inv8(q2, q3);

NAND2XI Nand1(q1, q3, q4);

endmodule
```