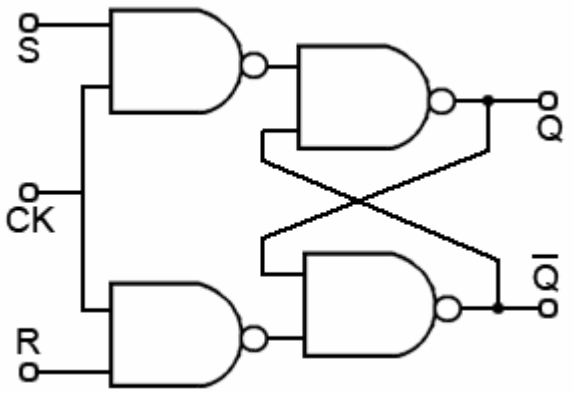
BITS-Pilani Dubai Campus
I Sem 2021-22
Digital Design Laboratory / ECE/INSTR/CS F215
Submission Report
Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Hardware runs

Run 1: Clocked SR Latch using NAND gates

Diagram



Truth Table

Clock	S	R	Q(t)	Q(t+1)	Operation
0	0	0	x	Q(t)	No change
0	0	1	x	Q(t)	No change
0	1	0	x	Q(t)	No change
0	1	1	x	Q(t)	No change
1	0	0	0/1	0/1	Hold
1	0	1	1	0	Set
1	1	0	0	1	Reset
1	1	1	-	-	Invalid

Q: What is the disadvantage with an SR latch?

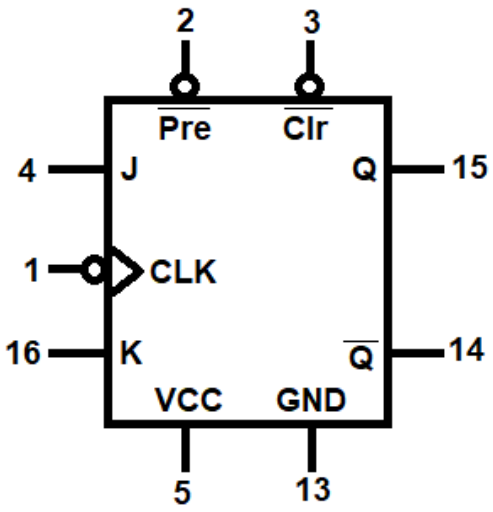
Disadvantage of SR latch is that S=1, R=1 has no output/real value. when the inputs are both 1, the outputs of both the NOR gates are 1. This is not right because the outputs are supposed to be complements of each other. Hence this particular configuration is not allowed resulting in wastage of one combination value.

Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Run 2: JK Flip Flop

Diagram



Truth Table

J	K	Pre	Clr	Clock	Q(t)	Q(t+1)	Operation
x	x	0	0	x	1	1	Invalid
x	x	0	1	x	x	1	Preset
x	x	1	0	x	x	0	Clear
0	0	1	1	-ve	x	Q(t)	No change
0	1	1	1	-ve	x	0	Reset
1	0	1	1	-ve	x	1	Set
1	1	1	1	-ve	x	Q'(t)	Toggle

Q: Are the preset and clear inputs synchronous or asynchronous.

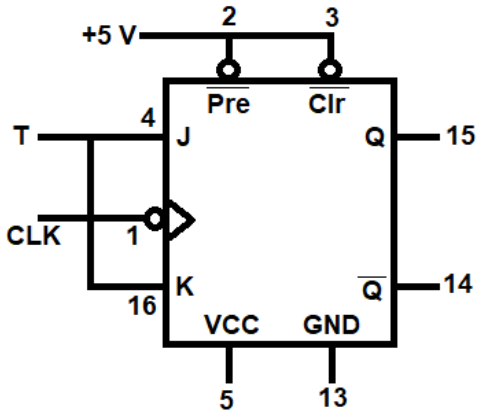
Asynchronous inputs on a flip flop have control over the outputs (Q and Q') regardless of clock input status. Preset and Clear do not depend upon the clock signals and are thus Asynchronous in nature.

Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Run 3: T Flip Flop (Using JK Flip Flop)

Diagram



Truth Table

T	Clock	Q(t)	Q(t+1)
0	0	x	Q(t)
1	0	x	Q(t)
0	-ve	x	Q(t)
1	-ve	x	Q'(t)

Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Software runs

Run 4: SR Latch and Flip-flop

1. Write the Verilog code for clocked SR latch as shown in run-1 of this experiment using four NAND gates (Gate level modeling). Write the testbench also for all possible scenarios and also check for undefined case in waveforms when both S = R = '1'.

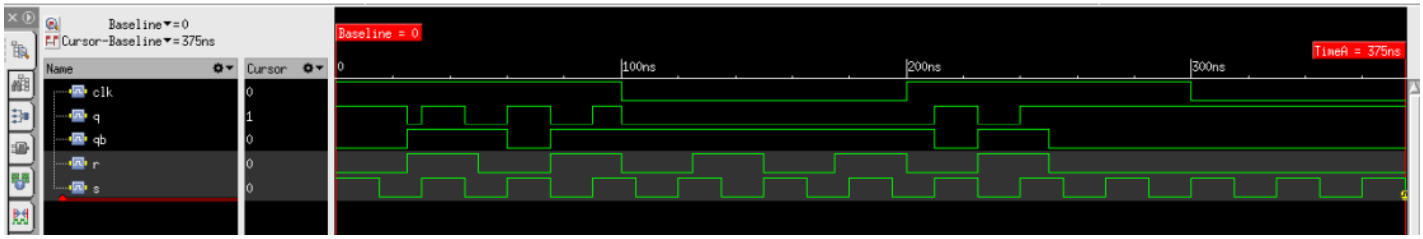
A: Verilog Code and testbench-

```
module run1 (clk,q,qb,s,r );
input s,r,clk;
output q,qb;
wire so,ro;
nand g1 (so,s,clk), g2 (ro,r,clk), g3 (q,so,qb), g4 (qb,ro,q);
endmodule

//testbench
module testbench();
reg r,s,clk;
wire q,qb;
initial begin
clk = 1'b1;
repeat(3) #100 clk = ~clk;
end
initial begin
r = 1'b0;
repeat(10) #25 r = ~r;
end
initial begin
s = 1'b1;
repeat(25) #15 s = ~s;
end
run1 hello (clk,q,qb,s,r);
endmodule
```

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.

A:



Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

2. Write Verilog code and testbench for clocked JK flip-flop and compare their response in waveforms. (please use exhaustive testbench).

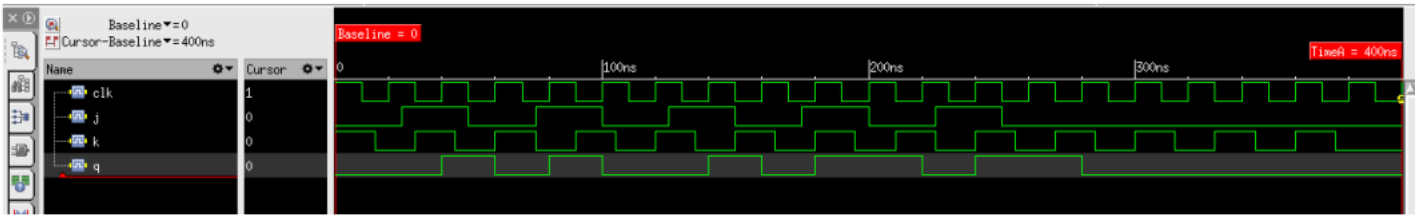
A: Verilog Code and testbench-

```
module run2 (input j, k, clk, output reg q);
always @ (posedge clk) begin
case ({j, k})
2'b00 : q <= q;
2'b01 : q <= 0;
2'b10 : q <= 1;
2'b11 : q <= ~q;
endcase
end
endmodule

//testbench
module testbench();
reg j,k,clk;
wire q;
initial begin
clk = 1'b1;
repeat(40) #10 clk = ~clk;
end
initial begin
j = 1'b0;
repeat(10) #25 j = ~j;
end
initial begin
k = 1'b1;
repeat(25) #15 k = ~k;
end
run2 h1 (j,k,clk,q);
endmodule
```

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.

A:



Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Run 5: D-Flipflop

1. Write Verilog code and testbench for positive edge triggered D-Flip-flop with asynchronous set and reset.

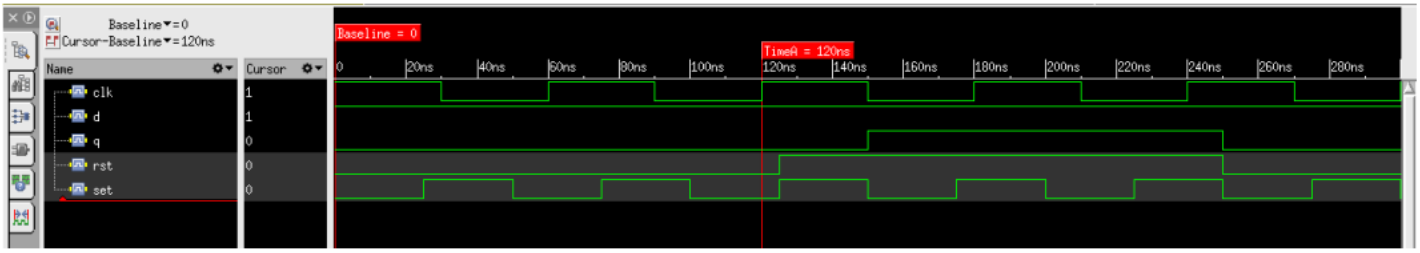
A: Verilog Code and testbench-

```
module run55 ( input d, set_b, rst_b, clk, output reg q);
always @ (posedge clk, negedge set_b, negedge rst_b) begin
if (rst_b == 1'b0)
q <= 0;
else if (set_b == 1'b0)
q <=1;
else q <= d;
end
endmodule

//testbench
module testbench();
reg d,set,rst,clk;
wire q;
initial begin
clk = 1'b1;
repeat(40) #30 clk = ~clk;
end
initial begin
set = 1'b0;
repeat(20) #25 set = ~set;
end
initial begin
rst = 1'b0;
repeat(4) #125 rst = ~rst;
end
initial begin
d = 1'b1;
repeat(20) #25 d = d;
end
run55 g1(d,set,rst,clk,q);
endmodule
```

Q: Paste the Image of your **Simvision** window where you get the waveforms for the above code.

A:



Experiment No.- 7 (Latches and Flipflops)

Name: V N Suchir Vangaveeti **ID Number: 2020A7PS0018U**

Assignment All assignments are to be submitted strictly before start of next lab session through online only. Late assignments will not be entertained and will be awarded '0' marks.

1. Verilog code and testbench for T Flip-Flop for positive edge triggered.
Ans: Link1: https://www.edaplayground.com/x/u7ER

2. Identify the logic from the Verilog code below. (hint: Create testbench to identify).

```
module circuit_1 (input A,B, output C);
assign C = A ? B : C;
// ? : is the conditional operator (e.g. w=x ? y : z ; if x=true, then w=y if x =false then w=z)
endmodule
```

Ans: Link2: https://www.edaplayground.com/x/X3sN
The logic is JK Flip Flop Logic

Self-Practice and self-evaluation
1. Verilog code and testbench for D Flip-Flop for negative edge triggered.
2. Identify the logic for the code below by writing the testbench

```
module circuit_2 (input D_in, en, rst, output q);
    assign q = !(rst ==1'b0) ? 0 : en ? D_in : q;
endmodule
```

3. Identify the logic for the code below by writing the testbench

```
module circuit_2 (output reg q, input d, en);
    always @ (en, d)
    if (en ==1'b1) q &lt;= d;
endmodule
```

4. Identify the logic for the code below by writing the testbench

```
module circuit_3 (q, q_bar, d, set, rst, clk);
    input d, set, rst, clk;
    output reg q;
    output q_bar;
    assign q_bar = !q;
    always @ (posedge clk) // code enters here only at rising edge or positive edge of clock
    // this also makes set and reset signals synchronous to clock edge only
    if (rst == 1'b0) q &lt;= 0; // operator '&lt;=' is the non-blocking assignment operator
    else if (set == 1'b0) q&lt;=1;
    else q &lt;= d;
endmodule
```