30 Days of RTL Coding

-By T Tushar Shenoy

Day 25

Problem Statement: Implementing Positive/Negative edge Triggered D Flip-Flops using NAND Gates (Gate Level Implementation).

Theory:

Triggering

Triggering means making the circuit active [allow it to receive inputs].Triggering makes a circuit synchronous, triggering is given in the form of clock or gate signal Depending on the type of triggering used circuit becomes active at specific states of clock pulse

–Level Triggering: Circuit becomes active when gating or clock pulse is at a particular level.

–Edge Triggering: Circuit becomes active at the negative edge or positive edge of the clock signal.

D Flip Flop

D flip flop is an electronic devices that is known as “delay flip flop” or “data flip flop” which is used to store single bit of data. D flip flops are synchronous or asynchronous. The clock signal required for the D flip flops. The D flip flop has two inputs, data and clock input which controls the flip flop. During Positive Edge of clock (Positive Edge Triggered), the data is transferred to the output of the flip flop otherwise, the output of the flip flop is held in its previous state.

Positive Edge Triggered D Flip-Flop

Inputs		Outputs	
D	C	Q^+	$\overline{Q}^+$
0	↑	0	1
1	↑	1	0
X	0	Q	$\overline{Q}$
X	1	Q	$\overline{Q}$

FIG: Positive Edge Triggered D Flip-Flop Truth Table

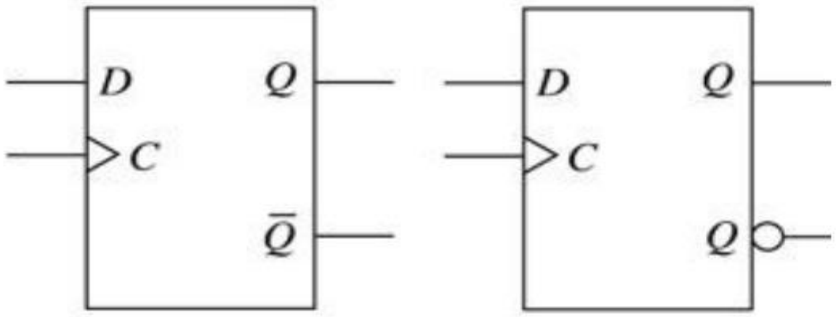


FIG: Positive Edge Triggered D Flip-Flop Symbol

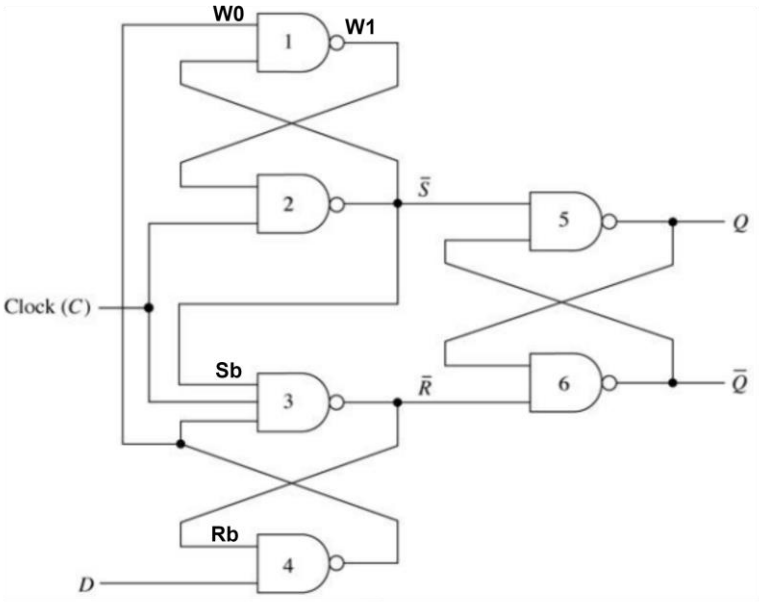


FIG: Positive Edge Triggered D Flip-Flop Using NAND Gate

Verilog Code:

```
//Verilog Code for Positive Edge Triggered D Flip-Flop
module D_FF_Pos_Trig(D,clk,Q,Qb);

input D,clk;
output Q,Qb;

wire [1:0]w;
wire Sb,Rb;

nand N1(w[1],w[0],Sb);
nand N2(Sb,w[1],clk);

nand N3(Rb,Sb,clk,w[0]);
nand N4(w[0],Rb,D);

nand N5(Q,Sb,Qb);
nand N6(Qb,Rb,Q);

endmodule
```

Testbench Code:

```
//Test bench code for Positive Edge Triggered D Flip-Flop
module D_FF_Pos_Trig_tb();

reg D,clk;
wire Q,Qb;

D_FF_Pos_Trig dut(.D(D),.clk(clk),.Q(Q),.Qb(Qb));

initial begin
clk=1'b0;
   D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#16 $finish;
end

always #5 clk=~clk;
endmodule
```

Schematic:

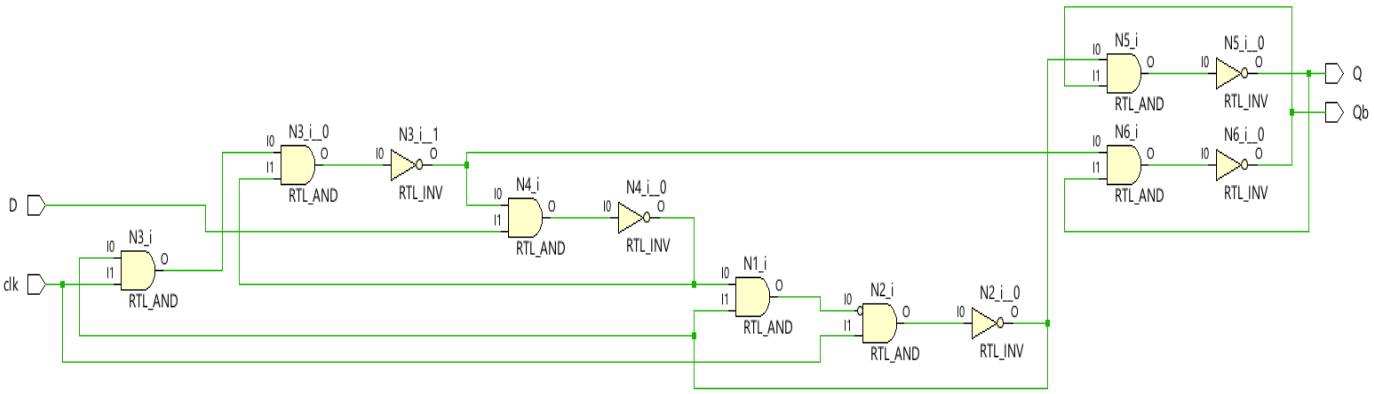
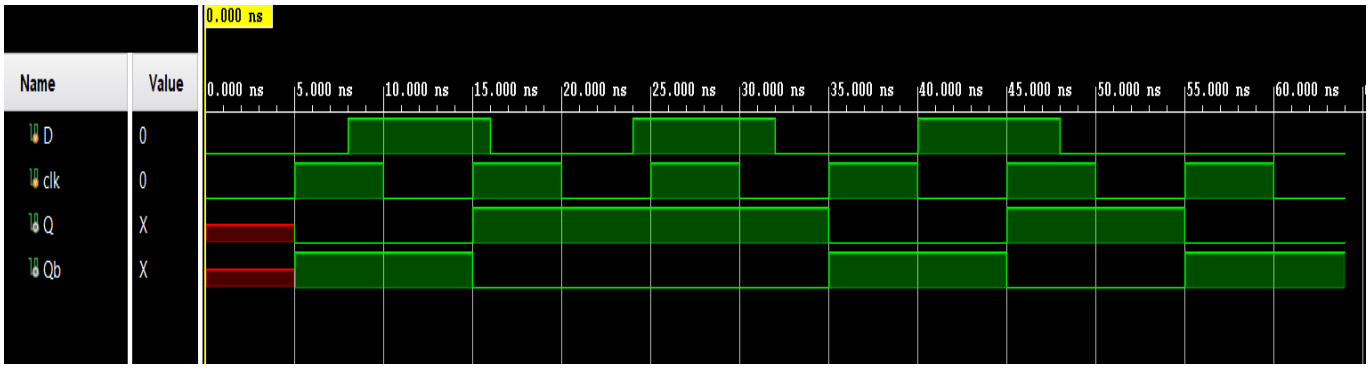


Fig: Schematic of Positive Edge Triggered D Flip-Flop

Simulation Output:



Negative Edge Triggered D Flip-Flop

Inputs		Outputs	
D	Clk	Q^+	$\overline{Q}^+$
0	↓	0	1
1	↓	1	0
X	0	Q	$\overline{Q}$
X	1	Q	$\overline{Q}$

FIG: Negative Edge Triggered D Flip Flop Truth Table

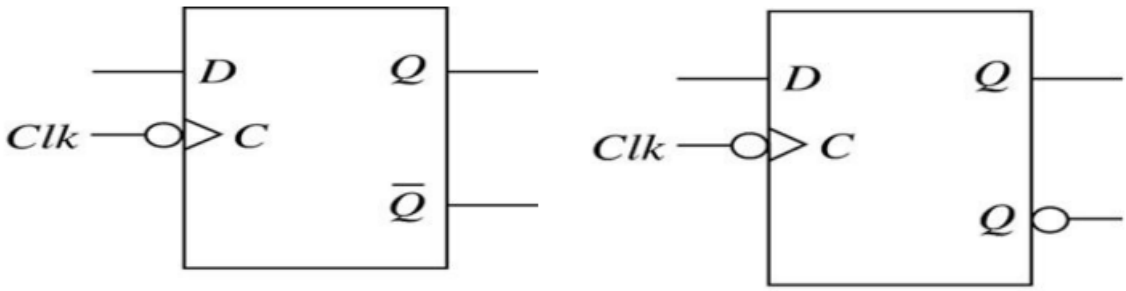


FIG: Negative Edge Triggered D Flip-Flop Symbol

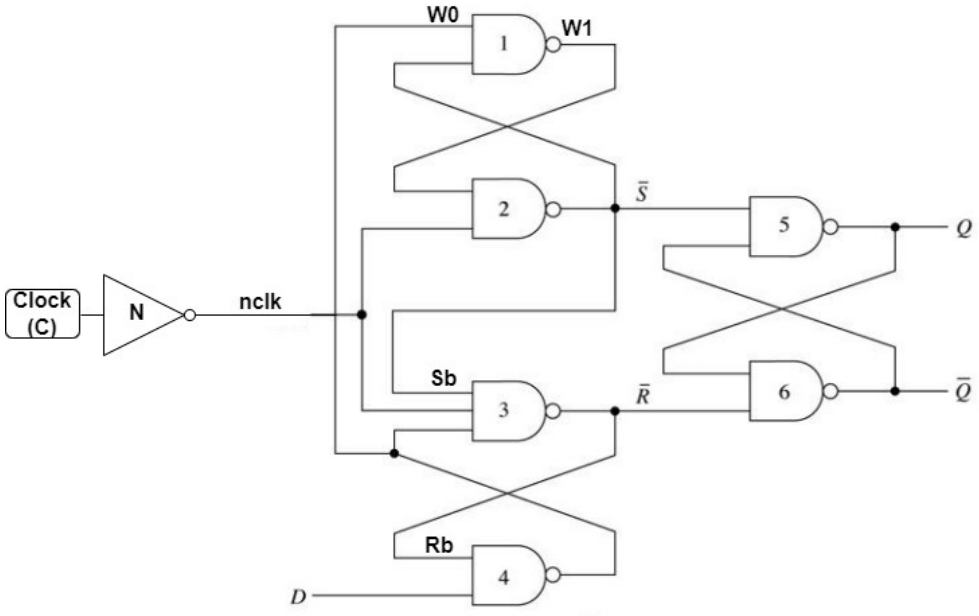


FIG: Negative Edge Triggered D Flip-Flop Using NAND Gate

Verilog Code:

```
//Verilog Code for Negative Edge Triggered D Flip-Flop
module D_FF_Neg_Trig(D,clk,Q,Qb);

input D,clk;
output Q,Qb;

wire [1:0]w;
wire nclk,Sb,Rb;

not N(nclk,clk);

nand N1(w[1],w[0],Sb);
nand N2(Sb,w[1],nclk);

nand N3(Rb,Sb,nclk,w[0]);
nand N4(w[0],Rb,D);

nand N5(Q,Sb,Qb);
nand N6(Qb,Rb,Q);

endmodule
```

Testbench Code:

```
//Test bench Code for Negative Edge Triggered D Flip-Flop
module D_FF_Neg_Trig_tb();

reg D,clk;
wire Q,Qb;

D_FF_Neg_Trig dut(.D(D),.clk(clk),.Q(Q),.Qb(Qb));

initial begin
clk=1'b0;
   D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#8 D=1'b1;
#8 D=1'b0;
#16 $finish;
end

always #5 clk=~clk;
endmodule
```

Schematic:

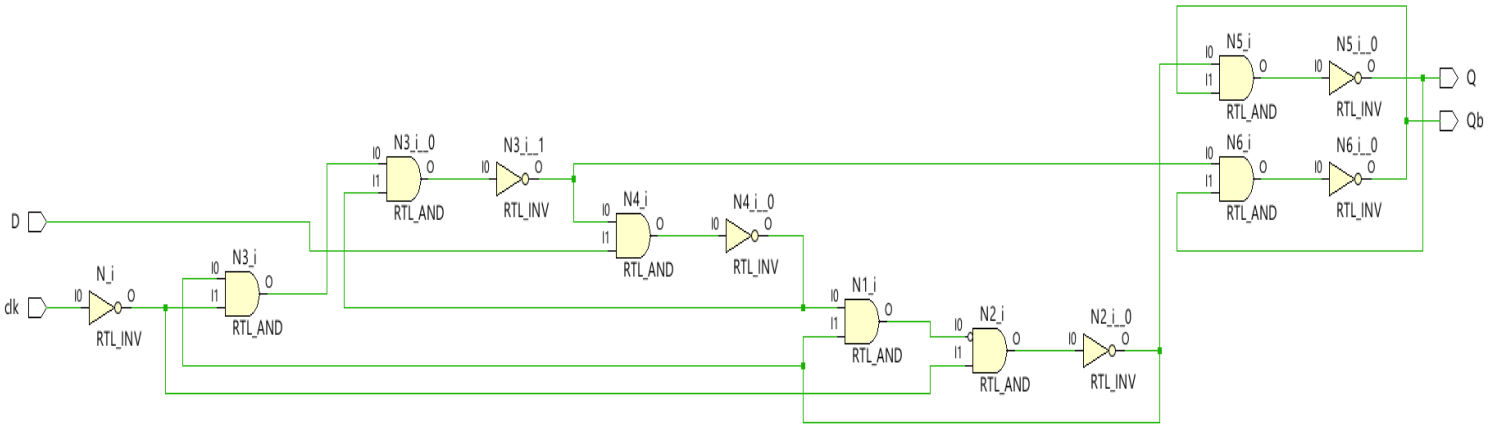
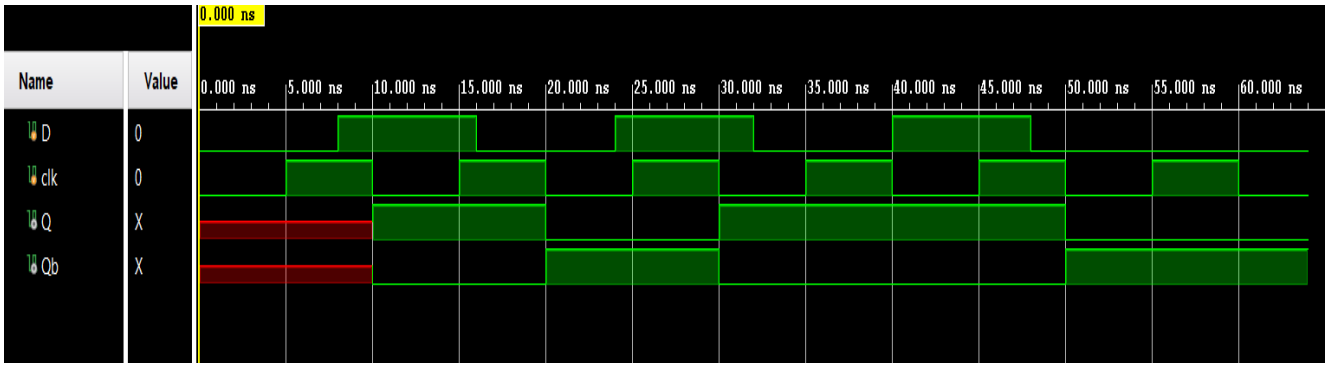


Fig: Schematic of Negative Edge Triggered D Flip-Flop

Simulation Output:



GitHub Repository URL: https://github.com/tusharshenoy/RTL-Day-25-D_Flip_Flop_Gate_Level