30 Days of RTL Coding

-By T Tushar Shenoy

Day 15

Problem Statement: Implementing a Master Slave JK Flip Flop.

Theory:

A master-slave flip flop is made by connecting two JK flip flops in a series configuration in which one acts as the master and another as a slave. The two inputs of slave are connected with the output of the master flip flop. Furthermore, the master flip flop inputs are fed back by the output of the slave flip flop. Apart from two flip-flops, it has one inverter. Here inverter and clock pulse are connected to ensure that the slave flip flop gets an inverted clock pulse. If CP =1 for the master flip flop, CP=0 for slave flip flop, and vice versa.

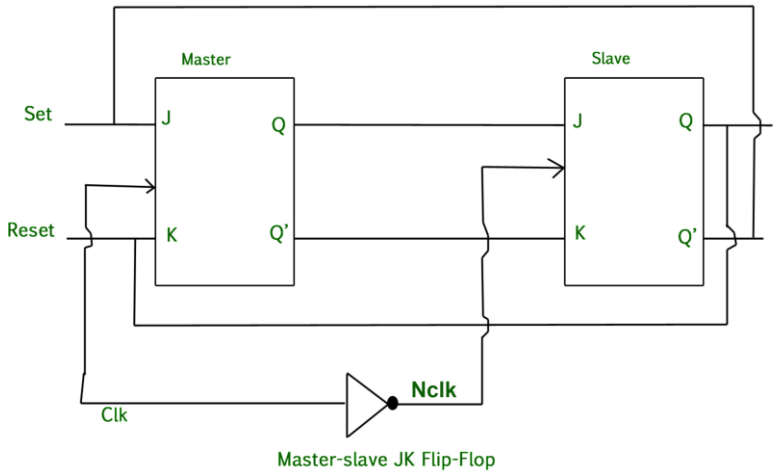


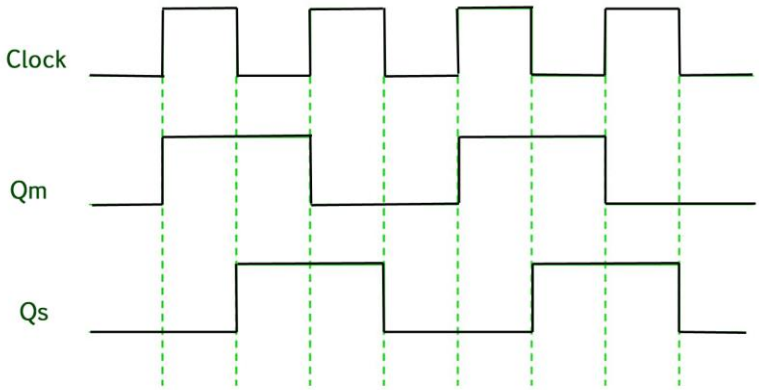
FIG: Master Slave JK Flip Flop Block Diagram

Working of a master slave flip flop –

1. When the clock pulse goes to 1, the slave is isolated; J and K inputs may affect the state of the system. The slave flip-flop is isolated until the CP goes to 0. When the CP goes back to 0, information is passed from the master flip-flop to the slave and output is obtained.
2. Firstly the master flip flop is positive level triggered and the slave flip flop is negative level triggered, so the master responds before the slave.

3. If J=0 and K=1, the high Q' output of the master goes to the K input of the slave and the clock forces the slave to reset, thus the slave copies the master.
4. If J=1 and K=0, the high Q output of the master goes to the J input of the slave and the Negative transition of the clock sets the slave, copying the master.
5. If J=1 and K=1, it toggles on the positive transition of the clock and thus the slave toggles on the negative transition of the clock.
6. If J=0 and K=0, the flip flop is disabled and Q remains unchanged.

Timing Diagram of a Master Slave flip flop –



1. When the Clock pulse is high the output of master is high and remains high till the clock is low because the state is stored.
2. Now the output of master becomes low when the clock pulse becomes high again and remains low until the clock becomes high again.
3. Thus toggling takes place for a clock cycle.
4. When the clock pulse is high, the master is operational but not the slave thus the output of the slave remains low till the clock remains high.
5. When the clock is low, the slave becomes operational and remains high until the clock again becomes low.
6. Toggling takes place during the whole process since the output is changing once in a cycle.

This makes the Master-Slave J-K flip flop a Synchronous device as it only passes data with the timing of the clock signal.

Verilog Code:

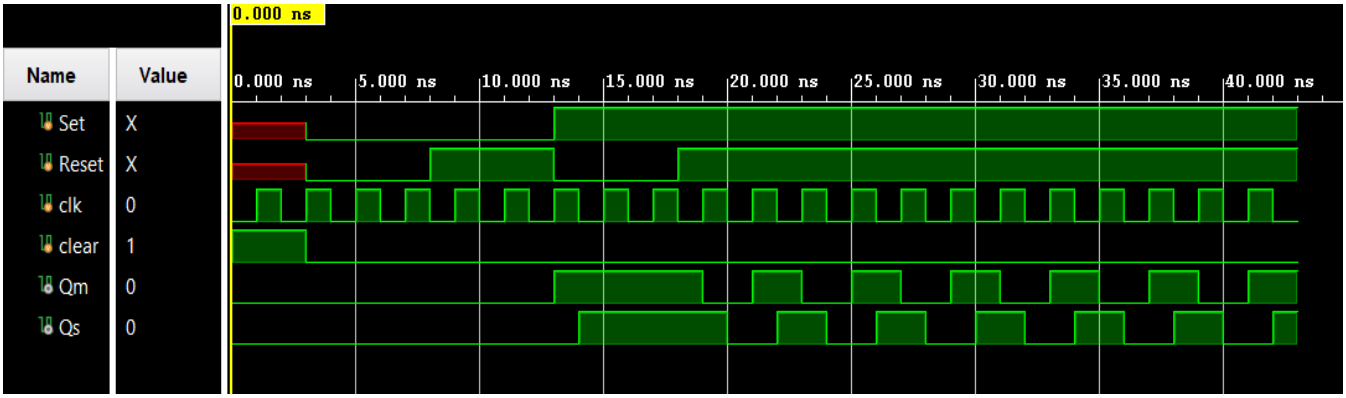
```
//JK Flip Flop Code Using Behavioural modelling
module JK_Flip_Flop(J,K,clk,reset,Q,Qb);

input J,K,clk,reset;
output reg Q,Qb;

always@(posedge clk,posedge reset)
begin
if(reset)
  Q=1'b0;
  else
  begin
      if(J==1'b0&&K==1'b0)
         Q=Q;
      else if(J==1'b0&&K==1'b1)
         Q=1'b0;
      else if(J==1'b1&&K==1'b0)
         Q=1'b1;
      else
         Q=~Q;
   end
  Qb<=~Q;
end

endmodule
```

```
//Verilog code for Master Slave JK Flip Flop
module Master_Slave_JK(Set,Reset,clk,clear,Qm,Qmb,Qs,Qsb);

input Set,Reset,clk,clear;
output Qm,Qmb,Qs,Qsb;

JK_Flip_Flop
JK1(.J(Set),.K(Reset),.clk(clk),.reset(clear),.Q(Qm),.Qb(Qmb));

wire nclk;
not(nclk,clk);

JK_Flip_Flop
JK2(.J(Qm),.K(Qmb),.clk(nclk),.reset(clear),.Q(Qs),.Qb(Qsb));
endmodule
```

Testbench Code:

```
//Testbench code fot Master Slave Jk Flip Flop
module Master_Slave_JK_tb();

reg Set,Reset,clk,clear;
wire Qm,Qs;

Master_Slave_JK
dut(.Set(Set),.Reset(Reset),.clk(clk),.clear(clear),.Qm(Qm),.Qs(Qs));

initial begin
clear=1'b1;
clk=1'b0;
#3 clear=1'b0;
    Set=1'b0;Reset=1'b0;
#5  Set=1'b0;Reset=1'b1;
#5  Set=1'b1;Reset=1'b0;
#5  Set=1'b1;Reset=1'b1;
#25  $finish;
end

always #1 clk=~clk;
endmodule
```

Simulation Output:



GitHub Repository URL: https://github.com/tusharshenoy/RTL-Day-15-JK-Master-Slave-Flip-Flop