**HDL code for Flipflop**

**D-FF: Source Code**

module d\_ff(input D, clk, output Q);

reg Q;

always @(posedge clk)

begin

Q <= D;

end

endmodule

**D-FF: Testbench Code**

module tb\_d\_ff();

reg d;

reg clk;

wire q;

d\_ff d1(d,clk,q);

initial begin

d=0;

clk=0;

end

always #50 clk=~clk;

initial begin

#100 d=1'b0;

#100 d=1'b1;

#100 d=1'b1;

#100 d=1'b1;

#100 d=1'b0;

#100

$stop;

end

endmodule

Result:



……………………………………………………………………………….

**D-FF with reset : Source code**

module d\_ff\_rst(input D, clk, clear, output Q);

reg Q;

always @(posedge (clk))

begin

if(~clear)begin

Q=1'b0;

end

else

begin

Q <= D ;

end

end

endmodule**D-FF with reset : Testbench code**

module tb\_d\_ff\_rst();

reg d;

reg clk;

reg clear;

wire q;

d\_ff\_rst d2(d,clk,clear,q);

initial begin

d=1;

clear=0;

clk=1;

end

always #50 clk=~clk;

initial begin

#100 d=1'b0;clear=1;

#100 d=1'b1;clear=1;

#100 d=1'b1;clear=1;

#100 d=1'b1;clear=1;

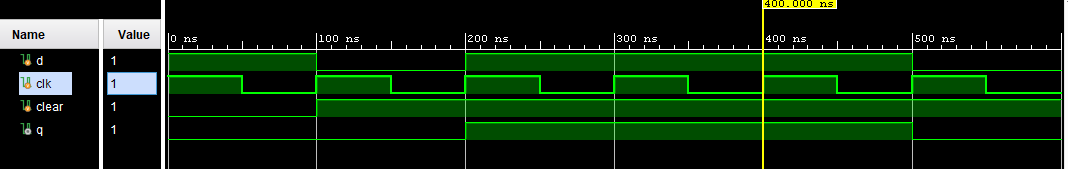
#100 d=1'b0;clear=1;

#100

$stop;

end

endmodule



**T-FF: source code**

module t\_ff(input T, clk, clear, output Q);

reg Q;

always @(T, posedge (clk))

begin

if(~clear)begin

Q=1'b0;

end

else

begin

Q <= T^Q ;

end

end

endmodule

**T-FF: Testbench code**

module tb\_t\_ff();

reg T;

reg clk;

reg clear;

wire q;

t\_ff t1(T,clk,clear,q);

initial begin

T=1;

clear=0;

clk=1;

end

always #50 clk=~clk;

initial begin

#100 T=1'b0;clear=1;

#100 T=1'b1;clear=1;

#100 T=1'b1;clear=1;

#100 T=1'b1;clear=1;

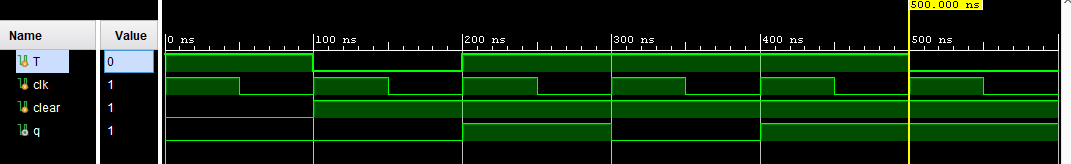
#100 T=1'b0;clear=1;

#100

$stop;

end

endmodule



**JK-FF: source code:**

module jk\_ff(input j, k , clk, clear, output Q);

reg Q;

always @(posedge (clk))

begin

if(~clear)begin

Q=1'b0;

end

else

begin

case ({j,k})

2'b00: Q<=Q;

2'b01: Q<=0;

2'b10: Q<=1;

2'b11: Q =~Q;

endcase

end

end

endmodule

**JK-FF: Testbench code:**

module tb\_jk\_ff();

reg j,k;

reg clk;

reg clear;

wire q;

jk\_ff jk1(j,k,clk,clear,q);

initial begin

j=1;k=0;

clear=0;

clk=1;

end

always #50 clk=~clk;

initial begin

#100 j=1'b0;k=1'b0;clear=1;

#100 j=1'b0;k=1'b1;clear=1;

#100 j=1'b1;k=1'b0;clear=1;

#100 j=1'b1;k=1'b1;clear=1;

#100 j=1'b1;k=1'b1;clear=1;

#100 j=1'b1;k=1'b1;clear=1;

#100

$stop;

end

endmodule

