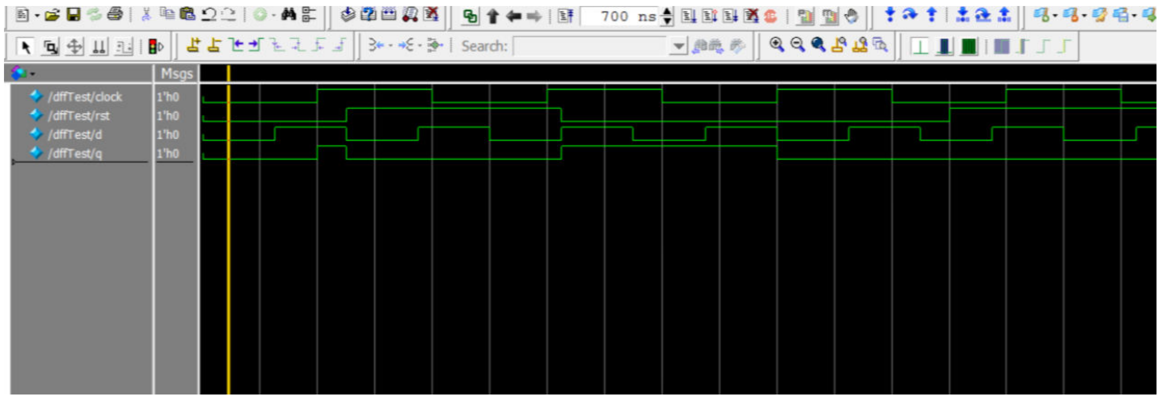
9.6.1 TASK 1: D flip flop

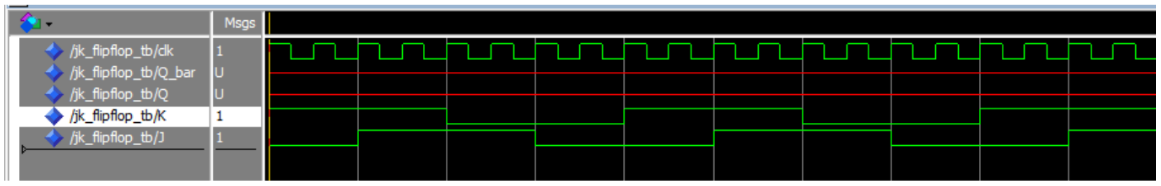
```
module DFlipFlop (D, Clk, Q, Qnot);
    input D, Clk;
    output reg Q;
    output Qnot;
    always @(posedge Clk) begin
        Q <= D;
    end
    assign Qnot = ~Q;
endmodule


module DFlipFlop_tb;
    reg D, Clk;
    wire Q, Qnot;
    DFlipFlop uut (.D(D), .Clk(Clk), .Q(Q), .Qnot(Qnot));
    initial begin
        Clk = 0;
        forever #5 Clk = ~Clk;
    end
    initial begin
        D = 0; #12
        D = 1; #10
        D = 0; #15
        D = 1; #20
    end
endmodule
```



9.6.2 TASK 2: JK flip flop

```
module D_flip_flop (D,Clk,Q);

    input D;

    input Clk;

    output reg Q;

    always @(posedge Clk) begin

        Q <= D;

    end

endmodule


module JK_flip_flop (

    input J,

    input K,

    input Clk,

    output Q

);

    wire D;

    assign D = (J & ~Q) | (~K & Q);

    D_flip_flop dff ( .D(D), .Clk(Clk), .Q(Q));

endmodule
```

```
module JK_flip_flop_tb;

    reg J, K, Clk;

    wire Q;

    JK_flip_flop jkff ( .J(J), .K(K), .Clk(Clk), .Q(Q));

    initial begin

        Clk = 0;

        Clk = ~Clk;

    end

    initial begin

        J = 0; K = 0;

        #10 J = 0; K = 1;

        #10 J = 1; K = 0;

        #10 J = 1; K = 1;

        #10 J = 0; K = 0;

        #10 J = 1; K = 0;

        #10 J = 0; K = 1;

        #10 J = 1; K = 1;

    end

endmodule
```



```
                     Msgs
/jk_flipflop_tb/clk    1
/jk_flipflop_tb/Q_bar  U
/jk_flipflop_tb/Q      U
/jk_flipflop_tb/K      1
/jk_flipflop_tb/J      1
```