Digital Design and Computer Organisation Laboratory

3rd Semester, Academic Year 2025

Date:28-09-2025

Name: Prajwal M	SRN:PES2UG24CS910	Section:C

Week Number: 5 Program Number: 5

TITLE:IMPLEMENTAION OF FLIP FLOPS

1. Design and implement D flipflop

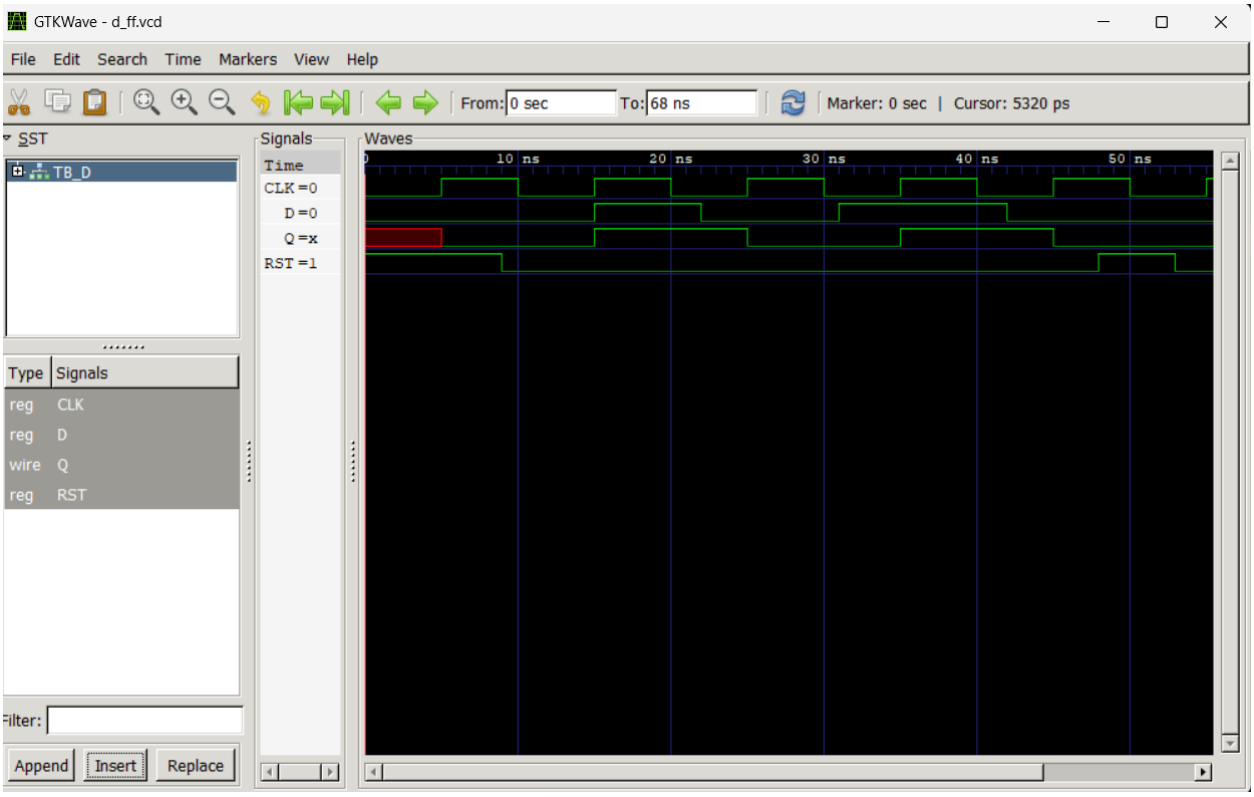
Dflipflop.v

```
1   module d_ff (input wire D, input wire clk, input wire rst, output reg Q);
2     always @(posedge clk)    "posedge": Unknown word.
3   begin
4     if (rst)      Q <= 1'b0;     // Asynchronous reset to 0
5    else      Q <= D;        // Sample D on rising edge of clock
6     end
7   endmodule    "endmodule": Unknown word.
8
```

d_using_t_tb.v

```
1   module TB_D_using_T;
2   reg D;
3   reg CLK;
4   reg RST;
5   wire Q;
6   d_using_t_ff newD (.D(D), .clk(CLK), .rst(RST), .Q(Q));
7   initial begin
8     CLK=1'b0;
9     forever #5 CLK=~CLK;
10  end
11  initial begin
12    $dumpfile("d_using_t.vcd");    "dumpfile": Unknown word.
13    $dumpvars(0, TB_D_using_T);    "dumpvars": Unknown word.
14  end
15  initial begin
16    $monitor("Time=%0t | RST=%b | D=%b | Q=%b", $time, RST, D, Q);
17  end
18  initial begin
19    RST=1'b1; D=1'b0; #8;
20    RST=1'b0; #6;
21    D=1'b0; #15;  // Q should follow D
22    D=1'b1; #10;  // Q should become 1
23    D=1'b0; #10;  // Q should become 0
24    D=1'b1; #10;  // Q should become 1
25    D=1'b1; #10;  // Q should stay 1
26    D=1'b0; #10;  // Q should become 0
27    RST=1'b1; #5;
28    RST=1'b0; #15;
29    $finish;
30  end
31  endmodule    "endmodule": Unknown word.
32
```

```
D:\iverilog lab\week 6>iverilog -o Dflipflop Dflipflop.v Dflipflop_tb.v

D:\iverilog lab\week 6>vvp Dflipflop
VCD info: dumpfile d_ff.vcd opened for output.
At time                    0, D = 0, RST = 1, Q = x
At time                 5000, D = 0, RST = 1, Q = 0
At time                 9000, D = 0, RST = 0, Q = 0
At time                15000, D = 1, RST = 0, Q = 1
At time                22000, D = 0, RST = 0, Q = 1
At time                25000, D = 0, RST = 0, Q = 0
At time                31000, D = 1, RST = 0, Q = 0
At time                35000, D = 1, RST = 0, Q = 1
At time                42000, D = 0, RST = 0, Q = 1
At time                45000, D = 0, RST = 0, Q = 0
At time                48000, D = 0, RST = 1, Q = 0
At time                53000, D = 0, RST = 0, Q = 0

D:\iverilog lab\week 6>gtkwave d_ff.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[68000] end time.
```



2. Design and implement T flip flop

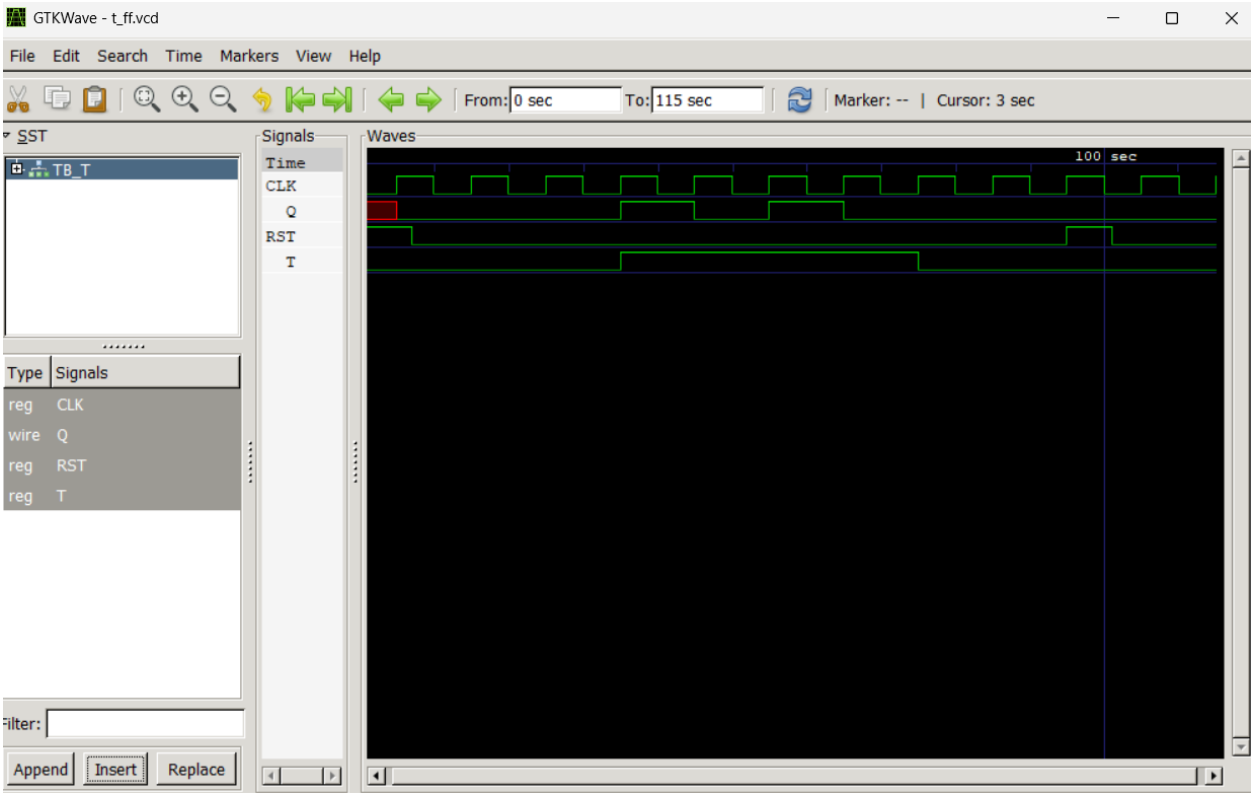
tflipflop.v

```v
module t_ff (input wire T, input wire clk, input wire rst, output reg Q);

always @(posedge clk )    "posedge": Unknown word.
    begin  if (rst)
                Q <= 1'b0;
            else if (T)    Q <= ~Q;   // toggle
     else    Q <= Q;     // hold
    end
    endmodule    "endmodule": Unknown word.

```

tflipflop_tb.v

```v
module TB_T;
 reg T;
reg CLK;
  reg RST;
wire Q;
 // Instantiate T flip-flop
 t_ff newT (.T(T), .clk(CLK), .rst(RST), .Q(Q));
  // Clock generation: 10 time units period
  initial begin
    CLK = 1'b0;
    forever #5 CLK = ~CLK;
  end
// Waveform dump
 initial
begin
 $dumpfile("t_ff.vcd");      "dumpfile": Unknown word.
 $dumpvars(0, TB_T);     "dumpvars": Unknown word.
end
 // Monitor prints automatically whenever any signal changes
initial
begin
 $monitor("Time=%0t | RST=%b | T=%b | Q=%b", $time, RST, T, Q);
  end
 // Stimulus
  initial
 begin
   RST = 1'b1;
T = 1'b0; #7;
   RST = 1'b0; #8;
  T = 1'b0; #20;  // hold
   T = 1'b1; #40;  // toggle
   T = 1'b0; #20;  // hold
 RST = 1'b1; #6;
 RST = 1'b0; #14;
$finish;
end

```

```
D:\iverilog lab\week 6>iverilog -o tflipflop tflipflop.v tflipflop_tb.v

D:\iverilog lab\week 6>vvp tflipflop
VCD info: dumpfile t_ff.vcd opened for output.
Time=0 | RST=1 | T=0 | Q=x
Time=5 | RST=1 | T=0 | Q=0
Time=7 | RST=0 | T=0 | Q=0
Time=35 | RST=0 | T=1 | Q=1
Time=45 | RST=0 | T=1 | Q=0
Time=55 | RST=0 | T=1 | Q=1
Time=65 | RST=0 | T=1 | Q=0
Time=75 | RST=0 | T=0 | Q=0
Time=95 | RST=1 | T=0 | Q=0
Time=101 | RST=0 | T=0 | Q=0

D:\iverilog lab\week 6>gtkwave t_ff.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[115] end time.
```



3. Design and implement JK flipflop

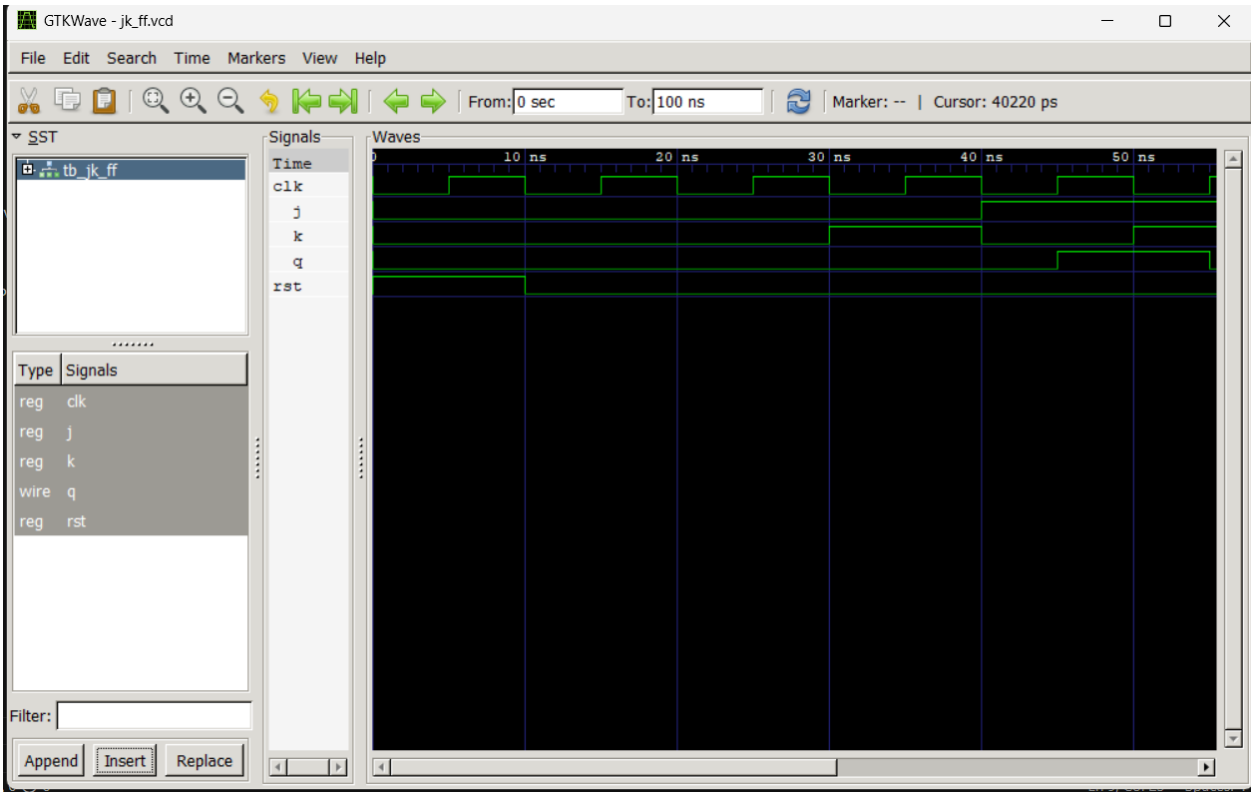
jkflipflop.v

```
module jk_ff (input wire j, input wire k, input wire clk, input wire rst, output reg q);
always @(posedge clk or posedge rst)    "posedge": Unknown word.
begin
 if (rst)
    q <= 1'b0;              // Reset to 0
 else begin
    case ({j, k})
      2'b00: q <= q;        // No change
      2'b01: q <= 1'b0;     // Reset
      2'b10: q <= 1'b1;     // Set
      2'b11: q <= ~q;       // Toggle
    endcase      "endcase": Unknown word.
 end
end
endmodule    "endmodule": Unknown word.

```

jkflipflop_tb.v

```
`timescale 1ns/1ps
module tb_jk_ff;
  reg j, k, clk, rst;
  wire q;
  jk_ff jk1 (
    .j(j),
    .k(k),
    .clk(clk),
    .rst(rst),
    .q(q)
  );
 initial begin
    clk = 0;
    forever #5 clk = ~clk;
  end
  initial begin
    $dumpfile("jk_ff.vcd");    "dumpfile": Unknown word.
    $dumpvars(0, tb_jk_ff);    "dumpvars": Unknown word.
  end
  initial begin
    $monitor("Time=%0t | RST=%b | J=%b K=%b | Q=%b", $time, rst, j, k, q);
    rst = 1; j = 0; k = 0;
    #10 rst = 0;
    #10 j = 0; k = 0; // Hold
    #10 j = 0; k = 1; // Reset
    #10 j = 1; k = 0; // Set
    #10 j = 1; k = 1; // Toggle
    #10 j = 1; k = 1; // Toggle again
    #10 j = 0; k = 1; // Reset
    #10 j = 1; k = 0; // Set
    #20 $finish;
  end
endmodule    "endmodule": Unknown word.


```

```
D:\iverilog lab\week 6>iverilog -o jkflipflop jkflipflop.v jkflipflop_tb.v

D:\iverilog lab\week 6>vvp jkflipflop
VCD info: dumpfile jk_ff.vcd opened for output.
Time=0 | RST=1 | J=0 K=0 | Q=0
Time=10000 | RST=0 | J=0 K=0 | Q=0
Time=30000 | RST=0 | J=0 K=1 | Q=0
Time=40000 | RST=0 | J=1 K=0 | Q=0
Time=45000 | RST=0 | J=1 K=0 | Q=1
Time=50000 | RST=0 | J=1 K=1 | Q=1
Time=55000 | RST=0 | J=1 K=1 | Q=0
Time=65000 | RST=0 | J=1 K=1 | Q=1
Time=70000 | RST=0 | J=0 K=1 | Q=1
Time=75000 | RST=0 | J=0 K=1 | Q=0
Time=80000 | RST=0 | J=1 K=0 | Q=0
Time=85000 | RST=0 | J=1 K=0 | Q=1

D:\iverilog lab\week 6>gtkwave jk_ff.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[100000] end time.
```



ASSIGNMENT QUESTIONS:

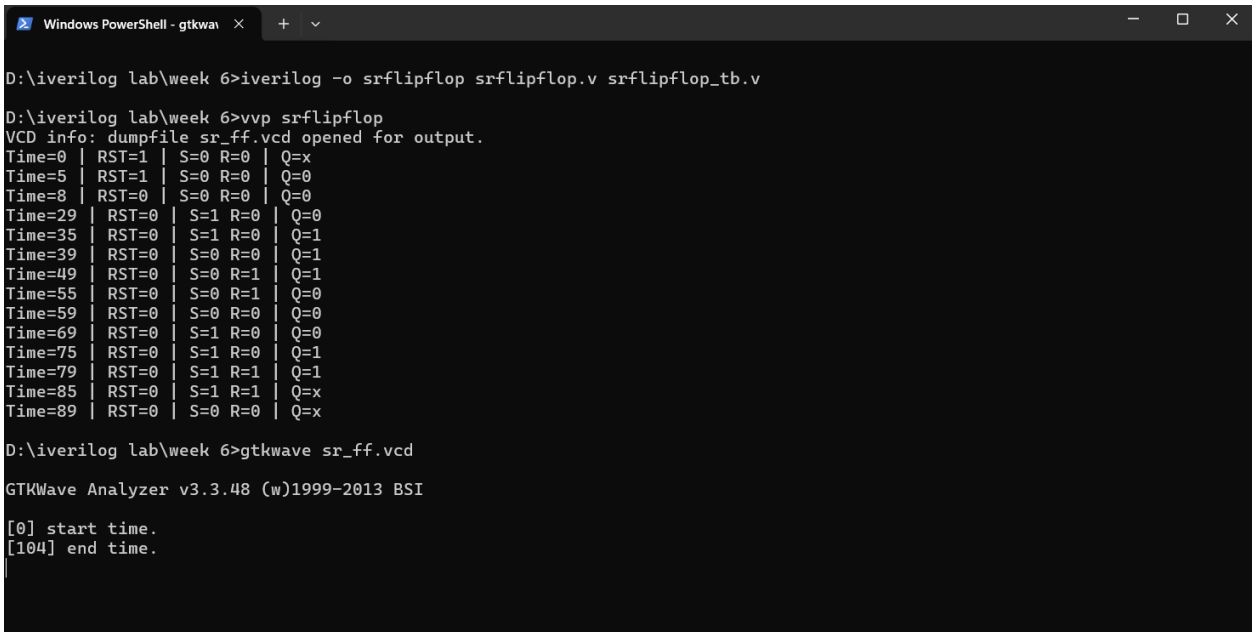
4. Implement S R flipflop

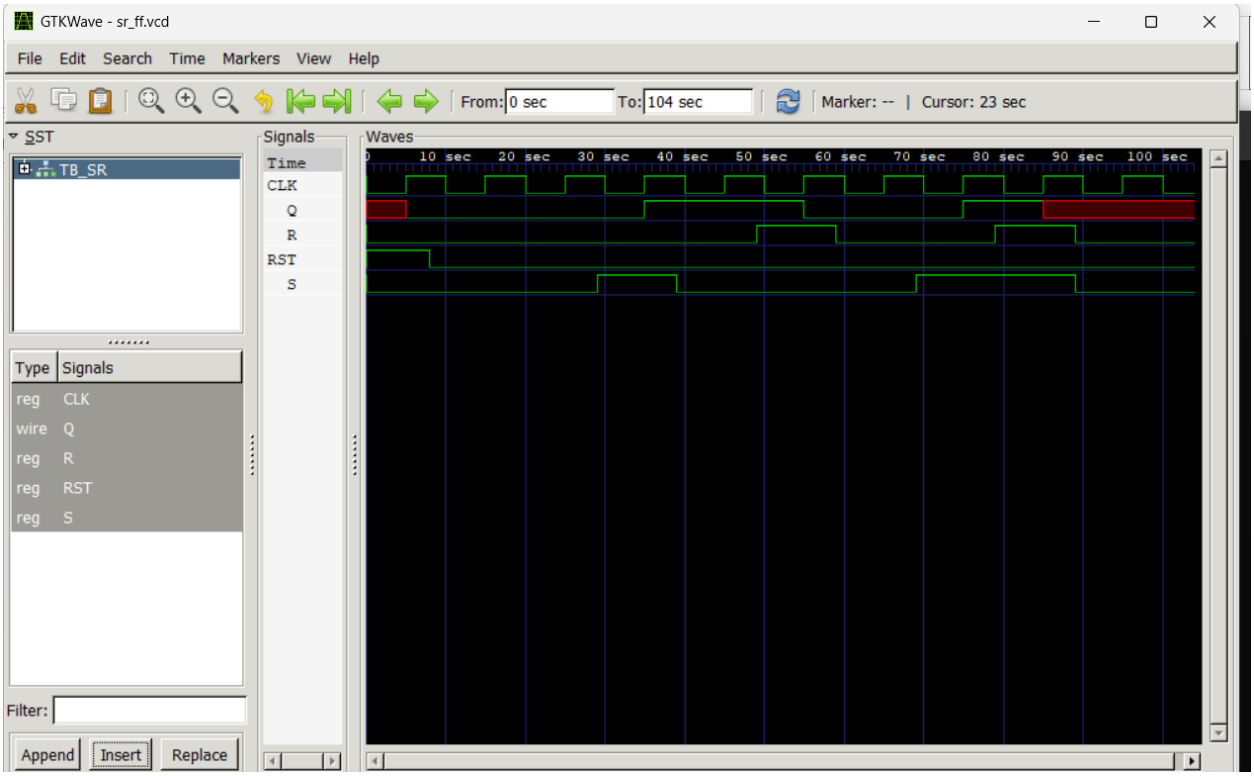
srflipflop.v

```v
module sr_ff (input wire S, input wire R, input wire clk, input wire rst, output reg Q);
always @(posedge clk)    "posedge": Unknown word.
begin
  if (rst)
    Q <= 1'b0;
  else begin
    case ({S,R})
      2'b00: Q <= Q;        // hold
      2'b01: Q <= 1'b0;     // reset
      2'b10: Q <= 1'b1;     // set
      2'b11: Q <= 1'bx;     // invalid
    endcase    "endcase": Unknown word.
  end
end
endmodule    "endmodule": Unknown word.

```

srflipflop_tb.v

```v
module TB_SR;
reg S;
reg R;
reg CLK;
reg RST;
wire Q;
sr_ff newSR (.S(S), .R(R), .clk(CLK), .rst(RST), .Q(Q));
initial begin
  CLK=1'b0;
  forever #5 CLK=~CLK;
end
initial begin
  $dumpfile("sr_ff.vcd");    "dumpfile": Unknown word.
  $dumpvars(0, TB_SR);    "dumpvars": Unknown word.
end
initial begin
  $monitor("Time=%0t | RST=%b | S=%b R=%b | Q=%b", $time, RST, S, R, Q);
end
initial begin
  RST=1'b1; S=1'b0; R=1'b0; #8;
  RST=1'b0; #6;
  S=1'b0; R=1'b0; #15;  // hold
  S=1'b1; R=1'b0; #10;  // set
  S=1'b0; R=1'b0; #10;  // hold
  S=1'b0; R=1'b1; #10;  // reset
  S=1'b0; R=1'b0; #10;  // hold
  S=1'b1; R=1'b0; #10;  // set
  S=1'b1; R=1'b1; #10;  // invalid
  S=1'b0; R=1'b0; #15;  // hold
  $finish;
end
endmodule    "endmodule": Unknown word.

```





5. To design and implement a D Flip-Flop using a T Flip-Flop in Verilog (iverilog) and verify its operation through simulation.

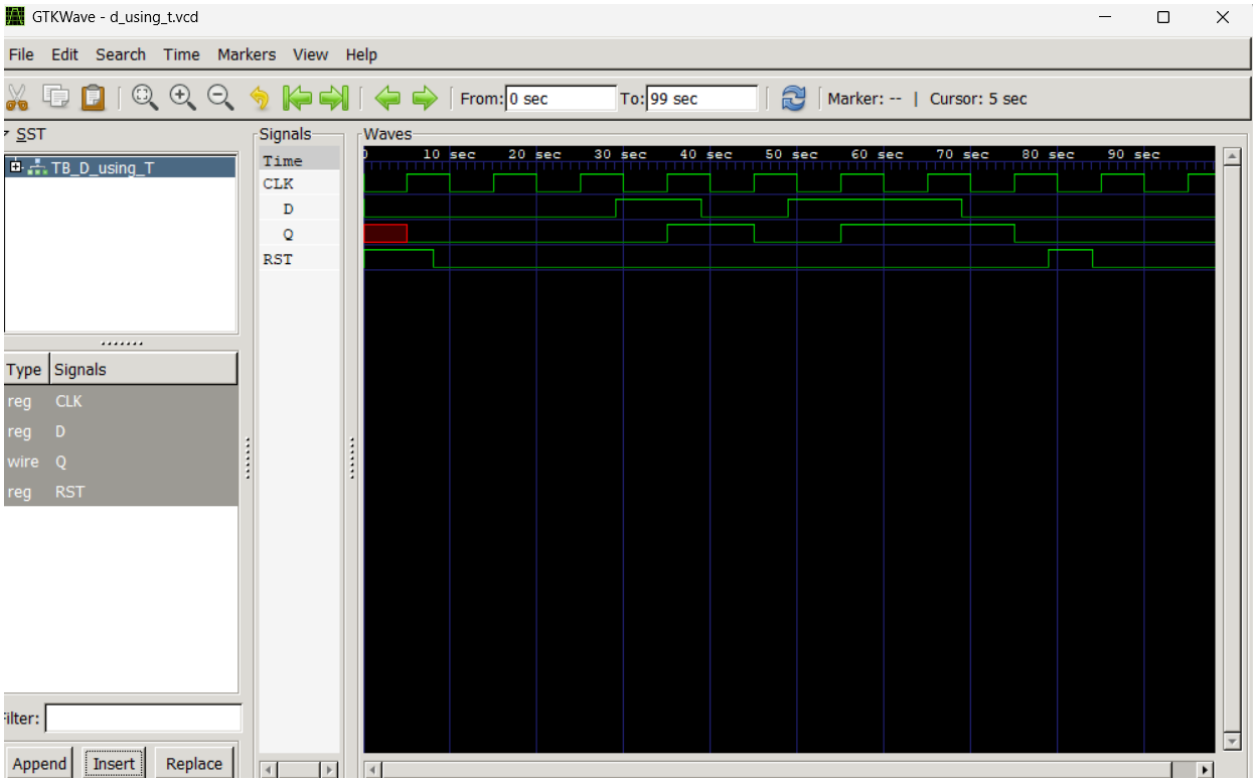
d_using_t.v

```v
module d_using_t_ff (input wire D, input wire clk, input wire rst, output wire Q);
wire T_input;
assign T_input = D ^ Q;  // XOR gate logic: T = D XOR Q
t_ff tff1 (.T(T_input), .clk(clk), .rst(rst), .Q(Q));
endmodule    "endmodule": Unknown word.

```

d_using_t_tb.v

```v
module TB_D_using_T;
reg D;
reg CLK;
reg RST;
wire Q;
d_using_t_ff newD (.D(D), .clk(CLK), .rst(RST), .Q(Q));
initial begin
  CLK=1'b0;
  forever #5 CLK=~CLK;
end
initial begin
  $dumpfile("d_using_t.vcd");    "dumpfile": Unknown word.
  $dumpvars(0, TB_D_using_T);    "dumpvars": Unknown word.
end
initial begin
  $monitor("Time=%0t | RST=%b | D=%b | Q=%b", $time, RST, D, Q);
end
initial begin
  RST=1'b1; D=1'b0; #8;
  RST=1'b0; #6;
  D=1'b0; #15;  // Q should follow D
  D=1'b1; #10;  // Q should become 1
  D=1'b0; #10;  // Q should become 0
  D=1'b1; #10;  // Q should become 1
  D=1'b1; #10;  // Q should stay 1
  D=1'b0; #10;  // Q should become 0
  RST=1'b1; #5;
  RST=1'b0; #15;
  $finish;
end
endmodule    "endmodule": Unknown word.

```

```
D:\iverilog lab\week 6>iverilog -o d_using_t .\d_using_t.v .\tflipflop.v .\d_using_t_tb.v

D:\iverilog lab\week 6>vvp d_using_t
VCD info: dumpfile d_using_t.vcd opened for output.
Time=0 | RST=1 | D=0 | Q=x
Time=5 | RST=1 | D=0 | Q=0
Time=8 | RST=0 | D=0 | Q=0
Time=29 | RST=0 | D=1 | Q=0
Time=35 | RST=0 | D=1 | Q=1
Time=39 | RST=0 | D=0 | Q=1
Time=45 | RST=0 | D=0 | Q=0
Time=49 | RST=0 | D=1 | Q=0
Time=55 | RST=0 | D=1 | Q=1
Time=69 | RST=0 | D=0 | Q=1
Time=75 | RST=0 | D=0 | Q=0
Time=79 | RST=1 | D=0 | Q=0
Time=84 | RST=0 | D=0 | Q=0

D:\iverilog lab\week 6>gtkwave d_using_t.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[99] end time.
```


GTKWave - d_using_t.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 99 sec
Marker: -- | Cursor: 5 sec
SST
TB_D_using_T
Type Signals
reg CLK
reg D
wire Q
reg RST
Filter:
Append Insert Replace
Signals
Time
CLK
D
Q
RST
Waves
10 sec 20 sec 30 sec 40 sec 50 sec 60 sec 70 sec 80 sec 90 sec