Digital Design and Computer Organisation Laboratory

3rd Semester, Academic Year 2025

Date:08-09-2025

Name: Aakash Desai	SRN:PES1UG24CS006	Section 3A

Week Number: 5 Program Number: 1

TITLE: Write a verilog program to model a D Flip-flop. Generate the VVP output and simulation waveform using GTKWave.

Deliverables

I. Verilog Code Screenshot
II. Verilog VVP Output Screen Shot
III. GTKWAVE Screenshot
IV. Output Table to be completed and included

I.Verilog Code Snippet

- Main code

```
module dff (
    input wire D,
    input wire clk,
    input wire rst,
    output reg Q
);
    always @(posedge clk or posedge rst) begin
        if (rst)
            Q <= 1'b0; // Asynchronous reset
        else
            Q <= D;    // Capture D on rising edge of clock
    end
endmodule
```

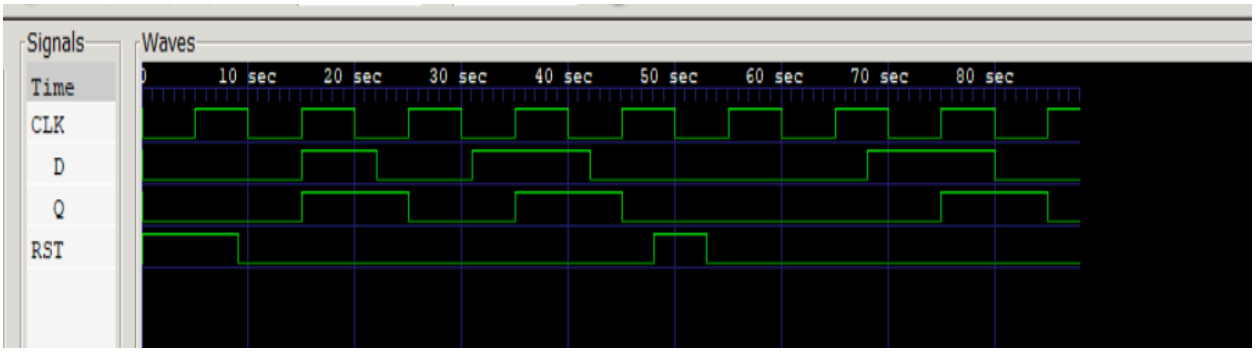
- Test Bench

```
module tb_dff;
    reg D;
    reg CLK;
    reg RST;
    wire Q;


    dff newD(.D(D), .clk(CLK), .rst(RST), .Q(Q));
    initial begin
        CLK = 1'b0;
        forever #5 CLK = ~CLK;  // 10 time units period
    end
    initial begin
        $dumpfile("dff.vcd");
        $dumpvars(0, tb_dff);
    end
    initial begin
        RST = 1'b1; D = 1'b0; #9;
        RST = 1'b0;              #6;
        D = 1'b1; #7;
        D = 1'b0; #9;
        D = 1'b1; #11;
        D = 1'b0; #6;
        RST = 1'b1; #5; RST = 1'b0; #15;
        D = 1'b1; #12;
        D = 1'b0; #8;
        $finish;
    end
    initial begin
        #200 $finish;
    end
    initial begin
        $display("Time\tRST\tCLK\tD\tQ");
        $monitor("%0t\t%b\t%b\t%b\t%b", $time, RST, CLK, D, Q);
    end

endmodule
```

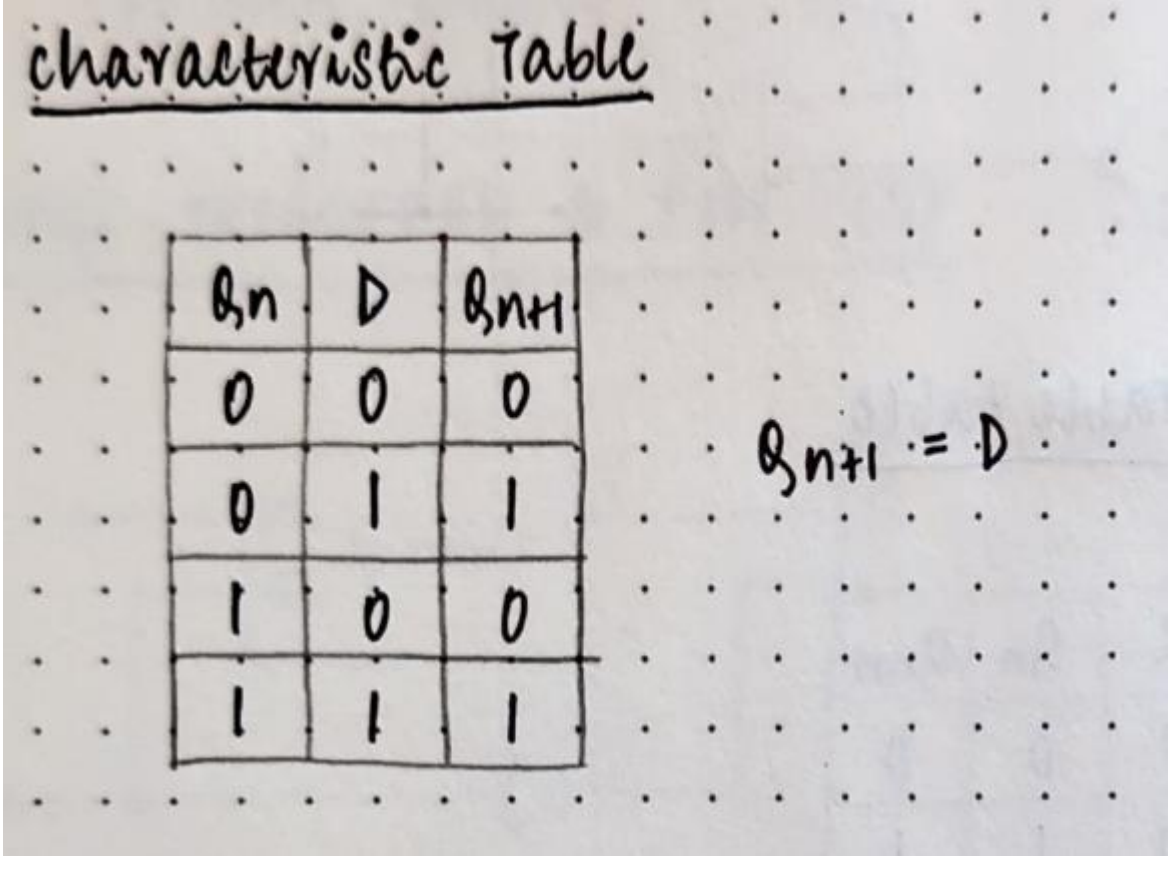
II Verilog VVP output ScreenShot

```
VCD info: dumpfile dff.vcd opened for output.
Time    RST     CLK     D       Q
0       1       0       0       0
5       1       1       0       0
9       0       1       0       0
10      0       0       0       0
15      0       1       1       1
20      0       0       1       1
22      0       0       0       1
25      0       1       0       0
30      0       0       0       0
31      0       0       1       0
35      0       1       1       1
40      0       0       1       1
42      0       0       0       1
45      0       1       0       0
48      1       1       0       0
50      1       0       0       0
53      0       0       0       0
55      0       1       0       0
60      0       0       0       0
65      0       1       0       0
68      0       1       1       0
70      0       0       1       0
75      0       1       1       1
80      0       0       0       1
85      0       1       0       0
```

III GTKWave ScreenShot



IV Characteristic Table



TITLE: Write a verilog program to model T Flip-flop.

Generate the VVP output and simulation waveform using GTKWave.

Deliverables

I. Verilog Code Screenshot
II. Verilog VVP Output Screen Shot
III. GTKWAVE Screenshot
IV. Output Table to be completed and included

I.Verilog Code Snippet

- Main Code

```
module t_ff (input wire T, input wire clk, input wire rst, output reg Q);
always @(posedge clk) begin
                Q = 1'b0;
        if (rst)
                Q <= 1'b0;
        else if (T)
                Q <= ~Q;
        else
                Q <= Q;
end
endmodule
```

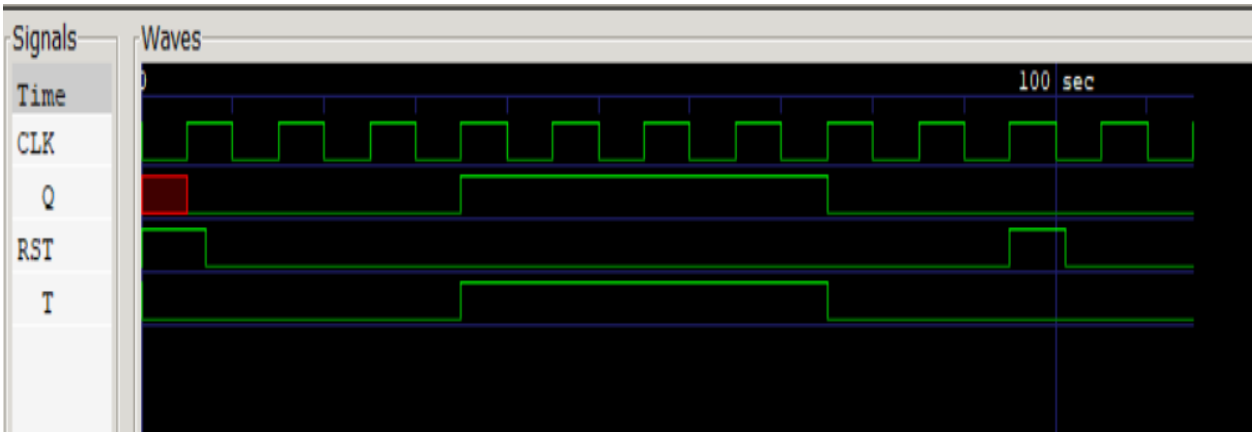
- Test Bench

```
module TB_T;
reg T;
reg CLK;
reg RST;
wire Q;
        t_ff newT (.T(T), .clk(CLK), .rst(RST), .Q(Q));
        initial begin
                CLK = 1'b0;
                forever #5 CLK = ~ CLK;
        end
        initial begin
                $dumpfile("t_ff.vcd");
                $dumpvars(0,TB_T);
        end
        initial begin
                $monitor ("Time=%0t | RST=%b | T=%b | Q=%b", $time, RST, T, Q);
        end
        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
endmodule
```

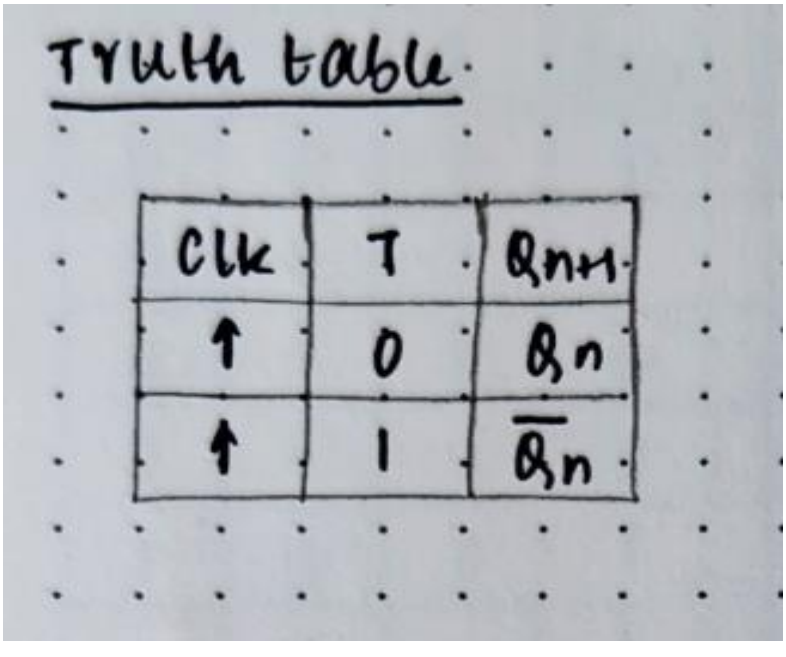
II Verilog VVP ScreenShot

```
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=1
Time=55 | RST=0 | T=1 | Q=1
Time=65 | RST=0 | T=1 | Q=1
Time=75 | RST=0 | T=0 | Q=0
Time=95 | RST=1 | T=0 | Q=0
Time=101 | RST=0 | T=0 | Q=0
```

III GTK wave Screenshot



IV Truth Table



TITLE: Write a verilog program to model J-K Flip Flop.Generate the VVP output and simulation waveform using GTKWave.

Deliverables

I. Verilog Code Screenshot
II. Verilog VVP Output Screen Shot
III. GTKWAVE Screenshot
IV. Output Table to be completed and included

- I Iverilog code

```
module jk_ff(
    input wire j,
    input wire k,
    input wire clk,
    input wire rst,
    output reg q
);
    always @(posedge clk or posedge rst) begin
        if (rst)
            q <= 1'b0;
        else begin
            case ({j,k})
                2'b00: q <= q;
                2'b01: q <= 1'b0;
                2'b10: q <= 1'b1;
                2'b11: q <= ~q;
            endcase
        end
    end
endmodule
```

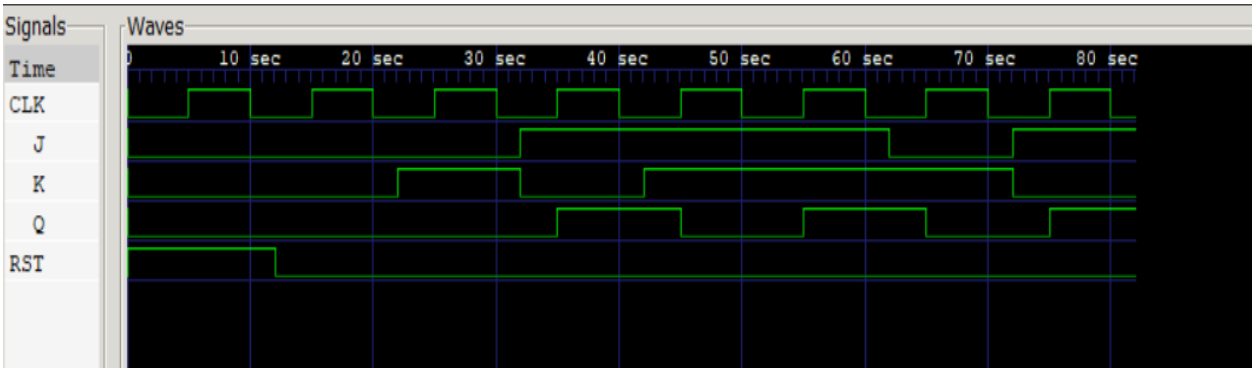
- Testbench

```
module tb_jk;
    reg J, K, CLK, RST;
    wire Q;
    jk_ff uut (
        .j(J),
        .k(K),
        .clk(CLK),
        .rst(RST),
        .q(Q)
    );
    initial begin
        CLK = 0;
        forever #5 CLK = ~CLK;
    end
    initial begin
        $dumpfile("jk.vcd");
        $dumpvars(0, tb_jk);
    end
    initial begin
        RST = 1; J = 0; K = 0; #12;
        RST = 0;
        J = 0; K = 0; #10;
        J = 0; K = 1; #10;
        J = 1; K = 0; #10;
        J = 1; K = 1; #10;
        J = 1; K = 1; #10;
        J = 0; K = 1; #10;
        J = 1; K = 0; #10;
        $finish;
    end
    initial begin
        $display("Time\tRST\tCLK\tJ\tK\tQ");
        $monitor("%0t\t%b\t%b\t%b\t%b\t%b", $time, RST, CLK, J, K, Q);
    end
endmodule
```

II VVP screenshot

```
C:\iverilog\bin\Week5>vvp dsn
VCD info: dumpfile jk.vcd opened for output.
Time    RST     CLK     J       K       Q
0       1       0       0       0       0
5       1       1       0       0       0
10      1       0       0       0       0
12      0       0       0       0       0
15      0       1       0       0       0
20      0       0       0       0       0
22      0       0       0       1       0
25      0       1       0       1       0
30      0       0       0       1       0
32      0       0       1       0       0
35      0       1       1       0       1
40      0       0       1       0       1
42      0       0       1       1       1
45      0       1       1       1       0
50      0       0       1       1       0
55      0       1       1       1       1
60      0       0       1       1       1
62      0       0       0       1       1
65      0       1       0       1       0
70      0       0       0       1       0
72      0       0       1       0       0
75      0       1       1       0       1
80      0       0       1       0       1
```

III GTKwave screenshot



IV Characteristic Table

characteristics table

Q_n	J	K	Q_{n+1}
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$$Q_{n+1} = J\overline{Q_n} + \overline{K}Q_n$$

TITLE: Write a verilog program to model SR Flip Flop. Generate the VVP output and simulation waveform using GTKWave.

Deliverables

I. Verilog Code Screenshot
II. Verilog VVP Output Screen Shot
III. GTKWAVE Screenshot
IV. Output Table to be completed and included

I.Verilog code

- Main code

```
module sr_ff(
    input wire s,
    input wire r,
    input wire clk,
    input wire rst,
    output reg q
);
    always @(posedge clk or posedge rst) begin
        if (rst)
            q <= 1'b0;
        else begin
            case ({s,r})
                2'b00: q <= q;
                2'b01: q <= 1'b0;
                2'b10: q <= 1'b1;
                2'b11: q <= 1'bx;
            endcase
        end
    end
endmodule
```

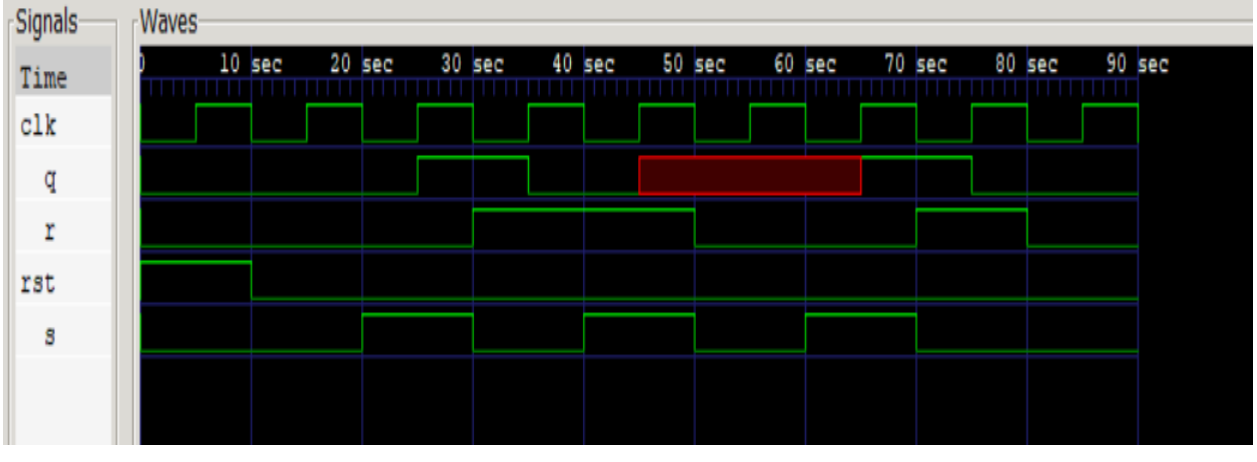
- Test bench

```
module tb_sr_ff;
    reg s, r;
    reg clk, rst;
    wire q;
    sr_ff uut(
        .s(s),
        .r(r),
        .clk(clk),
        .rst(rst),
        .q(q)
    );
    initial clk = 0;
    always #5 clk = ~clk;
    initial begin
        $dumpfile("sr_ff.vcd");
        $dumpvars(0, tb_sr_ff);
        s = 0; r = 0; rst = 1;
        #10 rst = 0;
        s = 0; r = 0; #10;
        s = 1; r = 0; #10;
        s = 0; r = 1; #10;
        s = 1; r = 1; #10;
        s = 0; r = 0; #10;
        s = 1; r = 0; #10;
        s = 0; r = 1; #10;
        s = 0; r = 0; #10;
        $finish;
    end
    initial begin
        $monitor("Time=%0t | S=%b R=%b | Q=%b", $time, s, r, q);
    end
endmodule
```

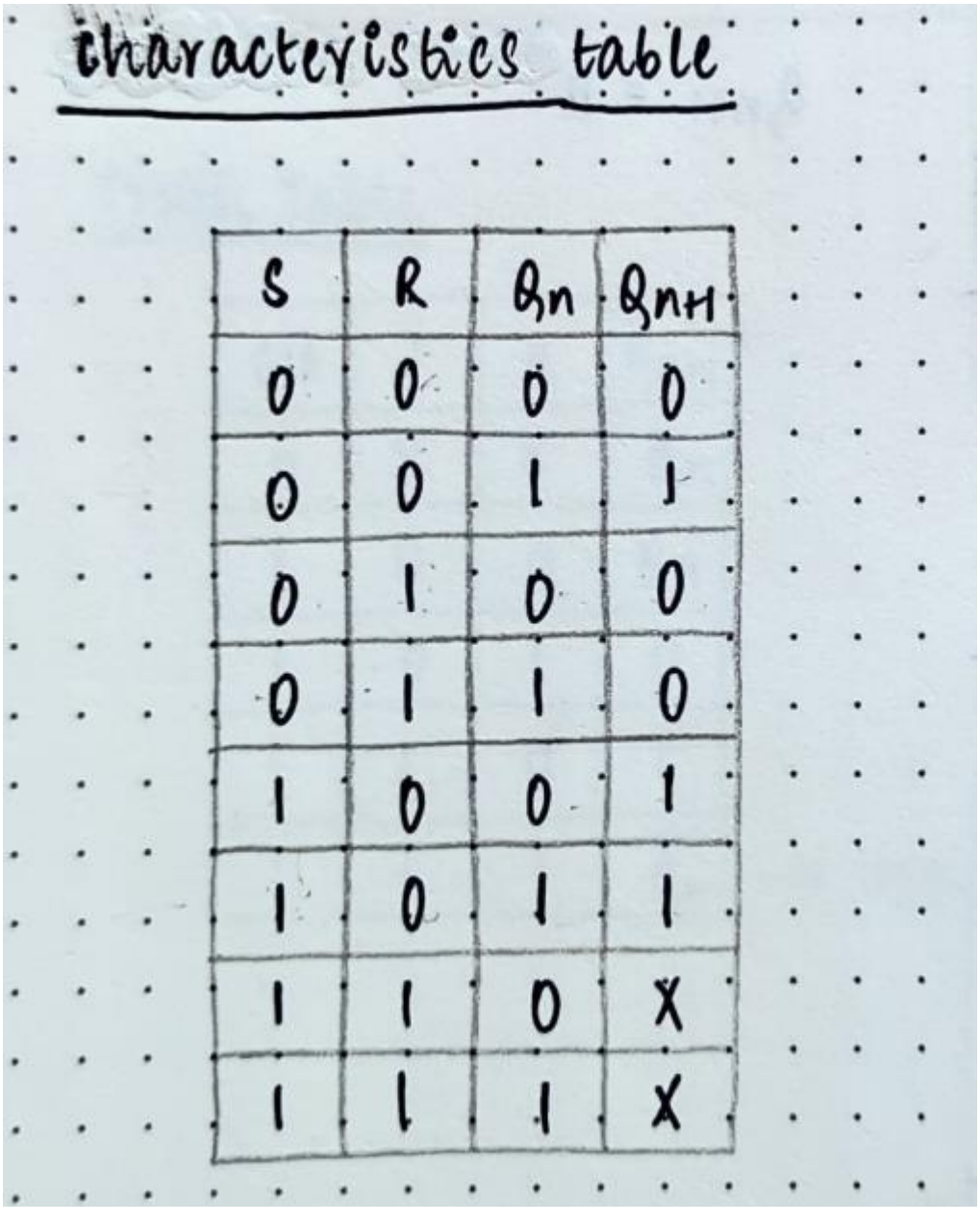
II VVP Screenshot

```
C:\iverilog\bin\Week5>vvp dsn
VCD info: dumpfile sr_ff.vcd opened for output.
Time=0 | S=0 R=0 | Q=0
Time=20 | S=1 R=0 | Q=0
Time=25 | S=1 R=0 | Q=1
Time=30 | S=0 R=1 | Q=1
Time=35 | S=0 R=1 | Q=0
Time=40 | S=1 R=1 | Q=0
Time=45 | S=1 R=1 | Q=x
Time=50 | S=0 R=0 | Q=x
Time=60 | S=1 R=0 | Q=x
Time=65 | S=1 R=0 | Q=1
Time=70 | S=0 R=1 | Q=1
Time=75 | S=0 R=1 | Q=0
Time=80 | S=0 R=0 | Q=0
```

III GTKWave output



IV Truth Table



characteristics table

S	R	Q_n	Q_{n+1}
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

TITLE: Write a verilog program to design and implement a D flip flop using a T flip flop. Generate the VVP output and simulation waveform using GTKWave.

Deliverables

I. Verilog Code Screenshot
II. Verilog VVP Output Screen Shot
III. GTKWAVE Screenshot
IV. Output Table to be completed and included

I Verilog Code

- Main code

```
module t_ff(
    input wire t,
    input wire clk,
    input wire rst,
    output reg q
);
    always @(posedge clk or posedge rst) begin
        if (rst)
            q <= 1'b0;
        else if (t)
            q <= ~q;
        else
            q <= q;
    end
endmodule
module d_ff_using_t(
    input wire d,
    input wire clk,
    input wire rst,
    output wire q
);
    wire t;
    assign t = d ^ q;
    t_ff uut (
        .t(t),
        .clk(clk),
        .rst(rst),
        .q(q)
    );
endmodule
```

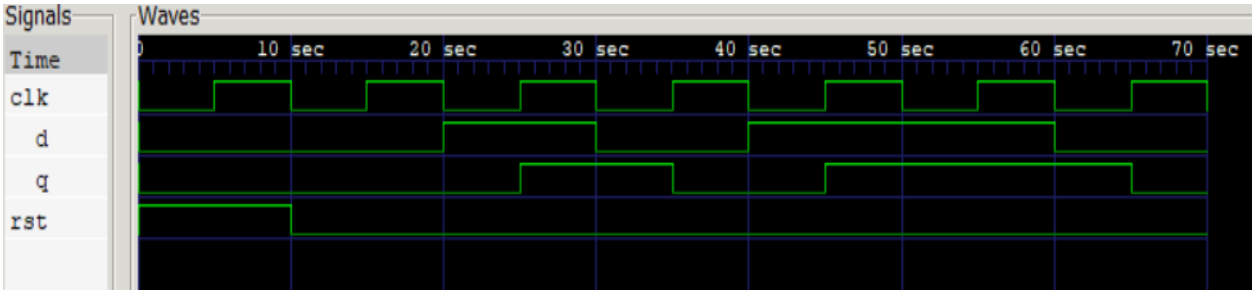
- Test bench

```
module tb_d_ff_using_t;
    reg d;
    reg clk, rst;
    wire q;
    d_ff_using_t uut(
        .d(d),
        .clk(clk),
        .rst(rst),
        .q(q)
    );
    initial clk = 0;
    always #5 clk = ~clk;
    initial begin
        $dumpfile("d_ff_using_t.vcd");
        $dumpvars(0, tb_d_ff_using_t);
        d = 0; rst = 1;
        #10 rst = 0;
        d = 0; #10;
        d = 1; #10;
        d = 0; #10;
        d = 1; #10;
        d = 1; #10;
        d = 0; #10;

        $finish;
    end
    initial begin
        $monitor("Time=%0t | D=%b | Q=%b", $time, d, q);
    end
endmodule
```

II VVP output Screenshot

```
C:\iverilog\bin\Week5>vvp dsn
VCD info: dumpfile d_ff_using_t.vcd opened for output.
Time=0 | D=0 | Q=0
Time=20 | D=1 | Q=0
Time=25 | D=1 | Q=1
Time=30 | D=0 | Q=1
Time=35 | D=0 | Q=0
Time=40 | D=1 | Q=0
Time=45 | D=1 | Q=1
Time=60 | D=0 | Q=1
Time=65 | D=0 | Q=0
```

III GTKwave output screenshot



IV Truth Table

Current State	D Input	**Required T Input**	T Flip-Flop Action	Next State
Q_n	D	$T = D \oplus Q_n$		Q_{n+1}
0	0	$0 \oplus 0 = 0$	Hold	0
0	1	$1 \oplus 0 = 1$	Toggle	1
1	0	$0 \oplus 1 = 1$	Toggle	0
1	1	$1 \oplus 1 = 0$	Hold	1

Export to Sheets