CALIFORNIA STATE UNIVERSITY, NORTHRIDGE

Department of Electrical and Computer Engineering

ECE 526L

LAB – 4

EDGE TRIGGER D-FILPFLOP

Professor : Orod Haghighara

Written By : Gunupudi. Tarun Kumar

INTRODUCTION :

In this lab we will model an edge triggered flip-flop using a hierarchal modelling approach using Synopsys VCS in Linux OS environment and different terminal commands

METHODOLOGY :

The first thing is being familiar with the Linux and Synopsys VCS that will navigate your system using terminal

Now to write a Verilog code for the module using a text editor. always use linux based text editors and the file should have ".v" extension.

In this lab we will model an edge triggered flip-flop using a hierarchical modelling approach. Using primitive gates in the question we will write a code for SR_latch. And now save as "SR_Latch2".

The module name and file name should be consistent

SR_Latch2 - Notepad

File Edit Format View Help

```
`timescale 1 ns / 100 ps

`define PRIMARY_OUT  2
`define FAN_OUT_1    0.5
`define FAN_OUT_2    0.8
`define FAN_OUT_3    1
`define TIME_DELAY_1 3
`define TIME_DELAY_2 4
`define TIME_DELAY_3 5

module SR_Latch2(Q, Qnot, s0, s1, r0, r1);
  input s0, s1, r0, r1;
  output Q, Qnot;

  nand #(`TIME_DELAY_3 + `FAN_OUT_2) NAND1(Q, s0, s1, Qnot);
  nand #(`TIME_DELAY_3 + `FAN_OUT_2) NAND2(Qnot, r0, r1, Q);

endmodule
```

This is the fig of "SR_Latch2 ", store it in new file, also in folder "lab4"

Now for D-flip flop circuit,write the code using above module SR Latch for design of D- flip flop and save it as “dff.v”

dff - Notepad

File Edit Format View Help

```
`timescale 1ns / 100ps

`define PRIMARY_OUT  2 // ns (primary outputs)
`define FAN_OUT_1    0.5 // ns (one output fanout)
`define FAN_OUT_2    0.8 // ns (two output fanout)
`define FAN_OUT_3    1 // ns (three output fanout)
`define TIME_DELAY_1 3 // ns (one input gates)
`define TIME_DELAY_2 4 // ns (two input gates)
`define TIME_DELAY_3 5 // ns (three input gates)


module dff(q,qbar,clock,data,clear);
        input clock, data, clear;
        output q, qbar;

        wire s, sbar, r, rbar, clkbar, cbar, dbar, clock, clear, data, clr, clk, d;

        not #(`TIME_DELAY_1 + `FAN_OUT_1) N1(cbar, clear);
        not #(`TIME_DELAY_1 + `FAN_OUT_1) N2(clkbar, clock);
        not #(`TIME_DELAY_1 + `FAN_OUT_1) N3(dbar, data);
        not #(`TIME_DELAY_1 + `FAN_OUT_1) N4(clr, cbar);
        not #(`TIME_DELAY_1 + `FAN_OUT_1) N5(clk, clkbar);
        not #(`TIME_DELAY_1 + `FAN_OUT_1) N6(d, dbar);

        SR_Latch2 s1(s, sbar, clr, clk, 1'b1, rbar);
        SR_Latch2 s2(r, rbar, s, clk, clr, d);
        SR_Latch2 s3(q, qbar, s, 1'b1, clr, r);
endmodule
```

The above figure is the “ dff.v ” deisgn code.

Then we should write a test bench module to verify the functionality of your flip-flop, in this module we use each of the output system tasks , $monitor, $display, $write and $strobe and name it as “dff_tb.v”.

This below figure is “dff_tb.v” test bench code.

dff_tb - Notepad

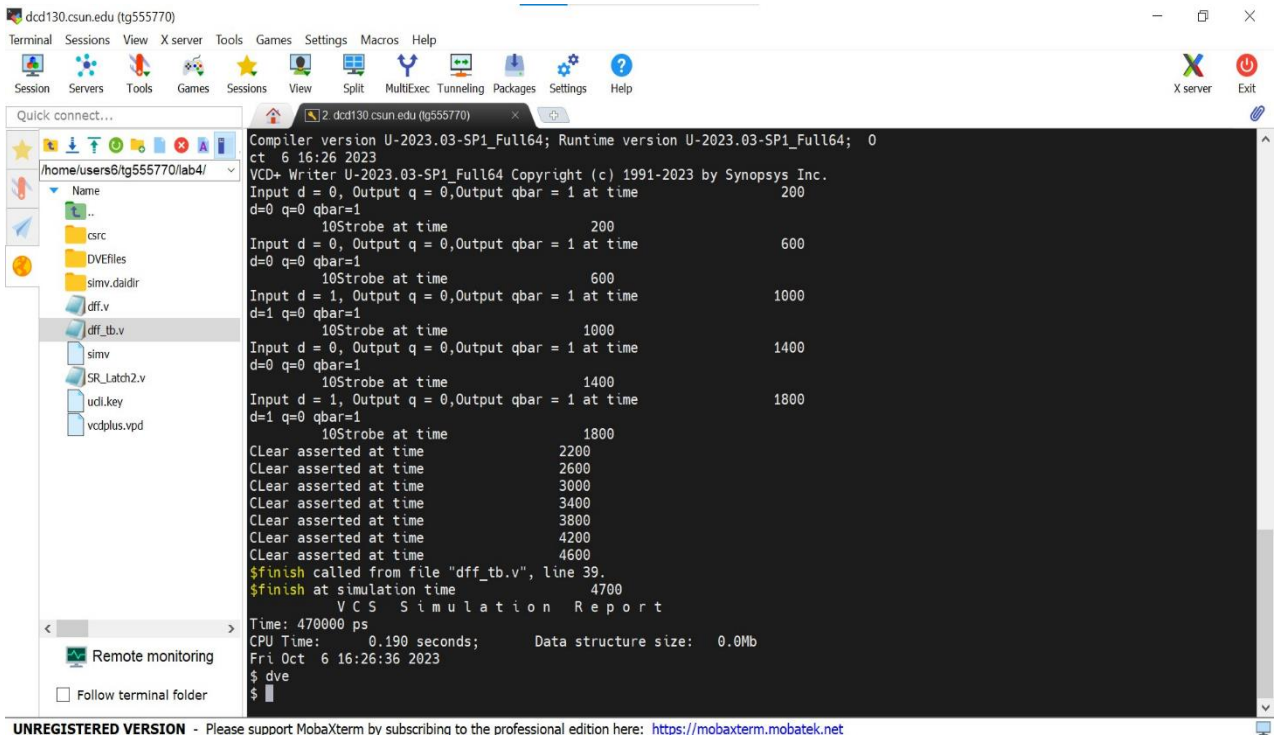
File Edit Format View Help

```
`timescale 1ns / 1ns

module dff_tb();

  reg clock;
  reg clear;
  reg data;
  wire q, qbar;

  dff UUT(q,qbar,clock,data,clear);

  initial begin
    $vcdpluson;
    clock=0;
    clear = 0;
    data = 0;

    forever #20 clock = ~clock;
 end

// Testbench behavior

   // Simulate test scenario
   initial begin
     #20 clear = 0;

     // Apply data to D input
     #50 data = 1;
     #50 data = 0;
     #50 data = 1;

     #50 clear = 1;

     #50 data = 1;
     #50 data = 0;
     #50 data = 1;

     // Finish simulation after some time
     #100 $finish;
   end

  always @(posedge clock or posedge clear) begin
     if (clear) begin

     // Use $monitor to display clear information
     $monitor("CLear asserted at time %t", $time);
   end else begin

     // Use $display to display D flip-flop input and output values
     $display("Input d = %b, Output q = %b,Output qbar = %b at time %t", data, q, qbar, $time);

     // Use $write to write input and output values to a file
     $write("d=%b q=%b qbar=%b\n", data, q, qbar);

     // Use $strobe to create a strobe signal for debugging
     $strobe(10, "Strobe at time %t", $time);
     end

end

endmodule
```

Now after saving the SR Latch and D-flipflop.Then use the command as follows

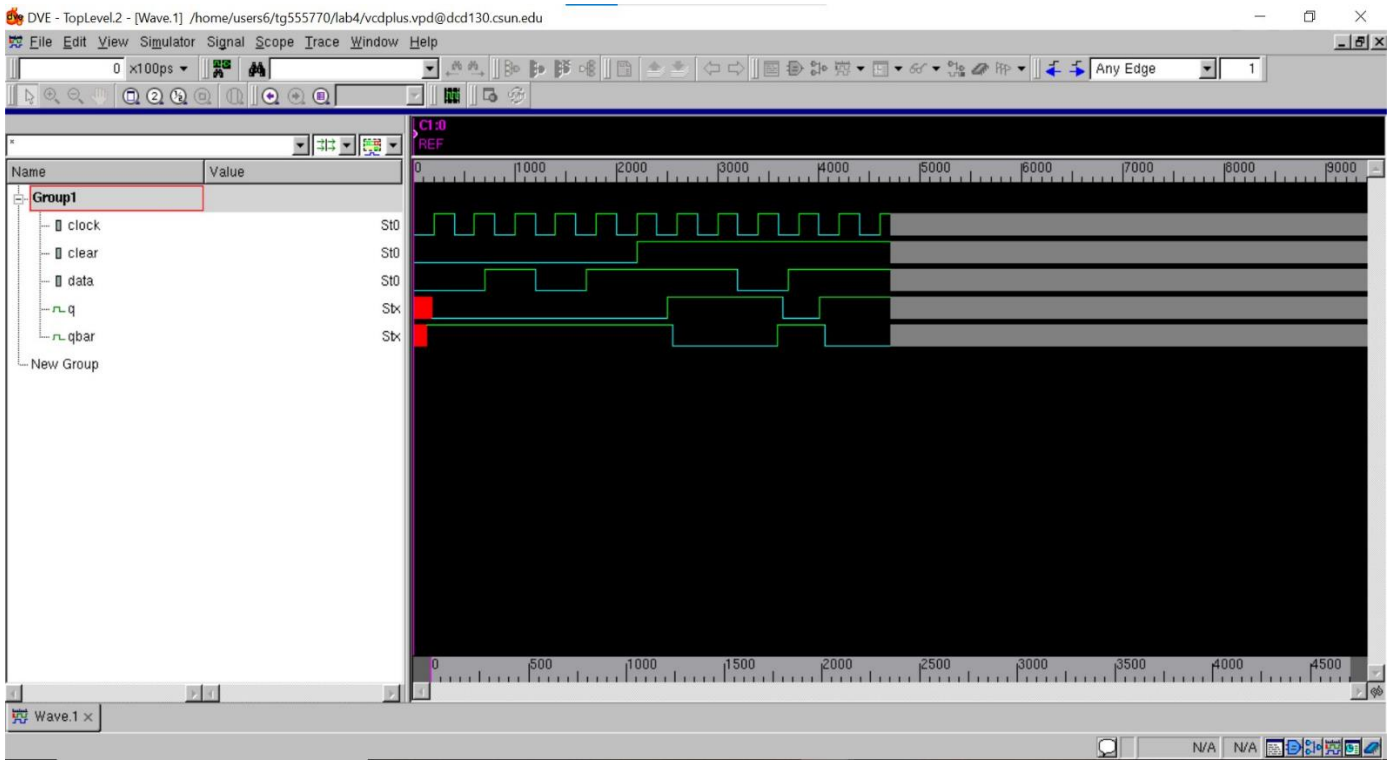
“ vcs -debug_access+all SRLatch.v dff.v dff_tb.v ”

If we don’t see any error messages if everything is typed correctly then output should look like below figure.



To see the wave form first type “dve” command line then we will get the blank simulation window then go to file and open database then choose vcdplus.vpd and open the file

Then finally select all the signals with the mouse,then right click and select to “ add to wave ->new wave view.



Analysis of result :

The D-Flip Flop's inputs and outputs are identical. That means the result will be a 0 if the input is 0->0. Additionally, if the input is 1->1, then the output will also be 1. Here, "d" stands for the data input provided by the table mentioned earlier details the D-input as 0,0,1,1 and the 'qbar' output as being identical to the data supplied as 0,0,1,1. Furthermore, the data's clock will alternate with the letter "q," which will be 0, 1, 0, 1. The D-Flip Flop completely expresses what I said in the above wave form.To provide alternative values for the clock, clear, and data, we also use a "NOT" gate here as inverter. which are substitute values for them. Here, the q and qbar outputs are obtained using the three SR-latches. simply adding the values specified in the code given by the values from the provided circuit. Finally positive edge triggered d flip flop becomes active when its clock signal goes from low to high and ignores the high-low transition.

Conclusion :

In this lab we done model an edge triggered flip-flop using a hierarchal modelling approach using Synopsys VCS in Linux OS environment and different terminal commands

I hereby attest that this lab report is entirely my own work.I have not copied either code or text from anyone ,nor have I allowed or I will anyone to copy my work.

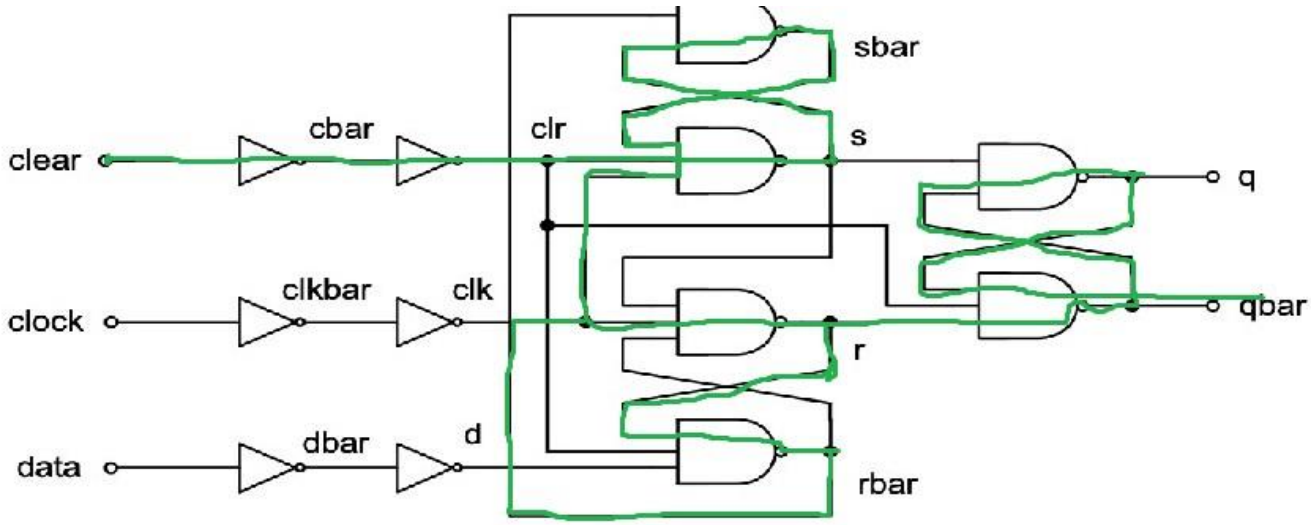
Name(printed) Gunupudi Tarun Kumar

Name(signed) G.Tarun Date 10/6/2023

LAB REPORT QUESTION :

1.Calculate the longest path delay, show how you calculated it (which gates and what delays) and show the value?

ANS.



Positve Edge-Triggered D Flip-Flop with Clear
Figure 2

The longest path is calculated as

=NOT+NOT+NAND+NAND+NAND+NAND+NAND+NAND

=(3+0.5)+(3+0.5)+(5+0.8)+(5+0.8)+(5+0.8)+(5+0.8)+(5+1.0)+(5+2.0)

=43.2ns

2.what is the maximum operating frequency for your circuit?

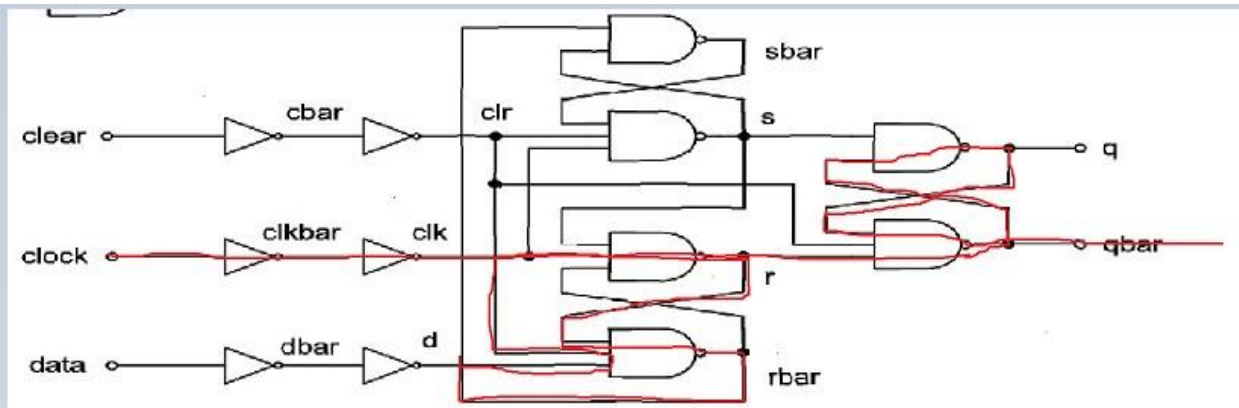
Ans. It is calculated by 1/ shortest delay

Clock to qbar : NOT+NOT+NAND+NAND+NAND+NAND

=(3+0.5)+(3+0.5)+(5+0.8)+(5+0.8)+(5+1.0)+(5+2.0)

=31.6ns

=1/31.6 = 3.174603175*10^7



Positve Edge-Triggered D Flip-Flop with Clear
Figure 2

This above picture is the shortest delay path.

Hierarchical DFF: