Fall 2019

California State University, Northridge

Department of Electrical and Computer Engineering

ECE 526L

Experiment # 3
Hierarchical Modeling

Sept. 26, 2019

Authors: Ridge Tejuco
Professor: Ronald Mehler Ph.D

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 will I allow anyone to copy my work.

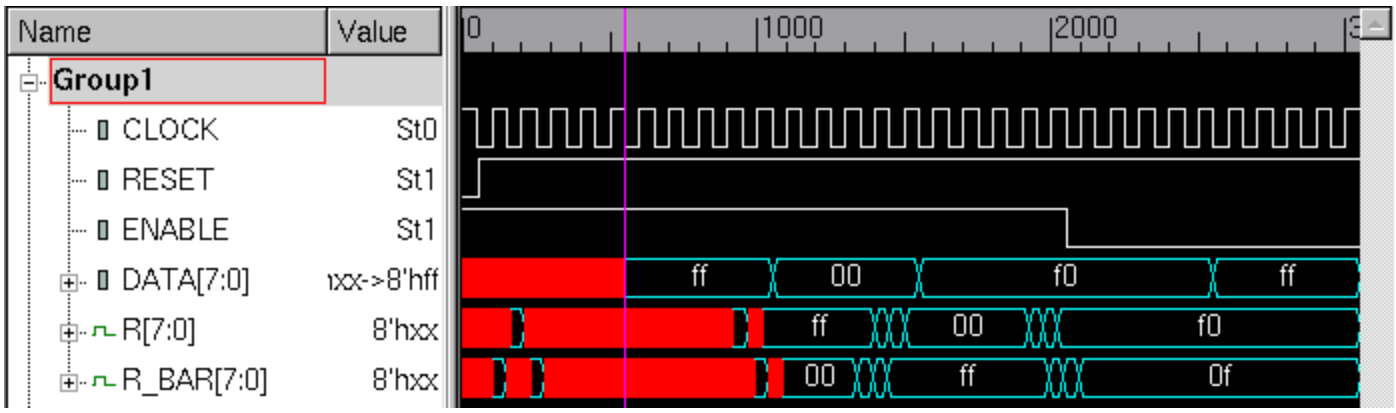
Name (printed) _____

Name (signed) _____

Date _____

Analysis

Figure 1. TB_REGISTER.v waveform



Looking closely at figure 1, the DATA input is closely followed by the R output. The input tested was FF, 00, then F0.

After F0, the ENABLE input was dropped to 0 and the DATA input was given FF; However, the R output did not follow the DATA input as expected.

Looking closer at figure 2, the RESET is set initially 0. The R output eventually gets a correct value of 00. These results show that the design is logically correct.

Figure 2. Zoom on RESET

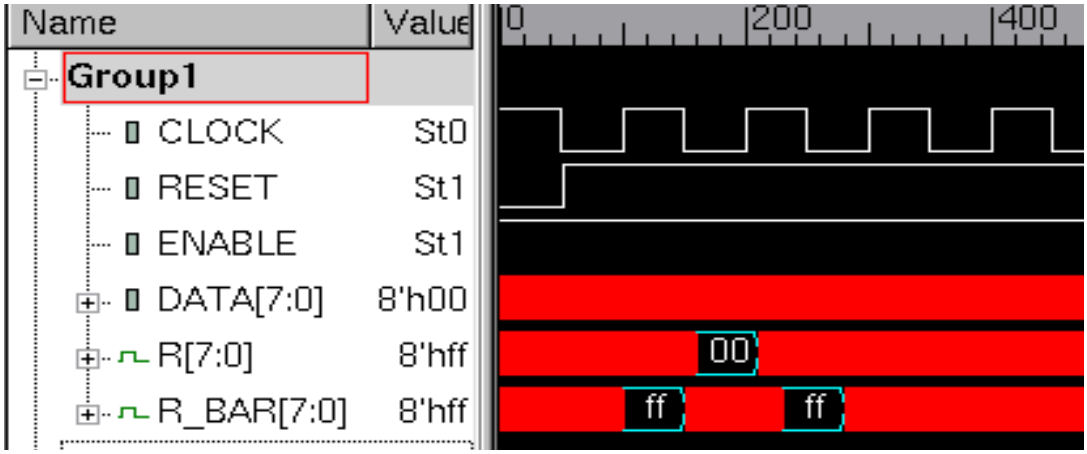
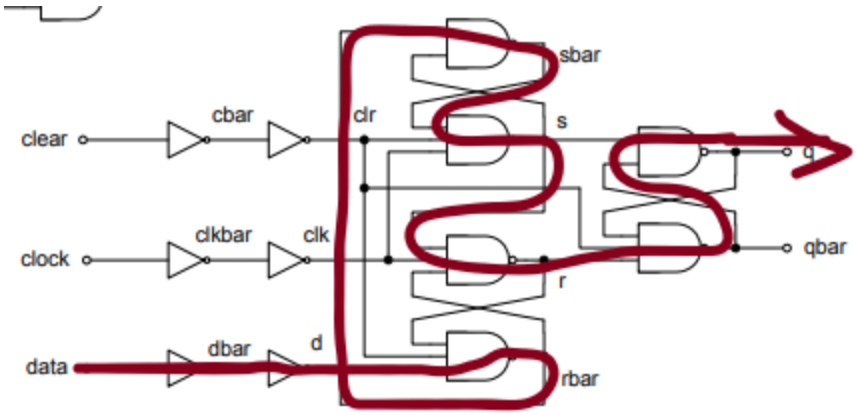


Figure 3. Longest path in DFF

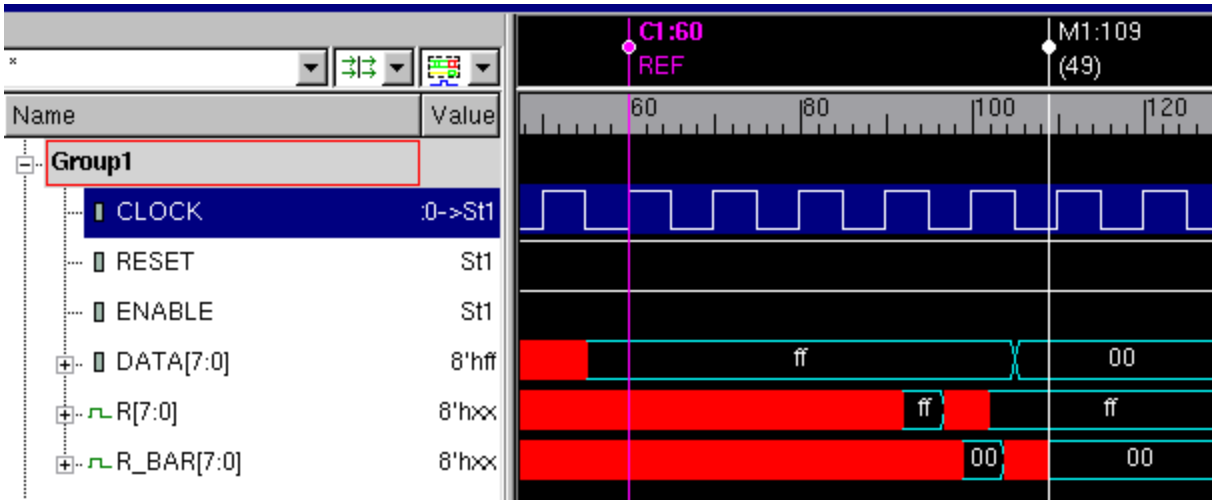


Positve Edge-Triggered D Flip-Flop with Clear

To calculate the operating frequency of the circuit, the longest path of the circuit was found. The longest path includes that data input of the multiplexer to the multiplexers data output to the DFF, and all the way to the DFF's output. By summing the delays of each gate and fanout, the period is calculated to be 35.7 ns.

$$F = \frac{1}{T} = \frac{1}{35.7\,ns} = 28.01\,MHz$$

Figure 4. Time delay at first input

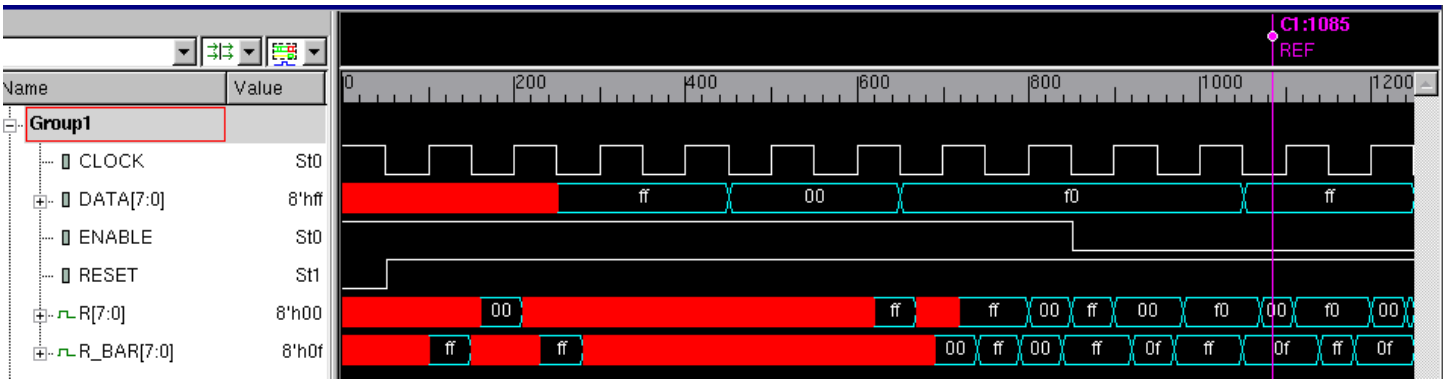


Looking closer at figure 4, the maximum time delay is seen. The DATA=FF input is followed by R = FF and R_BAR = 00 at 109 ns. The max time delay simulated by the circuit is 49 ns. The max operating frequency of the circuit is calculated as follows.

$$F = \frac{1}{T} = \frac{1}{49\,ns} = 20.408\,MHz$$

The simulated operating frequency is lower than our calculated because the simulation accounts for the feedback and stabalization of the output.

Figure 5. Input delay of 20 ns



Looking closer at figure 5, the max operating frequency was exceeded by setting the delay between inputs by only 20 ns. Looking at the results of the waveform, the FF input is able to propagate; However, after inputting the DATA = 00 and DATA = F0, the output never stabilizes and begins oscillating between F0 and 00.

```
/*=================================================
CSUN - Experiment 3
Ridge Tejuco
SEPT 19, 2019
=================================================
TB_REGISTER.v
GOALS
- stimulate parallel registers (DATA[7:0] ==> R[7:0])
- stimulate a clock ( #5 CLK = ~ CLK)
- stimulate a reset (RST = 0)
- stimulate enable vs disable
- account for time delay of output and calculate frequency
=================================================*/
`timescale 1 ns / 100 ps;
`define ALL_ONE DATA = 8'b11111111;
`define ALL_ZERO DATA = 8'b00000000;
`define LSB_HALF_ZERO DATA = 8'b11110000;

module TB_REGISTER ();
  parameter SIZE = 8,
            DELAY = 50;
  reg CLOCK, RESET, ENABLE;
  reg [SIZE - 1: 0] DATA;
  wire [SIZE - 1: 0] R,R_BAR;
  //REGISTER(R,R_BAR,DATA,ENA,CLK,RST)
  REGISTER UUT (R,R_BAR,DATA,ENABLE,CLOCK,RESET);
  //CLOCK INPUT
  always begin
     #5 CLOCK = ~CLOCK;
  end
  initial begin
     $vcdpluson;
     RESET = 0; ENABLE = 1;CLOCK = 1;
     #5.1 RESET = 1;
     #DELAY `ALL_ONE
     #DELAY `ALL_ZERO
     #DELAY `LSB_HALF_ZERO
     #DELAY ENABLE = 0;
     #DELAY `ALL_ONE
     #DELAY $finish;
  end
endmodule
```

```
/*================================================
CSUN - Experiment 3
Ridge Tejuco
SEPT 19, 2019
================================================
SR_LATCH.sv
Q = ~(S0 & R0 & Q_NOT);
Q_NOT = ~(R1 & R0 & Q);
================================================*/
`timescale 1ns / 100ps
`ifndef SINGLE
  `include "DELAYS.v"
`endif

module SR_LATCH(Q, Q_NOT, S0, S1, R0,R1);
  parameter NAND1_DELAY = 3.0;
    output wire Q,Q_NOT;
    input S0,S1,R0,R1;
  nand #3.0 (Q_NOT,Q,R1,R0);
  nand #NAND1_DELAY (Q,S0,S1,Q_NOT);
endmodule
```

```
/*================================================
CSUN - Experiment 3
Ridge Tejuco
SEPT 19, 2019
================================================
DFF.v
3 instances of SR_LATCH
connections
CLEAR ==>> R0 of L0    CLOCK ==>> R1 of L0
CLEAR ==>> R0 of L1    CLOCK ==>> S1 of L1
CLEAR ==>> R0 of L2    DATA  ==>> R1 of L1
L0   ==>> S0 of L2    L1   ==>> R1 of L2
L0   ==>> S0 of L1    L1_NOT => S0 of L0
================================================*/
`timescale 1 ns / 100 ps
`ifndef SINGLE
  `include "DELAYS.v"
`endif
module DFF(Q,QBAR,CLOCK,DATA,CLEAR);
    output Q,QBAR;
    input CLOCK,DATA,CLEAR;

    wire `PRIMARY Q, QBAR;
    wire `FANOUT1 CBAR, CLR;
    wire `FANOUT1 CLKBAR, CLK;
    wire `FANOUT1 DBAR, D;
    wire `FANOUT3 S, SBAR, R, RBAR;

  //Buffers
    not `SINGLE (CBAR,CLEAR);
    not `SINGLE (CLR,CBAR);
    not `SINGLE (CLKBAR,CLOCK);
    not `SINGLE (CLK,CLKBAR);
    not `SINGLE (DBAR,DATA);
    not `SINGLE (D, DBAR);

  //SR_LATCH(Q, Q_NOT, S2, S1, R2,R1)
    SR_LATCH #(2.0) L0(S,SBAR,CLR,CLK,RBAR,1'b1);    // Overwrite first
parameter
    SR_LATCH L1(R,RBAR,S,CLK,CLR,D);
    SR_LATCH #(2.0) L2(Q,QBAR,S,1'b1,CLR,R);         // 1'b1 logic high
endmodule
```

```
//=====================================================
// Ridge Tejuco - CSUN
//=====================================================
// LAB 1
// MUX2_1.v
// September 5, 2019
//=====================================================
`timescale 1 ns / 100 ps
`ifndef SINGLE
  `include "DELAYS.v"
`endif
module MUX2_1(OUT, A, B, SEL);
    //Port declarations
  output OUT;
    input A, B, SEL;

    //Internal variable declarations
    wire `FANOUT1 A1,B1,SEL_N;
  wire `SINGLE OUT;

    //The netlist=================================
    not `SINGLE (SEL_N, SEL);

    and `DOUBLE (A1, A, SEL);     //A1 = A and SEL and ENA;

    and `DOUBLE (B1,B,SEL_N);         //B1 = B and SEL_N and ENA;

    or `DOUBLE (OUT,A1,B1);           //OUT = A1 or B1;
endmodule
//=====================================================
```

```
/*=================================================
CSUN - Experiment 3
Ridge Tejuco
SEPT 19, 2019
=================================================
REGISTER.v
INPUTS: DATA[0],ENA,CLK,RST
RST => CLEAR of DFF
CLK => CLOCK of DFF
ENA => SEL of MUX
DATA[i - 1] => B of MUX
R[i] => A of MUX
instances of MUX2_1.v and DFF.v
=================================================*/
`timescale 1 ns / 100 ps;
module REGISTER(R,R_BAR,DATA,ENA,CLK,RST);
    parameter SIZE = 8;
  genvar index;
    input ENA,CLK,RST;
  input wire [SIZE - 1:0] DATA;
    output wire [SIZE - 1:0] R,R_BAR;
  wire [SIZE - 1:0] MUX2_1OUT;

  generate
  for(index = 0; index < SIZE; index=index+1) begin : MUX_LIST
     MUX2_1 MUX0(MUX2_1OUT[index],DATA[index],R[index],ENA);
  end
  for(index = 0; index < SIZE; index=index+1) begin  : DFF_LIST
     DFF DFF0(R[index],R_BAR[index],CLK,MUX2_1OUT[index],RST);
  end
  endgenerate
endmodule
```