Lab 9 – Data path and Control for the 4-digit, 7-Segment Displays

Name: Von Kaukeano *Date*: 11/6/18

Course: Digital Circuit Design Lab – ECE 2613 *Section #*: 001

Summary/Abstract

In this lab, we created a finite state machine that's uses a time dependent clock signal to cycle through the four 7 segment displays using sequential logic. We used the 7 segment decoder, anode decoder, and frequencies that create cycles so fast that the human eye appears that all four are active at one time. We then used the switches, four bits at a time, to display the numbers in hexadecimal from their binary inputs on all four displays rather than one at a time.

Introduction

The background material for this lab is understanding the 7 segment decoder lab, how the finite state machine, and data paths all work. We will use the 7 segment decoder design and drive a continuous clock signal to be able to see all four the displays at once. With the knowledge of sequential logic, we began this lab by dividing the clock signal to reach our design specifications and instantiating it with the previous lab for the 7 segment decoder.

Procedure

The procedure to this lab consisted of creating three different modules. The first module display_driver_dp consisted of instantiating two modules from past labs and a creation of a 4x1 multiplexer. The second is the creation of the display_drive_fsm, which we designed the finite state machine based on the diagram in the lab instructions. The last module created was the display_drive_module which was the instantiation the first two modules created and the divider module created in a previous lab. After creation, we simulated to get the correct timing diagram and inserted the bit file into our board.

Results

Design Code

```
//
// lab9 : version 10/29/2018
//
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////

module display_driver_dp(output logic [6:0] cathode, output logic [3:0] anode, input logic [3:0] digit3,
    input logic [3:0] digit2, input logic [3:0] digit1, input logic [3:0] digit0,
    input logic [1:0] digit_sel, input logic display_on, input logic [1:0] anode_sel);

    // insert your code here

logic [3:0] bcd_in;

always_comb begin
case (digit_sel)
2'b00: bcd_in = digit0;
2'b01: bcd_in = digit1;
2'b10: bcd_in = digit2;
2'b11: bcd_in = digit3;
endcase
end

svn_seg_decoder u1(.seg_out(cathode), .bcd_in(bcd_in), .display_on(display_on));
anode_decoder u2(.anode(anode), .switch_in(anode_sel));


endmodule
```

30:1 Verilog Tabs: 4

```
// lab9 : version 10/29/2018
//
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////

module display_driver_fsm(output logic [1:0] anode_sel, output logic [1:0] digit_sel,
    input logic clk, input logic rst, input logic m_sec);

    // define and enumerate state variables
    enum logic [1:0] {D0, D1, D2, D3} state, next_state;

    // sequential logic
    always_ff @(posedge clk) begin
        state <= next_state;
    end

    // combinational for state machine
    always_comb begin
// complete the combinational logic for the state machine
        // defaults
    next_state = state;
        // main logic
case(state)
D0: begin
anode_sel = 2'b00;
digit_sel = 2'b00;
if (m_sec == 1)
next_state = D1;
end
```

12:33 Verilog Tabs: 4

```
D1: begin
anode_sel = 2'b01;
digit_sel = 2'b01;
if (m_sec == 1) begin
next_state = D2;
end
end

D2: begin
anode_sel = 2'b10;
digit_sel = 2'b10;
if (m_sec == 1)
next_state = D3;

end

D3: begin
anode_sel = 2'b11;
digit_sel = 2'b11;
if (m_sec == 1)
next_state = D0;
end
endcase

      // priority logic
if (rst == 1'b1) begin
next_state = D0;
anode_sel = 2'b00;
digit_sel = 2'b00;
end
```

12:33 Verilog Tabs: 4

```
//
// lab9 : version 10/29/2018
//
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
//////////////////////////////////////////////////////////////////////////////////

module display_driver( output logic [6:0] cathode, output logic [3:0] anode,
    output logic [1:0] anode_sel, output logic m_sec,
    input logic [3:0] digit3, input logic [3:0] digit2, input logic [3:0] digit1,
    input logic [3:0] digit0, input logic display_on, input logic rst, input logic clk);

    // insert your code here


logic [1:0] digit_sel;


display_driver_fsm u1(.anode_sel(anode_sel), .digit_sel(digit_sel), .clk(clk), .rst(rst), .m_sec(m_sec));

display_driver_dp u2(.cathode(cathode), .anode(anode), .digit0(digit0), .digit1(digit1), .digit2(digit2), .digit3(digit3),
.digit_sel(digit_sel), .display_on(display_on), .anode_sel(anode_sel));

divider #(.BIT_SIZE(17)) u3(.tc(m_sec), .clk(clk), .ena(1'b1),.rst(rst) ,.init_count(17'd99999));

endmodule
```

Simulation Results

```
source xsim.dir/work.tb_display_driver/xsim_script.tcl
# xsim {work.tb_display_driver} -autoloadwcfg -tclbatch {tb_display_driver.tcl} -onerror quit
Vivado Simulator 2018.2
Time resolution is 1 ps
source tb_display_driver.tcl
## run 10100us
Time: 1000000 ns
Time: 2000000 ns
Time: 3000000 ns
Time: 4000000 ns
Time: 5000000 ns
Time: 6000000 ns
Time: 7000000 ns
Time: 8000000 ns
Time: 9000000 ns
Time: 10000000 ns
Simulation complete - no mismatches!!!
$finish called at time : 10 ms : File "/home/tuh42003/2613_2018f/lab9/tb_display_driver.sv" Line 105
run: Time (s): cpu = 00:00:00.02 ; elapsed = 00:00:08 . Memory (MB): peak = 1359.289 ; gain = 0.000 ; free physical = 6508 ; free v
irtual = 14114
## exit
INFO: [Common 17-206] Exiting xsim at Tue Oct 30 14:10:08 2018...
Compressing vcd file to lxt2 file. ...
```

Hardware Implementation

My design was demonstrated to Catherine on the afternoon of 10/29 during the lab period.

Conclusion

This lab was the first time seeing a clock signal first hand. The experience allowed me to understand sequential logic much better along with how finite state machines and data paths work. This design is used every day in many LED lights used in various clocks and other items.