Nicholas Scott AKA “White Cheddar”

A01255181 | Jan 22nd, 2023

ELEX 7660

Lab 2 – rotary encoder input

Contents

[Prelab – encoder.sv 2](#_Toc157059448)

[Lab2.sv Modifications 3](#_Toc157059449)

[Improvements – enc2bcd.sv 4](#_Toc157059450)

**Table of Figures**

[Figure 1 - 'encoder.sv' Simulation Waveforms 3](#_Toc157059463)

[Figure 2 - 'encoder.sv' Simulation Transcript 3](#_Toc157059464)

[Figure 3 - Compilation Report 5](#_Toc157059465)

[Figure 4 - Full RTL Netlist 0](#_Toc157059466)

[Figure 5 - 'encoder.sv' RTL Netlist 1](#_Toc157059467)

[Figure 6 - 'enc2bcd.sv' RTL Netlist - "The Abomination" 2](#_Toc157059468)

# Prelab – encoder.sv

I implemented the following code for the encoder.sv module:

// ELEX 7660 Lab 2

// Author: Nicholas Scott AKA "White Cheddar" - SN: A01255181

// Date: Jan 23rd, 2024

// Instructor: Sweet Bobby T

module encoder (input logic a, b, clk,

                output logic cw, ccw);

    logic [1:0] state;  // State of the encoder output

    logic [1:0] previous\_state;

    always\_ff @(posedge clk) begin

        state <= {a,b};

        previous\_state <= state;

        if (((state == 'b10) && (previous\_state == 'b00)) ||

            ((state == 'b11) && (previous\_state == 'b10)) ||

            ((state == 'b01) && (previous\_state == 'b11)) ||

            ((state == 'b00) && (previous\_state == 'b01))) begin    // True for all cases of cw rotation

                cw <= 1;

                ccw <= 0;

            end

        else if (((state == 'b01) && (previous\_state == 'b00)) ||

                ((state == 'b11) && (previous\_state == 'b01)) ||

                ((state == 'b10) && (previous\_state == 'b11)) ||

                ((state == 'b00) && (previous\_state == 'b10))) begin // True for all cases of ccw rotation

                    cw <= 0;

                    ccw <= 1;

                end

        else begin  // Set both cw and ccw to 0 if no rotation

            cw <= 0;

            ccw <= 0;

        end

    end

endmodule

Not the most elegant code I’ve ever written, but I don’t really give a darn. This code yielded the following simulation waveforms and transcript:

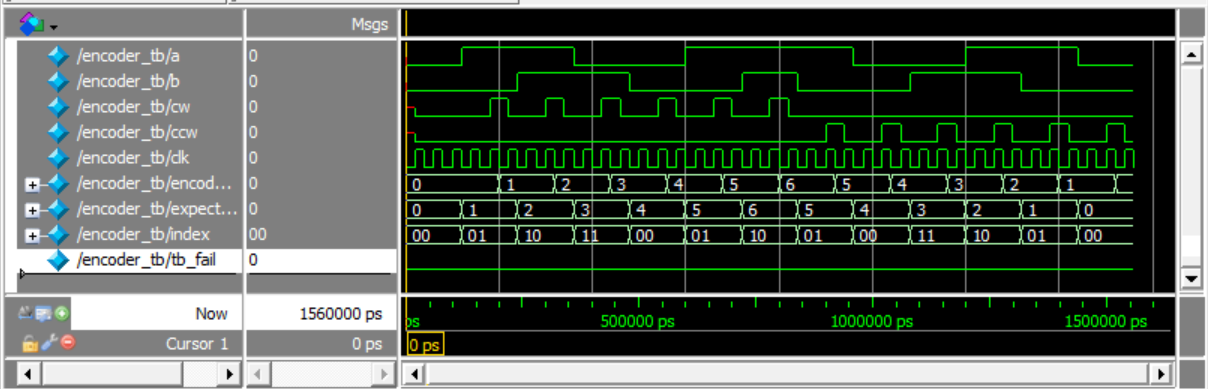


Figure - 'encoder.sv' Simulation Waveforms

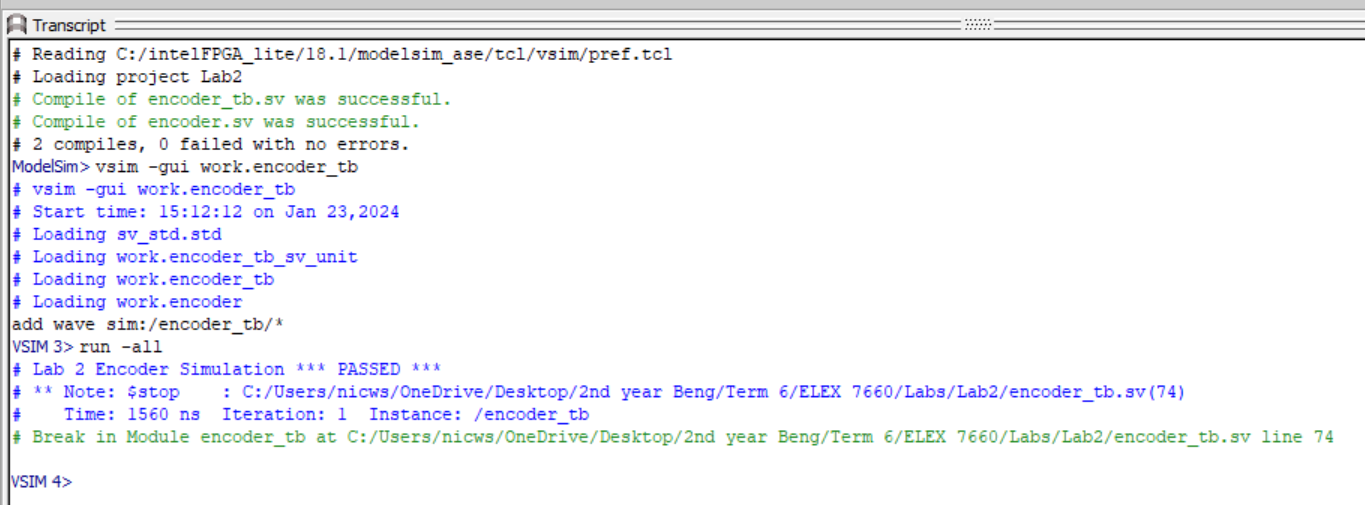


Figure - 'encoder.sv' Simulation Transcript

# Lab2.sv Modifications

I added the following code to lab2.sv to display the digits:

// Select digit to display (disp\_digit)

// Left two digits (3,2) display encoder 1 hex count and right two digits (1,0) // display encoder 2 hex count

  always\_comb begin

    case (digit)

      0 : disp\_digit = enc2\_count[3:0]; // Set digit 0 to encoder 2 count LSN

      1 : disp\_digit = enc2\_count[7:4]; // Set digit 1 to encoder 2 count MSN

      2 : disp\_digit = enc1\_count[3:0]; // Set digit 2 to encoder 1 count LSN

      3 : disp\_digit = enc1\_count[7:4]; // Set digit 3 to encoder 1 count MSN

    endcase

  end

It’s always been my life goal to coin a phrase. Maybe LSN (Least Significant Nibble) will catch on.

# Improvements – enc2bcd.sv

The code I wrote to this point worked great. I then began working on the lab improvements. I wrote the following code for the enc2bcd.sv:

// ELEX 7660 Lab 2

// Author: Nicholas Scott AKA "White Cheddar" - SN: A01255181

// Date: Jan 23rd, 2024

// Instructor: Sweet Bobby T

module enc2bcd (input logic clk, cw, ccw,

                output logic [7:0] bcd\_count);

    logic [7:0] countup;

    logic [7:0] countdown;

    always\_ff @(posedge clk) begin

        if (cw)

            countup <= countup + 1; // Count up for cw movement

        else if (ccw)

            countdown <= countdown + 1; // Count "down" for ccw movement

        if (countup >= 4) begin

            bcd\_count <= bcd\_count + 1; // Increment bcd\_count if countup reaches 4

            countup <= 0;

        end

        else if (countdown >= 4) begin

            bcd\_count <= bcd\_count - 1; // Decrement bcd\_count if countdown reaches 4

            countdown <= 0;

        end

        if ((cw) &&

            ((bcd\_count == 'h9) ||

            (bcd\_count == 'h19) ||

            (bcd\_count == 'h29) ||

            (bcd\_count == 'h39) ||

            (bcd\_count == 'h49) ||

            (bcd\_count == 'h59) ||

            (bcd\_count == 'h69) ||

            (bcd\_count == 'h79) ||

            (bcd\_count == 'h89)))   // If right digit is 9 and movement is cw

            bcd\_count <= bcd\_count + 7; // Increment count by 7 to skip hex A-F

        else if ((ccw) &&

                ((bcd\_count == 'h10) ||

                (bcd\_count == 'h20) ||

                (bcd\_count == 'h30) ||

                (bcd\_count == 'h40) ||

                (bcd\_count == 'h50) ||

                (bcd\_count == 'h60) ||

                (bcd\_count == 'h70) ||

                (bcd\_count == 'h80) ||

                (bcd\_count == 'h90))) // If right digit is 0 and movement is ccw

            bcd\_count <= bcd\_count - 7; // Decrement count by 7 to skip hex F-A

        else if ((cw) && (bcd\_count == 'h99))

                bcd\_count <= 0;                 // cw rollover from 99->0

        else if ((ccw) && (bcd\_count == 0))

                bcd\_count <= 'h99;              // ccw rollover from 0->99

    end

endmodule

This code is a bit of a disasterpiece, but it worked! That’s good enough for me. I instantiated two of these in lab2.sv and compiled. It yielded the following compilation report and RTL Netlists:

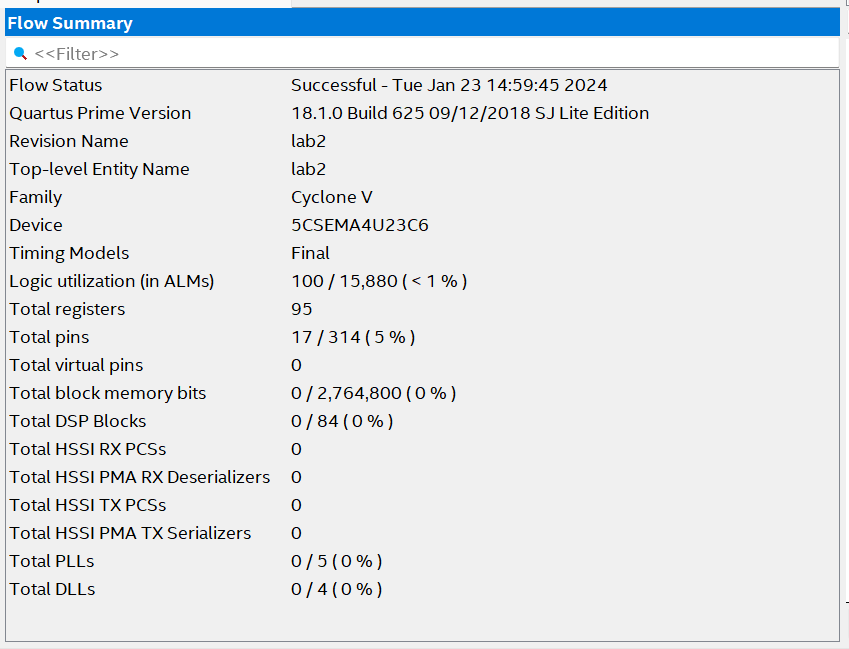


Figure - Compilation Report

I can’t get rid of this page for some reason. Word does weird stuff when you make some of the document ‘landscape’ and some ‘portrait’. I guess I’ll use the space to show the code I added to lab2.sv to instantiate these modules:

   enc2bcd enc2bcd\_1 (.clk(CLOCK\_50), .cw(enc1\_cw), .ccw(enc1\_ccw), .bcd\_count(enc1\_count));

   enc2bcd enc2bcd\_2 (.clk(CLOCK\_50), .cw(enc2\_cw), .ccw(enc2\_ccw), .bcd\_count(enc2\_count));

I then commented out this part:

  // encoder counts: enc1\_count & enc2\_count (increment when cw=1, decrement when ccw=1)

  // always\_ff @(posedge CLOCK\_50)  begin

  //   // encoder 1 count

  //   if (enc1\_cw) enc1\_count <= enc1\_count + 1'b1;

  //   else if (enc1\_ccw) enc1\_count <= enc1\_count - 1'b1;

  //   //encoder 2 count

  //   if (enc2\_cw) enc2\_count <= enc2\_count + 1'b1;

  //   else if (enc2\_ccw) enc2\_count <= enc2\_count - 1'b1;

  // end

The rest of lab2.sv was left as-is, aside from what I added before. I wasn’t sure if you wanted to see that or not.

A diagram of a computer program

Description automatically generated

Figure - Full RTL Netlist

A diagram of a computer

Description automatically generated

Figure - 'encoder.sv' RTL Netlist

A diagram of a machine

Description automatically generated

Figure - 'enc2bcd.sv' RTL Netlist - "The Abomination"