Nicholas Scott AKA “White Cheddar”

A01255181 | Jan 22nd, 2023

ELEX 7660

Lab 2 – rotary encoder input

Contents

[Introduction 1](#_Toc117435008)

[Lab Preparation 1](#_Toc117435009)

[Conclusion 1](#_Toc117435010)

**Table of Figures**

No table of figures entries found.

**Table of Tables**

No table of figures entries found.

# Prelab

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

// ELEX 7660 Lab 2

// Author: Nicholas Scott AKA "White Cheddar"

// Instructor: Sweet Bobby T

module encoder (input logic a, b, clk,

                output logic cw, ccw);

    logic [1:0] state = {a,b};

    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

                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

                    cw <= 0;

                    ccw <= 1;

                end

        else begin

            cw <= 0;

            ccw <= 0;

        end

    end

endmodule

This code yielded the following simulation waveforms:

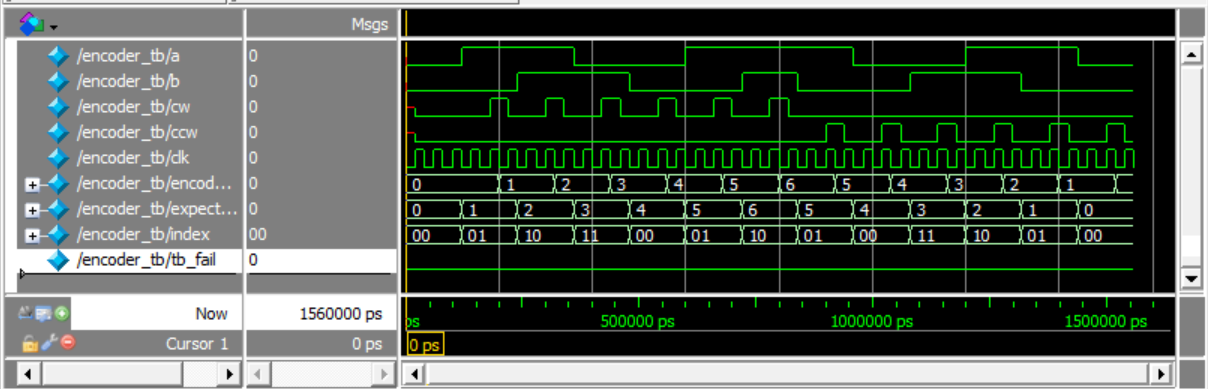


Figure - 'encoder.sv' Simulation Waveforms

A screenshot of a computer

Description automatically generated

Figure - Compilation Report