Encoders and Decoders Lab Report

Submitted to:

Dr. Gregg Chapman

Srinivastin Subramaniyan

Created by:

Group 18

Ansh Pachauri

Rushi Bhatt

Siwei Fan

ECE 2060

The Ohio State University

Columbus, OH

23 February 2024

Executive Summary

In order to create successfully working decoders and encoders, the goal of lab 3 was to investigate gate level logic design in conjunction with the use of VHDL files and ModelSim software. Using the truth table that was provided in the lab for both the encoder and decoder phases, the team created a set of Karnaugh maps. Thereafter, the two circuits were developed using the Quartus Prime application. The circuit designs were all successfully simulated in the ModelSim program after each circuit was finished and successfully compiled. The group then used the previously described circuit to create VHDL codes for the encoder and decoder. After that, this file was successfully used by the encoder and decoder in the ModelSim program.

This lab report's length focuses on the history and objectives of the experiment, then moves on to discuss the experimental design and methods that were employed to effectively complete the lab. The circuits that were constructed, the karnaugh maps that were used to create the devices, the VHDL code, and the ModelSim results are all included in the results section that follows. After discussing a series of discussion questions about different lab components, the group reviewed the lab's summary and confirmed that the main objectives had been achieved. Together with the references consulted in order to finish the lab and this report, the team has also provided acknowledgements for each member of the team.

Introduction

Creating a 2 to 4 decoder and a 4 to 2 encoder was the goal of lab 3. The group investigated developments in gate level logic design as well as familiarity with VHDL files and ModelSim software as a result of these devices. To create the proper encoder and decoder results, the team also used Karnaugh maps. All things considered, this lab enables the team to integrate many methods and tools into a single lab to connect the different applications and course skills that have been mastered. Below is a description of the lab 3 technique in more depth.

Experimental Methodology

To begin the lab, the team was tasked with designing a 2 to 4 decoder. To achieve this, Group 18 first drew a Karnaugh Map from a given truth table to find the associated Boolean expressions for each output. Group 18 found four Karnaugh Maps for the given inputs (A0, A1) and outputs (D0, D1, D2, D3) in a 2x2 cell format.

Group 18 first opened up the Quartus Prime application to construct the 2 to 4 decoder circuit. Once on this software they selected the tools menu, followed by the options button. They then selected the EDA tool options, and then in the ModelSim-Altera section, apply the code, “C:\intelFPGA_lite\18.1\modelsim_ase\win32aloem.” Group 18 used a combination of NOT gates and AND gates to realize the decoder. Next, the group simulated the design using ModelSim. Screenshots of each result were collected and recorded in the following section.

Group 18 developed a VHDL implementation for a 2 to 4 decoder by initially crafting a new VHDL file. With the given decoder code sans Boolean expressions, the team referenced expressions like “D(2) <= A(1) and not A(0);” as models, derived from equations discerned through Karnaugh maps, outlined later. After completing the decoder code, it was designated as the top-level entity and integrated into the project compilation.

For the subsequent part of the lab, Group 18 designed a 4 to 2 encoder employing analogous procedures utilized for the decoder. Beginning with K-maps generated from a provided truth table, boolean expressions were derived, accommodating 'don't care' outputs for unlisted input combinations. Group 18 streamlined these expressions by grouping ones and 'don't cares' to minimize hardware requirements for the encoder circuitry.

Following expression derivation via K-maps, the encoder was designed in a new .bdf file utilizing gates. Simulation in ModelSim ensued, employing clocks with varied periods (50 ps, 180 ps, 200 ps, and 400 ps) for a duration of 1800 ps. Simulation results were scrutinized against an anticipated output screenshot post the timed clock simulation.

Lastly, Group 18 replicated the design of the 4 to 2 encoder using VHDL, leveraging the provided reference code and encoder Karnaugh maps to formulate expressions. After writing the code, simulation in ModelSim was conducted, and the outcomes were documented.

Below are the encoder and decoder truth tables Group 18 used to construct each simulation.

Inputs		Outputs			
A1	A0	D3	D2	D1	D0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

Figure 1 - Decoder Truth Table

Inputs				Outputs	
A3	A2	A1	A0	D1	D0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

Figure 2 - Encoder Truth Table

Results

Following Lab 2, Group 18 has acquired diverse schematics and VHDL symbols pertaining to both the 2 to 4 decoder circuit and the 4 to 2 encoder gate. The VHDL symbol associated with the 2 to 4 decoder gate regulated its behavior, facilitating verification of output correctness against a truth table via compilation and execution of the code on Quartus Prime, utilizing a timed clock diagram. A similar process was undertaken for the 4 to 2 encoder gate. Group 18 effectively conceived and implemented both the 2 to 4 decoder and 4 to 2 encoder schematics, subsequently validated through timed clock diagram simulations. Each input-output pairing, whether true or false, precisely corresponded with the provided truth tables for both gates. Below are the outcomes collected by Group 18, encompassing decoder and encoder block diagrams, VHDL code, and simulation results.

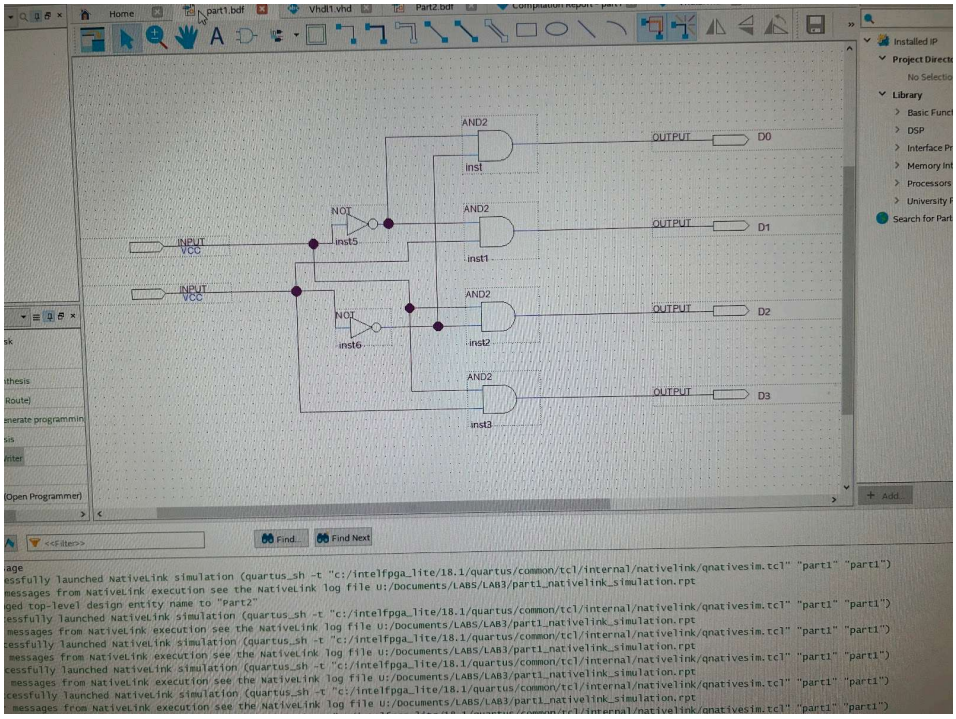

Figure 3 - Decoder Block Diagram

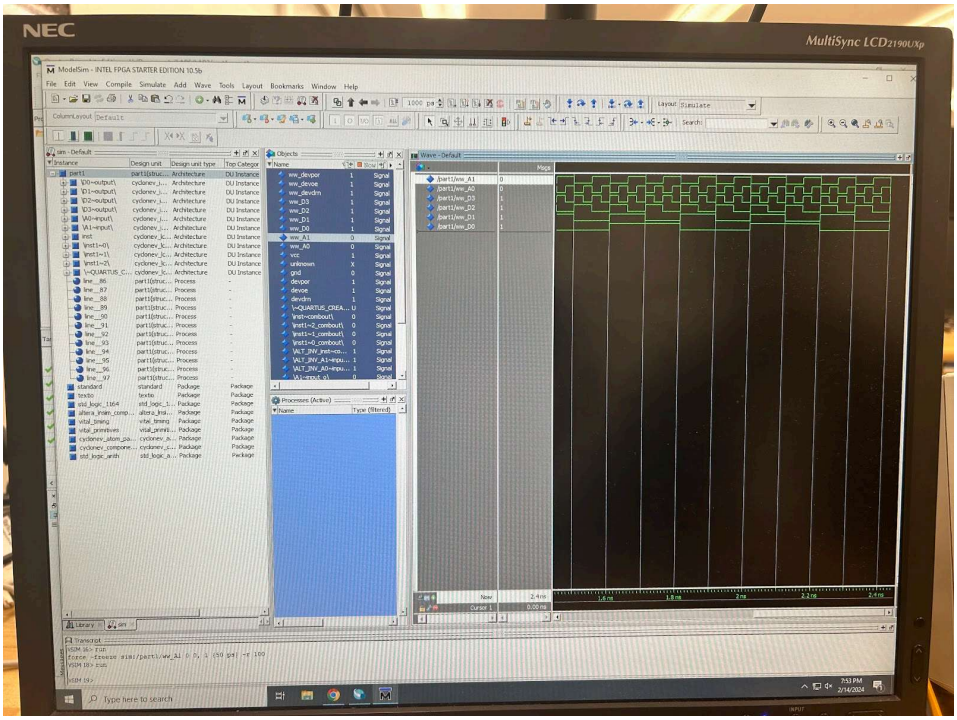

Figure 4 - Decoder Block diagram ModelSim Simulation

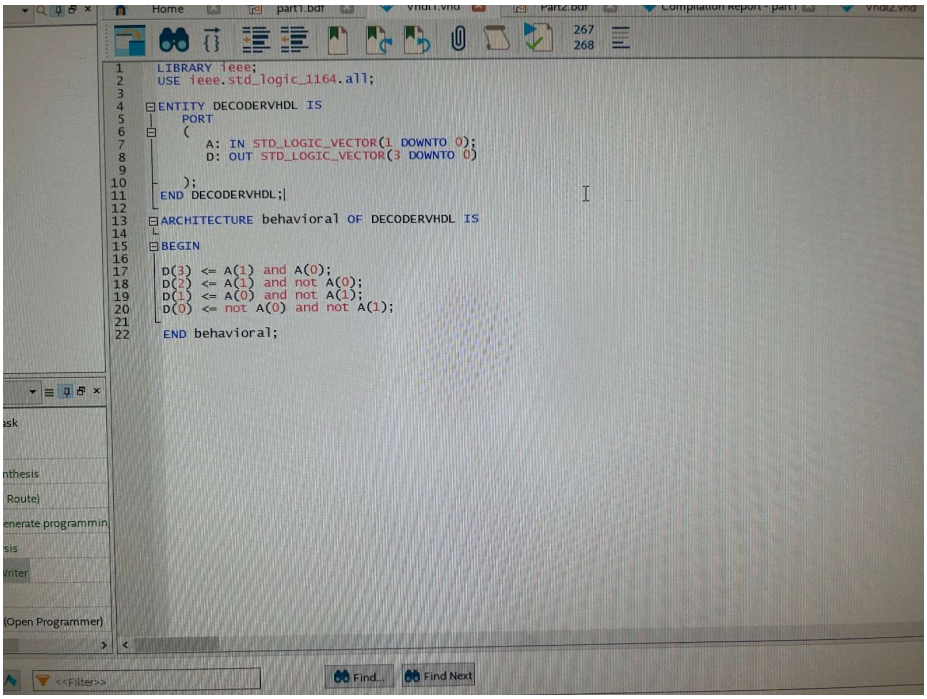

Figure 5 - Decoder VHDL

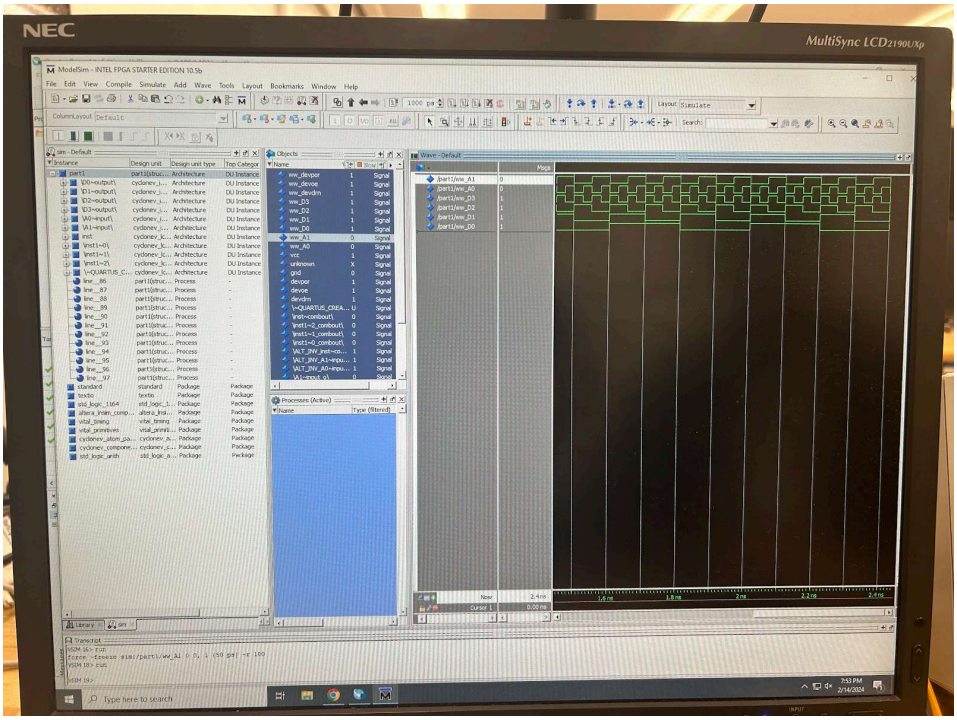

Figure 6 - Decoder VHDL ModelSim Simulation

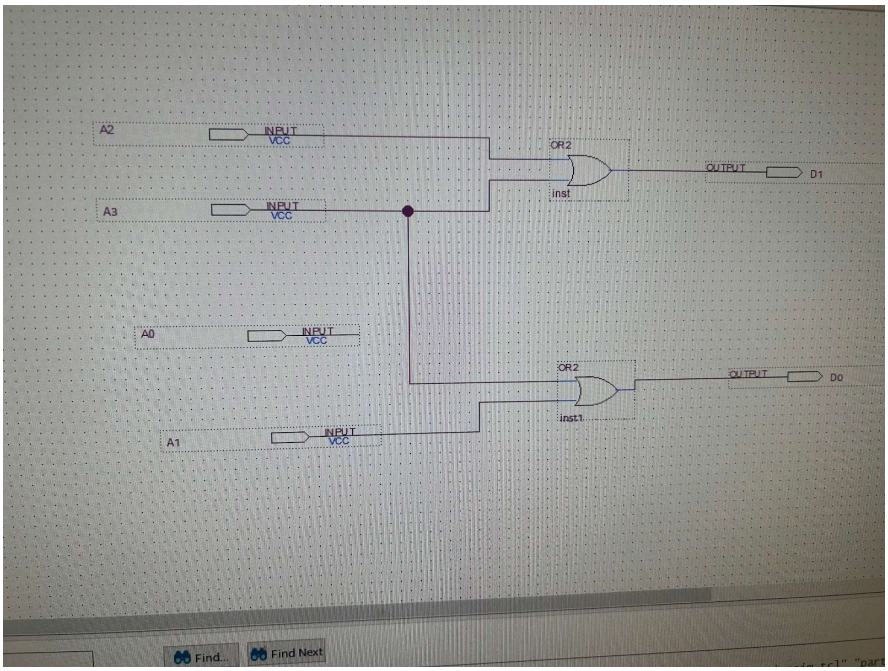

Figure 7 - Encoder Block Diagram

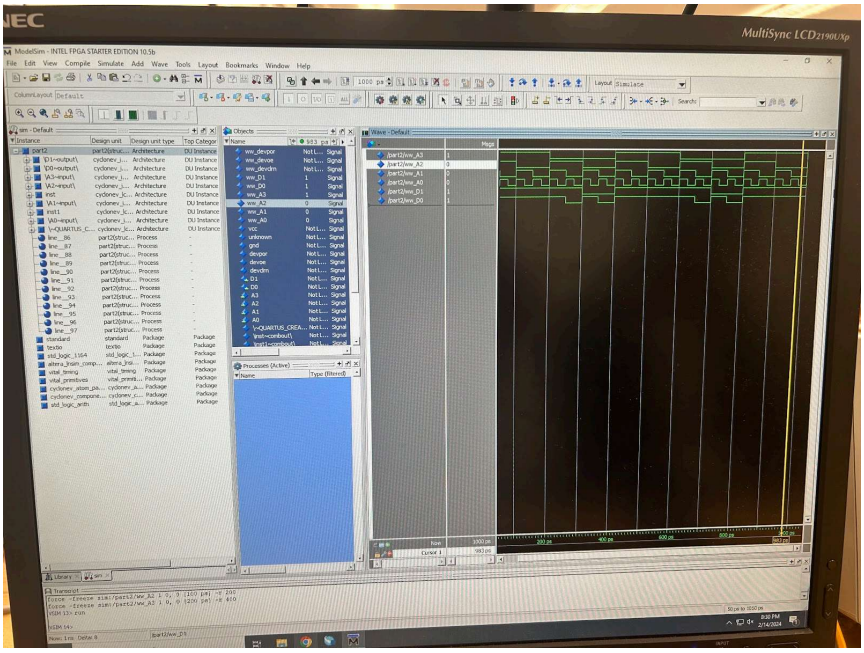

Figure 8 - Encoder Block Diagram ModelSim Simulation

```
1   LIBRARY ieee;
2   USE ieee.std_logic_1164.all;
3
4   ENTITY ENCODER IS
5       PORT
6       (
7           A: IN STD_LOGIC_VECTOR(3 DOWNTO 0);
8           D: OUT STD_LOGIC_VECTOR(1 DOWNTO 0)
9
10      );
11  END ENCODER;
12
13  ARCHITECTURE behavioral OF ENCODER IS
14
15  BEGIN
16
17      D(1) <= A(3) or A(2);
18      D(0) <= A(3) or A(1);
19
20  END behavioral;
```

Figure 9 - Encoder VHDL

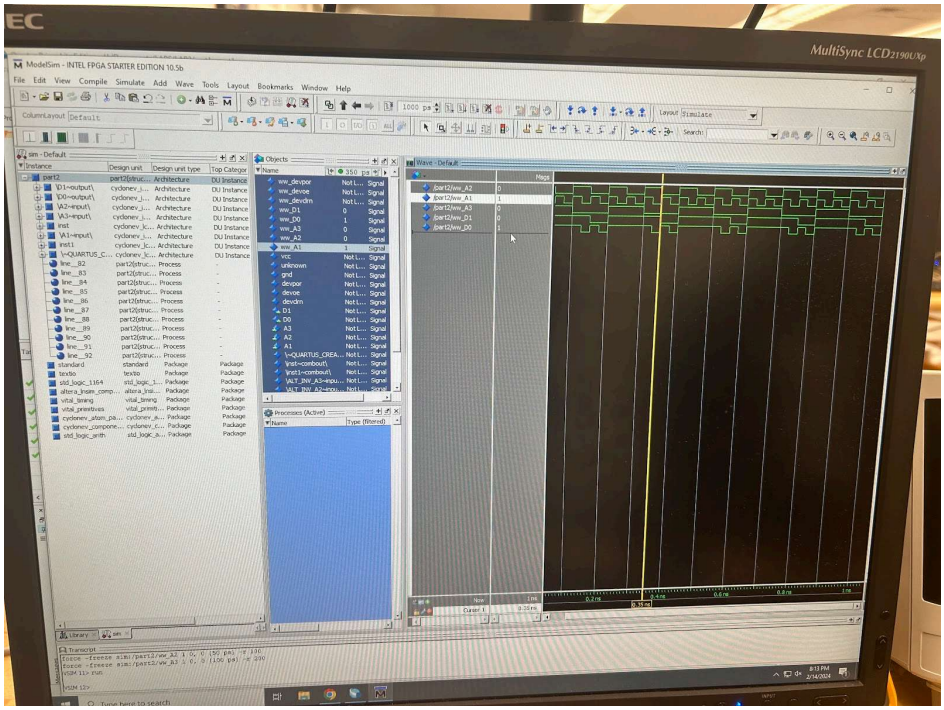

Figure 10 - Encoder VDHL ModelSim Simulation

Discussion

To monitor the status of 16 discrete digital inputs effectively, we employ a 4-to-16 line decoder acting as a multiplexer. This involves linking 4 bits from the 8-bit input to the address lines of the decoder, subsequently connecting the 16 discrete digitals to its outputs. The remaining 4 bits from the 8-bit input serve as control inputs, enabling the monitoring of the 16 discrete digital inputs. The proposed intermediate circuit integrates microcontrollers alongside a 3-to-8 line decoder and a 4-to-16 line decoder. The 3 bits from the microcontroller output interface with the 3-to-8 line decoder, facilitating the selection of one of its 8 outputs. Further, by routing 4 outputs from the microcontroller to the address lines of the 4-to-16 line decoder, we enable the selection of one of the 16 outputs from this decoder. Ultimately, the outputs of these two decoders are linked to the 14 discrete digital outputs to regulate their states, determined by the address line combination. This setup yields 24 outputs (6 + 18 controls), although only 14 outputs are required, leaving 18 unused output combinations and 1 unused bit. The Boolean functions for the outputs of an 8-to-3 encoder can be computed as follows: D2 = A4 + A5 + A6 + A7, D1 = A2 + A3 + A6 + A7, D0 = A1 + A3 + A5 + A7. These expressions are derived from the provided truth table in the laboratory resources, representing the behavior of an 8-to-3 encoder.

Conclusion

In conclusion, this lab was extremely beneficial to learn how decoders and encoders are computed and implemented through the QuartusPrime and ModelSim softwares. This lab allowed Group 18 to first learn how to design a 2 to 4 decoder by drawing a Karnaugh Map, deriving the expression, and building the circuit in QuartusPrime. The group learned collectivelyt how to write functional VDHL code and the varying logic behind it. The group learned to implement a 4 to 2 encoder by following these same steps. Overall, this lab was helpful for learning how to create a functioning circuit model from scratch and assess its success via ModelSim. The results highlighted above demonstrate a successful laboratory outcome from following the necessary steps taken to complete the assignment.

Acknowledgments

All members of Group 18 contributed equally and participated in all lab activities. In terms of the lab report, Siwei wrote the discussion and conclusion sections. Ansh described the executive summary, introduction, and edited the document. Rushi wrote the experimental methodology and results sections.

References

- Lab report template - Lab-Report-Submission-Instructions
- Lab 3 document - encoderdecoderLab