Lab 4 – Combinational Circuits

CS 2052 Computer Architecture

Dept. of Computer Science and Engineering, University of Moratuwa

Learning Outcomes

In this lab we will design a decoder and a multiplexer. Decoders and multiplexers are 2 of the key components of a microprocessor. After completing the lab, you will be able to:

- design and develop a 3-to-8 decoder using schematics
- design and develop a 8-to-1 multiplexer using schematics
- verify their functionality via simulation and on the development board

Introduction

A decoder converts binary data from *n* coded inputs to a maximum of 2^n unique outputs. The decoder that we are going to build also has an enable pin/input. Enable input must be on for the decoder to function, otherwise we assume its outputs as a "disabled" output. A multiplex receives binary data from 2^n lines and connect them to a single output line based on a given *n*-bit selection. We will also add an enable pin to the multiplexer.

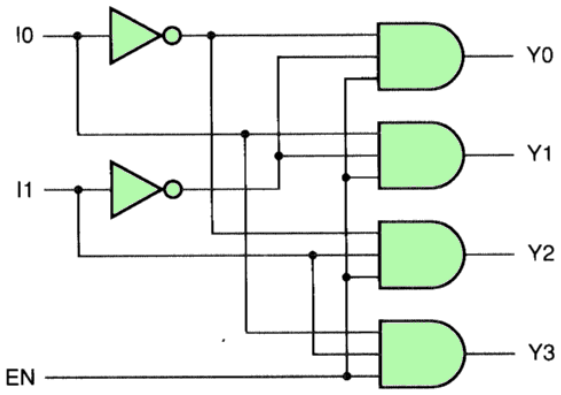
Decoders and multiplexers are 2 key components of a microprocessor. We will later use these components to build our simple microprocessor.

Building the Circuits

We will first build a 2-to-4 decoder. Then by combining two 2-to-4 decoders we will build a 3-to-8 decoder.

Step 1: Building a 2-to-4 decoder.

Build the decoder shown in the following logic circuit. Name the project as **Lab 4** and the schematic file as **Decoder_2_to_4**.



Source: http://users.cis.fiu.edu/~prabakar/cda4101/Common/notes/lecture08.html

Step 2: Simulating circuit.

Simulate the circuit using Xilinx ISim and make sure your decoder functions correctly. Name the Test Bench File as **TB_Decoder_2_to_4**.

Step 3: Create a decoder symbol and name it as **Decoder_2_to_4**.

. Save changes to the current schematic editor session before closing the schematic file and return to the Project Navigator main window.

Step 4: Build a 3-to-8 decoder.

Create a new schematic file and save it as **Decoder_3_to_8**.

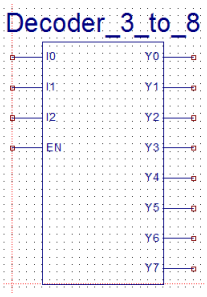
Now you should see **Decoder_2_to_4** listed as one of the symbols.

Using two 2-to-4 decoders and other gates, build a 3-to-8 decoder. Label your input pins as **I0** to **I2** and output pins as **Y0** to **Y7**. Make sure to add an enable pin and label is as **EN**. Hint: You may need AND and NOT gates.

Simulate the new decoder using Xilinx ISim and make sure it functions correctly. Name the Test Bench File as **TB_Decoder_3_to_8**. It is not essential to try all the possible combinations of inputs. Instead, you can try a subset of the possible inputs. Here is a method to try several input combinations based on your index number (this also enables your instructor to make sure each student uses a unique simulation pattern).

Consider the 6 digits of your index number. Then convert your index number to binary. Set **I2** to **I0** according to the 3 Least Significant Bits (LSBs) of your index number. Then try the next 3 LSBs, and so on. For example, suppose your index number is 123456R. Then its binary representation is 011 110 001 001 000 000 (ignoring the check digit). Try setting I**2** - **I0** as 000, then 001, 110, and so on. You may ignore repetitive values.

Using the **Symbol Wizard** create another symbol based on our 3-to-8 decoder. Name it as **Decoder_3_to_8**. Symbol of the completed decoder should be similar to the following.



Step 5: Designing an 8-to-1 multiplexer.

Take a piece of paper and draw a logic circuit of an 8-to-1 multiplexer built using 3-to-8 decoder. You may also need several AND and OR gates. Label your control pins as **S0** to **S2**. Label your input pins as **D0** to **D7** and enable as **EN**. Label the output as **Y**.

Step 6: Build the 8-to-1 multiplexer.

Name the schematic file as **Mux_8_to_1**.

Step 7: Simulating the multiplexer.

Similar to Step 4 try a set of input combinations based on your index number. Make sure your multiplexer is functioning correctly.

Step 8: Connecting inputs and outputs.

Input and output pins are connected only to the top-level design. Therefore, first we have to set our multiplexer as the Top Design Unit. Select **Mux_8_to_1** from the Sources window. Then use **Source → Set as Top Model** from the main menu. Now we are ready to assign pins.

Connect switches **SW0** to **SW7** as the inputs (from **D0** to **D7**) to **Mux_8_to_1**. Connect push button **BTN0** to **EN**. Connect **BTN1** to **BTN3** to control pins **S0** to **S2**. Set output **Y** to LED **LD0**.

Step 9: Test on BASYS2.

Generate the programming file (i.e., bitstream) and load it to the BASYS2 board.

Change the switches and push buttons on the BASYS 2 and verify the functionality of your multiplexer (check the output on LED).

Demonstrate the circuit to the instructor and get the Lab Completion Log singed.

Step 10: Lab Report

You need to submit a report for this lab. Your report should include the following:

- Student name and index number. Do not attach a separate front page
- State the assigned lab task in a few sentences
- All schematic circuits (label figures)
- All timing diagrams. Show all possible inputs for 2-to-4 decoder. For 3-to-8 decoder and 8-to-1 multiplexer use your index number as the input
- Conclusions from the lab

Submit the lab report at the beginning of the next lab.

Bibliography

- Digilent Inc., “Xilinx ISE WebPACK Schematic Capture Tutorial,” Feb. 27, 2010.

Prepared By

- Dilum Bandara, PhD – Feb 19, 2014.
- Updated on Sep 14, 2017.