SER232 - Assignment 3

[10 Points]

Description

This assignment covers the different representations of combinational circuit and requires you to apply the conversion process to convert between the different circuit representations.

In this assignment you have to create your own 1x8 demultiplexer (demux) and 3x8 decoder with the three basic logic gates: AND, OR, NOT, using the design process covered in class (designing a truth table, deriving Boolean equations from the table, creating a circuit from equations). With those components, you will complete a verification circuit (provided in a template) that will display the decimal value of the binary input on an array of hexadecimal displays.

Tasks

This assignment has two parts:

1. Create a Design Document for the demux and the decoder.
2. Create the circuit in Logisim.

1. Design Document

First, you will create a Design Document that captures the behavior of each component. The Design Document must include the truth table and the Boolean equations for each of the two components: a 1-bit 1x8 demux and a 1-bit 3x8 decoder.

Do the following tasks in order to create your Design Document:

1. Create the truth table for the 1-bit 1x8 demux. You must use the following labels: *i* input; *d0* through *d7* for the outputs; *s0*, *s1*, *s2* for the select inputs.
2. Derive the Boolean equation(s) from the truth table. Make sure to follow the circuit conversion process covered in the lectures. Do not simplify the Boolean equations, which means they will and must be in the canonical sum-of-minterms form.
3. Create the truth table for the 3x8 decoder. You must use the following labels: *a0* through *a2* for the input, and *e0* through *e7* for the outputs.
4. Derive the Boolean equation(s) from the truth table. Make sure to follow the circuit conversion process covered in the lectures. Do not simplify the Boolean equations, which means they will and must be in the canonical sum-of-minterms form.

Important:

- The truth tables must cover all possible input combinations.
- You are only allowed to use 0 or 1 for the truth table (except column headers).
- If you have inputs and/or outputs with the same letter and just a different index (e.g. input 0 through 3), sort these columns in decreasing order in the truth table (e.g. i7, i6, i5, i4, i3, i2, i1, i0).
- Use a proper tool/program to create the table (e.g. Google Sheets). Hand-written or -drawn submissions will not be accepted.
- Format your truth table properly (e.g. using the same content alignment for all cells, use of cell borders, etc.). Points will be deducted if the table is hard to read.

2. Circuit

Secondly, you will use the Design Document to create your circuits. With the derived Boolean equations from the Design Document you will create a *1-bit 1x8 Demux* and *3x8 Decoder* in Logisim.

Complete the following tasks:

1. Download the provided assignment template. (If Canvas adds a ".xml" extension during the download: Please remove it, so the file extension is ".circ" again.)
2. Use the Boolean equations of each component from your Design Document to create the circuits for *1-bit 1x8 Demux* and *3x8 Decoder.*
3. Test each subcircuit individually and make sure it works as intended. If it deviates from the expected behavior, double check your Design Document and the circuit. Make sure the 2-bit and 4-bit demux also works properly (those are already implemented and depend on your 1-bit demux implementation). For more details about multi-bit demux / 4-bit demux, see section *4-bit 1x8 Demultiplexer* below.
4. Complete the verification subcircuit. The verification circuit uses one 3-bit input *i* to input a 3-bit binary number into the system. The system then displays the decimal representation (or hexadecimal, since they are identical for the value range 0-7 in this assignment) of that binary number on one of the hex displays. The display showing the number matches the decimal value (e.g. if the value 011 is sent into the system, the third hex display (Hex 3) will show the value 3 and the remaining displays will show a 0). This behavior is realized by using the demux, which will route the input value to the respective hex display due to using the input value also as select bits.

   Note: Since the demux requires a 4-bit input, but the system only has a 3-bit input, you will get a *incompatible widths* error when directly connecting the system input to the demux input. In order to avoid this issue, a fourth bit has to be added. To not change the value of the input, a constant 0 can be added as the most significant bit. The constant component with value 0 was already added to the template and can be used in conjunction with a splitter to solve this issue as described above.

Lastly, the decimal dot of the currently used hex display (always only one will be used at a time) must be lit up, all other decimal dots must be off. To light up the decimal dot, the right 1-bit input of the hex display can be used (1 will make it light up, 0 will make it turn off). For this functionality, the decoder will be used, which will turn on exactly one output based on the input value.

4-bit 1x8 Demultiplexer

A 4-bit demux works the same as a 1-bit demux, the only difference is that the input and the outputs have 4-bit widths instead of a 1-bit width. Besides this difference the function and behavior is identical.

This 4-bit width is necessary for demux which must route a 4-bit signal (like in this assignment).

Deliverables

The following deliverables must be submitted on Canvas before the due date as a single submission:

1. Submit your Design Document as *.pdf*, named: *lastname_a3design.pdf*
2. Submit your Logisim assignment file, name: *lastname_a3.circ*

Important:

- You are not allowed to use the built-in demux or decoder of Logisim for this assignment. You will be given a 0 if you use them.
- Make sure to submit both files in one submission. Only the files of the last submission will be graded. If one is missing, you will receive 0 points for the missing part.
- Using the tunneling feature of Logisim is not allowed for this (or any other) assignment.
- Do not modify the template by removing existing elements or wiring. Removing elements from the template results into a deduction. Moving existing elements is not necessary to finish this assignment.