# COMP 2421 Computer Organization
# Digital Logic Programming Assignment

**Deadline: 11:59 pm, Apr. 14 (Sunday), 2024.**

## 1 OBJECTIVE

The objective of this assignment is to use combinational logic to implement some simple real-world applications, using Logisim. By completing the tasks, we will implement the most basic display function for an elevator – after pressing a button "1" representing floor 1, the 7-Segment will display the corresponding digit. Fig. 1 illustrates this function in details. Specifically, there are eight buttons numbered through 0 to 7, and a 7-Segment display, which are connected through some combinational logic. If button $i$ ($i = 0, 1, \cdots, 7$) is pressed, the 7-Segment will display the corresponding digit.

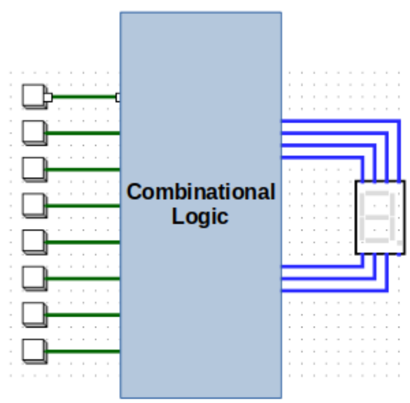


Fig. 1. Task Overview

Your task is thus to implement the combinational logic. This task is further divided to two major sub-tasks and one simple sub-task, described in the next section.

**Note: this task requires you to read the introduction and specifications of some components in Logisim, and learn by yourself based on what you have learned through the Lab sessions. Related references as well as examples are provided at the end of this document.**

## 2 TASK DESCRIPTION

The whole task is composed of three sub-tasks: (1) build up an encoder, (2) design the combinational logic, and (3) connect the two components.

## 2.1 Subtask One

First, you will need to design an encoder with 8 inputs and 3 outputs. Each of the input corresponds to one button and the 3 outputs (denoted as $E_0E_1E_2$) are the three-bit encoding of the input. Specifically, if button $i$ is pressed, the outputs $E_0E_1E_2$ would be the binary encoding of $i$. Example: if button 5 is pressed, $E_0E_1E_2 = 101$.

You will use the component Button, Priority Encoder, and Splitter from Logisim to implement this function, as shown in Fig. 2. Play with the Priority Encoder to make sure that it will produce the desired outputs. Especially, the bit width of the output of the Priority Encoder could be larger than 1. Thus, you will need a Splitter to split the output bits. To show the values of the outputs $E_0E_1E_2$, they are connected to three LED lights. As a result, when you press a button, the corresponding LED light(s) will turn on (become red). Play with this circuit until you got the desired results.

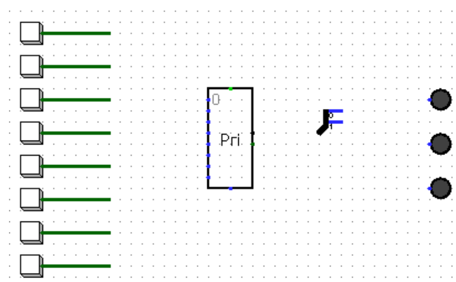


Fig. 2. Sub-task One.

## 2.2 Subtask Two

Second, we need to design the logic to control the states (ON/OFF) of each segment in the 7-Segment display. As shown in Fig. 3, the display contains seven segments labeled through $S_1$ to $S_7$. When provided with input bit 1, a segment will turn on (become red). Play with this 7-Segment display to know the correspondence between the inputs and the segments.

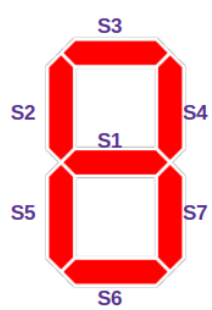


Fig. 3. 7-Segment Display

In this subtask, you will use three input lines $E_0E_1E_2$ to control the 7-Segment display. Specifically, $E_0E_1E_2$ present the binary form of a decimal digit $i$ ($i = 0, 1, \cdots, 7$), which will shown in the 7-Segment display. The designing process is similar to what you have learned in Lab 10. Specially, you will write a truth table with $E_0E_1E_2$

as inputs and $S_1S_2S_3S_4S_5S_6S_7$ as outputs. Carefully decide which segments will be turned ON for each input digit value. For example, when $i = 1$, we assume that segments $S_4$ and $S_7$ will be turned ON.

Based on the truth table, derive the functions relating each output $S_j$ with the inputs $E_0E_1E_2$. Finally, construct the circuit using AND and OR gates. To ease the grading process (and make the grader happy), you are suggested to organize the gates as shown in Fig. 4.

**You will use three "Pins" as the inputs $E_0E_1E_2$. The figure does not say that there are only three AND gates and two OR gates in the circuit. You can reuse the AND gates to reduce the number of gates.**

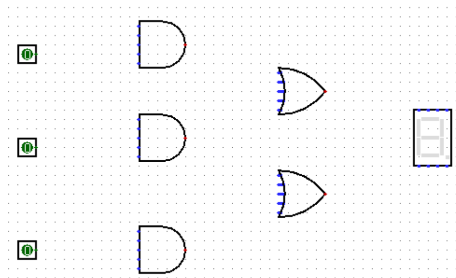

Fig. 4. Subtask Two.

## 2.3 Subtask Three

Now, you can combine the two circuits to achieve our desired functionality: click a button $i$, show the digit $i$ in the 7-Segment display. An example circuit is shown in Fig. 5. One thing to notice is that the orders of the bits for the output of the priority encoder and the input to the rest of the combinational logic might be different. You can easily adjust the order by modifying the attributes of the priority encoder.

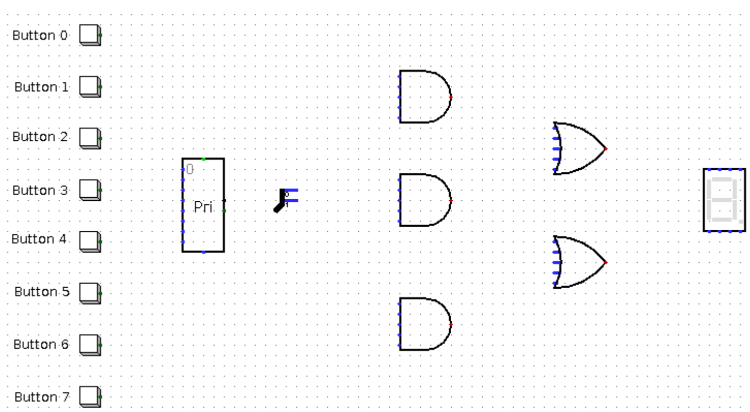


Fig. 5. Example of the whole circuit.

## 3 WHAT TO SUBMIT

(1) **Three circuits**. For each of the subtask, submit a circuit that implements the desired function. Name the files as " `subtask_one_XXXXXXXX.circ`", " `subtask_two_XXXXXXXX.circ`", and " `whole_circuit_XXXXXXXX.circ`", respectively, where "`XXXXXXXX`" represents your 8-digit student number (the last letter is not needed).

(2) **An experiment report**. The report should include the following contents:
- The truth table in subtask two and the functions relating the output $S_j$ and the input $E_0E_1E_2$.
- Screen shot of the three circuits in three subtasks.
- Three pictures. In each picture, you should show that when button $i$ is pressed, the digit $i$ is shown in the 7-Segment display (**Select three cases as you wish**).

## 4 ASSESSMENT

- Subtask One (30 pts)
- Subtask Two (50 pts)
- Subtask Three (10 pts)
- Report (10 pts)

## 5 USEFUL REFERENCES

(1) Specifications of all components in Logisim. http://www.cburch.com/logisim/docs/2.7/en/html/libs/plexers/index.html

(2) Example of using Priority Encoder. http://www.cs.unca.edu/ brock/classes/Fall2012/csci255/labs/lab03.html