LAB 2: COMBINATIONAL CIRCUITS (PART ONE)

This lab will focus on exploring the uses of various key combinational circuit components like encoders and decoders, comparators, etc., for a special use case: Tita Jo's seaside eatery.

As in Lab 1, please use Logisim-evolution or CircuitVerse when instructed to implement or design a circuit.

Part 1: Reading Orders

Tita Jo has 8 items on her eatery's menu: Sinigang, Lumpia, Bulalo, Manok Inasal, Lechon Kawali, Kanin, Taho, and Halo-Halo.

With lots of customers (and most of the waiters unavailable for most of the weekdays when things get busy), you will have to design a Customer Console system, with an "Ordering Machine" around three components:

Customers will get a *bubble sheet*, which they put into a *bubble card reader*. The bubble card readers can be considered 3-output machines, which return an output 0 for the corresponding output rail if the bubble is unfilled, and 1 if it does. *However, the bubble card reader can only read up to three bubbles at a time.*

These bubble sheet readers feed their outputs to an "Order Machine", which returns 8 possible outputs, connected to 8 different LEDs in the kitchen, assigned to turn on when an order is made.

The local electronics shop has completely run out of 3-bit decoders. As such, you will only have access to 2-bit decoders, AND, OR, and NOT gates. **With these resources, design and implement an "Ordering Machine."**

Part 2: Accepting by Availability

Availability of ingredients can vary greatly in the eatery from day to day, and *it is very important to ensure orders for unavailable dishes are not accepted*. In other words, if a customer orders a dish, but the ingredients are not available, we need the corresponding light indicator in the kitchen *not to turn on*. As such, the "Order Machines" implemented in Part 1 must receive an "Availability Mechanism" upgrade.

There are 8 ingredients in the pantry, and Tita Jo has provided you with her cookbook. Every ingredient is necessary, but some can be substituted for others:

Dish	Ingredients
Sinigang	Rice Pork or Vegetables
Lumpia	Pork or Vegetables
Bulalo	Beef or Chicken Rice or Vegetables
Manok Inasal	Chicken Rice Vegetables
Lechon Kawali	Pork Rice Vegetables
Kanin	Rice Tofu Vegetables
Taho	Sugar Tofu
Halo-Halo	Sugar Ice Cream

Inventory is taken every day, and input rails are provided to r each ingredient, to indicate when it is available or not.

Your local electronics shop, this time, has a backlog. Therefore, for this task, you will only have access to NOR and NOT gates. **Implement the "Availability Mechanism" upgrade to the Customer Console.**

Part 3: Payment

The last interaction left to automate is the customer payment. Each of the dishes cost an integer multiple of ₱5. (e.g., an order of Lumpia costs ₱5, an order of Lechon Kawali costs ₱10, and so on.) **Assign a price to each dish, and make sure each price is a unique multiple of ₱5, up to ₱40 at most.**

Tita Jo needs a new Payment Mechanism to be implemented into the Customer Console, as an additional check before an order can be accepted, in the same way the Availability Mechanism was a check before an order can be accepted (i.e., for the appropriate LED in the kitchen to switch on).

When paying, customers deposit their coins into a coin machine, where only ₱5 coins are accepted. This coin machine has a useful feature: it provides an 8-bit output C $(C_1, C_2, …, C_8)$, such that the jth bit (and all less significant bits) are turned "on" once j many coins are deposited. In particular, if the Customer pays j many ₱5 coins (totaling a payment of ₱(5*j)), then C_m =1 for $m \leq j$, while C_k = 0 for k > j.

Implement the "Payment Mechanism" update to the Customer Console. This time, the electronics shop will finally give you access to all of the components within. (In practical terms, this only includes components discussed in class up to the assignment of this lab.) However, money is also tight; the business cannot afford more than _three_ additional components in this design.