CMPE 212- Project 2

**Finite State Machine Controller**

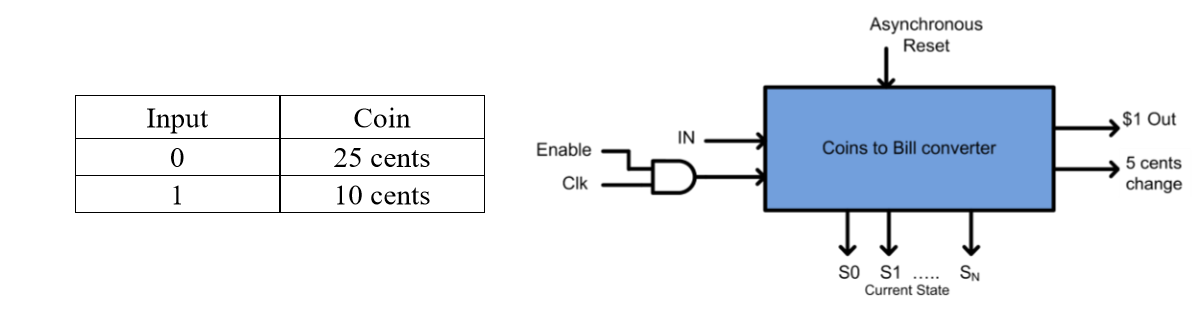
**Objective**

The objective of this project is to design a synchronous sequential circuit and implement it in Verilog.

**Project Description**

Almost all of the real-world electronic systems are designed with a help of one or multiple finite state machines. This project considers a finite state machine controller for a ‘coin to bills converter’ machine. To help needy children, a generous company has decided to offer $1 food voucher for every 70 cents dropped to the machine. You are required to design the state machine for the specifications provided below.

**Specifications**

The machine accepts only quarters (25 cents) and dimes (10 cents). The output $1 food voucher should be dispensed when the total coin amount reaches or exceeds 70 cents. If the amount exceeds 70 cents, the change should be applied as credit towards the next transaction, unless it equals 5 or 15 cents. For 5-cents change a nickel should be given back (dispensed); meanwhile a change of 15 cents should cause the machine to apply 10 cents towards the next transactions and give back a nickel. For the project, there should be a 1 bit input which specifies the following:

The figure illustrates the input and output to the controller. The asynchronous Reset input is to reset the state machine to its IDLE state. As the food voucher is dispensed, the state machine should reset itself to its initial (IDLE) state or a state that reflects the amount to be credited.

There should be 2 outputs denoting the following:

• 1 bit – Denotes the $1 out, which should be turned ON when the input amount reaches or exceeds 70 cents

• 1 bit – Denotes that a nickel is be dispensed and becomes ON in case of having 5 or 15 cents change.

You are to develop the state table and determine the number of bits “N” required for denoting the current state (and determine the number of flip-flops).

**To Do**

The designed logic should have minimum number of gates.

• The design should be verified using Verilog with a test bench demonstrating 3 examples of the working state machine. In the first example, the test bench should only input 0’s (quarters) after resetting the state machine. In the second example, the test bench should only input 1’s (dimes) after resetting the state machine. In the third example, the test bench should input a combination of an equal number of 0’s and 1’s. In all examples, the test bench should run the state machine long enough so that a $1 out is produced, change is made, and the state machine begins to count up to 70 cents again.

• The design should be simulated with Verilog and a screenshot of the testbench output should be printed in the report.

• The combinational logic in your sequential circuit should be designed in ‘structural model’, using only 2-input and 3-input gates. The D flip flops in your sequential circuit may be designed in ‘behavioral model’.