



CMPE 212L, Principles of Digital Design Laboratory

Project #2

Due: Fri 5/6/2016

Objective

The objective of this project is to experiment the design and implementation of synchronous sequential circuits.

Project Description

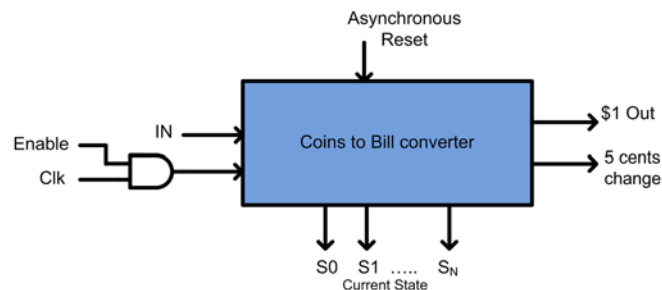
Almost all of the real world electronic systems are designed with a help of one or multiple finite state machines. This project considers a finite state machine controller for a 'coin to bills converter' machine. To help needy children, a generous company has decided to offer \$1 food voucher for every 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:

Input	Coin
0	25 cents
1	10 cents



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 to 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).

The ‘clk’ input should be given from the function generator with the following settings: FUNC = SQUA, FREQ = 0.5Hz, AMPL = 2.5v, OFFSET = 2.5v

To Do

- The designed logic should have minimum number of gates, try to simplify so that it will help in wiring.
- 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 using NCLaunch and the graphical timing diagram 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 JK flip flops in your sequential circuit may be designed in 'behavioral model', and you may use the JK flip flop code given below and also posted on blackboard.
- You can use the IC in your box. Determine the required IC numbers ahead of time.

Project Report

- A project report is due on 5/9/2016. Make sure to explain your state diagram with neat titles, state machine, inputs and outputs, K-Map / Quine-McCluskey reductions, Boolean equations, circuit diagrams and all other details.

Submission

- On 5/6/2016, the design has to be implemented in the breadboard and verified in the lab.
- Bring the state diagram of your design and also a table specifying which state corresponds to what value. Failure to do so will result in losing points.
- Once the implementation is complete, the TA will verify the state machine with random inputs.
- The submission deadline for the report and Verilog files (including testbench) is 5/9/2016. Submit **all files** using blackboard.

Please do NOT email your project submissions to the TA or the instructor, unless you have a problem with for your blackboard account.

Important Notes:

- This has to be an individual work; any act of plagiarism will be penalized and reported.
- You MUST come to the lab on 5/6/2016 with your completed design and schematic.
- On 5/6/2016, there will be no discussion session. The lab will start at 2:00pm and run until 4:00pm. This will allow sufficient time for implementing your design.

```

module JKflipflop(q, _q, J, K, clk, clear, set);
input J, K, clk, clear, set;
output q, _q;
reg q, _q;

//As long as clear and set are not low, then q will wait for the
//negedge of the clock and change according to the values
//at J and K.
//      If J and K are both low, q holds.
//      If J and K are not equal, q takes on the value at J.
//      If J and K are both high, then q toggles.
always@(negedge clk or negedge clear or negedge set)
    begin

        if(~clear) begin
            q <= 1'b0;
            _q <= 1'b1;

        end

        else if(~set) begin
            q <= 1'b0;
            _q <= 1'b0;

        end

        else begin
            case({J,K})
                2'b00: begin
                    q <= q;
                    _q <= _q;
                end

                2'b01: begin
                    q <= 1'b0;
                    _q <= 1'b1;
                end

                2'b10: begin
                    q <= 1'b1;
                    _q <= 1'b0;
                end

                2'b11: begin
                    q <= ~q;
                    _q <= ~_q;
                end

            endcase
        end

    end
endmodule

```