ENGR-354

HW # 16

You have just been hired by a new start-up company who is planning to market good quality face masks from vending machines placed near public places where people may arrive having forgotten to bring a face mask and must have a face mask to enter. While the mechanics of the vending machine have been designed and individual parts tested, a controller is needed to made all the parts work together to dispense masks after proper payment is made. Your new boss has given the job of designing the controller to you.

Figure 1 below shows a block diagram of the electrical part of the vending machine. Subsystems that collect money, dispense product, give change, and control the whole operation are shown. Here are assumptions you can make about the subsystems:

1) All signals going in/out of the subsystems are logic level signals and are electrically compatible.
2) The 25c and 50c (25 cents and 50 cents) signals coming out of the coin receiver will be asserted for one clock cycle when a new coin is inserted.
3) The dispense signal needs to be asserted for one clock cycle (and only one cycle). Each time it is asserted only one product item will be dispensed.
4) The return25 signal needs to be asserted for one clock cycle (and only one cycle) each time a quarter needs to be returned. One quarter (and only one) will be released each time the return25 signal is asserted.
5) You can assume that once a coin is inserted in the coin receiver that additional coins will be inserted until the cost of a mask is achieved (i.e. there is no way to cancel getting a mask once you start putting coins in the coin receiver).
6) The cost of a mask is $1.00

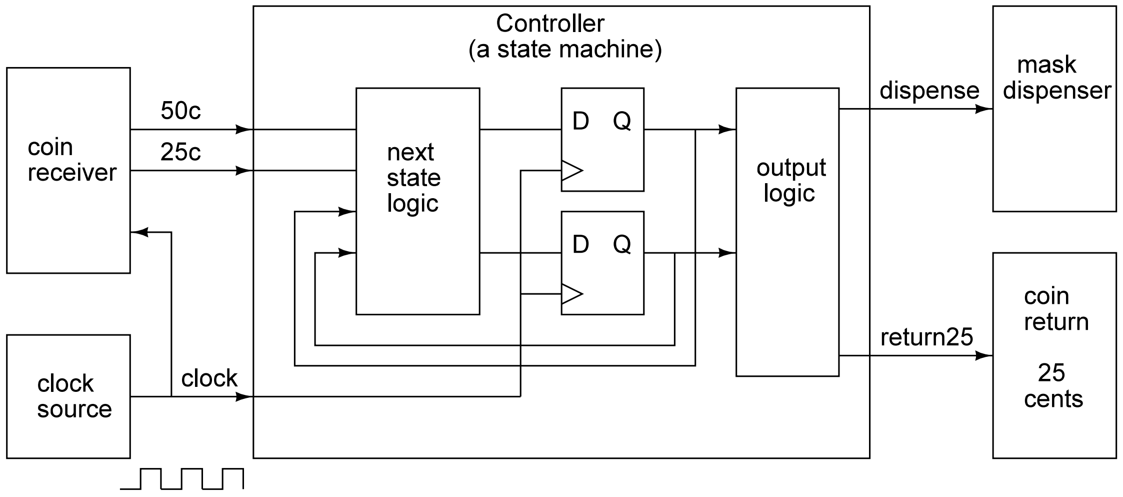

Figure 1 - system block diagram

To Do

Design the controller using a state machine that uses edge triggered D flip-flops as memory.

Follow this procedure for designing a state machine:
1) Determine design specifications (given).
2) Draw a block diagram (provided).
3) Create a fully documented state diagram using as few states as possible.
4) Make binary state assignments (likely you will need three state bits).
5) Create entered variable K-maps, one per state bit, recalling that the axis of the map represent. states and entered variables the input signals.
6) Loop-out and read minimized next state logic from the k-maps.
7) Develop output decoder logic by creating output K-maps and reading minimized logic.
8) Draw a logic diagram using AND and OR gates.
9) Revise the logic diagram using gate types available in our kit of parts.

For this homework you do not need to build the circuit, just produce the documentation defined above.

Keep in mind the following:
1) The sum of all branching conditions leaving each state must equal one.
2) Branching conditions leaving each state must be unique.
3) An output associated with a state will be active for the duration of that state unless it is what we call a Mealy output that depends on an input being active concurrent with a particular state (Mealy outputs are not needed for this homework).
4) The time spent in a state will be one clock cycle minimum. If there is no holding branch on a particular state then that state for sure will last only one clock cycle.
5) For the purposes of this design, unused states are considered Don't Cares.
6) Clock connects only to the clock input on the flip-flops. Clock will not appear as a branching condition on the state diagram or as an input to the next state logic.
7) For this problem, some of the states will likely represent an amount of money that has accumulated as a series of coins are inserted.

To Turn in:

As HW#16a - just your fully documented state diagram.
As HW#16b - the complete set of documentation you have created.