

**Massachusetts Institute of Technology**  
**Department of Electrical Engineering and Computer Science**  
**6.111 – Digital Systems Laboratory**  
**Problem Set #3**

Issued: Wednesday February 20, 2002

Due: Wednesday February 27, 2002 (**IN CLASS**)

**Problem 3.1 Fun with Counters**

Using the minimum amount of logic, configure the LS163 4-bit counter to repeatedly count the following patterns:

- a. from 1 to 6 repeatedly, i.e. 1, 2, 3, 4, 5, 6
- b. from 5 to 7, then 12 to 15, i.e. 5, 6, 7, 12, 13, 14, 15
- c. the sequence 0, 1, 2, 4, 7, 11, 13, 14

**Problem 3.2 Finite String Recognizer**

A finite string recognizer has one input X and one output Z. The output is asserted whenever the input sequence 110 or 101 has been observed, as long as the sequence 00 has never been seen. Design this Finite State Machine with the least number of states possible. (Hint: less than 9 states are required for this state machine)

Here is some sample input and output strings to clarify the FSM specification:

X: 1 1 1 0 1 0 1 1 0 1 1 0 0 1 1 0  
Z: 0 0 0 1 1 0 1 0 1 1 0 1 0 0 0 0

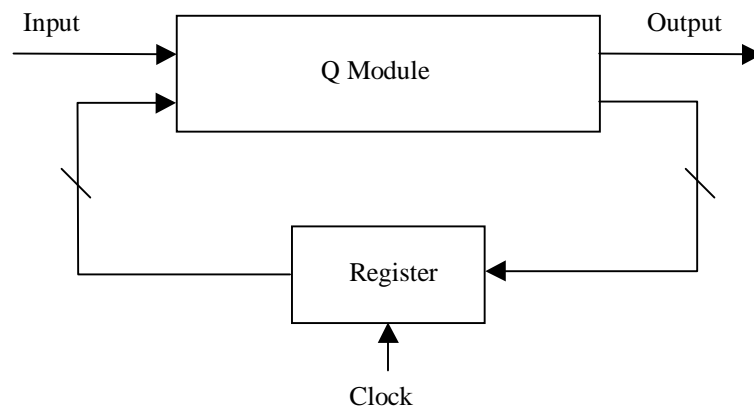
X: 1 0 1 0 1 1 0 1 1 1 1 1 0 0 1 1  
Z: 0 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0

Turn in your state diagram, carefully labeled. Be sure to indicate which state the FSM is after a reset.

### Problem 3.3 Twenty Cent Sodas!

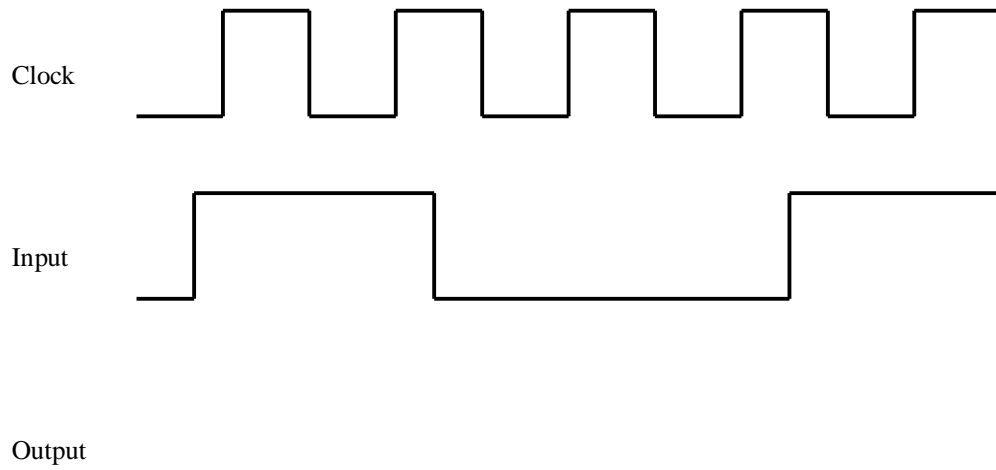
You have been hired by a soda company to produce a controller for their new soda machine. This soda machine is very limited in that it can only take nickels and dimes. Design a system so that it has one input  $X$ , representing the coin slot, and one output  $Z$ , representing the soda. The input  $X=0$  means that a nickel is added to the total, while the input  $X=1$  means that a dime is added to the total. The output  $Z$  is asserted high when a soda comes out of the machine, i.e. twenty or more cents have been deposited. However, any amount over twenty cents is lost, and the next person to purchase soda will have to pay the entire twenty cents.

- Draw a state machine representing the above system. Be sure to specify the starting state.
- The state-transition diagram is implemented in a clocked sequential machine shown below, consisting of a register of positive edge-triggered D flip-flops and a “Q module”, which is purely combinational logic. How many state variables are needed?



- Give the truth table for the Q module such that the sequential machine above implements the desired state-transition diagram. Make sure the initial state corresponds to all state variables being zero.

- d. Reproduce the diagram below and fill in the output for the sequential machine using the Q module and D flip-flops. You may neglect propagation delay in the logic, assuming them to be zero. Assume that there is no money in the machine at the start.



- e. Implement the FSM described above in VHDL.