## MASSACHUSETTS INSTITUTE OF TECHNOLOGY DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

## 6.111 Introductory Digital Systems Laboratory Fall 2008

## Lecture PSet #5 *Due: Thu, 09/23/08*

**Problem 1.** (Katz, problem 8.13) A finite state machine has one input and one output. The output becomes 1 and remains 1 thereafter when at least two 0's and two 1's have occurred as inputs, *regardless of the order of appearance*.

Assuming this is to be implemented as a Moore machine, draw a state transition diagram for the machine. Hint: You can do this in nine states.

**Problem 2**. FSMs are often used to generate sequences of waveforms necessary to communicate with a component. Older memory chips (DRAMs) required several control signals to be asserted in a particular sequence to perform a memory read. The following figure shows a control module and the waveforms it must generate in response to a read request:



The module sits idle with RAS=1, MUX=0 and CAS=1 until it detects REQ=1 on the rising edge of CLK. In the first cycle of the request it should assert RAS=0, in the second cycle MUX=1 and in the third cycle CAS=0. These signals are held during the fourth cycle and return to their idle values in the fifth cycle. The module then waits for a new request; it ignores requests made while in the middle of processing the last request.

- (A) Draw a state transition diagram for a FSM that will generate the appropriate sequence of signals.
- (B) Write the Verilog for the module, choosing a state encoding and providing the appropriate combinational logic for generating the next state and output signals.