HW #5

Due Date: February 28th

Note: submit both the code and the simulation output for each problem requiring simulation.

1. Describe the operation of the logic block which is coded below and give a function table for it.

```
entity mystery is
port(  D: in  std_logic_vector(7 downto 0);
        Q: buffer std_logic_vector(7 downto 0);
       EN: in  std_logic;
      CLK: in std_logic);
end mystery;

architecture synthesis of mystery is
 signal M_out: std_logic_vector(7 downto 0);
 begin

   process
   begin
     wait until clk='1';
       Q <= M_out;
   end process;

   M_out <= D when EN = '1' else Q;

end synthesis;
```

2. Write a behavioral VHDL model for the following Moore state diagram (see figure 1). Your VHDL code will implement the state transitions for the states described in the diagram below. Your code will have two processes. The first will be a synchronous process using a case statement to generate the next state based on the input and the current state. The second will be a synchronous process that will implement the reset or setting the state to the next state. Let the input be X, output Z, and use positive edge-triggered flip-flops for the state register. Simulate in Quartus or Modelsim, being sure to force the circuit to traverse all states.

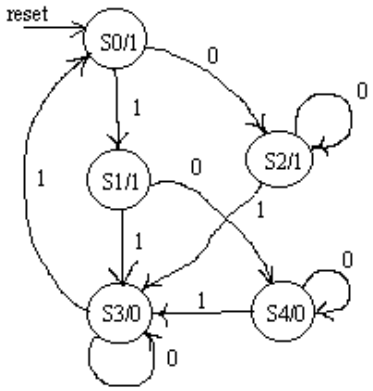

Figure 1. Moore FSM for Q2

3. Complete question 6.11, pg. 154, from the textbook. The code for this problem is in the appendix so **Be sure that you fully understand the code!** Use a **testbench** simulation to verify that the design functions correctly.

5. Design a 4 bit counter which will output the sequence 4, 9, 12, 3, 7 repeatedly. Verify your design in Quartus by doing the following: construct an LPM_COUNTER megafunction for 4 bits and use a block diagram with the required logic to give you the correct outputs and trigger a repeat of the sequence.

6. Design a datapath that can execute the two statements A=B+C and A=A+3 using just one adder. (Note that A, B, C are all 8 bits.)

7. Construct a 4 bit wide dedicated datapath to generate and output the numbers from 1 to 10. Indicate any inputs from an external source and a controller as well as any status signals generated in the datapath but there is no need to design the controller.

8. Design a dedicated datapath for inputting two 8-bit unsigned numbers (each presented one clock cycle after the other at the input port) and then output the larger number. The datapath should have only one input port and one output port. Label clearly all of the control and status signals (You can assume that the "greater than" comparison block is given to you. The LPM_COMPARE megafunction in Quartus could be used to generate this output.)

Winter 2013