CS220: Practice Problems for Lab

1. Implement the finite state machine the next state function of which is shown in Table 1. The machine takes a two-bit input (Y[1:0]). The output function is not specified and can be ignored for this question. This is the same question from the mid-semester exam. Suppose a state incrementer can be used to compute the next state along

Table 1. Next state function

Current state	Input (Y[1:0])	Next state
S_0	2'bxx	S_1
S_1	2'bxx	S_2
S_2	2'bxx	S_3
S_3	2'b00	S_4
S_3	2'b01	S_5
S_3	2'b1x	S_6
S_4	2'bxx	S_7
S_5	2'bxx	S_7
S_6	2'bxx	S_7
S_7	2'bxx	S_8
S_8	2'bxx	S_9
S_9	2'bxx	S_{10}
S_{10}	2'b00	S_{11}
S_{10}	2'b01, 2'b1x	S_{12}
S_{11}	2'bxx	S_0
S_{12}	2'bxx	S_0

with two dispatch ROMs, a microcode ROM, and a state selection multiplexer. The microcode ROM takes only the current state as input to look up a row and outputs the branch control for computing the next state (note that Y cannot be used as an input to the microcode ROM). The two dispatch ROMs are used to encode the two branches in the state diagram. The first branch occurs when the current state is S_3 and the second branch occurs when the current state is S_{10}. The first dispatch ROM stores the next states when the current state is S_3. The second dispatch ROM stores the next states when the current state is S_{10}. The input Y is used to look up a row in the dispatch ROMs. I have included the solution in the following for your convenience.

Solution: Let us denote the branch controls for incremented state as 0, for S_3 dispatch ROM as 1, for S_{10} dispatch ROM as 2, for converging back from S_4 and S_5 to S_7 as 3, and for converging back from S_{11} and S_{12} to S_0 as 4. The contents of the ROMs are shown below. The microcode ROM is indexed by the current state and outputs the branch control. The dispatch ROMs are indexed by Y[1:0] and output the next state. The multiplexer's selection lines are tied to the output lines of the microcode ROM. Since the microcode ROM generates three-bit output, there are three selection lines. However, the selection lines will attain only five possible values corresponding to the branch control output of the microcode ROM: 0, 1, 2, 3, 4. The corresponding five inputs to the multiplexer are respectively current state+1, S_3 dispatch ROM output, S_{10} dispatch ROM output, 0111, and 0000. Since the multiplexer selects the next state, each of the five inputs to the multiplexer is four-bit wide.

Table 2. Microcode ROM

0
0
0
1
3
3
0 or 3
0
0
0
2
4
4

Table 3. Dispatch ROM for next state of S_3

0100
0101
0110
0110

Table 4. Dispatch ROM for next state of S_{10}

1011
1100
1100
1100

The ROMs should be implemented as arrays and initialized within an initial block. The current state should be displayed using four LEDs. The state change must take place every time a new input is entered (use any means that you know of to accept the two-bit input Y). Additionally, after a state change, if a new input is not entered for two seconds, the state will change based on the last input. You can use the standard 50 MHz clock of the FPGA to count two seconds.

2. Construct a hardware that takes as input four three-bit values treated as unsigned numbers and computes the index of the smallest value. For example, if the inputs are 110, 010, 001, and 111, the output is 2 indicating that the input at position two is the smallest (input index starts at zero and ends at three). Use any means that you know of to accept the inputs. Display the inputs in decimal in the first line of the LCD with a comma and a space between two consecutive inputs. For the aforementioned four inputs, the first line of the display should be 6, 2, 1, 7. Display the output in decimal in the second line of the LCD.

3. Consider a hardware that takes a four-element bitvector X[0:3] as input. It sorts the elements and stores the sorted result in another vector Y[0:3]. While doing the sorting, the hardware also remembers the original index of an element in X. This is stored in another array Z where each element is two-bit long. While sorting, a tie between two elements is broken by putting the left element first and then the other element in Y. As an example, suppose X[0:3] = 4'b1010. Then Y[0:3] = 4'b0011 and Z[0] = index of X[0] in Y = 2'b10, Z[1] = index of X[1] in Y = 2'b00, Z[2] = index of X[2] in Y = 2'b11, Z[3] = index of X[3] in Y = 2'b01. Take the inputs through the switches. Display the input bitvector and Y in the first line of the LCD with sufficient space between the two. Display each element of Z in decimal in the second line of the LCD with a comma and a space between two consecutive elements of Z. For the aforementioned example, the first line would be 1010   0011 and the second line would be 2, 0, 3, 1.