cs281: Computer Organization

Lab7

1 Moore Machine Sequence Detector

In a *Moore* circuit, the outputs depend only on the present state. A general model of a Moore sequential circuit consists of two combinational circuits – one to compute the next state and associated flip flop input values from the present state and the inputs, and a second circuit to compute the output based on the present state. These two combinational circuits are combined with a set of 1-bit memories (called the "state register"), which encode and remember the present state. The state register normally consists of D or JK flip-flops.

A *sequence detector* is a circuit that serially examines a string of 0's and 1's applied to the X input and can generate an output Z when the sequence matches a particular pattern. For this lab, the pattern you are looking for is the subsequence 111. For this specification, the circuit does not reset when the output is matched. A typical input sequence and the corresponding output sequence might be:

X	=	0	0	1	1	0	1	1	1	1	1	0	1	1	1	0	0
Z	=	0	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0

1. Design a Moore-style finite state machine that describes the operation of this circuit. Identify how many 1-bit memory elements are required to represent your set of states.
2. Draw a truth table that describes the next state function of the finite state machine.
3. Draw a truth table that describes the output function of the finite state machine.
4. Use K-maps to simplify the boolean expressions for each of your outputs.
5. Design a sequential circuit from your boolean expressions. Use D flip-flops.
6. Implement your circuit in Logisim. Make the combinational parts of the circuit into *separate* circuits (called NextState and Output), but keep the D flip-flops on the Logisim *main* circuit. Demonstrate for your instructor.
7. If there is time, re-implement your circuit on a bread board. The D flip-flop IC number is 7474. Demonstrate your completed circuit for your instructor