

DigSim Assignment 3: Finite State Machine Simplifications

Due: Tuesday December 10, 2002

Objective

The objective of this assignment is to design and simplify a moderately complex a finite state machine.

Assignment

For this project you will design and implement in DigSim a Moore-type finite state machine with one input bit and one output bit. The machine's output depends on the previous two input bits as well as the machine's previous output and is specified as follows. If the previous two input bits were 00, then the output is 0. If inputs were 01, then the machine flips its output bit (0 becomes 1 and 1 becomes 0). If the inputs were 10, then the output bit remains the same. Finally, if the previous two input bits were 11, then the output is 1. You may assume that the initial state of the machine corresponds to the state of the machine when it has seen two 0's as input. For example, if the machine is given inputs 0011011000, the outputs would be 0011101100.

In order to complete the project, you must accomplish the following tasks. Note that for this project, you will submit both your design work (on paper) and the DigSim implementation of the resulting finite state machine (electronically).

1. Draw a transition diagram for the finite state machine described above. Recall that in a Moore-type finite state machine, the output bit depends directly only on the state of the machine and not the input bits. Thus, the output of the machine should be specified at each state, rather than each transition. Reduce the number of states if necessary. You should be able to design a machine with 6 states.
2. Assign bit patterns to each state of the machine. Use the heuristics for state assignment presented in class (also available on the course lecture topics web page). Do this step carefully as it will have a large effect on the complexity of your final circuit. If you succeeded in designing a machine with 6 states, then you should only need 3 bits to specify the states.
3. Fill in the state transition table for the finite state machine given below. In this table, A, B and C are the bit patterns assigned to the states, x is the input bit and z is the output bit. Then A', B' and C' are the next states to be stored in the flip flops.
4. Use the Karnaugh maps provided to simplify the Boolean formulas for a finite state machine to be implemented using D flip-flops. You will need Karnaugh maps for A', B', C' and z.
5. Using the excitation table for J-K flip-flops, fill in the columns JA, KA, JB, KB, JC and KC in the truth table below. The columns JA and KA represent the settings to the inputs of the first J-K flip-flop that will cause it to store the value A'. The columns JB, KB, JC and KC are analogous for the second and third J-K flip-flops.
6. Use the Karnaugh maps provided to simplify the Boolean formulas for the J and K inputs to each J-K flip flop. You will need Karnaugh maps for JA, KA, JB, KB, JC and KC. The formula for the output z will be the same as in Step 4.
7. Decide whether you want to use D flip-flops or J-K flip-flops. Briefly justify your choice.
8. Implement the resulting circuit in DigSim.

What to submit

In class on Tuesday December 10, turn in your finite state diagram, truth table, Karnaugh maps and the resulting Boolean formulas on paper. Make copies of these, if you still need them to implement your circuit in DigSim.

Save your circuit as you did in DigSim Assignment 1. Submit the circuit file using the Unix submit command as in previous assignments. The submission name for this assignment is: `digsim3`.

Excitation Table for J-K Flip-Flops

Q	Q'	J	K
0	0	0	<i>d</i>
0	1	1	<i>d</i>
1	0	<i>d</i>	1
1	1	<i>d</i>	0

Truth Table:

	A	B	C	x	A'	B'	C'	z	JA	KA	JB	KB	JC	KC
0	0	0	0	0										
1	0	0	0	1										
2	0	0	1	0										
3	0	0	1	1										
4	0	1	0	0										
5	0	1	0	1										
6	0	1	1	0										
7	0	1	1	1										
8	1	0	0	0										
9	1	0	0	1										
10	1	0	1	0										
11	1	0	1	1										
12	1	1	0	0										
13	1	1	0	1										
14	1	1	1	0										
15	1	1	1	1										

Question:

Should you use D flip-flops or J-K flip-flops to implement this circuit? Why?



