Computer Organization and Assembly Language Programming (CMSC 313)
Section 0102 Fall 2003

Digsim #2

Objective

In this assignment, you are to design two arbitrary code counters that have the same functionality, only using different flip flops and combinational logic for each design. Part of the submission will be made using facilities afforded by “Digsim”.

Problem Description

A synchronous counter is to be designed which follows one of two repeating sequences, depending on a control input C. When C = 0, the sequence is ‘00’, ‘01’, ‘11’. When C = 1 the sequence is ‘00’, ‘11’, ‘01’.

Implementation Notes

You should also use Gray Code sequencing to represent the states during the state assignment phase of design. Use the design procedure outlined in the lecture notes. It will not be necessary to reduce the number of states in this design. You should have no more than 4 states to get this FSM to work. Use K-Maps to aid in combinational logic reduction. Be aware that you must include the condition where the count happens to come up as ‘10’ in your state transition diagram. You can use the FF’s and 2 or 3 input AND / OR gates and inverters as provided in digsim to construct the combinational logic unit section of the FSM. In one design use positive edge-triggered D-type FF’s, in a second design, use JK FF’s to store the machine’s state bits.

You are to implement your FSM in digsim. Note that the FF’s are positive edge triggered. To change the state of the FF change the input when the clock is low, then bring the clock from low to high; At this instant the D input will be stored in the FF. You can also use the oscillator running at slow speed with LED indicators or the waveform generator to help see the FSM at work.

Submission Requirements for each design

Indicate the number of FF’s required to store the state bits. Draw the state transition diagram and a state transition table. Draw up a modified state transition table with your choice of state assignments. Draw a truth table and develop K-Maps for the FF input functions and the outputs. Submit this hardcopy, in the correct design sequence, during the class timetabled for Dec 8.

Draw up, test and debug your circuits in digsim. Take care to lay the circuit out with the combinational logic unit compactly presented and separate from the FF section as shown in the worked examples depicted in the course textbook. Give yourself sufficient time to complete the circuit layout, test and debug phases. Note that to reinitialize the FSM, you need to save the circuit and load it back into digsim.

Save your two circuits in two separate files using the approach adopted for the digsim 1 homework. Submit each circuit file using the UNIX submit command as in the previous assignment. The Unix command script should look similar to this:

```
submit cs313_0102 digsim2 fsm.sim
```

Notes: (1) Approved textbook: “Principles of Computer Architecture” M J Murdocca & V P Heuring Prentice Hall 2000 ISBN 0-201-43664-7