University of California at Berkeley
College of Engineering
Department of Electrical Engineering and Computer Science

EECS150, Spring 2010

Homework Assignment 11: FSMs, Synchronizers, Counters, and Parallel Prefix
Due April 16^{th}, 2pm

Homework submission will only be through SVN. Email submissions will not be accepted! Please format your homework as plain text with either PNG or PDF for any necessary figures. Microsoft Visio is installed on the machines in 125 Cory, and is a useful tool for drawing figures of all kinds.

1. DDCA 3.26
2. DDCA 3.35
3. DDCA 3.37
4. A parallel prefix AND-tree was used in class as a means to speed the simultaneous generation of the following functions over the 8 inputs **`A`** through **`H`**: **`A`**, **`AB`**, **`ABC`**, **`ABCD`**, **`ABCDE`**, **`ABCDEF`**, **`ABCDEFG`**, and **`ABCDEFGH`**. An alternative approach would use a separate balanced binary AND-tree for each of the 8 functions. Compare these two approaches in terms of total number of AND gates and worst-case delay (in terms of unit gate delays).
5. Imagine you are given a specification for a circuit that has three inputs, **`clk`**, **`reset`**, and **`input`** and an output, **`output`**. Your job is to design an FSM that detects the sequences 01 and 10. The FSM continuously inspects its input and generates a 1 at is output when the input sequence 01 or 10 has been detected, otherwise it outputs a 0. Sequences can overlap. For instance, the input sequence 0101 will output a pulse for three consecutive cycles. Show the finite state machine diagrams and the state transition tables for both the Moore *and* Mealy implementations of this circuit. Then, for each implementation, show the gate level diagrams of the FSM using both binary encoding and one-hot encoding.
6. Now, suppose you want to add to your circuit from the last problem an output, **`ten`**, that goes high iff, since the last reset, **`output`** has been high for a number of cycles equal to a multiple of ten. In addition you want to have a 4-bit output, **`remaining`** that shows how many more sequences must be detected before **`ten`** is high again. Pick any one of your circuits from the last problem to use to implement your solution to this problem.
7. Review the state transition table on Page 21 of Lecture 21, and find simplified boolean expressions for **`NS2`**, **`NS1`**, and **`NS0`**.
8. DDCA 3.27
9. Repeat DDCA 3.27 using one-hot encoded states