CS 150
Lab Lecture 2

Schematic Entry & Simulation, Part 010_2
Finite State Machines
Laura Todd, 9-8-2000

Homework Quiz

- NMP - Not My Problem. ;P

Overview

- Homework Quiz ✓
- Administrivia
- Objectives
- Prelab
- Brief Description
- To-do
- Hints, tips, and tricks

Administrivia

- Discussion Sections:
  - Monday 1-2 285 Cory
  - Tuesday 1-2 310 Soda
  - Wednesday 1-2 71 Evans
  - Wednesday 2-3 4 Evans
  - Thursday 1-2 310 Soda
- Office Hours (TAs)
  - Laura Wed. 3-4 204b Cory
  - Jimmy Tu. 4-5 204b Cory
  - Norm Th. 3:30-5 204b Cory
  - Po Th. 1-2 204b Cory
  - Ramki F. 9:30-11:30 tentatively 464 Soda

Objectives

- Practice using Xilinx software
- Enter a FSM design from logic equations
- Simulate the FSM
- Work in a group

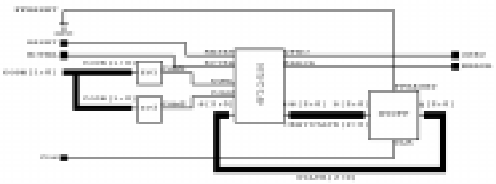
Prelab

- **Get a partner in YOUR lab section**
  - **For the first few labs, your partner must be from your own lab section**
- Complete the **IN1** and **IN2** blocks
- Write a script file to test your design
- Why do we have a prelab assignment?
  - Because there's lots to do, maybe more than you can do in 3 hours
  - Much of the design work can be done on paper beforehand

Brief Description – the "High Level Spec"

- The high-level spec is a description of the problem in "plain English"
  - This is a very general description of the system you're trying to build, in our case, a combination lock.
    - Our lock will have a RESET button, an ENTER button, and two binary switches: CODE1 and CODE0.
    - To open the lock, the user must enter a pair of 2-bit codes. (ie. 01 <enter> 11 <enter>)
    - If the user enters the right codes, the lock will OPEN, otherwise it will indicate an ERROR
  - Designing from a high-level specs is like those old math word problems: you have to come up with a way to get the answer all on your own. But don't despair, we do it for you this time!

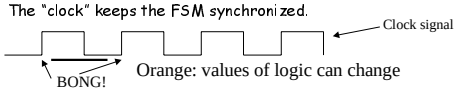
Brief Description – Lower Levels: Down the rabbit hole…



System-level view: these are the building blocks we need

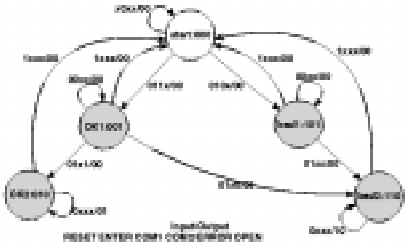
Brief Description – the Clock

- The "clock" keeps the FSM synchronized.



- When the clock goes "BONG!" the FSM will change state
- During the part of the clock marked with orange, the logic has time to figure out what it's next state is going to be
- When the clock signal value goes from "0" to "1", we call that a rising edge. The time from one rising edge to another is called the "clock cycle" time.
- Let's look at the FSM for the lab→

Brief Description – Lower Levels: Round and round she goes…



Logical view: the finite state machine

Brief Description – the IN* blocks

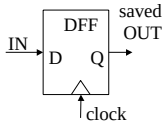
- The IN1 and IN2 blocks compare the code the user inputs to the "secret" code that will open the lock.
  - If code matches, assert a TRUE on the output (COM*)
  - Otherwise, assert a FALSE.
- Choose your own combination. Write it down.
- The two combinations must be different:
  - Good: 01 11
  - Bad: 10 10
- IN1 and IN2 are very simple blocks. Just some AND gates and inverters.

Brief Description – MyCLB

- CLB = Combinational Logic Block => and, or, inv…
- The MyCLB block acts as the controller for our lock, it's the brains
- Truth table and logic equations are in the printout
  - You're expected to implement this block during lab time.
- This block is a little messy:
  - So, keep your schematic neat (it'll be easier to debug)
  - Watch for mis-labeled wires
- KISS-Keep It Simple, Stupid
  - Use the "multi-level" equations, they're much easier to debug
  - But take the time to understand them!

Brief Description – MyDFF

- What the heck are D flip-flops????
  - Remember sequential logic? We need a way of saving the current state!
  - Welcome to the world of memory!
  - D flip-flops are memory elements
  - Put what you want to save on the D pin
  - When the clock has a rising edge the DFF will look at the value on D and move it to Q
  - Now until the next rising edge on the clock, Q will remain the same, giving us a saved output



- This is the simple version: lots more later in class!

ToDo's – What you need for credit

- Enter the IN1, IN2, MyCLB schematics
- Make a test script for your CLB. Run it. Make sure it's correct. (You want this to work, don't you?)
- Make the MyDFF block.
- Wire up all the individual pieces you just made.
- Write a script to simulate scenarios on the full lock
  - Successful/Unsuccessful entries of the combination
  - Use of the RESET and ENTER buttons
  - Do all the paths on the state diagram work correctly?
  - Create a log file to show your TA

Hints, Tips, and Tricks

- Buses are your friends.
  - CS150 buses: collections of wires that have a similar purpose
  - Usually same wire name + a number, as in DATA7, DATA6, … DATA1, DATA0
  - Simplify, simplify, simplify…
- Scripts are your friends.
  - Note: the pipe character " | " will comment out a line.
- Don't print things!
- Xilinx software is buggy. When in doubt, restart it.

The End… for now.

Ya ain't seen nuthin' yet…