EECS150 Lab Lecture 0

Kris Pister, Vincent Lee, Ian Juch, Albert Magyar
Electrical Engineering and Computer Sciences
University of California, Berkeley
What is lab lecture?

• Additional lecture to make your life easier in lab...
• Overview of next week’s lab and prelab work
• Tips to avoid pitfalls and problems during the next lab
• Will be held the Friday before a lab week
• Time for you to ask questions about a lab
• Lab logistics and administrative announcements
• Brief overview of prelab or demonstration
Why should I care about labs?

- Labs and project are core part of this course
- Digital design is not just design and coding, it’s also about figuring out the tools
- Labs will build up to final project – some modules used in final project
- Industry likes people who can do design AND code in verilog
- Can’t be good at verilog and design unless you practice
Lab Resources

• Same as class resources
• Website: inst.eecs.berkeley.edu/~cs150/fa12
• Newsgroup: Piazza
• 1 x Professor Pister
• 3 x TAs
• ~60 x your fellow peers
Lab Sections

Lab Lecture

• Fri 2:00-3:00PM  306 Soda

Lab Sections

• Tues 5:00-8:00PM  125 Cory  Section 011
• Tues 5:30-8:30PM  125 Cory  Section 011   Vincent and Ian
• Wed 9:00-12:00PM  125 Cory  Section 012
• Thurs 5:00-8:00PM  125 Cory  Section 012   Vincent
• Wed 5:00-8:00PM  125 Cory  Section 013   Ian and Albert

• You must attend at least one lab section
• If you’re in Section 012, FILL OUT THE SURVEY
Floating Lab Section

• Proposed times are:
  – Monday 2-5 PM
  – Monday 4-7 PM
  – Monday 5-8 PM
  – Thursday 5-8 PM

• If you are in the Section 012 and can attend Section 011 or 013, please change to that section until it is full – there will be two TAs

• Section 012 will most likely be one TA depending on schedule

• Sections will strictly be capped at 30 students
Lab Logistics

• Labs start next week
• Lab instructions found on the inst website
• Labs due one week from the week they are assigned and due by the end of the last section
• Check offs will only be done in lab or OH
• All labs done individually for the first few labs
• **No partial** check offs
• Late check offs will incur penalty but still given credit
• Finish all labs – important for final project
• No dropped labs
More Lab Logistics...

• Make sure to pick up ONE account form in lab
• Project and lab questions go on Piazza
• Posting source code on Piazza violates academic dishonesty
• Grades entered to bspace – check frequently
• All enrolled students should have lab cardkey access
  – Verify yours works or notify TA immediately
Lab Policies

• Standard lab expectations apply
  – Do the pre-lab and come prepared
  – Neglecting the pre-lab makes it harder/impossible to finish during the lab section
  – Read lab manual before section
  – The trajectory of all food and/or drink shall not be towards lab equipment
# Lab Schedule Overview

<table>
<thead>
<tr>
<th>Lab</th>
<th>Date</th>
<th>Checkoff Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lab 0: Structural Verilog</td>
<td>8/28, 8/29</td>
<td>9/4, 9/5</td>
</tr>
<tr>
<td>Lab 1: Behavior Synthesis</td>
<td>9/4, 9/5</td>
<td>9/11, 9/12</td>
</tr>
<tr>
<td>Lab 2: ALU Design and Verification</td>
<td>9/11, 9/12</td>
<td>9/18, 9/19</td>
</tr>
<tr>
<td>Lab 3: List Processor and Chipscope</td>
<td>9/18, 9/19</td>
<td>9/25, 9/26</td>
</tr>
<tr>
<td>Lab 4: Serial I/O</td>
<td>9/25, 9/26</td>
<td>10/9, 10/10</td>
</tr>
</tbody>
</table>
Project Logistics

• Project will begin around the 6th week of instruction

• Project group sizes are <= 2

• Pick your partner carefully when the time comes

• Partner does not have to be enrolled in same lab

• Additional details announced closer to project
## Project Overview

<table>
<thead>
<tr>
<th>Project Checkpoint</th>
<th>Specification Out</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelined Processor Design Review</td>
<td>TBA</td>
<td>10/10</td>
</tr>
<tr>
<td>MIPS Processor Implementation</td>
<td>TBA</td>
<td>10/19</td>
</tr>
<tr>
<td>Processor Interrupt Implementation</td>
<td>TBA</td>
<td>11/2</td>
</tr>
<tr>
<td>Frame Buffers and Graphics Acceleration</td>
<td>TBA</td>
<td>11/16</td>
</tr>
<tr>
<td>Audio</td>
<td>TBA</td>
<td>11/30</td>
</tr>
<tr>
<td>Final Project Report</td>
<td>TBA</td>
<td>12/8 @ 11:59PM</td>
</tr>
<tr>
<td>Extra Credit: BIOS Implementation</td>
<td>TBA</td>
<td>12/8 @ 5PM</td>
</tr>
<tr>
<td>Final Project Demonstration</td>
<td>N/A</td>
<td>12/8 4PM-5PM</td>
</tr>
</tbody>
</table>

*Schedule subject to change*
Academic Dishonesty and Copying Code

• There’s a fine line between collaborating and cheating
• Don’t copy code. We will catch you and the wrath of the gods will descend upon you and your partners in crime.
• All incidents will be dealt with appropriately and referred to the professor
• The point of doing labs and writing code is to prepare you for the projects and industry
• Project checkpoints build on each other so you’ll have to understand your code anyways
FPGA Development Platform

- Xilinx Virtex 5 xc5vlx110t
- Do not break – makes it harder to program
- Do not touch heat sink – it gets hot...
- FPGA is pointy - if you get into a fight, the FPGA will always win
- Intentionally damaging the lab equipment will affect enrollment
This Week’s Lab

• Objectives:
  – Cover basic structural verilog constructs
  – Get familiar with the development platform

• Pre-lab requirements:
  – Read Chapter 5: Configurable Logic Blocks of the Virtex-5 User Guide and answer the pre-lab questions
  – Write these answers down somewhere and bring them to lab
This Week’s Lab

• **Structural verilog**
  – Simple primitive gates – AND, OR, XOR, NAND, etc.
  – Exact wiring configurations and interconnections
  – Tells the synthesis tools exactly what you want

• **Behavioral verilog**
  – Covered next week
  – Tools infer equivalent logic
  – Specify “behavior” of circuit

```verilog
Decoder(output x0,x1,x2,x3;
    input a,b)
{
    wire abar, bbar;
    inv(bbar, b);
    inv(abar, a);
    and(x0, abar, bbar);
    and(x1, abar, b   );
    and(x2, a, bbar   );
    and(x3, a, b      );
}
```

Structural Verilog Code
This Week’s Lab

• This week only, we will indicate where to put your code in each of the files

• Indicated by /***/YOUR CODE HERE*****/ and /***/END CODE****/ in code base

• Do not modify any code outside these indicators... things will probably break

• Delete specified code – it’s in the documentation
This Week’s Lab

• Files to modify:
  – /lab0/src/FA.v
  – /lab0/src/Mux2_1.v
  – /lab0/src/Adder.v
  – /lab0/src/ml505top.v

• Check off requirements:
  – Working full adder
  – Working 2-1 mux
  – Working ripple adder
  – Answers to pre-lab
  – Due by Wednesday 8/29 @ 8PM
Project File Structure

Makefile

- Configuration settings to build project
- Don’t modify. All hell will break loose...

Synthesis and Build commands
- Run “make” command to build and synthesize
- Run “make impact” to program board
- Run “make clean” to clean project synthesis

src Directory
- Contains all verilog source code
- ml505top.v file – top level module

build Directory
- Only available after synthesis
- Contains results of synthesis

cfg Directory
- Contains configuration information about synthesis
Lab Problems

• If you have trouble running the tools or synthesis, ask one of us

• Try to debug your code first before asking for help...
  – Debugging is an invaluable skill for the final project
  – TAs will not always be around

• Shooting for two TAs per lab section

• Office hours will be held in the lab
  – TBA
  – Bring lab questions to office hours
Next Week’s Lab

• Next week’s lab lecture: FPGA Editor and Development Flow (Ian Juch)

• Next week’s lab: Behavioral Synthesis
Questions, comments, or concerns?
Acknowledgements

• Some points and content for slides taken from presentations by:
  – Chris Fletcher & John Wawrzynek (2/2010)
  – Chen Sun (2008-2009)
  – Sarah Swisher (2008)