Honors Section

About This Class

The Honor's class is self paced. We have provided all of the materials you need to finish assignments as quickly as you want, and we will grade your assignments as they come in but have provided deadlines to help you pace yourself so that you don't have a mountain of work at the end of the semester. You are expected to go to Honors office hours for questions and lab credit.

We cover CMOS, ECC, Karnaugh maps, FPGA boards, and some advanced MIPS concepts. The FPGA section of the course consists of four labs and a final project. The MIPS section consists of a written homework and a take-home lab. You are expected to work at your own pace, keeping up with the deadlines for each topic.

For the FPGA labs, you must attend at least one office hour before you can checkout a board and work on your own. You will need to show your finished results for each lab to an Honors instructor before the deadline.

Assignments that are through git (FPGA, MIPS++) will be pushed soon for students in the class.

Schedule

<table>
<thead>
<tr>
<th>Finish by</th>
<th>Topic</th>
<th>Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sept 27</td>
<td>Complementary Metal-Oxide Semiconductor (CMOS) circuits</td>
<td>Watch the following video lectures</td>
</tr>
<tr>
<td></td>
<td>Error-Correcting Codes (ECC)</td>
<td>CMOS lecture: <a href="https://youtu.be/1tZyGL1K5Qi">https://youtu.be/1tZyGL1K5Qi</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Error-Correcting Codes (Hamming Distance and Parity): <a href="https://youtu.be/RXzqXzDapZs">https://youtu.be/RXzqXzDapZs</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(7,4) Hamming Code: <a href="https://youtu.be/2BI7wvmdFE8">https://youtu.be/2BI7wvmdFE8</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Homework</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Homework 101 and 102 on PrairieLearn</td>
</tr>
<tr>
<td>Sept 27</td>
<td>FPGA Intro</td>
<td>FPGA Intro Video</td>
</tr>
<tr>
<td></td>
<td>FPGA Lab 0</td>
<td>FPGA Intro Slides</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPGA Lab 0 Handout</td>
</tr>
<tr>
<td></td>
<td></td>
<td>FPGA Lab 0 Tutorial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vivado Intro Slides</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vivado Tutorial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Basys3 Reference Manual</td>
</tr>
<tr>
<td>Oct 4</td>
<td>Karnaugh Maps</td>
<td>Watch the following video lectures</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Karnaugh Maps: <a href="https://youtu.be/CpsJoAwreqo">https://youtu.be/CpsJoAwreqo</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Don't Cares: <a href="https://youtu.be/95xLk7gHgIE">https://youtu.be/95xLk7gHgIE</a></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Homework</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Complete the worksheet and submit into the homework dropbox in the Siebel basement (near the vending machines). Print-outs of the worksheet are available just outside the CS 233 TA office (0212 Siebel)</td>
</tr>
<tr>
<td>Oct 4</td>
<td>FPGA Lab 1</td>
<td>FPGA Lab 1 Handout</td>
</tr>
</tbody>
</table>
Oct 11 | FPGA Lab 2 | FPGA Lab 2 Handout
Oct 18 | FPGA Lab 3 | FPGA Lab 3 Handout
Oct 25 | MIPS++ Part 1 | Watch the following video lectures
| | | https://youtu.be/m0LUcSkwyAM
| | | Homework
| | | Complete the worksheet. Please scan or take pictures of your completed handout and submit using a PRIVATE piazza post entitled “MIPS++ handout submission”. Be sure to include your netid so we can give you points.

Nov 1 | MIPS++ Part 2 | Complete the Lab assignment (distribution of lab files coming soon)
Nov 1 | Final Project Idea Proposal | FPGA Final Project Handout
Nov 29 | Final Project Check-in | Schedule a meeting with your assigned Honors instructor on or before this day to demo your current progress.
Week of Dec 10 | Final Project Demo | Exact schedule details coming later. Everyone will demo their finished projects to the class on this day.

**Honors Office Hours**

See calendar on front page

**Grade Breakdown**

<table>
<thead>
<tr>
<th>Course</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECC Homework</td>
<td>2.5%</td>
</tr>
<tr>
<td>CMOS Homework</td>
<td>2.5%</td>
</tr>
<tr>
<td>Kmap Homework</td>
<td>5%</td>
</tr>
<tr>
<td>Lab 0</td>
<td>10%</td>
</tr>
<tr>
<td>Lab 1</td>
<td>10%</td>
</tr>
<tr>
<td>Lab 2</td>
<td>10%</td>
</tr>
<tr>
<td>Lab 3</td>
<td>10%</td>
</tr>
<tr>
<td>MIPS++ part 1</td>
<td>10%</td>
</tr>
<tr>
<td>MIPS++ part 2</td>
<td>10%</td>
</tr>
<tr>
<td>Final Project</td>
<td>30%</td>
</tr>
</tbody>
</table>

**Example FPGA Final Project Ideas**

<table>
<thead>
<tr>
<th>Project Title</th>
<th>Description</th>
</tr>
</thead>
</table>


<table>
<thead>
<tr>
<th><strong>Calculator</strong></th>
<th>Example: <a href="https://www.theonlinecalculator.com/">https://www.theonlinecalculator.com/</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Build a multi-function calculator (must at least perform addition, subtraction, multiplication, and division as well as a small stack-based memory that can remember the outcome of prior computations that the user selects)</td>
<td></td>
</tr>
<tr>
<td>The calculator should use the keypad or keyboard peripheral to allow a user to enter numbers and operators (Your calculator must NOT rely on the switches other than for an overall reset)</td>
<td></td>
</tr>
<tr>
<td>Consider displaying hex digits instead of decimal for this calculator.</td>
<td></td>
</tr>
<tr>
<td>Must use a FSM to respond to sequences of button presses.</td>
<td></td>
</tr>
<tr>
<td>Use a keyboard to enter characters</td>
<td></td>
</tr>
<tr>
<td>Use a FSM to keep track of rotors</td>
<td></td>
</tr>
<tr>
<td><strong>Simon (Game)</strong></td>
<td><a href="https://en.wikipedia.org/wiki/Simon_(game)">https://en.wikipedia.org/wiki/Simon_(game)</a></td>
</tr>
<tr>
<td>Use push buttons and LEDs to display patterns that the player needs to copy. The pattern should get progressively more challenging.</td>
<td></td>
</tr>
<tr>
<td>Use a pseudo-random number generator or look-up table to create the pattern.</td>
<td></td>
</tr>
<tr>
<td>Use the sound peripherals to generate music for the game.</td>
<td></td>
</tr>
<tr>
<td><strong>Typing tutor (Game)</strong></td>
<td>Similar to Whac-a-Mole (<a href="https://en.wikipedia.org/wiki/Whac-A-Mole">https://en.wikipedia.org/wiki/Whac-A-Mole</a>)</td>
</tr>
<tr>
<td>Use 7-segment displays to display timer and score.</td>
<td></td>
</tr>
<tr>
<td>Use OLED or VGA to display a letter or number that the player needs to press. If the player presses the correct key on the keyboard within a time limit, they get a point. If they fail to press the correct key in the time limit or press the wrong key, they lose a point.</td>
<td></td>
</tr>
<tr>
<td>Optional: Player can set difficulty to make the timer shorter or longer or even display multiple keys to press at a time.</td>
<td></td>
</tr>
<tr>
<td><strong>Morse Code Reader</strong></td>
<td>Design a project to read Morse Code. Since every letter cannot be displayed on the 7-segment display it suffices to only recognize numbers. You could also come up with a clever use for the LEDs.</td>
</tr>
</tbody>
</table>