CSE141L Course Schedule

<table>
<thead>
<tr>
<th>Lecture #</th>
<th>Date</th>
<th>Time</th>
<th>Room</th>
<th>Topic</th>
<th>Assignment Due</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tue. 6/28</td>
<td>6 - 7:15 PM</td>
<td>Center 105</td>
<td>Assignment 1: ISA</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Tue. 7/5</td>
<td>6 - 7:15 PM</td>
<td>Center 105</td>
<td>Assignment 2: Assembler &amp; ISS</td>
<td>#1</td>
</tr>
<tr>
<td>3</td>
<td>Tue. 7/12</td>
<td>6 - 7:15 PM</td>
<td>Center 105</td>
<td>Assignment 3: Datapath</td>
<td>#2</td>
</tr>
<tr>
<td>4</td>
<td>Tue. 7/19</td>
<td>6 - 7:15 PM</td>
<td>Center 105</td>
<td>Assignment 4: CPU</td>
<td>#3</td>
</tr>
<tr>
<td></td>
<td>Tue. 7/26</td>
<td></td>
<td></td>
<td>Lab # 4 Report and Electronic Submission Due No Lecture</td>
<td></td>
</tr>
</tbody>
</table>

July 26 - 29: Students demonstrate their CPU Design to TAs by appointment.

Lab Due Dates

- Lab 4 Due:
  - Before 6:00 PM Tuesday, July 26th
  - You must make an appointment to demonstrate your CPU July 27 - July 29
- You cannot make any changes to the design between the time you submit to your demo to TA
- No late submissions!
Microprocessor Design Steps

- Design Instruction Set Architecture (ISA)
- Develop software generation tools
- Code applications
- Develop instruction set simulator (ISS)
- Design datapath, verify it
  - Design the Processor, simulate logic
  - Verify the processor
  - (Fabricate the chip: not in this class!)

Lab 4 Assignment

- Design logic for complete 8-bit processor architecture
- Use LogicWorks 5 to design logic
- Simulate the operation executing 3 programs from Lab 1
Processor Organization

Note: Blocks in your architecture equivalent to those in blue are to be implemented in Lab4

What you must include

- Reset logic
- Program Counter Logic
  - Non-control transfer instructions
  - Control transfer instructions
  - Halt instruction
- Data path modified for Load/Store instructions
- Control logic
- Instruction counter
  - Initialize on reset
  - Freeze on HALT instruction
What you will turn in for this Lab

- Summary of your ISA from Lab 1 and assembly code with machine code for 3 programs.
- Printed schematics for the top level CPU as well as all the lower level modules you designed in LogicWorks 5.
- All the LogicWorks 5 files you created (to be submitted via electronic submission).
- Answers to following questions.

Questions

- What changes did you make in your original ISA and why?
- What is instruction count for each one of the three programs? How do the numbers compare with those for the ISS. If the numbers are different, why?
- What are the strengths of your design?
- What are the deficiencies of your design?
- Which instruction will determine the clock frequency of your processor, i.e. is responsible for the critical path?
- Which instruction is most expensive in terms of the number of gates required?
Question Continued

• Having gone through a complete CPU design experience, what would you do differently in your ISA to:
  – Decrease static and dynamic instruction count
  – Simplify data path design
  – Simplify CPU design
• If you were to pipeline the execution, what would the pipeline stages be? Give at least three issues that will complicate the design of your processor.

Lab 4 Grading

• It is your responsibility to make an appointment with one of the TAs before 7/26/05
• Show TA that your CPU design works before end of Friday, July 29th.
• You should test the programs using the data patterns given in Lab1.
• The TAs may test your design for correct functionality using their own data files that satisfy the constraints outlined in Lab 1.
Useful Hints

- Build hierarchical design
- Test thoroughly at every level of hierarchy
  - Connect binary switches and hex keypads to provide inputs
  - Connect binary and hex displays to observe behavior
- Write an assembly program to test individual instructions in your CPU
  - Self-checking programs are ideal!

What is Functional Verification?

- Making sure that your design is functionally correct!
  - Reset behavior
  - Instructions
    - Addressing modes
    - Algorithms in hardware, e.g. setting carry
    - Corner cases
    - Control transfer: branch/jump
  - Memory access
  - Special features
    - e.g. HALT in our case
Importance of Verification

- General purpose processor must run, without a flaw, any application running on it!
- Programmers will use the CPU in ways you never imagined!
- Processor may be used in mission critical applications
- It is costly to fix bugs in processors
  - Chip mask and fabrication costs
  - System HW and SW redesign
  - Lost market opportunity

A Program to Test 8-bit CPU
Before you leave

• Remember to make appointment with the TAs to show your operational design