CSE 141L
Computer Architecture Lab
Summer Session I, 2004

Lecture 3
Pramod V. Argade
CSE141L: Computer Architecture Lab

Instructor: Pramod V. Argade (p2argade@cs.ucsd.edu)
Office Hours:
   Tue.  7:30 - 8:30 PM (Center 109)
   Wed. 4:30 - 5:30 PM (AP&M 4141)

TA:
   Hua (Kevin) Yang (huhu@cs.ucsd.edu)
   Office Hour: Mon/Tue 3 - 4:30 PM

Tutors:
   Anthony Choi (buchoi@ucsd.edu)
   Office Hour: Tue/Wed 1:30 - 3 PM
   Dianne Shaw (dsshaw@ucsd.edu)
   Office Hour: Tue 4 - 5:30 PM, Thu 4:30 - 6
   Hou Choi (hochoi@ucsd.edu)
   Office Hour: Mon 1:30 - 3 PM, Fri. 3 - 4:30

Lecture: Tue. 6 - 7:15 PM, Center 109

Reference Book: LogicWorks4, Capilano Computing Systems
   Available in Bookstore

Web-page: http://www.cse.ucsd.edu/classes/su04/cse141L
# 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. 6/29</td>
<td>6 - 7:15 PM</td>
<td>Center 109</td>
<td>Assignment 1: ISA</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>Tue. 7/6</td>
<td>6 - 7:15 PM</td>
<td>Center 109</td>
<td>Assignment 2: ISS</td>
<td>#1</td>
</tr>
<tr>
<td>3</td>
<td>Tue. 7/13</td>
<td>6 - 7:15 PM</td>
<td>Center 109</td>
<td>Assignment 3: Datapath</td>
<td>#2</td>
</tr>
<tr>
<td>4</td>
<td>Tue. 7/20</td>
<td>6 - 7:15 PM</td>
<td>Center 109</td>
<td>Assignment 4: CPU</td>
<td>#3</td>
</tr>
<tr>
<td></td>
<td>Tue. 7/27</td>
<td></td>
<td></td>
<td>No Lecture</td>
<td></td>
</tr>
</tbody>
</table>

July 26 - 29: Students demonstrate their working CPU Design to TAs by appointment. Assignment #4 reports are due during the demonstration.
Lab Due Dates

• Lab 3 Due:
  ⇒ In AP&M 2444
  Before 5:30 PM Tuesday, July 20\textsuperscript{th}

• No late submissions!

• I plan to disclose a range of clock cycles from your Lab 2 reports (no names)
Microprocessor Design Steps

✓ Design Instruction Set Architecture (ISA)
✓ Code applications
✓ Develop software generation tools
✓ Develop instruction set simulator (ISS)
  • Design datapath, verify it
  • Design the Processor, simulate logic
  • Verify the processor
  • Fabricate the chip
Processor: A Stage Machine!

Flipflop(s)

Combinational Logic
What is a data path?

- Path along which data in a processor passes
- Data path consists of
  - Internal storage
    - General purpose register file
    - Special registers
  - ALU
  - Some control logic
Example Instruction Formats

Example 1: ADD Instruction

ADD $RD, $Constant

$RD = $RD + $Constant

Example 2: MOV Instruction

MOV $RD, $RS

$RD = $RS
Note: Blocks in your architecture equivalent to those in green are to be implemented in Lab3
Lab 3 Assignment

• Design data path for your 8-bit processor architecture
• Use LogicWorks 4 to design the data path
• Simulate the data path
What you must include

• All internal storage in your architecture.
  – General purpose register file
  – Stack
  – Accumulator
  – Special registers, status flag(s), etc.

• Arithmetic Logical Unit
  – Implement data path for all instructions that manipulate data in some way
    • MOV, both immediate-to-register and register-to-register
    • ADD, SUB, XOR, SHIFT, …

• Interconnections between modules

• Logic to control components of ALU
What you don’t have to implement

- Following instructions
  - Control transfer instructions (branches, jumps, etc.)
  - Memory load/store instructions
- Program Counter (PC)
- I-Mem
- D-Mem
How you should test data path

• Stimulus generation:
  – Binary switches
  – Hex keypads

• Mechanisms to observe results:
  – Binary displays
  – Hex displays

• Internal variables on the timing diagrams
What you will turn in for this Lab

- Summary of your ISA from Lab 1.
- Printed schematics for the top level data path as well as all the lower level modules you designed in LogicWorks 4.
- Printout that shows example input data, ALU opcode and the result from the ALU for each instruction.
- All the LogicWorks 4 files you created (to be submitted using turnin script).
- Description of the procedure to follow in order to test the operation of the data path for each instruction for your processor. This is so your TA can test your design.
- Answers to following questions.
Questions

• Which instruction is the most expensive in terms of number of gates it requires (no need to give the exact gate count, just give the reasoning).
• What tricks did you use to decrease the logic in your data path by sharing the logic among more than one instruction?
• How does your "move constant to register" and "move register to register" instruction work? Is it a special case of another ALU instruction, or does it use special data path elements?
• Using your data path, explain how you will load a register with a value from a memory location and store contents of a register to a memory location.
You should show on waveforms:

- Ability to load a constant with the required number of bits for your ISA into any appropriate register (e.g. any general purpose register).
- Correct operation of the ALU for all the ALU opcodes supported.
- Ability to have the same register to be the source and destination of an instruction.
- Show the ALU operation for interesting input data. For example, if the carry from the adder is saved in your architecture, use data that both does and does not set the carry bit.
Working knowledge of LogicWorks 4 Assumed!

- Various components available in Standard Libraries, such as, D Flip-flop, logic gates, multiplexors, registers, adders, clock, binary switch, binary display, hex keypad, hex display, etc.
- How to connect a bus to various components.
- How to define a sub-circuit bottom up, i.e. create a circuit and then use it to define the pins on the parent symbol.
- How to simulate a circuit and generate waveforms.
- How to print a circuit and waveforms.
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
  – Test combinations of control signals
  – Test all corner cases for data

• Bugs buried deep inside hierarchy are hard to find!
Lab Due Dates

• Lab 3 Due:
  ⇒ In AP&M 2444
  Before 5:30 PM Tuesday, July 20th

• No late submissions!

• I plan to disclose a range of clock cycles from your Lab 2 reports (no names)