Introduction

- Administration
- Overview of this course
- Assignment #1
- Examples
- Download design tool
- Get into groups of 3

Please put your cellphone on vibrate
http://cseweb.ucsd.edu/classes/su09/cse141L/index.html

| INSTRUCTOR         | Isaac Chu  
|--------------------|------------
|                    | ichu@ece.ucsd.edu (ichu@cs.ucsd.edu)  
|                    | Mobile Office (TAs have my phone #)  
| TEXTBOOK           | None  
| TA                 | Hung-wei Tseng  
|                    | h1tseng@cs.ucsd.edu  
|                    | Lab Office hours: Mon/Tue 2-6pm  
|                    |  
|                    | Vasileios Kontorinis  
|                    | vkontori@cs.ucsd.edu  
|                    | Lab Office hours: Thu/Fri 2-6pm  
|                    |  
|                    | Ankur Jain  
|                    | ajain@ucsd.edu  
|                    | Lab Office hours: Sat/Sun 1-5pm  
| Lab Room           | Ebu 3b, Room B240  
| Grades             | 20% Assignment 1  
|                    | 20% Assignment 2  
|                    | 20% Assignment 3  
|                    | 20% Assignment 4  
|                    | 20% Assignment 5  
|                    | No late turn-in for all assignments  

week 1 Page 2
What is Instruction Set Architecture?

A list of instructions a processor can execute
- Arithmetic — add, sub
- Logical — and, or, xor
- Data — load, store
- Control — branch, jump
- (skip) Interrupts, I/O, exceptions

Op-code definition (machine code)
- Registers — which registers
- Memory location offsets
- Address mode (jump address)

Hardware definition
- Registers — How many regs?
- Memory — How big?
- Computing — what operations are supported? add, sub, shift
- Jump (program counter) — How many bits?
Reduced Instruction Set Computer

- Simple instructions (simple op-code decoding).
- RISC processors are faster.
- Achieve higher performance through pipelining.
- There are more instructions in RISC.
- But compiler must generate more assembly "code".
Complex Instruction Set Computer

- Each instruction can execute several low-level operations
  - Ex: add data from several memory locations & store back to memory in one instruction (more hardware).
- Complex instructions (i.e. string operation, complex numbers)
- Faster operation (hardware) but CPU is more complex & larger.
CISC vs. RISC

- Both RISC & CISC have evolved.
- CISC has pipelining.
- RISC is becoming more complex.
- RISC is in MIPS, Play-station, iPod, & iPhone.
- CISC is in INTEL x86.
- INTEL got $$ and market share.
1. Invent your assembly language to:
   a. Sort an array of integers (array size 10 max)
      i. Use this sequence: 1, 6, 31, 9, 3, 18, 17, 0, 1, 5, 2
      ii. (TA will change numbers during grading)
   b. Plus one of the following features:
      i. Determine how many 1's are in the largest number of the array (ex. Largest number is 10110 -> 3 ones)
      ii. Determine how many odd numbers are in the array
2. Define your ISA (instruction set architecture)
   a. What are your instructions? Must have branching, memory, and ALU instructions.
   b. How many registers do you need?
   c. How many bits is your instruction set?
   d. Etc...
3. Write an assembler to convert your assembly language to machine code (in any language: c, java, c#)
4. Demo your program to 141L TA/tutor. No written report required.
5. Install Xilinx Webpack Tool V9 or V10
6. Due next Friday 6pm 8/14 (demo to TA & save work to TA's USB drive). Grading breakdown for this assignment:
   a. Your ISA definition 25%
   b. Your bubble sort program + 1-feature 25%
   c. Your assembler design - opcodes (25%)
   d. Your assembler machine code (25%)
Ex 1 - assembly language

- 32-bit MIPS assembly language will be used as example for the lecture, you must create your own ISA.
- In this example, I only show "bubble sort" w/ 5 numbers

```
.rdata
$MSG:
    .asciiz "hello world...\n"

.text
.globl main

main:
    ori  $t0, $zero, 5  #
    sw   $t0, 0($v0)    # store value 5 into memory space 0
    ori  $t0, $zero, 2  #
    sw   $t0, 4($v0)    # store value 2 into memory space 1
    ori  $t0, $zero, 9  #
    sw   $t0, 8($v0)    # store value 9 into memory space 2
    ori  $t0, $zero, 1  #
    sw   $t0, 12($v0)   # store value 1 into memory space 3
    ori  $t0, $zero, 4  #
    sw   $t0, 16($v0)   # store value 4 into memory space 4

    addi $t0, $zero, 0
    addi $t1, $v0, 20

    # now sort it using bubble sort
    or   $t0, $zero, $v0  # $t0 get the memory address [0]
    addi $t0, $t0, -4     #
    addi $t3, $v0, 16     # stopping point [4] for var j
    addi $t2, $t0, -4     # initialize t2 <> t0

OUTER:
    addi $t0, $t0, 4
    beq  $t0, $t2, OUTERDONE # outer loop is done
    or   $t2, $zero, $t0

INNER:
    addi $t2, $t2, 4

    lw    $a0, 0($t0)
    lw    $a1, 0($t2)
    slt   $t8, $a0, $a1  # if (a0 < a1) t8 = 1, else t8 = 0
    beq   $t8, $zero, SWAP
```
NOSWAP:
    sw $a0, 0($t2)
    sw $a1, 0($t0)

SWAP:
    beq $t2, $t3, OUTER
        # inner loop done, go to outer loop
    j INNER

OUTERDONE:
    addi $t1, $v0, 20
    or $t0, $zero, $v0
        # $t0 get the memory address [0]
    beq $t1, $t0, DONE
        # loop is done

Print2:
    lw $a0, 0($t0)
        # load memory
    ori $v0, $zero, 1
        # Service = print_int
    syscall
    addi $t0, $t0, 4
        # increment the memory index
    j Print2

DONE:
    jr $ra
        # Return to caller
Ex 2 - assembler program

1. Show the 141MIPSAssembler in VS2008.
2. Show the output machine code.
• [www.xilinx.com](http://www.xilinx.com)
• Webpack is free. Must register online to use it.
• Version 11 is large & buggy from the start. Avoid it!
• You can use previous version V9 or V10.
  ○ I have a copy here with me. If you have USB drive, you may copy it right after class (soon).
  ○ TA will have them on USB Drive as well.
Lab Partners

Do not leave today until you:

1. Get in a group of **THREE**.
2. Exchange contact info.
3. Choose a group leader.
4. Have the group leader come and enter the names in excel-sheet and remember your group #.
5. Get start on your project early (this weekend)!
Reminder

You may now turn on your cellphone ringer