CS2011: Introduction to Machine Organization and Assembly Language

Class 1: Introduction

Today’s Class:
- What we’ll be learning!
- Logistics of the course
  - Who’s who
  - Syllabus
  - Grading
  - etc.
- Introduction to Machine Organization (very brief!)
- Introduction to Assembly Language
  - What
  - Why
  - Brief Example

CS2011: What We’ll be Learning

- Major parts to the course:
  - Data representation (homework 1)
  - Machine language (homework 2)
  - Assembly language (homeworks 3-5)
  - Machine architecture (logic, circuits, microprogramming, etc.) (homework 6)

Teaching Staff

- Instructor: Janet Burge
  - office hours: TBD in FL144
  - email: jburge@cs.wpi.edu
- TAs:
  - Andrei Dancus, Chris Shoemaker
  - Office hours (to be determined)
    - will be in FL A20 (sub-basement)
  - cs2011_ta@cs.wpi.edu
Class Mailing Lists

- For the instructor and TAs:
  - cs2011_ta@cs.wpi.edu
  - This is what you should use for questions on the classwork!
- For the whole class (also includes instructor and TAs):
  - cs2011@cs.wpi.edu
- You should be added to these automatically when you register for the course.

Class Web Page

- http://www.cs.wpi.edu/~jburge/courses/c01/cs2011
- This will be updated throughout the course.
- This is also the best place to check to see when office hours are.

Textbook

- Required:
  - K.R. Irvine, *Assembly Language for Intel-Based Computers*
- Optional:
  - A.S. Tannenbaum, *Structured Computer Organization*
    - Five copies of the previous edition are on reserve at the library.

Syllabus

- From web-page
Lectures

- Lecture slides will be (mostly) available on the web in advance of the lecture (PDF format).
- Most lectures will also include in-class exercises on the material covered in the lecture.
  - these are NOT quizzes
  - they are done in groups
  - you will be expected to have your book with you in class
  - you should take notes on them!

Labs

- You must attend your scheduled lab section:
  - Section C02: 12:00 Washburn, 216
  - Section C03: 1:00 Washburn, 216
  - Section C04: 2:00 Kaven, 202
  - Section C05: 8:00 Kaven, 202
- Bring:
  - your text (Irvine)
  - a floppy disk
  - a pen or pencil – you will be turning in your lab results
- You will need a Novell account!!!

Homework

- Due at the START of class on the day it is due. Homework is always due each Friday. Homework due-dates will not be extended until Monday unless Friday’s class was cancelled.
- No late homework will be accepted under any circumstances (but lowest grade will be dropped)
- Homework will be returned during lab the following week.
- If you do not pick it up in lab, you can pick it up from the TA assigned to record grades for your lab section during their office hours (this will be posted on the web)

Exams

- Closed book
- You can bring one 8 ½ by 11 sheet of paper (with notes, hand-written, both sides)
- NO CALCULATORS
- No make-up exams except in cases of serious health-related emergencies with appropriate documentation signed by a medical doctor.
- Sample exam questions will be posted one week prior to the actual exam. Sample exam answers will not be posted. Some problems will be worked out in-class, for others you will need to check your answers with a TA.
Grading Policy

- Two Exams – 25% each: 50%
- Six Homework Assignments, 7% each, lowest dropped: 35%
- Six Laboratory Assignments, 2% each, lowest dropped: 10%
- In-Class Exercises – 5% total, lowest dropped

Academic Honesty Policy

- All assignments must be performed individually.
  - Students may ask general questions of each other but should not see each other's homework assignments under any circumstances.
  - Working together on a homework assignment is considered an academic honesty violation.
  - Comparing homework answers is also an academic honesty violation.
- It is expected that students follow the policies of this class. There will be opportunities for collaborative work during class but homework assignments are not collaborative.

Academic Honesty Policy (continued)

- If cheating is suspected the following procedure will be followed:
  - The students will receive a photocopy of their homework assignment rather than the original.
  - The instructor will examine the homework to determine if an academic honesty violation has occurred.
  - The instructor will meet with the students prior to computing final grades for the course to discuss the situation and give an appropriate penalty.

Computer Organization
CPU Components

- Control Unit – fetches instructions, decodes instructions, causes instructions to be carried out.
- Arithmetic logical unit (ALU) – performs arithmetic operations (addition, etc.) on data.
- Buses – communication pathways connecting different devices/components.
- Registers – high speed memory cells (don’t need to go through the bus to access). They vary in number and purpose on different machines.

Registers

- 8, 16, or 32 bit high-speed storage locations inside the CPU
- They can be accessed at a much higher speed than conventional memory.
- When optimizing for speed, use registers.
- Four types: general purpose, segment, index, status, and control

What is Assembly Language?

- From Irvine:
  “Assembly language is a machine-specific programming language with a one-to-one correspondence between its statements and the computer’s native machine language”
- What does this mean?
  - Each computer has its own machine language and, therefore, its own assembly language
  - Each assembly language statement is translated (by an Assembler) into one machine language statement
What is Machine Language?

• Machine language is the language understood by the computer.
• The language is made of numbers which can be interpreted by the processor.
• The processor has a microprogram that translates machine instructions into hardware signals

Example

• Assembly language instruction:
  ```
  mov al, 5 ;move 5 to the al register
  (registers are high speed storage locations in the CPU)
  ```
• Machine language instruction:
  ```
  101100000000101
  10110000: operation – move operand to the AL register
  00000101: operand – the number 5 (in binary)
  ```

Why Learn Assembly Language?

• Some programming MUST be done in assembler:
  – direct communication may be needed with the operating system or peripheral devices
  – size and speed constraints (embedded programming)
  – custom hardware with no high level language support
• Assembly can be used in debugging
• A learning tool!

Context

• table 1 from the text
Comparison of Assembly Language and High Level Languages

- table 2 from the text

Differences in Assembly Languages

- Our move example:
  - Intel: MOV AL, 5
  - MC6800: MOVQ #5, D0
- Defining memory:
  - Intel: COUNT DW 5
  - MC68000: COUNT DS.W 1
    reserves space only!

Assembly Language Instructions

- A symbolic representation of a single machine instruction.
- Consists of a mnemonic and zero or more operands:
  clc ; just a mnemonic
  inc ax ; single operand
  mov ax, bx ; two operands
- Operand types:
  10 (immediate operand)
  count (variable or memory operand)
  AX (register operand)
  [0200] (memory location)

Sample Program

- Figure 5 from text