Computer Organization and Design

CS 251 / Winter 2013

Dr. Igor Ivkovic

iivkovic@uwaterloo.ca

Motivational Quotes

- By the time you've sorted out a complicated idea into little steps that even a stupid machine can deal with, you've certainly learned something about it yourself. – Douglas Adams

- It's hardware that makes a machine fast. It's software that makes a fast machine slow.
– Craig Bruce

- People who are really serious about software should make their own hardware. – Alan Kay

- The protean nature of the computer is such that it can act like a machine or like a language to be shaped and exploited. – Alan Kay

[BrainyQuote. Online, 2013. http://www.brainyquote.com]

Syllabus /1

- Computer Organization and Design
  CS 251 (Sections 001 & 002) – Winter 2013
- **Instructor:** **Dr. Igor Ivkovic**
- **Email:** **iivkovic@uwaterloo.ca**
- Office hours: 4:30-6:00W in DC 2555E
- Lectures:
  (Section 001) 02:30-03:20MWF in MC 2054
  (Section 002) 03:30-04:20MWF in MC 2054

My CARE™ Approach

- **What can you expect from me as your professor?**

  Clear and Technical Lectures

  Approachable and Helpful

  Resources and Material for Studying

  Engaging Theory and Practical Applications

Syllabus /2

- **Course Description:**
  - Develop an accurate mental model of the architecture and organization of physical computers
  - Explain theoretical and practical concepts relevant to the structure and design of modern digital computers
  - Enable students to understand and predict the behaviour and performance of computer programs executing on real machines
  - Prerequisites: CS 134 or 136 or 145; Computer Science students only
  - Antirequisites: ECE 222, PHYS 353, SE 141

Syllabus /3

- **Learning Objectives:**
  - Design simple combinational and sequential hardware at the logic gate level in order to implement simple algorithms and computations, such as addition, multiplication, and data-path control for processors
  - Describe number systems used by computer hardware, including IEEE floating point and two's complement at the binary level
  - Explain the limits of the binary representations, including rounding error and overflow conditions
  - Explain how machine language is executed by hardware, and describe a simple processor architecture for executing a RISC machine language

Syllabus /4

- **Learning Objectives:**
  - Explain some simple processor implementation optimization techniques, and how differences in code ordering can impact performance for processors using these optimizations
  - Analyze performance of a code sequence executing on a pipelined architecture, and reorganize (given) code in order to improve performance on a pipelined architecture
  - Explain basic cache and virtual-memory architectures, and how they can impact performance
  - Analyze a memory-access trace relative to these architectures in order to predict performance
  - Describe a basic multicore and multiprocessor architecture, and explain the key factors that will affect performance and scalability of programs written for such an architectures

Syllabus /5

- Announcements and Lecture Notes on LEARN:
  - https://learn.uwaterloo.ca/
- Assignments and Grades:
  - http://www.student.cs.uwaterloo.ca/~cs251/
- Attendance:
  - **You are highly encouraged to attend all lectures and keep detailed class notes**
- **Required Readings:**
  - CS 251 Course Notes (available from Media.doc, MC 2018)
  - Patterson and Hennessy: "Computer Organization and Design". 4th Revised Edition, Morgan Kaufmann, 2011, ISBN 0123747503 (previous editions are acceptable)

Syllabus /6

- Grading Scheme:
  - Assignments — 20% (five assignments + bonus 1% for A0)
  - Midterm Exam — 30% (held on Feb 14th, 04:30-06:20Th)
  - Final Exam — 50% (to be scheduled by the Registrar)
- Exam Policy:
  - Students are required to obtain a passing grade for the weighted grade of the midterm and the final exam. **If the weighted grade for the midterm and the final exam is less than 50%, the assignments grade will not be counted.**
  - A missed midterm exam will receive a mark of 0, unless there is a valid documented reason. If a documented reason is provided for missing the midterm, its weight is applied to the final exam.

Syllabus /7

- Final Exam Policy:
  - If a documented reason is provided for missing the final exam, a grade of incomplete (INC) is given, and the final exam must be written at the end of the next term the course is offered.
  - **A copy of the documented reason must be given to the instructor.** For a missed final exam, upon the receipt of a valid doctor's note, the student's term work is evaluated to determine if a grade of INC is suitable.
  - If an INC is granted, the student's grade will be calculated using the weightings of the course components from the term the student was registered, the student's term marks from the registered term, and the mark from the final exam when it is written. The final exam should be written with the scheduled exam in the following semester.
  - **A copy of the documented reason must also be given to the Instructional Support Coordinator (ISC) for the course.**

Syllabus /8

- Assignments:
  - There will be five course assignments. The assignments will be due throughout the term, and will be typically due on Fridays. Assignments must be submitted in the assignment drop boxes in the MC building.
  - **No late submissions will be accepted.**
  - An assignment not handed in receives a mark of 0, unless there is a documented reason. If a documented reason is supplied, the weight of the missing assignment is distributed across the other assignments. A copy of the documented reason must be given to and approved by the instructor.

Syllabus /9

- Assignments/Midterms Delivery:
  - **Graded material will be returned to students in class or via email if applicable.**
- Group Work Policy:
  - You are permitted to discuss the assignment questions with your classmates.
  - **However, all assignment submissions must be done individually, and no copying is allowed.**

Syllabus /10

- Assignment Remarking:
  - Attach a sheet of paper to your assignment clearly stating the questions that you want remarked. Include any supporting evidence for your case.
  - No more than two weeks after the assignment is handed back, return your assignment with the remarking page to the IA during a tutorial or an office hour.
  - **Note that the entire assignment is examined when remarking, so the assignment could receive a grade lower than the one originally assigned.**

Syllabus /11

- In-Class Discipline:
  - **Students are encouraged to attend all lectures, but are required not to be disruptive during lectures out of respect for their classmates and for the instructor.**
  - Disruptive behaviour includes talking with people next to them, playing YouTube videos, Facebook and Twitter updates, etc. Also note that comments and feedback about the course are welcome, but outside of lecture time.
  - A student who is found to be disruptive during lectures will be given only one verbal warning for the term.
  - At the instructor's discretion, if the same student continues to be disruptive, for each incident of disruption, they will be penalized 3% of their final mark for the course, and may be asked to leave the lecture during which the disruption incident occurred. They would also be liable for any missed milestones during the lecture.

Syllabus /12

- Academic Integrity:
  - In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check the Office of Academic Integrity's website for more information,
  - http://uwaterloo.ca/academic-integrity/
- Grievance:
  - A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy70.htm
  - When in doubt please be certain to contact the department’s administrative assistant who will provide further assistance.

Syllabus /13

- Academic Discipline:
  - A student is expected to know what constitutes academic integrity to avoid committing academic offenses and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offense, or who needs help in learning how to avoid offenses (e.g., plagiarism, cheating) or about “rules” for group work/collaboration should seek guidance from the course professor, academic advisor, or the undergraduate associate dean. For information on categories of offenses and types of penalties, students should refer to Policy 71, Student Discipline,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy71.htm
  - For typical penalties check Guidelines for the Assessment of Penalties,
  - http://www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm

Syllabus /14

- Avoiding Academic Offenses:
  - Most students are unaware of the line between acceptable and unacceptable academic behaviour, especially when discussing assignments with classmates and using the work of other students. For information on commonly misunderstood academic offenses and how to avoid them, students should refer to the Faculty of Mathematics Cheating and Student Academic Discipline Policy,
  - http://www.math.uwaterloo.ca/navigation/Current/cheating_policy.shtml
- Appeals:
  - A decision made or penalty imposed under Policy 70, Student Petitions and Grievances (other than a petition) or Policy 71, Student Discipline may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72, Student Appeals,
  - http://www.adm.uwaterloo.ca/infosec/Policies/policy72.htm

Syllabus /15

- Note for Students with Disabilities:
  - The Office for Persons with Disabilities (OPD), located in Needles Hall, Room 1132, collaborates with all academic departments to arrange appropriate accommodations for students with disabilities without compromising the academic integrity of the curriculum. If you require academic accommodations to lessen the impact of your disability, please register with the OPD at the beginning of each academic term.

Syllabus /16

- Course Topics Overview:
  - Introduction (2 hours)
    - Overview of computer organization. Measuring performance.
  - Digital Logic Design (6 hours):
    - Gates, truth tables, and logic equations. Combinational logic and basic components. PLAs and ROMs. Memory elements. Finite-state machines (informally).
  - Data Representation and Manipulation (6 hours)
    - Signed and unsigned integer representations. Addition and subtraction. Using (informal) finite-state machines to control datapaths. Multiplication. IEEE floating-point representation.
  - Basic Processor Design (5 hours)
    - Basic processor datapaths. Processor design using single-cycle control.

Syllabus /17

- Course Topics Overview Continued:
  - Pipelining (5 hours)
    - Pipelined datapaths. Data hazards. Branch hazards. Load-use hazards.
  - Input/Output (3 hours)
    - Buses and bus arbitration. Storage (disks). Interrupts. Point-to-point (routed) networks. Networks and clusters.
  - Memory Hierarchies (3 hours)
    - Caches: direct-mapped, fully-associative, set-associative. Virtual memory. Page tables and TLBs.
  - Multiprocessing (3 hours)
    - Multi-processor systems and core processors. Synchronization and locking. Cache consistency.

Code Changes that Affect Performance /1

- **Consider the following C/C++ program:**

```
#define NR 10000
#define NC 10000

int a[NR][NC];

int main() {
    int i, j;
    for (i=0; i < NR; i++){
        for (j=0; j < NC; j++){
            a[i][j] = 32767;
        }
    }
    return 0;
}
```

User time: ~0.6s, Real time: ~1s

  - **Compile and run the program, and time it using “time”**

Code Changes that Affect Performance /2

- **Consider the following C/C++ program:**

```
#define NR 10000
#define NC 10000

int a[NR][NC];

int main() {
    int i, j;
    for (i=0; i < NR; i++){
        for (j=0; j < NC; j++){
            a[j][i] = 32767;
        }
    }
    return 0;
}
```

User time: ~4s, Real time: ~4.4s

  - **Compile and run the program, and time it using “time”**

Mapping Code to Binary /1

- **Consider the following C/C++ code fragment:**

```
swap(int v[], int k) {
   int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

- **Simplified mapping to MIPS assembly:**

```
swap:
   muli $2, $5, 4
   add $2, $4, $2
   lw $15, 0($2)
   lw $16, 4($2)
   sw $16, 0($2)
   sw $15, 4($2)
   jr $31
```

Mapping Code to Binary /2

- **Mapping to MIPS assembly via compiler explained:**

```
swap:
    muli $2, $5, 4 # k is stored in $5
                   # multiply by 4 to get k-th address
    add $2, $4, $2 # v is stored in $4
                   # get k-th address in the array v
    lw $15, 0($2) # load contents of v[k] into $15
    lw $16, 4($2) # load contents of v[k+1] into $16
    sw $16, 0($2) # store contents of $16 into v[k]
    sw $15, 4($2) # store contents of $15 into v[k+1]
    jr $31 # jump to the address in $31
```

Mapping Code to Binary /3

- **Mapping to MIPS binary via assembler:**

```
00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Welcome to the matrix! ☺

- We will explain the above mapping later in the course

Food for Thought

- Download and Read Assignment #0 Specifications
  - **Assignment #0 is intended as a simple introduction to the MIPS assembly language that we will use later in the term**
  - This assignment is optional, but if you complete it you can earn up to 1% bonus mark (the bonus will not apply if your final course grade is 100%)