Lecture 1 – Course Introduction

Rami Abielmona
University of Ottawa
January 13, 2015
ITI 1100 B
Digital Systems I

Presentation Outline

- Introduction to computing
- Digital systems and abstraction levels
- Basic structure of a computer
- Course details
- Programs at EECS
- Computer engineering design
- Key terms

Introduction to Computing

- Computer engineering is a very dynamic field
  - Chips double in speed every 18 months (Moore's Law), due to
    - Decreasing transistor sizes
    - Increasingly efficient computer architectures
- What will you learn in this course ?
  - How digital systems work
  - How to digital systems are designed and realized
  - You will design many yourselves!
- Performance/Power tradeoff
  - Performance = [frequency * IPS] / # of instructions
  - Power = 0.5 * capacitance * voltage^2 * frequency

Transistor Size

- Sizes of transistors are decreasing every year
- Moreover, we can integrate multiple logic circuits (simple and/or complex) within the same chip
  - e.g. System-on-Chip

	Year					
	1999	2001	2003	2006	2009	2012
Transistor gate length	0.14 μm	0.12 μm	90 nm	65 nm	50 nm	35 nm
Transistors per cm^2	14 million	16 million	24 million	40 million	64 million	100 million
Chip size	800 mm^2	850 mm^2	900 mm^2	1000 mm^2	1100 mm^2	1300 mm^2

Table 1.1. A sample of the SIA Roadmap.

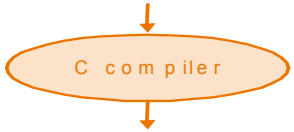
What is a Digital System?

- A system can be
  - Digital or analog
  - Sequential or combinational
  - Synchronous or asynchronous
  - Data or control
- A digital system contains circuits that are composed of logic gates, that are themselves composed of transistors
  - We have to move away from the design details
  - We have to move up the abstraction chain towards the behavioral level

Levels of Abstraction

High-level language program (in C)

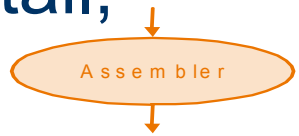
```
swap(int v[], int k)
{int temp;
  temp = v[k];
  v[k] = v[k+1];
  v[k+1] = temp;
}
```



- Delving into the depths reveals more information

Assembly language program (for MIPS)

```
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
```

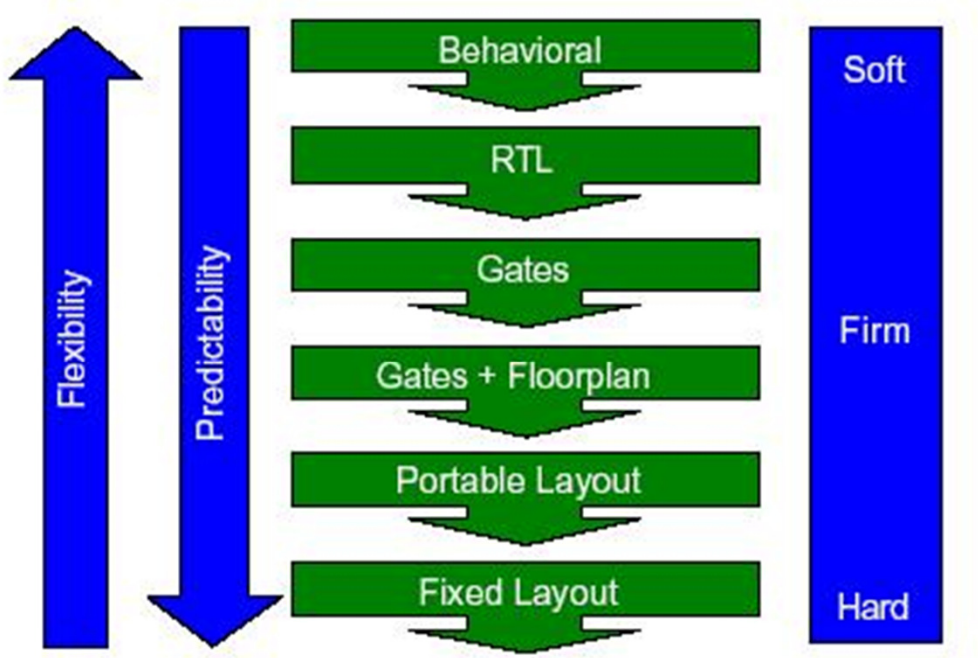


- An abstraction omits unneeded detail, helps us cope with complexity

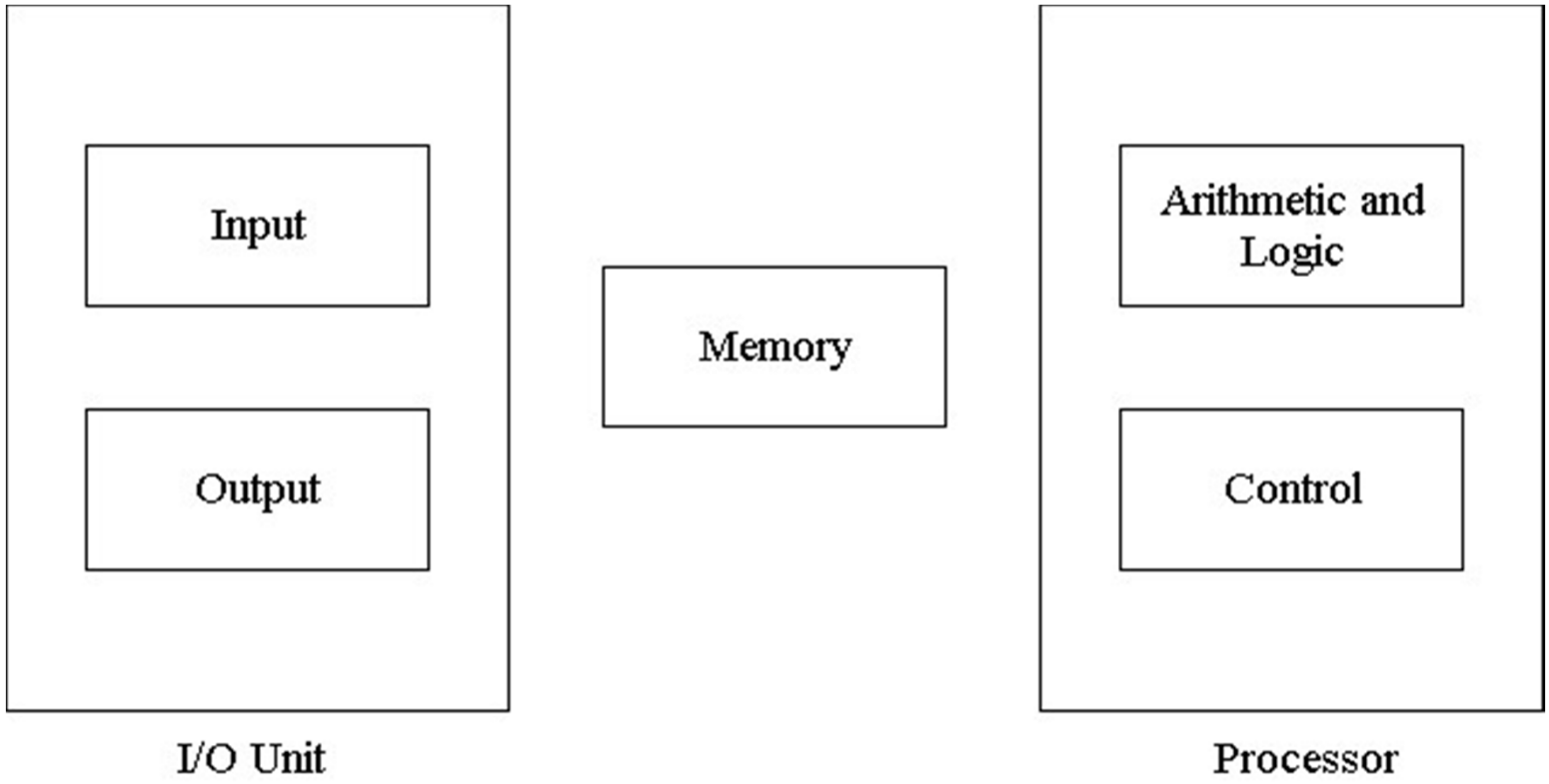
Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000100001110000110000010000 1
10001100011000100000000000000000
10001100111110010000000000000100
10101100111110010000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

Levels of Abstraction (2)



Basic Structure of a Computer



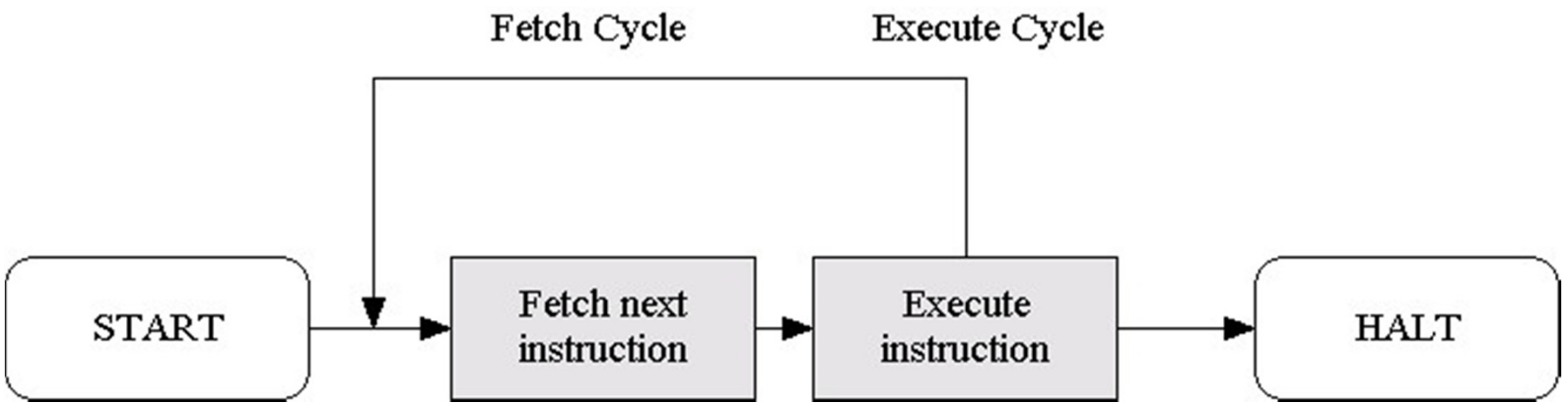
Basic Structure of a Computer (2)

- Input Unit
  - Keyboards, joysticks, trackballs, microphones and mice
- Output Unit
  - Printers and graphic displays
- Memory Unit
  - Primary (cache, RAM, HDD) and secondary (CD-ROM, tape drives)
- Arithmetic and Logic Unit (ALU)
  - Executions completed here and stored in fast-access registers
- Control Unit (CU)
  - Provides control to all other units, including timing signals

Basic Operation of a Computer

1. The computer *accepts* information in the form of programs and data through an input unit and *stores* it in memory
2. The information stored in memory is *fetched*, under program control, and *processed* in an ALU
3. The processed information *leaves* the computer through an output unit
4. All activities inside the computer are *directed* by the control unit

Basic Instruction Cycle



Computer performs the instruction cycle forever! (or at least until it is turned off, faces an error or is instructed to do so)

ITI 1100B Details

Course Website	http://www.site.uottawa.ca/rabielmo/iti1100b_w15
Required Textbook	Mano and Ciletti, Digital Design, 5th edition, Pearson-Prentice Hall, 2012.
Recommended Textbook	Wakerly, Digital Design: Principles and Practices, 4th edition, Prentice Hall, 2005.
Prerequisites	None
Teaching Assistants	TBD

Chapters Covered in Book

- Digital Systems and Binary Numbers (Chapter 1)
  - How do we represent and process numbers?
- Boolean Algebra and Logic Gates (Chapter 2)
  - Boolean functions and forms
  - Digital logic gates
- Gate-Level Minimization (Chapter 3)
  - Karnaugh-map method
  - Various implementations
- Combinational Logic (Chapter 4)
  - How do we add, subtract, compare, encode, etc?
- Synchronous Sequential Logic (Chapter 5)
  - How do we maintain a state?
- Registers and Counters (Chapter 6)
  - How do we store information? How do we count?

Important Dates

- **Final Exam - TBD - TBD - TBD**
- Lab #6 – April 14/15, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Assignment #6 – April 06, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- Lab #5 – March 31/April 01, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Assignment #5 – March 23, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- Lab #4 – March 17/18, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Quiz #3 – March 17, 2015 – 14:30-15:00 – MRT 218
- Lab #3 – March 10/11, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Assignment #4 – March 09, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- **Midterm Exam – TBD – TBD – TBD**
- Lab #2 – February 24/25, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Assignment #3 – February 23, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- Lab #1 – February 10/11, 2015 - 20:00 - ITI1100B Box (STE 1st floor)
- Quiz #2 – February 10, 2015 – 14:30-15:00 – MRT 218
- Assignment #2 – February 09, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- Assignment #1 – February 02, 2015 - 20:00 – ITI1100B Box (EECS 1st floor)
- Quiz #1 – January 27, 2015 - 14:30-15:00 – MRT 218

Labs and Exams

- LABS
  - 6 labs have been organized
  - You will be working in groups of 2 students
  - The same group will work together in the same session thoughout the semester
  - A lab report is expected from each group for each lab
  - The report should be prepared according to the guidelines found in the laboratory manual
- EXAMS
  - The mid-term and final exams are closed-book exams and cover all the material in the weeks prior to the exam
  - Their exact dates will be disclosed in the next couple of weeks

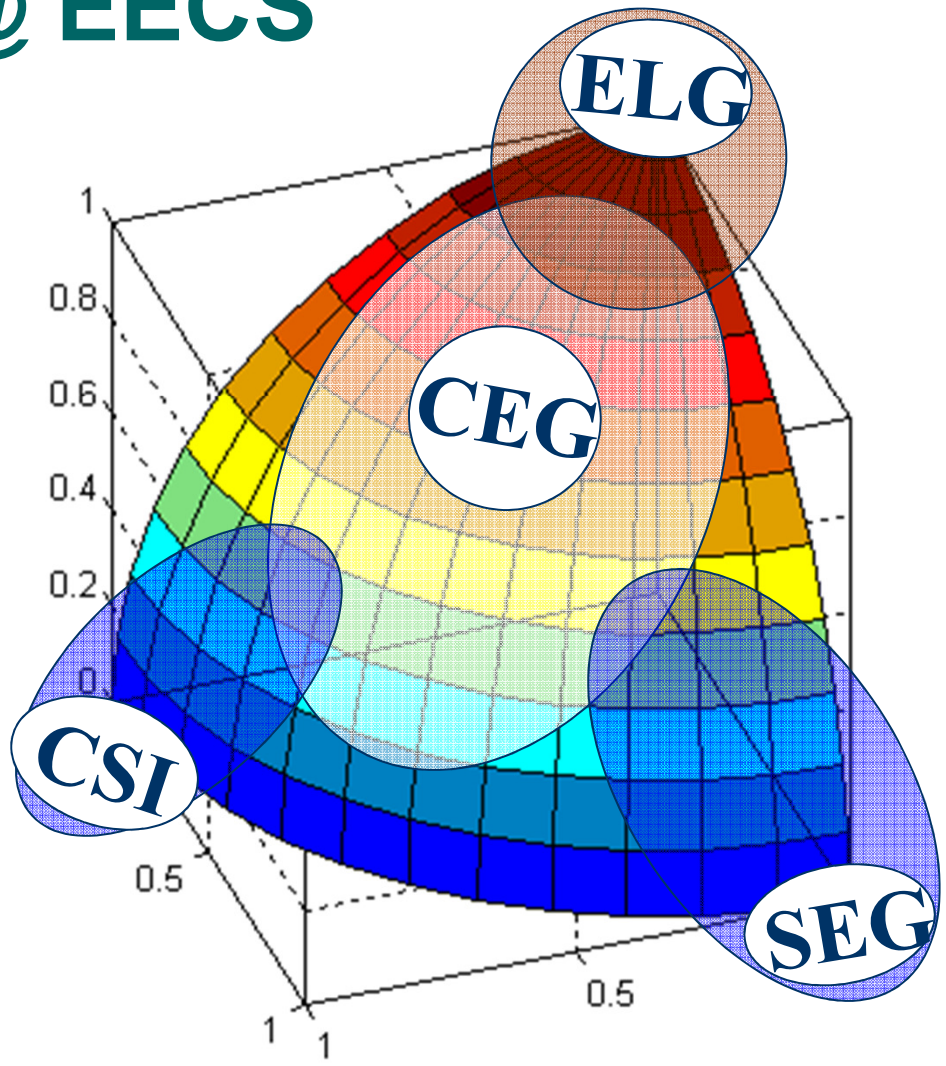
Additional Announcements

- Please consult the course web site for updates, course marks, as well as a host of other material related to this course;
- Assignment and lab rules and regulations will be posted on the course web site, as well as how to deal with late copies of the aforementioned;
- Appointments can be set up through email if extra consultation hours are needed;
- Any quiz can be replaced by the submission of a 1-page article summary for a paper that the professor provides you;
- Advanced lectures will be given on topics such as VHDL, synthesis and advanced optimization methods. These will be announced on the course web site;
- Note that **to pass this course**, you need at least 50% on the final exam.
- ***Plagiarism is strictly not allowed*** (http://www.uottawa.ca/plagiarism.pdf)
  - Copying on exams, assignments, lab reports, etc
  - Would be turned over to Faculty for investigation and possible sanctions

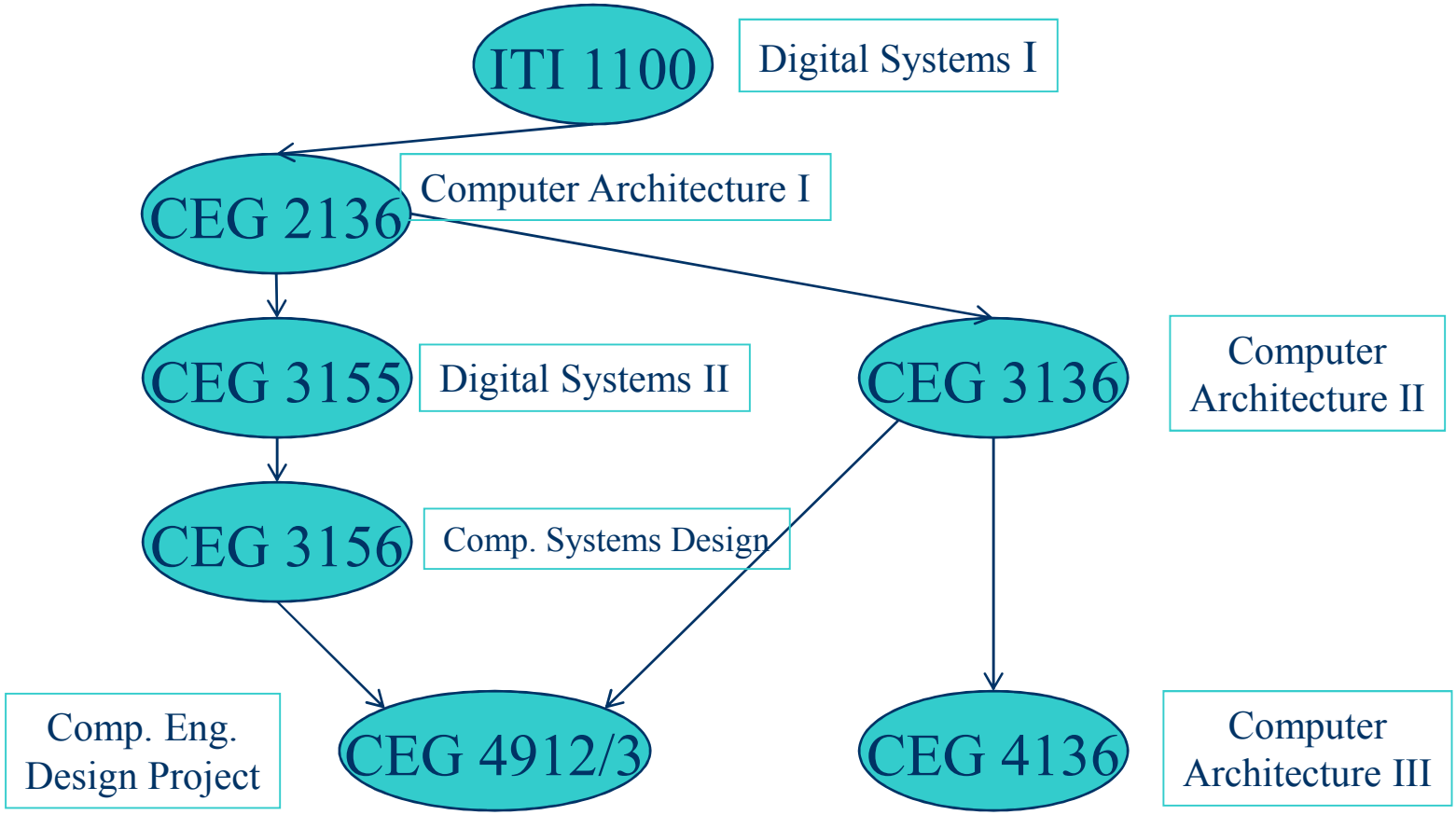
Course Evaluation

Deliverable	*Percentage*	*Comments*
Quizzes	5%	Best 2 out of 3
Assignments	10%	6 in total
Laboratories	15%	6 in total
Midterm Exam	20%	Non-transferable to final
Final Exam	50%	Must get at least 50% to pass

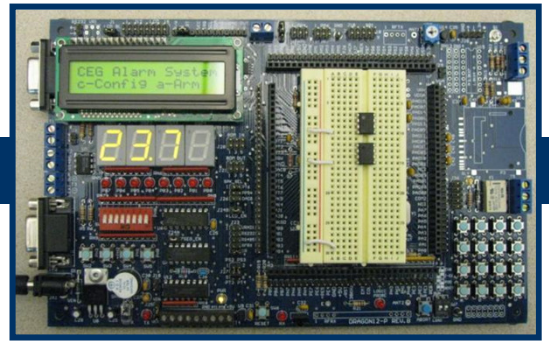
Programs @ EECS



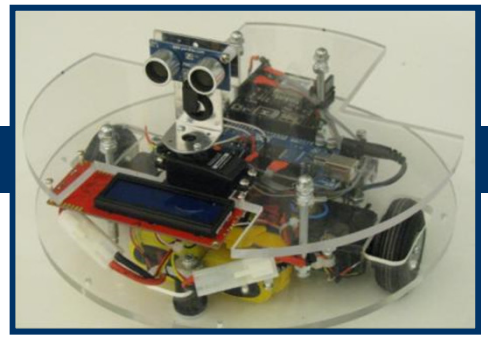
Hardware/Software at our Faculty

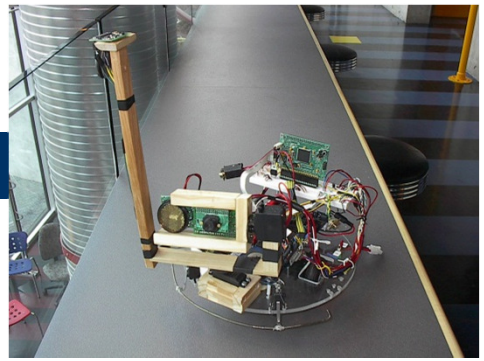


Computer Engineering Design



CEG3136 – Computer Architecture II





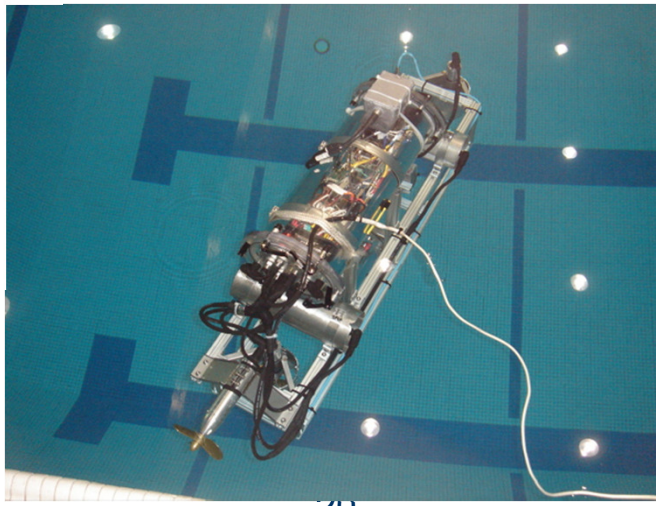
CEG4912/CEG4913 – Capstone Design Project

Unmanned Airplane (UAV)

Wind Turbine

Arduino Microcontroller Board

Autonomous Underwater Vehicle



Key Terms and Review Points

- Abstraction Levels
- Analog System
- Arithmetic Logic Unit (ALU)
- Asynchronous Circuit
- Combinational Logic
- Control Unit (CU)
- Digital System
- Input/Output Unit (I/O)
- Instruction Cycle
- Memory Unit
- Performance vs Power Tradeoff
- Sequential Logic
- Synchronous Circuit
- System-on-Chip

See you next class!