Chapter 1: Introduction

Prof. Ben Lee
Oregon State University
School of Electrical Engineering and Computer Science

Contents

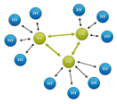
1.1 The Role of Computers in Modern Society

Embedded System

- Importance of desktops, servers, and laptops is unmistakable.
- But, we often overlook another facet of computers – embedded systems!
- Embedded systems are designed to perform one or a few dedicated functions, and more importantly, are embedded as a part of a complete device that often includes hardware and other mechanical parts.
- The meaning of embedded system has evolved - any system with a computer that is not a desktop or laptop!

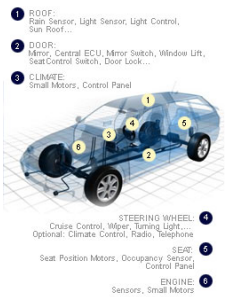
Example Embedded Systems

802.15.4/ZigBee
LR-WPAN Devices



Wireless Sensor Networks

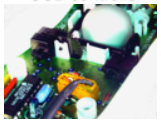
Automotive Applications



RFID

Motor Control

USB controller



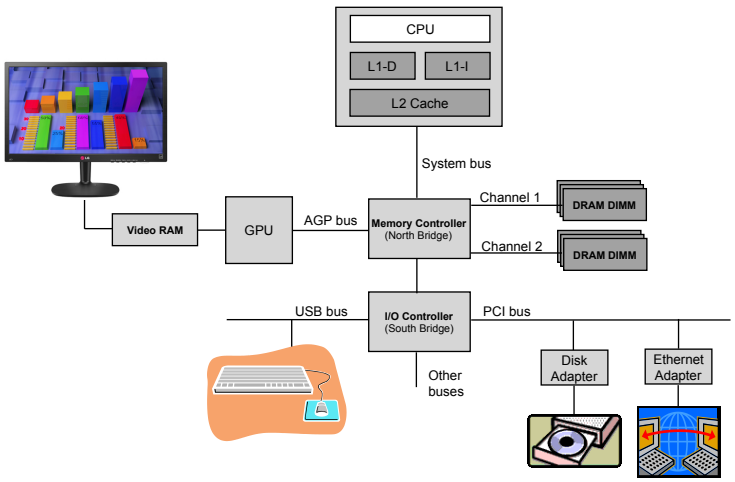
Remote Access Controller

Why Focus on Embedded System?

- More than 95% of devices with computers are embedded systems.
  - They account for the most of the world's production of microprocessors!
- Therefore, understanding how they are programmed and how their internal structure is organized are essential for future engineers and computer scientists.

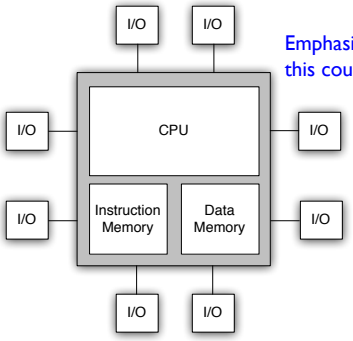
1.2 Spectrum of Computers and Their Processors

Desktop System



Embedded System

System on Chip (SOC)



Designed to control I/O devices.

Emphasis of this course

- Lower performance, complexity, power, and cost.
- Low-end embedded system:
  - 8-bit or 16-bit processor @ few to tens of MHz.
  - Memories of tens to hundreds of Kilobytes (KB).
- High-end embedded systems:
  - 32-bit processors clocked at several hundred MHz.
  - Memories in the order of Megabytes (MB)
  - e.g., Portable Media players, GPS, car infotainment systems, feature phones, etc.

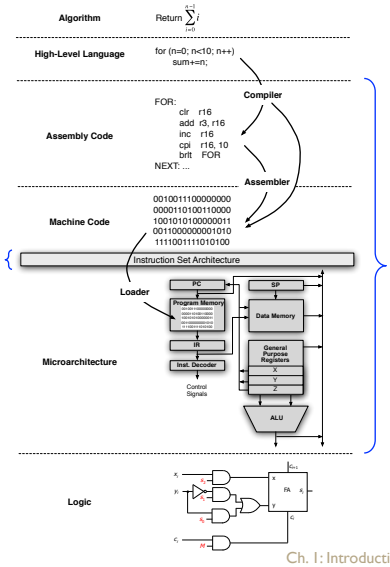
Embedded or General-Purpose?

- Recently, a new class of mobile devices have emerged that saddle between embedded systems and general-purpose computers:
  - Smartphones and pad/tablet devices with large memories, OS, graphics processor, and multi-cores.
- Thus, the line between general-purpose and embedded systems has become blurred!

1.3 Objectives of this Course

Computing System Hierarchy

Interface between software and hardware



Emphasis of this course

Objectives of the Course

- Understand the interrelationship between hardware and software:
  - These two topics are not distinct.
  - This is the course where ECE and CS disciplines merge.
- Understanding the essence of these concepts makes both software and hardware designers better at what they do:
  - Programmers can write better programs by understanding how processor execute their programs.
  - Hardware designers can design better processors by understanding the operational requirements of programs.

Course Logistics

Logistics

Instructor:	Prof. Ben Lee
Office:	3013 Kelley Engineering Center
Tel:	7-3148
E-mail:	benl@eecs.orst.edu
Web:	http://www.eecs.orst.edu/~benl/Courses/ECE375_fa15.html
Textbook:	- *Computer Organization and Assembly Language: Embedded Systems Perspective*, by Ben Lee (will be provided) - ATmega 128 Datasheet
Office Hours:	4 - 5PM TR and by Appt.
TAs:	TBA
Labs:	Section 010: Tuesday 10:00AM - 11:50AM Section 011: Wednesday 6:00PM - 7:50PM Section 012: Tuesday 12:00PM - 1:50PM Section 013: Monday 12:00PM - 1:50PM Section 014: Wednesday 4:00PM - 5:50PM Section 015: Monday 10:00AM - 11:50AM

Grading Policy

- Midterm 25%
- Quizzes (2) 10%
- Laboratory (7) 30%
- Assignments (4) 10%
- Final Exam 25%

Course Outline (1)

Chapter 1: Introduction

Chapter 2: Assembly Language Fundamentals
- Introduction
- How Do We Speak the Language of the Machine
- Instruction Set Architecture
- Instruction Format
- A Pseudo-ISA

Chapter 3: Computer Organization Fundamentals
- Introduction
- Memory
- Microoperations
- Organization of the Pseudo-CPU

Course Outline (2)

Chapter 4: Atmel's AVR 8-bit Microcontroller, Part 1 - Assembly Programming
- Introduction
- General Characteristics
- Addressing Modes
- Instructions
- Assembly to Machine Instruction Mapping
- Assembler Directives
- Expressions
- Assembly Coding Techniques
- Mapping Between Assembly and High-Level Language
- Anatomy of an Assembly Program

Course Outline (3)

Chapter 5: AVR 8-bit Microcontroller, Part 2 - Input/Output
- I/O Ports
- Interrupts
- Timers/Counters
- USART

Chapter 7: Digital Components
- Introduction
- Multiplexers
- Decoders
- Memory Elements
- Registers
- Memory
- Register File
- ALU

Course Outline (4)

Chapter 8: AVR 8-bit Microcontroller, Part 3 – Microarchitecture
- Microarchitecture
- Instruction Format
- Basic Datapath Components
- Multicycle Implementation
- Execution of More Complex Instructions
- Control Unit Design
- Pipeline Implementation

Questions?