CENG 336 Introduction to Embedded Systems Development

Lecture 1: An Introduction to Computers and Embedded Systems
Course Schedule

• Lecture:
  – Section 1: Volkan Atalay Tue 10:40 BMB2 Thu 10:40,11:40 BMB1
  – Section 2: Ali Doğru Tue 10:40 BMB3 Thu 10:40,11:40 BMB2

• Lab:
  – To be announced
  – TAs: Alper Kilic akilic@ceng.metu.edu.tr, Fatih Gokce fgokce@ceng.metu.edu.tr

• Text Book
  – No Textbook

• Reference Material
  – PIC 16F87X Data sheets (http://www.microchip.com)

• Course webpage:
Grading Policy (tentative)

<table>
<thead>
<tr>
<th>MIDTERM</th>
<th>QUIZES</th>
<th>FINAL</th>
<th>LAB</th>
</tr>
</thead>
<tbody>
<tr>
<td>25%</td>
<td>15%</td>
<td>35%</td>
<td>35%</td>
</tr>
</tbody>
</table>
What is this course about?

CENG 336 Introduction to Embedded Systems Development (2-2)3


Prerequisite: CENG 232.

http://catalog.metu.edu.tr/ceng.php#desc
What is an embedded computer system?

• What is a computer?
  – [Merriam-Webster Dictionary] one that computes; specifically: a programmable electronic device that can store, retrieve, and process data.

• Classification of Computers
  – by price and computing power
  – Dedicated controllers – Embedded controllers
  – Personal computers
  – Mainframes
  – Supercomputers
Types of Computers

• Mainframes
  – largest and most powerful
    • massive amounts of memory
    • use large data words…64 bits or greater
    • mostly used for military defense and large business data processing
    • examples: IBM 4381, Honeywell DPS8
• Microcomputers
  – range from small controllers that work with 4 bit words to the PCs we are familiar with that work with 32 bit words
  • modern microcomputers are becoming indistinguishable from early minicomputers...functionally speaking
  • large variety of uses from specialized controls like a printer to personal publishing
  • the CPU is usually 1 Integrated Circuit (IC) called a microprocessor
  • examples: Intel 8051 controller chip, IBM PC, Apple Macintosh

• Supercomputers
  – fastest and most powerful mainframes
  • contains multiple central processors
  • used for scientific applications, and number crunching
  • now have teraflop performance
• Types of Computers
• **Microcomputer Structure**
• Microprocessor Evolution
• Internal Architecture
• Introduction to Programming the 8086
Microcomputer Structure

• Central Processing Unit (CPU)
• Memory
• Input/Output (I/O) circuitry
• Buses
  – Address bus
  – Data bus
  – Control bus
Memory

• A mixture of RAM and ROM…may also include magnetic hard disks and optical disks

• 2 purposes of memory
  – store the binary codes for the sequence of instructions specified by programs
  – store binary data that the computer needs to execute instructions
I/O

• the way the computer communicates with the outside world

• peripherals are connected to the I/O ports
  – printers, modems, keyboard, mouse, scanner
  – Universal Serial Bus (USB)

• ports
  – physical devices needed to interface with the computer’s internal buses
  – actually a set of D flip-flops connected in parallel
  – how do we distinguish between an input port and an output port?
CPU

• the “brains” of the computer
• its job is to fetch instructions, decode them, and then execute them
• contains:
  – an Instruction Pointer register which contains the address of the next instruction
  – general purpose registers for temporary storage
  – circuitry to generate signals to the control bus
Address bus

- consists of 16, 20, 24, or 32 parallel signal lines (wires)
  - these lines contain the address of the memory location to read or written
  - just how many unique addresses can an address bus specify?
Data bus

- consists of 8, 16, or 32 parallel signal lines
  - these are bi-directional... meaning that data can be read from/written to either memory or a port
  - only one device at a time can have its outputs enabled, even though many will have their outputs connected to the same data bus
  - this requires the devices to have three-state output
Control bus

- consists of 4 to 10 parallel signal lines
- CPU sends signals along these lines to memory and to I/O ports
  - examples: Memory Read, Memory Write, I/O Read, I/O Write
Typical CPU Behavior

Goal:
– Read a word of data from a memory location

Process:
– CPU first sends out the address along the address bus to the memory device
– CPU then sends the Memory Read signal along the control bus
– the output from the memory device travels back to the CPU along the data bus
• Types of Computers
• Microcomputer Structure
• Microprocessor Evolution
• Internal Architecture
• Introduction to Programming the 8086
Microprocessor Evolution

- microcomputers are commonly categorized by the number of bits that their ALU can work with at a time
  - regardless of the number of address lines or data lines
    - the first commercial microprocessor was the Intel 4004…a 4-bit device combined with other devices to make a calculator
    - next came the Intel 8008…an 8-bit device, but it required many additional devices to be a functional CPU
    - Intel 8080…another 8-bit device, but it only required 2 additional devices
      - it also used different transistors making it much faster, and started the 2nd generation of microprocessors
      - then Motorola entered the market with the MC6800
Directions taken by Microprocessors

- **Embedded controllers**
  - used to control smart machines
  - printers, auto braking systems
  - also called microcontrollers

- **Bit-slice processors**
  - custom-designed hardware and custom-designed instruction set made by connecting devices…each part becomes a slice needed for a specific application
  - these were created because general-purpose CPUs were not fast enough or did not have a rich enough instruction set
General Purpose CPUs

• Intel released the 8086, a 16-bit microprocessor, in 1978
• Motorola followed with the MC68000 as their 16-bit processor
• the 16-bit processor works with 16 bit words, rather than 8 bit words…
  – instructions are executed faster
  – provide single instructions for more complex instructions…multiply and divide
32 bit Processors

• 16 bit processors evolved into 32 bit processors
  – now able to work with gigabytes \(10^9\) bytes and terabytes \(10^{12}\) bytes
  – Intel released the 80386
  – Motorola released the MC68020
The 8086 Microprocessor Family

• Characteristics
  – 16 bit microprocessor
  – 16 bit data bus
    • it can read from or write to memory and I/O ports either 8 or 16 bits at a time
  – 20 bit address bus
    • it can address $2^{20}$ memory locations
      – each location is 1 byte (8 bits) wide, thus 16 bit words will require consecutive memory locations
Members of this family

- **8088**
  - same as 8086 but has an 8 bit data bus
- **80186/80188**
  - enhanced instruction set…but still backwards compatible
- **80286**
  - designed for a multi-user, multi-tasking microcomputer
  - users a virtual address mode to prevent collision of users’ programs
- **80386/80486**
  - first Intel 32 bit processor
  - can directly address up to 4 GB of memory
Story of the Humble Transistor

- 1947 – Shockley, Brattain, and Bardeen invented the transistor at Bell Labs
- 1961 – First commercial IC by Fairchild/TI
- 1963 – CMOS invented
- 1965 – Moore’s law
- 1968 – State of the art: 64 transistor chip
- 1978 – IA 8086: ~10K transistors
- 1986 – IA 386: ~100K transistors
- 1990 – IA 486: ~1 M transistors
- 1998 – IA P2: ~10 M transistors
- 2004 – IA P6: ~1 B transistors
Moore’s Law

Transistor capacity doubles every 18 months

Source: Intel
Embedded System = Computers Inside a Product
Definition

• **Embedded system**: any device that includes a programmable computer but is not itself a general-purpose computer.

• **Computer purchased as part of some other piece of equipment**
  - Typically dedicated software (may be user-customizable)
  - Often replaces previously electromechanical components
  - Often no “real” keyboard
  - Often limited display or no general-purpose display device: don’t need all the general-purpose bells and whistles.
Embedded systems: Applications

- **Consumer segment**, e.g. cameras, camcorders, VCRs, washers, microwave ovens, …
- **Automobiles**, e.g., engine control, anti-lock brake, air bags, …
- **Office automation**, e.g., copiers, printers, FAX machines, …
- **Telecommunications**, e.g., cellular phones, PDAs, interactive game boxes, answering machines, …
- **Other industrial products**, e.g., door locks in hotel rooms, automatic faucets, …
Embedded Systems: Typical Characteristics

• Perform a single or tightly knit set of functions (not usually “general purpose”)
• Is part of a larger system that may not be a “computer”
• Works in a reactive and time-constrained environment
• Employs a combination of hardware & Software
  – Software provides features and flexibility
  – Hardware provides performance (Application specific processor design)
What makes an embedded system unique? – Hardware perspective

- Microprocessor technology
  - Personal computers
    - Pentium Chips, Power-PC chips
    - General purpose computers, raw computing power
    - 75 million units per year
  - Embedded controllers
    - Special purpose microcontrollers
    - 2.5 billion units per year, 30 times larger in unit volume
• GartnerGroup estimates 70 Billion µP used in embedded systems in 2001
• Other estimates say 50 to 120 Billion µP
• Average embedded system has 4 µP
• Of all µP sold, 90% go into “non-computers”, 10% in “computers”
• You will most likely end up working with a “non-computer” at some point in your career
What makes an embedded system unique? – Software perspective

• No operating systems
• Execute a single program, tailored exactly to the controller hardware
• Assembly language (vs. High-level language)
  – Not transportable, machine specific
  – Programmer need to know CPU architecture
  – Speed
  – Program size
  – Uniqueness
An embedded system example --
a digital camera

- Single-functioned -- always a digital camera
- Tightly-constrained -- Low cost, low power, small, fast
- Reactive and real-time -- only to a small extent
BMW 850i brake and stability control system

• Anti-lock brake system (ABS): pumps brakes to reduce skidding.
• Automatic stability control (ASC+T): controls engine to improve stability.
• ABS and ASC+T communicate.
  – ABS was introduced first---needed to interface to existing ABS module.
BMW 850i, cont’d.
Example: BMW 745i

- 2,000,000 LOC
- Windows CE OS
- 53 8-bit µP
- 11 32-bit µP
- 7 16-bit µP
- Multiple Networks
- Buggy!
Components of Embedded Systems

- Memory
  - Flash
  - DRAM
  - Memory Card

- Controllers
  - Flash Adapter / Memory Control
  - I/O Control

- Interface
  - RS-232C Interface
  - IrDA
  - LED
  - Bluetooth Module
  - Ethernet PHY
  - Ethernet Interface

- Processor
  - CPU
  - JPEG Co-Processor
  - Graphics Controller
  - LCD Interface
  - Video Encoder

- Coprocessors
  - Camera DSP
  - A/D
  - CCD

- Converters
  - Video Out

- Software
  - (Application Programs)
• Analog Components
• Sensors, Actuators, Controllers, …
• Digital Components
• Processor, Coprocessors
• Memories
• Controllers, Buses
• Application Specific Integrated Circuits (ASIC)
• Converters –A2D, D2A, …
• Software
• Application Programs
• Exception Handlers
Characteristics

• **Application Specific**
  – Applications are known a priori
  – Optimize for cost, area, power, and performance

• **Digital Signal Processing**
  – Signals are represented digitally

• **Reactive**
  – Reacts to changes in the system’s environment

• **Real-time**
  – Compute certain tasks before deadline

• **Distributed, Networked, …**
• **Reliability**
  - Probability of system working correctly provided that it was working at $t=0$

• **Maintainability**
  - Probability of system working correctly $d$ time units after error occurred.

• **Safety**
  - Not harmful for user

• **Security**
  - Confidential and authentic communication
Traditional Design Challenges

- Low cost
- Light weight
- Reliability
- Low power
- Portable
- Complexity
- Ease of use

- Mixed digital/analog requirements
- Shrinking time-to-market
- Short product lifetime
- Real-time processing
- Inherent concurrency
- HW/SW co-design
Recent Design Challenges

• **Design Complexity**

• **Ultra low power**
  – Highly adaptive
  – Active power management (voltage scaling, etc.)
  – Alternative energy source (scavenge, solar, etc.)

• **Internet aware**
  – Incorporate RF technologies
  – Networking capabilities

• **Verification**

• **Security**
Processor technology

- The architecture of the computation engine used to implement a system’s desired functionality
- Processor does not have to be programmable
  - “Processor” *not* equal to general-purpose processor

General-purpose (“software”)

Application-specific

Single-purpose (“hardware”)
General-purpose processors

• Programmable device used in a variety of applications
  – Also known as “microprocessor”

• Features
  – Program memory
  – General datapath with large register file and general ALU

• User benefits
  – Low time-to-market and NRE costs
  – High flexibility

• “Pentium” the most well-known, but there are hundreds of others
Single-purpose processors

• Digital circuit designed to execute exactly one program
  – a.k.a. coprocessor, accelerator or peripheral
• Features
  – Contains only the components needed to execute a single program
  – No program memory
• Benefits
  – Fast
  – Low power
  – Small size
Application-specific processors

- Programmable processor optimized for a particular class of applications having common characteristics
  - Compromise between general-purpose and single-purpose processors

- Features
  - Program memory
  - Optimized datapath
  - Special functional units

- Benefits
  - Some flexibility, good performance, size and power
• Applications