Fundamentals of Microprocessor and Microcontroller

Dr. Farid Farahmand

Updated: Monday, February 5, 2018
Evolution

First came transistors

Integrated circuits
- SSI (Small-Scale Integration) to ULSI
- Very Large Scale Integration circuits (VLSI)

1- Microprocessors (MPU)
- Microcomputers (with CPU being a microprocessor)
- Components: Memory, CPU, Peripherals (I/O)
- Example: Personal computers

2- Microcontroller (MCU)
- Microcomputers (with CPU being a microprocessor)
- Many special function peripheral are integrated on a single circuit
- Types: General Purpose or Embedded System (with special functionalities)
An embedded system is a special-purpose computer system designed to perform one or a few dedicated functions often with real-time.

An integrated device which consists of multiple devices:
- Microprocessor (MPU)
- Memory
- I/O (Input/Output) ports

Often has its own dedicated software.
Embedded Systems
Amazon Warehouse

Kiva Robot
Microprocessor-Based Systems

- Central Processing Unit (CPU)
- Memory
- Input/Output (I/O) circuitry
- Buses
  - Address bus
  - Data bus
  - Control bus
Microprocessor-based System
Microprocessor-Based System with Buses: Address, Data, and Control
Microprocessor-based Systems
Microprocessor

- The microprocessor (MPU) is a computing and logic device that executes binary instructions in a sequence stored in memory.

- Characteristics:
  - General purpose central processor unit (CPU)
  - Binary
  - Register-based
  - Clock-driven
  - Programmable
Microprocessor-based Systems

Microprocessor

- the “brains” of the computer
  - its job is to fetch instructions, decode them, and then execute them
  - 8/16/32/etc –bit (how it moves the data)
- contains:
  - Arithmetic Logic Unit
  - Register Arrays
  - Control Unit

**ALU** performs computing tasks – manipulates the data/ performs numerical and logical computations

**Registers** are used for temp. storage

**Control unit** is used for timing and other controlling functions – contains a program counter (next instruction’s address and status register)

**System software:** A group of programs that monitors the functions of the entire system
Remember
Microprocessor-based Systems

Memory

- Memory is a group of registers
  - 16 register – address: 0-15 – in binary: 0-1111; Address lines: A0-A3

- Serves two major purposes
  - storing the binary codes for the sequence of instructions specified by programs (program)
  - storing binary data that the computer needs to execute instructions (data)
Microprocessor-based Systems
Memory Types

- **R/W: Read/Write Memory; also called RAM**
  - It is volatile (losses information as power is removed)
  - Write means the processor can store information
  - Read means the processor can receive information from the memory
  - Acts like a Blackboard!

- **ROM: Read-Only memory**
  - It is typically non-volatile (permanent) – can be erasable
  - It is similar to a Page from your textbook
Microprocessor-based Systems
Memory Classification

Basic Technologies:
Semiconductor
Magnetic
Optical
(or combination)

System Memory

Read/Write Memory
R/WM

- Static R/WM
Expensive
Fast/

- Dynamic R/WM
Cheap
Slow

Read-Only Memory
ROM

- Erasable Memory

- EPROM
- EEPROM
- Flash Memory

- Permanent Memory

- Masked ROM
- PROM

Electronically Erasable
PROM

Onetime programmable
Microprocessor-based Systems

Memory Classification

- **Expensive**
  - One transistor and one capacitor to store a bit
  - Leakage problem, thus requires refreshing
  - Used for dynamic data/program storage
  - Cheap and slow!

- **Fast/Cheap**
  - 4/6 transistor to save a single bit
  - Volatile
  - Fast but expensive

- **Slow**
  - Static RAM
  - Dynamic RAM

Read-Only Memory

Erasable Memory
- EPROM
- EEPROM
- Flash Memory

Permanent Memory
- Masked ROM
- PROM

Onetime programmable

Electronically Erasable PROM
Erasable ROMs

- Marked Programmed ROM
  - Programmed by the manufacturer
- Programmable ROM (PROM)
  - Can be programmed in the field via the programmer
- Erasable Programmable ROM (EPROM)
  - Uses ultraviolet light to erase (through a quartz window)
  - OTP refers to one-time programmable
- Electrically Erasable Programmable ROM (EEPROM)
  - Each program location can be individually erased
  - Expensive
  - Requires programmer
- FLASH
  - Can be programmed in-circuit (in-system)
  - Easy to erase (no programmer)
  - Only one section can be erased/written at a time (typically 64 bytes at a time)
Microprocessor-based Systems

I/O Ports

- The way the computer communicates with the outside world devices
- I/O ports are connected to Peripherals
  - Peripherals are I/O devices
    - Input devices
    - Output devices
  - Examples
    - Printers and modems,
    - keyboard and mouse
    - scanner
    - Universal Serial Bus (USB)
Microprocessor-based Systems - BUS

- The three components – MPU, memory, and I/O – are connected by a group of wires called the BUS

- **Address bus**
  - consists of 16, 20, 24, or 32 parallel signal lines (wires) - unidirectional
  - these lines contain the address of the memory location to read or written

- **Control bus**
  - consists of 4 to 10 (or more) 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

- **Data bus**
  - consists of 8, 16, or 32 parallel signal lines
  - **bi-directional**
  - only one device at a time can have its outputs enabled,
  - this requires the devices to have three-state output
Expanded Microprocessor-Based System

1. Note the directions of busses

2. What is the width of the address bus?

3. What is the value of the Address but to access the first register of the R/WM?

You must know how to draw it!
Example

How much memory do we have?
Example

Total of 4K bytes of memory: $2^{12} (\text{FFF}) \rightarrow 12 \text{ bits} \text{ ; last values } 2^{12}-1 = 4096-1$
Example

8 bits

000

001

1G bytes of memory

30 bits!

$2^{30} - 1 = \text{3FFF FFFF}$

Next number: 4000 0000 (in Hex)
Example of an 8-bit MPU
So what are microcontrollers?
What is a Microcontroller?

- A microcontroller is a small computer on a single integrated circuit containing:
  - processor core,
  - memory,
  - programmable input/output peripherals
- Used for specific (embedded) applications
Embedded controllers

- Used to control smart machines
- Examples: printers, auto braking systems
- Also called microcontrollers or microcontroller units (MCU)
Embedded controllers
Software Characteristics

- 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
Microcontroller Unit (MCU) Block Diagram

- An integrated electronic computing and logic device that includes **three** major components on a **single chip**
  - Microprocessor
  - Memory
  - I/O ports
- Includes support devices
  - Timers
  - A/D converter
  - Serial I/O
  - Parallel Slave Port
- All components connected by common communication lines called the system bus.
First Microcontrollers

- IBM started using Intel processors in its PC
  - Intel started its 8042 and 8048 (8-bit microcontroller) – using in printers
- Apple Macintosh used Motorola
- 1980 Intel abandoned microcontroller business
- By 1989 Microchip was a major player in designing microcontrollers
  - PIC: Peripheral Interface Controller
## Different Microcontrollers (MCU)

- ARM core processors (from many vendors)
- Atmel AVR (8-bit), AVR32 (32-bit), and AT91SAM (32-bit)
- Cypress Semiconductor PSoC (Programmable System-on-Chip)
- Freescale ColdFire (32-bit) and S08 (8-bit)
- Freescale 68HC11 (8-bit)
- Intel 8051
- Infineon: 8, 16, 32 Bit microcontrollers[^9]
- MIPS
- Microchip Technology PIC, (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24), (32-bit PIC32)
- NXP Semiconductors LPC1000, LPC2000, LPC3000, LPC4000 (32-bit), LPC900, LPC700 (8-bit)
- Parallax Propeller
- PowerPC ISE
- Rabbit 2000 (8-bit)
- Renesas RX, V850, Hitachi H8, Hitachi SuperH (32-bit), M16C (16-bit), RL78, R8C, 78K0/78K0R (8-bit)
- Silicon Laboratories Pipelined 8051 Microcontrollers
- STMicroelectronics ST8 (8-bit), ST10 (16-bit) and STM32 (32-bit)
- Texas Instruments TI MSP430 (16-bit)
- Toshiba TLCS-870 (8-bit/16-bit).

[^9]: More detail can be found [here](http://en.wikipedia.org/wiki/Microcontroller)

What is the difference?
- 8/16/24/32 bits
- Architecture
- Package
- Capability
- Memory
- Software (IDE)/cloud
- ADC (10-12 bit)

A more complete list is here: [http://en.wikipedia.org/wiki/Category:Microcontrollers](http://en.wikipedia.org/wiki/Category:Microcontrollers)
MCU Architecture

- RISC
  - Reduced instruction set computer
  - Simple operations
  - Simple addressing modes
  - Longer compiled program but faster to execute
  - Uses pipelining
  - Most embedded system

- CISC
  - Complex instruction set computer
  - More complex instructions (closer to high-level language support)
  - x86 standard (Intel, AMD, etc.), but even in the mainframe territory CISC is dominant via the IBM/390 chip

Bench marks: How to compare MCUs together
MIPS: Million Instructions / second (Useful when the compilers are the same)
CISC vs RISC

<table>
<thead>
<tr>
<th>CISC</th>
<th>Pentium/x86 are CISC-based</th>
<th>RISC</th>
<th>ARM-based</th>
<th>Most mobile-phones</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex instructions require multiple cycles</td>
<td>Reduced instructions take 1 cycle</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Many instructions can reference memory</td>
<td>Only Load and Store instructions can reference memory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Instructions are executed one at a time</td>
<td>Uses pipelining to execute instructions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Few general registers</td>
<td>Many general registers</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

RISC and CISC architectures are becoming more and more alike.
*Read the LINK on the web site!*
CISC vs RISC

The Performance Equation
The following equation is commonly used for expressing a computer's performance ability:

\[
\frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}
\]

RISC does the opposite, reducing the cycles per instruction at the cost of the number of instructions per program.

The CISC approach attempts to minimize the number of instructions per program, sacrificing the number of cycles per instruction.

RISC and CISC architectures are becoming more and more alike. *Read the LINK on the web site!*
8-bit Controllers…
(Main Players)

- **Microchip**
  - RISC architecture (reduced instruction set computer)
  - Has sold over 2 billion as of 2002
  - Cost effective and rich in peripherals

- **Motorola**
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 68HC05, 68HC08, 68HC11

- **Intel**
  - CISC architecture
  - Has hundreds of instructions
  - Examples: 8051, 8052
  - Many difference manufacturers: Philips, Dallas/MAXIM Semiconductor, etc.

- **Atmel**
  - RISC architecture (reduced instruction set computer) – with CISC instruction set!
  - Cost effective and rich in peripherals
  - Claims to be very code efficient – less memory for the same code!
  - AVR (Advanced Virtual RISC): TunyAVR, MegAVR, XmegaAVR

- **Freescale**

- **Ziglog** (Z8)
A Bit About ARMs Architecture (Advanced RISC Machine)

- ARM design takes the RISC based computer design approach – Linux –like architecture
- ARM is a British semiconductor (and software) design company that designs and licenses ARM processor cores to semiconductor manufacturers
  - They just sell the ARM core
  - Other manufacturers license the core from them and then design microcontrollers around that core by adding in peripherals and memory to suit their design goals
- There are different cores for different applications
  - Cortex-M0/M0+, Cortex-M3, or Cortex-M4.
ARM Processor IP

Cortex-A15
- ARMv7-A
- High performance
- 32-bit CPU
- with enterprise class feature set

Cortex-A17
- ARMv7-A
- High performance
- 32-bit CPU
- with lower power and smaller area

Cortex-A57
- ARMv8-A
- Highest performance
- 64/32-bit CPU

Cortex-A9
- ARMv7-A
- Smallest and lowest power CPU

Cortex-A5
- ARMv7-A
- High efficiency
- 32-bit CPU
- big.LITTLE™ compatible

Cortex-A7
- ARMv7-A
- High efficiency
- 32-bit CPU
- big.LITTLE™ compatible

Cortex-A53
- ARMv8-A
- High efficiency
- 64/32-bit CPU
- big.LITTLE™ compatible

Cortex-R4
- Real-time standard

Cortex-R5
- Functional safety

Cortex-R7
- High performance
- 4G modem and storage

Cortex-M0
- Lowest cost
- Lowest power

Cortex-M0+
- Highest energy efficiency

Cortex-M3
- Performance efficiency

Cortex-M4
- Mainstream Control & DSP

Cortex-M7
- Maximum Performance
- Control & DSP

Enhanced system integration features

High Performance
High Efficiency
Real-time
Control
Applications of ARM-Based Microcontrollers

Who is using ARM? Check this out!

Most Cellphones!
Design Examples ..... 

Microcontrollers vs. Microprocessors
MPU-Based Time and Temperature System
MCU-Based Time and Temperature System
References

- Read about microcontrollers: [http://www.mikroe.com/en/books/picbook/2_01chapter.htm](http://www.mikroe.com/en/books/picbook/2_01chapter.htm)
- I used a few slides from here: [http://www.ceng.metu.edu.tr/courses/ceng336/_documents/introduction.pdf](http://www.ceng.metu.edu.tr/courses/ceng336/_documents/introduction.pdf)
- ARM related references:
  - [http://mc2.unl.edu/2013/10/03/getting-started-with-arm-microcontrollers/](http://mc2.unl.edu/2013/10/03/getting-started-with-arm-microcontrollers/)
  - [http://www4.cs.fau.de/Lehre/SS06/HS_AKES/slides/ARM.pdf](http://www4.cs.fau.de/Lehre/SS06/HS_AKES/slides/ARM.pdf) - Very good reference!
References - RISC

- http://cse.stanford.edu/class/sophomore-college/projects-00/risc/
- http://arstechnica.com/articles/paedia/cpu/pipelining-1.ars/4