# PIC Microcontroller Introduction

Vojtěch Krmíček

vojtec@ics.muni.cz

# Microcontroller I

- Also MCU or µC
- Computer-on-a-chip
  - microprocessor
  - high integration
  - low power consumption
  - self-sufficiency and cost-effectiveness
- Usually integrates additional elements
  - read-write memory for data storage
  - read-only memory, such as flash for code storage
  - EEPROM for permanent data storage
  - peripheral devices and input/output interfaces
- Clock speeds of a few MHz, but this is adequate for typical applications
- Frequently used in automatically controlled products and devices
  - automobile engine control systems, remote controls, office machines, appliances, power tools, and toys

# Microcontroller - Achitecture

- Von-Neuman Architecure
  - single "data" bus that is used to fetch both instructions and data
  - program instructions and data are stored in a common main memory
  - when such a controller addresses main memory, it first fetches an instruction, and then it fetches the data to support the instruction (if such data is needed).
- Harvard Architecture
  - separate data bus and an instruction bus
  - data and instructions are stored into separate memories that are accessed separately

# Micocontroller – CISC and RISC

- Complex Instruction Set Computer (CISC)
  - a large set of instructions that can perform complex tasks
  - e.g. the Intel 80X86 series, The Zilog Z80, 8051, 6HC11 etc.
  - features many instructions, addressing modes and takes more than 1 internal clock cycle to execute
- Reduced Instruction Set Computer (RISC)
  - a quite small set of instructions which carries out less task per command
  - complicated operations are carried out by combining many simple instructions
  - examples include usage in ARM, SPARC, Atmel AVR MIPS, PowerPC, PIC

# PIC microcontroller

- Harvard architecture microcontrollers by Microchip Technology
- "Programmable Interface Controller" or "Programmable Intelligent Computer“
- Popular due to low cost, wide availability, large user base, extensive collection of application notes, low cost or free development tools, serial programming, re-programming with flash memory capability

# Core Architecture I

- separate code and data spaces (Harvard architecture)
- a small number of fixed length instructions (RISC architecture)
- most instructions are single cycle execution (4 clock cycles)
- a single accumulator (W)
- a hardware stack for storing return addresses
- a fairly small amount of addressable data space (typically 256 bytes), extended through banking
- data space mapped CPU, ports, and peripheral registers
- the program counter is also mapped into the data space and writable
- no distinction between "memory" and "register" space because the RAM serves the job of both memory and registers

# Core Architecture II

- Data space – RAM
- Code Space - EPROM, ROM, or flash ROM
- Hardware stack
- Constant interrupt latency
- 35 to 70 intructions, skip intruction, conditional execution, branching

# Programming

- Only a single accumulator
- A small instruction set
- Some instructions can address RAM and/or immediate constants, while others can only use the accumulator
- Direct referencing of memory in arithmetic and logic operations
- Register-bank switching is required to access the entire RAM
- Conditional skip instructions are used instead of conditional branch instructions

- Wide range of device programmers (we will use PIC PRESTO)
- Microchip provides a freeware IDE package called MPLAB, which includes an assembler, linker, software simulator, and debugger

# Family Core Architectural Differences

- Baseline Core Devices - PIC10 series, as well as some PIC12 and PIC16 devices
  - 12-bit wide code memory, and a tiny two level deep call stack
- Mid-Range Core Devices - PIC12 and PIC16
  - 14-bit wide code memory, and 8 level deep call stack
- PIC17 High End Core Devices – not so popular, supressed by PIC18 architecture
  - a memory mapped accumulator, read access to code memory (table reads), direct register to register moves (prior cores needed to move registers through the accumulator)
  - an external program memory interface to expand the code space
  - an 8bit x 8bit hardware multiplier, a second indirect register pair
- PIC18 High End Core Devices
  - much deeper call stack (31 levels deep)
  - the call stack may be read and written
  - conditional branch instructions
  - indexed addressing mode (PLUSW)
  - extending the FSR registers to 12 bits, allowing them to linearly address the entire data address space
  - the addition of another FSR register (bringing the number up to 3)
- PIC24 and dsPIC 16-bit Microcontrollers
- PIC32MX 32-bit Microcontrollers

# What do the numbers and prefix stands for?

- prefix 12 is for chips with 8 pins
- prefix 16 is for 12-bit and 14-bit core chips with more than 8 pins
- prefix 18 is for 16-bit core chips
- the letter after number tells the memory type: C is for EPROM (OTP or windowed (except 16C84 that has EEPROM), F is for flash chips and JW is for chips that have windowed EPROM (UV erasable)
- The number (2 or 3 digits) after this letter identifies specific chip version
- Improved new version of certain PIC types are identified by appending an A to the type (A chips are in most aspects identical to their non-A predecessors, but there can be some differences usually on chip programming)