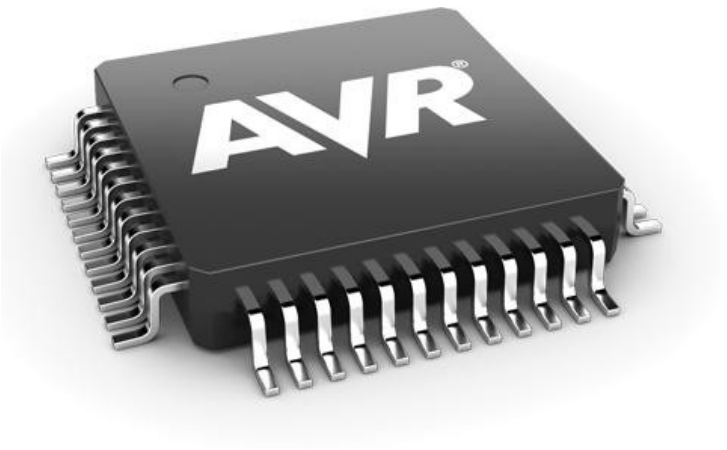
Sistemas Baseados em Microprocessadores

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

The AVR family of microcontrollers

João Paulo de Sousa

AVR Family



- Architecture:
  - Harvard - Separate program and data
  - Flash/SRAM/EEPROM
  - Reduced instruction set (RISC)
  - Most instructions are *single clock*

- But what does AVR stands for anyway?
  - Advanced Virtual RISC?...
  - Alf and Vegard RISC?...
  - Nothing at all?...

AVR Family - different market segments

- Tiny AVR
  - Program Memory: 512Byte to 8kByte flash
  - Footprint: 6 to 32 pins
  - Few internal peripherals

- Mega AVR
  - Program Memory: 4 to 256kByte flash
  - Footprint: 28 to 100 pins
  - Many internal peripherals

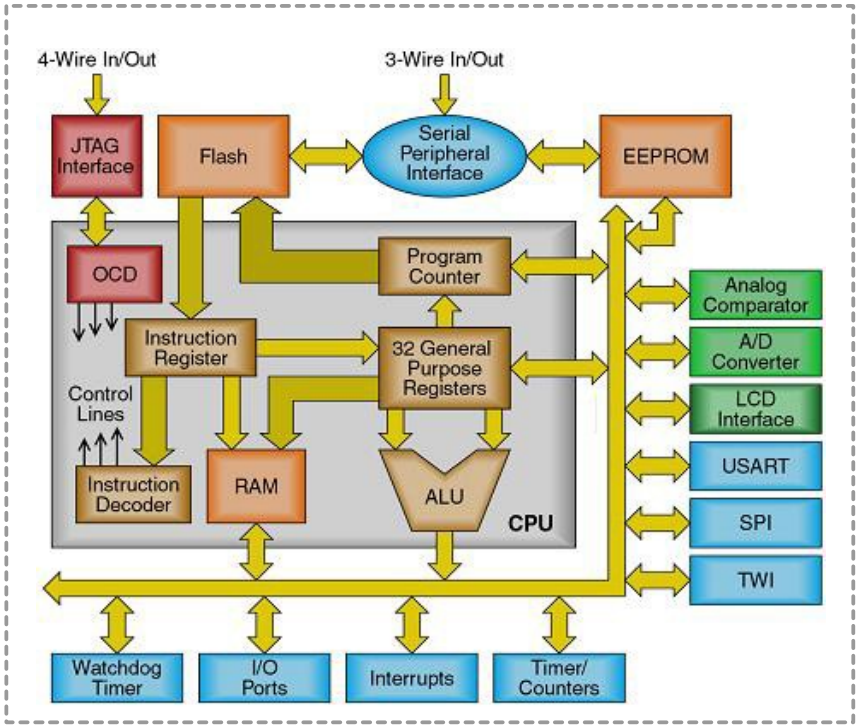
AVR Family - different market segments

- XMega AVR
  - Program Memory: 16 to 384kByte flash
  - Footprint: 32 to 100 pins
  - Many internal peripherals

- Application Specific AVR
  - AVR core
  - Specific application peripherals (USB, CAN,...)

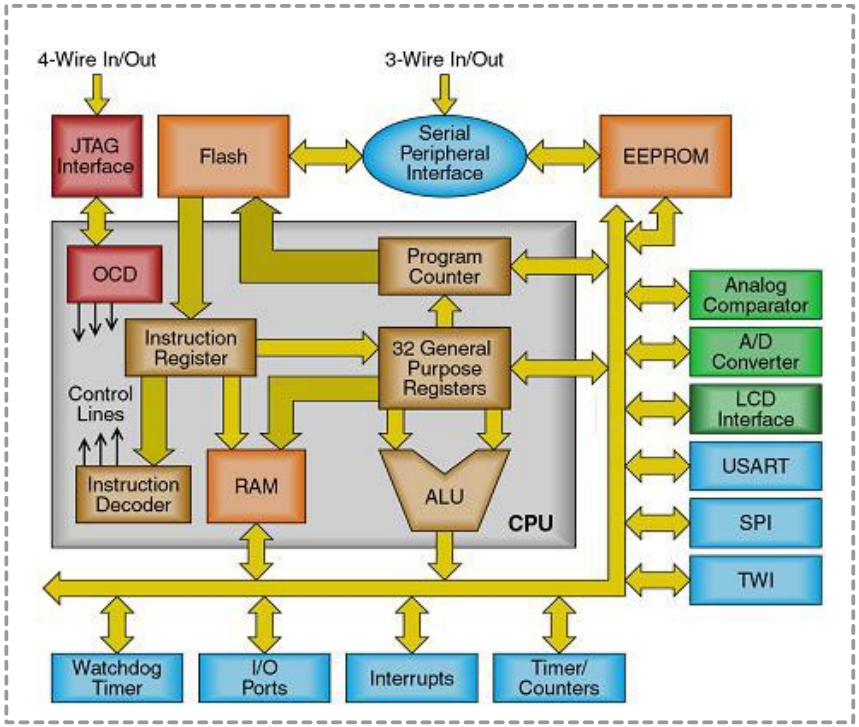
AVR Family - internal architecture

- CPU:
  - 32 working registers
  - Low power
  - RISC
    - 131 instructions
    - Mostly 1cycle
  - Harvard
    - Program memory
    - Data memory



AVR Family - internal architecture

- Peripherals:
  - I/O pins
  - Timers, WDT
  - ADC 10 bits, 6-8 channels
  - Communications:
    - I2C
    - SPI
    - USART



AVR Family - internal architecture

- Registers:
  - R0..R31 (8bits)
    - Z=R31:R30
    - Y=R29:R28
    - X=R27:R26
  - PC (program counter)
  - SP (stack pointer)
  - Many special function registers

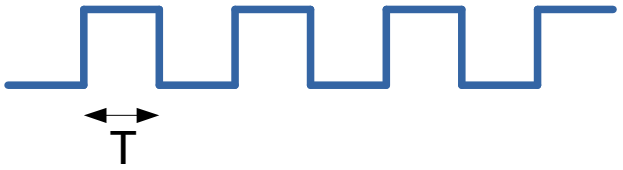
- Instructions:
  - A few examples:

Syntax	Operation	Cycles
LDI Rd,k	Rd ← k	1
IN Rd, A	Rd ← I/O(A)	1
OUT A, Rs	I/O(A) ← Rs	1
LD Rd, Z	Rd ← (Z)	2
ST X, Rs	(X) ← Rs	2
EOR Rd,Rs	Rd ← Rd ⊕ Rs	1
RJUMP k	PC ← PC+k+1	2

AVR Family - Assembly language example

```
;; ------------------------------------
;; sqr.asm
;;   A simple square wave generator
;;   written in assembly language
;;   Created on: 25/08/2018
;;       Author: jpsousa@fe.up.pt
;; ------------------------------------

#include <avr/io.h>   ; register definitions

    .section .text
    .global main

main: ldi r18, 0b100000  ; Set bit 5 of R18

lp:   in   r19, PORTB    ; Read port B (1)
      eor  r19, r18      ; XOR in R19 (1)
      out  PORTB, r19    ; Write to port B (1)
      rjmp lp            ; Infinite loop (2)

;; ------------------------------------
;; Total cycles inside the loop:
;;      1+1+1+2=5

    .end
```

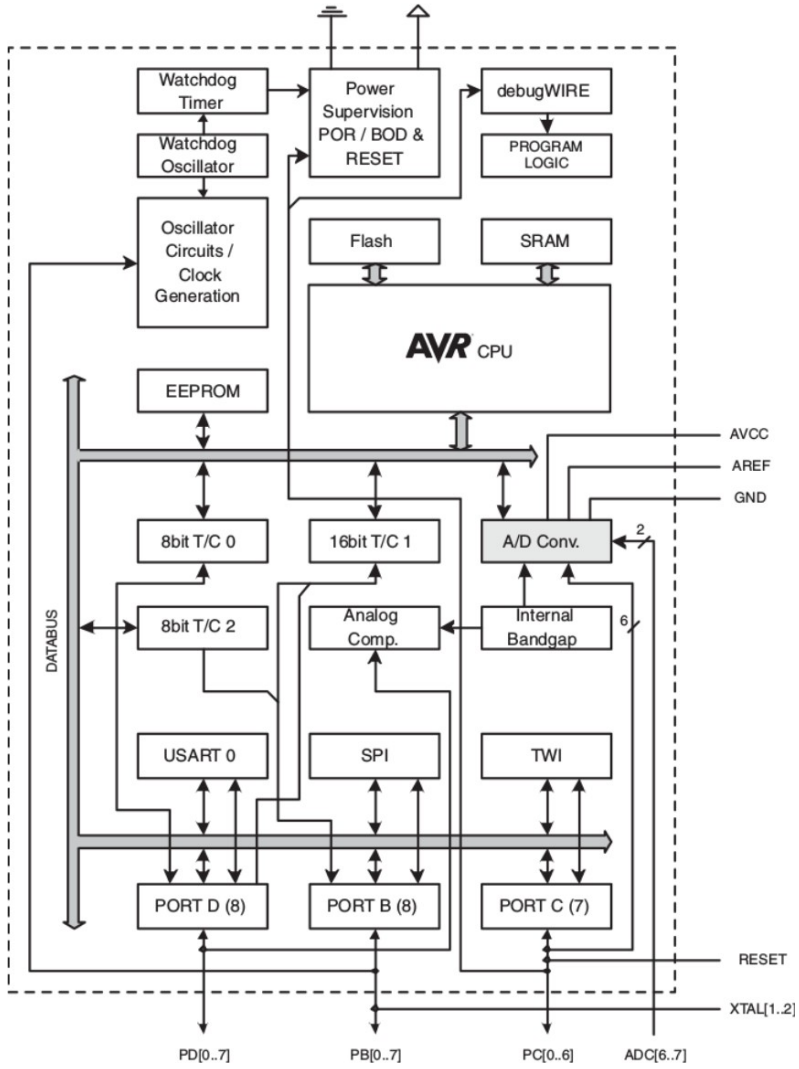
PB5:



- F_{XTAL} = 16MHz
- T = 3,125 uS
  (5 cycles @ 16MHz)
- f = 1/2T = 1.6MHz

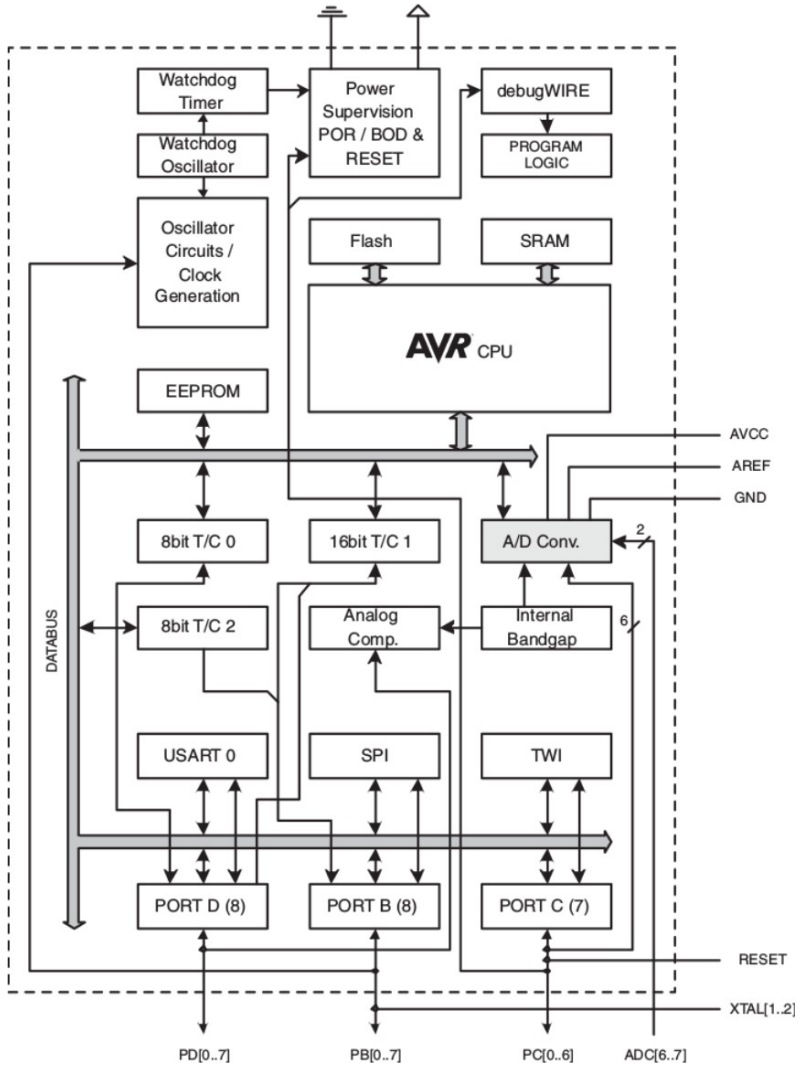
ATmega328P

- Memory (harvard):
  - 32 kB Flash ROM (Program)
  - 2kB SRAM (Data)
  - 1kB EEPROM (NV Data)
- Input/output interfaces:
  - 23 I/O pins,
  - 3 Timers, WDog
  - ADC 10 bits, 6-8 channels
  - Comms: I2C, SPI, USART



ATmega328P

- Voltage: 1.8V to 5.5V
- Frequency:
  - @1.8V: up to 4MHz
  - @2.7V: up to 10MHz
  - @4.5V: up to 20MHz
- Current (1MHz, 1.8V, 25°C):
  - Active: 200uA
  - Power-save: 0,75uA
  - Power-down: 100nA



To further explore...

- Atmel Flash Microcontrollers Portfolio
- The history of the AVR family
- Interview with Vegard Wollan
- The AVR Freaks community
- AVR042: AVR Hardware Design Considerations
  - Providing a robust analog and digital power supply
  - Connecting the reset line