

5197 - Sistema Digitais

Bacharelado de Informática

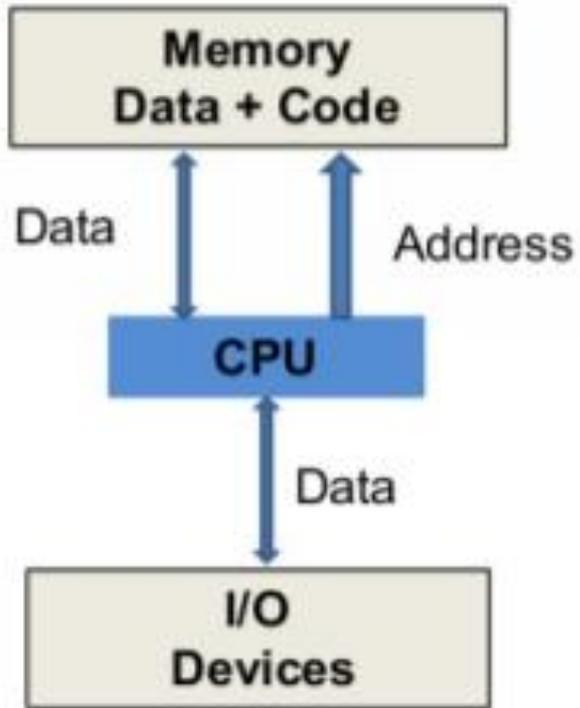
UEM – DIN - Prof. Elvio

v. 17a

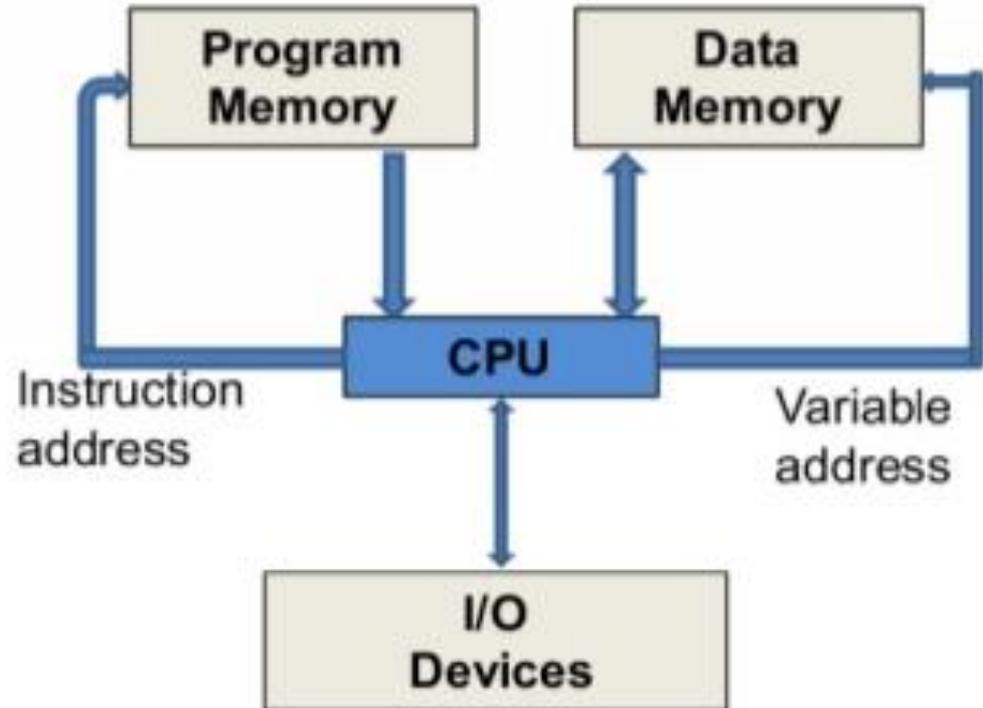
Roteiro

- Informação Preliminar
- Introdução à Microcontroladores

Arquiteturas Harvard e Von Neumann



Von Neumann Machine



Harvard Machine

Arquiteturas Harvard e Von Newmann

- Arquitetura Von Newmann
 - Arquitetura mais simples porque tem apenas um banco de memória
 - Mais lento porque não permite acessos simultâneos
 - Usualmente empregada em processadores CISC (Complex Instruction Set Computer)
- Arquitetura Harvard
 - Arquitetura mais complexa porque divide a área de memória em dois bancos
 - Mais rápido porque permite acessos separados à instruções e dados
 - Usualmente empregada em processadores RISC (Reduced Instruction Set Computer)

Algumas Abreviações

- CISC: Complex Instruction Set Computer
 - Possui um conjunto amplo de instruções
 - Instruções são complexas e podem executar várias tarefas
- RISC: Reduced Instruction Set Computer
 - Possui um conjunto reduzido de instruções
 - Instruções são simples e de mesmo comprimento
- DSP: Digital Signal Processor
 - Processador otimizado a processamento digital de sinais

Microcontrolador

- Microcontrolador (abreviado por μC , uC ou MCU) é um sistema computacional completo em um único encapsulamento
 - Inclui CPU, memória, oscilador de relógio, periféricos e interface de E/S em um único circuito integrado (chip)
- Características:
 - Pequeno, possui toda a memória e E/S de que precisa
 - Não expansível (não possui barramento para interface externa)
 - Baixo custo (a partir de US\$ 1)
 - Baixa velocidade (relógio a partir de 10 kHz até 20 MHz)
 - Baixo consumo de energia, ou extremamente baixo em modo "*sleep*"
 - Arquitetura pequena, comumente de 8 bits, mas chegando até 32 bits
 - Pequeno volume de memória
 - Interface de E/S limitada
- Oferece a vantagem de um projeto programável a um custo e tamanho que sistemas computacionais de propósito geral não conseguem atender
- Utilizado em Sistemas Embarcados (*Embedded Systems*)

Sistema Embarcado

- Sistema que usa um ou mais microcontroladores para executar tarefas bem definidas
 - Usualmente hardware e software são otimizados para a tarefa desejada
 - São diferentes dos computadores de mesa, que não são projetados para executar apenas um tipo de programa
- Alguns exemplos:
 - Sistema de alarme e segurança
 - Controle de velocidade automotivo (piloto automático)
 - Controle de aquecimento ou refrigeração (ar condicionado)
 - Forno de microondas
 - Controlador ABS e anti-derrapagem
 - Controlador de semáforos
 - Máquina de vendas
 - Bomba de gasolina
 - Controlador do sistema de irrigação
 - Mars Rover

Entendendo as Diferenças

- Abordagem para uso geral: máquina é a parte principal de um sistema que executa aplicações que podem ser instaladas ou desenvolvidas pelo usuário; **usuário pode alterar a programação**
- Abordagem para uso embarcado: máquina é apenas um elemento de um sistema inteligente maior; **usuário não pode alterar a programação**
- Características de arquitetura e organização diferentes podem tornar microprocessadores/microcontroladores mais adequados à determinada aplicação
- A vantagem ou desvantagem de um determinado microprocessador/microcontrolador só deve ser avaliada no contexto da aplicação à que se destina

Entendendo as Diferenças

- Aplicações de propósito geral exigem μP com as seguintes características:
 - Suporte a sistemas operacionais que implementam “*time-sharing*”
 - Suporte a memória virtual, com “*cache*” multicamada e RAM dinâmica
 - Suporte a E/S controlada por DMA
 - Conjunto grande de registradores
 - Matemática de ponto flutuante implementada em hardware
 - Arquitetura com múltiplos núcleos
 - Visão geral é que interrupções são “irritações (chamadas de “exceções”)

Entendendo as Diferenças

- Aplicações embarcadas exigem μC com as seguintes características:
 - Estrutura flexível para interrupções (interrupções são fundamentais para sistemas baseados em eventos)
 - Mudança rápida de contexto (geralmente implica em conjunto pequeno de registradores)
 - Mistura de E/S digital e analógico (para facilitar a interface com uma grande variedade de dispositivos externos)
 - Possibilidade de “remendos” (patching) em linguagem Assembly (para segmentos de código no caminho crítico)

Programação Embarcada

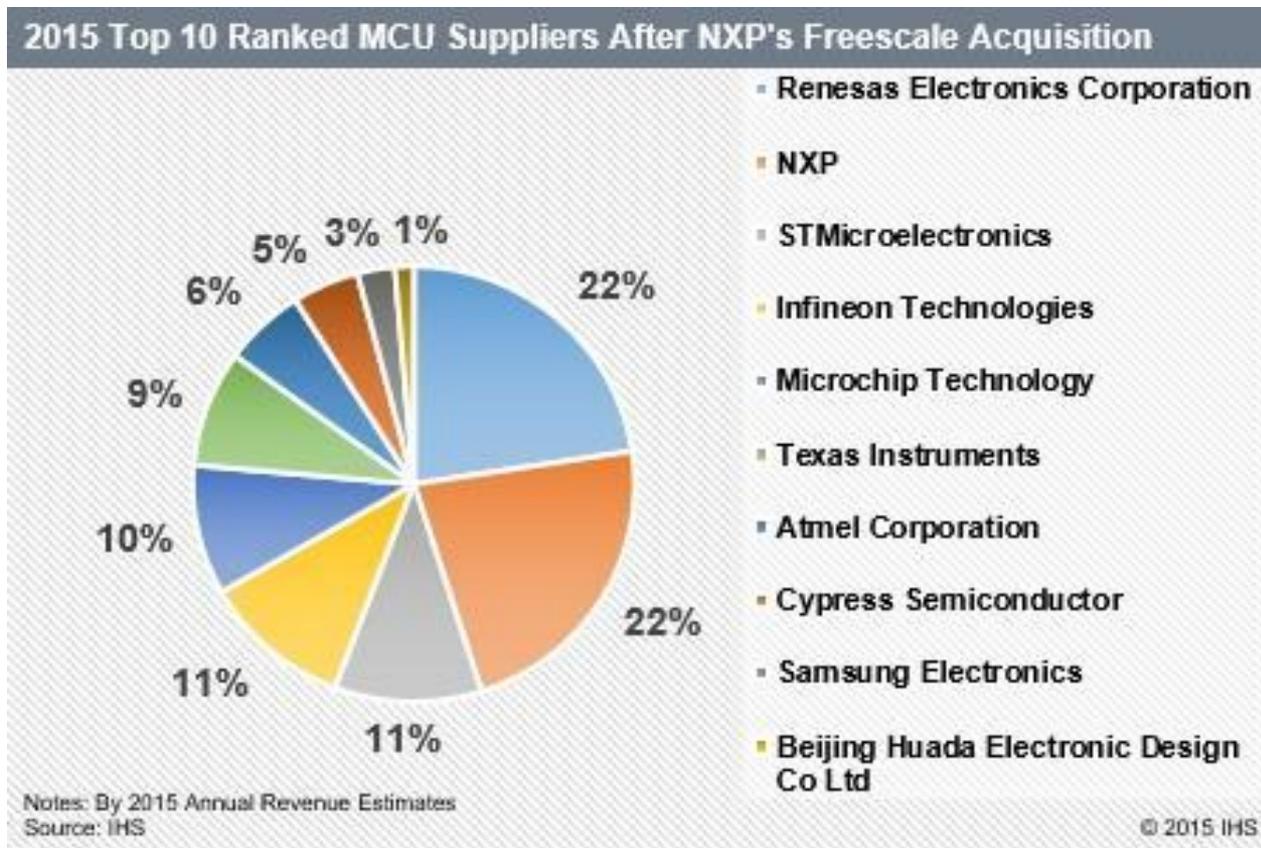
- Programas devem trabalhar próximos e em colaboração com hardware (muitas vezes hardware específico e dedicado)
- Ao contrário de computadores de mesa:
 - Hardware diverso
 - Grande oferta de sistemas operacionais
- Programador deve
 - conhecer como programar e utilizar diversos dispositivos e interfaces
 - conhecer e usar equipamentos de depuração e teste, como multímetros, osciloscópios, analisadores lógicos e similares
- Sistemas embarcados são
 - limitados em termos de memória e capacidade de processamento
 - (muitas vezes) sensíveis ao consumo de energia
 - (muitas vezes) sistemas de tempo real

Linguagem de Programação

- Duas linguagens dominam programação embarcada: C e Assembly
- A mais utilizada: C
 - Linguagem estruturada mas com grande flexibilidade
- Aquela em extinção: Assembly
 - Cada vez menos utilizada mas, ainda assim, importante
- Outras:
 - C++
 - Pode ser interpretada (traduzida) para C
 - Utilizada em projetos maiores, onde programação orientada a objetos pode ser uma vantagem
 - Java
 - Também orientada a objetos, mas que precisa maior capacidade de processamento
 - Utilizada em sistemas de 32 bits
 - JavaScript e Python
 - Fácil aprendizagem
 - Outras linguagens desenvolvidas especificamente para sistemas embarcados

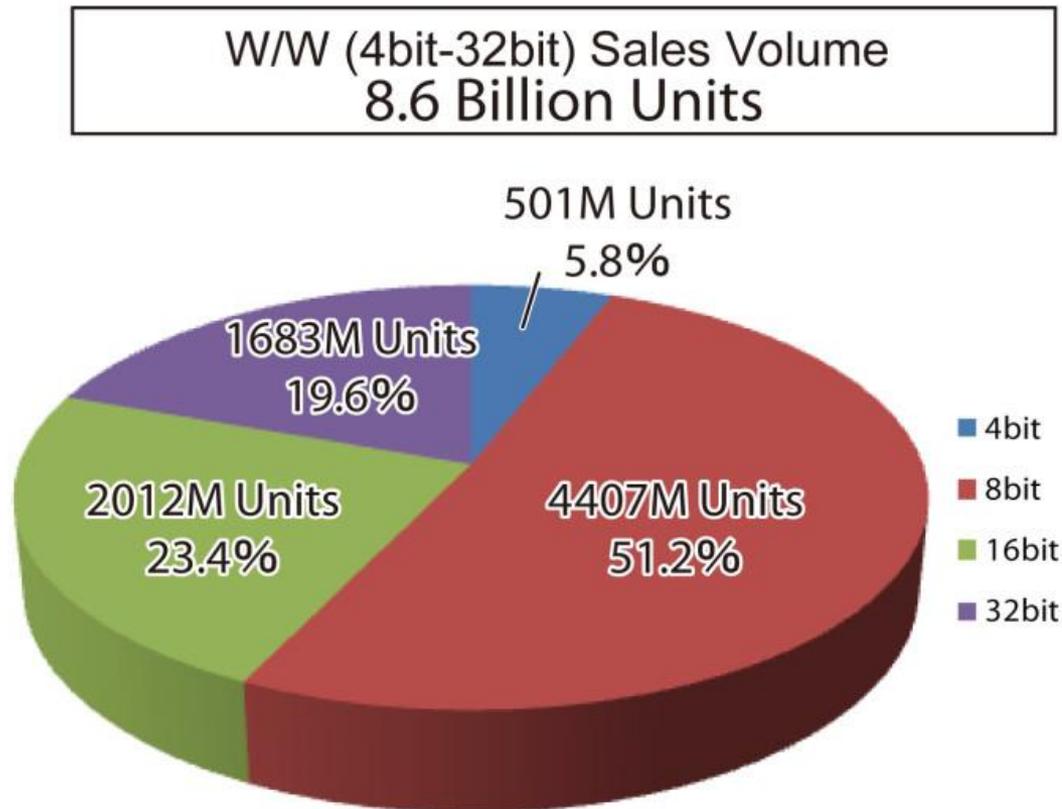
Mercado

- Mercado por vendas (2015)



Mercado

- Mercado por largura de barramento (2013)

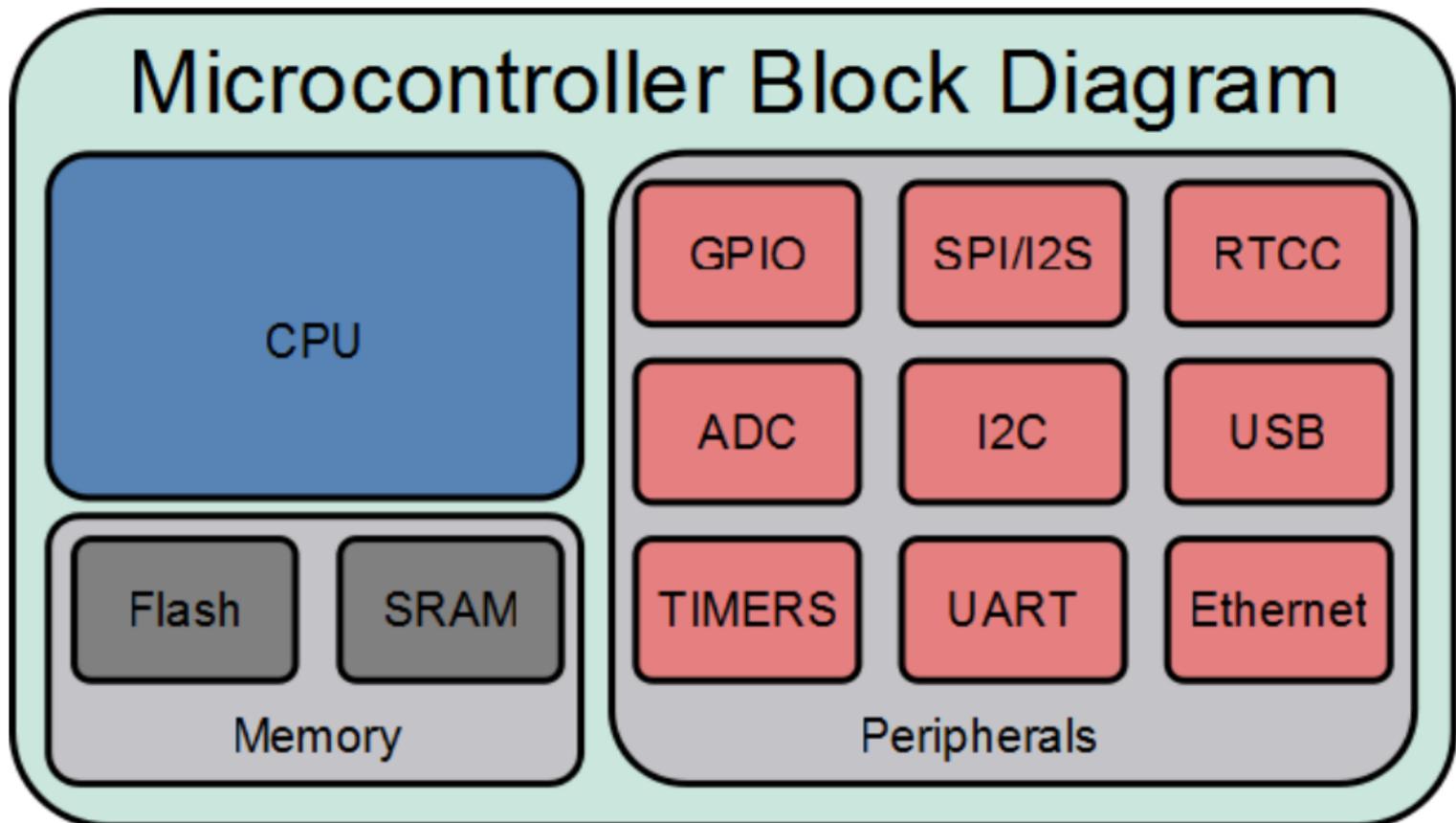


Microcontroladores

- CPU, memória flash e RAM
- Periféricos:
 - GPIO (General Purpose I/O): controla portas de E/S
 - RTCC (Real Time Calendar/Clock) e Timers: temporizadores e base de tempo
 - ADC (Analog Digital Converter): conversor A/D
 - SPI (Serial Peripheral Interface), I2S (Inter IC Sound), I2C (Inter-Integrated Circuit), UART (Universal Asynchronous Receiver/Transmitter) e USB (Universal Serial Bus): interfaces seriais
 - Ethernet: interface de rede (também serial)

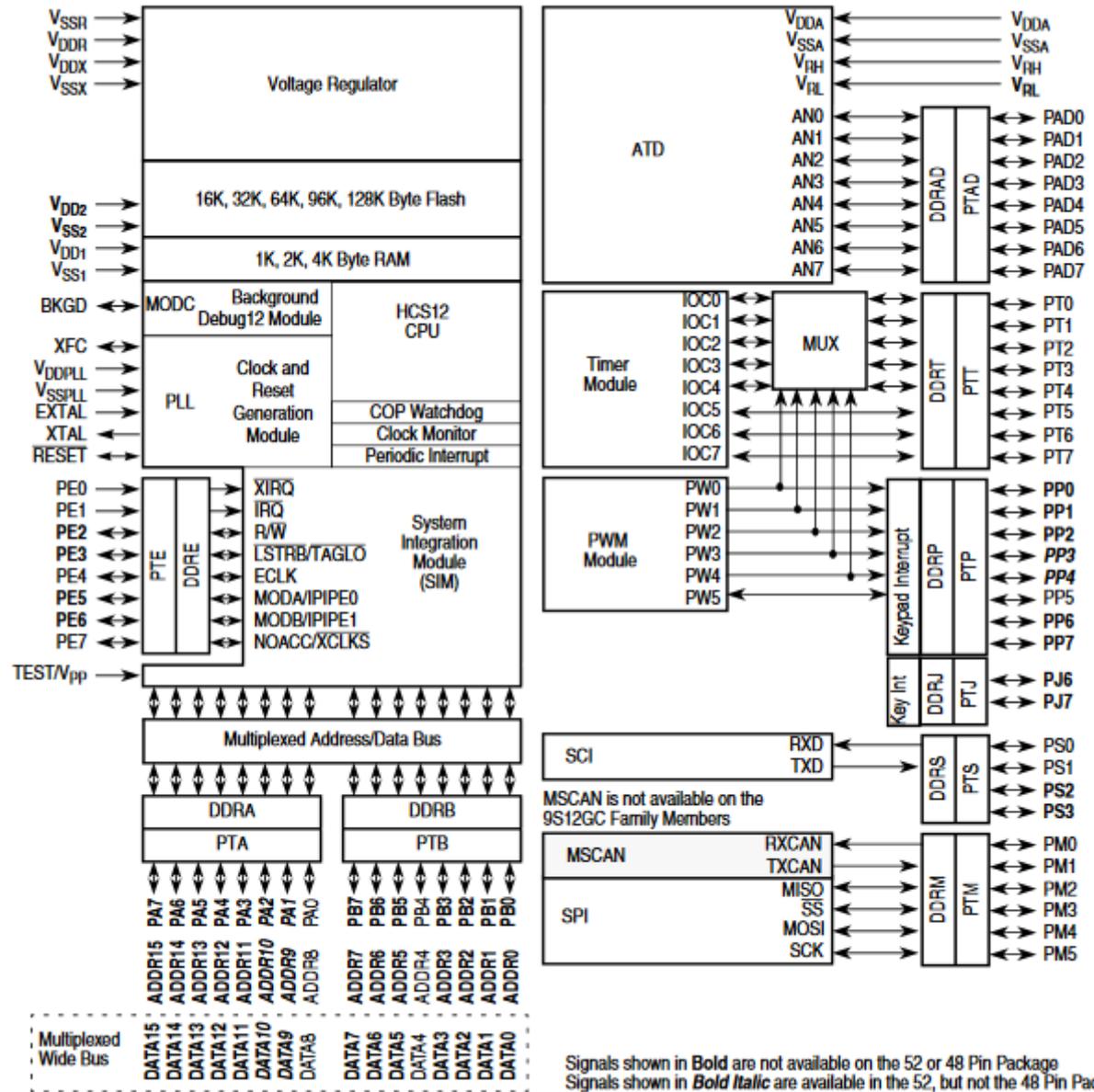
Introdução à Microcontroladores

- Diagrama em blocos (geral)



Exemplos

- MC9S12C128 (Motorola, depois Freescale, depois NXP)

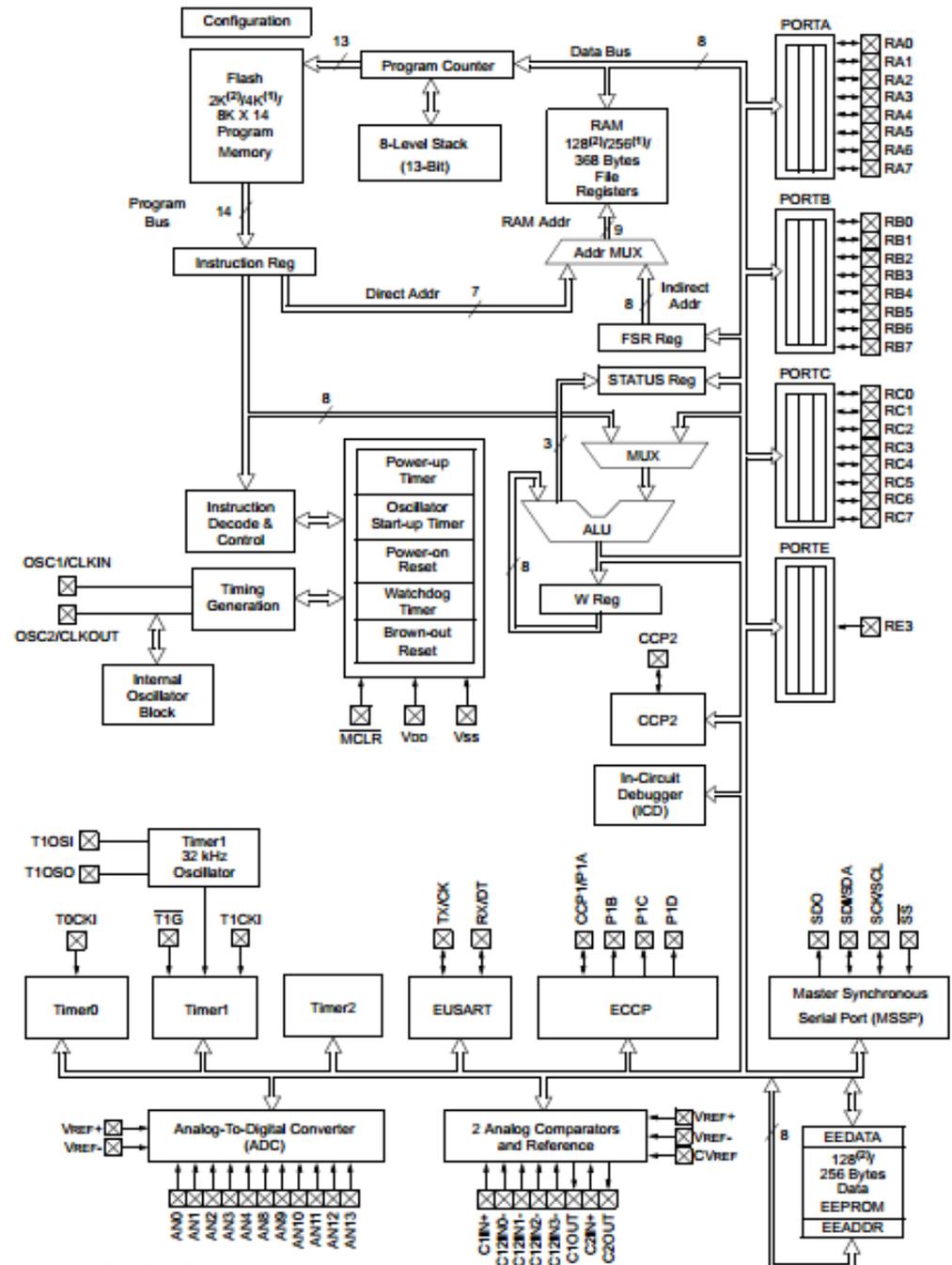


Exemplos: MC9S12C128

- Núcleo HCS12 (16 bits)
- Até 12 opções para interrupção de “wake-up”
- Opções de memória: 16 KB até 128 KB flash, 1 KB até 4 KB RAM
- Conversor A/D com 10 bits de resolução
- Interface CAN (Controller Area Network) versão 2.0 A de 1 Mbps
- Módulo temporizador, com 8 canais
- Módulo PWM (Pulse Width Modulator), de 8 ou 16 bits
- Interfaces seriais: SCI (Serial Communications Interface) assíncrona e SPI (Serial Peripheral Interface) síncrona
- Módulo gerador de relógio e “reset”, com “watchdog”, interrupção de tempo real, monitor de relógio, relógio de referência entre 0,5 MHz a 16 MHz, com possibilidade de sincronismo com fonte externa
- Regulador interno de tensão de 2,5V, permite alimentação com tensões entre 2,97 V e 5,5 V
- Apoio a depuração com o BDM (Background Debug Mode) que permite execução com informação de depuração enviada por interface dedicada
- Encapsulamentos de 48 pinos, 52 pinos ou 80 pinos

Exemplos

- PIC16F882
(Microchip)



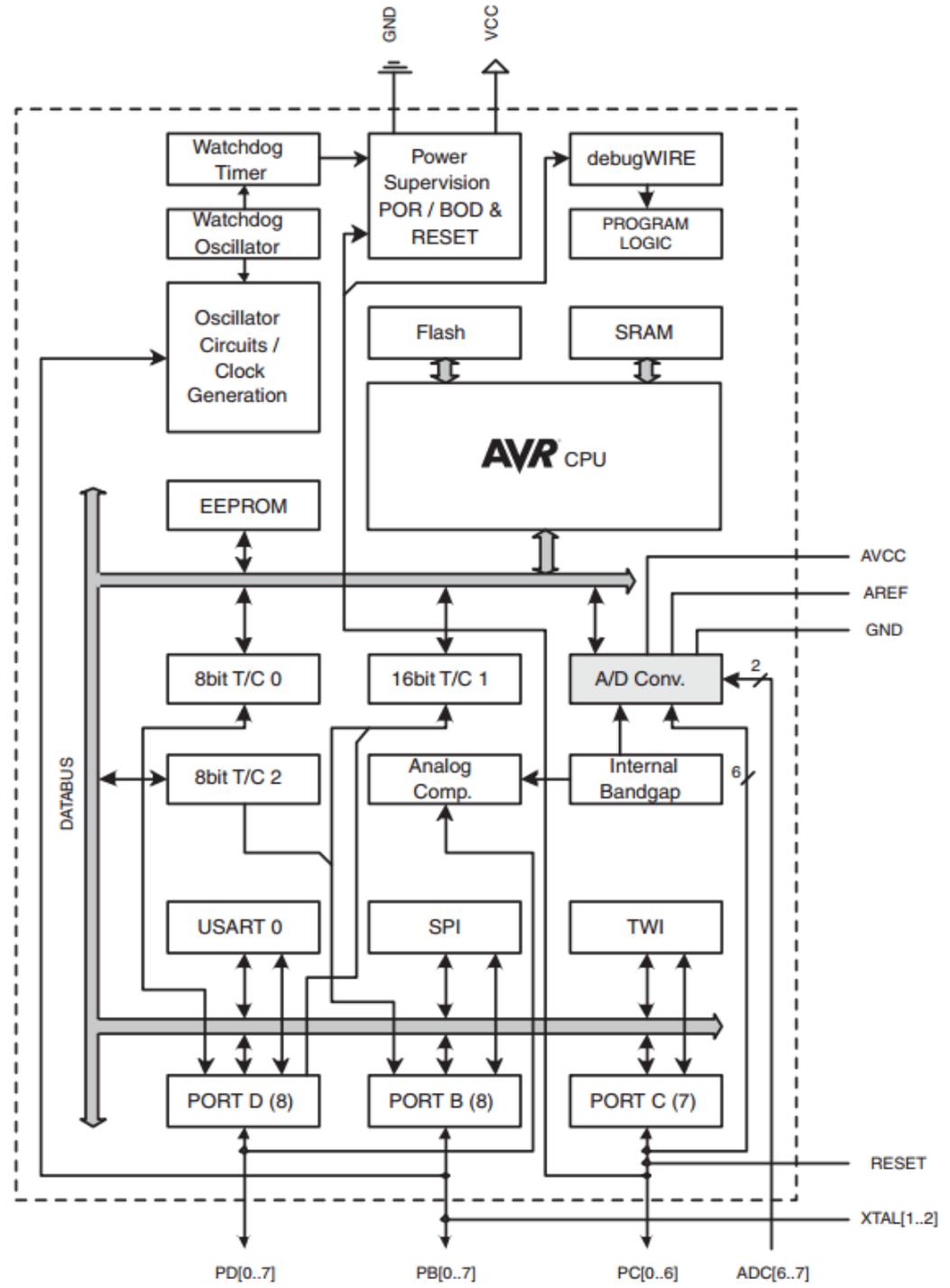
Note 1: PIC16F883 only.
 Note 2: PIC16F882 only.

Exemplos: PIC16F882

- Apenas 35 instruções (arquitetura RISC)
- Relógio programável até 20 MHz com oscilador interno
- Modos “sleep” de economia de energia
- Tensão de alimentação entre 2 V e 5,5 V
- “Power-on Reset”
- Proteção do código
- Módulo de depuração
- Até 35 pinos de E/S com alta corrente
- Comparador analógico
- Conversor A/D com 10 bits de resolução e até 14 canais
- Dois temporizadores (8 e 16 bits)
- Módulo PWM (Pulse Width Modulator)
- Módulo USART (Universal Synchronous/Asynchronous Receiver/Transmitter) com RS-485, RS-232, and LIN (Local Interconnect Network) 2.0, SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit)

Exemplos

- ATmega328 (Atmel)



Exemplos: ATmega328

- Núcleo Atmel AVR de 8 bits
- 131 instruções (arquitetura RISC)
- 32 registradores de 8 bits
- Opções de memória: 4 KB a 32 KB de flash; 0,4 KB a 1 KB de EEPROM e 0,5 KB a 2 KB de RAM
- Proteção de código
- Três temporizadores (2 de 8 bits e um de 16 bits)
- Módulo PWM (Pulse Width Modulator) com 6 canais
- Conversor A/D com 10 bits de precisão e até 8 canais
- Módulos USART (Universal Synchronous/Asynchronous Receiver/Transmitter), SPI (Serial Peripheral Interface) e I2C (Inter-Integrated Circuit)
- Temporizador de “watchdog”
- Interrupção e “wake-up” dependente de entrada
- “Power-on reset”
- Interrupções por fontes externas e internas
- Encapsulamentos de 28 pinos e 32 pinos
- Tensão de operação de 1,8 V a 5,5 V

Exemplos



Microchip PIC32MZ



Freescale (NXP) KL03



Atmel Atmega328P