

Arquitetura de Computadores

Introdução

(slides adaptados do professor Tiago Dias)

Bib: A – secção 6.4.6; B – secções 1.1, 1.2 e 1.3

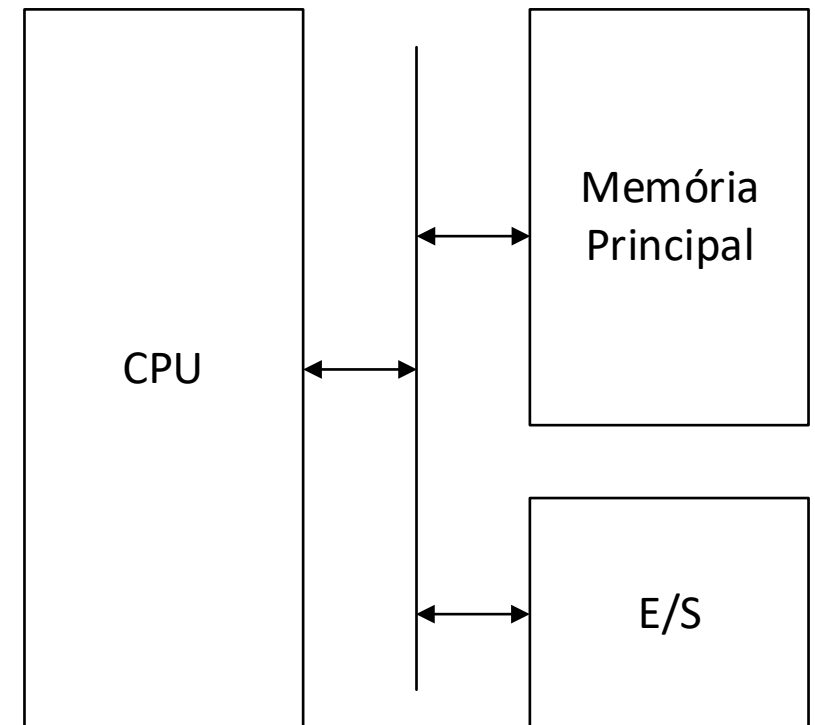
João Pedro Patriarca (jpatri@cc.isel.ipl.pt), Gabinete F.0.23 do edifício F
ISEL, ADEETC, LEIC

Projeto de sistema digitais

- Desenho específico para resolver um problema
 - Recursos concretos para implementar uma determinada funcionalidade
 - Sistema de controlo para orquestrar os módulos funcionais
- Exemplo do sistema multiplicador
 - Registo para guardar o valor do multiplicando
 - Contador para decrementar o multiplicador
 - Registo acumulador para guardar o resultado de sucessivas adições
 - Somador para adicionar o valor do multiplicando com o valor acumulado
 - Sistema de controlo para controlar os dispositivos anteriores

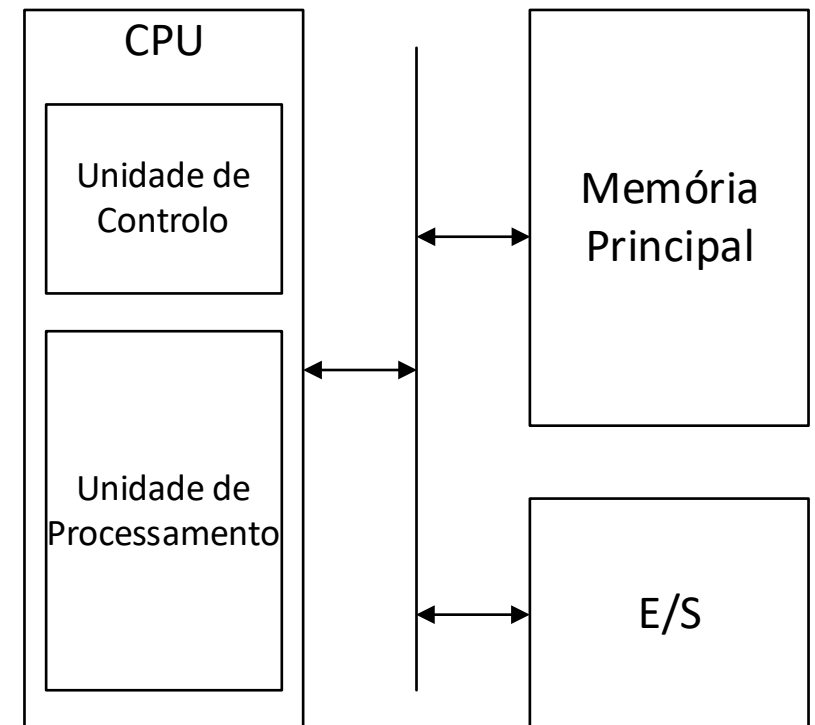
Sistema computacional

- Sistema baseado em CPU (*Central Processing Unit*) capaz de executar diversos algoritmos e processamento de dados
- Na sua essência, é constituído por três subsistemas:
 1. Processador ou CPU
 2. Subsistema de memória principal
 3. Subsistema de entrada e de saída de dados (E/S)



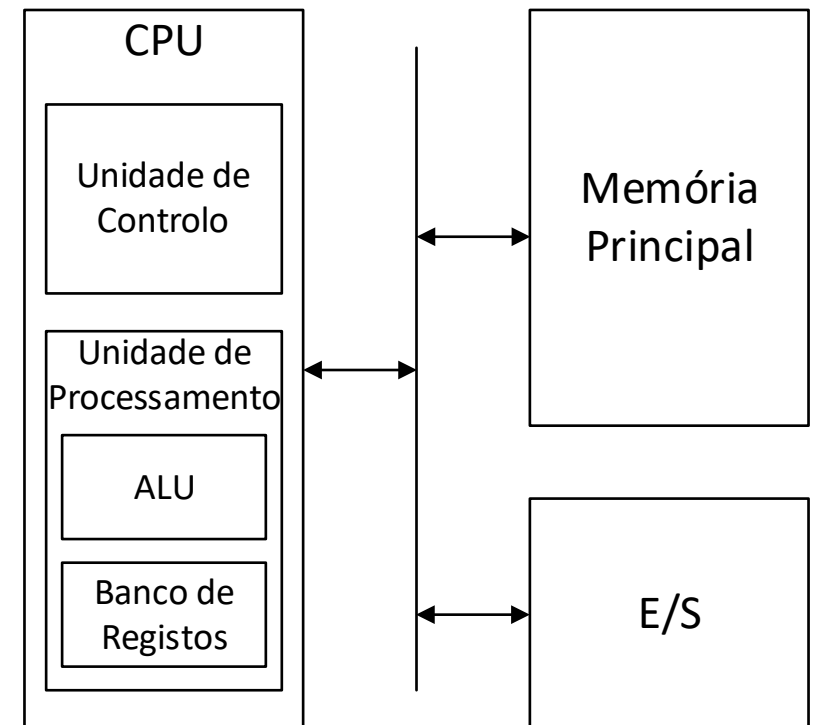
Processador

- O processador executa instruções guardadas em memória
- A sua estrutura interna é constituída por duas unidades principais:
 - Unidade de processamento
 - Unidade de controlo



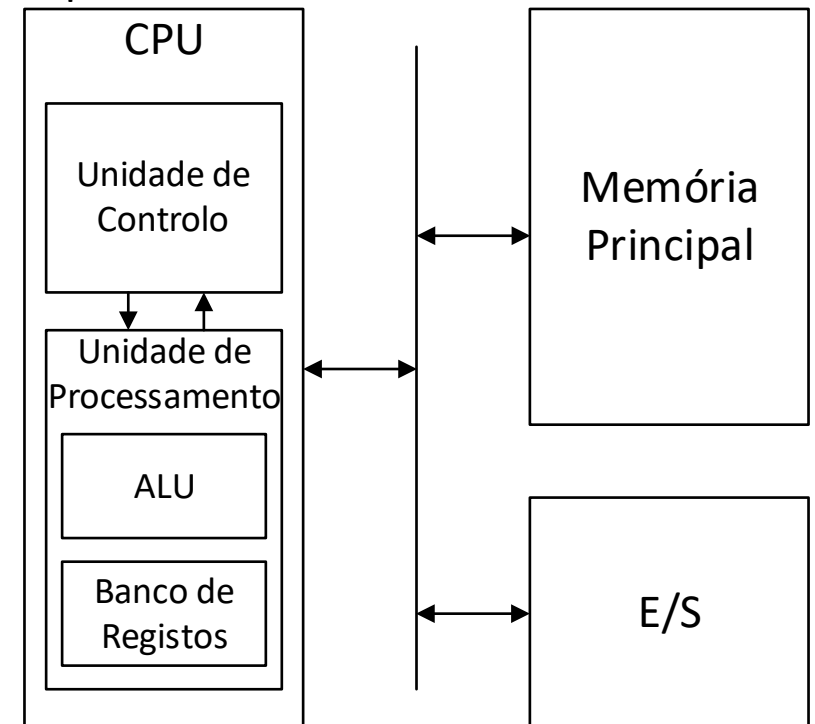
Unidade de processamento

- A unidade de processamento é constituída, essencialmente, por uma unidade aritmética e lógica (ALU – *Aritmetic and Logic Unit*) e por um banco de registos
- Unidade aritmética e lógica
 - Determina a capacidade de processamento de dados
 - Diverge de CPU para CPU e depende do objetivo da arquitetura
- Banco de registos (*Register File*)
 - Servem para guardar operandos e resultados produzidos pela ALU
 - Por norma, estão organizados num bloco e cada registo é identificado por um número
 - Tipicamente, o número de registos correspondem a uma potência de 2



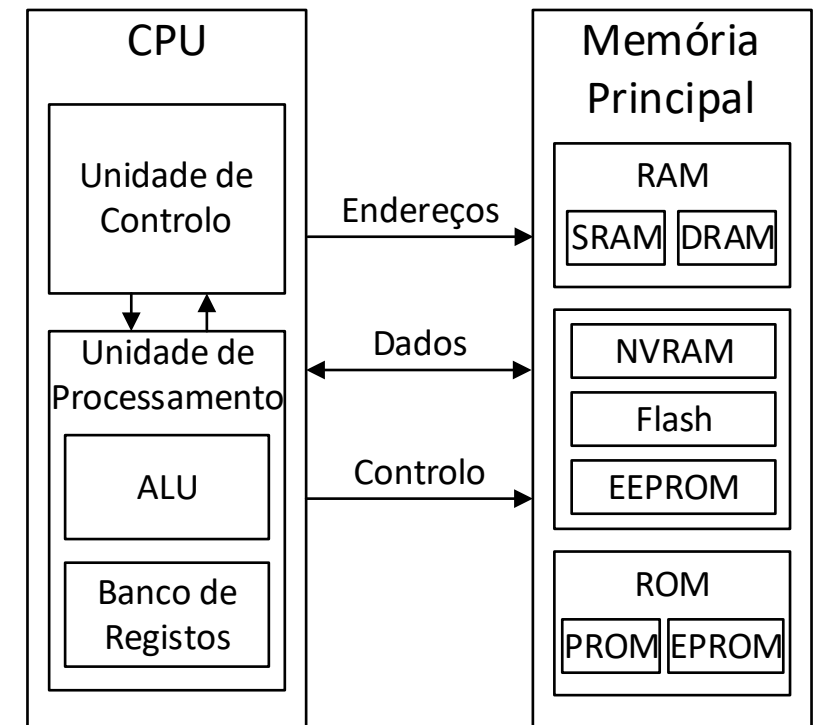
Unidade de controlo

- A unidade de controlo é responsável pelo funcionamento do processador:
 - Interação com memória e periféricos
 - Com base no código de uma instrução, gera a palavra de controlo que define a funcionalidade pretendida
 - Avalia *flags* produzidas pela ALU
- A palavra de controlo define, entre outros aspetos:
 - Operações a realizar pela ALU
 - Registos e constantes que servem como operandos na ALU
 - Destino do resultado produzido pela ALU
 - Sinais de encaminhamento dos dados
 - ...



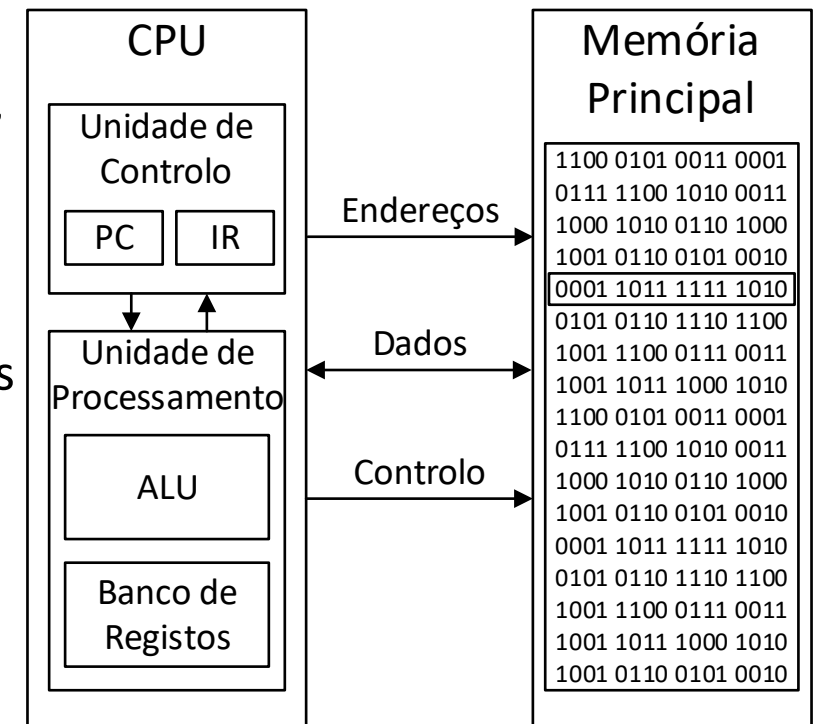
Subsistema memória

- A arquitetura do sistema computacional define a dimensão do seu espaço de endereçamento
- Nesse espaço de endereçamento podem ser mapeados diferentes dispositivos de memória, inclusive com características diferentes (volátil, não volátil, etc.)
- São usados três tipos de barramentos para interação com memórias
 - Barramento de endereços
 - Barramento de dados
 - Barramento de controle



Programa

- Um programa é representado por uma sequência de instruções em memória
- Cada instrução é constituída por um conjunto de bits, cuja codificação permite ao CPU definir a operação a realizar
- A execução de um programa corresponde ao processamento sequencial de instruções
- Na execução de uma instrução estão envolvidas, tipicamente, três fases
 - Obtenção da instrução da memória (*fetch*)
 - Decodificação da instrução (*decode*)
 - Execução da instrução (*execute*)
- Para este processamento cíclico de instruções são necessários os registos PC (*Program Counter*) e IR (*Instruction Register*)
- O registo PC guarda o endereço de memória da próxima instrução a executar
- O registo IR guarda o código da instrução que está atualmente a ser executada
- A execução de um novo programa corresponde a carregar o código do novo programa em memória (não envolve alterações do sistema computacional)



Instrução

- O código de uma instrução corresponde a uma sequência de bits (código máquina)
- É inadequado escrever programas em código máquina
- A linguagem *assembly* oferece um maior nível de abstração ao programador mas não dispensa o conhecimento da arquitetura
- Uma instrução *assembly* corresponde a uma instrução código máquina representada por uma mnemónica e operandos
 - A mnemónica identifica o tipo de instrução
 - Os operandos dependem do tipo de instrução
- Por exemplo, a instrução *assembly* SUB R0, R1, R2 realiza a subtração entre dois registos e guarda o resultado num terceiro registo
- O tradutor de *assembly* traduz um programa escrito em linguagem *assembly* para código máquina

QUIZ 1 - Sistema computacional

Access code: FR79K