

Microprocessadores

Memórias

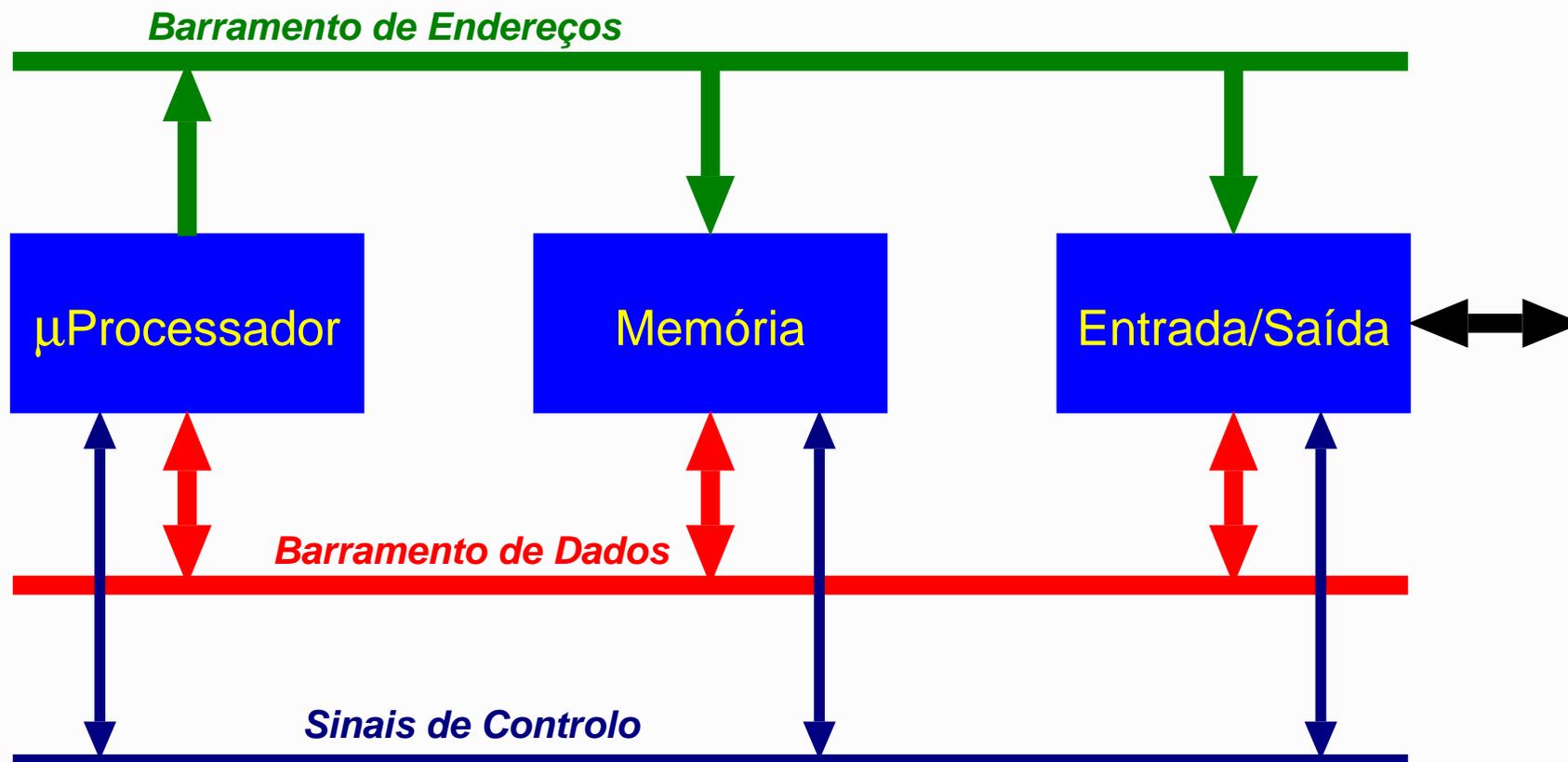
António M. Gonçalves Pinheiro

Departamento de Física
Universidade da Beira Interior
Covilhã - Portugal

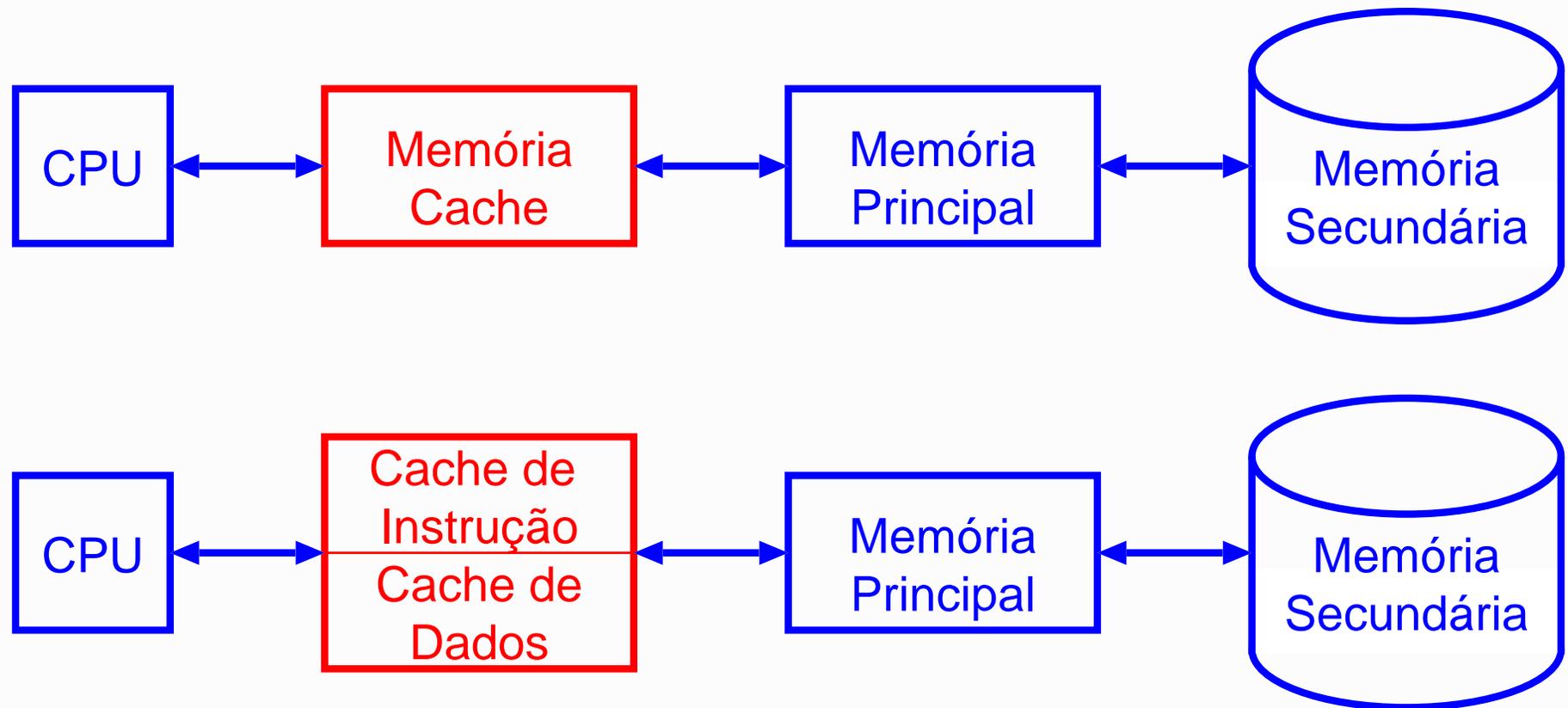
pinheiro@ubi.pt

Arquitetura de Microcomputador

Modelo de Von Neumann



Hierarquia de Memória



Hierarquia de Memória

- ***Memória Cache***

Memória que funciona à velocidade do μ Processador (ou a velocidades aproximadas).

Normalmente está incluído no mesmo CI do próprio microprocessador.

A sua função consiste em ter cópias das posições da memória principal mais utilizadas, para que o seu acesso seja mais rápido.

- ***Memória Principal***

Composta pela ROM e pela RAM do sistema.

Nela são colocados os programas que se pretende serem executados pelo sistema e respectivos dados necessários.

A ROM inclui os programas de inicialização do sistema.

A RAM é a memória volátil do sistema.

Esta é a memória incluída no modelo de Von Newman.

- ***Memória Secundária***

Memória de massa, normalmente constituída pelo disco, e também por CD's, "disketes" e outros tipos de memória removível.

Contem programas para execução e dados do sistema.

Pode ser também utilizada para paginação quando a Memória Principal não tem dimensão suficiente.

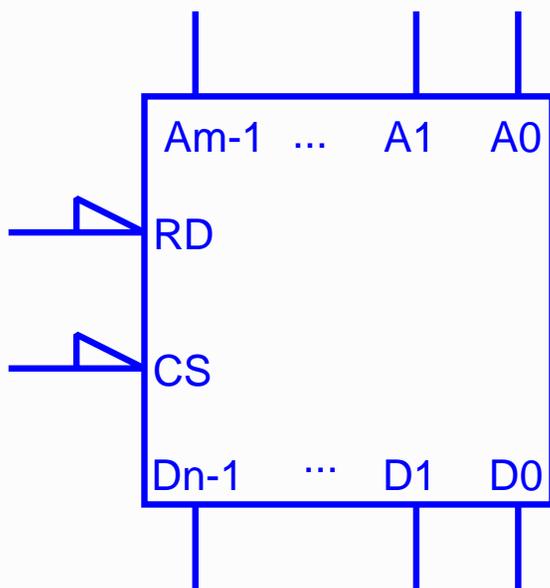
De qualquer forma o acesso a essa memória é muito mais lento, e é feito por operações de Entrada/Saída, já que se encontra exterior ao modelo de Von Newman.



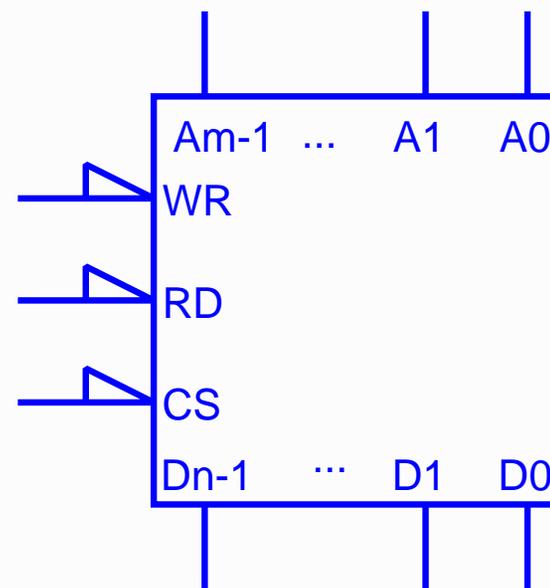
Circuitos de Memória

Tipos de Memória

ROM - Read Only Memory



RAM - Random Access Memory



- **n** - Dimensão da palavra (n bits).
- **m** - Dimensão da memória (2^m palavras de n bits).

Circuitos de Memória

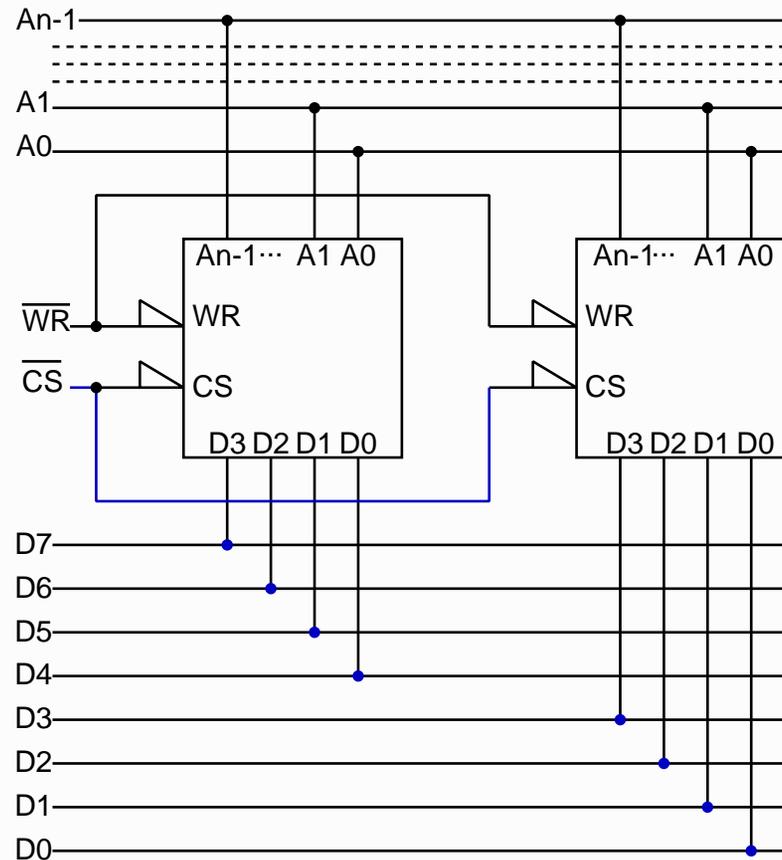
Ligação de Memórias em Paralelo

- Aumentar a Dimensão da Palavra
- Aumentar o Número de Palavras



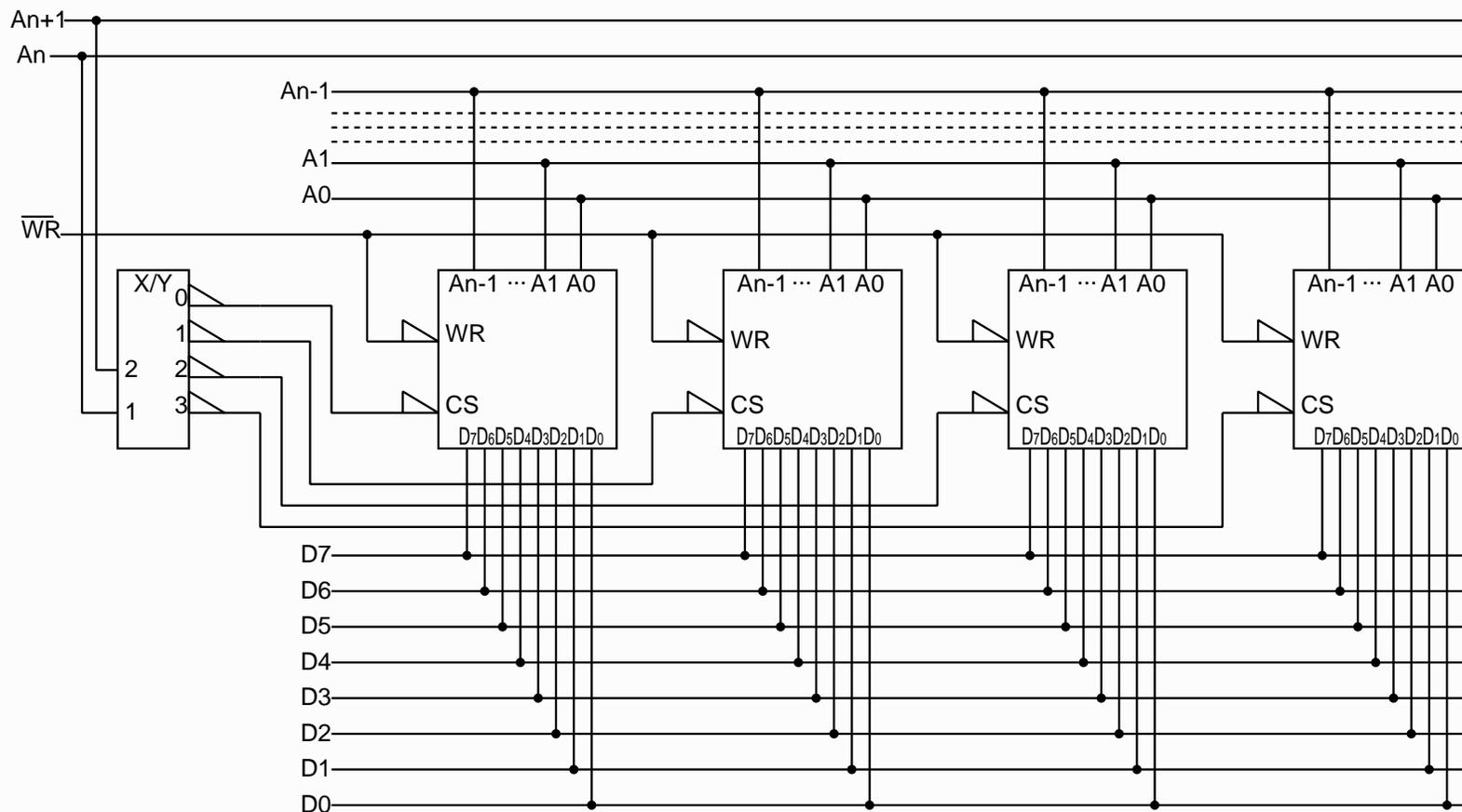
Circuitos de Memória

Aumentar a Dimensão da Palavra (4→8 bits)



Circuitos de Memória

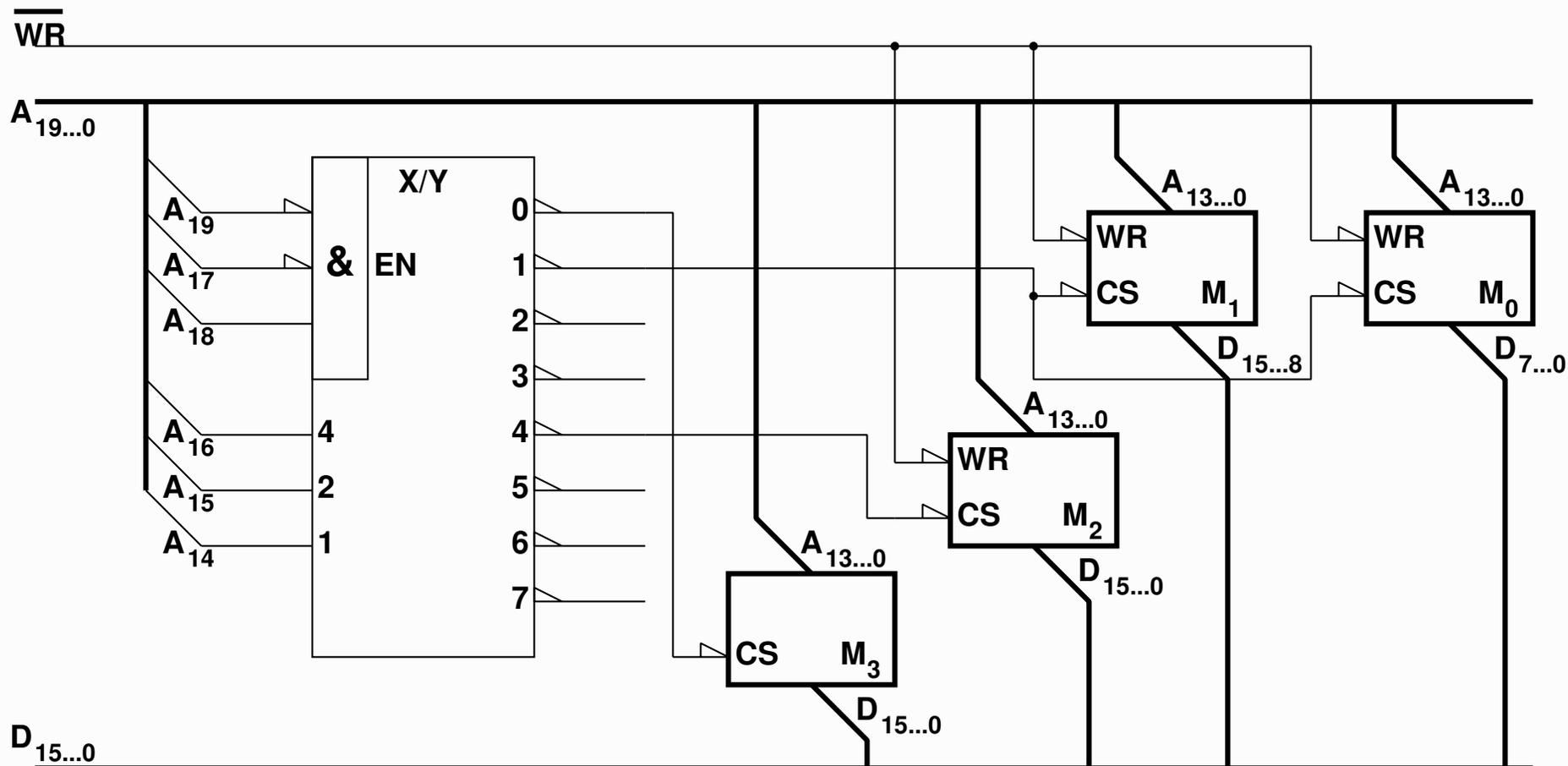
Aumentar o Número de Palavras ($2^n \rightarrow 2^{n+2}$ palavras)



Circuitos de Memória

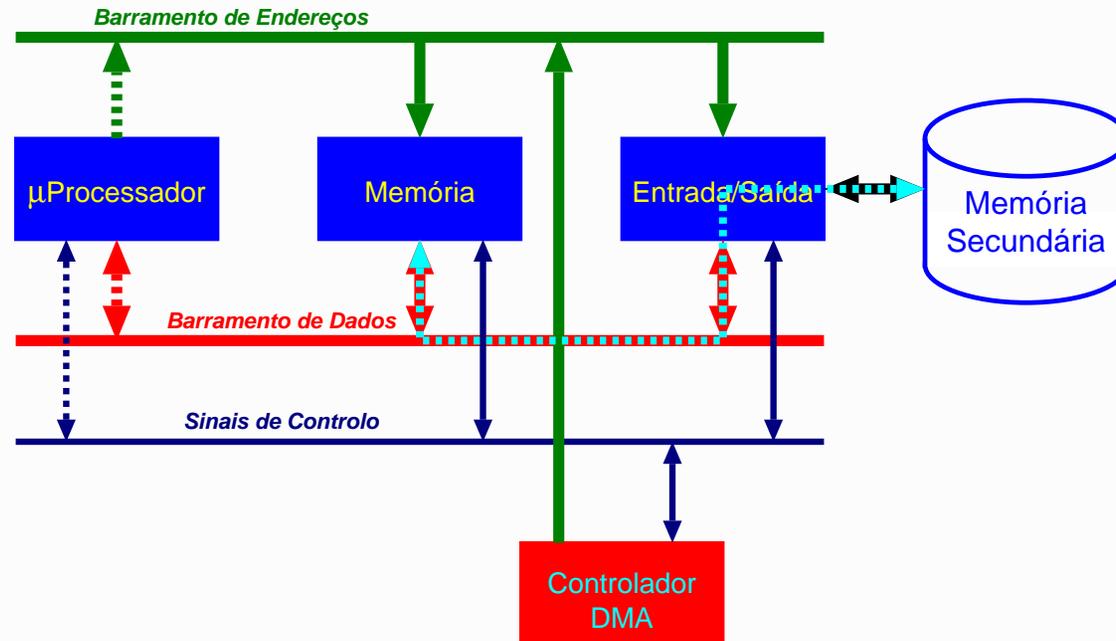
Exemplo de Ligação em Paralelo

Quais os endereços ocupados pelos circuitos de memória?



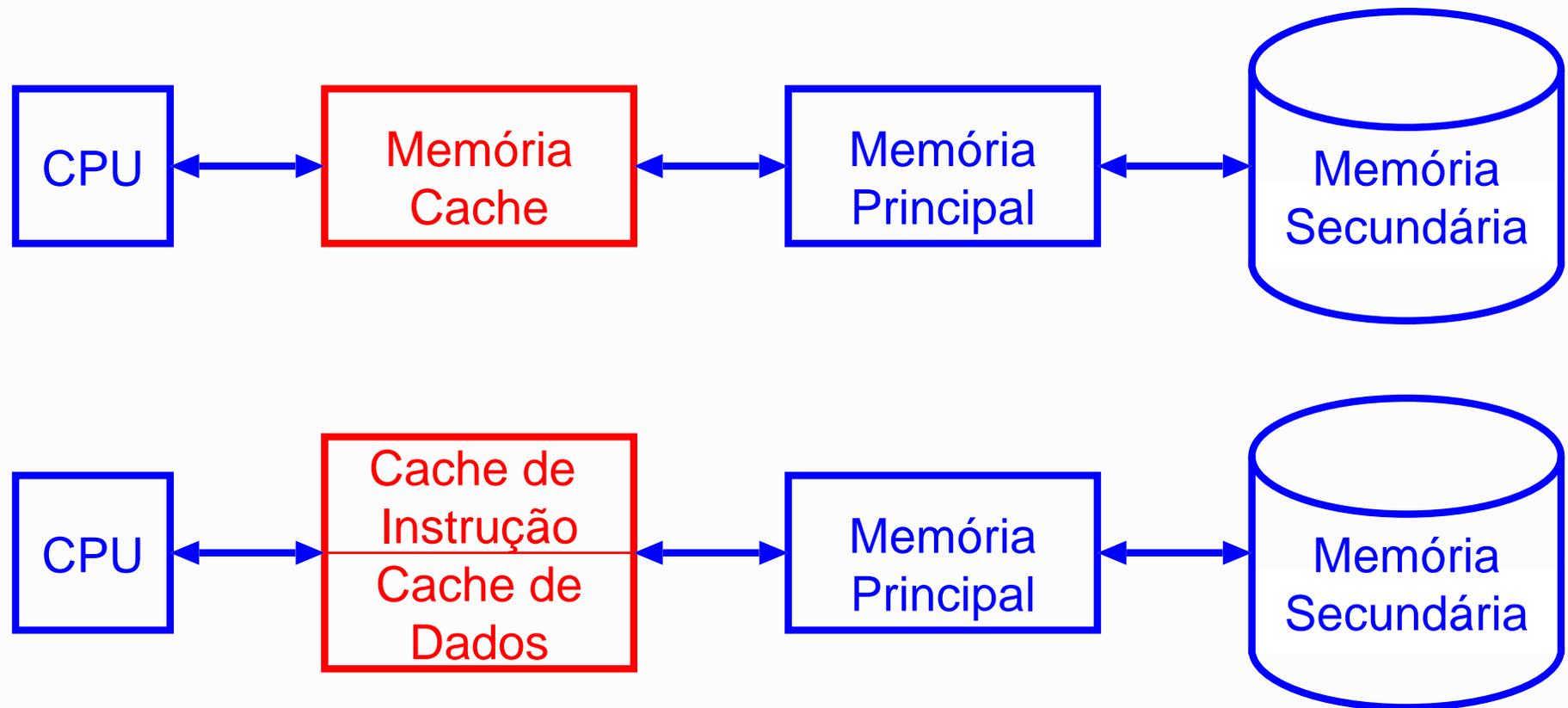
Acesso Directo à Memória

DMA- “Direct Memory Access”



- O controlador DMA sinaliza ao Microprocessador que pretende fazer transferência.
- O Microprocessador coloca os barramentos em estado de alta impedância e sinaliza ao Controlador DMA que pode assumir o controlo dos barramentos.
- O controlador DMA assume o controlo dos barramentos e inicia o processo de transferência.

Hierarquia de Memória



Hierarquia de Memória

Definições

↪ Localização Temporal

Se uma localização de dados é referenciada, então tende a ser referenciada num futuro muito próximo.

↪ Localização Espacial

Se uma localização de dados é referenciada, então localizações com endereços próximos tendem a ser referenciada num futuro muito próximo.

NOTA: *Ex. de Biblioteca:*

Um estudante escolhe um livro de uma biblioteca - tende a consultá-lo mais que uma vez

Um estudante escolhe um livro de uma biblioteca - tende a escolher outros livros da mesma estante

NOTA: *Ex. de Programa de Computador:*

Localização Temporal - Os programas são tipicamente constituídos por Ciclos e Funções que tendem a ser acedidas múltiplas vezes.

Localização Espacial - Os programas, sendo estruturas sequenciais, tendem a usar instruções que estão em posições de memória consecutivas.

Memória Cache

Leitura da memória cache

↪ Cada instrução origina pelo menos um acesso à memória (ciclo de busca)

↪ μ Processadores muito mais rápidos do que memórias

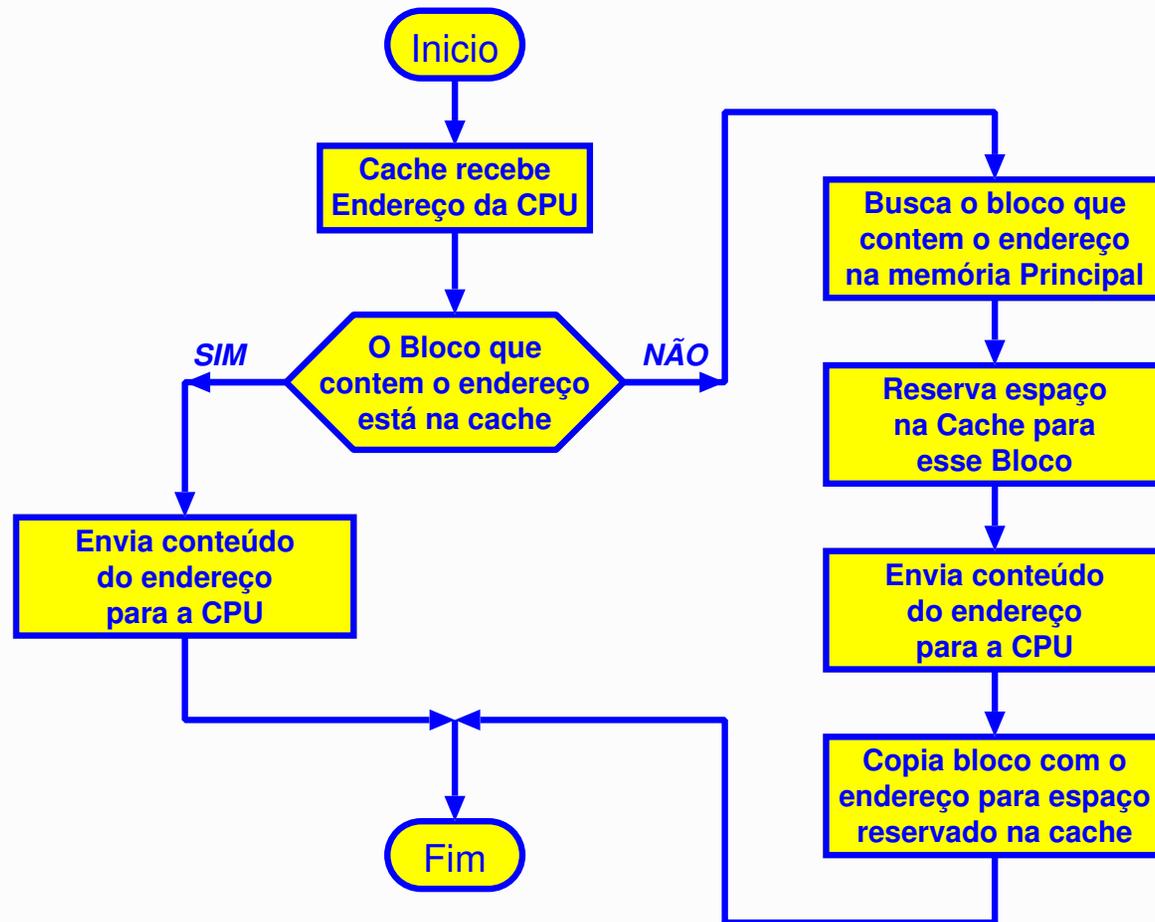
Memória "Cache"

- memória relativamente pequena;
- memória rápida (velocidades similares ao μ Processador);
- coloca-se entre μ Processador e a memória principal (normalmente no CI do μ Processador).

Notas:

- "Cache hit": Quando a palavra de memória está na cache.
- "Cache miss": Quando a palavra de memória não está na cache.

Memória Cache



Memória Cache

Técnicas de mapeamento da memória

- Estas técnicas definem a forma como se faz o mapeamento da memória principal na memória cache, de forma a ser eficientemente acedida.
- As posições de memória escolhidas e que são colocadas na cache, são aquelas que mais vezes são acedidas.
- No entanto tem que haver uma forma eficiente de identificar se uma determinada posição de memória está colocada na cache ou não.

Nota: A memória cache, além de armazenar um conjunto de palavras da memória que em princípio devem ser as mais utilizadas, tem que armazenar o endereço dessas palavras na memória principal.

Memória Cache

Cache mapeada directamente

Endereço(*bloco*) de memória dividido em:

- TAG : parte mais significativa
- INDEX : parte menos significativa

$$\boxed{\text{ENDEREÇO}} \equiv \boxed{\text{TAG}} \boxed{\text{INDEX}}$$

INDEX é fixo

TAG varia em função da palavra escolhida para a cache

Exemplo

(sistema de memória com 256 palavras e cache de 8 palavras):

Para cada INDEX existem 2^5 possíveis palavras de memória.

A que deve ocupar a cache, deve ser a que é mais utilizada de entre as 2^5 .

Endereço		DADO
TAG	INDEX	
0 0 1 1 0	0 0 0	
0 0 0 0 0	0 0 1	
0 1 0 1 0	0 1 0	
0 0 1 1 0	0 1 1	
1 0 0 0 1	1 0 0	
1 0 0 0 1	1 0 1	
0 0 0 0 0	1 1 0	
0 0 0 0 0	1 1 1	



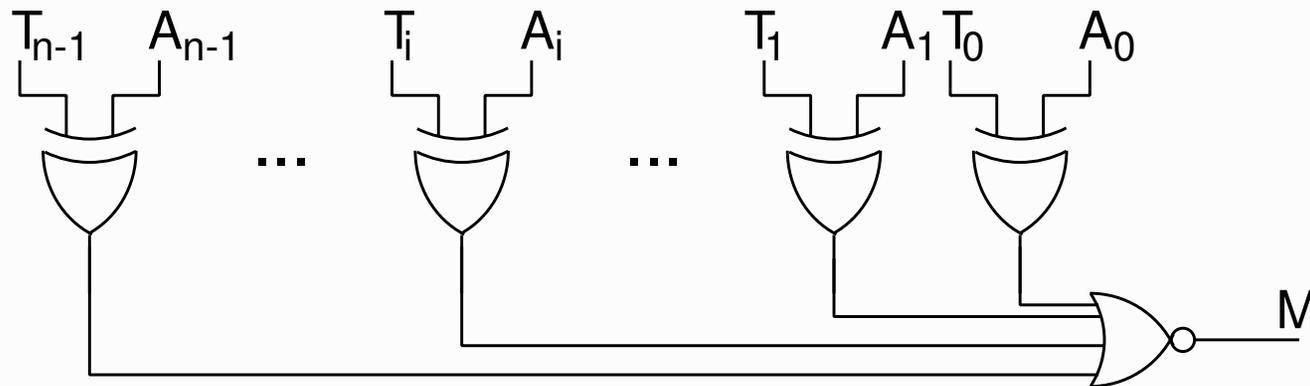
Memória Cache

Cache mapeada associativamente

Não existe INDEX. Logo a TAG contem o endereço(*bloco*) completo.

Então a cache, pode conter em principio, as palavras que são mais utilizadas pelo μ Processador sem qualquer limitação do "hardware".

O problema desta filosofia, está relacionado com a lógica complementar que é necessária à identificação dessa posição de memória.



Este circuito só resulta em 1 quando o endereço(*bloco*) armazenado na TAG e o endereço(*bloco*) requerido à memória está na posição de cache respectiva.

O problema, é que cada endereço(*bloco*) armazenado em cache tem que ter um circuito destes associados, aumentando muito a complexidade.

Memória Cache

Cache mapeada em conjuntos associativos

Combina as duas técnicas prévias.

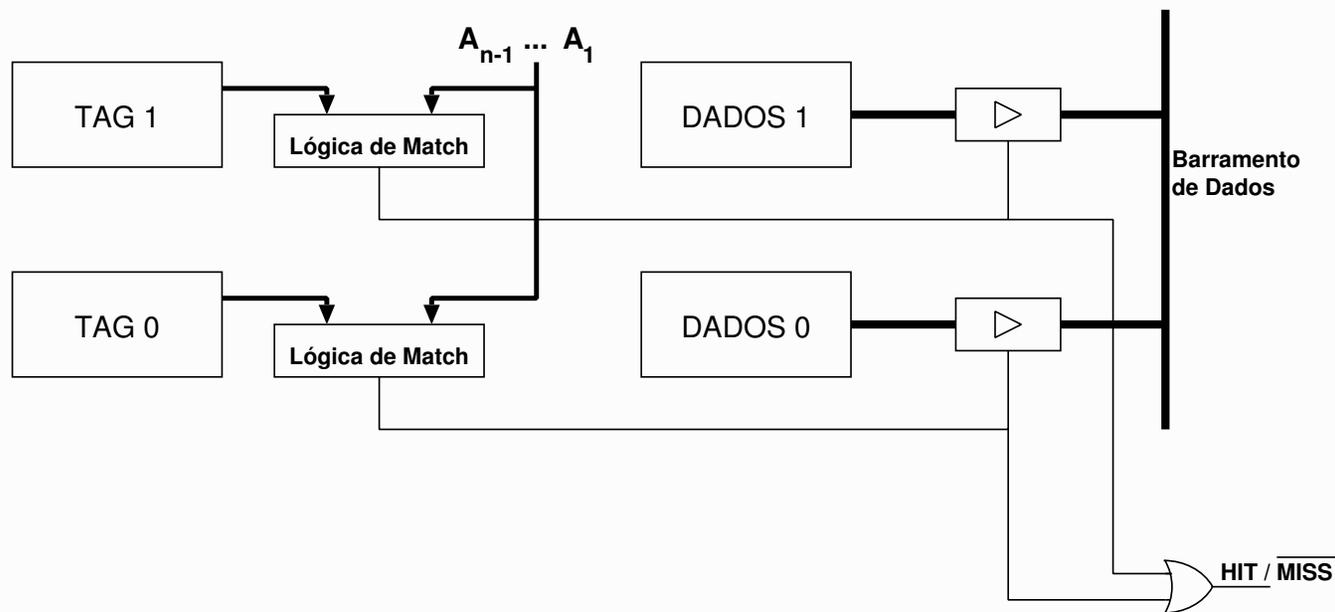
Reduz a complexidade, sem perda de eficiência significativa.

Um endereço (*bloco*) é novamente dividido em TAG e INDEX.

Exemplo:

Circuito com INDEX de 1 bit, A_0 .

$HIT / \overline{MISS} = 1$ significa que a palavra está na cache.



Memória Cache

Substituição do conteúdo da cache

- *Substituição Aleatória*
- *Substituição FIFO*
A posição mais antiga é substituída.
- *Substituição LRU* - Least Recently Used
É considerada a melhor técnica, já que é menos provável voltar a aceder à posição que não é usada à mais tempo.
No entanto, a implementação é complicada, e usualmente usam-se técnicas que a aproximam.
- *Substituição LFU* - Least Frequently Used

Memória Cache

Métodos de Escrita na cache

Quando uma palavra é escrita na memória, e o endereço está na cache, o que é que se deve fazer?

As técnicas são as seguintes:

- *WRITE THROUGH*

Escreve-se na cache e na memória principal.

Como a escrita na memória principal provoca atrasos, usam-se buffers para escrita na memória principal.

Este buffer armazena o endereço e o conteúdo final do dado respectivo, até que seja escrito em definitivo.

- *WRITE BACK*

- A escrita só se faz na cache.

- Cada posição de memória na cache tem um bit extra associado, chamado “Dirty bit” que sinaliza se a respectiva palavra foi ou não alterada.

- Quando as posições de memória em cache são substituídas, e se têm o “Dirty bit” a sinalizar a alteração, são escritas na memória principal.



Memória Virtual

Memória Virtual

Permite que seja usada mais memória do que a presente fisicamente na **memória principal**.

Para isso, usa a memória secundária do sistema (nos sistemas de entrada/saída, como os discos)



Programas são divididos em **PAGEs**

Memória Virtual

PAGE



Memória Cache

Bloco

Page Fault



Memory Miss

Um Page Fault

- Significa que o dado não está na memória principal.
- Leva milhões de ciclos a processar

Endereço Virtual - Endereço de memória virtual que é traduzido (por Software/Hardware) por **Mapeamento de Endereço** para um **Endereço Físico** quando se acede à memória.

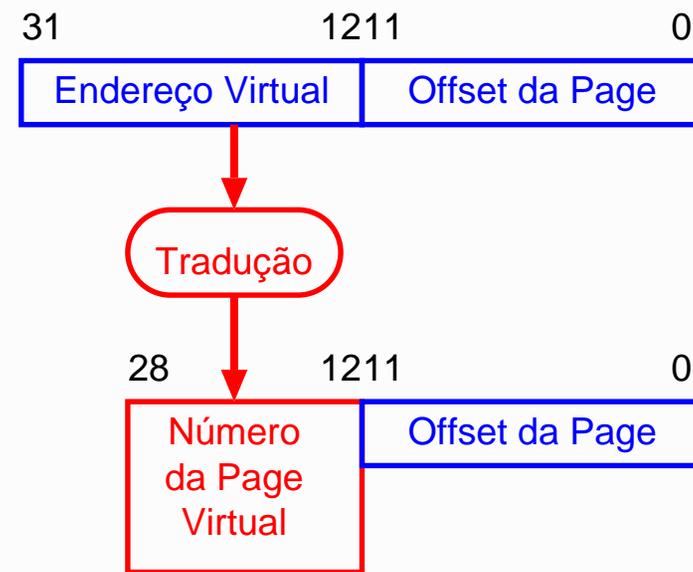
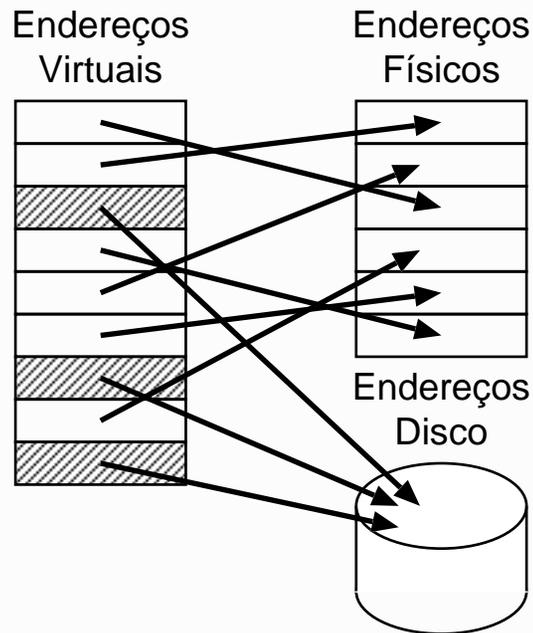
Memória Virtual - Implementa a tradução do espaço de endereçamento do programa para o endereço Físico real.



Memória Virtual

Mapeamento de Endereço - Tradução de Endereço

Processo que mapeia um **Endereço Virtual** no endereço usado para endereçar a memória



Escrita na Memória Virtual - Usado o método **Write-Back**

(Write-Through resultaria em tempos de atraso elevadíssimos, tornando-a impossível)