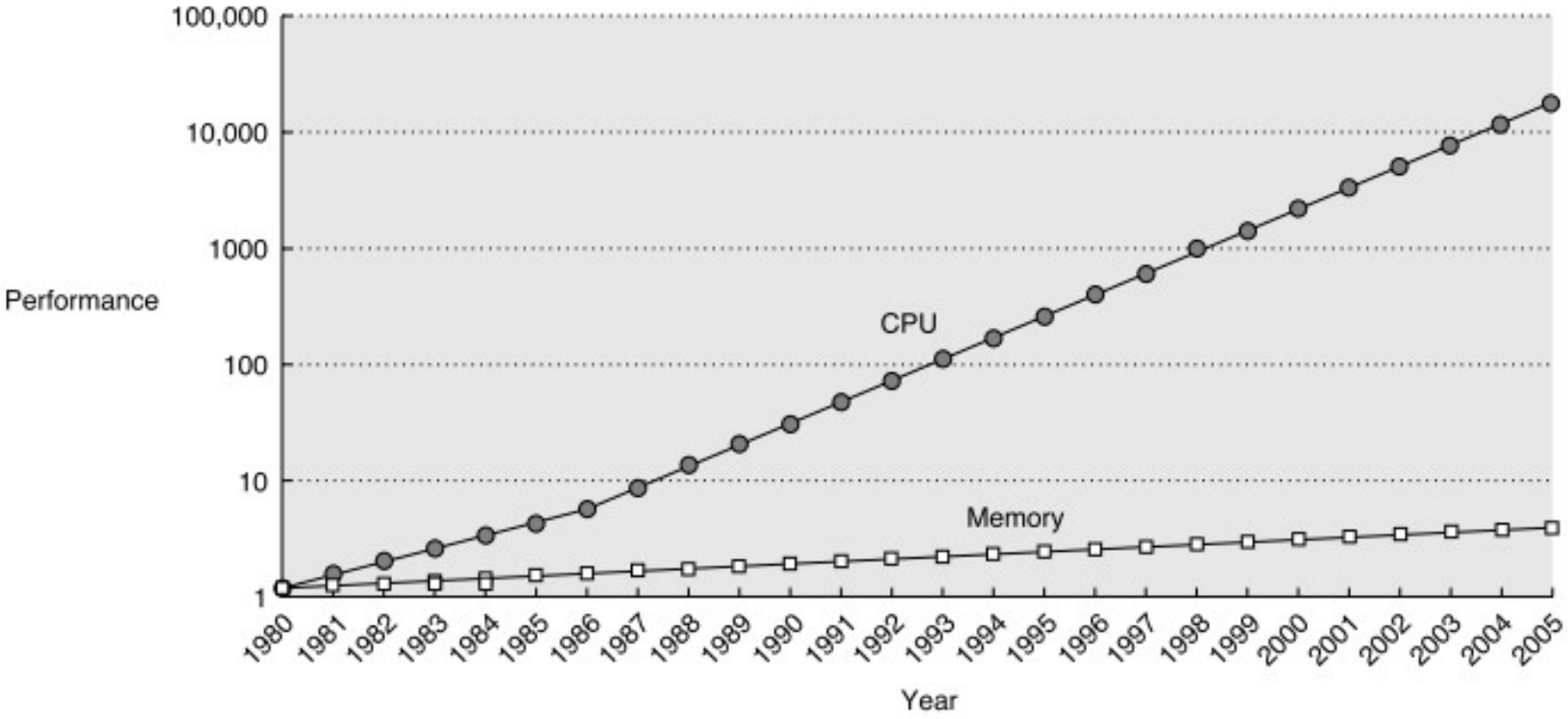
Memória Cache

Prof. Tiago Gonçalves Botelho

Desempenho de CPU *vs.* Memória

Evolução histórica do desempenho de CPU e dos circuitos de memória principal



Desempenho de CPU *vs.* Memória

- Se a CPU fizer uma requisição na memória, ela não obterá a palavra de que necessita por muitos ciclos de clock.

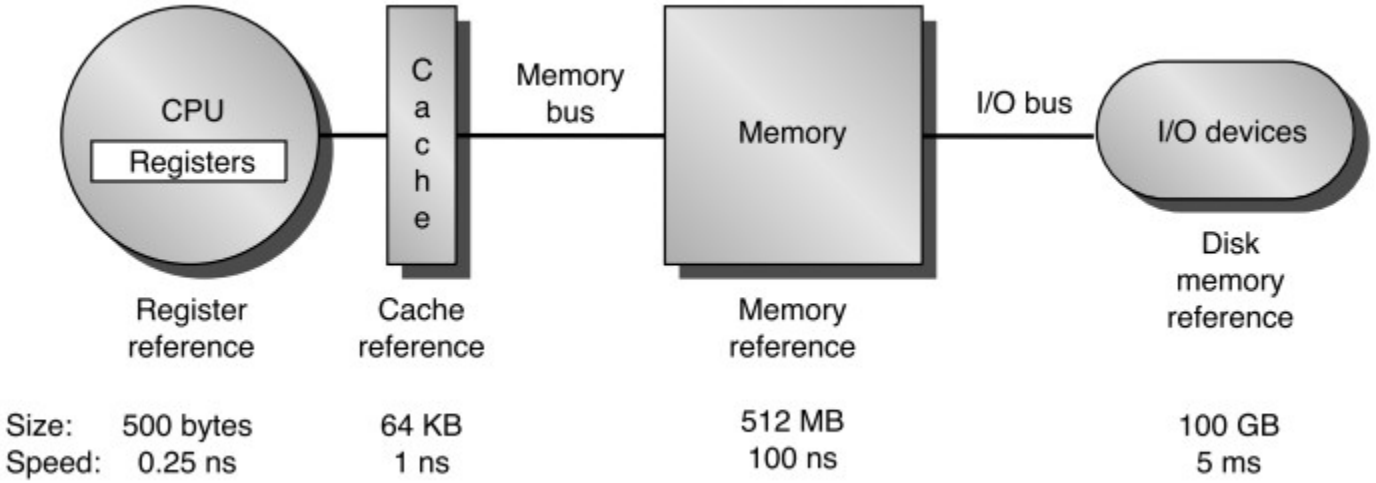
Soluções propostas

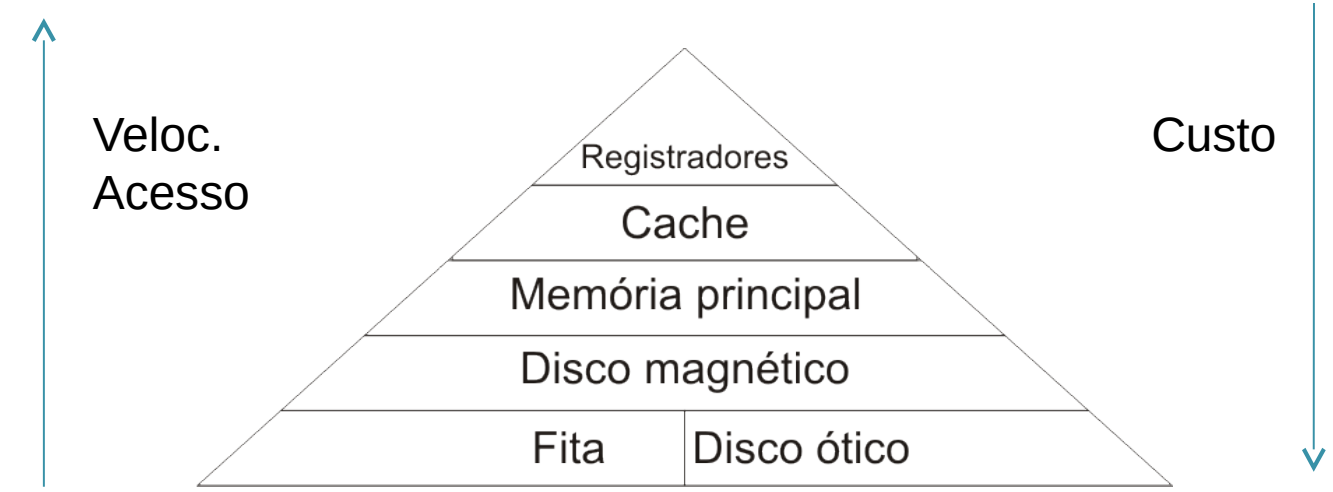
- O processador deve executar outras instruções enquanto aguarda acesso a memória. Porém isto nem sempre é possível e é difícil de implementar.
- Colocar a memória principal no chip do processador. Isto tornaria o chip maior e mais caro.

Soluções propostas

- Devido a grande diferença de velocidade existente entre o processador e a memória principal, foi desenvolvido um elemento intermediário que tem o propósito de minimizar o impacto desse problema no sistema de computação: **A memória Cache.**

Hierarquia de Memória





Idéia básica da Memória Cache

- As palavras de memória mais usadas pelo processador devem permanecer armazenadas na cache;
- Se o número de acessos a cache é grande, o tempo médio de acesso à memória diminui significativamente;
- Algumas constatações: Uso mais frequente de dados recém usados, de dados de loop e de dados matriciais.

Memória Cache

- **Princípio da proximidade:** Programas tendem a reutilizar os dados e as instruções usados recentemente. Existem 2 tipos de proximidade:
  1. **proximidade temporal:** elementos acedidos recentemente têm maior probabilidade de ser acedidos a seguir;
  2. **proximidade espacial:** elementos colocados em posições de memória próximas tendem a ser acedidos em instantes consecutivos.

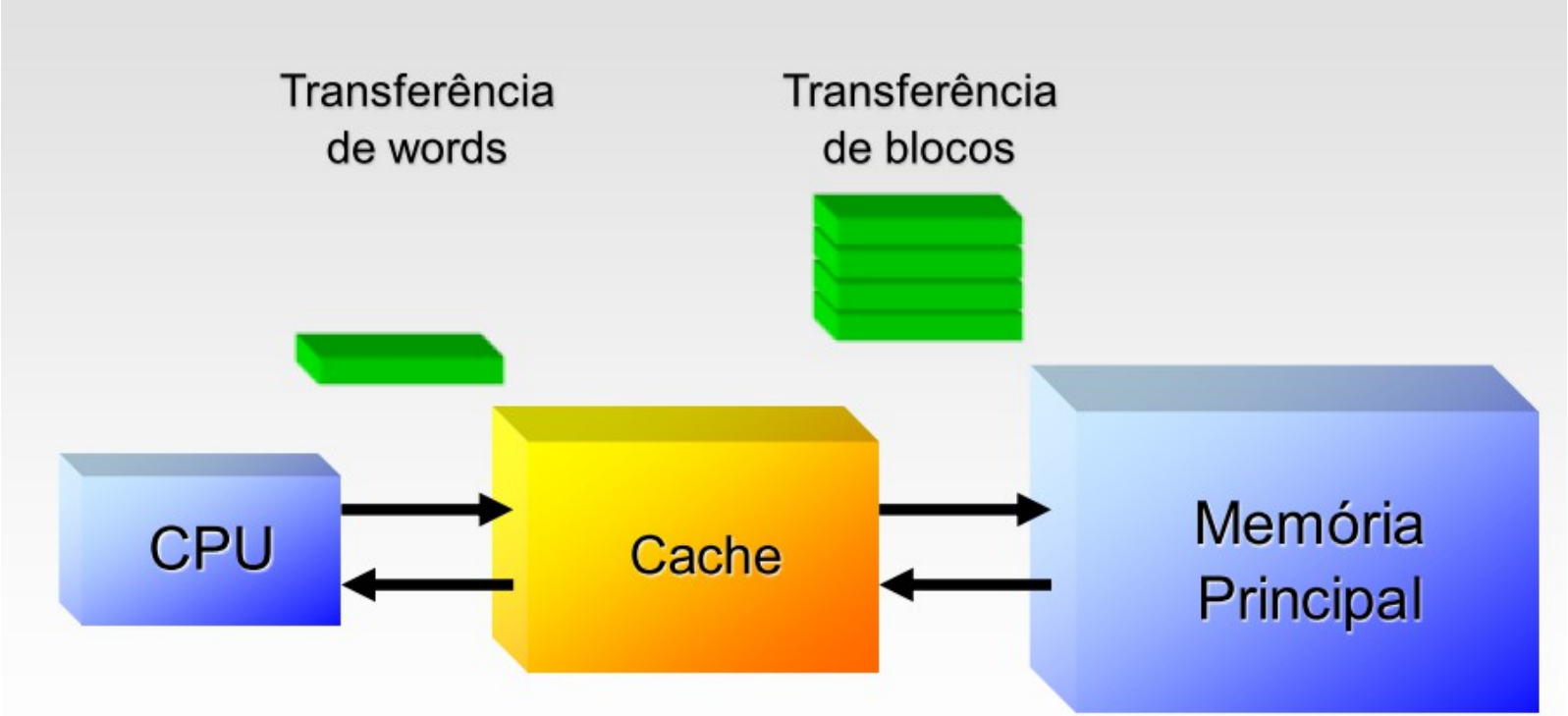
Memória Cache

- A Memória Cache (MC) se baseia fundamentalmente nos princípios de localidade temporal e espacial;
- Funciona como um elemento intermediário entre a CPU e a Memória Principal (MP) e armazenando as informações que muito provavelmente serão requisitadas pela CPU.

Utilização da MC

- Uma vez introduzida no sistema de computação o funcionamento do sistema é alterado de forma que antes de realizar um acesso direto a MP a informação é primeiramente buscada na Memória Cache.
  - Se a informação requisitada estiver presente na Memória Cache ocorre um acerto(hit) e a informação é transferida para a CPU em “alta” velocidade.
  - Caso contrario ocorre uma falta (miss) e o sistema busca a informação na MP, e a transfere para a memória Cache (juntamente com outras informações determinadas pelo principio da proximidade).

Utilização da MC



Considerações sobre a Utilização da MC

- Para haver aumento de desempenho do sistema é necessário que hajam muito mais acertos (hits) do que faltas (misses) de maneira que as eventuais perdas de desempenho com faltas seja compensada pela taxa de acertos.
- A taxa de acertos mais comum em sistemas atuais varia entre 80% e 99%, o que garante um ganho de desempenho considerável com a utilização de memórias cache.

Tempo médio de acesso à cache

- $T_{ma} = h*Tc+(1-h)*Tm$

  - Tma = Tempo médio de acesso
  - Tc = Tempo de acesso a cache
  - Tm = Tempo de acesso a Memória Principal
  - h = Taxa de acerto

Tempo médio de acesso à cache

- Exemplo: Se a taxa de acerto na execução de um programa foi de 85% e o tempo de acesso à cache é de 10ns e o tempo de acesso à memória principal é de 80ns, então o tempo médio de acesso pode ser caculado da seguinte maneira:

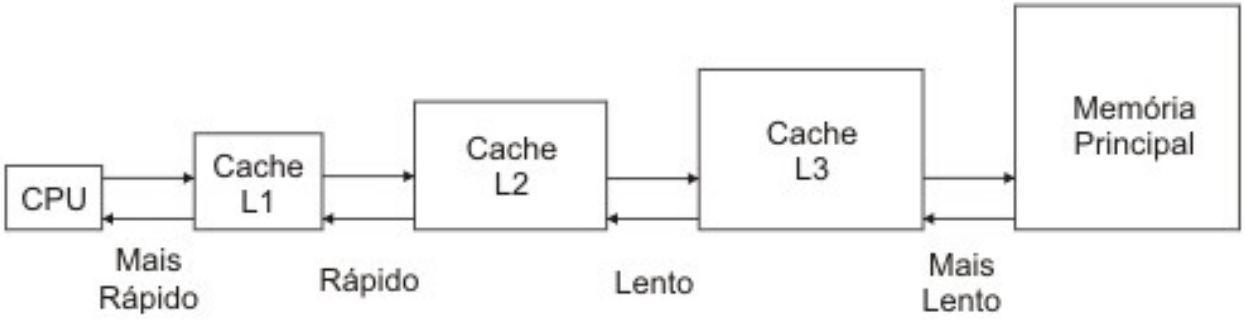
  $T_c = 10\ ns,\ Tm = 80\ ns,\ h = 0,85$

  $Tma = 0,85*10+0,15*80$

  $Tma = 20,5\ ns$

Níveis de Memória Cache

- Com o aumento crescente da velocidade da CPU e visando minimizar um grande impacto no custo da Memória Cache os fabricantes vêm estabelecendo diferentes níveis de memória cache.



Níveis de Memória Cache

- Cache L1:
  - Uma pequena porção de memória estática presente dentro do processador;
  - Geralmente tem entre 16KB e 128KB; hoje já encontramos processadores com at é 16MB de cache.

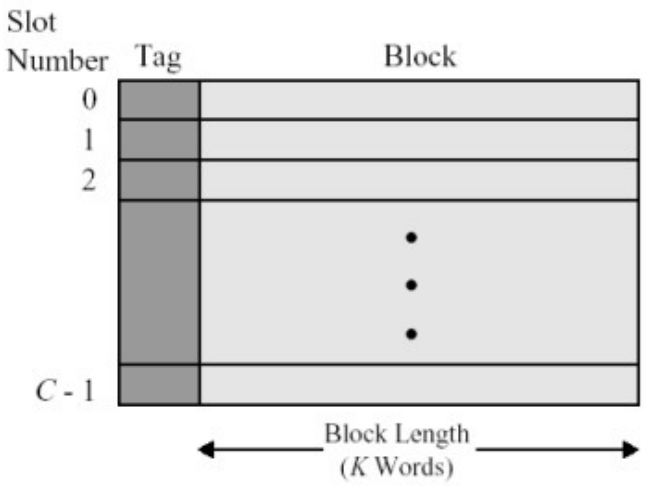
Níveis de Memória Cache

- Cache L2:
  - A partir do Pentium II passou a ser instalada dentro do processador;
  - Tamanho maior que a cache L1

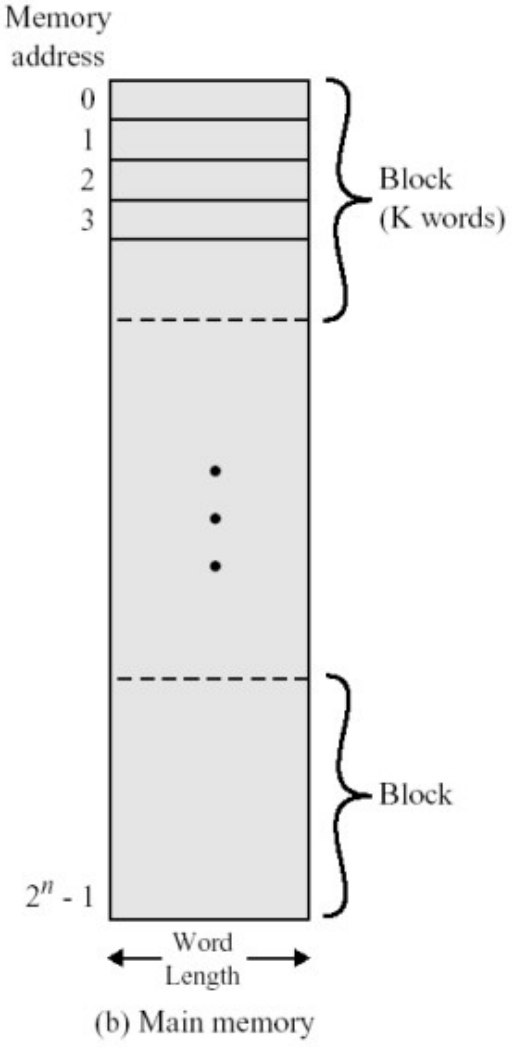
Níveis de Memória Cache

- Cache L3:
  - Terceiro nível de cache de memória. Maior que a cache L2;
  - Inicialmente utilizado no AMD K6-III;
  - Hoje é utilizada em CPUs da Intel i3, i5, i7.

Estrutura Cache/Mem. Principal



(a) Cache



(b) Main memory

Projeto de memória cache

1) Tamanho da cache;
2) Linha da cache;
3) Modo de organização da cache;
4) Instruções e dados mantidos na mesma cache (cache unificada) ou em caches diferentes (Arquitetura Harvard);
5) Número de caches.

Funções de Mapeamento

- Técnicas para estabelecer uma associação entre as células da MP e os blocos da MC.
  - Mapeamento Direto;
  - Mapeamento Associativo;
  - Mapeamento Associativo por Conjuntos.

Funções de Mapeamento- Mapeamento direto

- É o método mais simples, sendo cada bloco da memória principal mapeado em uma única linha da cache. Para que isso seja feito podemos pensar no seguinte mapeamento:

$$I = J \ MOD \ M$$

I = Numero da linha cache

J = Numero do bloco da memória principal

M = Quantidade de linhas da cache

Funções de Mapeamento-Mapeamento associativo

- Cada bloco da memória principal pode ser alocado em qualquer posição da cache.
- A busca por um bloco é feita ao mesmo tempo em paralelo em todas as entradas da cache.

Funções de Mapeamento-Mapeamento Associativo por Conjuntos

- Criada com o objetivo de eliminar os problemas das técnicas de mapeamento direto e mapeamento associativo;
  - Blocos da MP são associados a um conjunto de linhas na MC.

$$M = V \times K;$$

$$I = J \text{ MOD } V;$$

I : número do conjunto da cache;

J : número do bloco na memória principal;

M : número de linhas na cache;

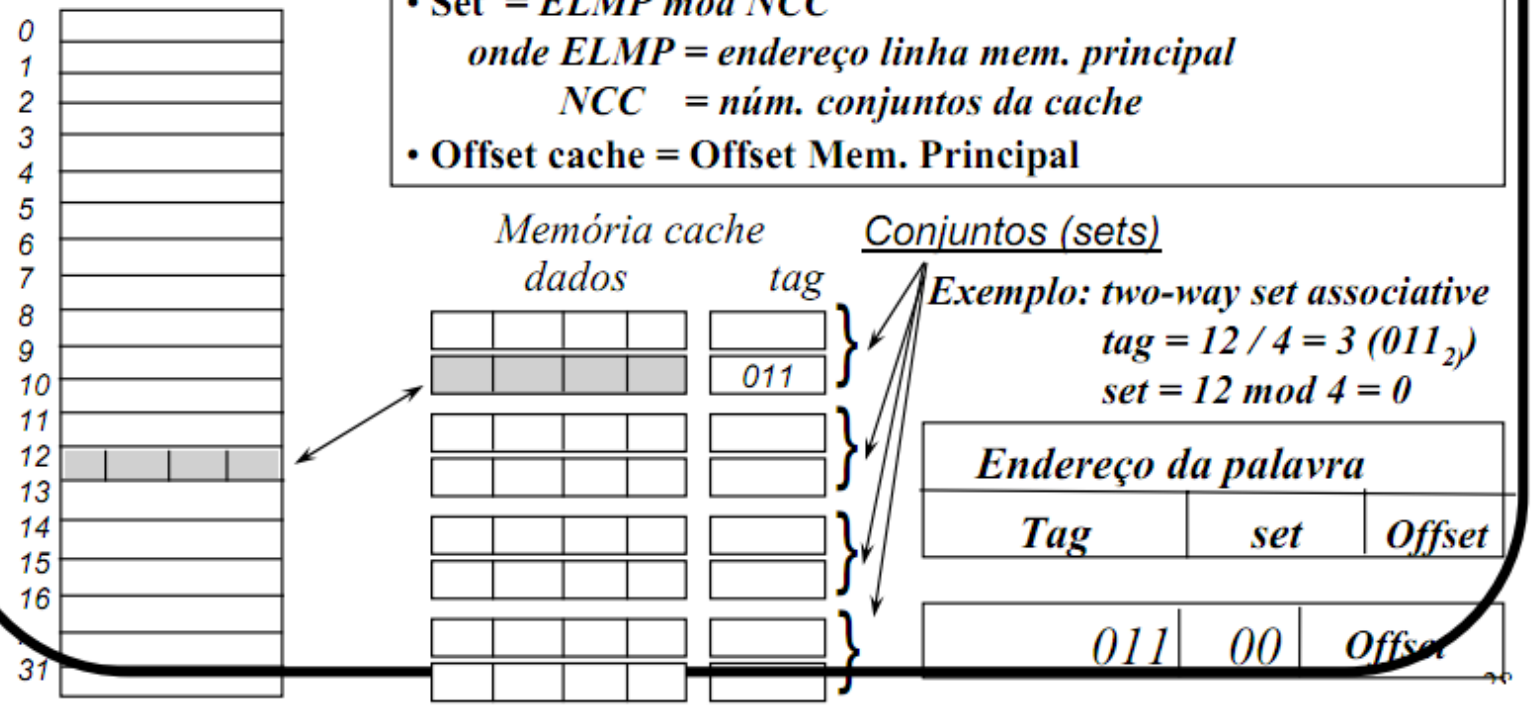
V : número de conjuntos;

K : número de linhas da cache.

Mapeamento Associativo por Conjunto

Uma linha na memória principal pode ocupar qualquer posição dentro de um conjunto definido de linhas da cache

- **Tag =** $\lfloor$ ***ELMP / NCC*** $\rfloor$
- **Set = *ELMP mod NCC***
  onde ELMP = endereço linha mem. principal
  NCC = núm. conjuntos da cache
- **Offset cache = Offset Mem. Principal**



Algoritmos de substituição de dados

- A capacidade de armazenamento da MP é muito maior do que a capacidade de armazenamento da MC;
- Os algoritmos de substituição de dados na MC têm a tarefa de definir qual dos blocos já armazenados na Memória Cache deve ser retirado para o armazenamento de um novo bloco.

Algoritmos de substituição de dados

- Algoritmos:
- **LRU (Least Recently Used):** Determina como candidatos à substituição os que não foram acessados recentemente.
- **FIFO (First-In-First-Out):** Seleciona como candidado para substituição o bloco que foi armazenado primeiro na MC;
- **LFU (Least Frequently Used):** o sistema de controle selecionará o bloco que tem tido menos acessos por parte do processador;
- **Escolha Aleatória:** O sistema de controle da memória Cache escolhe aleatoriamente o bloco que será removido.

Políticas de escrita

- Mecanismos para garantir a integridade das informações processadas no sistema, apesar das transferências entre a MP e a MC;
- Escrita em Ambas (write through)
  - toda modificação de dados na Cache acarreta uma modificação na MP.
- Escrita somente no retorno (write back)
  - A informação modificada na Cache só será repassada para a MP quando estiver a ponto de ser substituída.

Tamanho da Memória Cache

- A definição da faixa de tamanho (capacidade de armazenamento) de uma cache depende:
  - Capacidade de armazenamento da MP;
  - Razão acertos/falhas aceitavel;
  - Tempo de acesso da MP e da MC;
  - Custo da MP e MC;
  - Natureza dos programas em execução.
- Estudos apontam que capacidades aceitáveis para MC como:
  - Entre 32K e 256Kbytes para Caches L1;
  - Entre 64K e 4Mbytes para Caches L2;

Evolução de cache na Intel

Problema	Solução	Processador
Memória externa mais lenta que o barramento do sistema	Acrescentar cache externa usando tecnologia de memória mais rápida	386
O aumento da velocidade do processador torna o barramento externo um gargalo para o acesso a MC	Mover a cache externa para o chip, trabalhando na mesma velocidade do processador	486
Cache interna um tanto pequena, devido ao espaço limitado do chip.	Acrescentar cache L2 externa usando tecnologia mais rápida que a memória principal	486
Quando ocorre uma disputa entre o mecanismo de pré-busca de instruções e a unidade de execução no acesso simultaneo à memória cache. Nesse caso a busca é adiada até o término do acesso da unidade de execução dos dados.	Criar caches separadas para dados e instruções	Pentium
Maior velocidade do processador torna o barramento um gargalo para o acesso a cache L2.	Criar barramento back-side separado dedicado a cache L2	Pentium Pro
	Mover a cache L2 para o chip do processador	Pentium II
Algumas aplicações lidam com BD enormes, e precisam de acesso rárpido. As caches dos chip são muito pequenas	Acrescentar cache L3 externa	Pentium III
	Mover cache L3 para o chip	Pentium 4

Bibliografia:

- Stallings, W. Arquitetura e Organização de Computadores. 8 ed – Editora Pearson, 2009.
- Tanembaum, A. S. Organização Estruturada de Computadores. 5 ed – Editora Pearson, 2007.