Organização e Arquitetura de Computadores I

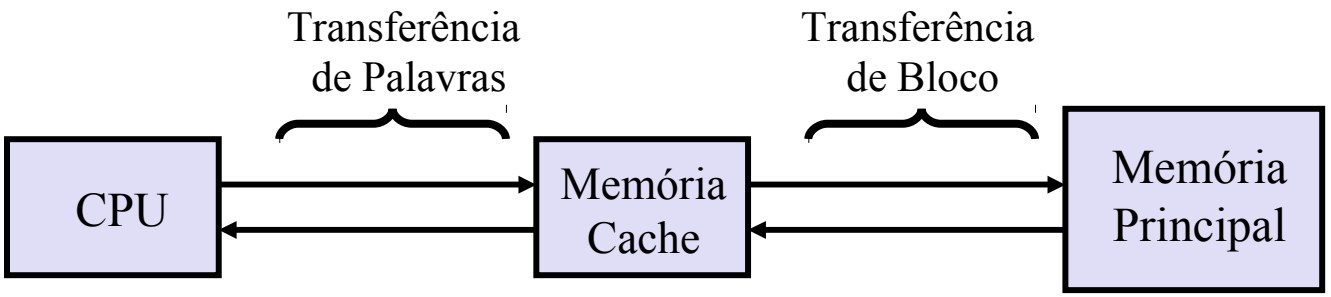
Memória Cache

- Introdução
- Projetos de Memória Cache
  - Tamanho
  - Função de Mapeamento
  - Política de Escrita
  - Tamanho da Linha
  - Número de Memórias Cache
  - Cache em Níveis

Introdução

- O uso da memória cache, visa obter uma velocidade de acesso à memória próxima da velocidade das memórias mais rápidas e, ao mesmo tempo, disponibilizar no sistema uma memória de grande capacidade, a um custo equivalente ao das memórias de semicondutor mais barata.
- A memória cache é combinada com uma memória principal relativamente grande e lenta com a finalidade de obter um melhor desempenho.

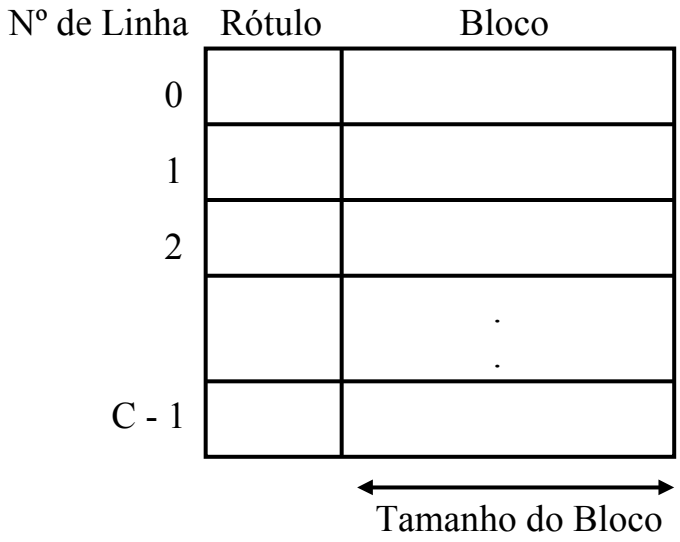
Introdução


Transferência de Palavras
Transferência de Bloco
CPU
Memória Cache
Memória Principal

Introdução

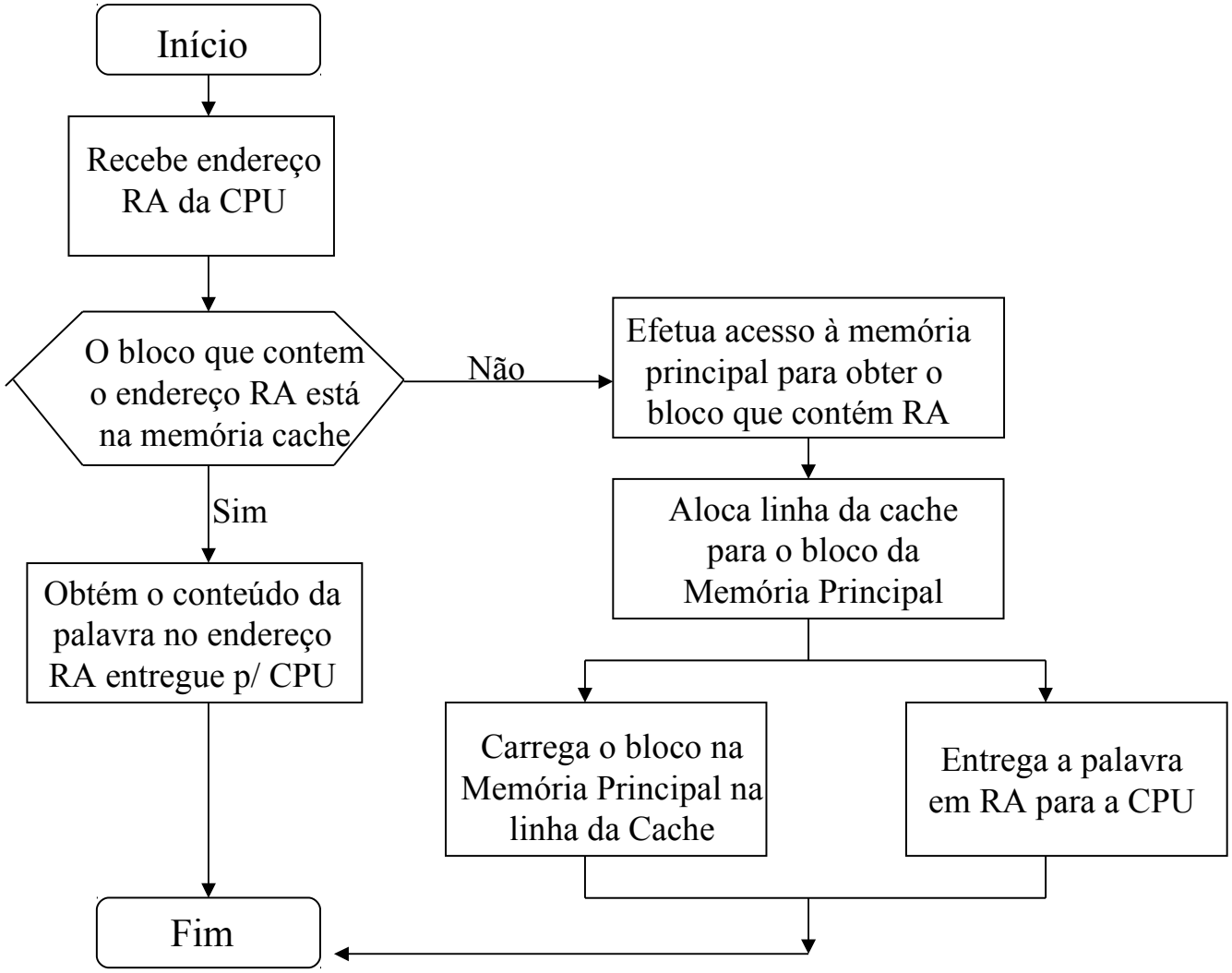
- A Memória Cache é constituída por várias linhas de palavras, sendo o número de linhas consideravelmente menor do que o número de blocos da memória principal. Ao ler um bloco da memória principal, ele é transferido para uma das linhas da memória cache, assim não é possível que uma linha armazene um mesmo bloco permanentemente, por isso, é necessário que cada linha inclua um rótulo, que identifica qual é o bloco de memória nela armazenada. Este rótulo é geralmente uma parte do endereço de memória principal.

Introdução



C = número de linhas da memória

Organização e Arquitetura de Computadores I



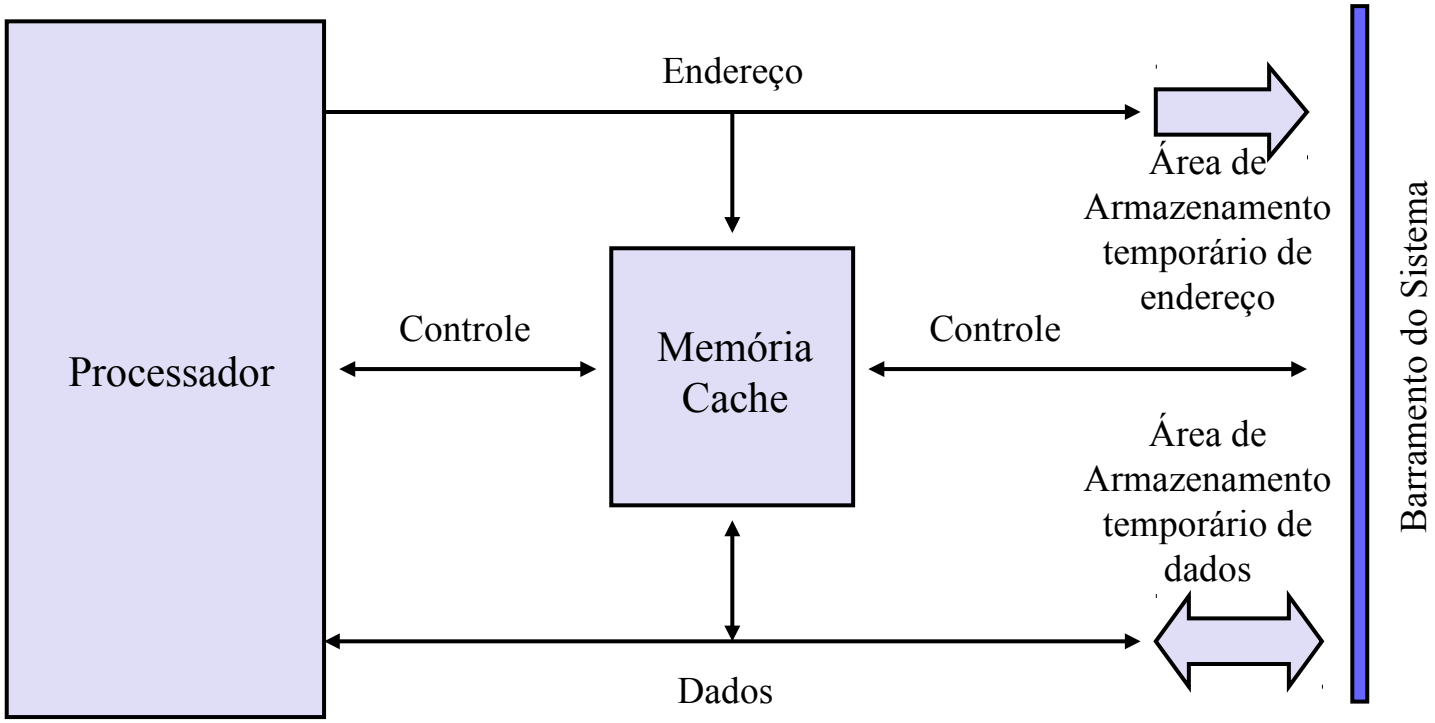
Introdução

- O processador gera um endereço, RA, da palavra a ser lida
- Se essa palavra estiver contida na memória cache, ela será entregue ao processador
- Caso contrário, o bloco que contém essa palavra será carregado na memória cache e a palavra entregue ao processador
- Essas duas últimas operações ocorrem paralelamente, refletindo a organização típica de memórias cache modernas

Introdução

- Em memórias cache modernas, as linhas de dados e de endereços são também conectadas a áreas de armazenamento temporário de dados e de endereços, que se conectam ao barramento do sistema, por meio do qual é feito o acesso à memória principal.

Introdução


Processador
Endereço
Área de Armazenamento temporário de endereço
Controle
Memória Cache
Controle
Área de Armazenamento temporário de dados
Barramento do Sistema
Dados

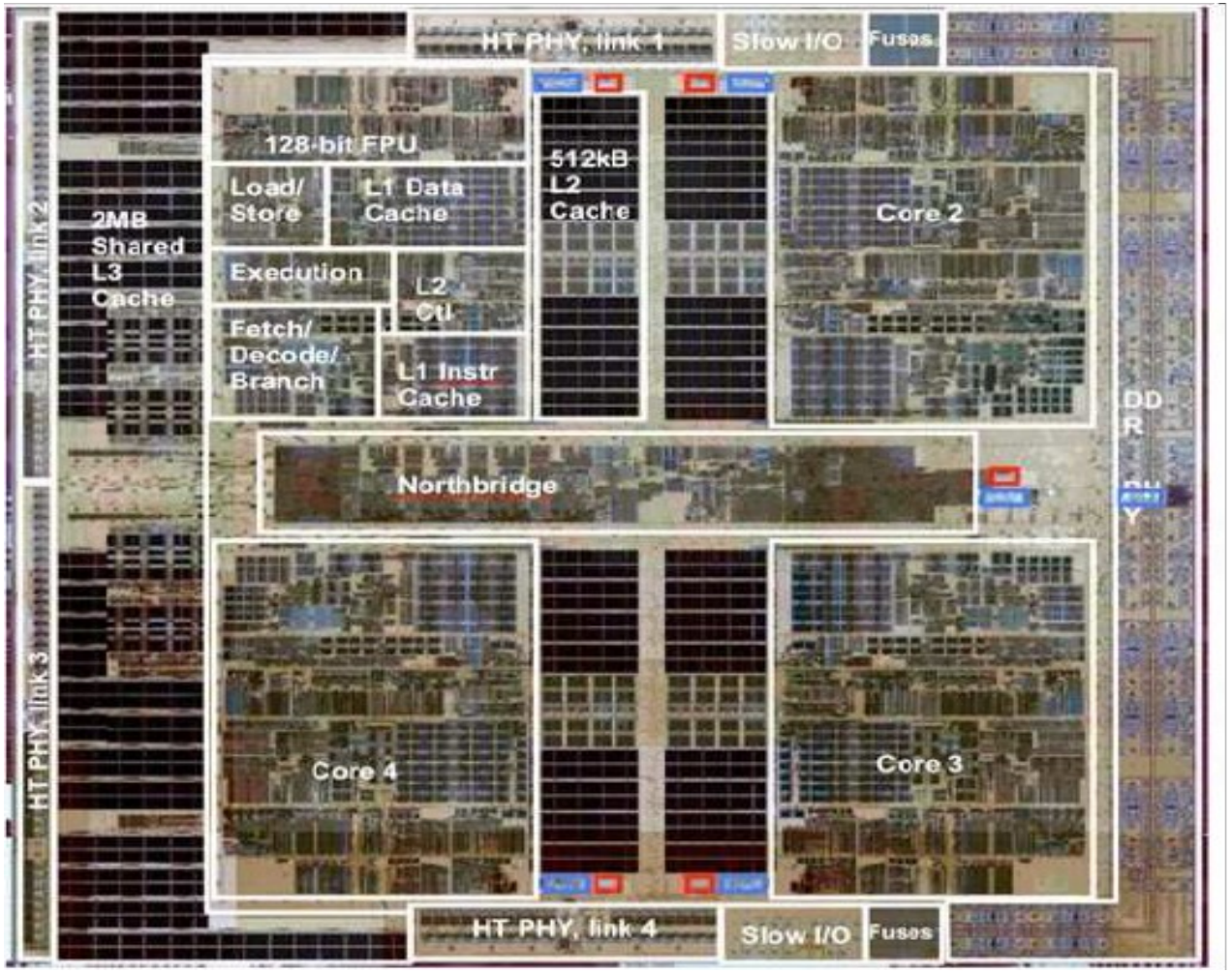
Projetos de Memória Cache

- **Tamanho da Memória Cache**
  - O tamanho de uma memória cache deve ser suficientemente pequeno, para que o custo total médio por bit seja próximo do custo por bit da memória principal e deve ser grande o suficiente para que o tempo de acesso médio a memória seja próximo ao tempo de acesso da memória principal.
  - O tamanho da memória cache também é limitado pelo tamanho da pastilha e da placa de circuito.
  - Quanto maior é a memória cache maior é o número de portas envolvidas no seu endereçamento e mais lento será seu processamento.

Projetos de Memória Cache

- **Tamanho da Memória Cache**
  - Em suma, como o desempenho de uma memória cache é muito sensível à natureza da carga de trabalho imposta, é impossível determinar um tamanho ótimo.

Organização e Arquitetura de Computadores I



Projetos de Memória Cache

- **Função de Mapeamento**
  - Como o número de linhas de memória cache é menor do que o de blocos da memória principal, é necessário um algoritmo para mapear os blocos da memória principal em linhas da memória cache, é necessário ainda um mecanismo para determinar o bloco da memória principal que ocupa uma dada linha da memória cache, determinando como a memória cache é organizada.

Projetos de Memória Cache

- **Função de Mapeamento**
  - Os exemplos que seguirão, incluem os seguintes elementos:
    - A memória cache pode conter 64 Kbytes.
    - Os dados são transferidos entre a memória principal e a memória cache em blocos de 4 bytes. Dessa forma, a memória cache é organizada com 16K = 214 linhas de 4 bytes cada uma.
    - A memória principal com 16 Mbytes, sendo cada byte endereçável diretamente por um endereço de 24 bits.

Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Direto**
    - Cada bloco da memória principal é mapeado em uma única linha de cache. É representado pela seguinte equação:

      i = j módulo m

      i = número da linha da memória cache.
      j = número do bloco da memória principal.
      m = número de linhas na memória cache.

    - Temos ainda:

      w: que indica uma única palavra ou byte dentro de um bloco da memória principal.

      s: que indica um único bloco da memória principal.

      r: que indica uma linha da memória cache.

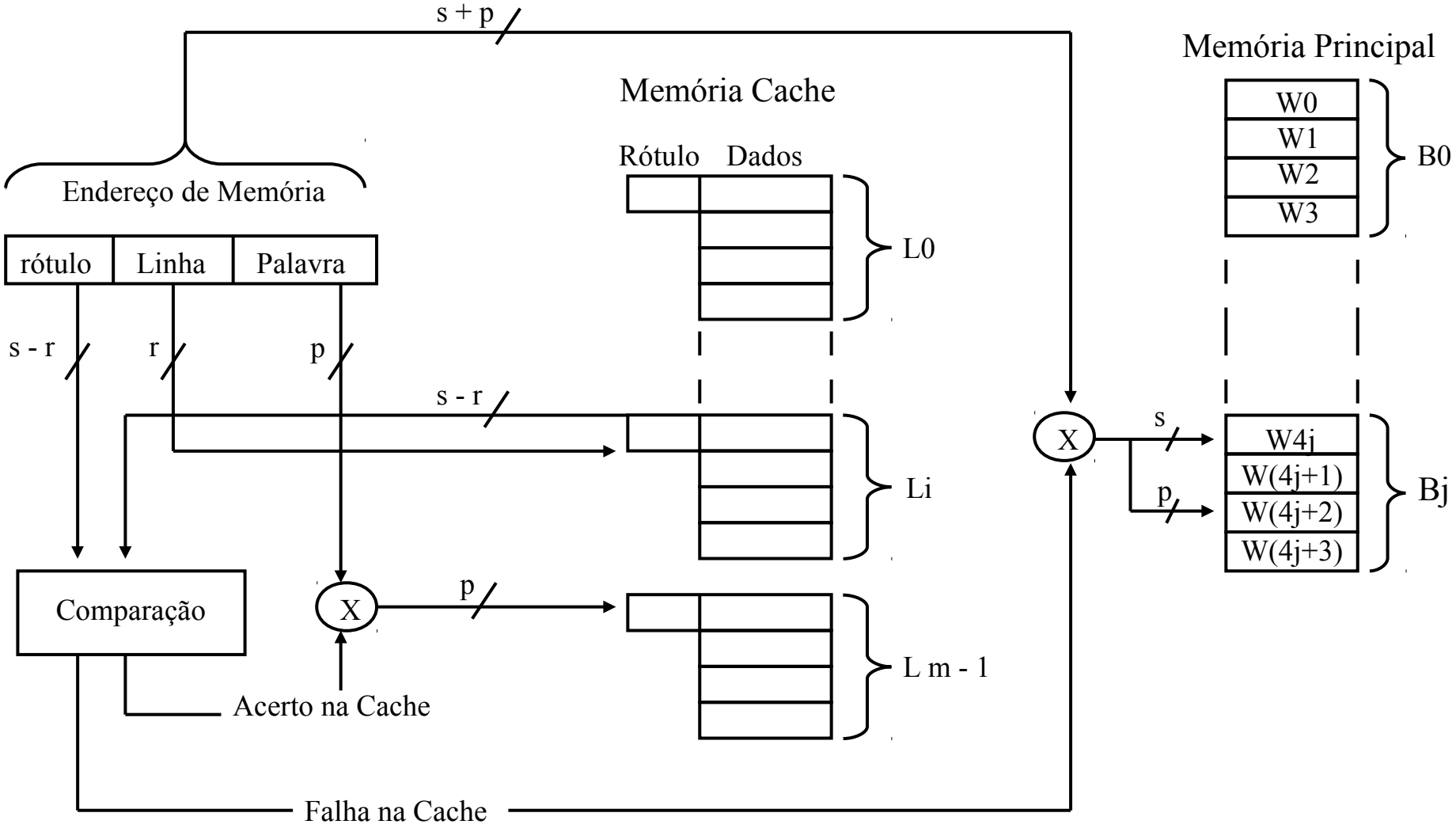
      s – r: rótulos que interpretam na memória cache os blocos da memória principal.

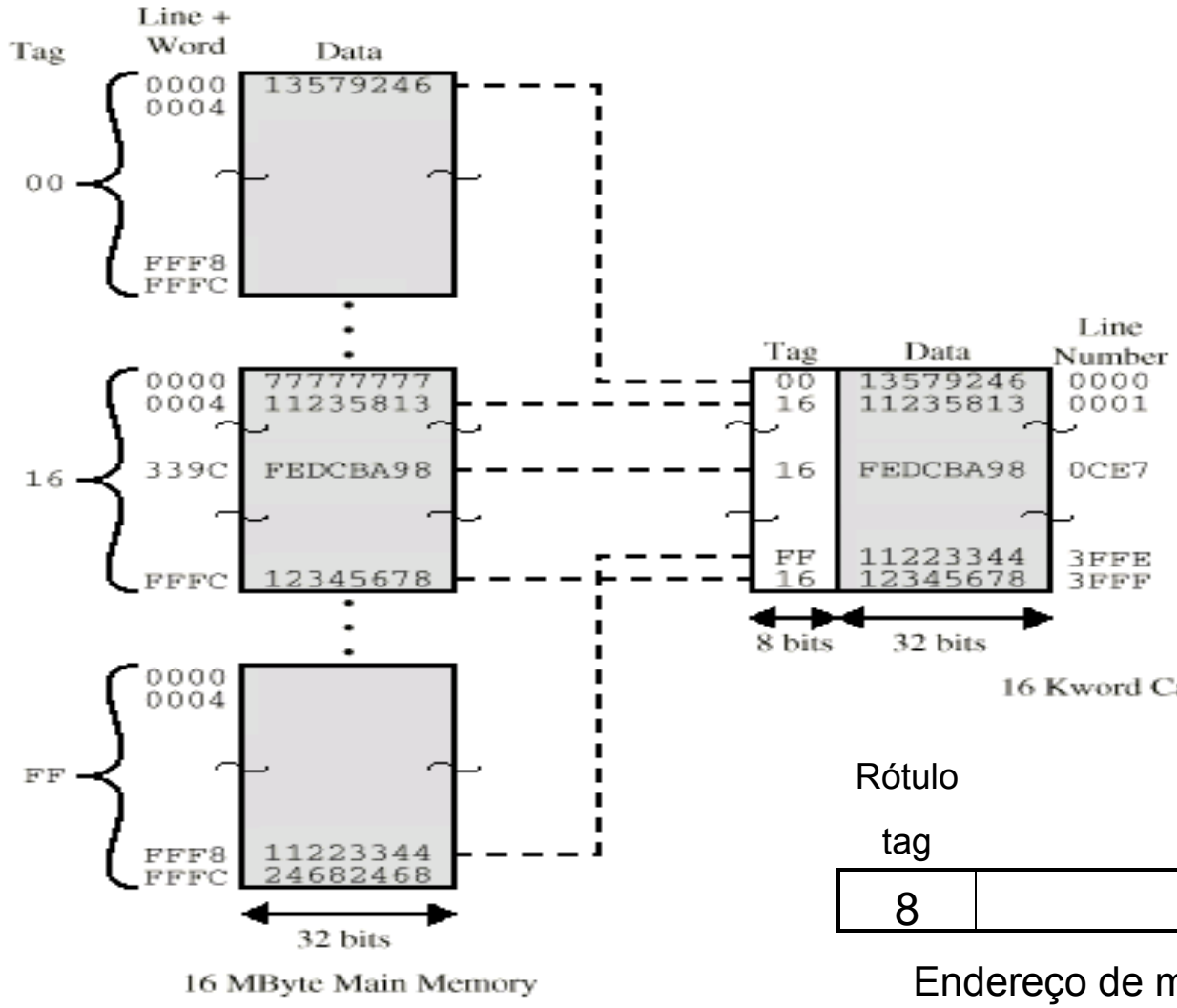
Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Direto**
    - Os blocos da memória principal são armazenados nas linhas da memória cache da seguinte forma:

Linha da Memória Cache	Blocos da Memória Principal Mapeados
0	0, m, 2m, ..., 2/s – m
1	1, m + 1, 2m + 1, ..., 2/s – m + 1
:	
m-1	m – 1, 2m – 1, ..., 2/s - 1

Organização e Arquitetura de Computadores I





Rótulo	Linha	Palavra
tag	line	word
8	14	2

Endereço de memória Principal

FFFFFC	11111111-	11111111111111-	11
endereço	tag	line	word
	FF	3FFF	3
16FFFC	00010110-	11111111111111-	11
endereço	tag	line	word
	16	3FFF	3

Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Direto**
    - O Mapeamento Direto é simples e tem custo de implementação baixo, sua principal desvantagem é que cada bloco é mapeado em uma posição fixa na memória cache, assim se um programa fizer repetidas referências a palavras de dois blocos distintos, mapeados em uma mesma linha, esses blocos serão trocados continuamente na memória cache e a taxa de acertos à memória cache será baixa.

Projetos de Memória Cache

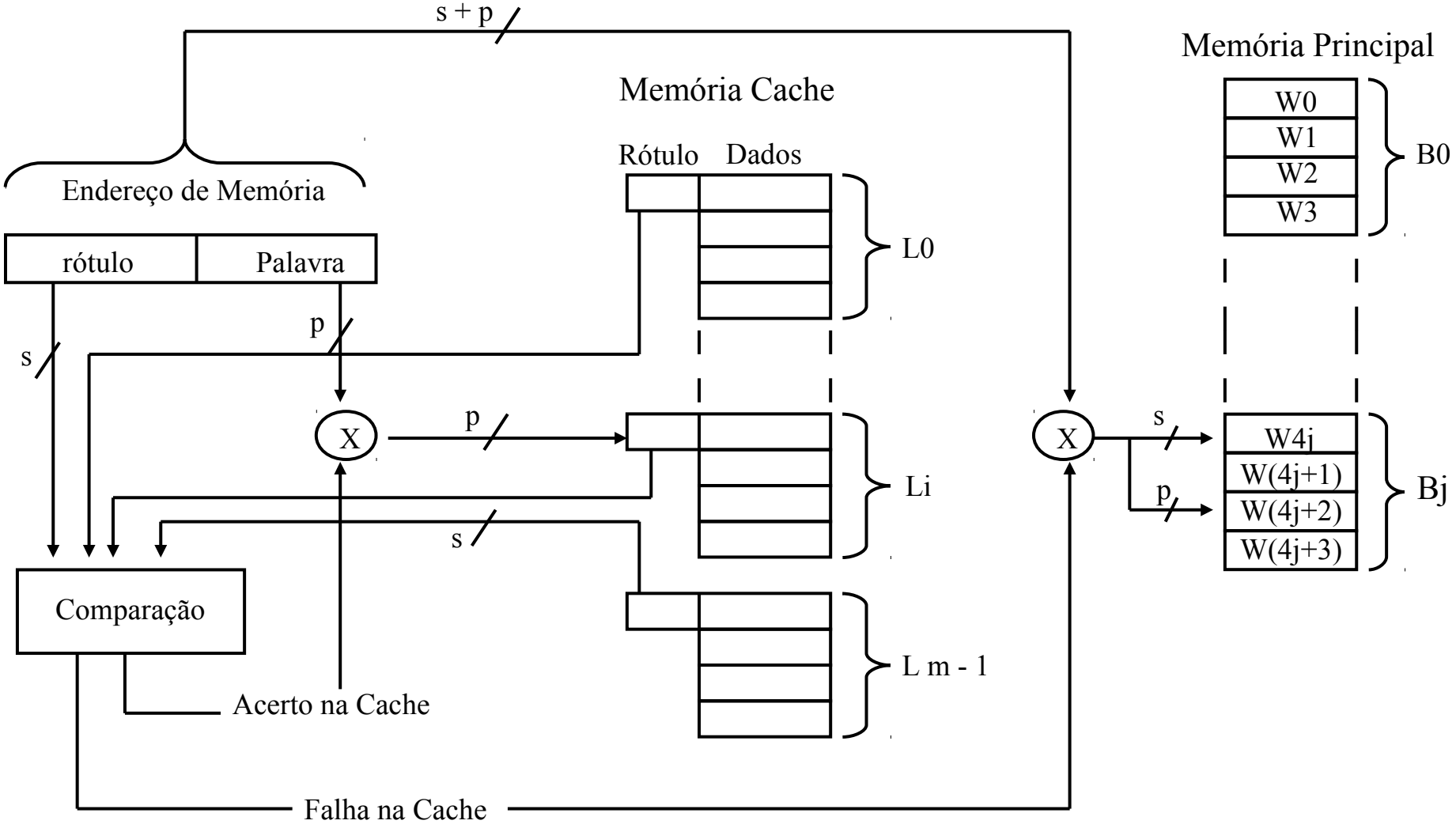
- **Função de Mapeamento**
  - **Mapeamento Associativo**
    - Evita a desvantagem do mapeamento direto, permitindo que cada bloco da memória principal seja carregado em qualquer linha da memória cache. Assim, o controle da memória cache interpreta um endereço de memória como constituído apenas por rótulo e campo de palavra. Onde o rótulo indica um bloco da memória principal, e para determinar se um bloco esta na memória cache, compare-se simultaneamente o campo de rótulo do endereço do bloco acessado, com os rótulos de todas as linhas.

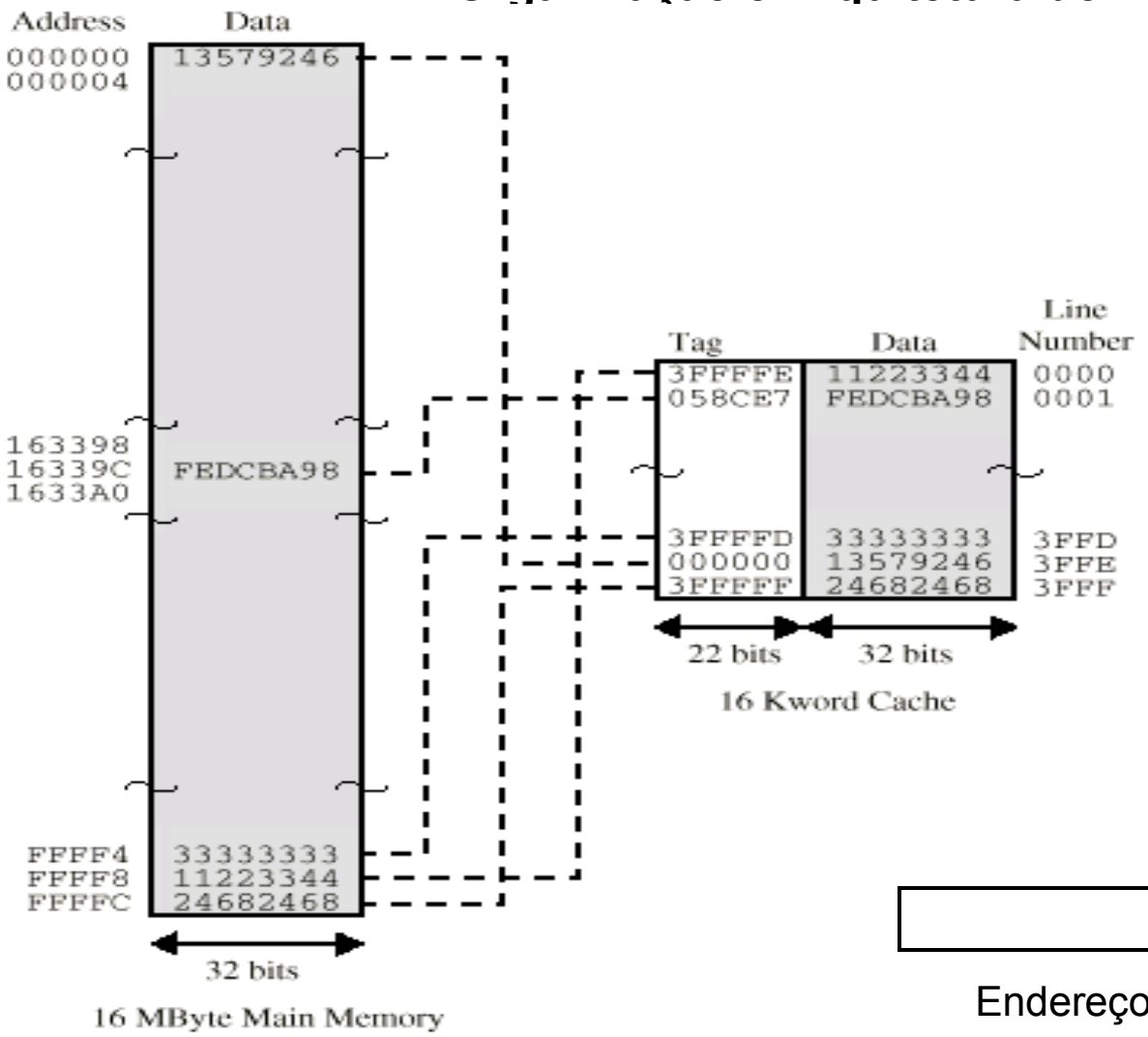
Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Associativo**
    - Um endereço de memória principal, é composto de um rótulo de 22 bits e de um número de byte de 2 bits;
    - Os 22 bits do rótulo são armazenados junto com os 32 bits de dados do bloco, em cada linha de memória cache;
    - Dessa forma, o endereço hexadecimal de 24 bits, 16339C, tem rótulo igual a 058CE7. Isso pode ser visto mais facilmente utilizando a notação binária:

Endereço de memória	0001 0110 0011 0011 1001 1100	(bin)
	1 6 3 3 9 C	(hex)
Rótulo	00 0101 1000 1100 1110 0111	(bin)
	0 5 8 C E 7	(hex)

Organização e Arquitetura de Computadores I





Rótulo	Palavra
22	2

Endereço de memória Principal

Projetos de Memória Cache

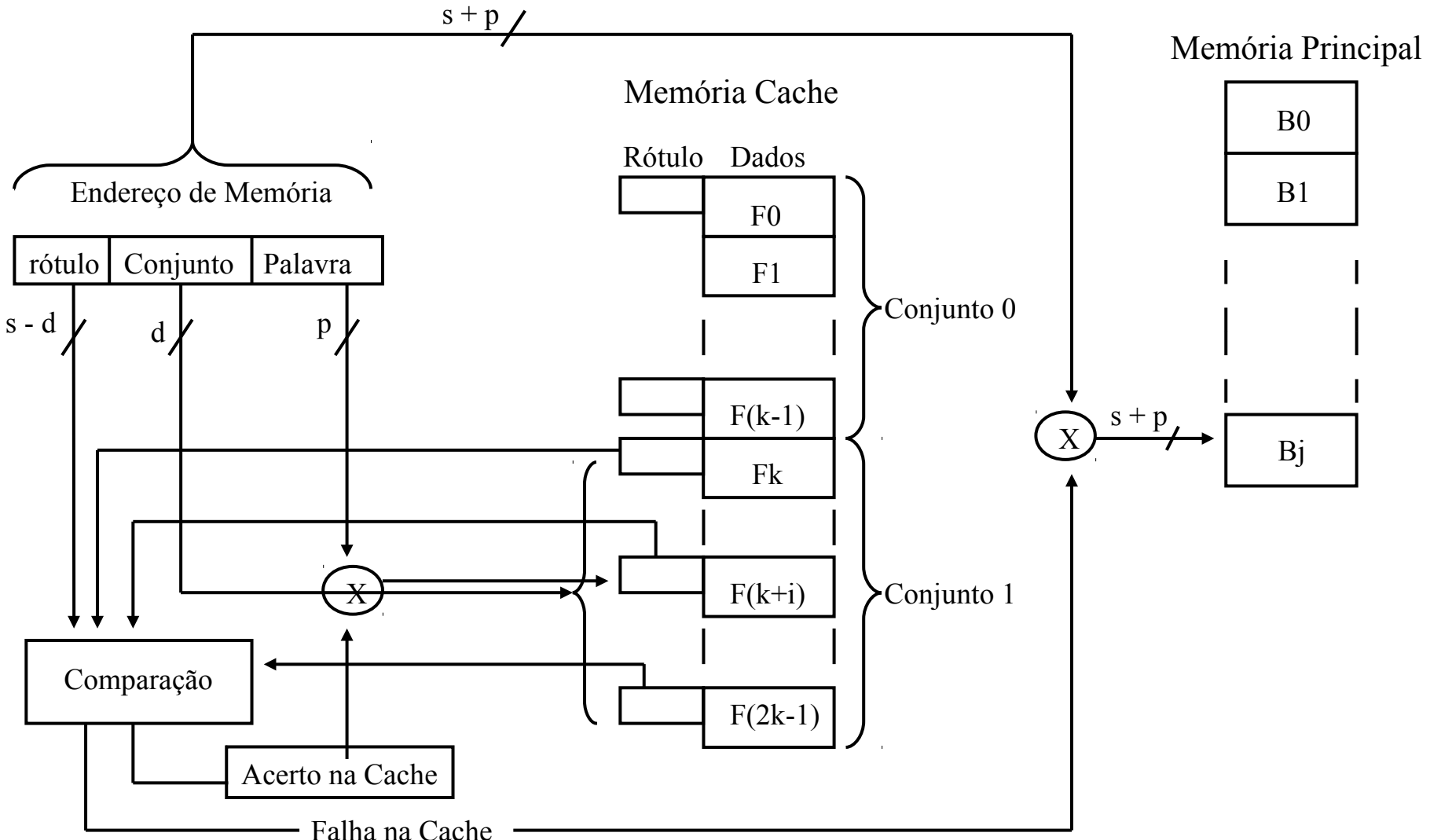
- **Função de Mapeamento**
  - **Mapeamento Associativo**
    - Ele oferece maior flexibilidade para a escolha do bloco a ser substituído quando um novo bloco é trazido para a memória cache. São utilizados algoritmos de substituição para maximizar a taxa de acertos na cache.
    - A principal desvantagem do mapeamento associativo é a complexidade do conjunto de circuitos, necessários para a comparação dos rótulos de todas as linhas da memória cache em paralelo.

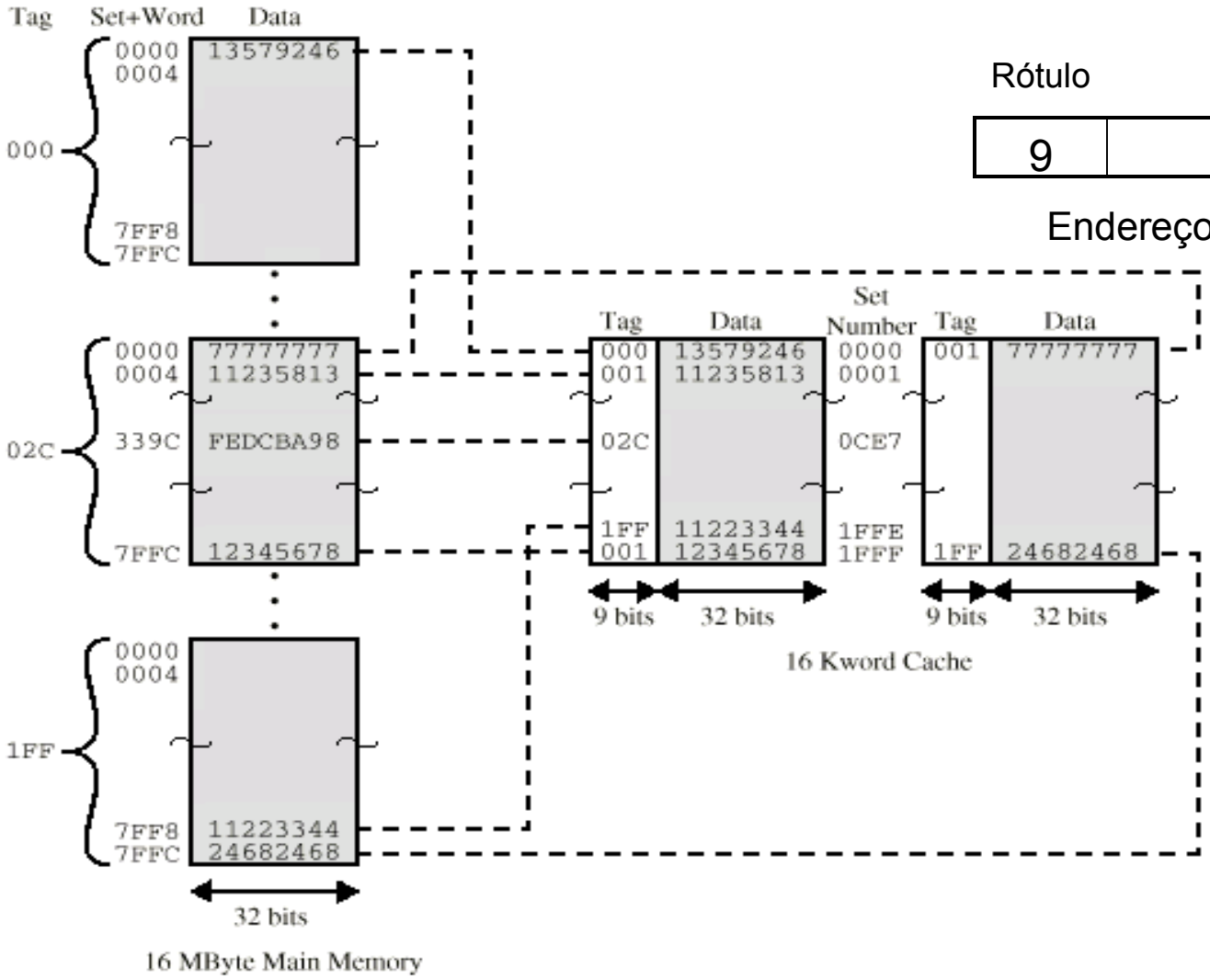
Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Associativo por Conjuntos**
    - Nesse mapeamento, são combinadas vantagens do mapeamento direto e do mapeamento associativo, e diminui suas desvantagens.
    - A memória cache é dividida em v conjuntos, cada qual com k linhas.
    - Em um mapeamento totalmente associativo, o rótulo de um endereço de memória é muito grande e é comparado com o rótulo de cada linha de memória cache.

Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Associativo por Conjuntos**
    - Em um mapeamento associativo por conjuntos de k linhas, o rótulo de um endereço de memória é muito menor e é comparado apenas com k rótulos de um mesmo conjunto.
    - A figura a seguir, mostra nosso sistema utilizando um mapeamento associativo por conjuntos com duas linhas em cada conjunto, denominado mapeamento associativo por conjuntos de duas linhas.


s + p
Memória Principal
Memória Cache
Rótulo
Dados
B0
B1
Endereço de Memória
F0
F1
rótulo
Conjunto
Palavra
Conjunto 0
s - d
d
p
F(k-1)
Fk
X
s + p
Bj
F(k+i)
Conjunto 1
Comparação
F(2k-1)
Acerto na Cache
Falha na Cache



Rótulo	Linha	Palavra
9	13	2

Endereço de memória Principal

Endereço FFFFFC 11111111111111111111100

111111111-11111111111100 1111111111111

tag line +word line

1FF 7FFC 1FFF

Endereço 16FFFF 00010110111111111111111

000101101-11111111111111 11111111111111

tag line+word line

02C 7FFC 1FFF

Projetos de Memória Cache

- **Função de Mapeamento**
  - **Mapeamento Associativo por Conjuntos**
    - A organização de mapeamento associativo por conjuntos mais comum é a que usa duas linhas por conjunto (v = m/2, k = 2).
    - Sua taxa de acertos é significativamente maior do que no caso do mapeamento direto.
    - Um maior número de linhas por conjunto não apresenta melhoras significativas de desempenho.

Projetos de Memória Cache

- **Algoritmo de Substituição**
  - Quando um novo bloco é trazido para a memória cache, um dos blocos existentes deve ser substituído.
  - No mapeamento direto, cada bloco é mapeado em uma única linha, o que determina o bloco a ser substituído, não havendo alternativa possível.
  - Nos demais mapeamentos, existe o uso de um algoritmo de substituição. Esse algoritmo é implementado em hardware para aumentar a velocidade de acesso à memória.

Projetos de Memória Cache

- **Algoritmo de Substituição**
  - Os quatro algoritmos mais comuns são:
    - ***Least recently used* – LRU**: substitui o bloco menos recentemente usado. É usado um bit de USO para indicar quando uma linha foi referenciada.
    - ***First-in-First-out* – FIFO**: substitui o bloco do conjunto que foi armazenado primeiro na memória cache.
    - ***Least frequently used* – LFU**: substitui o bloco do conjunto menos freqüentemente utilizado. É usado um contador a cada linha da memória cache.
    - **Substituição aleatória**.

Projetos de Memória Cache

- **Política de Atualização**
  - Antes que um bloco residente na memória cache possa ser substituído, é necessário verificar se ele foi alterado:
    - Se isso não ocorreu, então o novo bloco pode ser escrito sobre o bloco antigo.
    - Caso contrário, se pelo menos uma operação de escrita foi feita, sobre uma palavra dessa linha de memória cache, então a memória principal deve ser atualizada.

Projetos de Memória Cache

- **Política de Atualização**
  - Dois problemas devem ser considerados:
    - Alteração da memória principal por um dispositivo de E/S
    - Múltiplos processadores com múltiplas memórias cache

Projetos de Memória Cache

- **Política de Atualização**
  - A técnica de atualização mais simples é denominada **escrita direta** (*write-through*);
    - Todas as operações são feitas tanto na memória cache, quanto na memória principal
    - A desvantagem é que ela gera um tráfego de memória considerável
  - Um técnica alternativa é a **escrita de volta** (*writeback*)

Projetos de Memória Cache

- **Política de Atualização**
  - Um técnica alternativa é a **escrita de volta** (*writeback*)
    - As escritas são feitas apenas na memória cache
    - O bloco só é substituído se seu bit de atualização estiver com o valor 1
    - Desvantagem: acesso de módulo de E/S direto à cache

Projetos de Memória Cache

- **Tamanho da Linha**
  - O tamanho da linha da memória cache, está relacionado ao tamanho do bloco da memória principal.
  - Quando aumentamos o número de linhas e blocos, temos um aumento de acerto na memória cache, devido a maior capacidade de armazenamento e reutilização da palavra, que foi utilizada e ainda continua armazenada na cache.
  - Com um tamanho de bloco maior, mais dados úteis são trazidos para a memória cache.
  - Com um grande aumento dos blocos a taxa de acerto tende a diminuir, porque a probabilidade de utilizar os dados buscados recentemente se torna menor do que a probabilidade de reutilizar os dados que foram substituídos.

Projetos de Memória Cache

- **Número de Memórias Cache**
  - Existem dois aspectos importantes do projeto de sistemas com múltiplas memórias cache: o número de níveis e o uso de memórias cache unificadas ou separadas.
  - Com o aumento da capacidade tecnológica dos circuitos integrados, foi possível acoplar a memória cache na mesma pastilha do processador, reduzindo a atividade externa do processador com barramentos, aumentando a velocidade e o desempenho global do sistema.
  - Assim, com o surgimento da cache interna (L1), se tornou possível a utilização de mais uma memória cache no sistema, a cache externa (L2).

Projetos de Memória Cache

- **Número de Memórias Cache**
  - Com o surgimento da cache interna tornou-se comum o uso de duas memórias cache: uma dedicada para dados e outra para instruções
  - Vantagens das memórias cache unificadas:
    - Se um padrão de execução envolve um número muito maior de buscas de instruções do que de busca de dados, a memória cache tende a ser preenchida com instruções, se o padrão envolve maior número de busca de dados ocorre o contrário.
    - Apenas uma memória cache precisa ser projetada e implementada.
  - Vantagens das memórias cache separadas:
    - Elimina a disputa por acesso à memória cache entre o processador de instruções e a unidade de execução.
    - Realiza buscas de instruções a serem processadas antecipadamente.

Projetos de Memória Cache

- **Cache em Níveis**
  - **Cache L1**
    - Uma pequena porção de memória estática presente dentro do processador. Em alguns tipos de processador, como o Pentium 2, o L1 é dividido, em dois níveis: dados e instruções, que "dizem" o que fazer com os dados. A partir do Intel 486, começou a se colocar a L1 no próprio Processador.

Projetos de Memória Cache

- **Cache em Níveis**
  - **Cache L2**
    - Possuindo o Cache L1 um tamanho reduzido e não apresentando uma solução ideal, foi desenvolvido o cache L2, que contém muito mais memória que o cache L1. Ela é mais um caminho para que a informação requisitada não tenha que ser procurada na lenta memória principal.
    - Alguns processadores colocam essa cache fora do processador, por questões econômicas, pois uma cache grande implica num custo grande, mas há exceções, como no Pentium II, por exemplo, cujas caches L1 e L2 estão no mesmo cartucho onde está o processador.

Projetos de Memória Cache

- **Cache em Níveis**
  - **Cache L3**
    - Terceiro nível de cache de memória. Inicialmente utilizado pelo AMD K6-III (por apresentar o cache L2 integrado ao seu núcleo), utilizava o cache externo presente na placa-mãe, como uma memória de cache adicional.
    - É um tipo de cache raro, ainda hoje, mas a complexidade dos processadores atuais, com suas áreas chegando a milhões de transistores por micrômetros de área, ela ainda será muito útil.