Arquitetura e Organização de Computadores

Capítulo 4

Memória cache

Características dos Sistemas de memória do Computador

- Localização.
- Capacidade.
- Unidade de transferência.
- Método de acesso.
- Desempenho.
- Tipo físico.
- Características físicas.
- Organização.

Localização

- Interna:
  - — na CPU: registradores
  - — memória principal: cache e outras.
- Externa: p.e. disco e fita

Capacidade

- Tamanho de palavra (memória interna):
  - —A unidade de organização natural.
  - —Expresso em bits
- Número de palavras (memória externa):
  - —Expresso em Bytes.

Unidade de transferência

- Interna:
  - —Normalmente controlada pela largura do barramento.
- Externa:
  - —Normalmente um bloco que é muito maior que uma palavra.
- Unidade endereçável:
  - —Menor local que pode ser endereçado exclusivamente, também chamada de célula.
  - —N unidades endereçáveis = 2^A , onde A é o tamanho de bits de um endereço.

Métodos de acesso

- Sequencial:
  - —Começa no início e lê em ordem.
  - —Tempo de acesso depende da localização dos dados e local anterior.
  - —Por exemplo, fita.
- Direto:
  - —Blocos individuais possuem endereço exclusivo.
  - —Acesso saltando para vizinhança, mais uma busca sequencial.
  - —Tempo de acesso depende da localização e local anterior.
  - —Por exemplo, disco.

Métodos de acesso

- Aleatório:
  - —Endereços individuais identificam localizações com exatidão.
  - —Tempo de acesso é independente da localização ou acesso anterior e é constante.
  - —P.e., memória principal DRAM e algumas caches.
- Associativo:
  - —Dados são localizados por uma comparação com conteúdo de uma parte do armazenamento e não por um endereço.
  - —Tempo de acesso é independente do local ou acesso anterior e é constante.
  - —P.e., cache.

Desempenho

- Tempo de acesso (latência):
  - —Tempo entre a apresentação do endereço e obtenção dos dados válidos.
- Tempo de ciclo de memória:
  - – Tempo que pode ser exigido para a memória se "recuperar" antes do próximo acesso.
  - —Tempo de ciclo = t. de acesso + recuperação.
- Taxa de transferência:
  - —Taxa em que os dados podem ser movidos.
  - —Para a DRAM é 1/(tempo de ciclo)

Tipos físicos

- Semicondutor:
  —RAM e Flash
- Magnético:
  —Disco e fita.
- Óptico:
  —CD e DVD.
- Outros:
  —Holograma. Novos discos HVD.

Características físicas

- Deterioração.
- Volatilidade.
- Apagável.
- Consumo de energia.

Organização

- Arranjo físico dos bits em palavras.
- Arranjo óbvio nem sempre usado.
- P.e., intercalada.
- P.e., 1MBytes organizados em 2x 512KBytes

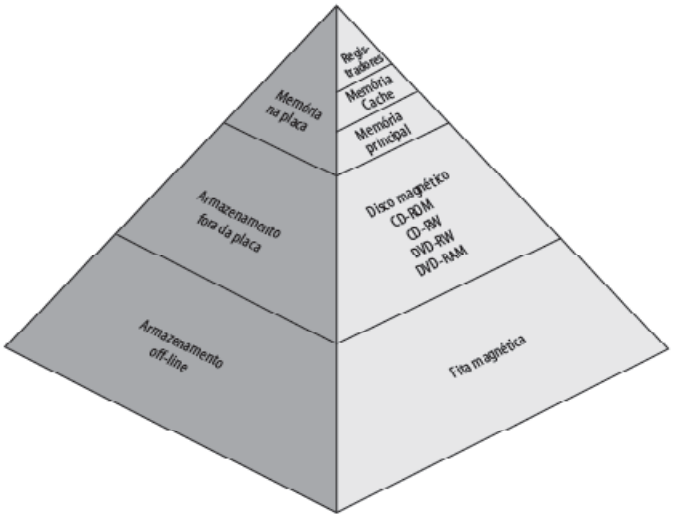
A conclusão

- Quanto?
  —Capacidade.
- Que velocidade?
  —Tempo é dinheiro.
- Com que custo?

Algumas relações

- Tempo de acesso mais rápido, maior custo p/ bit
- Maior capacidade, menor custo p/ bit
- Maior capacidade, tempo de acesso mais lento

Hierarquia de memória – Diagrama



Lista de hierarquia

- Registradores (na CPU).
- Cache L1.
- Cache L2.
- Cache L3
- Memória principal.
- Cache de disco.
- Disco.
- Óptica.
- Fita.

Descendo na hierarquia ocorre:

a) Diminuição do custo p/ bit.
b) Aumento da capacidade.
c) Aumento do tempo de acesso.
d) Diminuição da frequencia de acesso à memória pelo computador

Obs.: a chave é o item d) diminuição da frequência de acesso. Explicação em Memória cache neste capítulo e em memória virtual no capítulo 8.

Então queremos velocidade?

- É possível montar um computador que usa apenas RAM estática (veja adiante).
- Este seria muito rápido.
- Este não precisaria de cache.
- Isso sairia muito caro.

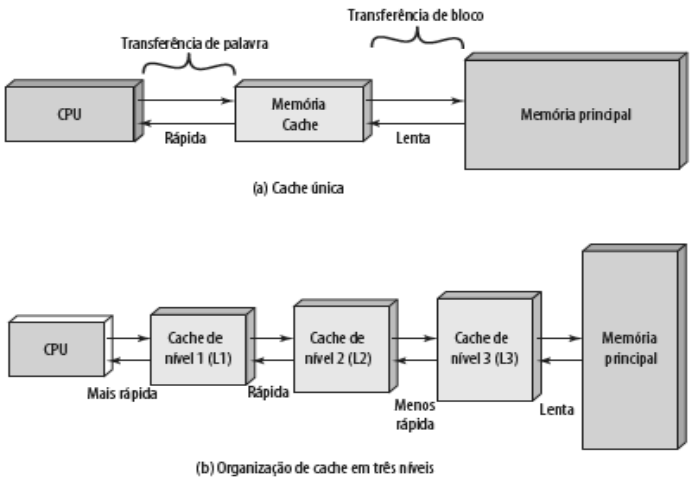
Localidade de referência (Denning, 1968)

- “Durante o curso da execução de um programa, as referências à memória pelo processador, para instruções e dados, tendem a se agrupar”.
- P.e., loops interativos.

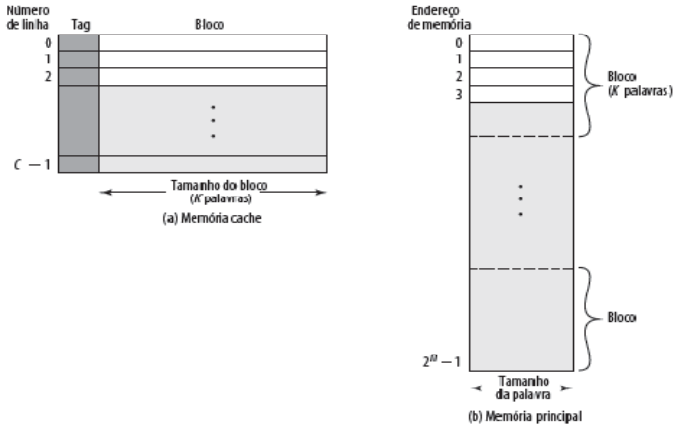
Princípios da memória cache

- Pequena quantidade de memória rápida.
- Fica entre a memória principal grande e lenta e a CPU.
- A cache contém uma cópia de partes da memória principal
- Pode estar localizada no chip da CPU ou módulo.

Cache e memória principal



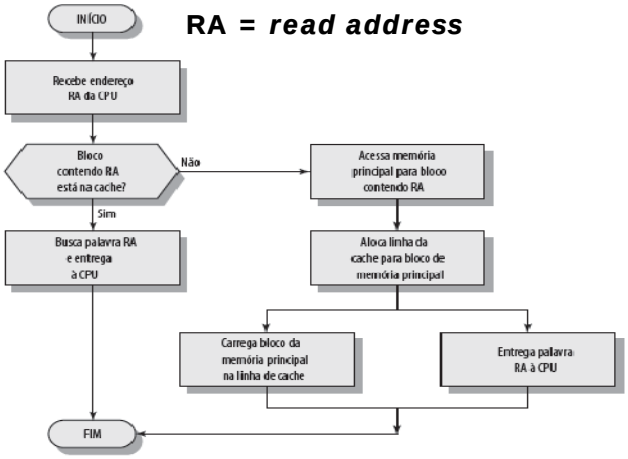
Estrutura de cache/memória principal



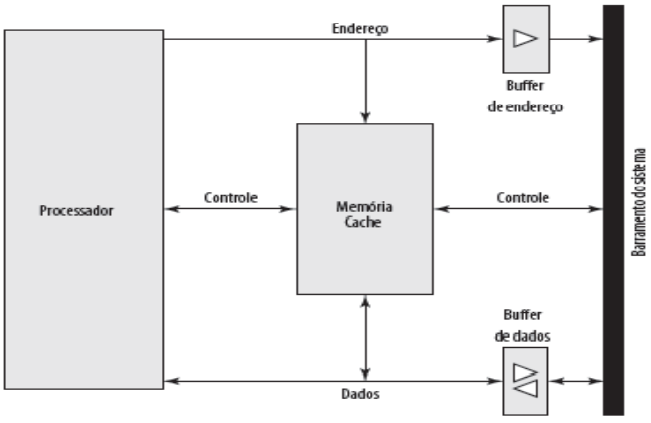
Operação da cache – visão geral

- CPU requisita conteúdo do local de memória.
- Verifica se os dados estão em cache.
- Se estiverem, apanha da cache (rápido).
- Se não, lê bloco solicitado da memória principal para a cache.
- Depois, busca da cache.
- Cache inclui Tags para identificar qual bloco da memória principal está em cada slot da cache.

Operação de leitura de cache – fluxograma



Organização típica da memória cache



Projeto de memória cache

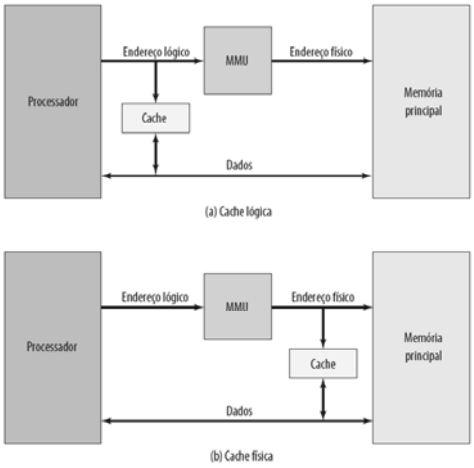
- Endereçamento.
- Tamanho.
- Função de mapeamento.
- Algoritmo de substituição.
- Política de escrita.
- Tamanho de bloco.
- Número de caches.

Endereçamento de cache

- Onde fica a cache?
  - Entre processador e MMU (unidade de gerenciamento de memória virtual). Explicada no capítulo 8.
  - Entre MMU e memória principal.
- Cache lógica (cache virtual) armazena dados usando endereço virtual.
  - Processador acessa a cache diretamente, sem passar pela MMU.
  - Vantagem: Acesso à cache lógica é mais rápido, pois a cache responde antes da tradução de endereço da MMU.
  - Endereços virtuais usam o mesmo espaço de endereços para diferentes aplicações.
    - Deve esvaziar cache a cada troca de contexto.
- Cache física armazena dados usando endereços físicos da memória principal.

Caches lógicas e físicas

Figura 4.7 Caches lógicas e físicas



Tamanho ideal da cache é impossível

Importa:

- Custo
  - —Mais cache é caro.
- Velocidade:
  - —Mais cache é mais rápido (até certo ponto).
  - —Verificar dados na cache leva tempo.

Comparação de tamanhos de memória cache

Processador	Tipo	Ano de introdução	Cache L1[a]	Cache L2	Cache L3
IBM 360/85	Mainframe	1968	16 a 32 KB	—	—
PDP-11/70	Minicomputador	1975	1 KB	—	—
VAX 11/780	Minicomputador	1978	16 KB	—	—
IBM 3033	Mainframe	1978	64 KB	—	—
IBM 3090	Mainframe	1985	128 a 256 KB	—	—
Intel 80486	PC	1989	8 KB	—	—
Pentium	PC	1993	8 KB/8 KB	256 a 512 KB	—
PowerPC 601	PC	1993	32 KB	—	—
PowerPC 620	PC	1996	32 KB/32 KB	—	—
PowerPC G4	PC/servidor	1999	32 KB/32 KB	256 KB a 1 MB	2 MB
IBM S/390 G4	Mainframe	1997	32 KB	256 KB	2 MB
IBM S/390 G6	Mainframe	1999	256 KB	8 MB	—
Pentium 4	PC/servidor	2000	8 KB/8 KB	256 KB	—
IBM SP	Servidor avançado/ Supercomputador	2000	64 KB/32 KB	8 MB	—
CRAY MTA[b]	Supercomputador	2000	8 KB	2 MB	—
Itanium	PC/servidor	2001	16 KB/16 KB	96 KB	4 MB
SGI Origin 2001	Servidor avançado	2001	32 KB/32 KB	4 MB	—
Itanium 2	PC/servidor	2002	32 KB	256 KB	6 MB
IBM POWER5	Servidor avançado	2003	64 KB	1,9 MB	36 MB
CRAY XD-1	Supercomputador	2004	64 KB/64 KB	1 MB	—
IBM POWER6	PC/servidor	2007	64 KB/64 KB	4 MB	32 MB
IBM z10	Mainframe	2008	64 KB/128 KB	3 MB	24 a 48 MB

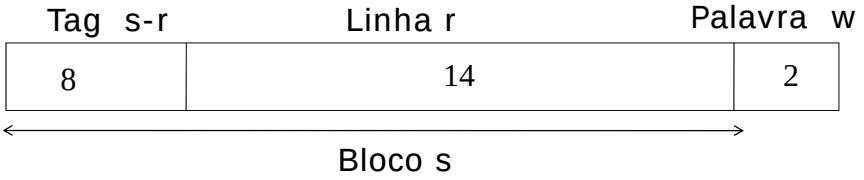
Função de mapeamento - Exemplo

- Cache de 64 KB.
- Bloco de cache de 4 bytes.
  - —Ou seja, cache é de 16k (2^{14}) linhas de 4 bytes.
- 16 MB de memória principal.
- Endereço direto de 24 bits.
  - —(2^{24}=16MBytes)
- Memória principal tem 4M blocos de 4 bytes

Mapeamento direto

- **Cada bloco de memória principal mapeado apenas para uma linha de cache possível**.
  - —Ou seja, se um bloco está na cache, ele deve estar em um local específico.
- **w** bits menos significativos identificam word exclusiva.
- **s** bits mais significativos especificam um bloco de memória.
- Lógica de cache interpreta os **s** bits em 2 partes: um campo de **linha de cache** r e um campo de **tag** de s-r (parte mais significativa).

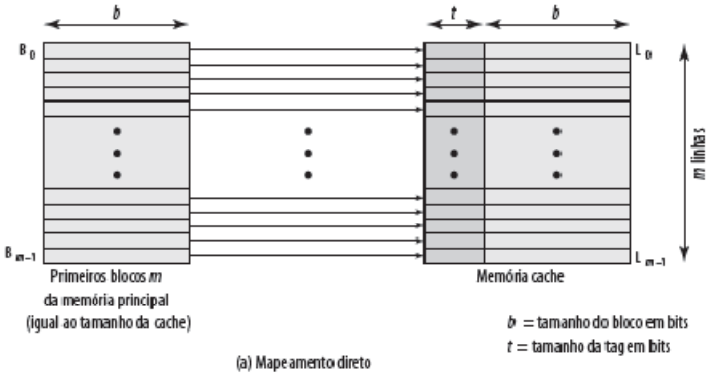
Mapeamento direto (exemplo) Estrutura de endereços



- Endereço da memória principal de 24 bits (2^{24} =16 MB).
- Identificador de **palavra** de 2 bits (blocos de 4 bytes).
- Identificador de **bloco** de 22 bits (4M blocos).
  - —**Linha de cache** de 14 bits (16K linhas).
  - —**Tag** de 8 bits (=22-14).
- Dois blocos na mesma linha não têm o mesmo campo de tag.
- Verifica-se o conteúdo da cache localizando linha e verificando tag.

Mapeamento direto da cache para memória principal

Cada bloco de memória principal mapeado apenas para uma linha de cache possível.

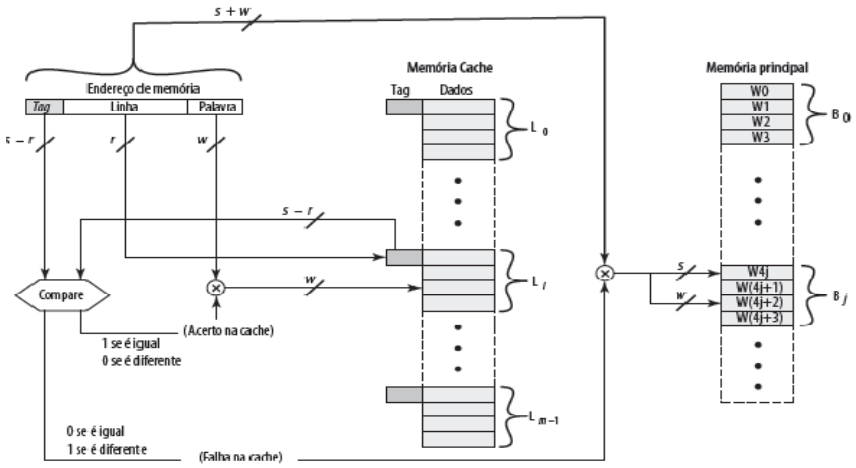


(a) Mapeamento direto

Mapeamento direto
Tabela de linhas de cache

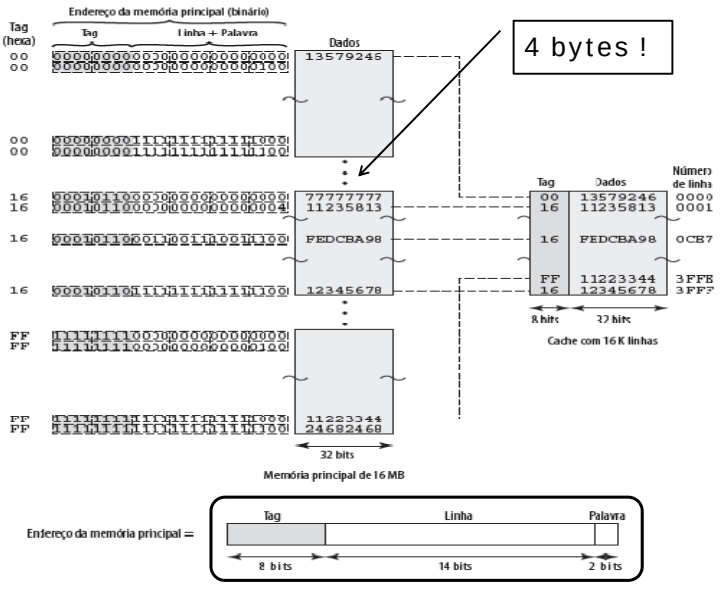
Linha de cache (m linhas)	Blocos de memória principal mapeados
0	0, m, 2m, 3m…2^s-m
1	1,m+1, 2m+1…2^s-m+1
…	
m-1	m-1, 2m-1,3m-1…2^s-1

Organização da cache com mapeamento direto



Lógica de cache

Exemplo de mapeamento direto



Resumo de mapeamento direto

- Tamanho de endereço= (s + w) bits.
- Número de unidades endereçáveis= 2^{s+w} palavras ou bytes.
- Tamanho de bloco= tamanho de linha= 2^{w} words ou bytes.
- Número de blocos na memória principal= 2^{s}.
- Número de linhas na cache = m = 2^{r}.
- Tamanho da tag= (s – r) bits.

Prós e contras do mapeamento direto

- Simples.
- Barato.
- Local fixo para determinado bloco, isto é, cada bloco sempre será carregado na mesma linha de cache.
  - —Mas, se um programa referenciar palavras repetidamente de dois blocos diferentes, então os blocos serão continuamente trocados e razão de acerto será baixa (*trashing*).

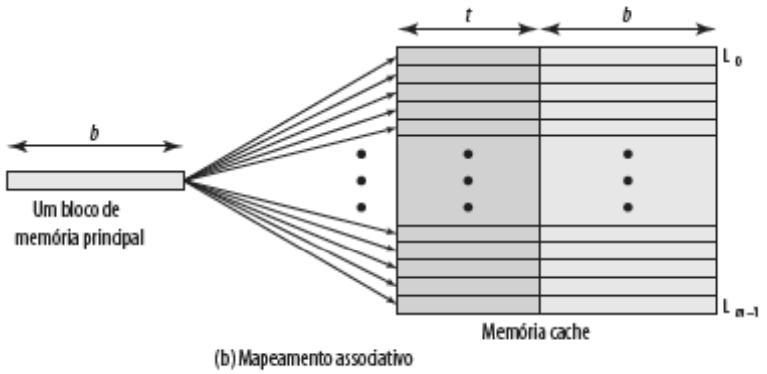
Cache vítima (uma técnica)

- Guarda o que foi descartado.
  - —Já buscado.
  - —Usa novamente com pouca penalidade.
- Menor penalidade de falha.
- Totalmente associativa.
- Normalmente de 4 a 16 linhas de cache.
- Entre cache L1 (mapeada diretamente) e nível de memória seguinte.

Mapeamento associativo

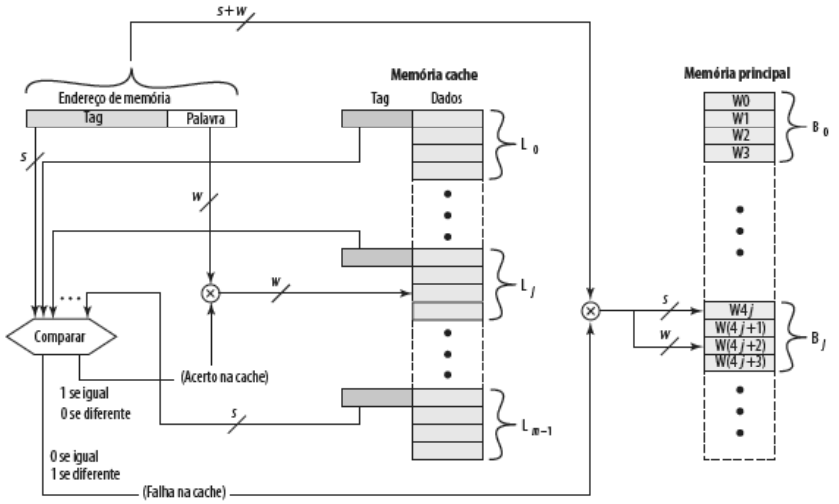
- Um bloco de memória principal pode ser carregado em qualquer linha de cache.
- Endereço de memória é interpretado como tag e palavra.
- Tag identifica exclusivamente o bloco de memória.
- Tag de cada linha é examinada em busca de combinação.
- Pesquisa da cache é dispendiosa.

Mapeamento associativo da cache para a memória principal



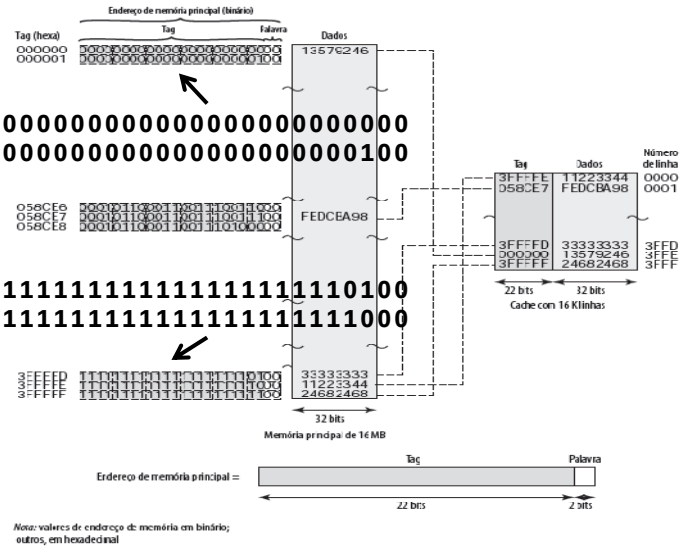
(b) Mapeamento associativo

Organização de cache totalmente associativa



Lógica de cache

Exemplo de mapeamento associativo



Mapeamento associativo
Estrutura de endereço

Tag	Palavra w
22	2

- Tag de 22 bits armazenado a cada bloco de 32 bits de dados.
- Compara campo de tag do endereço com entrada de tag na cache para procurar acerto.
- 2 bits menos significativos do endereço identificam qual word de 16 bits é exigida do bloco de dados de 32 bits.
- P.e. (no slide anterior):

	Endereço	Tag	Dados	Linha de cache
—	FFFFFC	3FFFFF	24682468	3FFF

Resumo do mapeamento associativo

- Tamanho do endereço= (s + w) bits.
- Número de unidades endereçáveis= 2^{s+w} words ou bytes.
- Tamanho do bloco= tamanho de linha= 2^{w} palavras ou bytes.
- Número de blocos na memória principal= $2^{s+w}/2^{w} = 2^{s}$.
- Número de linhas na cache= indeterminado.
- Tamanho da tag= s bits.
- **Maximiza a razão de acerto, mas circuito de comparação de Tags é complexo**

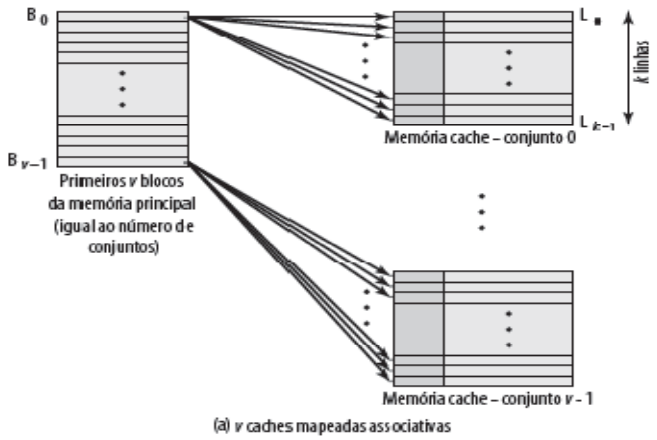
Mapeamento associativo em conjunto

- Cache é uma série de conjuntos.
- Cada conjunto contém uma série de linhas.
- Determinado bloco é mapeado a qualquer linha em determinado conjunto.
  - —P.e., bloco B pode estar em qualquer linha do conjunto i.
- P.e., 2 linhas por conjunto:
  - —Mapeamento associativo com 2 linhas.
  - —Determinado bloco pode estar em uma de 2 linhas em apenas um conjunto.

Mapeamento associativo em conjunto Exemplo

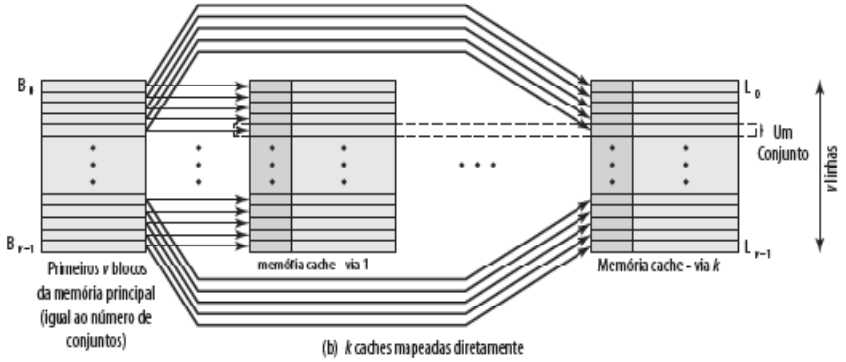
- Número de conjunto com 13 bits
- Número de bloco na memória principal é módulo 2^{13}
- 000000, 008000, … , FF8000 mapeados no mesmo conjunto

Mapeamento da memória principal para cache: associativo com *v* caches

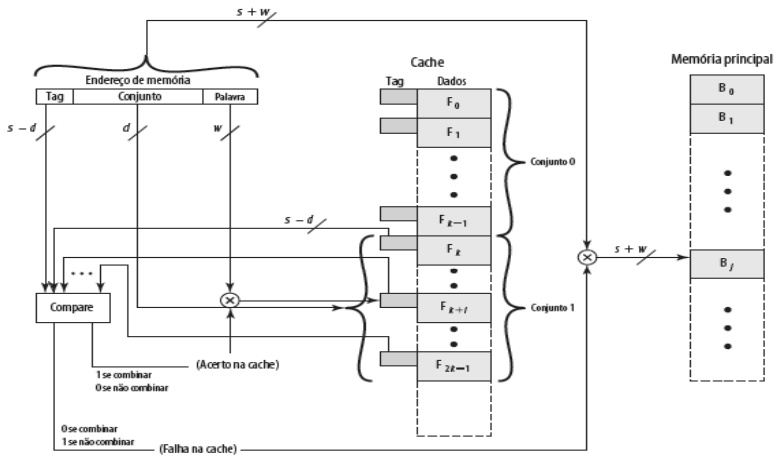


(a) *v* caches mapeadas associativas

Mapeamento da memória principal para cache: associativo c/ *k* caches mapeadas diretamente



Organização da cache associativa em conjunto com *k* linhas por conjunto



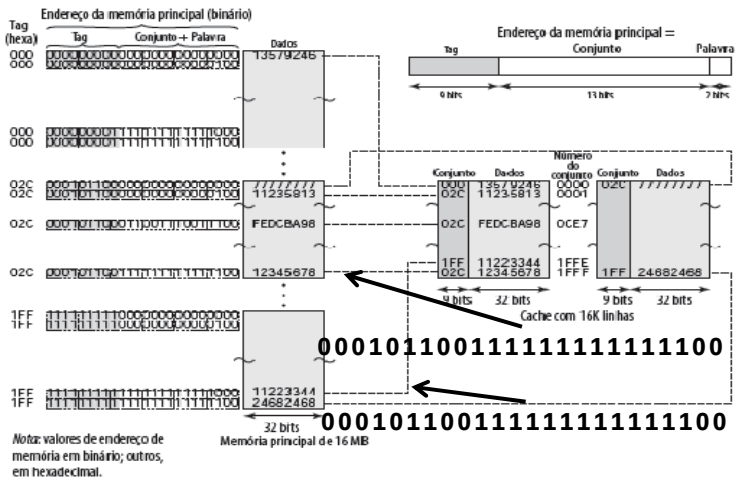
Mapeamento associativo em conjunto
Estrutura de endereços

Tag	Conjunto	Palavra w
9	13	2

- Usa campo de conjunto para determinar conjunto de cache a examinar.
- Compara campo de tag para ver se há um acerto.
- P.e. (ver próximo slide),

—Endereço	Tag	Dados	Conjunto
—FFFFF8	1FF	11223344	1FFE
—167FFC	02C	12345678	1FFF

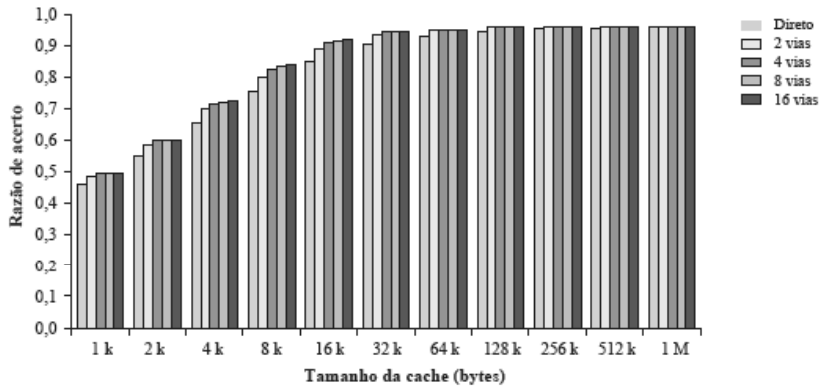
Exemplo de mapeamento associativo em conjunto com duas linhas



Resumo de mapeamento associativo em conjunto

- Tamanho do endereço= (s + w) bits.
- Número de unidades endereçáveis= 2^{s+w} palavras ou bytes.
- Tamanho do bloco= tamanho da linha= 2^{w} palavras ou bytes.
- Número de blocos na memória principal= 2^{d}.
- Número de linhas no conjunto= k.
- Número de conjuntos= v= 2^{d}.
- Número de linhas na cache= kv= k * 2^{d}.
- Tamanho da tag= (s – d) bits.
- **Tag é muito menor que no associativo simples e só é comparada com as k tags de um único conjunto**

Associatividade variável pelo tamanho da cache



Cache direta e associativa em conjunto

Diferenças de desempenho

- Significativo até pelo menos 64KB para 2 linhas.
- Diferença entre 2 e 4 linhas em 4 KB muito menor do que ao passar de 4 KB para 8 KB.
- Complexidade da cache aumenta com a associatividade.
- Complexidade não justificada contra o aumento da cache para 8kB ou 16kB.
- Acima de 32 KB não gera melhoria.
- (resultados de simulação)

O que fazer em caso de cache cheia ?
Algoritmos de substituição (de blocos)

Mapeamento direto

- Sem escolha.
- Cada bloco mapeado apenas a uma linha.
- Essa linha será substituída.

Algoritmos de substituição

Associativa e associativa em conjunto

- Algoritmo implementado no hardware (velocidade).
- Least Recently Used (LRU).
  - —Substitui o bloco usado recentemente
- P.e., na associativa em conjunto com 2 linhas.
  - —Qual dos 2 blocos é LRU?
- First In First Out (FIFO).
  - —Substitui bloco que está na cache há mais tempo.
- Least Frequently Used (LFU).
  - —Substitui bloco que teve menos acertos.
- Aleatório.

Política de escrita

- Não se deve sobrescrever bloco de cache a menos que a memória principal esteja atualizada.
- Múltiplas CPUs podem ter caches individuais.
- E/S pode endereçar memória principal diretamente.

Write-through

- Todas as escritas vão para a memória principal e também para a cache.
- Múltiplas CPUs podem monitorar o tráfego da memória principal para manter a cache local (à CPU) atualizada.
- Isso gera muito tráfego e atrasa as escritas.
- Pode haver caches *write-through* falsos!

Write-back

- Atualizações feitas inicialmente apenas na cache.
- Bit de atualização para slot de cache é definido quando ocorre a atualização.
- Se o bloco deve ser substituído, escreve na memória principal apenas se o bit atualizado estiver marcado.
- Outras caches saem de sincronismo.
- E/S deve acessar a memória principal através da cache.
- 15% das referências de memória são escritas.
- Para sistemas de alto desempenho (HPC) pode chegar a até 50%.

Tamanho de linha

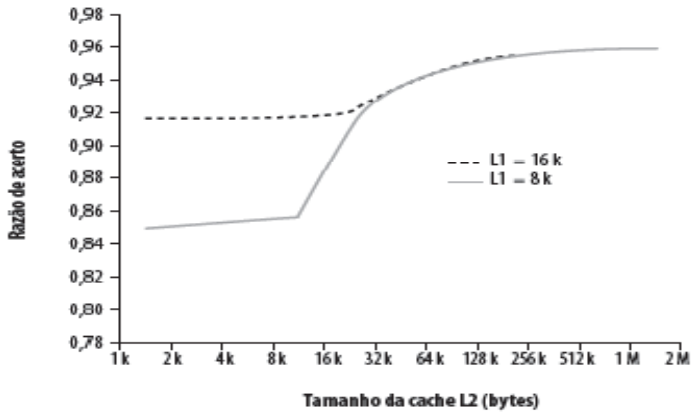
- Recuperar não apenas a palavra desejada, mas também uma série de palavras adjacentes.
- Tamanho de bloco aumentado aumentará razão de acerto a princípio.
  - —O princípio da localidade.
- Razão de acerto diminuirá à medida que o bloco se tornar ainda maior.
  - —Probabilidade de uso de informações recém-buscadas torna-se menor que probabilidade de reutilizar informações substituídas.

- Blocos maiores:
  - —Reduzem número de blocos que cabem na cache.
  - —Dados sobrescritos pouco depois de serem buscados.
  - —Cada palavra adicional é menos local, de modo que é menos provável de ser necessária.
- Nenhum valor ideal definitivo foi descoberto.
- 8 a 64 bytes parece ser razoável.
- Para sistemas HPC, 64 e 128 bytes mais comum.

Caches multinível

- Alta densidade lógica permite caches no chip.
  - —Mais rápido que acesso ao barramento.
  - —Libera barramento para outras transferências.
- Comum usar cache dentro e fora do chip.
  - —L1 no chip, L2 fora do chip na RAM estática.
  - —Acesso L2 muito mais rápido que DRAM ou ROM.
  - —L2 normalmente usa caminho de dados separado.
  - —L2 pode agora estar no chip.
  - —Resultando em cache L3.
    - – Acesso ao barramento agora no chip.

Razão de acerto total (L1 & L2)
Para L1 de 8 KB e 16 KB



Caches unificadas *versus* separadas

- Uma cache para dados e instruções ou duas, uma para dados e uma para instruções.
- Vantagens da cache unificada:
  - Maior taxa de acerto.
    - Equilibra carga entre buscas de instrução e dados.
    - Apenas uma cache para projetar e implementar.
- Vantagens da cache separada:
  - Elimina disputa pela cache entre a unidade de busca/decodificação de instrução e a unidade de execução.
    - Importante quando existe pipeline de instruções.

Pentium 4 – cache

- 80386 – nenhuma cache no chip.
- 80486 – 8 KB usando linhas de 16 bytes organização associativa em conjunto com 4 linhas.
- Pentium (todas as versões) – duas caches L1 no chip.
  - Dados e instruções:
- Pentium III – cache L3 adicionada fora do chip.
- Pentium 4:
  - Caches L1.
    - 8 KB.
    - Linhas 64 bytes.
    - Associativa em conjunto com 4 linhas.

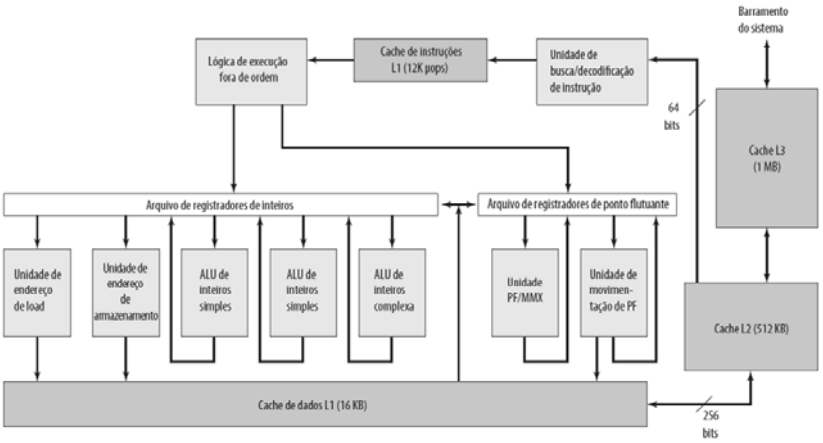
—Cache L2:
  – Alimentando ambas as caches L1.
  – 256k.
  – Linhas de 128 bytes.
  – Associativa em conjunto com 8 linhas.
—Cache L3 cache no chip.

Evolução de cache da Intel

Problema	Solução	Processador em que o recurso apareceu inicialmente
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 à memória cache..	Mover a cache externa para o chip, trabalhando na mesma velocidade do processador	486
Cache interna um tanto pequena, devido ao espaço limitado no 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 simultâneo à memória cache. Nesse caso, a busca antecipada é adiada até o término do acesso da unidade de execução aos dados.	Criar caches separadas para dados e instruções	Pentium
Maior velocidade do processador torna o barramento externo um gargalo para o acesso à cache L2.	Criar barramento back-side separado, que trabalha com velocidade mais alta que o barramento externo principal (front-side). O barramento back-side é dedicado à cache L2.	Pentium Pro
	Mover cache L2 para o chip do processador.	Pentium II
Algumas aplicações lidam com bancos de dados enormes, e precisam ter acesso rápido a grandes quantidades de dados. As caches no chip são muito pequenas.	Acrescentar cache L3 externa.	Pentium III
	Mover cache L3 para o chip.	Pentium 4

Diagrama em blocos do Pentium 4

Figura 4.18 Diagrama em blocos do Pentium 4



Processador Pentium 4 Core

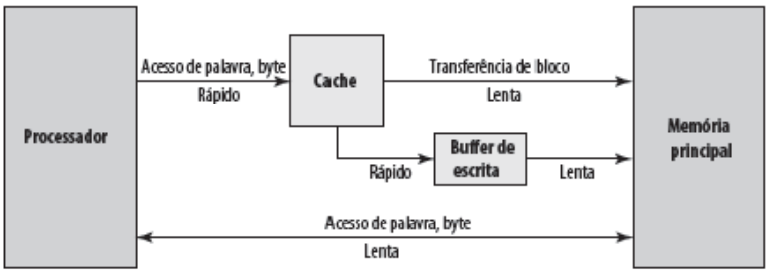
- Unidade de busca/decodificação:
  - Busca instruções da cache L2.
  - Decodifica para micro-operações.
  - Armazena micro-operações na cache L1.
- Lógica de execução fora de ordem:
  - Escalona micro-operações.
  - Baseada em dependência de dados e recursos.
  - Pode executar especulativamente.
- Unidades de execução:
  - Executa micro-operações.
  - Dados da cache L1.
  - Resultados em registradores.
- Subsistema de memória.
  - Cache L2 e barramento do sistema.

Raciocínio de projeto do Pentium 4

- Decodifica instruções para RISC como micro-operações antes da L1.
- Micro-operações de tamanho fixo.
  - —Pipelining e escalonamento superescalar.
- Instruções Pentium longas e complexas.
- Desempenho melhorado separando decodificação do escalonamento e pipelining.
  - —(Mais adiante – Capítulo 14)

- Cache de dados é *write-back.*
  - —Pode ser configurada para *write-through*.
- Cache L1 controlada por 2 bits no registrador.
  - —CD= Cache Disable.
  - —NW= Not *write-through*.
  - —2 instruções para invalidar (esvaziar) cache e *write-back* depois invalidação.
- L2 e L3 associativas em conjunto com 8 linhas.
  - —Tamanho de linha 128 bytes.

Organização da cache e do buffer de escrita do ARM



Organização de cache da ARM

- Pequeno buffer de escrita FIFO.
  - —Melhora o desempenho de escrita da memória.
  - —Entre cache e memória principal.
  - —Pequena cache, conforme figura
  - —Dados colocados no buffer de escrita na velocidade de clock do processador.
  - —Processador continua a execução.
  - —Escrita externa em paralelo até que esteja vazio.
  - —Se buffer encher, processamento adiado (*stall*).
  - —Dados no buffer de escrita não disponíveis até serem escritos na memória principal.
    - – Daí, buffer de escrita ser pequeno.

Características da memória cache do ARM

Núcleo	Tipo de cache	Tamanho de cache (kB)	Tamanho da linha de cache (palavras)	Associatividade (linhas por conjunto)	Local	Tamanho do buffer de escrita (palavras)
ARM720T	Unificada	8	4	4 linhas por conjunto	Lógico	8
ARM920T	Separada	16/16 D/I	8	64 linhas por conjunto	Lógico	16
ARM926EJ-S	Separada	4-128/4-128 D/I	8	4 linhas por conjunto	Lógico	16
ARM1022E	Separada	16/16 D/I	8	64 linhas por conjunto	Lógico	16
ARM1026EJ-S	Separada	4-128/4-128 D/I	8	4 linhas por conjunto	Lógico	8
Intel StrongARM	Separada	16/16 D/I	4	32 linhas por conjunto	Lógico	32
IntelXscale	Separada	32/32 D/I	8	32 linhas por conjunto	Lógico	32
ARM1136-JF-S	Separada	4-64/4-64 D/I	8	4 linhas por conjunto	Físico	32

Fontes na Internet

- Sites de fabricantes:
  - —Intel.
  - —ARM.
- Procure sobre cache.