

QXD0133 - Arquitetura e Organização de Computadores II



Universidade Federal do Ceará - Campus Quixadá

Thiago Werlley
thiagowerlley@ufc.br

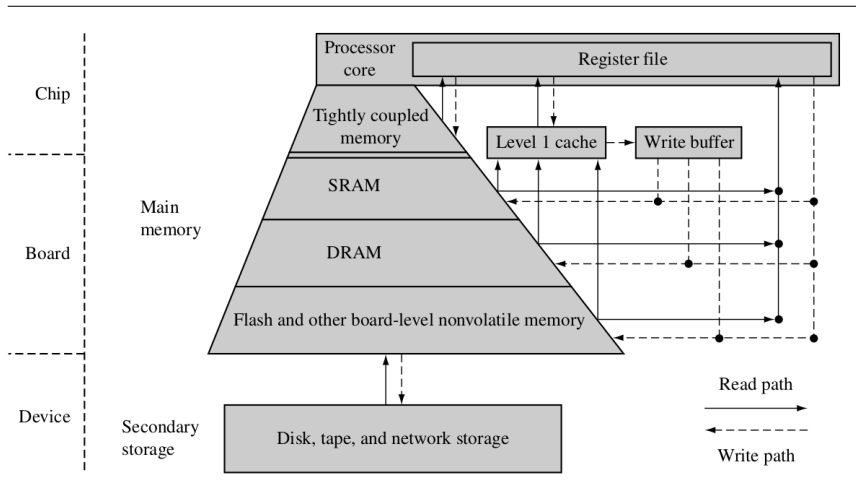
18 de outubro de 2025

Capítulo 12

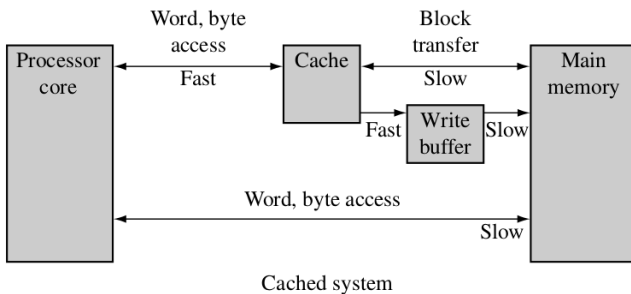
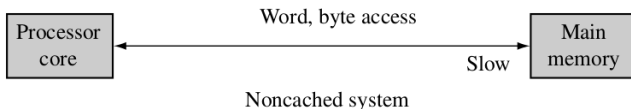
Capítulo 12

Sistema de memória do ARM (Cache)

Introdução



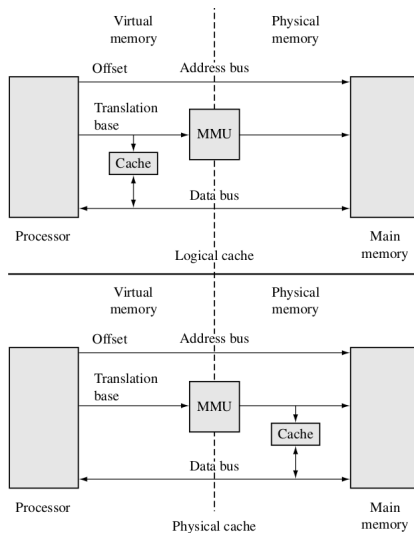
Introdução



Cache e unidade de gerenciamento de memória

- Uma cache lógica armazena os dados em um espaço de memória virtual.
- A cache lógica é localizada entre o processador e a MMU.
- A cache física armazena os dados usando endereço físico de memória.
- A cache física é localizada entre a MMU e a memória principal.

Cache Física e Local



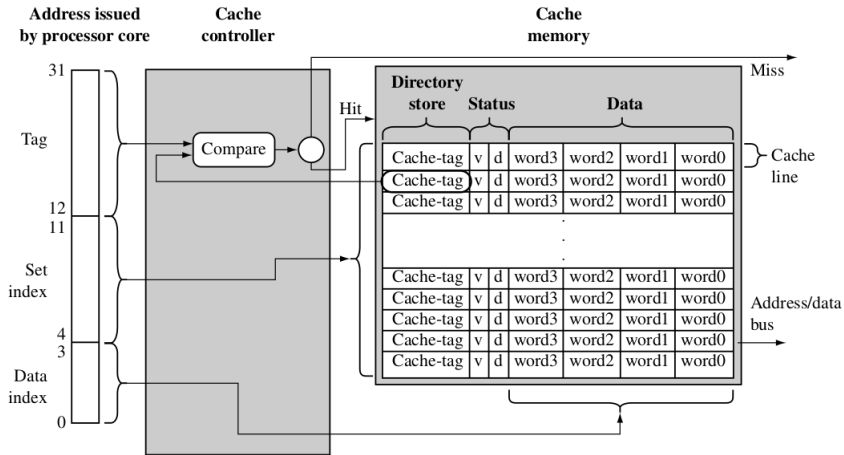
Princípio da localidade

- A melhoria oferecida por uma cache é possível porque os programas de computador são executados de maneiras não aleatórias.
- O princípio da localidade temporal diz que um dado acessado recentemente tem mais chances de ser usado novamente, do que um dado usado há mais tempo.
- O princípio da localidade espacial diz que há uma probabilidade de acesso maior para dados e instruções em endereços próximos àqueles acessados recentemente.

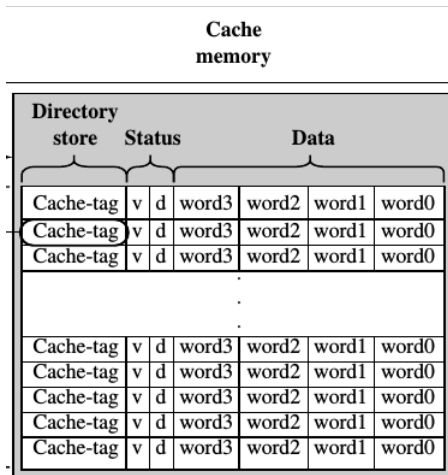
Arquitetura da Cache

- O ARM usa duas arquiteturas de barramento em seus núcleos em cache, o Von Neumann e o Harvard.
- Nos núcleos do processador que usam a arquitetura Von Neumann, há um único cache usado para instruções e dados - `cache unificado`.
- Nos núcleos do processador que usam a arquitetura Harvard, existem dois caches: um cache de instruções (`cache I`) e um cache de dados (`cache D`) - `cache dividida`.

Arquitetura da cache



Cache Line



- **Directory store**

- Endereço da memória principal de onde os dados da linha foram copiados
- Cache tag

- **Status**

- Bit v → Valid (dado disponível)
- Bit d → Dirty (conteúdo diferente a memória)

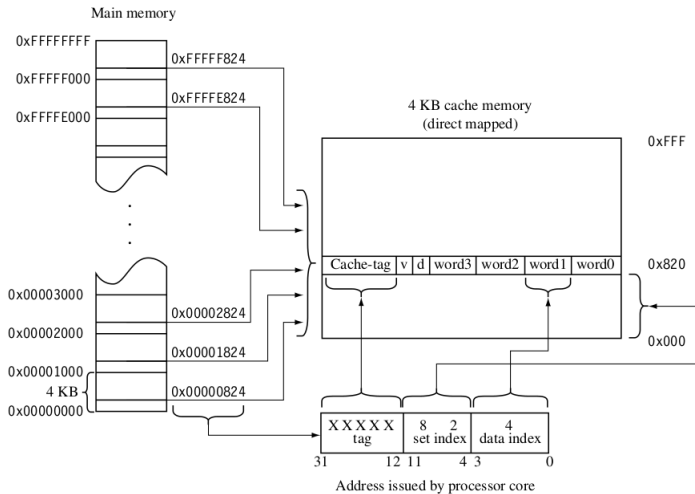
- **Data**

- Dados copiados da memória Tamanho da linha

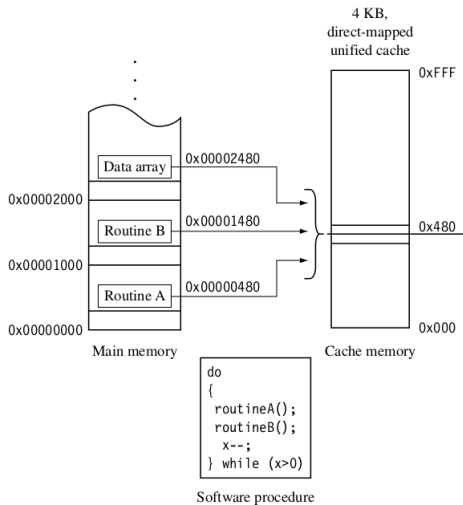
Operações básicas de um controlador de cache

- O controlador de cache é um hardware que copia código ou dados da memória principal para armazenar em cache automaticamente.
- Ele executa essa tarefa automaticamente para ocultar a operação de cache do software que ele suporta.
- O controlador de cache intercepta solicitações de memória de leitura e gravação antes de transmiti-las ao controlador de memória.

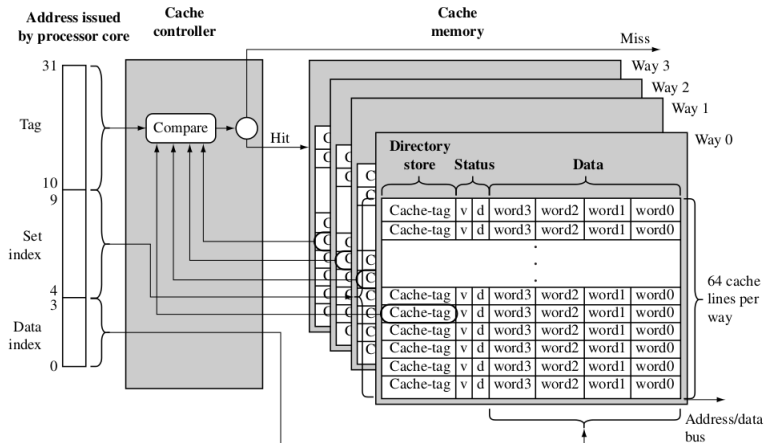
Cache - Mapeamento direto



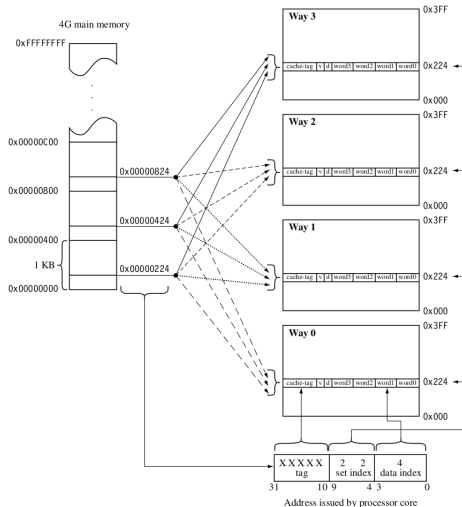
Thrashing



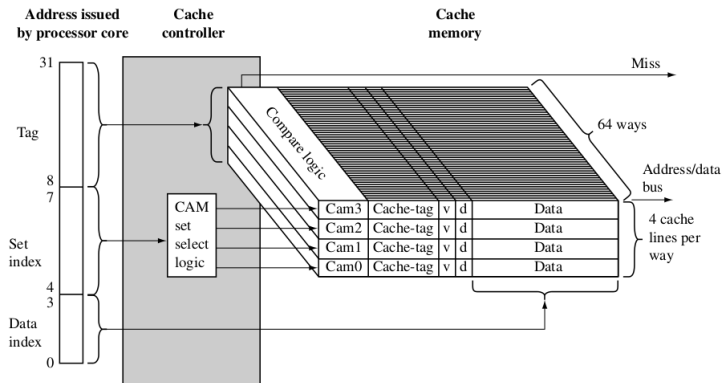
Associativa em conjunto



Associativa em conjunto



Cache Associativa



Eficiência de cache

$$\textit{hit rate} = \left(\frac{\textit{cache hits}}{\textit{memory requests}} \right) \times 100$$

Buffer de Escrita

- Ele permite uma redução no tempo do processador gasto para gravar pequenos blocos de dados seqüenciais na memória principal.
- A memória FIFO do buffer de escrita está no mesmo nível na hierarquia de memória que o cache L1.
- Os dados gravados no buffer de escrita não estão disponíveis para leitura até que tenham saído do buffer de escrita na memória principal.
- Esse é um dos motivos pelos quais a profundidade FIFO de um buffer de escrita é geralmente bastante pequena, apenas algumas linhas de cache.

Buffer de Escrita

- Um `buffer de escrita` é um buffer de memória FIFO rápido muito pequeno que retém temporariamente os dados que o processador normalmente gravaria na memória principal.
- Em um sistema sem o buffer de escrita, o processador grava diretamente na memória principal.
- Em um sistema com o buffer de escrita, os dados são gravados em alta velocidade no FIFO e depois esvaziados para a memória principal mais lenta.

Politica de Cache

- Existem três políticas que determinam a operação de uma cache:
 - política de escrita;
 - política de substituição;
 - política de alocação.

Política de escrita

- Leituras são muito mais numerosas que escritas ($\approx 21\%$ das instruções que acedem à cache de dados).
- Existem duas grandes estratégias de escrita (assumindo que o bloco está presente em cache):
 - *write through* - A informação é escrita simultaneamente em **cache** e na **memória** do nível seguinte.
 - *write back* - A informação é escrita **apenas em cache**: o bloco modificado é escrito em memória quando é substituído.
- É usado um bit (por bloco) para indicar se o bloco em cache foi modificado: *dirty bit*.
- Um bloco não modificado (limpo) não precisa de ser escrito em memória em uma substituição.

Política de escrita

- Vantagens de *write-back*:
 - escritas à velocidade da cache;
 - múltiplas escrita no mesmo bloco requerem apenas um acesso à memória (reduz consumo de largura de banda e de potência).
- Vantagens de *write-through*:
 - memória cache sempre “limpa” (uma falha de leitura não resulta em escritas na memória de nível seguinte);
 - o nível seguinte mantém sempre a versão mais recente dos dados (facilita coerência de memória).
- Com *write-through*, o CPU pode protelar à espera que uma escrita termine - usar um *write buffer*.
- Comportamento numa falha de escrita:
 - *read-write-allocate* - um bloco é alocado para guardar o novo valor;
 - *read-allocate* - apenas a memória de nível seguinte é atualizada; a cache não é afetada.

Política de substituição de linha de cache

- Quando ocorre um cache miss o controlador de cache deve selecionar uma linha de cache do conjunto disponível para armazenar as novas informações da memória principal.
- A linha de cache selecionada para substituição é conhecida como vítima.
- Se a vítima contiver dados válidos e sujos, o controlador deverá gravar os dados sujos da memória cache na memória principal antes de copiar novos dados na linha de cache da vítima.

Política de substituição de linha de cache

- O processo de seleção e substituição de uma linha de cache da vítima é conhecido como despejo.
- A estratégia implementada em um controlador de cache para selecionar a próxima vítima é chamada de política de substituição.
- A política de substituição seleciona uma linha de cache do conjunto de membros associativos disponíveis

Política de substituição de linha de cache

A política `round-robin` simplesmente seleciona a próxima linha de cache.

- O algoritmo de seleção usa um contador de vítimas sequencial e incremental que é incrementado cada vez que o controlador de cache aloca uma linha de cache.
- Quando o contador da vítima atinge um valor máximo, ele é redefinido para um valor base definido.

Política de substituição de linha de cache

A política de substituição pseudo-aleatória seleciona aleatoriamente a próxima linha de cache.

- o controlador incrementa o contador da vítima selecionando aleatoriamente um valor de incremento e adicionando esse valor ao contador da vítima.
- Quando o contador da vítima atinge um valor máximo, ele é redefinido para um valor base definido.

Política de substituição de linha de cache

- A maioria dos processadores ARM oferece suporte a ambas as políticas.
- A política de substituição round-robin tem maior previsibilidade, o que é desejável em um sistema embarcado.
- No entanto, uma política de substituição round-robin está sujeita a grandes mudanças no desempenho devido a pequenas mudanças no acesso à memória.

QXD0133 - Arquitetura e Organização de Computadores II



Universidade Federal do Ceará - Campus Quixadá

Thiago Werlley
thiagowerlley@ufc.br

18 de outubro de 2025

Capítulo 12