

**Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Unidade Acadêmica de Sistemas e Computação  
Curso de Bacharelado em Ciência da Computação**

# **Organização e Arquitetura de Computadores I**

**(Memória - Parte I)**

**Profa. Joseana Macêdo Fechine Régis de Araújo**  
[joseana@computacao.ufcg.edu.br](mailto:joseana@computacao.ufcg.edu.br)

**Carga Horária: 60 horas**



# Tópicos

## Organização Básica de Computadores

### Composição (memória)

# Memória - Composição

## Composição básica de um computador digital

- Processador
- **Memória**
- Dispositivos de entrada e saída interligados

# Memória - Composição

Um exemplo:



# Memória - Composição

## Memória de Acesso Aleatório

- ❑ RAM (*Random Access Memory*)
  - acesso aleatório
  - qualquer posição de memória acessada no mesmo intervalo de tempo
  
- ❑ RAM estática (SRAM)
  - baseada em flip-flops
  - conteúdo persiste enquanto circuito alimentado
  - mais rápida
  
- ❑ RAM dinâmica (DRAM)
  - baseada em capacitores
  - carga deve ser restaurada periodicamente
  - menor, mais econômica

# Memória - Composição

## Memórias Somente de Leitura

- ❑ ROM (*Read-Only Memory*)
  - Simples: decodificador, linhas de saída e portas lógicas
  - Aplicações de grande volume
- ❑ PROM (*Programmable ROM*)
  - Baixo volume e protótipos
  - Conteúdo escrito com um queimador de PROM
- ❑ EPROM (*Erasable PROM*)
  - Apagada por luz ultra-violeta
- ❑ EEPROM (*Electrically Erasable PROM*)
- ❑ ROM Flash

Obs.: ROM pode ser um tipo de RAM.

# Memória - Composição

## Problemas (Processador – Memória)

- ❑ Acesso à memória, leitura e escrita, é um dos motivos para menor velocidade de processamento.
- ❑ Processador é muito mais rápido do que a transferência de dados.

## Soluções:

- ❑ Processador deve executar outras instruções enquanto aguarda acesso à memória. Isto nem sempre é possível e é difícil de implementar.
- ❑ Colocar memória principal no Chip do processador. Isto tornaria o chip maior e mais caro.
- ❑ Uso de uma **memória menor e mais rápida** (em relação à memória principal) chamada Memória Cache.

- *Cache*, em inglês: lugar seguro para esconder ou guardar algo.
- *Cacher*, em francês: esconder, guardar.

# Memória - Composição

## Memória Cache

### Princípio da localidade

- **Localidade temporal**
  - Uma posição de memória referenciada recentemente tem boas chances de ser referenciada novamente
  - iterações e recursividade
- **Localidade *espacial***
  - Uma posição de memória vizinha de uma posição referenciada recentemente tem boas chances de ser referenciada
  - dados tendem a ser armazenados em posições contíguas

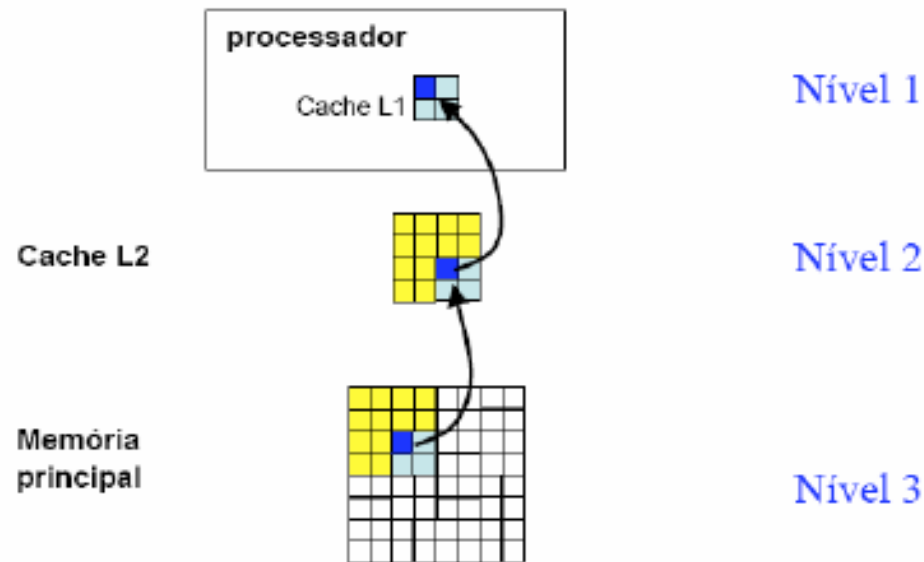


# Memória - Composição

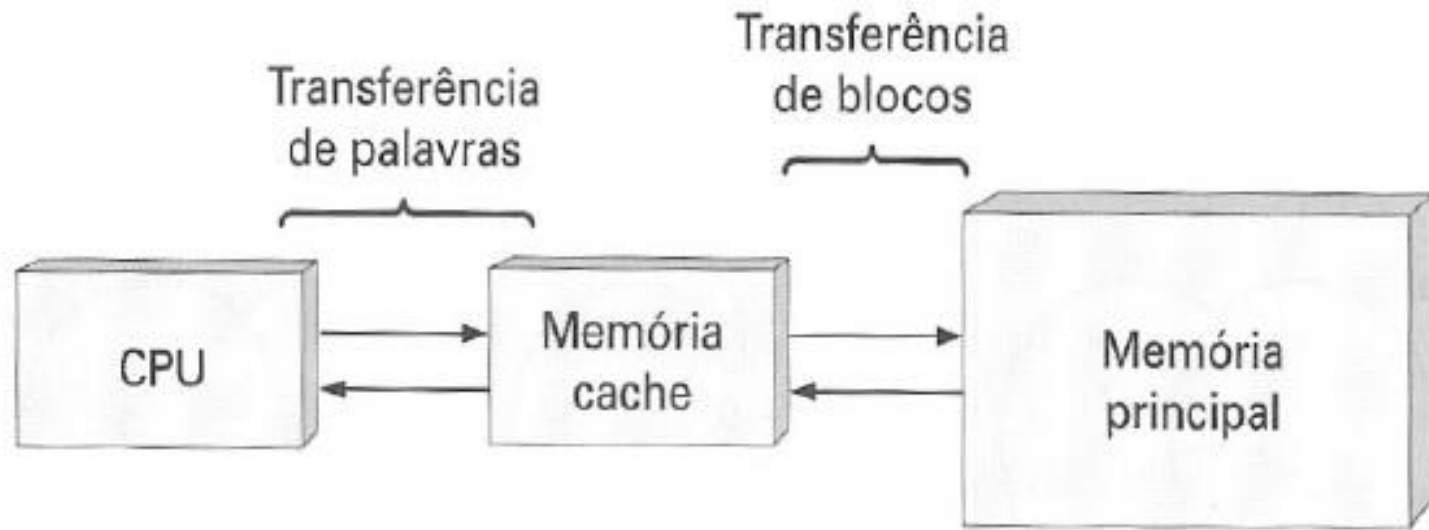
## Memória Cache

### □ Princípio básico

- Na execução de um programa de computador, muitas das referências são a um pequeno conjunto de posições de memória.



# Memória - Composição



Memória Cache e Memória Principal.

**Wait States:** é a manutenção do endereço no barramento por vários ciclos devido à diferença de velocidade entre processador e memória.

# Memória - Composição

## ❑ Memória Cache – Exemplo de funcionamento do sistema

- UCP vai buscar uma nova instrução (ou dado), após a busca inicial  $\Rightarrow$  acesso à memória cache.
- Instrução (ou dado) na cache  $\Rightarrow$  acerto (*hit*).
- Instrução (ou dado) não está na cache  $\Rightarrow$  falta/falha (*miss*)

**Falha no acesso à cache** – ocorre sempre que o processador procura uma informação na cache e essa informação não está armazenada neste local, havendo necessidade de buscá-la na memória principal.

# Memória - Composição

## Memória Cache – Definições

- ❑ **Taxa de acertos ou razão de acertos** (*hit ratio*) = fração dos acessos à memória encontrados no nível superior (com frequência, é usada como medida de desempenho do sistema de memória).
- ❑ **Taxa de faltas** (= 1- taxa de acertos) = fração de acessos à memória não encontrados no nível superior.

# Memória - Composição

## Memória Cache – Definições

- ❑ **Tempo de acerto** (*hit time*) = tempo necessário para acessar o nível superior da hierarquia, que inclui o tempo necessário para determinar se a tentativa de acesso à informação vai gerar um acerto ou uma falta.
- ❑ **Penalidade por falta** (*fault penalty*) é o tempo necessário para substituir um dos blocos do nível superior pelo bloco do nível inferior que contém a informação desejada, mais o tempo para enviar a informação ao processador.

# Memória - Composição

## Memória Cache

- ❑ Número de linhas de memória cache  $<$  número de blocos da memória principal. Solução:
  - Utilizar um algoritmo para mapear os blocos da memória principal em linhas da memória cache;
  - Utilizar um mecanismo para determinar o bloco da memória principal que ocupa uma dada linha da memória cache.
  
- ❑ **Técnicas utilizadas:**
  - Mapeamento direto
  - Mapeamento associativo
  - Mapeamento associativo por conjuntos

# Memória - Composição

## Memória Cache

- **Mapeamento direto**
  - Cada bloco da memória principal é mapeado em uma única linha de cache.
  - **Vantagem:** simplicidade e baixo custo
  - **Desvantagem:** cada bloco é mapeado em uma posição fixa da memória cache.
- A maioria das caches que usa mapeamento direto o faz usando o seguinte processo:

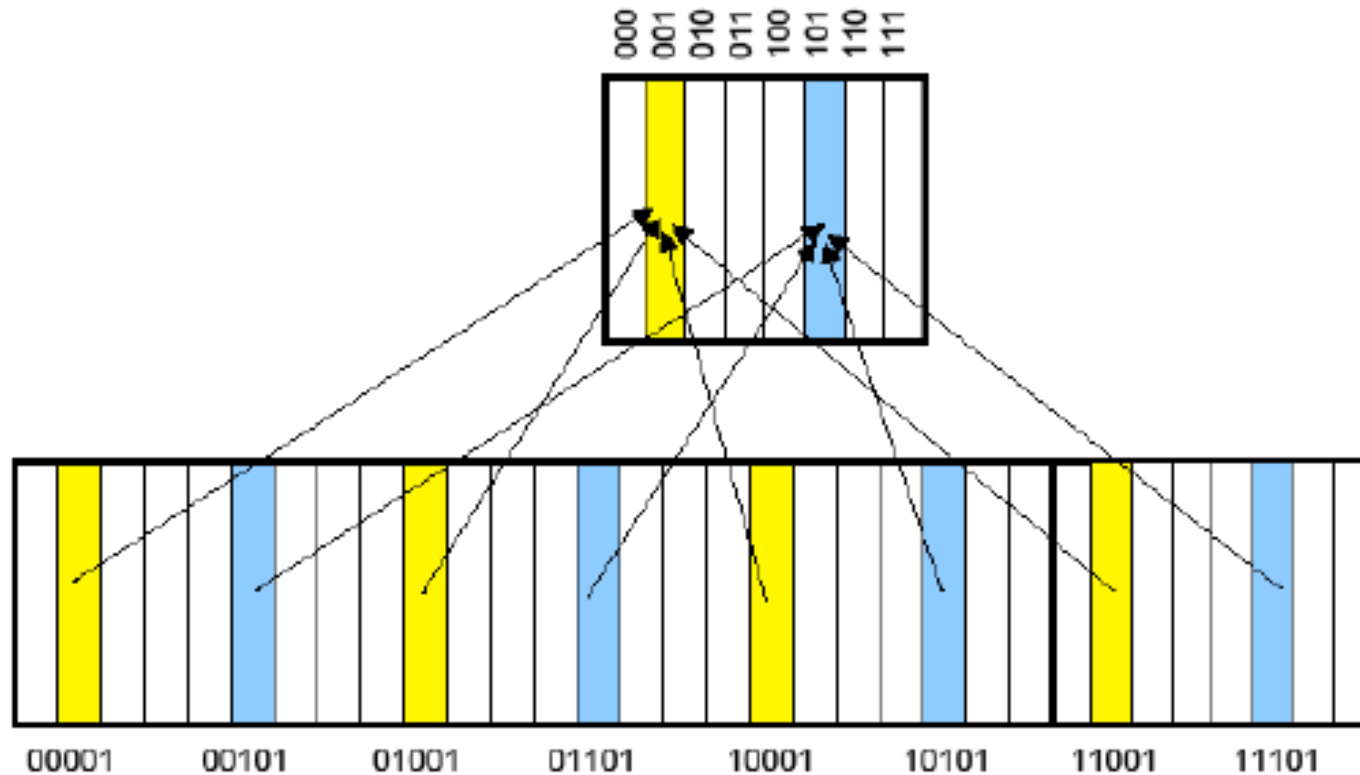
(Endereço do bloco) módulo (Número de blocos da cache)

↑  
Endereço absoluto  
(i.e., em relação à  
memória principal)

↑  
resto da divisão inteira

# Memória - Composição

## Mapeamento Direto para Memória Cache





# Memória - Composição

## Mapeamento Direto para Memória Cache

Dado que uma “entrada” da cache pode armazenar o conteúdo de diversos endereços da memória (principal), como identificar se o dado armazenado na cache corresponde ao solicitado?

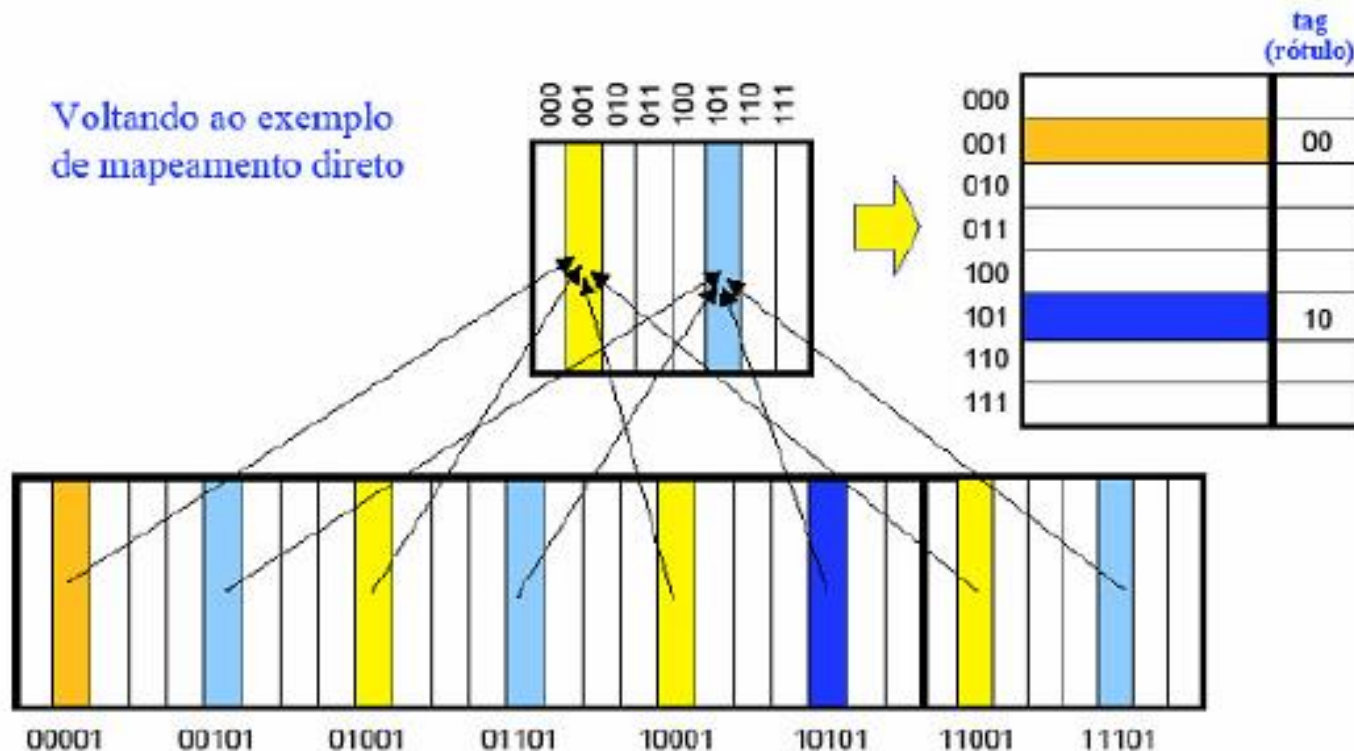
# Memória - Composição

## Mapeamento Direto para Memória Cache

- ❑ **Solução:** atribuir à cache um conjunto de rótulos (*tags*)
- ❑ Os rótulos são usados em conjunto com o endereço do mapeamento, de modo a compor o endereço completo, com relação à memória principal.

# Memória - Composição

## Mapeamento Direto para Memória Cache



# Memória - Composição

## Mapeamento Direto para Memória Cache

Como reconhecer se um bloco da cache possui uma informação válida? (Quando o processador é inicializado, por exemplo, algo deve sinalizar que a cache está “vazia”)

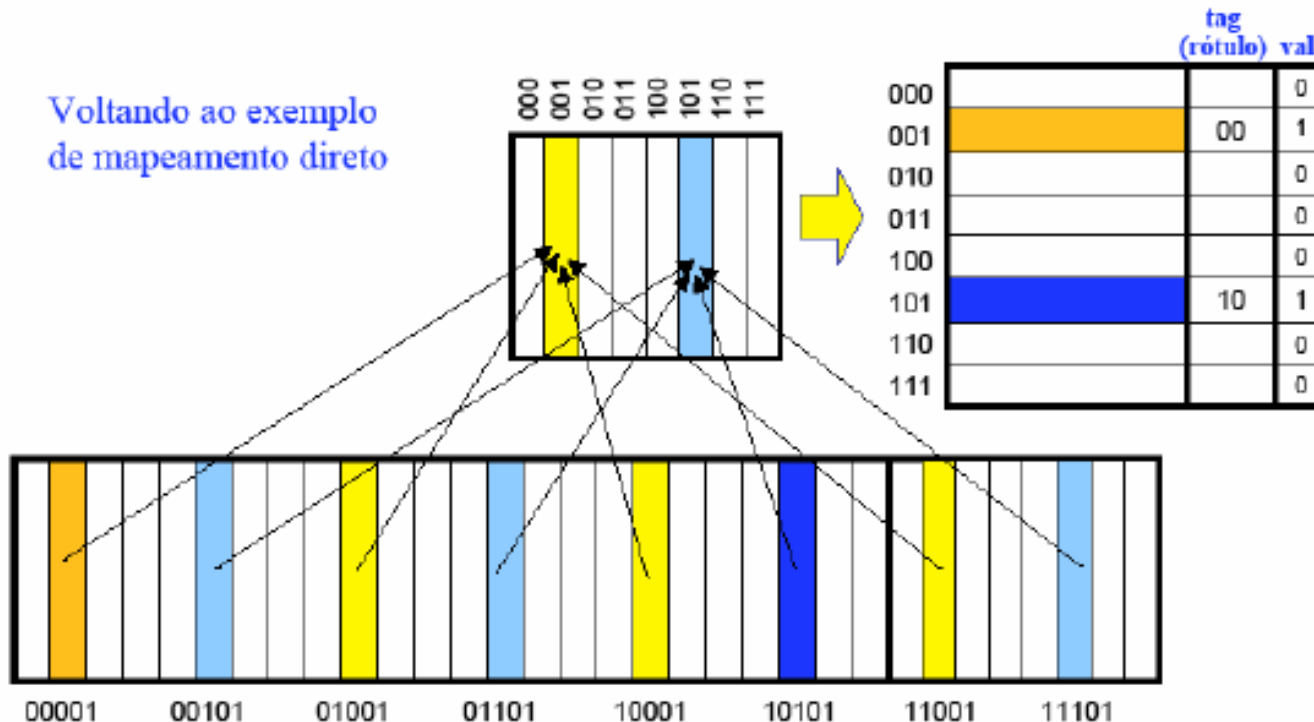
# Memória - Composição

## Mapeamento Direto para Memória Cache

- ❑ **Solução:** incluir um bit de validade.
- ❑ Se o bit de validade = 0, a informação contida naquele bloco da cache não é válida.

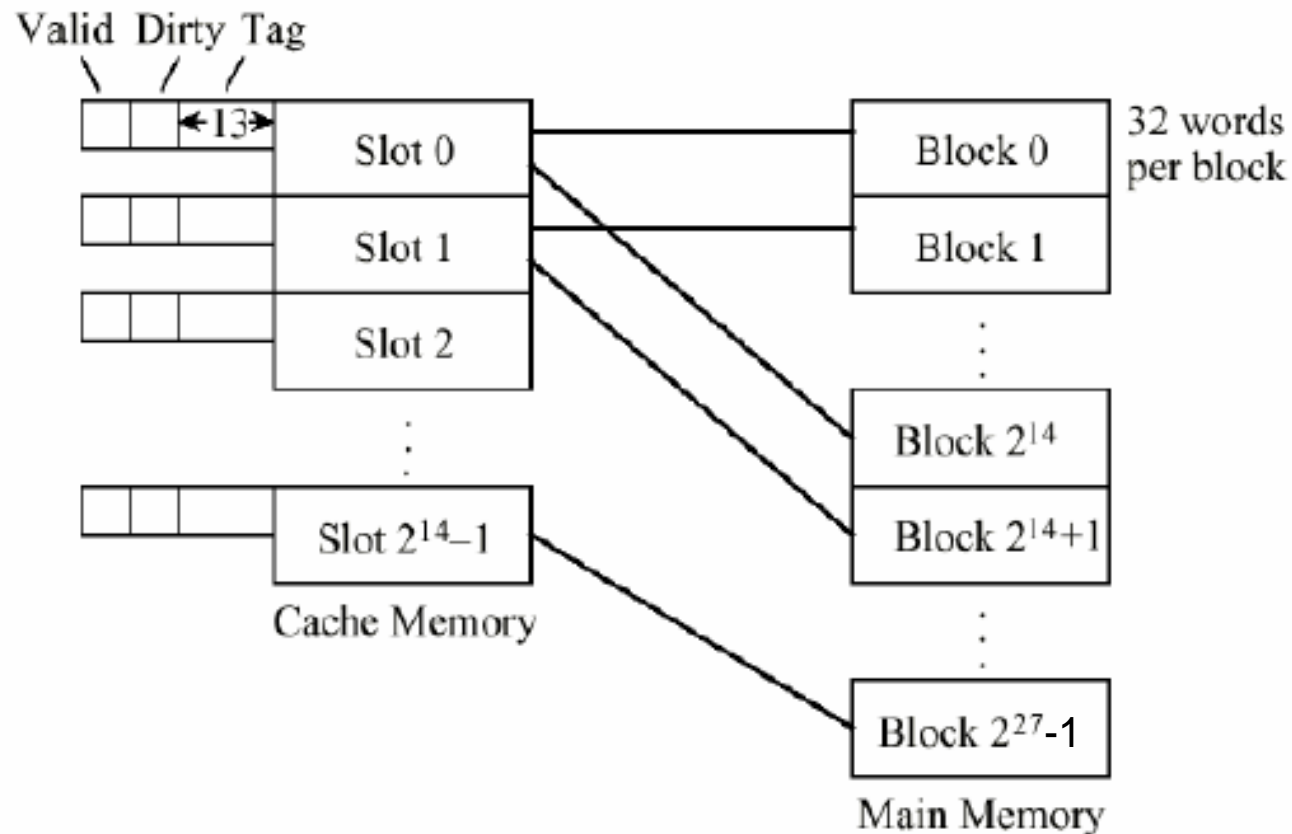
# Memória - Composição

## Mapeamento Direto para Memória Cache



# Memória - Composição

## Mapeamento Direto para Memória Cache



# Memória - Composição

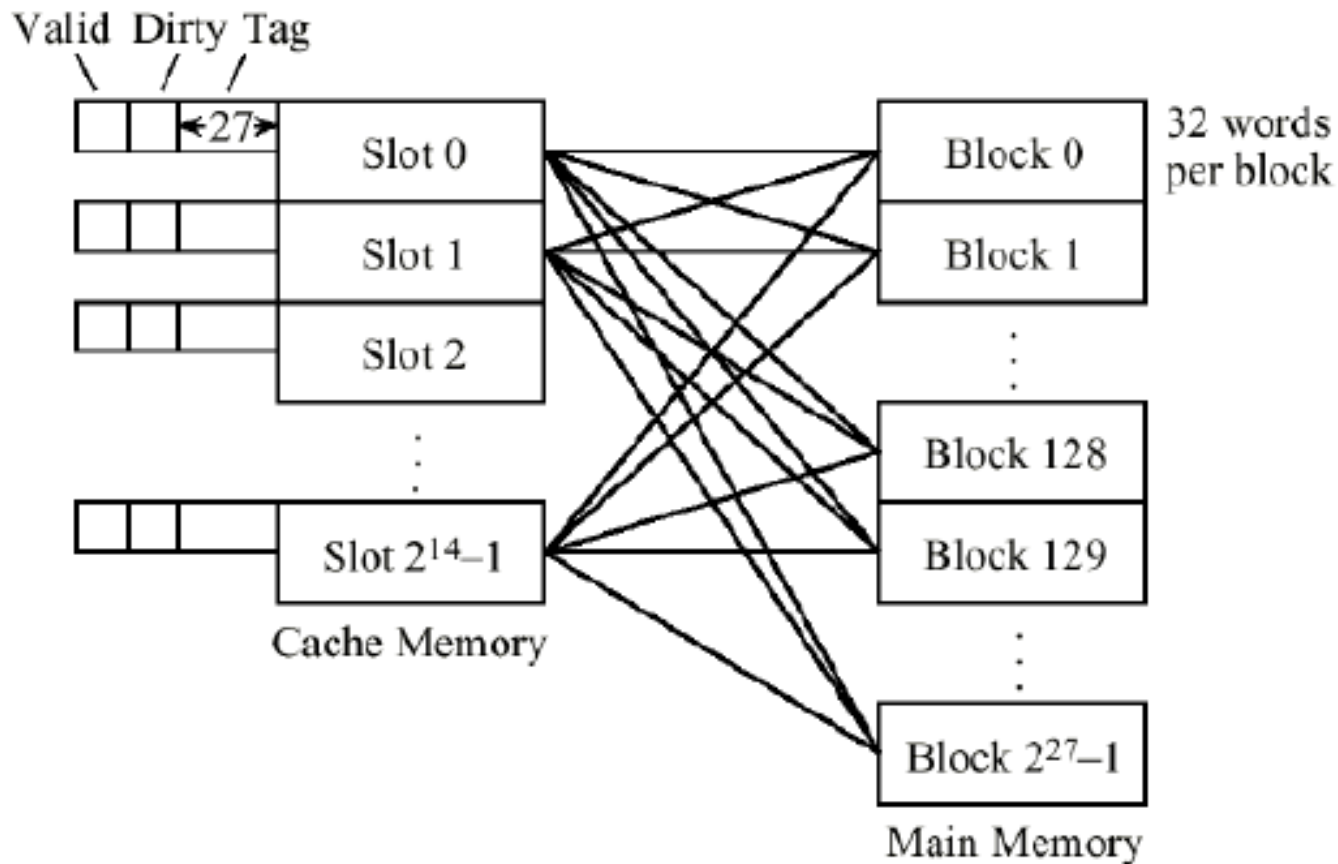
## Memória Cache

- **Mapeamento associativo**
  - Cada bloco da memória principal é carregado em qualquer linha da memória cache
  - **Vantagem:** oferece maior flexibilidade para escolha do bloco a ser substituído quando um novo bloco é trazido para a memória cache.
  - **Desvantagem:** complexidade do conjunto de circuitos necessários.



# Memória - Composição

## Mapeamento Associativo para Memória Cache



# Memória - Composição

## Memória Cache

- ❑ Quando um novo bloco é trazido para a memória cache, um dos blocos existentes deve ser substituído.
  - **Mapeamento direto** – cada bloco é mapeado em uma única linha, o que determina o bloco a ser substituído, não há alternativa.
  - **Mapeamento Associativo e Associativo por Conjuntos** – uso de algoritmo de substituição.

# Memória - Composição

## Memória Cache

### ❑ Algoritmos de substituição

- **LRU** (*Least Recently Used*) – substituir o bloco usado menos recentemente.
- **FIFO** (*First-in-first-out*) – substituir o bloco que está no conjunto há mais tempo.
- **LFU** (*Least Frequently Used*) – substituir o bloco que foi utilizado menos vezes.
- **Outra técnica** - substituir aleatoriamente uma das linhas candidatas.

Esquema *write-through*: a memória principal é atualizada a cada modificação da memória *cache*.

Esquema *write-back*: a memória principal só é atualizada quando um bloco da *cache* precisa ser substituído.

# Memória - Composição

## Memória Cache – Formas utilizadas

### ❑ Cache unificada

- Instruções e dados usando a mesma cache;
- Mais simples de projetar;
- Estabelece automaticamente um equilíbrio entre as buscas de instruções e as buscas de dados.

### ❑ Cache dividida (Arquitetura de Harvard)

- Uma cache para instruções e outra para dados;
- Permite acesso em paralelo a instruções e a dados.

Arquitetura de Harvard – alusão ao computador Mark III projetado por Howard Aiken, que tinha memórias diferentes para instruções e para dados.

# Memória - Composição

## Memória Cache – Formas utilizadas

### ❑ Cache unificada

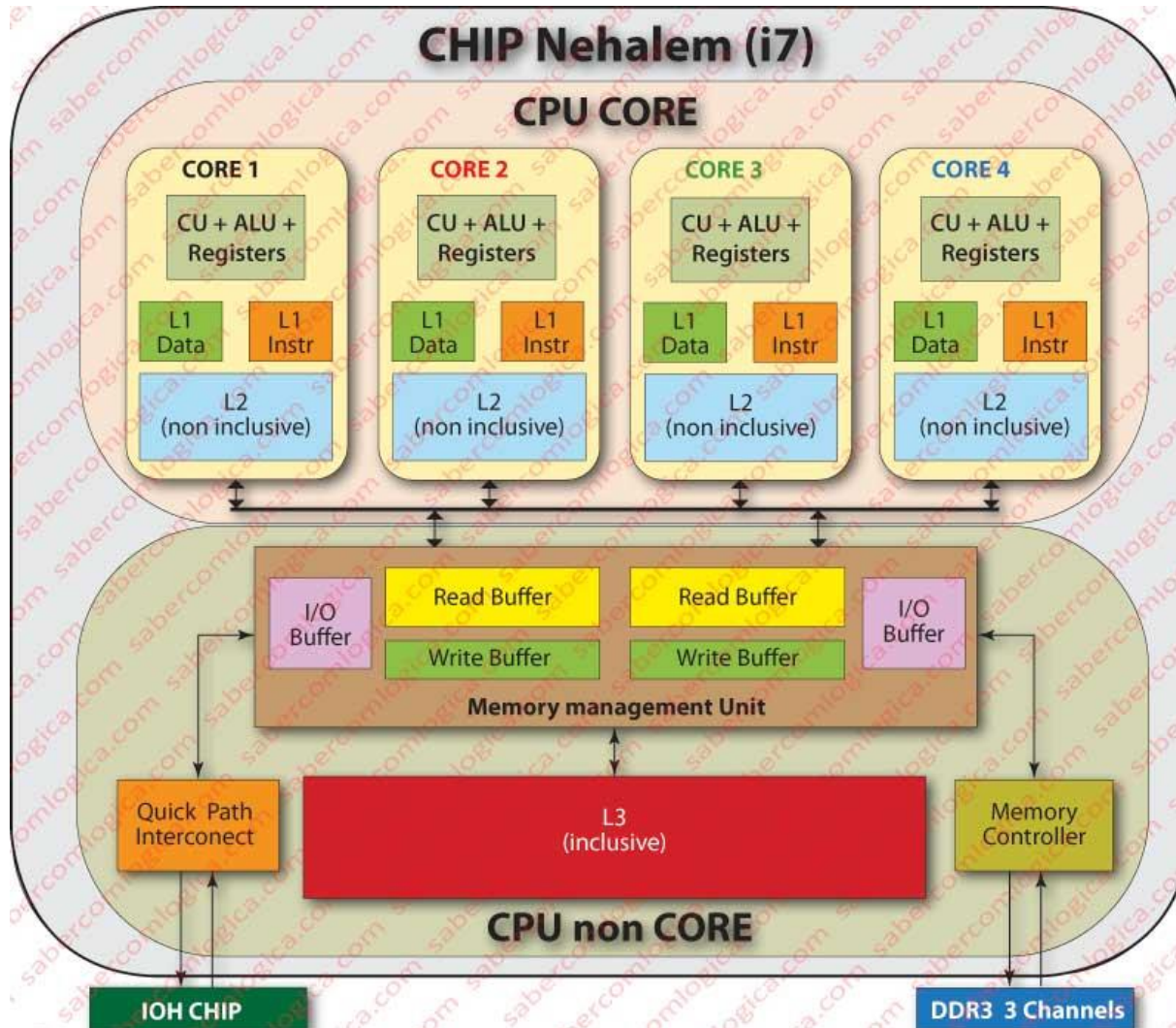
- Instruções e dados usando a mesma cache;
- Mais simples de projetar;
- Estabelece automaticamente um equilíbrio entre as buscas de instruções e as buscas de dados.

### ❑ Cache dividida (Arquitetura de Harvard)

- Uma cache para instruções e outra para dados;
- Permite acesso em paralelo a instruções e a dados.

Arquitetura de Harvard – alusão ao computador Mark III projetado por Howard Aiken, que tinha memórias diferentes para instruções e para dados.

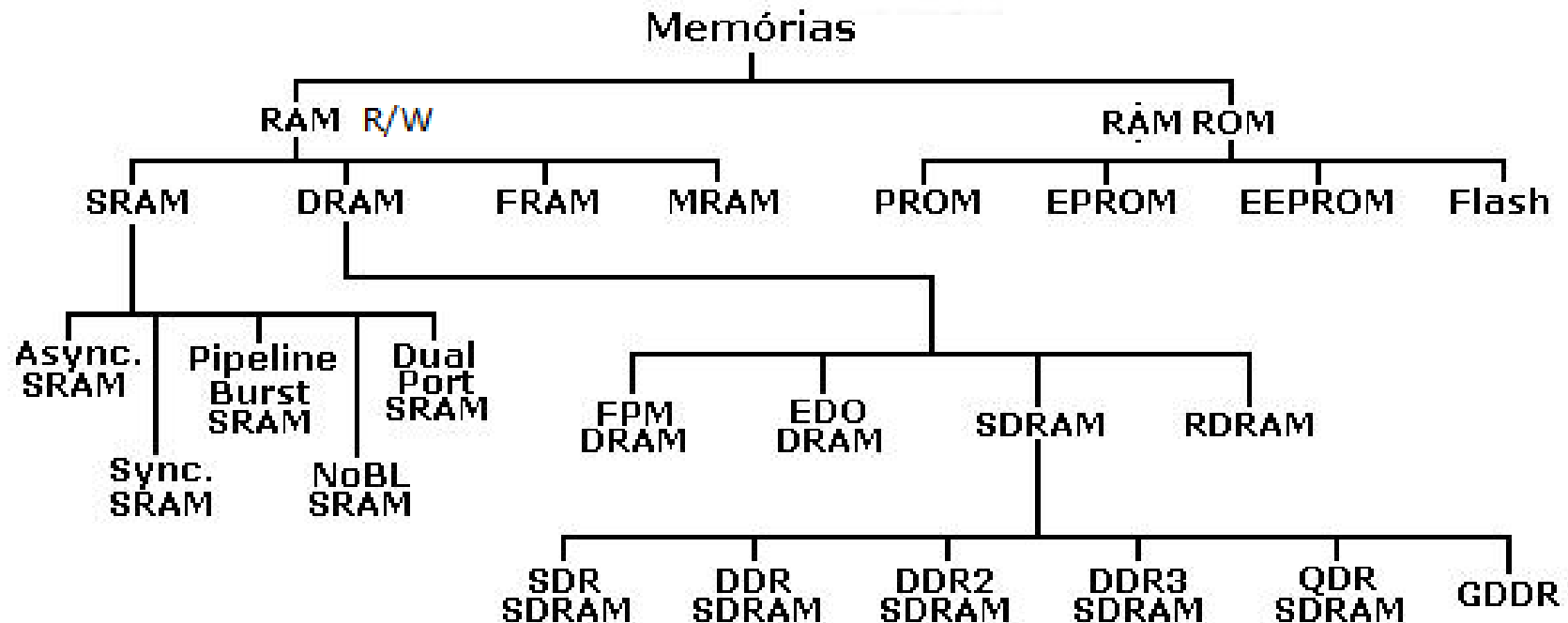
# Memória - Composição



# Memória - Composição

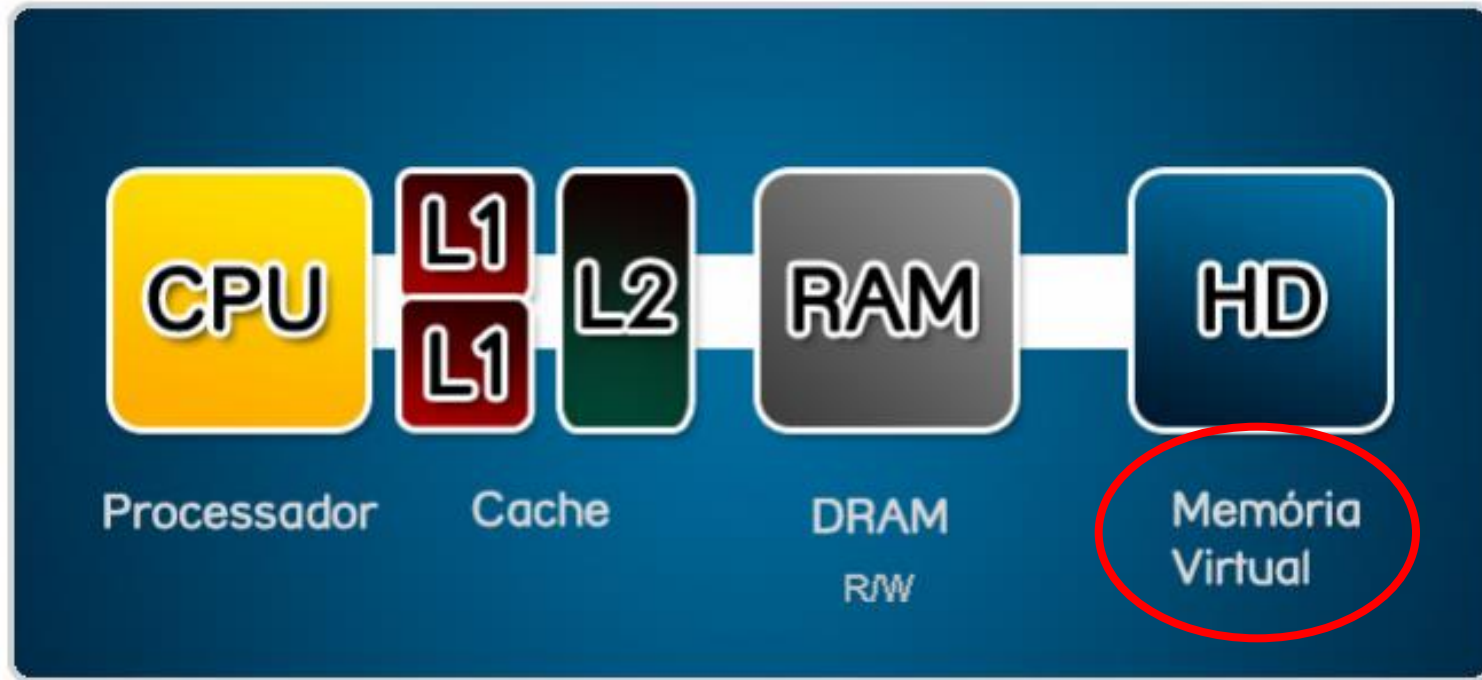
	Core i7-7820X	Core i7-8700K	Core i9-9900K	Core i7-9700K
Release Date	June 2017	October 2017	October 2018	
Cores / Threads	8/16	6/12	8/16	8/8
Base Frequency	3.6 GHz	3.5 GHz	3.6 GHz	3.6 GHz
Max Boost Frequency	4.3 GHz	4.7 GHz	5.0 GHz	4.9 GHz
L2 Cache	8 MB	1.5 MB	2 MB	2 MB
L3 Cache	11 MB	12 MB	16 MB	12 MB
Memory Config	Quad-Channel	Dual-Channel		
Max Mem Support	DDR4-2666			
TDP	140 W	95 W		
MSRP	\$600	\$360	\$500	\$374

# Memória - Composição

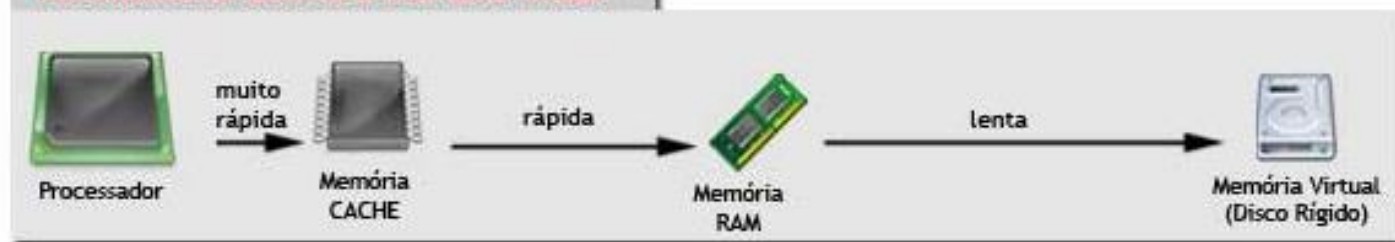




# Memória Virtual



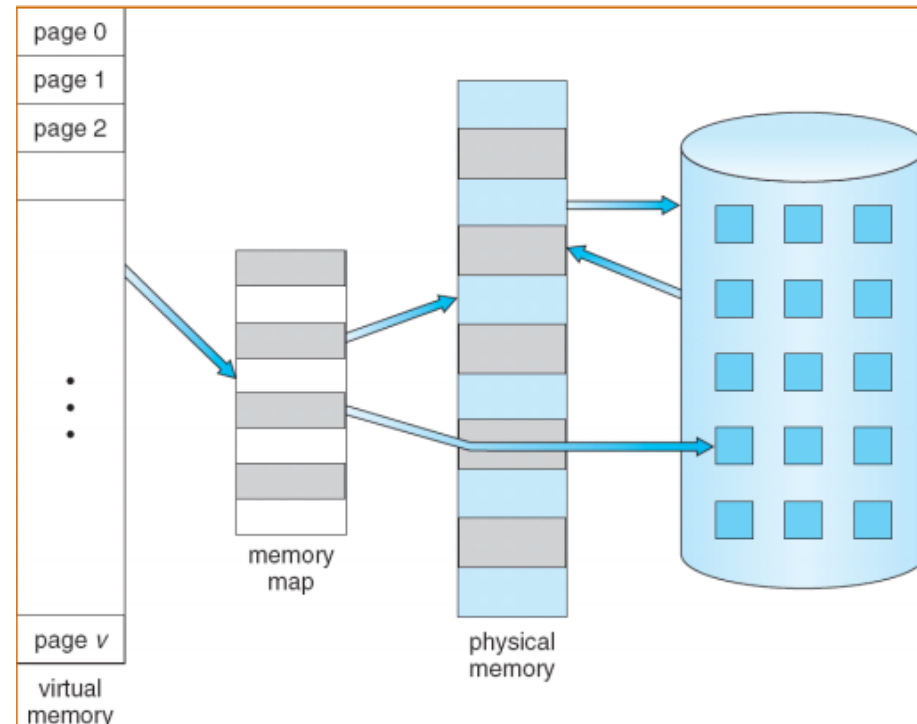
## velocidade de acesso a memória virtual



# Memória Virtual

## Conceitos Relevantes

- Paginação
- Segmentação



# Memória Virtual

## Política de Substituição de Páginas

- ❑ Quando ocorre um problema de falha de página, o Sistema Operacional deve substituir uma moldura, página da memória física, por uma página do disco.
  
- ❑ **Problema: Qual página deve ser devolvida ao disco?**
- ❑ O S. O. deve escolher, automaticamente, a página de mais baixa probabilidade de vir a pertencer ao conjunto de trabalho.
- ❑ Os algoritmos de substituição chamados **LRU** e **FIFO** podem ser usados nesses casos.
  - Algoritmo LRU (*Least Recently Used* – Não usado há mais tempo)
  - O algoritmo FIFO (*First In First Out* – Primeiro a Entrar, Primeiro a Sair)

# Memória Virtual

## Memória Virtual e Memória Cache

- ❑ Conceitualmente, essas duas memórias são idênticas; a única diferença é que elas atuam em níveis diferentes de hierarquia.
  
- ❑ As principais diferenças entre elas são:
  - As falhas no acesso à cache são tratadas por hardware, enquanto as falhas de páginas são tratadas pelo Sistema Operacional.
  - Os blocos da cache são muito menores do que as páginas.