

# AULA 1 – HIERARQUIA DE MEMÓRIA

## OBJETIVO DA AULA

Conhecer os diferentes níveis de dispositivos de armazenamento existentes nos computadores.

## APRESENTAÇÃO

O termo *memória* se refere a todo um sistema utilizado para armazenar dados e informações no computador, embora normalmente empregamos esse termo para nos referirmos à memória principal.

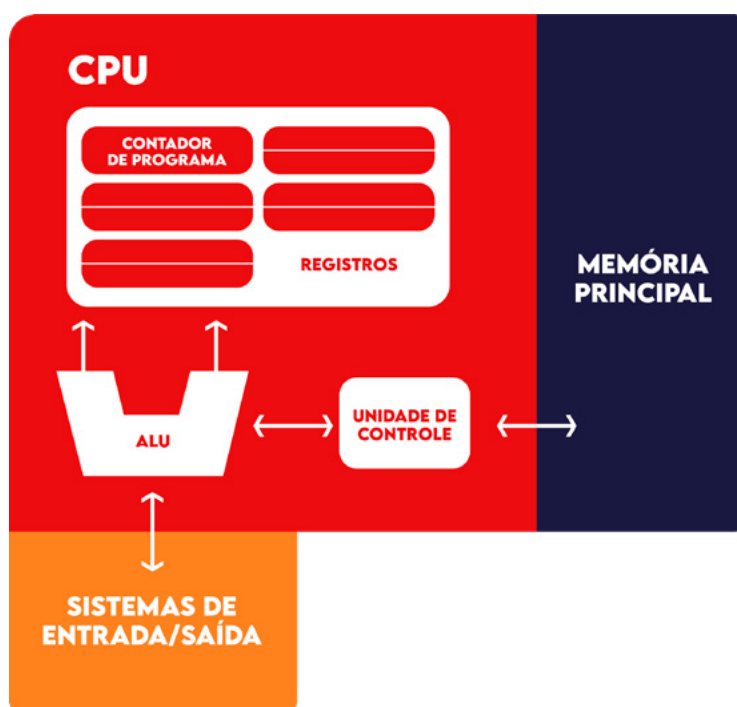
Vamos estudar aqui nesta aula como a memória do computador é dividida em níveis com características distintas e porque isso deve ser assim.

## 1. HIERARQUIA DE MEMÓRIA

Para iniciarmos nossa aula é importante sabermos que nossos computadores são organizados segundo o que conhecemos como *Arquitetura de Von Neumann*.

Basicamente essa arquitetura se caracteriza por um processador (unidade de controle e unidade lógico aritmética) executando um programa armazenado em uma memória.

FIGURA 1 | A Arquitetura de Von Neumann



Livro Eletrônico  
Fonte: adaptado de Wikipedia.org.

Agora vamos ver como a memória é organizada para atender da melhor forma possível às demandas do processador.

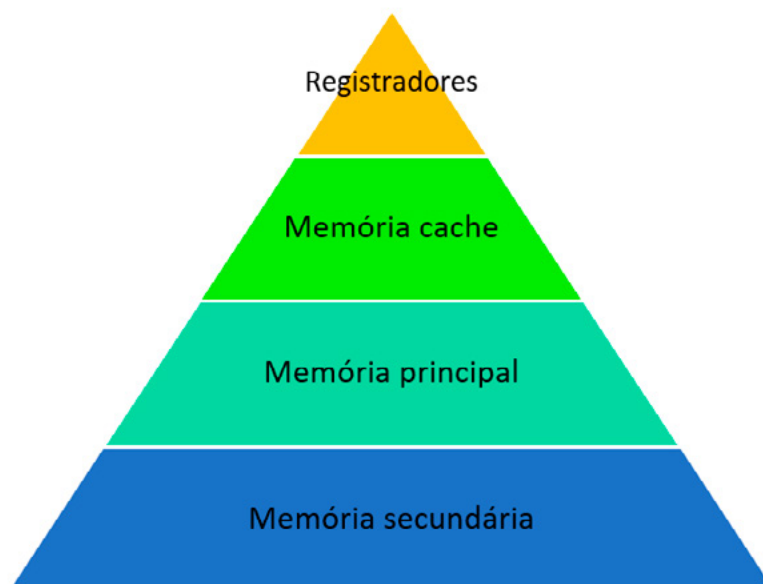
Antes de passarmos à organização do sistema de armazenamento, vale a pena conhecermos alguns conceitos importantes sobre memórias.

Quanto à forma de acessar as informações na memória, temos duas formas principais.

- **Acesso sequencial:** nesta forma de acesso, para conseguirmos gravar ou ler algo, precisamos percorrer a memória, endereço por endereço, até chegarmos à posição desejada. Um exemplo disso é visto nas antigas fitas cassete ou VHS, onde tínhamos que “correr” a fita até chegarmos na posição desejada;
- **Acesso randômico:** nesta forma de acesso, o endereço desejado é alcançado diretamente, sem necessidade de passar pelos endereços anteriores. Todo o sistema de armazenamento dos computadores atuais têm acesso randômico, desde os níveis mais baixos até os mais altos.

O sistema de armazenamento é organizado de forma hierárquica, basicamente em quatro níveis, conforme podemos ver na Figura 2.

FIGURA 2 | **Hierarquia de memória**



Fonte: <https://blog.maxieduca.com.br/memoria-cache-computador/a-hierarquia-de-memoria>.

Como vemos na Figura 2, a hierarquia de memória é representada na forma de pirâmide, o que reflete os aspectos que debateremos agora.

Em qualquer nível de memória há duas operações possíveis: leitura e escrita. A leitura ocorre quando se busca algo armazenado na memória e a escrita ocorre quando gravamos (salvamos) algo na memória.

Em primeiro lugar precisamos compreender quais são os atributos de cada nível de memória. São eles:

- Custo: é o preço por bit armazenado;
- Velocidade: diz respeito ao tempo de acesso. Esse tempo é aquele transcorrido entre uma solicitação de leitura ou escrita. Quanto menor for esse tempo, mais veloz será a memória;
- Capacidade de armazenamento: é a quantidade de bytes que a memória consegue armazenar;
- Volatilidade: diz respeito à capacidade da memória de reter informações em relação à alimentação elétrica. As memórias voláteis são aquelas que só retêm informações quando há energia elétrica e as memórias não voláteis são aquelas que conseguem reter informações independentemente da presença da energia elétrica.

Analisando os quatro níveis de memória de baixo para cima na pirâmide temos:

- Memória secundária – é composta pelo disco rígido e demais mídias portáteis, tais como pendrives, CDs, DVDs, fitas magnéticas, cartões magnéticos etc. É também conhecida como *memória de massa* por sua capacidade de armazenar grandes quantidades de informação. São memórias não voláteis, por não dependerem de energia elétrica para reter as informações;
- Memória principal – é o segundo nível de baixo para cima. Essa memória fica fisicamente instalada na placa mãe e é aquela a que nos referimos quando avaliamos a configuração de um computador que queremos comprar. Em geral, os vendedores informam qual o processador, qual o tamanho do Hard Disk e qual a quantidade de memória. Tal memória é a principal. Todos os programas executados pelo computador devem ser carregados para ela;
- Memória cache – é um tipo de memória intermediária entre a memória principal e o processador e surgiu para melhorar o desempenho dos computadores de forma que vamos estudar detalhadamente na próxima aula;
- Registradores – nível mais alto da pirâmide, esse é um tipo de memória que fica dentro do próprio processador. O banco de registradores armazena as informações que o processador está usando de forma mais imediata.

Analisaremos agora os atributos de memória a que nos referimos do ponto de vista da pirâmide.

Em primeiro lugar, quanto ao custo, enquanto subimos na pirâmide esse custo aumenta. Assim, armazenar um bit em um registrador é mais caro do que em um pendrive (memória secundária), por exemplo.

Em seguida temos a questão da velocidade. Também enquanto subimos na pirâmide, os dispositivos de memória vão ficando mais rápidos, ou seja, com um tempo de acesso menor. Assim, acessar uma informação na cache, por exemplo, gasta menos tempo do que acessar uma informação na memória principal.

Quanto à capacidade de armazenamento, enquanto subimos na pirâmide, essa capacidade diminui. Isso nos leva a concluir que o banco de registradores, por exemplo, armazena muito menos informações do que a memória principal.

Finalmente, quanto à volatilidade, somente os dispositivos que compõem a memória secundária são não voláteis. Em todos os outros níveis os dispositivos são voláteis, o que significa que quando desligamos o computador, tudo o que está armazenado ali é perdido. Daí a necessidade de salvarmos periodicamente tudo aquilo que estamos criando no computador. Salvar, portanto, significa copiar as informações de um dispositivo volátil para um dispositivo não volátil.

Aqui vale uma observação importante. Algumas pessoas têm dúvida e se confundem quanto à volatilidade do disco rígido, já que este deve ser ligado à fonte da placa mãe. Na verdade, essa ligação é necessária apenas porque o mecanismo do disco rígido necessita da energia elétrica para as operações de acesso, já que estas precisam mover o braço de leitura/escrita e para o movimento de rotação do próprio disco. Mas isso não tem nada a ver com a retenção dos dados e informações. Até porque tudo o que instalamos no computador, desde o sistema operacional até os aplicativos e jogos que usamos, bem como nossos arquivos, fica armazenado nesse disco mesmo quando desligamos o computador.

Agora vamos ver o que leva os projetistas a organizar a memória do computador desta forma hierárquica.

Em primeiro lugar, você poderia estar se perguntando:

Por que não organizar todo o sistema de memória com a mesma tecnologia utilizada nos registradores, já que eles são muito mais rápidos do que os outros dispositivos?

Porque não organizar todo o sistema de memória com a mesma tecnologia utilizada nas memórias secundárias, já que elas são muito mais baratas do que os outros dispositivos?

Respondendo à primeira questão, diríamos que, se toda a memória do computador utilizasse a tecnologia do banco de registradores, teríamos os seguintes problemas:

- Computadores absurdamente caros, inviáveis do ponto de vista financeiro;
- Computadores gigantescos, já que o sistema de memórias não poderia ser tão compacto em razão de aspectos tecnológicos;
- Consumo de energia inviável, já que os registradores consomem alta quantidade de energia proporcionalmente a seu tamanho e, conseqüentemente, também geram muito calor, criando a necessidade de sistemas de dissipação de calor muito potentes, o que consumiria ainda mais energia.

No caso da segunda questão, a resposta é muito simples: com a tecnologia da memória secundária sendo utilizada em todo o sistema de armazenamento, os computadores se tornariam terrivelmente lentos, não conseguindo atender à maior parte das demandas a que atendem atualmente.

Sendo assim, a organização da memória precisa prover uma relação custo/benefício que torne o computador atrativo para as pessoas.

Ser atrativo significa ser tão eficiente que as pessoas *queiram* comprar e com um preço tão acessível que as pessoas *possam* comprar.

Veremos agora como isso funciona.

Em algumas situações precisamos de muito espaço para armazenamento dos nossos dados e programas. Por exemplo, o primeiro programa a ser instalado no computador é o sistema operacional. Então ele deve ficar armazenado em um dispositivo com bastante espaço.

Além disso, nossos aplicativos, editores de texto, planilhas, jogos etc. precisam também de espaço de armazenamento e devem estar disponíveis sempre que usarmos o computador.

Sendo assim, eles ficam na memória secundária, no hard disk (ou no SSD, conforme veremos mais adiante).

Para serem executados, os programas devem ser carregados para a memória principal, que deve ter uma velocidade maior do que a memória secundária. Ela não precisa de tanto espaço já que só serão carregados nela os programas que formos executar. Então, o espaço e a velocidade que a memória principal oferece são suficientes para isso.

É importante lembrar que, por ser volátil, a memória principal só retém as informações enquanto o computador estiver ligado. Assim, tudo o que quisermos guardar em definitivo deverá ser copiado para alguma memória não volátil.

Mas a memória principal tem um tempo de acesso muito grande, sendo muito lenta do ponto de vista do processador. Por causa disso, o processador, sempre que precisa acessar a memória (e isso é muito frequente), passa muito tempo aguardando a conclusão das operações de leitura/escrita. Assim, para minimizar esse problema, foi necessário introduzir um meio de armazenamento mais rápido (embora com menos espaço disponível) entre a memória principal e o processador. Esse meio é a memória *cache*.

O termo *cache*, literalmente, significa *esconderijo*, mas, no sistema de armazenamento do computador, a *cache* é o local onde ficam temporariamente armazenadas as informações com maiores chances de serem usadas.

Você pode estar se perguntando como saber quais são as informações com mais chances de serem usadas. Isso é possível devido ao comportamento muitas vezes previsível dos programas. Mas estudaremos isso detalhadamente quando formos abordar o relacionamento entre as memórias *cache* e *principal*.

Finalmente, o processador trabalha executando instruções que manipulam dados e produzem informações. Tudo o que ele precisa a cada momento fica no banco de registradores, que é um dispositivo de armazenamento extremamente veloz e que se situa no núcleo (core) do processador.

Assim fica claro que cada nível de memória tem suas razões de existência e a presença de todos eles distribuídos no sistema de armazenamento torna o computador uma ferramenta eficiente, barata e acessível.

É graças a essa organização da memória em níveis hierárquicos que os computadores são ferramentas tão populares, com desempenho excelente e preço acessível.

## CONSIDERAÇÕES FINAIS

Acabamos de ver como a memória do computador é organizada e as razões dessa escolha.

Vimos que o computador precisa ter um desempenho e um preço que o tornem atraente e o sistema de memórias, com o processador, tem uma participação decisiva nesses aspectos.

A memória sendo organizada de forma hierárquica consegue atender a todos os tipos de demanda, permitindo que tanto aquilo que deve ser guardado definitivamente quanto o que é necessário de forma temporária têm seus dispositivos de armazenamento adequados.

Na próxima aula abordaremos de forma mais aprofundada o relacionamento entre a memória principal e a cache, vendo como essa última foi fundamental para a melhoria do desempenho dos computadores.

Até lá!

## MATERIAIS COMPLEMENTARES

<https://www.youtube.com/watch?v=3L5wP4VFxGQ> – Esse vídeo é um resumo dos assuntos abordados nessa aula.

<https://www.youtube.com/watch?v=K-trdnFIHhg> – Nesse vídeo você poderá saber um pouco mais sobre as memórias somente de leitura, que são as memórias ROM.

## REFERÊNCIAS

STALLINGS, William. *Arquitetura e Organização de Computadores: projeto para o desempenho*. 8ª edição. Editora Pearson. Livro. (642 p.). ISBN 9788576055648. Disponível em: <<https://middleware-bv.am4.com.br/SS0/iesb/9788576055648>>. Acesso em: 16 out. 2022.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. 3ª edição. Editora Pearson. Livro. (674 p.). ISBN 9788576052371. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576052371>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Organização estruturada de computadores*. 6ª edição. Editora Pearson. Livro. (628 p.). ISBN 9788581435398. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788581435398>>. Acesso em: 16 out. 2022.

## AULA 2 – MEMÓRIA CACHE E MEMÓRIA PRINCIPAL

### OBJETIVO DA AULA

Abordar o relacionamento entre memória principal e memória secundária e sua influência no desempenho do computador.

### APRESENTAÇÃO

Olá!

Nesta aula vamos conhecer melhor o relacionamento entre as memórias cache e principal.

Veremos aqui de que forma a cache, que é mais rápida, contribui para a melhoria do desempenho do computador.

Mas antes disso vamos entender um pouco do funcionamento de cada uma delas, bem como suas vantagens e limitações.

Mãos à obra!

### 1. MEMÓRIA CACHE E MEMÓRIA PRINCIPAL

Na última aula, estudamos a hierarquia de memórias e vimos a importância de dividir o sistema de armazenamento do computador em diferentes níveis hierárquicos para garantir uma relação custo/benefício viável para os computadores.

Agora vamos ver como funcionam a memória cache e a principal, já que são nelas que ficam os programas que executamos em nosso computador.

Antes de mais nada, comentaremos aqui um problema chamado *Gargalo de Von Neumann*, que está presente em todos os nossos computadores, já que são Arquiteturas de Von Neumann.

De forma simples, o Gargalo de Von Neumann é o problema que acontece pelo fato de que processador ter uma velocidade muito maior do que as memórias. Sendo assim, sempre que precisa acessá-las, ele passa muito tempo aguardando a conclusão de seus pedidos de leitura e escrita.

Portanto, esse problema afeta diretamente a performance do computador, uma vez que o processador tem seu trabalho prejudicado.

Há diversas formas de tentar mitigar esse problema e uma delas é a introdução das memórias cache.

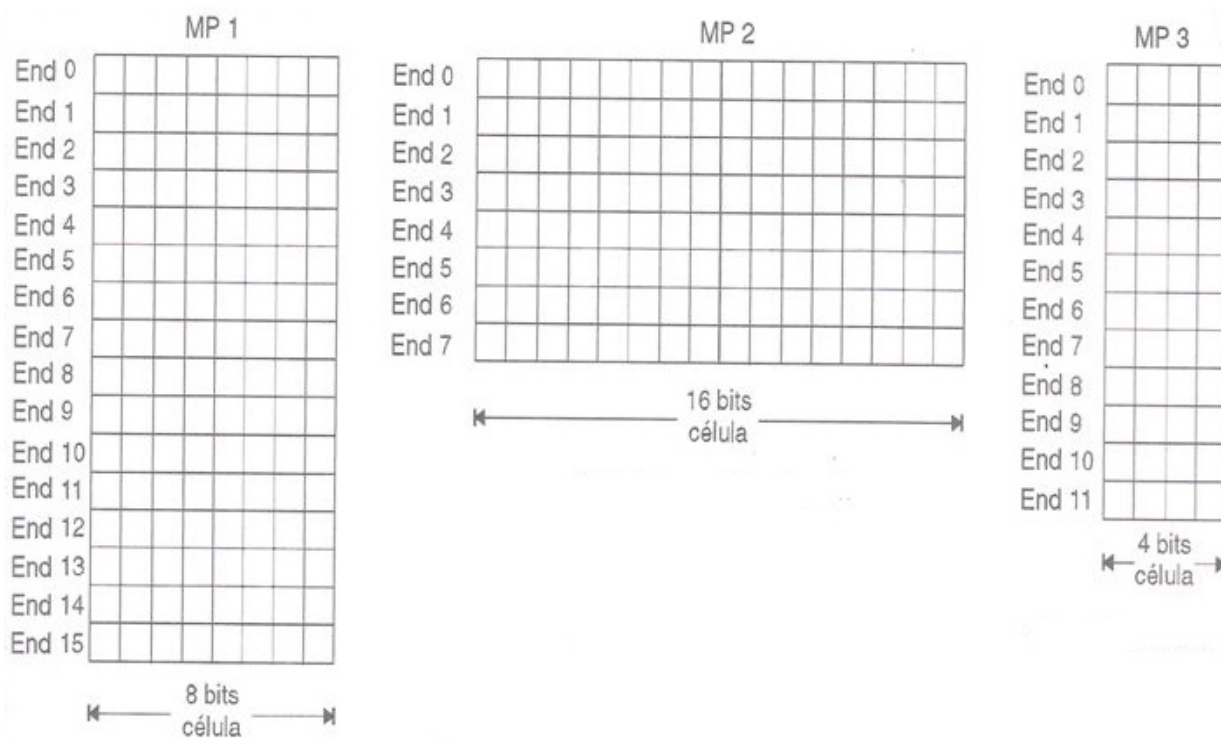
Livro Eletrônico



Como vimos na pirâmide da aula passada, a memória cache é um intermediário entre o processador e a memória principal, sendo mais veloz, porém com menor capacidade de armazenamento.

A memória principal é estruturalmente organizada em células, cada uma identificada por um endereço que identifica a localização física da célula na memória.

**FIGURA 1 | Exemplos de organização de memória**



Fonte: [http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula\\_4.html](http://www.dpi.inpe.br/~carlos/Academicos/Cursos/ArqComp/aula_4.html).

Na Figura 1 temos três tipos de memória. Na primeira as células possuem oito bits cada temos dezesseis endereços (0 a 15) para localizar cada célula. Na segunda são células de dezesseis bits e temos oito endereços (0 a 7). Finalmente, na terceira, temos doze células de quatro bits.

O conteúdo da memória é acessado através dos endereços das células.

A memória principal é formada por capacitores e estes têm como uma de suas propriedades físicas o fato de “perderem” os dados depois de um tempo. Isso torna necessária uma operação chamada *refresh*.

O *refresh* é uma varredura, bit a bit, na memória principal para “lembrar” a cada um deles o seu valor (0 ou 1). Essa operação ocorre várias vezes por segundo e é em razão dela que a memória principal é chamada de DRAM (Dynamic RAM).

Por ocorrer várias vezes por segundo, o *refresh* contribui para o aumento do tempo de acesso da memória principal, tornando-a mais lenta.

Já a memória cache é formada por circuitos que dispensam o *refresh*, porém geram mais calor e por isso não permite uma alta densidade de integração, consumindo também mais energia. Ela é chamada de SRAM (*Static RAM*).

Observe o quadro abaixo.

Quadro 1 | **Comparação entre memórias DRAM e SRAM**

	DRAM (principal)	SRAM (cache)
<b>Vantagens</b>	Alta densidade de integração; Menor consumo de energia; Menos geração de calor; Custo menor.	Alta velocidade; Não precisa de <i>refresh</i> .
<b>Desvantagens</b>	Baixa velocidade; Necessidade de <i>refresh</i> .	Baixa densidade de integração; Maior consumo de energia; Mais geração de calor; Custo maior.

Fonte: elaboração própria.

Pela Tabela 1 podemos entender porque a cache é mais rápida (não precisa de *refresh*) e porque tem menor capacidade de armazenamento (baixa densidade de integração, ou seja, menos bits por espaço físico).

Agora vamos ver de que forma a cache contribui para a melhoria do desempenho do processador.

Como sabemos, a cache é mais rápida e menor do que a memória principal. A utilização da cache, portanto, se dará da seguinte forma: nela ficará somente o que for mais importante para o processador a cada momento. Isto significa que o processador terá quase sempre aquilo de que precisa no meio de armazenamento mais rápido.

Mas como saber o que é mais provável de ser necessário ao processador?

Isso é possível devido a uma característica de comportamento previsível dos programas, o que permite deduzirmos o *princípio de localidade*, que é subdividido em:

**Princípio da localidade espacial**, segundo o qual, sempre que o processador acessar um endereço, ele provavelmente acessará também os endereços vizinhos.

**Princípio da localidade temporal**, segundo o qual, sempre que o processador acessar um endereço, ele provavelmente voltará a acessar esse endereço novamente em um curto espaço de tempo.

Graças ao princípio da localidade, é possível saber na maioria das vezes o que é mais provável de ser necessário ao processador a cada instante. Isto porque os programas apresentam um comportamento parecido na maioria das vezes.

Então, como a cache armazena o que é mais provável de ser necessário, o processador, toda vez que precisa buscar algo na memória, tenta primeiro encontrar aquilo que precisa na cache. Caso não consiga, ele vai procurar na memória principal.

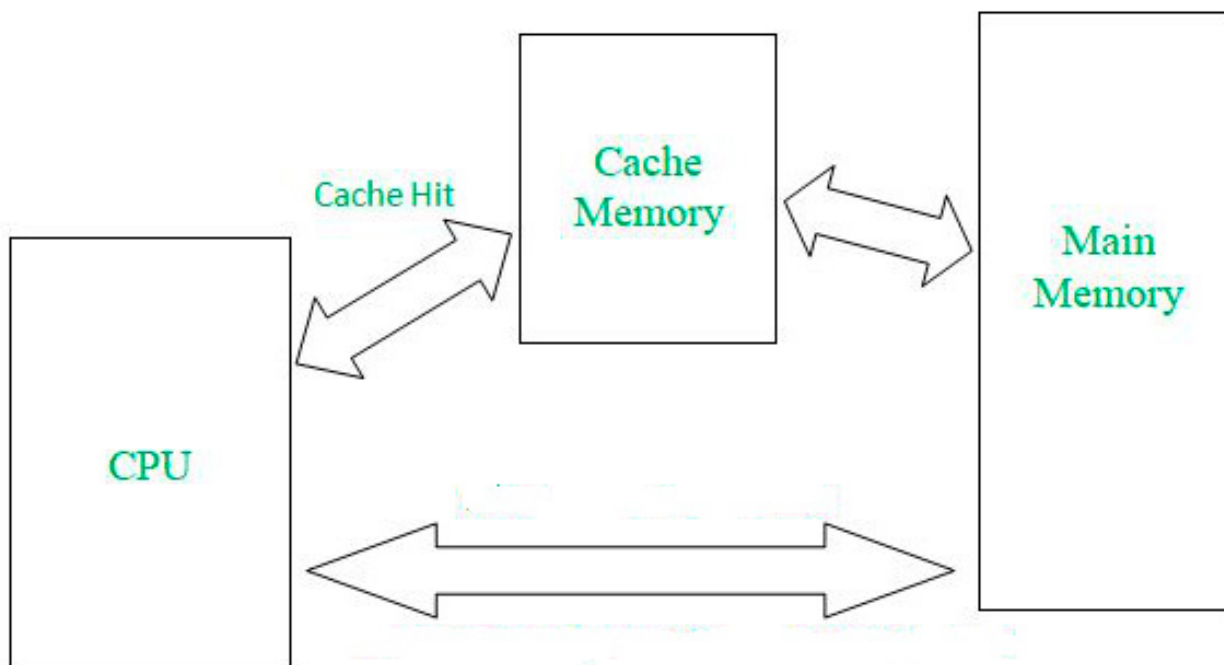
Quando o processador consegue o que precisa na cache, chamamos esse evento de *cache hit* (hit de cache – sucesso na cache) e quando não consegue, chamamos esse evento de *cache miss* (miss de cache – falta na cache).

Porém, na ocorrência de um *cache miss*, o processador tem um trabalho, que envolve os seguintes passos:

- 1) Ir à memória principal (mais lenta) achar o que não achou na cache;
- 2) Trazer o que pegou na memória principal para a memória cache, já que, pelo princípio de localidade, provavelmente precisará acessar essas informações novamente num breve espaço de tempo;
- 3) Caso a cache esteja cheia, escolher qual dos blocos presentes nela dará lugar àquele que acabou de chegar;
- 4) Efetuar essa substituição.

A essa sequência de passos damos o nome de *miss penalty*.

FIGURA 2 | **Relacionamento entre processador e sistema de memórias**



Fonte: <https://acervolima.com/acessos-simultaneos-e-hierarquicos-ao-cache/>.

Importante ressaltar que as informações são transferidas entre as memórias principal e cache em blocos por conta da localidade espacial.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

Para o correto funcionamento do esquema cache/memória principal, é importante considerarmos as seguintes questões:

1) Como escolher o bloco a ser substituído quando é necessário trazer um novo bloco da memória principal para a cache?

2) Como fazer a correlação entre os endereços de memória principal e cache, já que se trata de duas memórias fisicamente distintas?

À primeira questão respondemos: através dos algoritmos ou políticas de substituição de blocos.

À segunda questão respondemos: através do mapeamento de cache.

As políticas de substituição de blocos são as seguintes:

- FIFO (*First In First Out*) – este algoritmo funciona basicamente como uma fila, ou seja, o bloco que está há mais tempo na cache (o primeiro que chegou) é escolhido para ser substituído;
- LFU (*Least Frequently Used* – menos frequentemente usado) – de acordo com esse algoritmo, o bloco a ser substituído é aquele que tiver a menor quantidade de acessos;
- LRU (*Least Recently Used* – menos recentemente usado) – nesse caso, o bloco a ser utilizado é aquele que tiver sido usado pela última vez há mais tempo, ou seja, aquele que está há mais tempo ocioso.

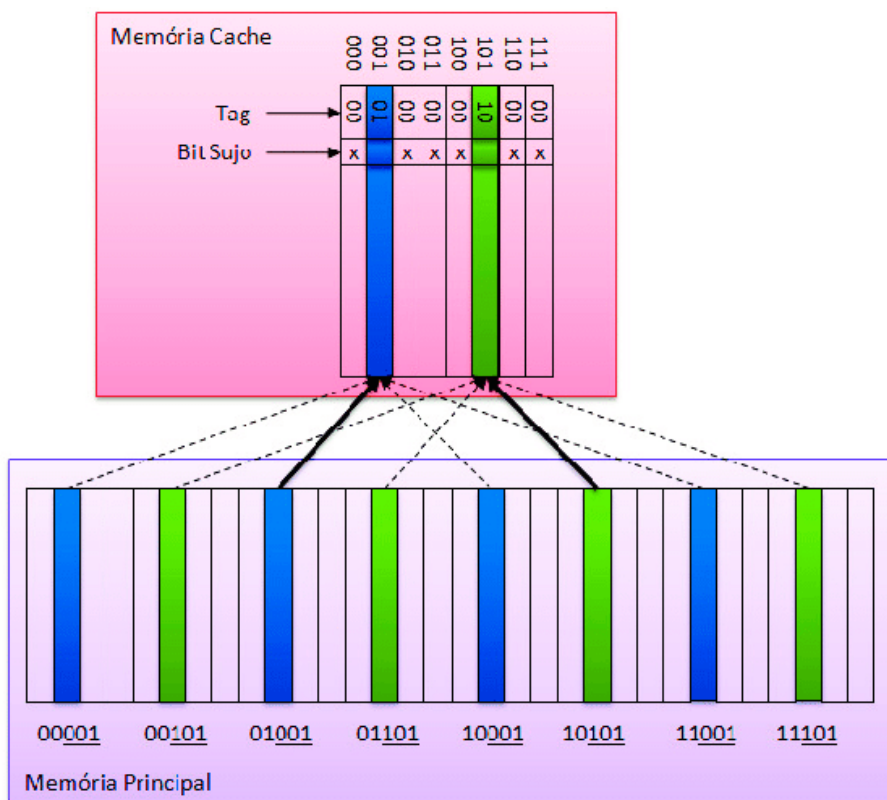
Quanto ao mapeamento da cache, também temos três tipos: o mapeamento direto, o mapeamento totalmente associativo e o mapeamento associativo por conjuntos.

**Mapeamento direto** – Neste mapeamento, cada bloco da memória principal é mapeado para um quadro da cache. O quadro a ser usado é obtido pelo resto da divisão do endereço do bloco da memória principal pela quantidade de quadros da cache. Cada quadro da cache tem três campos: o índice, o *tag* e o endereço de memória. O *tag* é usado para validar se a linha procurada é a mesma que está na cache.

Na Figura 3 podemos observar que, para cada quadro da cache, teremos a possibilidade de quatro blocos da memória principal. Assim, precisamos de dois bits no *tag* para identificar qual dos blocos da memória principal está carregado ali.

A desvantagem deste tipo de mapeamento é que se dois blocos da memória tiverem endereços com o mesmo resto de divisão, eles nunca poderão estar simultaneamente na cache, já que devem ocupar o mesmo quadro.

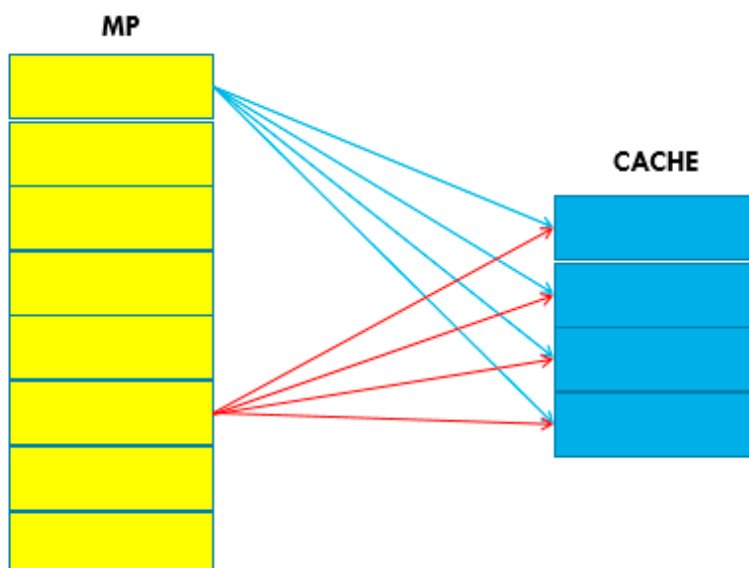
**FIGURA 3 | Mapeamento Direto**



Fonte: [https://www.researchgate.net/figure/Figura-39-Diagrama-de-uma-memoria-cache-com-mapeamento-direto-apresentando-a-tag-e-o\\_fig5\\_256293104](https://www.researchgate.net/figure/Figura-39-Diagrama-de-uma-memoria-cache-com-mapeamento-direto-apresentando-a-tag-e-o_fig5_256293104).

**Mapeamento (totalmente) associativo** – Neste mapeamento, um bloco de memória pode estar em qualquer quadro da cache. Neste caso, sempre que precisar encontrar algo, o processador deverá varrer toda a cache até encontrar (se encontrar) o que procura.

**FIGURA 4 | Mapeamento Totalmente Associativo**



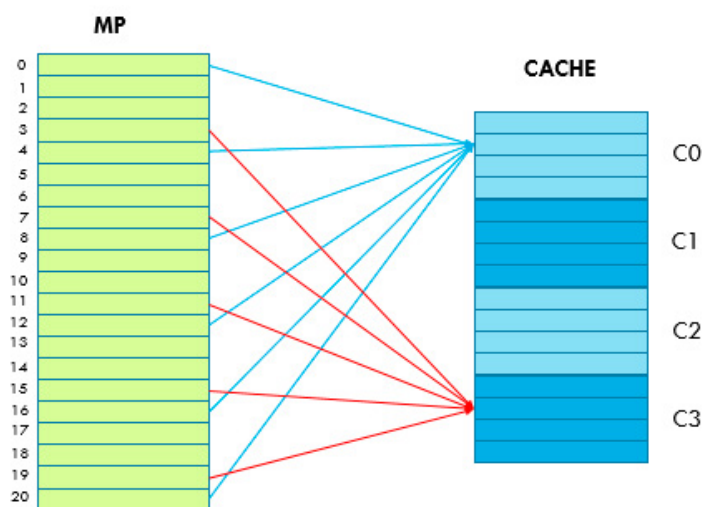
Fonte: elaborado pelo autor.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

Na Figura 4 podemos ver que um bloco da memória principal pode estar em qualquer quadro da cache.

**Mapeamento associativo por conjuntos** – Este mapeamento é um híbrido entre os mapeamentos direto e totalmente associativo. Nele, os quadros da cache são divididos em conjuntos. O bloco de memória usará então o quadro conforme o resto da divisão entre o endereço da memória principal e a quantidade de conjuntos da cache (como no mapeamento direto). Dentro do conjunto, o bloco da memória principal poderá estar em qualquer um dos quadros (como no mapeamento totalmente associativo).

FIGURA 5 | **Mapeamento Associativo por Conjuntos**



Fonte: elaborado pelo autor.

Na Figura 5 podemos ver que os blocos da memória principal cujo resto da divisão por quatro (quantidade de conjuntos da cache) é igual a 0, podem ser alocados em qualquer quadro da cache do conjunto 0 (setas azuis) e os blocos da memória principal cujo resto da divisão por quatro é igual a 3 podem ser alocados em qualquer quadro do conjunto 3.

## CONSIDERAÇÕES FINAIS

Vimos nesta aula um pouco mais sobre o subsistema de armazenamento do computador, entrando em mais detalhes sobre o relacionamento entre as memórias principal e cache.

A cache, que é uma memória mais rápida do que a memória principal, guarda o que é mais provável do processador usar. E isso é possível graças ao princípio da localidade, subdividido em localidade espacial e localidade temporal.

Vimos também que, para que a memória cache possa realmente melhorar a performance do computador, é importante que os *hits* de cache sejam superiores aos *misses* de cache, caso contrário, o *miss penalty* acabaria prejudicando em vez de ajudar o computador.

Finalmente, vimos que o correto e eficiente funcionamento da cache passa por políticas de substituição de quadros e mapeamento entre blocos de memória principal e quadros da cache.

Vamos falar um pouco sobre processadores? Hora de irmos para nossa próxima aula.

Até lá!

## MATERIAIS COMPLEMENTARES

<https://www.youtube.com/watch?v=qTvsyk-phEo> – Neste vídeo você poderá aprofundar seus conhecimentos sobre mapeamento de memória cache.

<https://www.youtube.com/watch?v=vVK6ffd9Aw4> – Conheça mais sobre os algoritmos de substituição de quadros da cache.

## REFERÊNCIAS

STALLINGS, William. *Arquitetura e Organização de Computadores: projeto para o desempenho*. 8ª edição. Editora Pearson. Livro. (642 p.). ISBN 9788576055648. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576055648>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. 3ª edição. Editora Pearson. Livro. (674 p.). ISBN 9788576052371. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576052371>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Organização estruturada de computadores*. 6ª edição. Editora Pearson. Livro. (628 p.). ISBN 9788581435398. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788581435398>>. Acesso em: 16 out. 2022.



# AULA 3 – ARQUITETURA E COMPONENTES DO PROCESSADOR

## OBJETIVO DA AULA

Conhecer os componentes do processador e sua participação no ciclo de instrução.

## APRESENTAÇÃO

Olá!

Agora é hora de estudarmos o processador, também conhecido como CPU (*Central Processing Unit*).

Vamos ver quais são seus componentes, como eles trabalham e como se comunicam entre si.

O processador executa continuamente as sequências de instruções que compõem os programas, repetindo o *ciclo de instrução*. Veremos o que é esse ciclo e como cada componente do processador participa dele.

Mãos à obra!

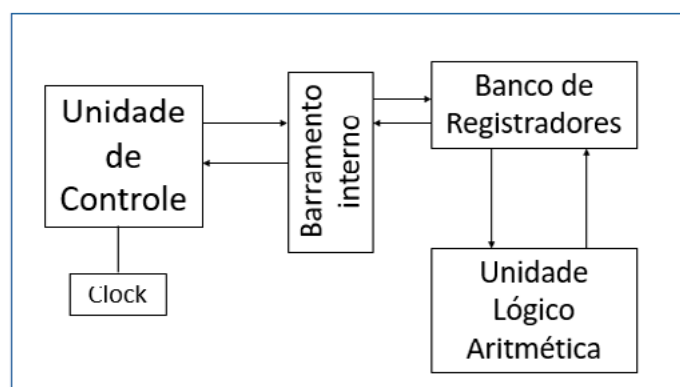
## 1. ARQUITETURA E COMPONENTES DO PROCESSADOR

Nas aulas anteriores estudamos o subsistema de armazenamento e sua organização hierárquica. Mas o sistema de memória trabalha em parceria com o processador, armazenando dados, informações e programas que o processador executa.

Vamos, a partir de agora, estudar o processador, seus componentes e seu funcionamento.

O processador, ou CPU, é organizado em três componentes principais: a Unidade de Controle (UC), a Unidade Lógica Aritmética (ALU) e o Banco de Registradores.

FIGURA 1 | **Estrutura de um Processador**



Livro Eletrônico  
Fonte: elaborado pelo autor.



Na Figura 1 vemos a estrutura de um processador. Os componentes que ali aparecem são:

- **Unidade de controle;**
- **Clock;**
- **Banco de registradores;**
- **Unidade Lógico Aritmética;**
- **Barramento interno.**

O processador trabalha executando programas ininterruptamente, desde o momento em que ligamos o computador até o momento em que o desligamos. E os programas são compostos por sequências de instruções, que dizem o que fazer passo a passo para concluirmos uma tarefa.

Assim, podemos dizer que o processador repete continuamente o *ciclo de instrução*, que é composto de 5 passos:

**1) Buscar instrução** – neste passo, o processador vai à memória buscar a instrução que está no endereço apontado por um dos registradores;

**2) Decodificar a instrução** – significa interpretar o código de operação (opcode), que consta em parte dos bits da instrução, e verificar qual instrução lógica ou aritmética deverá ser executada;

**3) Buscar os operandos** – toda instrução trabalha com um ou mais operandos, que são os dados que serão manipulados por ela. Por exemplo, se for uma soma, os operandos serão os números a serem somados.

**4) Executar a instrução** – neste passo, a operação verificada no passo 2 é efetivamente executada com os operandos buscados no passo 3;

**5) Escrever resultado** – aqui o resultado obtido no passo 4 é colocado em um dos registradores.

Como se trata de um ciclo, podemos concluir que, ao chegar no último passo, o processador volta ao primeiro passo, buscando a próxima instrução a ser executada.

Vamos agora ver as funções dos componentes do processador, analisando sua participação no ciclo de instrução.

## 1.1. BARRAMENTO INTERNO

Um barramento tem a função de transmitir bits entre um componente e outro. Temos barramentos também na placa mãe, que conectam os diversos componentes do computador. No

caso do barramento interno, ele é o responsável pela transmissão dos bits entre a Unidade de Controle, o Banco de Registradores e a Unidade Lógico Aritmética.

A principal característica de um barramento é sua *taxa de transmissão*. Ela representa a quantidade de bits que esse barramento consegue transportar por segundo e sua medida é o *bps* (bits por segundo).

Essa taxa de transmissão depende de dois fatores: a largura e a frequência do barramento e a taxa de transmissão é o resultado do produto entre esses fatores.

Como exemplo, suponha um barramento hipotético com largura de 4 bits e uma frequência de 10 hz ( $4 * 10 = 40$  bps). Um outro exemplo não hipotético e com uma frequência alta é um processador Pentium com o barramento de 64 bits trabalhando a 66 MHz. O cálculo será:

1º – Transformar os bits em bytes: 64 bits = 8 bytes (1 byte é igual a 8 bits.  $8 * 8 = 64$  bits ou  $64 \text{ bits} / 8 \text{ bits} = 8 \text{ bytes}$ ).

2º – Calcular a taxa:

$$\text{Taxa} = 8 * 66 \text{ MHz}$$

$$\text{Taxa} = 528 \text{ MB/s}$$

## 1.2. CLOCK

O clock é um circuito ligado à Unidade de Controle e tem como função emitir pulsos que determinam o ritmo em que as coisas acontecem no processador.

Para termos uma ideia da velocidade dos processadores, um processador com clock de 2 GHz (dois giga-hertz) emite dois bilhões de ciclos por segundo. Consegue imaginar essa frequência.

É importante considerar que o aumento da frequência do clock provoca o aumento da produção de calor. Assim, é necessário um sistema de dissipação desse calor para evitar o superaquecimento do processador.

Por isso, embora muitas pessoas utilizem o recurso do *overclocking*, isso não é recomendado pelos fabricantes, pois reduz o tempo de vida útil do processador.

## 1.3. UNIDADE DE CONTROLE

Como o nome já diz, a unidade de controle comanda todas as operações realizadas pelo processador. Dela partem as “ordens” que ativam os diversos componentes, sendo ela a responsável por ativar os passos do ciclo de instrução que vimos anteriormente.

## 1.4. UNIDADE LÓGICO ARITMÉTICA (ULA)

Essa unidade contém os circuitos que executam o passo quatro do ciclo de instrução. São circuitos formados pelas portas lógicas que estudamos e, como exemplo, podemos citar o somador. Após a decodificação da instrução e busca dos operandos, a Unidade de Controle ativa o respectivo circuito da Unidade Lógico Aritmética para que ele execute a operação identificada.

## 1.5. BANCO DE REGISTRADORES

O Banco de Registradores, como já vimos quando estudamos o subsistema de memórias, é um meio de armazenamento interno, situado no núcleo do processador. A importância do Banco de Registradores está no fato de que as informações que estão sendo usadas pelo processador em um determinado momento devem estar armazenadas nele.

Os registradores são de dois tipos: registradores de propósito geral e registradores de propósito específico.

Os primeiros armazenam dados em geral utilizados pelas instruções a cada ciclo.

Já os registradores de propósito específico têm funções bem definidas.

Vamos ver agora alguns dos registradores de propósito específico.

- PC (*Program counter*) ou CI (Contador de instruções) – este registrador guarda o endereço da próxima instrução a ser buscada e executada. No primeiro passo do ciclo de instrução, o processador usa esse endereço para buscar a próxima instrução e, assim que isso é feito, o endereço do PC é atualizado para estar sempre apontando para onde está a instrução seguinte;
- IR (*Instruction Register*) ou RI (Registrador de Instruções) – este registrador tem a função de armazenar a instrução que está sendo executada no momento. Isso é importante porque na instrução estão as informações que o processador precisa. Por exemplo, o *opcode* (código de operação) é usado no segundo passo (a decodificação). Além disso, a instrução contém as informações de onde serão buscados os operandos;
- REM (Registrador de Endereço de Memória) – neste registrador fica o endereço indicando a posição da memória onde será realizada uma operação;
- RDM (Registrador de Dados de Memória) – neste registrador fica guardado o dado tanto de uma operação de leitura quanto de escrita na memória.

Com a evolução dos processadores esses componentes foram sendo aperfeiçoados, mas todos eles estão presentes desde os primeiros projetos de processadores.

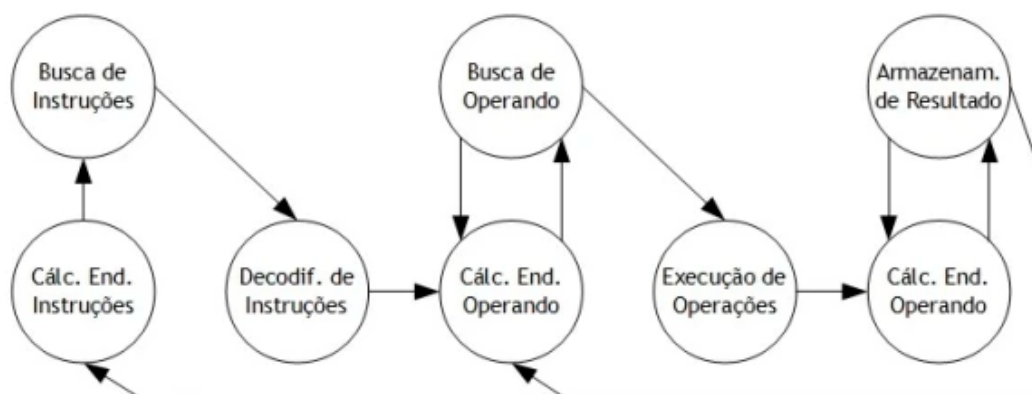
Agora veremos como os dados que o processador precisa para executar as instruções, os operandos, podem ser endereçados. Em outras palavras, veremos como o processador sabe quais os operandos que manipulará em cada instrução.

Os principais modos de endereçamento são os seguintes:

- Endereçamento direto – neste caso o endereço do operando é determinado pela própria instrução;
- Endereçamento indireto – neste caso o endereço é determinado por um registrador;
- Imediato – o operando vem na própria instrução. É o caso das constantes.

Agora que já conhecemos o ciclo de instrução e a participação de cada componente do processador nesse processo, vamos observar a Figura 2.

**FIGURA 2 | O ciclo de instrução**



Fonte: <https://cadernogeek.wordpress.com/2011/05/07/aoc-pipeline-i/>.

Pela Figura 2 podemos fazer algumas observações que nos ajudarão a entender melhor o ciclo de instrução.

Em primeiro lugar, repare que ao final do ciclo, depois que o resultado é armazenado, o endereço da próxima instrução é calculado e o PC é atualizado. Muitas vezes esse cálculo significa apenas incrementar o endereço atual com o tamanho da palavra de memória (quantidade de células que compõem uma instrução). Outras vezes esse cálculo envolve somar um número indicando que o programa dará um *salto* para uma instrução que não está em um endereço imediatamente seguinte. É o caso do que chamamos *instruções de desvio*, que não abordaremos nesta aula.

Outro fator importante a ser observado é que o endereço dos operandos também pode ser calculado e isso depende do modo de endereçamento usado na instrução e este modo é definido por um ou mais bits codificados dentro de tal instrução.

Para finalizar essa aula, vamos abordar dois aspectos relativos ao processador fundamentais para o funcionamento eficiente do computador.

O primeiro diz respeito à melhoria da performance. Há basicamente duas formas de tornarmos o processador mais eficiente: aumentar o clock ou aproveitar melhor os recursos que ele já tem.

No primeiro caso, o aumento do clock provoca o aumento do consumo de energia e geração de calor. Isso deve ser resolvido através de um sistema eficiente de dissipação de calor. Para isso, são utilizados recursos como *coolers*, que funcionam como ventoinhas jogando o calor para fora do computador ou a pasta térmica.

No segundo caso, vamos estudar na aula que vem uma série de técnicas e tecnologias que vêm sendo adicionadas aos processadores e que melhoraram sensivelmente sua performance.

Arquiteturas superescalares, *pipelining*, predição de desvios, coprocessadores de ponto flutuante, entre outras, serão estudadas quando abordarmos a evolução dos processadores.

Essas tecnologias, de maneiras diferentes, conseguiram melhorar a performance dos processadores sem mexer diretamente com a questão do clock.

## CONSIDERAÇÕES FINAIS

Vimos nesta aula que o processador possui basicamente três componentes principais – a Unidade de Controle, o Banco de Registradores e a Unidade Lógico Aritmética – interconectados pelo barramento interno e que trabalham no ritmo ditado pelo clock.

Importante também lembrar que o processador trabalha executando continuamente o ciclo de instrução, que é uma sequência de cinco passos.

Quanto aos registradores, vimos que alguns deles têm um propósito ou função específica e outros, de propósito geral, são utilizados para guardar os dados das instruções que estão sendo executadas.

É importante que você se certifique de que compreendeu o ciclo de instrução e a participação de cada componente do processador nele.

## MATERIAIS COMPLEMENTARES

<https://www.youtube.com/watch?v=gE5viQ1-7cY> – Esta é a cena de um filme clássico de 1959 (Ben Hur). Nela fica ilustrado o papel do clock do processador no personagem que toca o tambor. Quanto mais rápido ele toca, mais rápido os remadores têm que trabalhar.

[https://www.youtube.com/watch?v=\\_JEWsRu6ins](https://www.youtube.com/watch?v=_JEWsRu6ins) – Aqui você pode saber um pouco mais sobre o *overclocking* e a conveniência de sua utilização.

## REFERÊNCIAS

STALLINGS, William. *Arquitetura e Organização de Computadores: projeto para o desempenho*. 8ª edição. Editora Pearson. Livro. (642 p.). ISBN 9788576055648. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576055648>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. 3ª edição. Editora Pearson. Livro. (674 p.). ISBN 9788576052371. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576052371>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Organização estruturada de computadores*. 6ª edição. Editora Pearson. Livro. (628 p.). ISBN 9788581435398. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788581435398>>. Acesso em: 16 out. 2022.

## AULA 4 – FAMÍLIA E EVOLUÇÃO DOS PROCESSADORES

### OBJETIVO DA AULA

Abordar as características das diferentes famílias de processadores e as principais tecnologias inseridas para a aumentar sua performance.

### APRESENTAÇÃO

Olá!

Agora que você já conhece os componentes do processador e sua participação na execução das instruções, é hora de vermos todas as inovações que vêm sendo agregadas aos processadores ao longo dos anos.

Vamos entender o que são diferentes famílias de processadores e aprender como a inserção de unidades de ponto flutuante, *pipelining*, predição de desvios e outras tecnologias permitiram o aumento do desempenho dos processadores.

O assunto é bastante interessante e você verá como os projetistas conseguem a cada geração melhorar o que já parecia ótimo.

Mãos à obra!

### 1. INTRODUÇÃO

A presença de computadores nos mínimos detalhes do nosso dia a dia é fato mais do que conhecido por todos nós. Além disso, a demanda por altas capacidades de computação em áreas como a indústria, medicina, agricultura, educação, entre outras, faz com que a evolução dos computadores, e em especial dos processadores, siga um ritmo muito rápido e difícil de acompanharmos.

Vamos abordar aqui a evolução dos processadores, apresentando algumas das principais tecnologias que influenciaram decisivamente em sua velocidade e capacidade de executar tarefas de altíssima complexidade, inimagináveis há algumas décadas.

### 2. ISA E FAMÍLIAS DE PROCESSADORES

Para começar, vamos entender o que é o ISA (*Instruction Set Architecture* – Arquitetura do conjunto de instruções).

Livro Eletrônico

O ISA é parte de um modelo abstrato de um computador que define como o processador é controlado pelo *software*. Essa arquitetura age como uma interface entre o *software* e o *hardware*, especificando o que o processador consegue interpretar e executar. Em outras palavras, o ISA define como as instruções básicas são codificadas (representadas na forma binária) para que possam ser entendidas pelo processador. Esse conjunto de instruções pode ser visto como o “vocabulário” que o processador entende.

Desta definição podemos entender que processadores com conjuntos de instruções diferentes fazem parte de *famílias* diferentes.

Todo processador é capaz de realizar as mesmas operações, por exemplo, somas, subtrações, comparações etc., porém, essas operações são codificadas de forma diferente em famílias diferentes.

Para facilitar o entendimento, vamos fazer uma analogia. As palavras “água” e “water” significam a mesma coisa, respectivamente em português e inglês. Porém, são *codificadas* de formas diferentes. O mesmo acontece com os processadores. Enquanto numa determinada família uma soma de inteiros é codificada por uma determinada sequência de bits, em outra família a mesma soma é codificada por outra sequência de bits. Mas o que importa é que em ambas o resultado de  $2+2$  é igual a 4.

A definição do ISA por parte dos projetistas leva em consideração aspectos diversos, tais como o tipo de aplicação que tal família de processadores deve atender.

### 3. EVOLUÇÃO DOS PROCESSADORES

Os processadores, desde seus primórdios, vêm evoluindo como qualquer outro equipamento. Essa evolução passa pela tecnologia de fabricação, matéria-prima utilizada, capacidade de clock, arquitetura do conjunto de instruções e otimização dos recursos presentes.

Uma das principais formas de melhorar o desempenho do processador é aumentar a frequência do clock, mas isso não é simples e provoca o aumento da geração de calor e consumo de energia.

Vamos a partir de agora analisar algumas das tecnologias que permitiram verdadeiros saltos na capacidade e velocidade de processamento.



## 4. COPROCESSADORES ARITMÉTICOS

FIGURA 1 | **Coprocessador aritmético**



Fonte: <https://pt.wikipedia.org/wiki/Coprocessador>.

Os primeiros processadores fabricados pela Intel e que equipavam os computadores domésticos, que pertenciam à família x86, operavam basicamente com números inteiros. Porém, algumas funções matemáticas envolviam cálculos complexos envolvendo números não inteiros (ponto flutuante) e tais processadores efetuavam essas operações de forma lenta.

Surgiu então a figura do coprocessador aritmético que tinha a função de auxiliar o processador a executar esses cálculos de forma muito mais rápida.

Inicialmente o coprocessador era vendido como um acessório e era encaixado na placa mãe. Por ser muito caro, era pouco usado pelos usuários comuns.

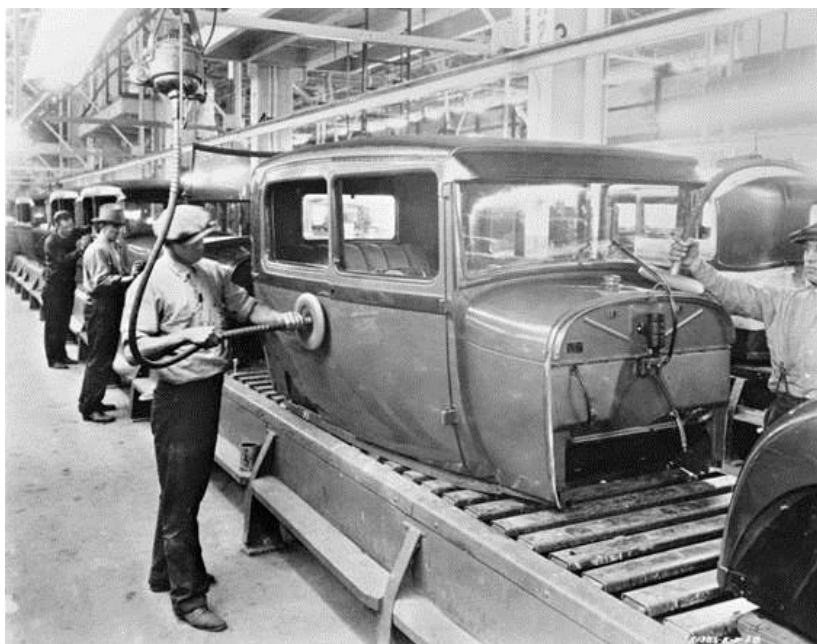
A partir da geração do 486DX o coprocessador foi incorporado no núcleo do processador, o que permitiu sua produção em maior escala e o barateamento dos preços.

## 5. PIPELINING

Esta tecnologia é baseada na ideia de linha de montagem das indústrias, criada por Henry Ford.

De forma resumida, a linha de montagem fez com que os carros, que eram construídos artesanalmente, passassem a ser construídos numa sequência de passos, cada um deles executado por um grupo de profissionais, permitindo que vários carros estivessem sendo simultaneamente montados, cada um em um estágio diferente.

FIGURA 2 | **Linha de montagem**



Fonte: <https://media.ford.com/content/fordmedia/fsa/br/pt/news/2016/02/11/ford-britanica-comemora-113-anos-com-mostra-vintage-de-fotos-e-s.html>.

Através da técnica de pipeline, os processadores passaram a executar instruções semelhantemente a uma linha de montagem. Cada instrução buscada começa seu ciclo e mesmo antes de acabar, outras instruções vão sendo buscadas, de forma que em um dado momento o processador está atendendo a mais de uma instrução, cada uma delas em um estágio diferente.

FIGURA 3 | **Execução de instruções sem pipeline e com pipeline**

B	D	BO	EX	ER	B	D	BO	EX	ER	B	D	BO	EX	ER
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

(a) – execução sem pipeline

B	D	BO	EX	ER		
	B	D	BO	EX	ER	
		B	D	BO	EX	ER
1	2	3	4	5	6	7

(b) – execução com pipeline

Fonte: elaborado pelo autor.

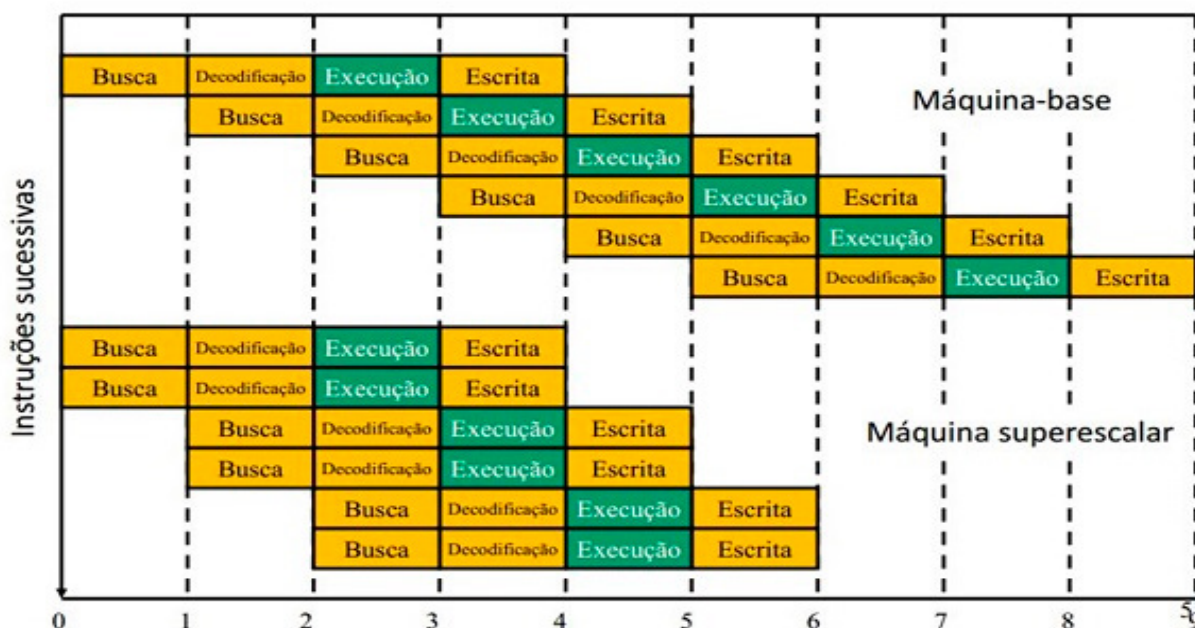
Como podemos ver na Figura 3, uma sequência com três instruções gastaria quinze ciclos para ser concluída sem a utilização do pipeline. Já a mesma sequência de instruções gastaria sete ciclos para ser concluída utilizando pipeline.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

## 6. ARQUITETURA SUPERESCALAR

As arquiteturas superescalares possibilitam que diferentes instruções sejam executadas simultaneamente em diferentes linhas de pipeline, pois permitem que se iniciem múltiplas instruções no mesmo ciclo de clock.

FIGURA 4 | **Modelo de arquitetura superescalar**



Fonte: [https://wiki.ifsc.edu.br/mediawiki/index.php/AULA\\_15\\_-\\_Microprocessadores\\_-\\_Gradua%C3%A7%C3%A3o](https://wiki.ifsc.edu.br/mediawiki/index.php/AULA_15_-_Microprocessadores_-_Gradua%C3%A7%C3%A3o).

Na Figura 4 temos a comparação de uma arquitetura simples e uma superescalar, onde todos os estágios do pipeline são duplicados.

## 7. PROCESSADORES MULTICORE

Os processadores multicore (múltiplos núcleos) são aqueles que possuem mais de um núcleo de processamento. O sistema operacional enxerga esses processadores como se fossem vários processadores de um núcleo (singlecore). Isso torna possível a busca e execução de dois ou mais fluxos de instruções completamente independentes entre si.

A grande limitação que as arquiteturas multicore enfrentam é a questão do acesso à memória. Embora os diversos núcleos possam ter suas caches internas, o acesso à memória principal é feito pelo mesmo barramento, o que configura uma espécie de gargalo.

FIGURA 5 | **Arquitetura Multicore**



Fonte: [https://www.researchgate.net/figure/Figura-2-modelo-de-arquitetura-multicore\\_fig2\\_282367700](https://www.researchgate.net/figure/Figura-2-modelo-de-arquitetura-multicore_fig2_282367700).

Observando a Figura 5, vemos que o barramento frontal (frontal *bus*), que liga o processador à memória principal na placa mãe, é o mesmo para todos os núcleos e isso deve ser gerenciado na busca das instruções que não estiverem nas caches internas dos núcleos.

## 8. PREDIÇÃO DE DESVIOS COM EXECUÇÃO ESPECULATIVA DE INSTRUÇÕES

Um dos principais motivos de atrasos no pipeline é a alta quantidade de ciclos que algumas instruções gastam. Entre essas estão as *instruções de desvio*.

Instruções de desvio são aquelas que testam uma condição e, dependendo do resultado deste teste, podem mudar o fluxo de instruções. Porém, esses testes são demorados e podem atrasar muito o início das instruções subsequentes.

A solução para minimizar esse problema é o processador executar especulativamente um dos fluxos de instruções possíveis se antecipando ao resultado do teste e guardando os resultados em registradores provisórios. Caso o fluxo executado seja aquele definido pelo resultado do teste, ótimo: o processador ganhou tempo e os resultados provisórios são tornados definitivos. Caso contrário, todos os resultados provisórios são descartados e o fluxo correto é executado.

## CONSIDERAÇÕES FINAIS

Chegamos ao fim dessa aula em que abordamos as famílias de processadores e sua evolução.

A evolução dos processadores não se resume a essas tecnologias que apresentamos aqui.

Muitas outras ideias vêm sendo incorporadas e um estudo mais aprofundado delas fugiria ao escopo desse curso.

Em todas elas a principal diretriz é explorar ao máximo os recursos que o processador já tem, utilizando-os de forma mais eficiente.

Elas permitem que a performance seja melhorada sem encarecer tanto o processador.

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

Como vimos no início, o aumento do clock também é um fator de melhoria de desempenho, mas como a velocidade dos sistemas de armazenamento não acompanha a velocidade do processador, essa melhoria acaba sendo prejudicada.

Vimos que os processadores têm uma espécie de vocabulário, que é a forma como suas instruções básicas são codificadas. A esse vocabulário chamamos de Arquitetura do Conjunto de Instruções. Portanto, as diferentes famílias de processadores são caracterizadas por diferentes arquiteturas do conjunto de instruções.

Vimos também que diversas tecnologias vêm sendo incorporadas aos processadores para melhorar sua performance e abordamos os coprocessadores aritméticos, *pipelining*, arquiteturas superescalares, arquiteturas multicore e predição de desvios.

Agora que vimos os dois primeiros subsistemas do computador, a memória e o processador, é hora de estudarmos os dispositivos de entrada e saída, que facilitam nossa comunicação com o processador.

Até lá!

## MATERIAIS COMPLEMENTARES

<https://www.youtube.com/watch?v=aje1Kj-jXOY> – Neste vídeo você poderá saber mais um pouco sobre os processadores multicore.

## REFERÊNCIAS

STALLINGS, William. *Arquitetura e Organização de Computadores: projeto para o desempenho*. 8ª edição. Editora Pearson. Livro. (642 p.). ISBN 9788576055648. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576055648>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. 3ª edição. Editora Pearson. Livro. (674 p.). ISBN 9788576052371. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576052371>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Organização estruturada de computadores*. 6ª edição. Editora Pearson. Livro. (628 p.). ISBN 9788581435398. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788581435398>>. Acesso em: 16 out. 2022.



# AULA 5 – DISPOSITIVOS DE ENTRADA E SAÍDA E INTERRUPÇÕES

## OBJETIVO DA AULA

Conhecer os dispositivos que permitem nossa interação com o computador e o mecanismo através do qual eles se comunicam com o processador.

## APRESENTAÇÃO

Nesta aula estudaremos os dispositivos de entrada e saída que compõem o terceiro subsistema do computador.

Esses dispositivos servem para promover a interface ou comunicação entre o processador e mundo externo a ele, que pode ser um usuário humano, outro computador ou algum dispositivo tal como um sensor de presença, de umidade relativa do ar ou um termômetro.

Veremos os principais dispositivos de E/S e alguns que funcionam tanto como saída quanto como entrada.

Então, hora de trabalhar!

## 1. DISPOSITIVOS DE ENTRADA E SAÍDA E INTERRUPÇÕES

Os dispositivos de entrada e saída (E/S) são componentes através dos quais o computador se comunica com o meio externo a ele.

Como sabemos, todo sistema computacional funciona executando o fluxo “Entrada → Processamento → Saída”. Isso significa que o computador recebe *dados* (entrada), realiza o processamento e fornece *informações* (saída) e, para que os dados entrem e as informações saiam, são necessários os dispositivos de E/S.

Estudaremos aqui os tipos e exemplos de dispositivos de E/S, a evolução desses dispositivos e o mecanismo de interrupções que, mediado pelo sistema operacional, provê a comunicação entre os dispositivos de E/S e o processador.

De forma simples podemos entender os dispositivos de entrada como aqueles que levam algo de fora para dentro do processador e os dispositivos de saída como aqueles que fazem a operação inversa.

Há também os dispositivos *híbridos*, que são aqueles que podem funcionar tanto como dispositivos de entrada quanto como dispositivos de saída.

Livro Eletrônico

Alguns exemplos de dispositivos de entrada são o microfone, o teclado, o mouse, o scanner, o leitor de código de barras, entre outros. E alguns exemplos de unidades de saída são o monitor, as caixas de som, impressora e outros. Finalmente, alguns exemplos de dispositivos híbridos são o *pendrive* e as telas *touch screen* dos celulares.

Os dispositivos de E/S também são conhecidos como *periféricos*, por estarem afastados do centro do processamento, conforme a Figura 1.

FIGURA 1 | **O computador e seus periféricos**



Fonte: <https://www.maxieduca.com.br/blog/tipos-dispositivos-perifericos/>.

Importante sabermos que o processador não se comunica diretamente com os dispositivos de E/S. Essa comunicação é realizada através de controladoras, que são uma espécie de processadores auxiliares, dos *drivers* e do mecanismo de interrupções.

Vamos ver agora como esses componentes participam das operações de E/S.

- **Controladora de dispositivos de E/S** – trata-se de uma placa de circuito impresso. Em sistemas mais antigos, o processador precisava conhecer cada dispositivo de E/S e isso causava dois problemas: a baixa flexibilidade em relação aos dispositivos que um computador poderia usar e o consumo de tempo do próprio processador, já que ele teria que comandar as operações de E/S. Em suma, uma controladora funciona como interface entre o dispositivo controlado e o processador;
- **Drivers de dispositivos** – são programas que ajudam o sistema operacional a prover a comunicação entre o processador e o dispositivo de E/S;

O conteúdo deste livro eletrônico é licenciado para GLEITON - 08303020692, vedada, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

- **Mecanismo de interrupções** – é a forma através da qual os dispositivos conseguem se comunicar com o processador quando precisam de algum dado para operar ou quando precisam informar o término de uma operação. Estudaremos esse mecanismo mais profundamente ainda nessa aula.

Outro fator importante em relação aos dispositivos de E/S é a questão das portas e conexões utilizadas para ligá-los ao computador. Você provavelmente já viu as diversas conexões que existem na parte de trás dos gabinetes ou nas laterais dos notebooks.

São três as principais portas onde podemos conectar os diversos dispositivos de E/S:

- Porta paralela – Era um tipo muito usado para conectar as impressoras, mas apresentava lentidão na capacidade de transferência de dados.

FIGURA 2 | **Porta paralela**



Fonte: [https://pt.wikipedia.org/wiki/Interface\\_paralela](https://pt.wikipedia.org/wiki/Interface_paralela).

- Porta serial – Conhecida também como RS 232. Costumava ser usada para a conexão de modems, mouse, scanners e outros equipamentos. Como o nome sugere, com esse tipo de porta os bits são transmitidos como numa fila, um após o outro.

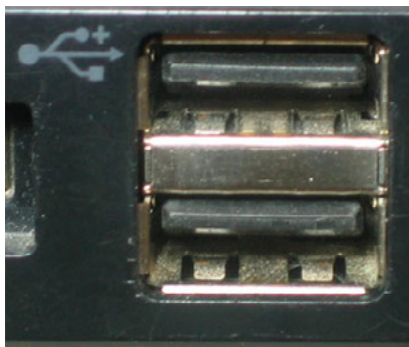
FIGURA 3 | **Porta serial**



Fonte: <https://pt.wikipedia.org/wiki/RS-232>.

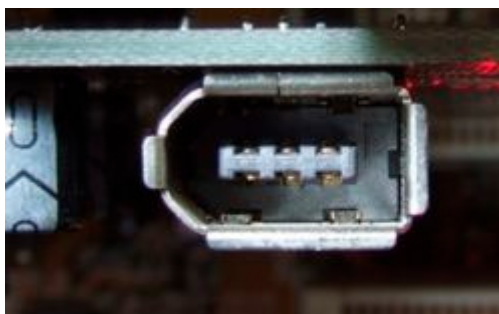
- USB (*Universal Serial Bus*) – São as portas mais usadas atualmente em virtude de sua versatilidade e facilidade de uso. Praticamente todos os dispositivos de E/S externos podem fazer uso dessas conexões.



FIGURA 4 | **Porta USB**

Fonte: [https://pt.wikibooks.org/wiki/Ficheiro:Porta\\_usb.jpg](https://pt.wikibooks.org/wiki/Ficheiro:Porta_usb.jpg).

- FireWire (IEEE 1394) – Um tipo de conexão semelhante a USB, mas com velocidades superiores. Usada para conectar dispositivos de vídeo digital, como filmadoras e câmeras digitais.

FIGURA 5 | **Portas FireWire**

Fonte: <https://pt.wikipedia.org/wiki/FireWire>.

Há vários outros tipos de portas e conectores, alguns já obsoletos e outros com uso bastante específico.

Como já mencionamos, a comunicação entre a CPU e os dispositivos de E/S é feita através do mecanismo de interrupções. Esse mecanismo é gerenciado pelo sistema operacional.

Os sistemas atuais são multiprogramados, o que significa que há diversos programas carregados na memória competindo pelo uso da CPU. Tais programas necessitam de operações de E/S para obter dados ou enviar informações e isso torna necessário o tratamento de solicitações dos periféricos.

O sistema de interrupção permite que a CPU atenda a essas solicitações sem precisar ficar “perguntando” aos periféricos se eles precisam de algo, o que traria sérios problemas de desperdício de tempo.

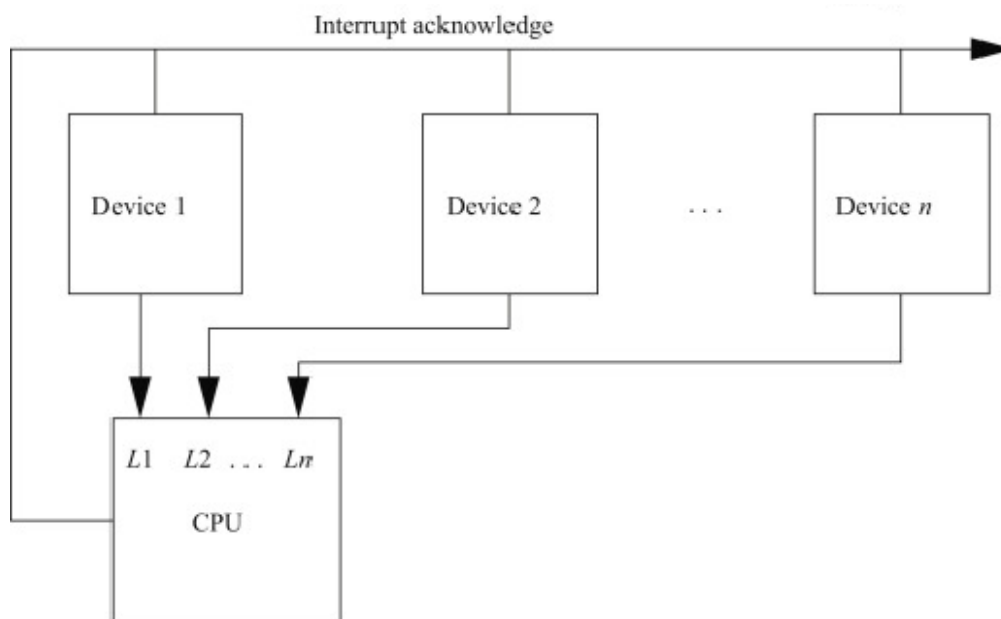
Para que as interrupções possam ocorrer, existem as IRQ's (*Interruption Requests*), que são canais de interrupções, e o PIC (*Programable Interruption Controller*), que é o responsável pelo encaminhamento das interrupções dos periféricos para a CPU.

O conteúdo deste livro eletrônico é licenciado para você, a partir de 02/02/2022, desde que, por quaisquer meios e a qualquer título, a sua reprodução, cópia, divulgação ou distribuição, sujeitando-se aos infratores à responsabilização civil e criminal.

A ocorrência de uma interrupção por parte de um periférico provoca os seguintes eventos:

- 1) O controlador de interrupções (PIC) ativa o sinal INT, que informa que uma interrupção ocorreu;
- 2) A CPU responde com sinal de confirmação (ACK), informando que está pronta para receber a interrupção;
- 3) O PIC informa qual o periférico que causou a interrupção através de uma das IRQ's;
- 4) A CPU utiliza o código da interrupção para acessar uma tabela chamada *vetor de interrupções*, onde estão os endereços das rotinas que tratam cada uma das interrupções codificadas;
- 5) A rotina é executada e a interrupção é tratada;
- 6) A CPU retoma a tarefa que havia sido suspensa para o tratamento da interrupção.

**FIGURA 6 | Esquema do mecanismo de interrupções**



Fonte: *Interrupt Request – an overview | ScienceDirect Topics*.

Todos esses passos são executados sob a supervisão do sistema operacional e o detalhamento deles será visto quando estudarmos o funcionamento dos sistemas operacionais.

Há basicamente três tipos de interrupção:

- Interrupção de *hardware*: causada pelos periféricos;
- Interrupção de *software*: em geral, causada pelo próprio sistema operacional em suas tarefas de gerência de recursos;
- Exceções: causadas por mau funcionamento do *software*, quando, por exemplo, tenta executar divisões por zero ou quando ocorre um *overflow*.

Quanto à prioridade das interrupções, elas são divididas em interrupções *maskáveis*, que são aquelas que podem ter seu tratamento cancelado ou adiado, e interrupções *não maskáveis*, que são aquelas cujo atendimento é imediato.

Um recurso importante para tornar as operações de E/S é o DMA (*Direct Memory Access* – Acesso Direto à Memória).

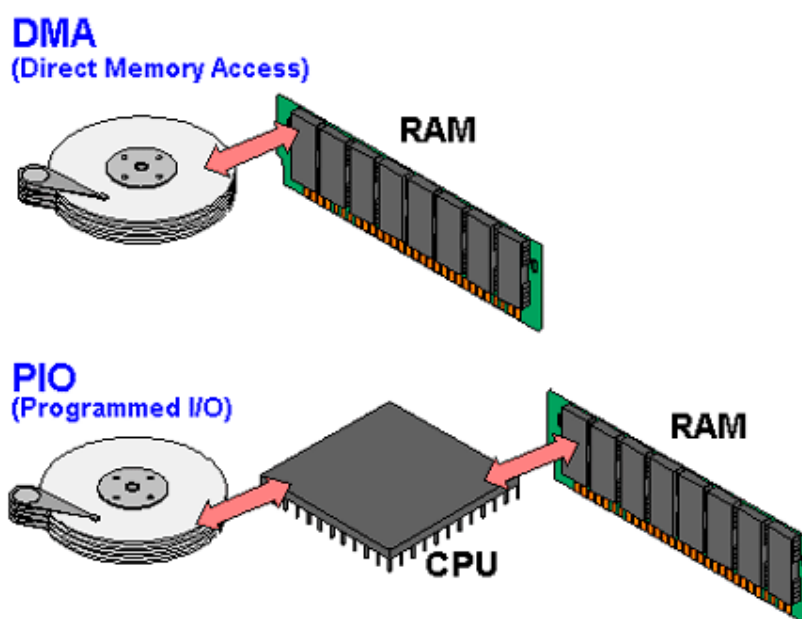
O DMA é muito útil porque minimiza a participação da CPU nessas operações. Este método faz com que os periféricos enviem ou recebam dados diretamente da memória, com pouca participação da CPU.

De forma simples, com o DMA o controlador de memória sabe onde buscar ou para onde levar os dados numa operação de E/S. Essa informação é passada no início da operação e, somente quando é concluída, a interrupção é realizada.

Sem o DMA essa transferência tem que ser comandada pela própria CPU, o que é um desperdício de um recurso tão importante.

Observe a Figura 7.

**FIGURA 7 | Operação de E/S com e sem DMA**



Fonte: DMA: o que é Direct Memory Access e como funciona? | Adrenaline

Na Figura 7 vemos uma operação de acesso ao disco rígido. Na parte de cima a operação é feita utilizando DMA e na parte de baixo a operação é sem DMA.

No segundo caso, a CPU ficará responsável pela transferência de bits entre *buffer* da controladora e a memória principal. Essa tarefa consumirá tempo e, enquanto isso, a CPU não estará atendendo aos programas do usuário ou do sistema operacional.

## CONSIDERAÇÕES FINAIS

Chegamos ao fim desta aula e agora sabemos quais são os dispositivos de E/S, como eles se comunicam com o processador e como essa comunicação pode ser agilizada através do mecanismo de interrupções e da técnica de acesso direto à memória.

Vimos que há dispositivos de entrada, de saída e híbridos, que executam as duas funções.

Com o mecanismo de interrupções, os periféricos podem solicitar a atenção da CPU para solicitar o entregar tarefas concluídas.

Também vimos que, com o acesso direto à memória (DMA), o processador passou a ter mais tempo para executar tarefas mais complexas e importantes, não desperdiçando tempo com transferências de bits entre controladoras de dispositivos de E/S e a memória principal.

## MATERIAIS COMPLEMENTARES

[Periféricos de entrada e saída – YouTube](#) – Assista a esse vídeo com um resumo sobre dispositivos de entrada e saída.

## REFERÊNCIAS

STALLINGS, William. *Arquitetura e Organização de Computadores: projeto para o desempenho*. 8ª edição. Editora Pearson. Livro. (642 p.). ISBN 9788576055648. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576055648>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Sistemas Operacionais Modernos*. 3ª edição. Editora Pearson. Livro. (674 p.). ISBN 9788576052371. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788576052371>>. Acesso em: 16 out. 2022.

TANENBAUM, Andrew S. *Organização estruturada de computadores*. 6ª edição. Editora Pearson. Livro. (628 p.). ISBN 9788581435398. Disponível em: <<https://middleware-bv.am4.com.br/SSO/iesb/9788581435398>>. Acesso em: 16 out. 2022.

## CONSIDERAÇÕES FINAIS DA UNIDADE

Chegamos ao fim desta unidade e tivemos a oportunidade de estudar os três grandes subsistemas em que os computadores são divididos: armazenamento, processamento e entrada e saída.

Começamos vendo que a memória é dividida de forma hierárquica, de forma a permitir um custo/benefício interessante para o computador, balanceando performance com preço.

O relacionamento entre a memória principal e a memória cache é projetado de forma a garantir que a cache, que é mais rápida e menor, consiga manter o que é mais provável de ser necessário ao processador. E isso é possível graças ao princípio da localidade, que reflete o comportamento padronizado da maioria dos programas.

Em seguida, estudamos os componentes do processador, suas funções e participações no ciclo de instrução, que é a rotina de trabalho do processador. Analisamos o que acontece em cada passo para que uma instrução seja concluída.

Estudamos a evolução dos processadores, abordando algumas das tecnologias inseridas ao longo dos anos para melhorar sua performance.

Finalmente, vimos o que são os dispositivos de entrada e saída, como eles se comunicam com o processador e como essa comunicação pode ser mais eficiente através do mecanismo de interrupções e do acesso direto à memória.