

# **Arquitetura e Organização de Computadores**

**Introdução a Arquiteturas de Alto  
Desempenho**

Prof. Ms. Wesley Viana

Unidade de Ensino: 4

Competência da Unidade: Aprofundar o conhecimento sobre arquiteturas de alto desempenho.

Resumo: Nesta unidade será apresentado as arquiteturas CISC e RISC, paralelas, multithread e multicore

Palavras-chave: CISC, RISC, paralelas, multithread e multicore

Título da Tele aula: Arquiteturas de Alto Desempenho

Tele aula nº: 04

# Contextualização

Você foi contratado como um dos **analistas do projeto** do desenvolvimento da arquitetura de um **novo processador** de uma importante fabricante internacional de processadores, conhecida pela qualidade e pela inovação.

Esse novo **processador** deve ser projetado para oferecer um **desempenho superior** ao dos concorrentes.

Existem diversas questões que implicam decisões impactantes para o futuro produto que exigem um compromisso (ou trade-off, em inglês) no desempenho desse **processador**.

Uma das primeiras decisões que a equipe de projeto deve tomar é em **relação à quantidade de instruções**. Ela deve escolher entre um **número maior de instruções com instruções mais complexas** ou um **número menor de instruções com sintaxe simplificada**.



www.shutterstock.com · 1039097026

Fonte: Shutterstock

# Contextualização

Haverá uma reunião de trabalho entre toda a equipe de trabalho e o gerente, e você ficou responsável de trazer diversos elementos para enriquecer a discussão.

Ao final dessa reunião vocês chegarão a um consenso sobre a **quantidade de instruções que o processador suportará, e a complexidade** de cada instrução.

O projeto já está atrasado em relação ao cronograma, e caso a decisão seja equivocada trará consequências nas próximas etapas de definição da arquitetura.

Além disso, o principal concorrente da sua empresa também está desenvolvendo um **projeto similar**.



www.shutterstock.com · 1039097026

Fonte: Shutterstock

# Contextualização

Você já participou de reuniões anteriores que não chegaram a um consenso sobre as características desejadas. Foi debatido o tamanho da instrução: se deveria ter a mesma **quantidade de elementos na instrução ou se quantidade desses elementos poderia variar.**

A conclusão dessa discussão contribuirá para definição da quantidade e complexidade das instruções com que o processador vai trabalhar.



www.shutterstock.com · 1039097026

Fonte: Shutterstock

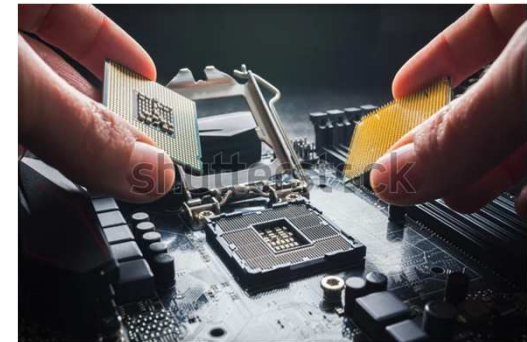
# **Introdução a Arquiteturas de Alto Desempenho**

## Sua missão:

Como material para essa reunião, você deve levantar diversas recomendações para o projeto com base nas arquiteturas **CISC e RISC**, indicando quais características seriam mais adequadas para o projeto.

Prepare um relatório com as características das arquiteturas **CISC, RISC e** as desejadas no projeto e verifique qual arquitetura está mais próxima do desejado.

O futuro desse importante projeto está em suas mãos.

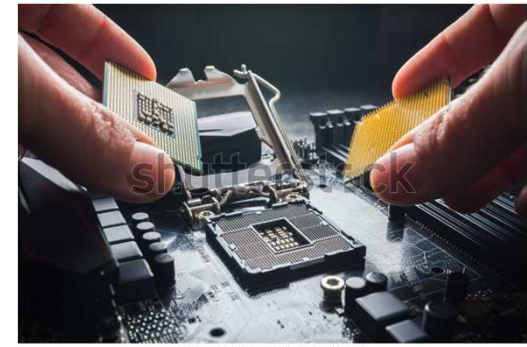


www.shutterstock.com - 1617600667

**Fonte:** Shutterstock

## Contextualizando

- Pesquisadores estudam métodos e técnicas para melhoria do desempenho e da capacidade de sistemas computacionais.
- **Software** = com desenvolvimento de **sistemas operacionais**, **Hardware** = arquitetura dos processadores.
- **Evolução do hardware:** a **velocidade de processamento da frequência do clock** (velocidade em ciclos por segundo – hertz – para executar as funções) do computador passou de poucos **MHz** em 1980 para **GHz na década de 2010**, além da criação de novas tecnologias de processamento **como pipelining e processamento paralelo**.
- **Linguagem de máquina x** linguagem de programação.
- Inicialmente foram utilizados conjuntos extensos de **instruções**, dezenas de modos de **endereçamento**.



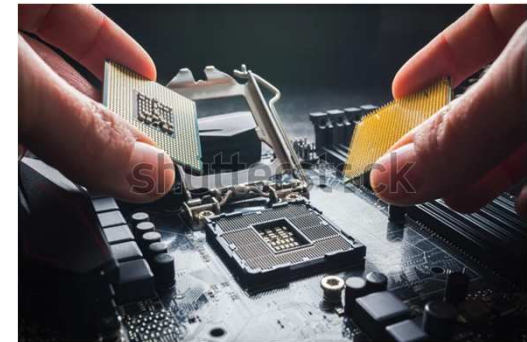
www.shutterstock.com - 1617600667

**Fonte:** Shutterstock



# Contextualizando

- Este tipo de arquitetura de computadores foi denominado **CISC – Complex Instruction Set Computers**, ou computadores com conjunto de instruções complexas.
- Avanços na área de arquitetura de processadores = com um conjunto menor de instruções, seria possível melhorar o desempenho do processador com uma arquitetura denominada **RISC - Reduced Instruction Set Computers**



www.shutterstock.com · 1617600667

**Fonte:** Shutterstock

## Características da Arquitetura CISC

A arquitetura de processadores **CISC** (Complex Instruction Set Computers) ou processadores com conjunto de instruções complexas, apresentava **grande quantidade de instruções com múltiplos modos de endereçamento**.

Os processadores com tecnologia CISC são capazes de processar centenas de conjuntos complexos de instruções simples. Isto significa que cada instrução isoladamente é considerada simples, curta e pouco potente, porém várias destas instruções agrupadas formam um conjunto complexo que é executado pelo processador.

A ideia dos fabricantes era produzir processadores cada vez mais potentes baseados na complexidade destes conjuntos de instruções (BROOKSHEAR, 2013).



www.shutterstock.com · 414568501  
**Fonte: Shutterstock**

# Características da Arquitetura CISC

Inicialmente, existia uma grande tendência a esta tecnologia de processadores, porém havia algumas desvantagens, como o desempenho reduzido justamente pelo excesso de instruções executadas pelo processador e pela velocidade de processamento ter que ser elevada para que o desempenho fosse melhorado.

Dessa forma, pode-se concluir que os projetistas de arquiteturas CISC consideram três aspectos básicos (MONTEIRO, 2010):

- Utilização de microcódigo.
- Construção de conjuntos com instruções completas e eficientes.
- Criação de instruções de máquina com complexidade similar aos comandos de linguagens de alto nível.

Na **arquitetura CISC**, o **objetivo** era a criação de um conjunto numeroso de funções, na maioria das vezes complexas e com vários operandos, o que aumentava o tempo de execução.



www.shutterstock.com · 414568501  
Fonte: Shutterstock

# Características da Arquitetura RISC

12

A ideia original era entregar um **conjunto mínimo de instruções que poderiam realizar todas as operações essenciais**: movimentação de dados, operações para Unidade Lógica Aritmética e desvios. Somente instruções explícitas LOAD e STORE tinham permissão para acessar a memória.

Alguns elementos constituem a base da arquitetura RISC: o pequeno conjunto de instruções como tamanho fixo e execução rápida da instrução.

A cada ciclo de clock uma instrução é selecionada (buscada) e executada. Essa arquitetura contém uma **menor quantidade de modos de endereçamento, maior quantidade de registradores e uso de pipelining**.



Fonte: Shutterstock

# Características da Arquitetura RISC

13

O pipeline (ou paralelismo) é a divisão de uma instrução em muitas partes, para que cada parte seja manipulada por uma parte de hardware dedicada e, com isso, todas as partes possam ser executadas em paralelo (TANEMBAUM, 2013).

Na arquitetura **RISC** existe uma diminuição da quantidade de instruções disponíveis e o modo de endereçamento em relação à arquitetura CISC.

As máquinas RISC utilizam os registradores da Unidade Central de Processamento para armazenamento dos parâmetros e variáveis em chamadas de funções e rotinas.

Processadores ARM usados em celulares são um bom exemplo de uso da arquitetura RISC. Outro exemplo de uso dessa arquitetura é em consoles, como o Nintendo e o Playstation.



Fonte: Shutterstock

## RISC x CISC

Uma vez que o tempo para buscar uma instrução não é mais uma preocupação e com memória barata para armazenar uma quantidade grande de instruções, não existe uma vantagem real em instruções CISC. (Murdocca, 2000).

Todas as instruções RISC têm tamanho fixo, uma palavra.

Todas as instruções RISC efetuam operações simples que podem ser iniciadas no pipeline à taxa de uma por ciclo de clock.

# Arquiteturas RISC x CISC

| CARACTERÍSTICAS        | RISC                    | CISC                |
|------------------------|-------------------------|---------------------|
| ARQUITETURA            | Registrador-registrador | Registrador-memória |
| TIPOS DE DADOS         | Pouca variedade         | Muito variada       |
| FORMATO DAS INSTRUÇÕES | Poucos endereços        | Muitos endereços    |
| MODO DE ENDEREÇAMENTO  | Pouca variedade         |                     |
| ESTÁGIOS DE PIPELINE   | Entre 4 e 10            | Entre 20 e 30       |
| ACESSOS AOS DADOS      | Via registradores       | Via memória         |

Fonte: produção.virtual.ufpb.br

# RISC e CISC



No relatório que você preparará para a reunião, os elementos quantidade e complexidade das instruções são importantes da arquitetura, e você deve levantar diversas características que sejam relevantes, tais como:

Arquiteturas **RISC** não apresentam microprogramação e microcódigo; todas as instruções são executadas diretamente pelo hardware.

Na arquitetura **CISC** era comum, em estruturas computacionais, adicionar uma camada de software entre o hardware e os demais programas, providenciando um novo nível de interpretação, o microcódigo. O microcódigo oferece a possibilidade de incluir ou alterar instruções sem alterar o hardware.

Entretanto, ter a maior flexibilidade não é considerada uma boa prática, pois o nível adicional de interpretação acarreta tempo de processamento para interpretação e, com isso, perda de desempenho, que não compensa às novas instruções que possam ser adicionadas.

As instruções devem ser simples de **decodificar, uma característica da arquitetura RISC**. Quanto menos complexas forem e menos formatos alternativos tiverem as opções de execução de uma instrução, será melhor.

Isto ocorre porque quando a instrução tem poucos formatos, o tempo de identificação da instrução será reduzido. Logo, o tempo de processamento será otimizado.

Somente as instruções LOAD e STORE devem referenciar a memória.

**O acesso à memória principal é uma operação mais demorada que o acesso a registradores.** Por esse motivo, é recomendado que apenas as instruções LOAD (leitura) e STORE (gravação) tenham acesso à memória. Isto ocorre na arquitetura **RISC**.

Maximize a taxa de execução de instruções.

Além do aumento da velocidade de clock, uma das formas de maximizar o desempenho do computador é por meio do **paralelismo, amplamente utilizado na arquitetura RISC.**

Implemente muitos registradores de uso geral.

Quando o processador não tem um registrador disponível para armazenar um valor resultante, ele transfere esse valor para a memória principal.

Já sabemos que a transferência de dados entre o processador e a memória principal é um processo mais lento do que a movimentação de dados dentro do processador. Assim, quanto mais registradores possíveis e de uso geral, melhor.

**A arquitetura RISC apresenta maior quantidade de registradores que a CISC.**

# **Arquiteturas Paralelas**

# Introdução a Arquiteturas Paralelas

“Uma arquitetura paralela fornece uma estrutura explícita e de alto nível para o desenvolvimento de soluções utilizando o processamento paralelo, através da existência de múltiplos processadores que cooperam para resolver problemas através de execução concorrente” (Duncan, 1990)

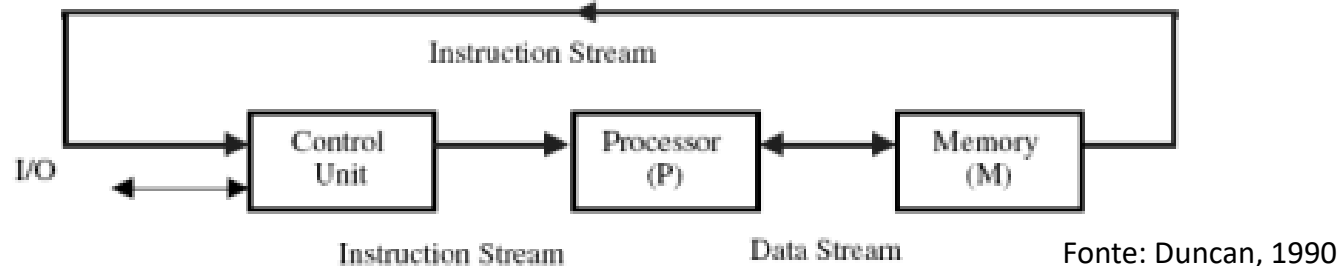
# Classificação de Máquinas Paralelas

- ✓ Classificação de Flynn – Classificação de arquiteturas paralelas genérica de acordo com as características do fluxo de instruções e o fluxo de dados (único ou múltiplo).

| Sequência de Instruções | Sequência de Dados | Nome        | Exemplos  |
|-------------------------|--------------------|-------------|---|
| 1                       | 1                  | <b>SISD</b> | Máquina clássica de Von Neumann                 |
| 1                       | Várias             | <b>SIMD</b> | Supercomputador vetorial, processador matricial |
| Várias                  | 1                  | <b>MISD</b> | Nenhum exemplo                                  |
| Várias                  | Várias             | <b>MIMD</b> | Multiprocessador, multicomputador               |

# SISD - Single Instruction, Single Data

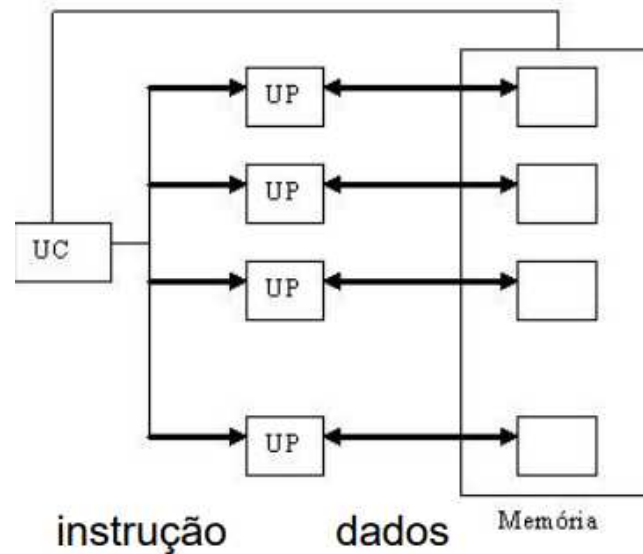
- Um único fluxo de instruções
- Um único fluxo de dados
- Contém as arquiteturas tradicionais não paralelas
- Máquinas de Von Neumann.



UC = unidade de controle UP = unidade de processamento  
MEM = memória

# SIMD - – Single Instruction, Multiple Data

- Um único fluxo de instrução
- Múltiplos fluxos de dados
- Execução síncrona da instrução para todos os dados
- Exemplos:
  - Processadores vetoriais (Cray 1)
  - GPU

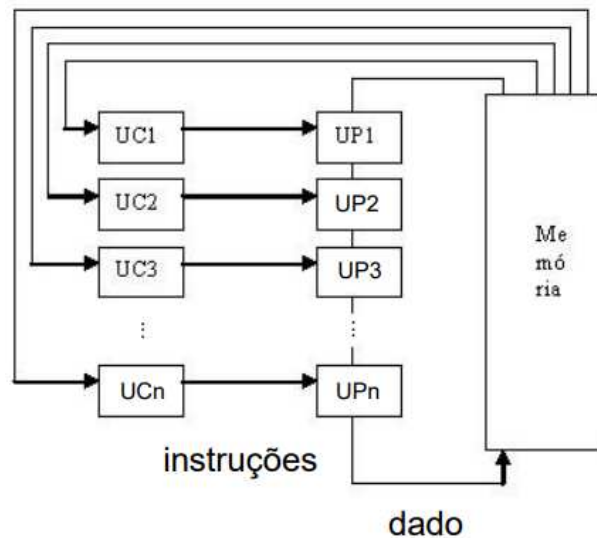


Fonte: Duncan, 1990



# MISD – Multiple Instruction, Single Data

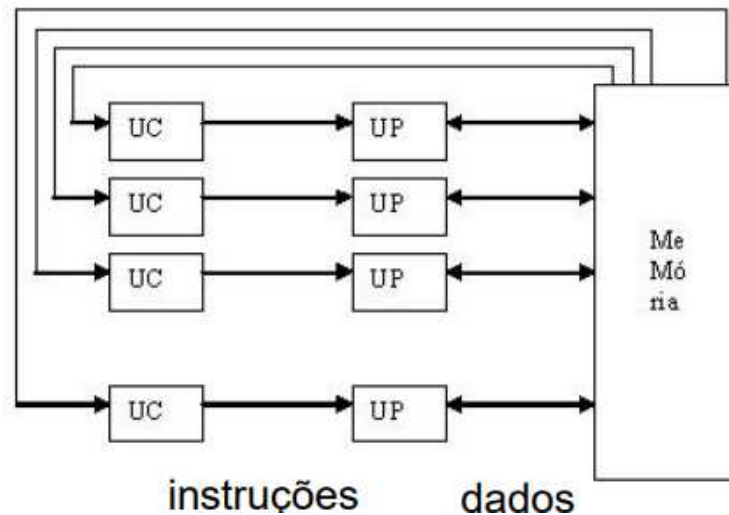
- Múltiplos fluxos de instrução
- Um único fluxo de dados – Execução de várias instruções em um único dado. – Não há exemplos.



Fonte: Duncan, 1990

# MIMD – Multiple Instruction, Multiple Data

- Múltiplos fluxos de instruções
- Múltiplos fluxos de dados – Permite a execução de instruções diferentes para cada um dos dados Exemplos:
- Arquiteturas paralelas atuais



Fonte: Duncan, 1990



Nesta etapa do projeto de novo processador de alto desempenho, foram apresentadas as arquiteturas para processamento paralelo com a implementação de pipeline, abordando suas classificações, características e utilizações. Foram detalhadas as classificações de Flynn.

Essa classificação divide os processadores da seguinte maneira: com instrução simples ou múltiplas instruções e com dados únicos ou dados múltiplos. Os quatros tipos são: SISD (única instrução, único dado), SIMD (única instrução, múltiplos dados), MISD (múltiplas instruções, único dado) e MIMD (múltiplas instruções, múltiplos dados).

No SISD há apenas um processador, e o exemplo são máquinas uniprocessadas, sendo recomendada para processamento de texto. SIMD executa uma instrução específica para vários grupos de dados, sendo recomendada para operações matriciais.

MISD não é uma arquitetura utilizada comercialmente. Já a arquitetura MIMD consiste em multiprocessadores, e a maior parte de sistemas paralelos utilizam essa arquitetura.

Nessa etapa do projeto será decidido o tipo de arquitetura em relação ao processamento, e você pesquisou sobre as arquiteturas paralelas segundo a classificação de Flynn.

Com base nos requisitos do projeto, para desenvolver um processador de alto desempenho a arquitetura mais adequada é a MIMD, mais avançada tecnologicamente e que apresenta elevado desempenho do sistema de computação.

É um conjunto de processadores que executam sequências de instruções diferentes simultaneamente em diferentes conjuntos de dados. Máquinas MIMD, que adotam vários pontos de controle, apresentam sequências de instruções e dados independentes.

Assim, nesse projeto, a arquitetura mais recomendada para ser utilizada é a **MIMD**.

Entenderam a importância da arquitetura?



Fonte: <https://gifer.com/en/XIOL9>

# **Arquiteturas Multithreaded**

# Sua missão

Você deve pesquisar as **arquiteturas multithread** e preparar um relatório com as suas principais características, que vai contribuir para a escolha da arquitetura mais apropriada para o projeto.



Fonte: Shutterstock

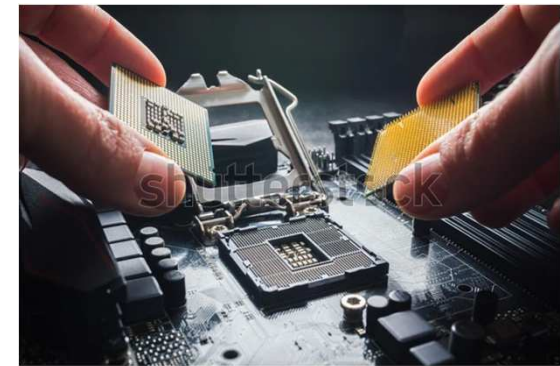


## Contextualizando

Dentro do projeto de desenvolvimento de um novo processador, iniciou-se a fase de **definição do fluxo de um processo e de seus componentes**, sendo as **threads** um desses componentes. Você deve conhecer a arquitetura do computador, que resulta em elementos visíveis a um programador.

Você já participou de reuniões anteriores que definiram por consenso que o projeto terá **troca de threads, que são mais rápidas do que as trocas de processos**.

Foi definido que a **arquitetura multithread** deve buscar as seguintes características: capacidade de envio de instruções **superescalares** e utilização de múltiplos contextos de threads. Entretanto, não houve a definição de arquitetura multithread a ser utilizada.



www.shutterstock.com - 1617600667

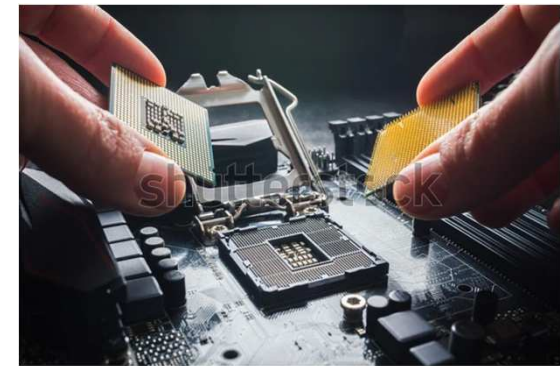
**Fonte:** Shutterstock

# Contextualizando

Desde os primeiros computadores, os cientistas se empenham para fazer as máquinas resolverem problemas de **forma melhor e mais rápida**.

A aplicação de técnicas de microeletrônica resultou em circuitos integrados (**CI**s) de alta complexidade e encapsulados em um único chip, isto é, o CI está dentro de um invólucro protetor.

Os **clocks ou relógios** aumentaram a frequência para a faixa de Gigahertz (GHz). Entretanto, existem barreiras físicas que delimitam o desempenho de uniprocessadores.



www.shutterstock.com - 1617600667

**Fonte:** Shutterstock

# Desempenho

A medida de desempenho mais relevante para um processador é a **taxa na qual ele executa as instruções**, que pode ser

$$\text{Taxa MIPS} = f * \text{IPC}$$

Onde:

**Taxa MIPS** representa a quantidade de milhões de instruções executadas em um segundo.

**$f$**  é a **frequência de clock** do processador, em MHz.

IPC é o número médio de instruções executadas por ciclo.



Fonte: Shutterstock

# Desempenho

36

Uma ordem diferente da forma que ocorrem no fluxo de instruções e começar a execução de instruções que podem nunca ser necessárias.

Entretanto, essa abordagem pode estar alcançando o limite, por causa da **complexidade e dos problemas** de consumo de energia.

Segundo Monteiro (2010), uma dessas estratégias para melhoria do desempenho é conhecida como ***superpipelining*** e ocorre quando um pipeline tem estágios que requerem menos da metade de um ciclo de relógio para executar.

Pode ser adicionado um **relógio interno**, o qual, ao realizar a



Fonte: Shutterstock

# Thread / Multithread

Como **não** é desejável o aumento da complexidade e do consumo de energia, outra abordagem que permite um grau mais alto de paralelismo em nível de instruções é denominada de **multithreading**.

Segundo Henessy e Petterson (2014, p. 193), “multithreading é uma técnica primária usada para expor mais paralelismo para o hardware”.

O fluxo de instruções é dividido em vários fluxos menores, conhecidos **como threads**, de forma que cada thread possa ser executada em paralelo.

Uma **thread** é um conjunto de tarefas existentes em um ou mais programas, executadas ao mesmo tempo pelo processador. Os **threads** definem como um processador funciona, recebendo e executando instruções. Isso acontece muito rapidamente e passa a sensação de que as ações são simultâneas.



Fonte: Shutterstock

# Thread/Processos

Uma **Thread** acontece quando um programa precisa resolver duas ou mais tarefas concorrentes (e em andamento). **Threads** não são **processos**, estando contidas **dentro de processos**.

Pode haver **múltiplas threads dentro de um mesmo processo** (ao menos uma sempre existe) e também é possível dividir recursos do computador (como memória e arquivos abertos, por exemplo), enquanto processos não podem fazê-lo.

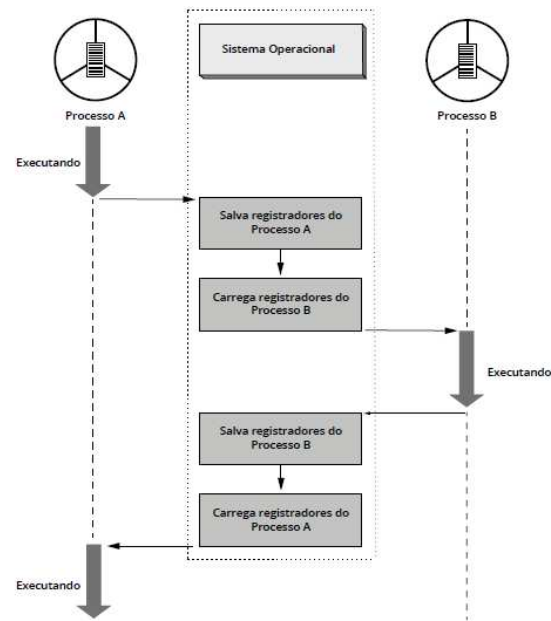
Segundo Machado e Maia (2013), pode-se definir **processo** como uma instância de um programa executando em um computador.



Fonte: Shutterstock

# Thread/Processos

O fluxo da troca de contextos está ilustrado na Figura e apresenta as etapas da troca de processo.



Fonte: Santos(2000,p.219)



Fonte: Shutterstock

# Multithreading

**Multithreading** é a capacidade que o sistema operacional possui de executar várias **threads** simultaneamente sem que uma interfira na outra.

Estas **threads compartilham os recursos do processo**, mas são capazes de ser executadas de forma independente.

Para possuir processamento **multithread “real”**, os processadores precisam ser capazes de atender **duas ou mais threads ao mesmo tempo** e não simular este efeito, atendendo-as uma por vez em um curto período de tempo.



Fonte: Shutterstock



**MULTITHREAD**

Analizando a situação-problema proposta, é necessário a determinação de qual arquitetura **multithread** será adequada.

Por sua vez, **multithreading simultâneo (SMT)** possibilita execução paralela e simultânea de várias threads. Esse é um sistema capaz de emitir diversas instruções ao mesmo tempo.

Se um thread apresenta um alto grau de paralelismo em nível de instruções, ela pode, em alguns ciclos, ser capaz de preencher todos os slots horizontais.

Em outros ciclos, as instruções de duas ou mais threads podem ser enviados. Se threads suficientes estão ativos, normalmente seria possível enviar o número máximo de instruções em cada ciclo, fornecendo um nível alto de eficiência.

Já no caso de chip multiprocessadores ou multicore, cada núcleo tem um processador superescalar de envio de duas instruções simultâneas.

Para cada processador é atribuído um thread, a partir do qual ele pode enviar até duas instruções por ciclo.

É possível combinar as potencialidades das arquiteturas **multithreading simultâneo (SMT) e chip multiprocessadores ao utilizar multithread** dentro de cada processador em um chip multicore, para potencializar o ganho de processamento, fato que é implementado em diversos processadores atuais.

# **Arquiteturas multicore**

# Sua Missão

Chegou o momento de mostrar seu conhecimento como analista de projetos para o desenvolvimento de um novo processador de alto desempenho.

Já está acertado com a equipe que serão utilizados processadores **multicore**, e isso tornará mais viável a ideia de paralelismo entre os núcleos do processador, ou seja: trabalhar com uma arquitetura que possibilite a execução simultânea, em que cada núcleo admite uma tarefa.

Você deverá apresentar para a equipe um relatório do funcionamento dos núcleos do processador em um ambiente Windows.



Fonte: Shutterstock

# Sua Missão

Você poderá utilizar a ferramenta de gerenciamento de tarefas para facilitar suas explicações.

Faça um print (cópia da tela) da execução das tarefas dos respectivos núcleos e, finalizando o seu relatório, contextualize o funcionamento dos threads e dos caches envolvidos em cada core (núcleo) do processador.

Fique à vontade para caracterizar um processador em específico.



**Fonte:** Shutterstock

# Introdução a Arquiteturas Multicore

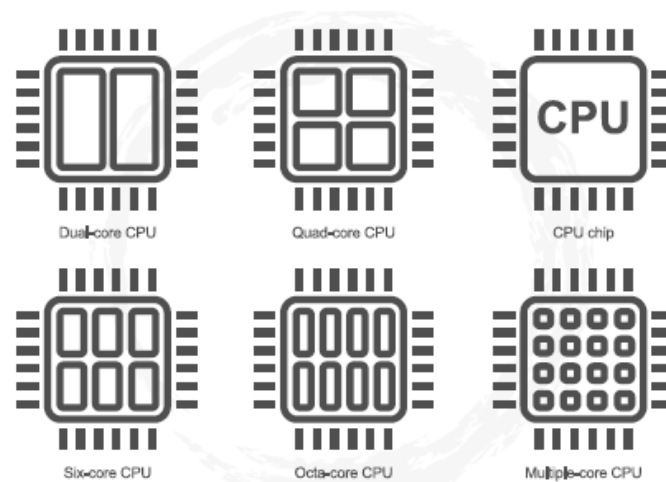
Hoje, com a necessidade de otimização de espaço físico, são desenvolvidos chips de processamento **multicore (múltiplos núcleos)**, nos quais é possível otimizar os transistores, gerando economia de energia e a redução de calor.

Um processador multicore pode ser definido como dois ou mais núcleos de processamento em uma única pastilha de silício (chip).

As tarefas a serem executadas são distribuídas entre esses núcleos do processador. Em outras palavras, cada núcleo pode executar múltiplos processos simultaneamente. A Figura ilustra a estrutura de processadores multicore.

## Processadores com múltiplos núcleos de processamento

São processadores com dois, quatro, um chip, seis, oito e dezesseis núcleos.





## Processadores com múltiplos núcleos de processamento

Cada núcleo de processamento é organizado como uma unidade de central de processamento independente, composta por unidade lógica aritmética e unidade de controle e registradores, Além dessas, há as memórias cache, que podem ser compartilhadas entre os núcleos e podem ser dedicadas. Por exemplo: temos as caches Level 1 (L1), Level 2 (L2) e Level 3 (L3) para dados e de instruções.

Ao visitar sites de fabricantes de processadores, temos um termo frequentemente utilizado, o Smart Cache (cache inteligente). Esse é o nome dado para uma tecnologia desenvolvida pela Intel, em que as caches são compartilhadas entre os núcleos de processamento do processador multicore, sendo as caches L2 e L3 compartilhadas entre os núcleos.

# Processadores com múltiplos núcleos de processamento

Os engenheiros implementaram alterações na organização de processadores permitindo o aumento do nível de paralelismo em nível de instruções, de forma a executar mais de uma instrução a cada ciclo de clock.

Posteriormente, com o surgimento da arquitetura superescalar, foram implementados vários pipelines com recursos de execução replicados para possibilitar a execução de instruções em pipelines paralelos.

Como forma de melhorar o desempenho **da arquitetura superescalar**, foi desenvolvida a arquitetura multiprocessada (Simultaneous Multiprocessing – **SMP**), que consiste em arquitetura com mais de um processador.

## Processadores com múltiplos núcleos de processamento

Esse tipo de arquitetura **tem recursos de memória, disco compartilhados e roda em um mesmo sistema operacional.** Nesse sentido, **dois processos podem ser executados ao mesmo tempo em dois processadores.**

Ainda como evolução, surgiu a arquitetura **multithread simultânea (SMT, do inglês Simultaneous Multithreading),** também chamada de **hyper-threading.**

## Processadores com múltiplos núcleos de processamento

Nessa técnica é explorado o **pipelining em nível de threads** em um processador. Assim, um processador **físico** pode simular dois processadores **lógicos**. Cada um dos processadores lógicos tem o seu próprio controlador de interrupção, bem como um banco de registradores.

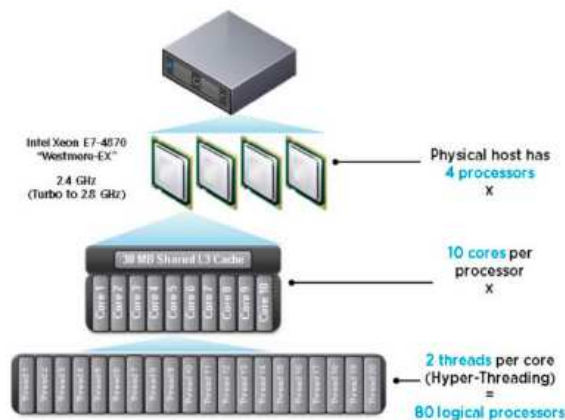
Essa arquitetura conta também com alguns recursos compartilhados, tais como, cache, ULA e barramentos.

Na próxima Figura é apresentada uma arquitetura de quatro processadores físicos, sendo dez núcleos por processador que serão compartilhados com a memória cache L3 (30 MB compartilhado entre os dez core).

# Processadores com múltiplos núcleos de processamento

Cada core tem 2 threads, ou seja, 80 processadores lógicos (quatro núcleos físicos multiplicados por dez processadores lógicos; o resultado também multiplicado por duas threads por core).

Ex: Intel Xeon 4870

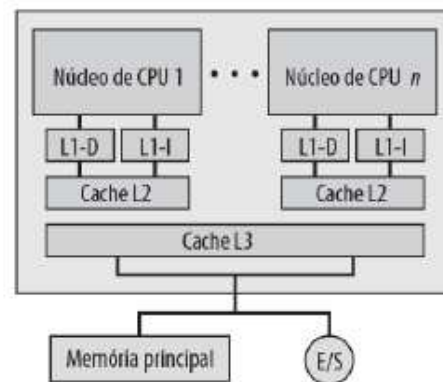


# Organização multicore

54

Com o crescimento da memória disponível e do número de núcleos, a utilização de uma memória cache L3 compartilhada, que combina o compartilhamento de cache L2 e cache L2 dedicados por núcleo, tende a resultar em um desempenho melhor que uma cache L2 compartilhada.

Os processadores **Intel i7** utilizam esse tipo de organização.



# Processadores multicore

55

Outro tipo de decisão em um projeto de sistemas com múltiplos núcleos é se os núcleos implementarão **multithread simultâneo** (SMT) ou serão **superescalares**.

Podemos destacar que o **Intel Core Duo** utiliza núcleos **superescalares**, enquanto o Core i7 utiliza **núcleos SMT**, que aumentam o número de threads suportados pelos sistemas multicore.

| Características                        | Intel Core i9 | Intel Core i7 | Intel Core i5 |
|--|---------------|---------------|---------------|
| Número de núcleo                       | 8             | 8             | 6             |
| Número de threads                      | 16            | 8             | 6             |
| Máxima frequência do processador (GHz) | Até 5,0       | Até 4,9       | Até 4,6       |
| Tamanho da memória cache (MB)          | 16            | 2             | 9             |
| Número de canais de memória            | 2             | 2             | 2             |
| Tecnologia Intel Hyper-Threading       | Sim           | Sim           | Não           |

**Novo processador**

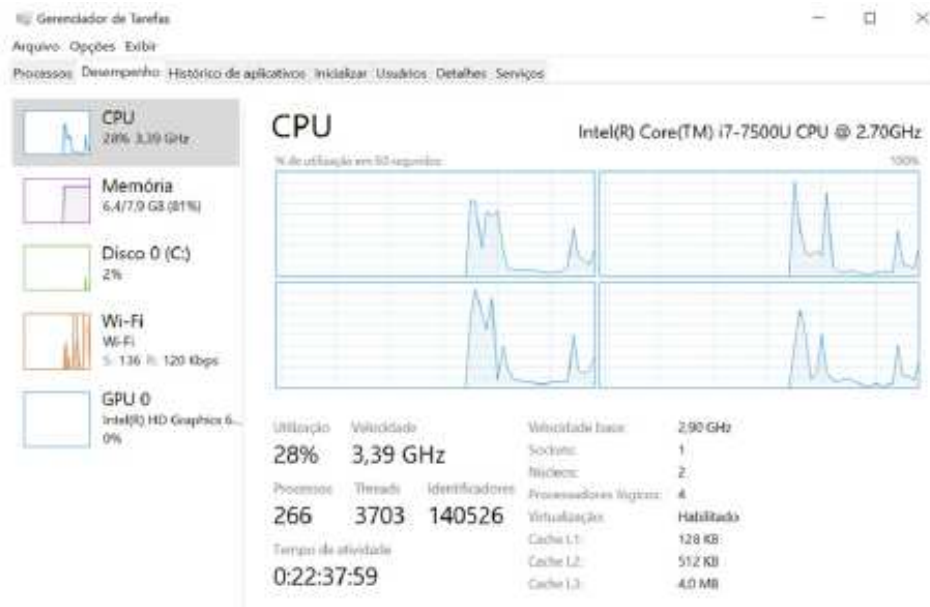


Você ficou encarregado de elaborar um relatório para a equipe de desenvolvimento do novo processador, pelo qual deverá mostrar o funcionamento dos núcleos do processador em um ambiente Windows.

Nesse caso específico foi realizada uma demonstração pelo gerenciador de tarefas do Windows, mas você poderá usar outras ferramentas para suas colocações. Finalize o relatório com uma contextualização sobre o funcionamento dos threads e dos caches envolvidos em cada core (núcleo) do processador.

Funcionamento dos núcleos do processador em um i7 de 7ª geração da Intel. Core™ i7-7500U CPU @ 2.70GHz

# Core™ i7-7500U CPU @ 2.70GHz



CPU Intel® Core™ i7-7500U CPU @ 2.70GHz

Velocidade base: 2,90 GHz

Sockets: 1

Núcleos: 2

Processadores lógicos: 4

Virtualização: Habilitado

Cache L1: 128 KB

Cache L2: 512 KB

Cache L3: 4,0 MB

Utilização: 23%

Velocidade: 3,36 GHz

Tempo de atividade: 0:23:04:53

Processos: 268

Threads: 3731

Identificadores: 143226

Fonte: captura de tela do gerenciador de tarefas do Windows elaborada pelo autor.

Podemos considerar que a execução das threads no processador pode ter o compartilhamento das informações. Cabe relatar que esse armazenamento na cache pode ser feito distintamente em outros core do processador.

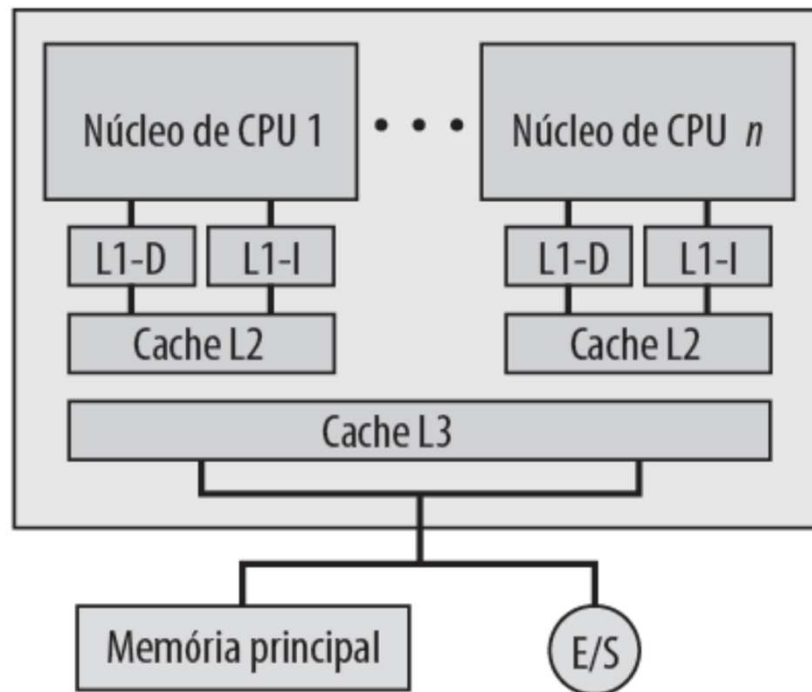
Lembre-se de que as memórias caches apresentam três níveis: a L1 é a de menor nível e menor espaço, porém muito mais rápida que a L2 e L3. A L1 é dividida em duas partes, uma para instruções e outro para os dados. A L2 é destinada para gravação de dados e informações, e a L3 é a maior entre os três níveis, e pode dar apoio a qualquer core para execuções de tarefas.

De acordo com Stallings (2013), existe um ganho quando há um compartilhamento do cache L2 e em paralelo há cache L2 dedicada, como ocorre no caso de Intel Core i7, em que cada núcleo tem a sua cache L2 dedicada, e os núcleos compartilham uma memória cache L3.

Uma das vantagens é a que interferência construtiva pode reduzir as taxas de falhas do sistema, ou seja, se uma thread em um core acessa uma posição de memória, esse acesso traz para linha de conteúdo a posição de referenciada para a cache compartilhada. Caso outra thread de outro core tente acessar depois, as posições de memória já estarão disponíveis na cache compartilhada no chip, gerando maior rapidez de acesso.

Utilizando algoritmos adequados de substituição de linha, existe uma alocação dinâmica da quantidade de cache compartilhada para cada core, de modo que threads com maiores conjuntos de trabalho apresentam mais cache. Além disso, a comunicação entre processadores é facilitada quando implementada por meio das posições de memória compartilhadas.

Assim, a estrutura multicore do processor pode ser representada na Figura.



Para finalizar, a 10ª geração de processadores Intel® Core™ dispõe de gráficos Intel® Iris® Plus, que proporcionam uso de inteligência artificial (IA) no computador em larga escala para acelerar o desempenho.

Esses processadores possibilitam **um novo nível de integração** para aprimorar as experiências com PCs atuais e as necessárias para o futuro.

Como pudemos verificar, a organização e arquitetura do processador evoluíram muito com o tempo. Como essa arquitetura pode mudar ainda nos próximos anos?

# Recapitulando

- Introdução à arquitetura de alto desempenho;
- Arquiteturas Paralelas;
- Arquiteturas Multithreaded;
- Arquitetura Multicore.



