

ARQUITETURA DE COMPUTADORES



AULA 09



AGENDA

PARALELISMO A NÍVEL DE INSTRUÇÃO

~~PIPELINING (PIPELINE)~~

~~HAZARD DE DADOS: Forwarding X Stalls~~

SUPERSCALARIDADE

EXECUÇÃO FORA DE ORDEM

AGENDAMENTO DINÂMICO

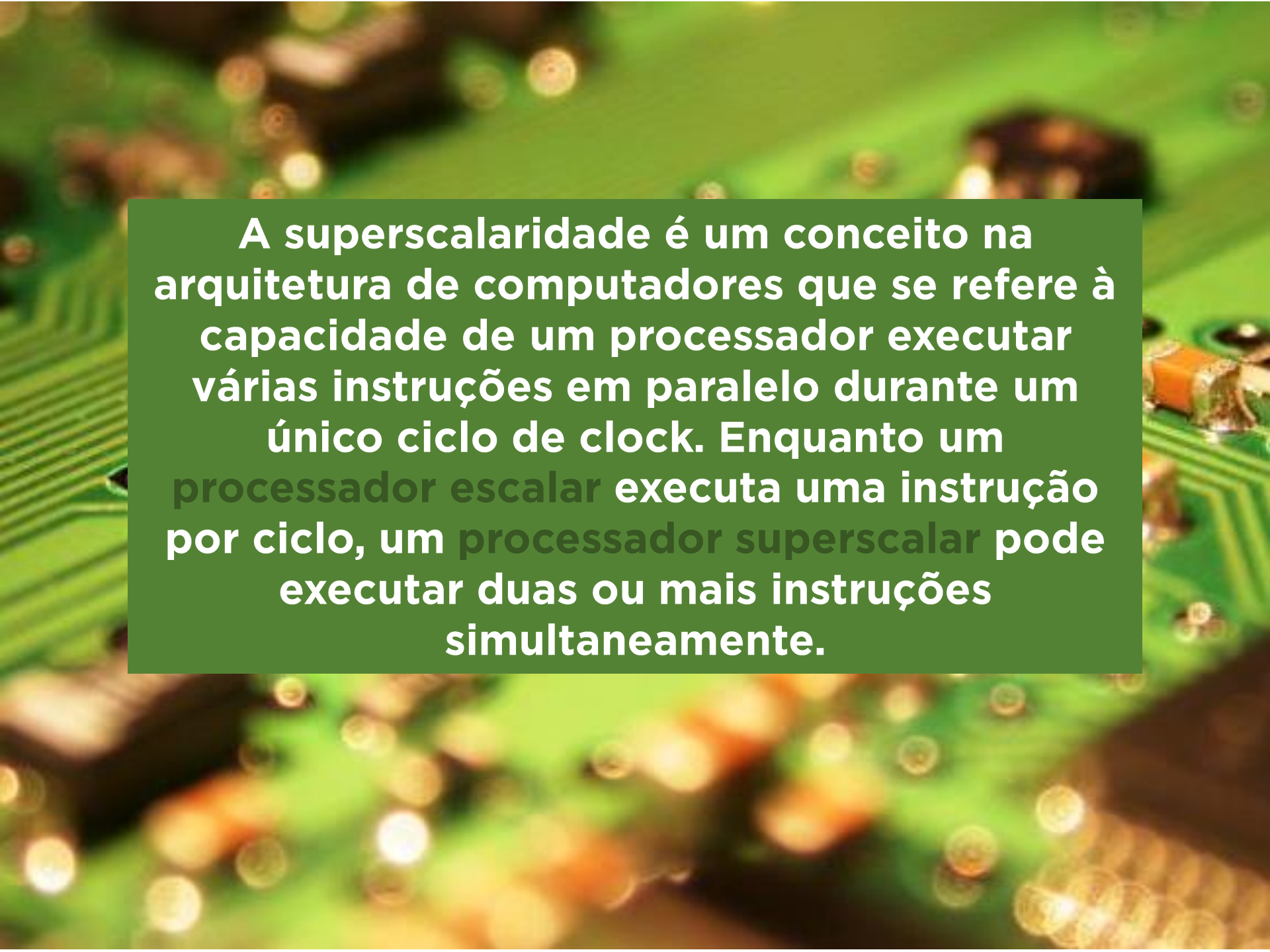


PARALELISMO A NÍVEL DE INSTRUÇÃO

Paralelismo a nível de instrução (ILP) refere-se à capacidade de um processador executar várias instruções em paralelo, aumentando assim o desempenho



SUPERSCALARIDADE
?

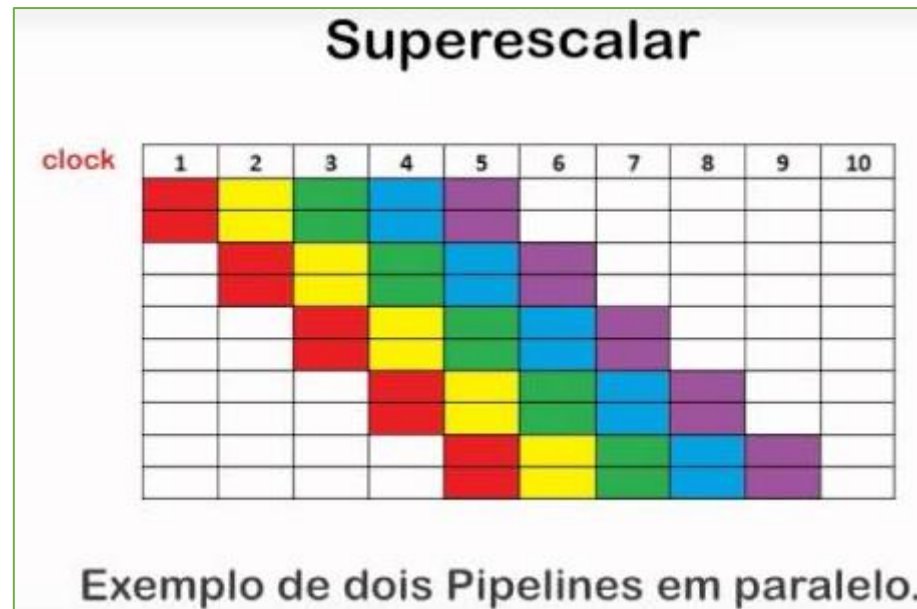


A superscalaridade é um conceito na arquitetura de computadores que se refere à capacidade de um processador executar várias instruções em paralelo durante um único ciclo de clock. Enquanto um processador escalar executa uma instrução por ciclo, um processador superscalar pode executar duas ou mais instruções simultaneamente.



SUPERESCALARIDADE

“a superscalaridade é uma técnica avançada de design de processadores que busca otimizar o desempenho, permitindo a execução paralela de várias instruções. Ela representa uma evolução em relação aos processadores escalares tradicionais, oferecendo uma abordagem mais eficiente para a execução de instruções em um único ciclo de clock”.





SUPERESCALARIDADE

RISC CLÁSSICO: 5 ESTÁGIOS

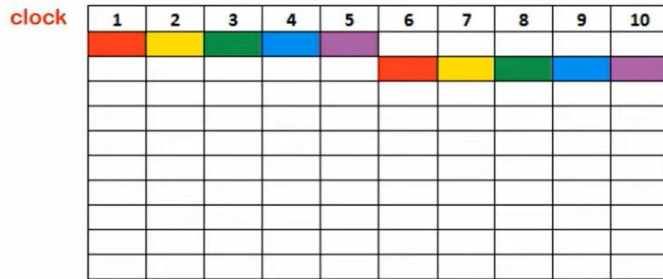
Busca as Instruções

ARQUITETURA DE COMPUTADORES



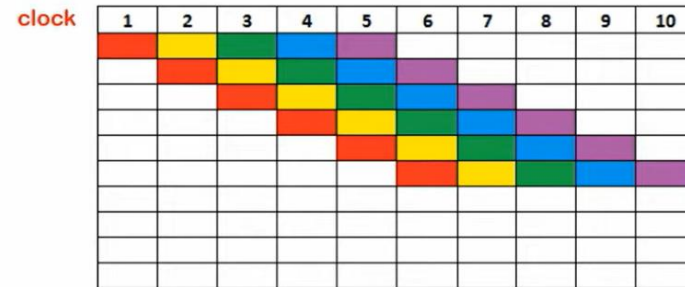
SUPERESCALARIDADE

Sem Pipeline



Espera acabar um ciclo para começar outro.

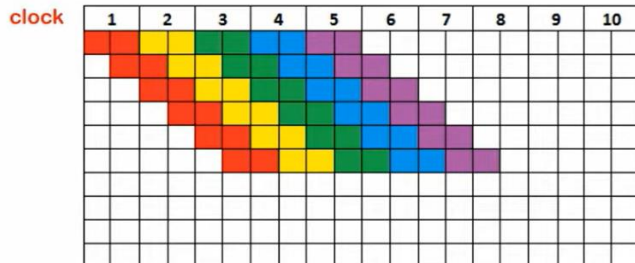
Pipeline



Começa um estágio assim que ele é desocupado.
Não espera terminar o ciclo.

CREATED USING
RAW

Superpipeline



Exemplo: Meio clock para cada instrução



Superescalar



Exemplo de dois Pipelines em paralelo.

CRE



SUPERESCALARIDADE **Principais Características**

Unidades de Execução Múltiplas:

- ❑ Processadores superscalares têm múltiplas unidades de execução para operações aritméticas, lógicas e outras instruções.
- ❑ Isso permite que diferentes instruções sejam processadas simultaneamente.

Decodificação de Instruções:

- ❑ A fase de decodificação de instruções é aprimorada para identificar instruções independentes que podem ser executadas em paralelo.
- ❑ As instruções são divididas em grupos que podem ser processados simultaneamente.

Pipeline de Instruções Aprimorado:

- ❑ A superscalaridade muitas vezes se integra a uma arquitetura de pipeline mais avançada.
- ❑ Os estágios do pipeline são otimizados para permitir a execução simultânea de várias instruções.



SUPERSCALARIDADE

Principais Características

Emissão Dinâmica de Instruções:

- ❑ Em vez de seguir rigidamente a ordem de instruções do programa, um processador superscalar pode reorganizar dinamicamente as instruções para otimizar a execução paralela.
- ❑ Isso é feito considerando as dependências de dados e as características de execução de cada instrução.

Problema de Dependência de Dados:

- ❑ A dependência de dados ocorre quando uma instrução depende do resultado de outra.
- ❑ Mecanismos como forwarding (adiantamento) e detecção de hazards são usados para mitigar o impacto das dependências de dados e permitir a execução paralela.



SUPERSCALARIDADE BENEFÍCIOS

✓ Melhoria do Desempenho:

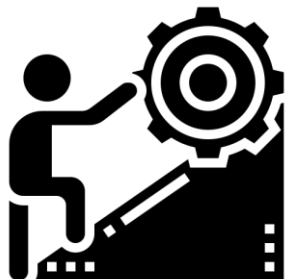
A capacidade de executar várias instruções simultaneamente leva a um aumento significativo no desempenho do processador.

✓ Aproveitamento Máximo dos Recursos:

A superscalaridade permite um aproveitamento mais eficiente das unidades de execução e dos estágios do pipeline.

✓ Execução Dinâmica e Adaptação:

O processador pode ajustar dinamicamente a ordem de execução das instruções com base nas condições em tempo de execução.



SUPERESCALARIDADE “DESAFIOS”

❖ Controle de Hazards:

Lidar com hazards, como dependências de dados, é complexo e requer técnicas avançadas, como forwarding e detecção dinâmica de hazards.

❖ Aumento da Complexidade do Projeto:

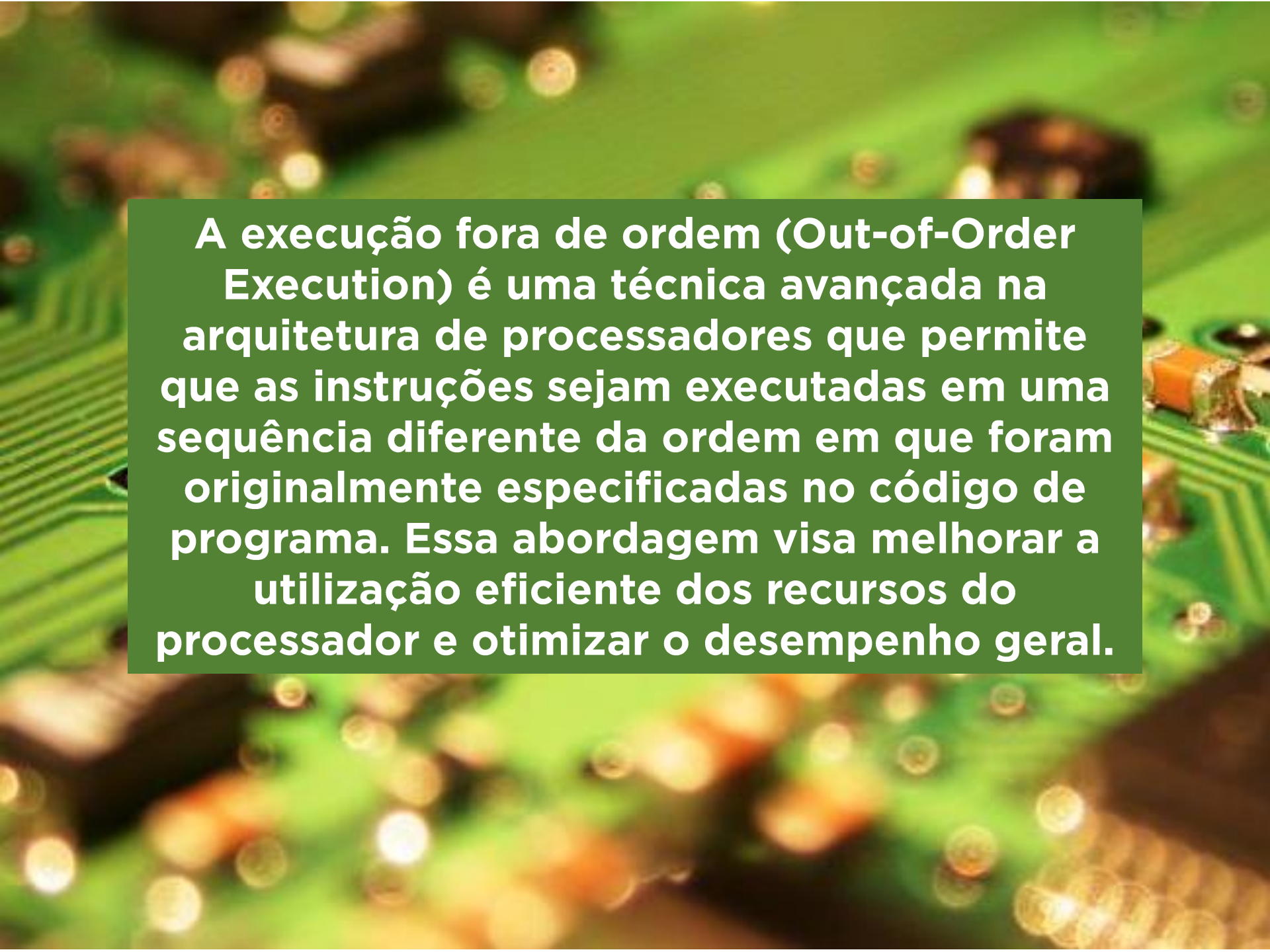
Projetar e fabricar processadores superscalares é mais complexo devido à necessidade de coordenar várias unidades de execução.

❖ Consumo de Energia:

A execução de várias instruções simultaneamente pode aumentar o consumo de energia, o que é uma consideração importante em dispositivos móveis e ambientes com restrições de energia.



**EXECUÇÃO FORA DE
ORDEM?**

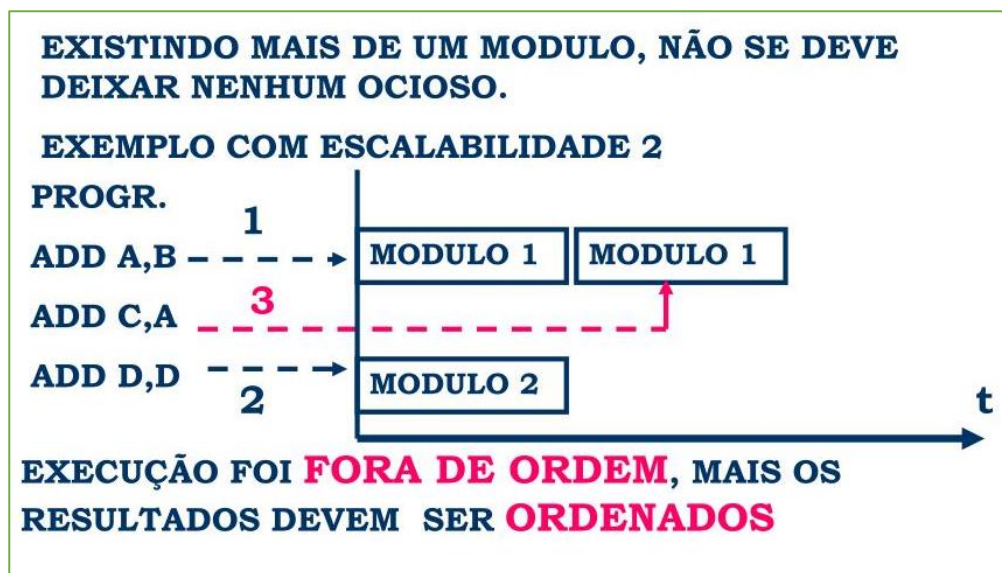


A execução fora de ordem (Out-of-Order Execution) é uma técnica avançada na arquitetura de processadores que permite que as instruções sejam executadas em uma sequência diferente da ordem em que foram originalmente especificadas no código de programa. Essa abordagem visa melhorar a utilização eficiente dos recursos do processador e otimizar o desempenho geral.



EXECUÇÃO FORA DE ORDEM

A execução fora de ordem é uma estratégia avançada que busca maximizar a eficiência do processador, permitindo que instruções sejam reorganizadas dinamicamente para otimizar o uso de recursos e melhorar o desempenho geral.





EXECUÇÃO FORA DE ORDEM COMO FUNCIONA

Decodificação e Reorganização Dinâmica:

- ❑ As instruções são decodificadas e reorganizadas dinamicamente durante a fase de execução.
- ❑ O processador avalia as dependências de dados e a disponibilidade de recursos antes de decidir a ordem de execução.

Paralelismo de Instruções:

- ❑ Instruções independentes são identificadas e executadas em paralelo, mesmo que originalmente estivessem fora de ordem no fluxo do programa.
- ❑ Isso aumenta a utilização eficiente das unidades de execução e melhora o desempenho geral.

Detecção e Solução de Hazards:

- ❑ Mecanismos, como renomeação de registradores e reordenação dinâmica de instruções, são utilizados para contornar hazards (conflitos) e manter a corretude do resultado final.



EXECUÇÃO FORA DE ORDEM BENEFÍCIOS

✓ **Aproveitamento Máximo dos Recursos**

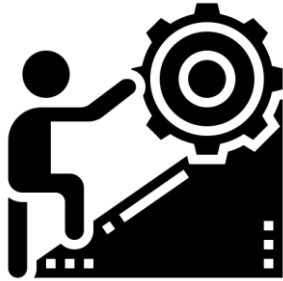
Permite que o processador execute instruções que não estão dependendo dos resultados de instruções anteriores, mesmo que estas ainda não tenham sido concluídas.

✓ **Redução de Ociosidade**

Minimiza os ciclos de clock ociosos, pois o processador não precisa esperar pela conclusão de todas as instruções anteriores antes de iniciar as próximas.

✓ **Melhoria do Desempenho**

Resulta em um aumento geral no desempenho, especialmente em situações em que o código possui dependências de dados e outras instruções podem ser executadas enquanto algumas estão aguardando dados.



EXECUÇÃO FORA DE ORDEM “DESAFIOS”

❖ **Controle de Hazards:**

A detecção e resolução de hazards tornam-se mais complexas, exigindo mecanismos avançados para garantir que as instruções sejam executadas corretamente.

❖ **Dificuldade no Predict Branches:**

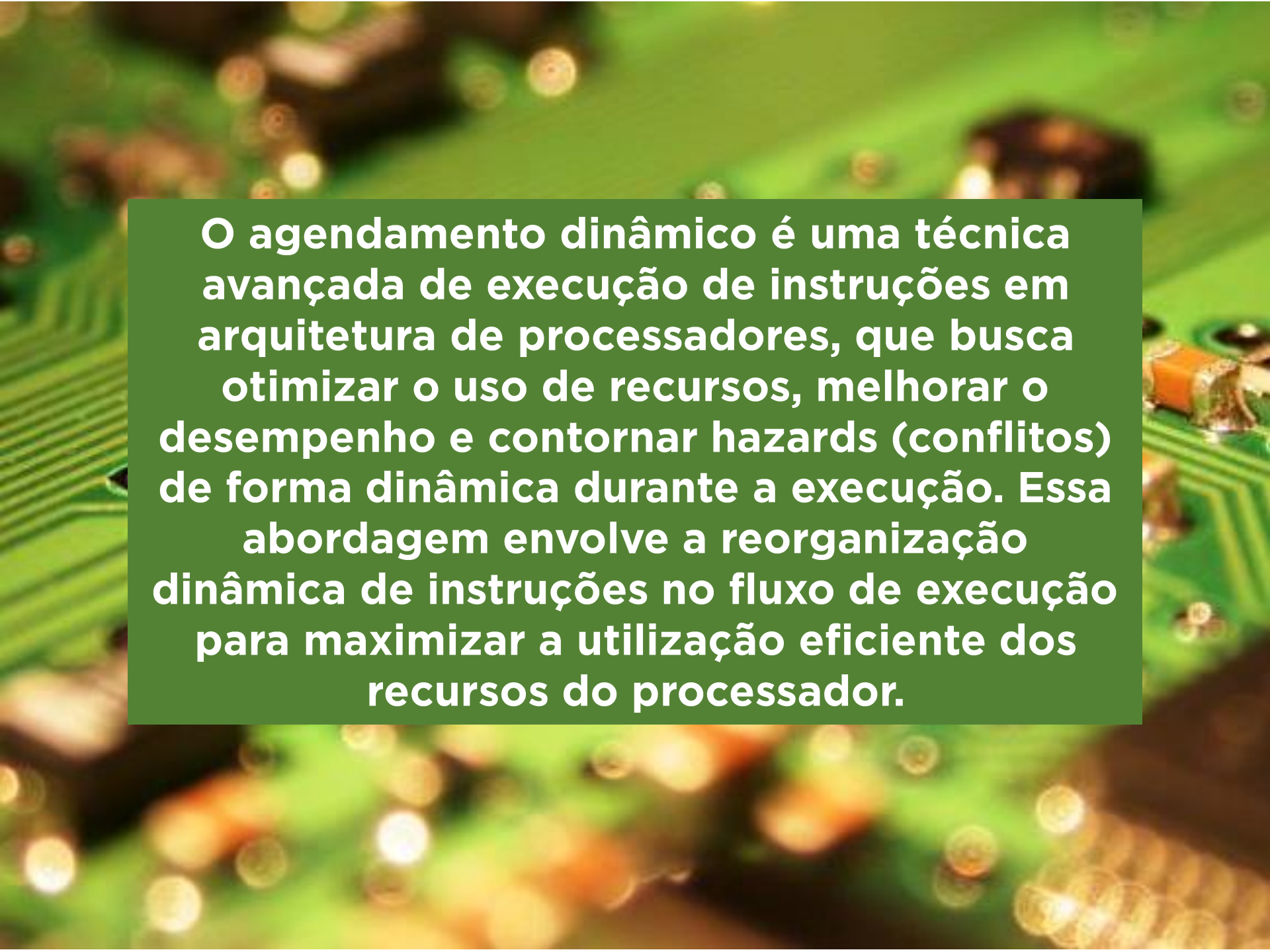
Prever condicionalmente os desvios (branches) se torna mais desafiador, pois as instruções podem ser reorganizadas dinamicamente.

❖ **Complexidade de Projeto:**

Implementar uma execução fora de ordem requer uma arquitetura de processador mais complexa, aumentando a dificuldade de projeto.



AGENDAMENTO DINÂMICO?



O agendamento dinâmico é uma técnica avançada de execução de instruções em arquitetura de processadores, que busca otimizar o uso de recursos, melhorar o desempenho e contornar hazards (conflitos) de forma dinâmica durante a execução. Essa abordagem envolve a reorganização dinâmica de instruções no fluxo de execução para maximizar a utilização eficiente dos recursos do processador.



AGENDAMENTO DINÂMICO

COMO FUNCIONA	
Decodificação Inicial	As instruções são decodificadas inicialmente na ordem do código do programa.
Análise de Dependências	O processador avalia as dependências de dados entre as instruções. Identifica quais instruções podem ser executadas independentemente e quais estão sujeitas a dependências.
Execução Dinâmica	Instruções independentes são executadas fora da ordem original, se possível, para evitar atrasos causados por dependências.
Renomeação de Registradores	Técnica comum para contornar hazards de dados, onde registradores virtuais são usados para evitar conflitos.
Reordenação do Resultado	Após a execução fora de ordem, os resultados são reordenados para refletir a sequência original das instruções.



DIFERENÇAS

enquanto o **agendamento dinâmico** abrange uma variedade de técnicas para otimizar a execução, a **execução fora de ordem** é uma estratégia específica que prioriza a execução de instruções independentes de forma a evitar ciclos ociosos no pipeline.

Ambos são conceitos associados à otimização do desempenho em arquiteturas de processadores modernas.



DIFERENÇAS

Agendamento Dinâmico	Execução Fora de Ordem
Envolve a reorganização dinâmica de instruções durante a execução.	Apenas as instruções que não estão sujeitas a dependências são executadas fora da ordem.
Busca otimizar o uso de recursos e contornar hazards de forma dinâmica.	O foco principal é evitar atrasos causados por dependências de dados.
Pode incluir técnicas como a renomeação de registradores e adaptação a condições variáveis.	Pode envolver a renomeação de registradores, mas o objetivo principal é a maximização da utilização do pipeline.



Atividade - *ILP* Apresentação