

# **ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES**

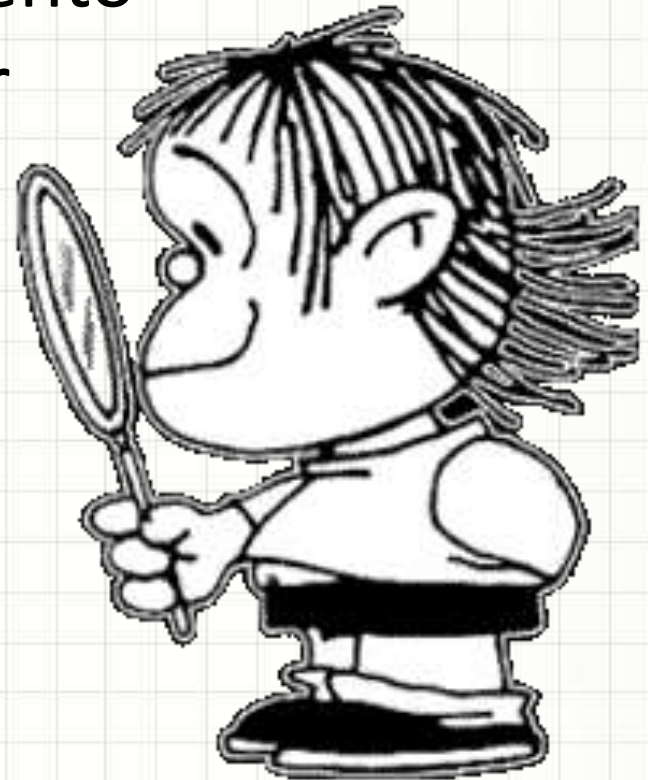
## **INTRODUÇÃO AO PARALELISMO: PROCESSADORES SUPERESCALARES**

Prof. Dr. Daniel Caetano

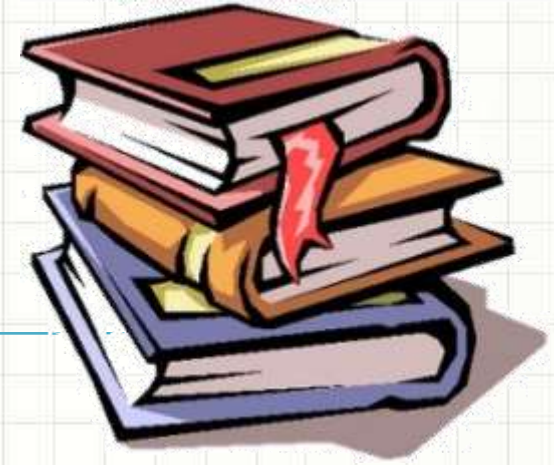
2012 - 2

# Lembretes

- Compreender o funcionamento da Arquitetura Superpipeline
- Compreender o funcionamento da Arquitetura Superescalar
- Introduzir o conceito de multiprocessamento
- Apresentar o conceito de computador completo
- **Lembretes:**
  - Lista 2!



# Material de Estudo



---

## Material

## Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>  
(Aula 12)

Apresentação

<http://www.caetano.eng.br/>  
(Aula 12)

Material Didático

-

Biblioteca Virtual

Arquitetura e Organização de Computadores, páginas  
647 a 652

---



# INTRODUÇÃO

# Introdução

- Aula passada: vantagem do Pipeline

Sequência no Tempo	SEM pipeline		COM pipeline	
	Busca	Execução	Busca	Execução
0	I1	-	I1	-
1	-	I1	I2	I1
2	I2	-	I3	I2
3	-	I2	I4	I3
4	I3	-	I5	I4

- Ocupar partes ociosas do processador



# **ARQUITETURA SUPERPIPELINE**

# Pipeline de Múltiplos Estágios

- Como visto na aula passada...

Tempo	BI	DI	CO	BO	EI	EO
0,00	I1	-	-	-	-	-
0,33	I2	I1	-	-	-	-
0,66	I3	I2	I1	-	-	-
1,00	I4	I3	I2	I1	-	-
1,33	I5	I4	I3	I2	I1	-
1,66	I6	I5	I4	I3	I2	I1
2,00	I7	I6	I5	I4	I3	I2

# Pipeline de Múltiplos Estágios

- Como visto na aula passada...

Tempo	BI	DI	CO	BO	EI	EO
0,00						
0,33						
0,66						
1,00						
1,33						
1,66						
2,00	17	16	15	14	13	12

**Será que todos estes  
estágios demoram  
exatamente o mesmo  
tempo para serem  
concluídos?**



# Superpipeline

- No pipeline tradicional
  - **Tempo da instrução:** Limitado pelo estágio mais lento
- Exemplo:

$\phi$	Tempo	E1	E2	E3	E4
1	0,00	I1	-	-	-
2	0,50	I2	I1	-	-
3	1,00	I3	I2	I1	-
4	1,50	I4	I3	I2	I1
5	2,00	I5	I4	I3	I2
6	2,50	I6	I5	I4	I3
7	3,00	I7	I6	I5	I4

Tempos Iguais  
 $E_n = 1$  ciclo

1 Instrução  
por Ciclo

# Superpipeline

- No pipeline tradicional
  - **Tempo da instrução:** Limitado pelo estágio mais lento
- Exemplo:

$\phi$	Tempo	E1	<b>E2</b>	E3	E4
1	0,00	I1	-	-	-
2	0,50	I2	I1	-	-
3	1,00	I2	I1	-	-
4	1,50	I3	I2	I1	-
5	2,00	I3	I2	-	I1
6	2,50	I4	I3	I2	-
7	3,00	I4	I3	-	I2

E1/E3/E4 = 1 ciclo  
**E2 = 2 ciclos**

0,5 Instrução  
por Ciclo

# Superpipeline

- Tempo de cada estágio: controlado pelo clock
  - Se estágio lento: 2 ciclos de clock...
  - Uma instrução a cada 2 ciclos
- Acelerar o clock geral... resolve?
  - Acelera todos os estágios, mas...
  - Se estágio lento: 2 ciclos de clock...
  - Continua uma instrução a cada 2 ciclos!
  - **Periféricos e memória podem deixar de funcionar!**
- O que fazer, então?

# Superpipeline

- Permitir um **clock interno** diferente (+ rápido)
  - Execução mais rápida dos estágios mais lentos
  - Exemplo: clock interno = 2x clock externo...
    - Agora estágio lento é cumprido em 2 ciclos **internos**
    - Mas cada ciclo externo equivale a dois internos...
- Superpipeline: pipeline com clock interno da CPU maior que o clock externo

# Superpipeline

- Exemplo Superpipeline
- Exemplo:

$\phi_e$	$\phi_i$	t	E1	<b>E2</b>	E3	E4
1	1	0,00	I1	-	-	-
1	2	0,25	I2	I1	-	-
2	3	0,50	I2	I1	-	-
2	4	0,75	I3	I2	I1	-
3	5	2,00	I3	I2	-	I1
3	6	2,50	I4	I3	I2	-
4	7	3,00	I4	I3	-	I2

E1/E3/E4 = 1 ciclo  
**E2 = 2 ciclos**  
internos

1 Instrução  
por Ciclo  
externo



# **ARQUITETURA SUPERESCALAR**

# Paralelismo em Nível de Instruções

- Arquitetura pipeline: domínio inicial da Intel
- AMD vinha “colada”... Intel tinha que inovar
  - *“Executar várias instruções ao mesmo tempo!”*
  - Múltiplos pipelines: instruções distribuídas
  - Ao invés de acelerar uma fábrica...
    - Vamos ter várias fábricas!
- Sempre posso executar instruções simultaneamente?
- **Instruções Independentes**
- O que são?

# Instruções Independentes

- Exemplo...
- LD           A, 17                             ; Carrega A com 17
- ADD          A, 20                            ; Soma 20 em A
- Quanto vale A?
- Se inverter as instruções o resultado é o mesmo?
- ADD          A, 20                            ; Soma 20 em A
- LD            A, 17                            ; Carrega A com 17
- Quanto vale A?
- O resultados dependem da ordem das instruções



# Instruções Independentes

- Vejamos um outro programa
- LD           A, 17           ; Carrega A com 17
- ADD         A, 20           ; Soma 20 em A
- LD           C, A           ; Guarda resultado (37) em C
- LD           A, 30           ; Carrega A com 30
- ADD         A, 10           ; Soma 10 em A
- LD           D, A           ; Guarda resultado (40) em D
- Essas linhas são dependentes?

# Instruções Independentes

- Vejamos um outro programa
- LD            A, 17            ; Carrega A com 17
- ADD          A, 20            ; Soma 20 em A
- LD            C, A             ; Guarda resultado (37) em C
- LD            A, 30            ; Carrega A com 30
- ADD          A, 10            ; Soma 10 em A
- LD            D, A             ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- Vamos trocar os registradores...
  - Trocar A por B na parte “azul”

# Instruções Independentes

- Vejamos um outro programa
- LD            A, 17            ; Carrega A com 17
- ADD          A, 20            ; Soma 20 em A
- LD            C, A                ; Guarda resultado (37) em C
- LD            B, 30            ; Carrega B com 30
- ADD          B, 10            ; Soma 10 em B
- LD            D, B               ; Guarda resultado (40) em D
- Essas linhas são dependentes?
- E vamos reorganizar as linhas...

# Instruções Independentes

- Vejamos um outro programa
  - LD            A, 17            ; Carrega A com 17
  - LD            B, 30            ; Carrega B com 30
  - ADD          A, 20            ; Soma 20 em A
  - ADD          B, 10            ; Soma 10 em B
  - LD            C, A            ; Guarda resultado (37) em C
  - LD            D, B            ; Guarda resultado (40) em D
- 
- Ainda são dependentes?

# Instruções Independentes

- Vejamos um outro programa

- LD A, 17

- LD B, 30

- ADD A, 20

- ADD B, 10

- LD C, A

- LD D, B

**Otimizador de  
código para  
processadores  
superescalares**

- Ainda são dependentes?

# Instruções Independentes

- Vejamos um outro programa

**Superescalar:**

**Excelente para**

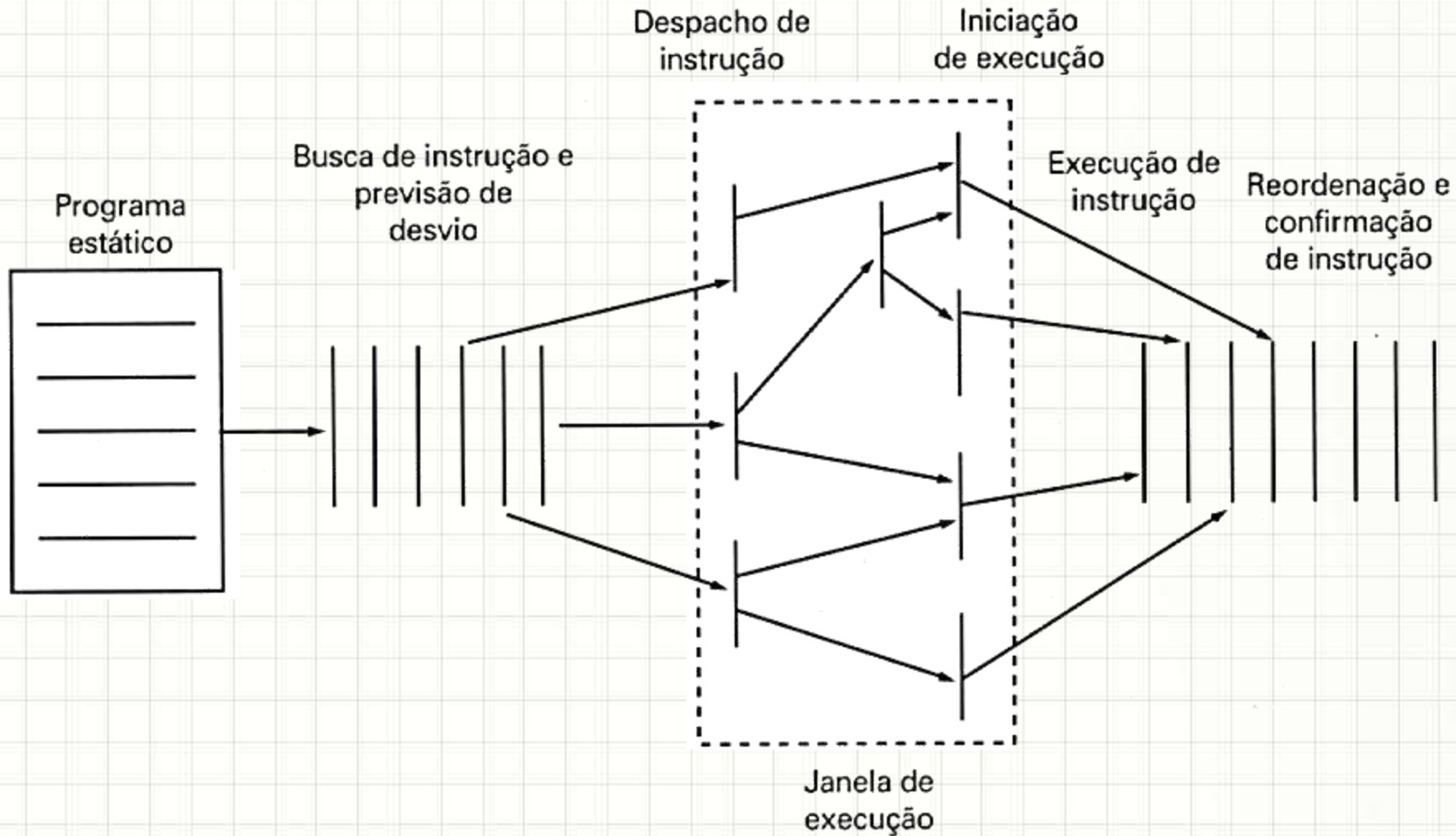
**processamento de imagens:**

**Pixels são Independentes!**

- LD                    D, B                    ; Guarda resultado (40) em D

- Ainda são dependentes?

# Lógica Superescalar

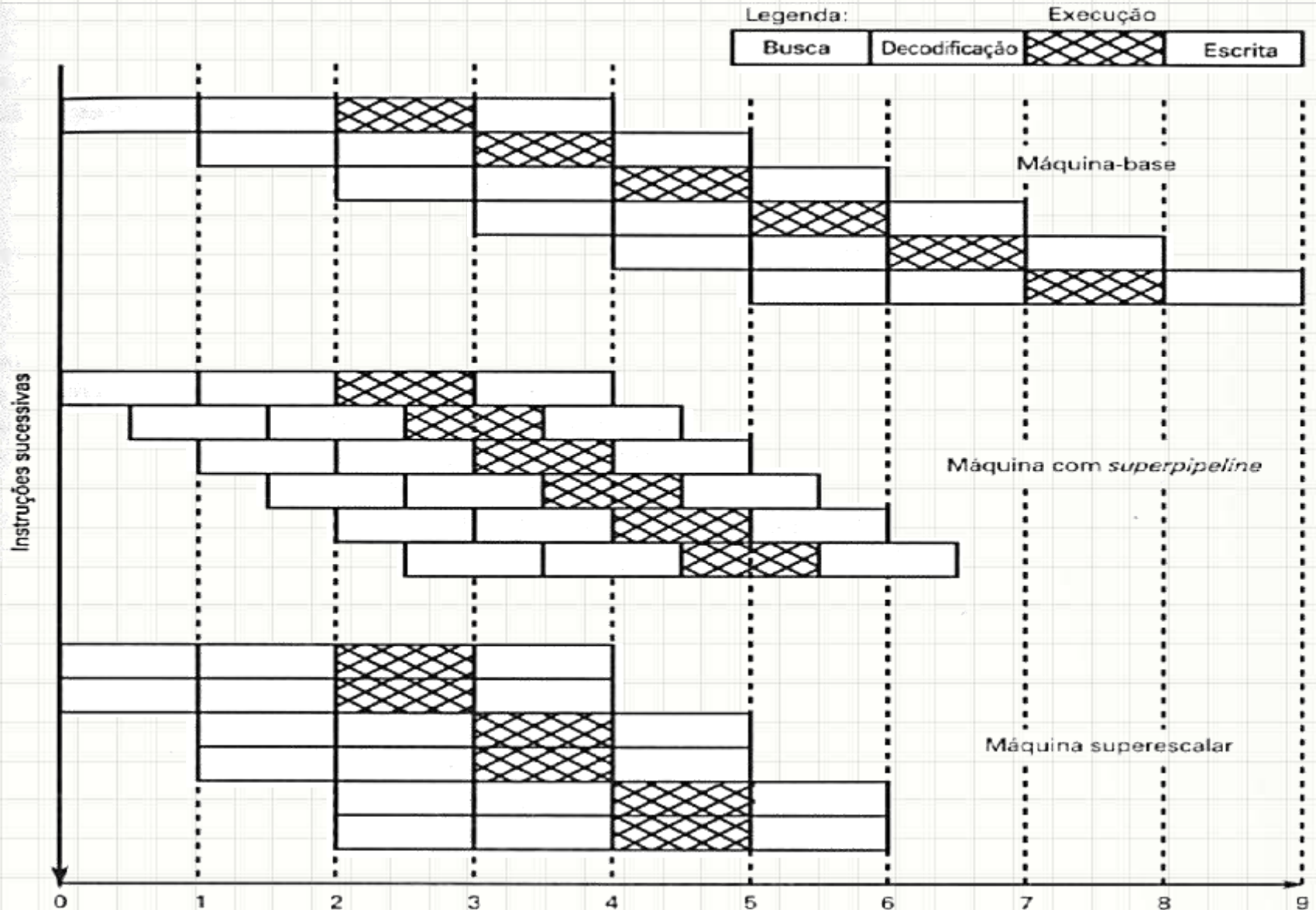




**COMPARATIVO**



# (Super)Pipeline x Superescalar





# FUNDAMENTOS DO MULTIPROCESSAMENTO

# Tipos de Multiprocessamento

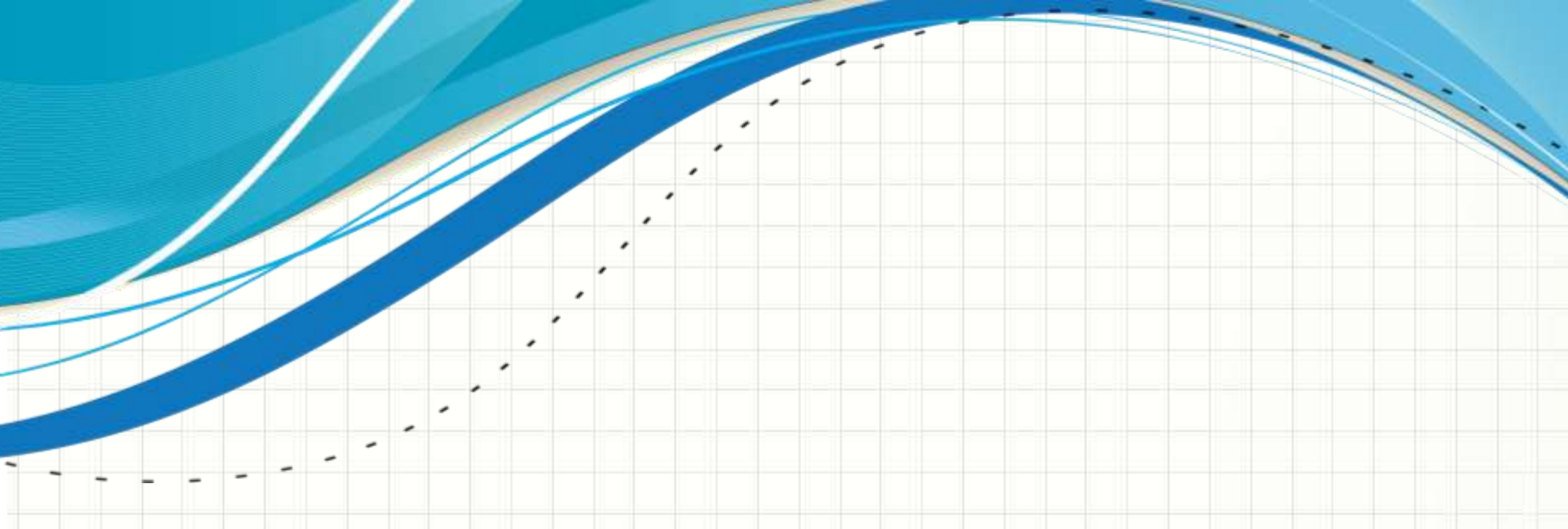
- Superescalar
  - Paralelismo em nível de instruções
- Múltiplos programas...
  - Paralelismo em nível de processos?
- Os tipos mais conhecidos são
  - **SMP**: Symetric MultiProcessing
  - **Clusters**
  - **NUMA**: Non-Uniform Memory Access
- Detalhes na próxima aula...!

# Tipos de Multiprocessamento



# Tipos de Multiprocessamento

- Classificação do Processamento:
  - **SISD**: Single Instruction, Single Data
    - Pipelines e Processadores Escalares
  - **SIMD**: Single Instruction, Multiple Data
    - Processamento Vetorial e Matricial
  - **MISD**: Multiple Instruction, Single Data
    - Teórico!
  - **MIMD**: Multiple Instruction, Multiple Data
    - SMP e Clusters
- Processamento vetorial?
  - Supercomputadores: imagens, simulação de partículas etc.



# **COMPUTADOR COMPLETO**

# Conceito

- O que é um computador completo?
  - *Whole Computer*
- Computador com processamento, unidades de entrada e saída e memória próprias
- **SMP** e **NUMA**: um único computador, com várias unidades de processamento
- **Clusters**: vários computadores completos interligados por rede
  - Cada computador completo: **nó** de processamento

# Consequência

- Um cluster com  $N$  nós...
  - **Pode** ser dividido em  $N$  computadores independentes
- Um sistema SMP/NUMA com  $N$  processadores
  - **NÃO pode** ser dividido em  $N$  computadores independentes





**CONCLUSÕES**

# Resumo

- Multiprocessamento: busca aumento do desempenho de processamento
  - Em nível de instruções x Em nível de processos
- Multiprocessamento em nível de instruções
  - Muitos cuidados são exigidos com o programa
- Multiprocessamento em nível de processos
  - Muitos cuidados são exigidos com o hardware
- **TAREFA**
  - **Lista de Exercícios 2!**

# Próxima Aula



- Como funcionam o multiprocessamento em nível de processos?
  - Qual a diferença entre SMP/NUMA e Clusters?
  - Qual deles é melhor?



**PERGUNTAS?**



**BOM DESCANSO  
A TODOS!**