

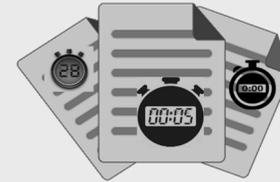


# PCS5761 Especificação de Sistemas de Tempo Real



Prof. Dr. Jorge Rady  
de Almeida Jr.

1

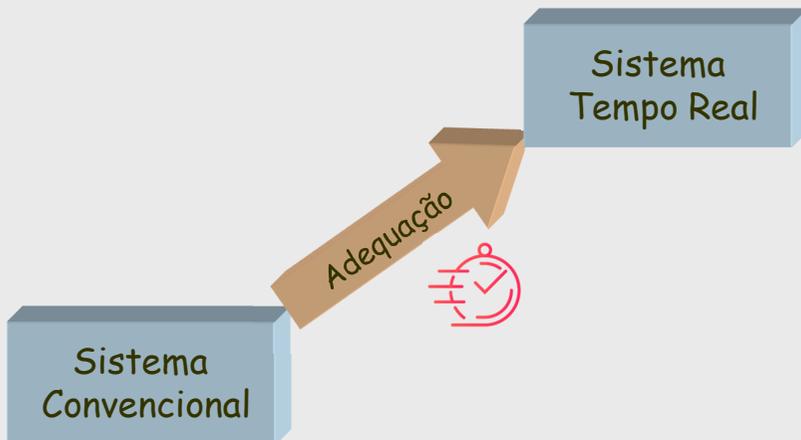


# Hardware para Tempo Real



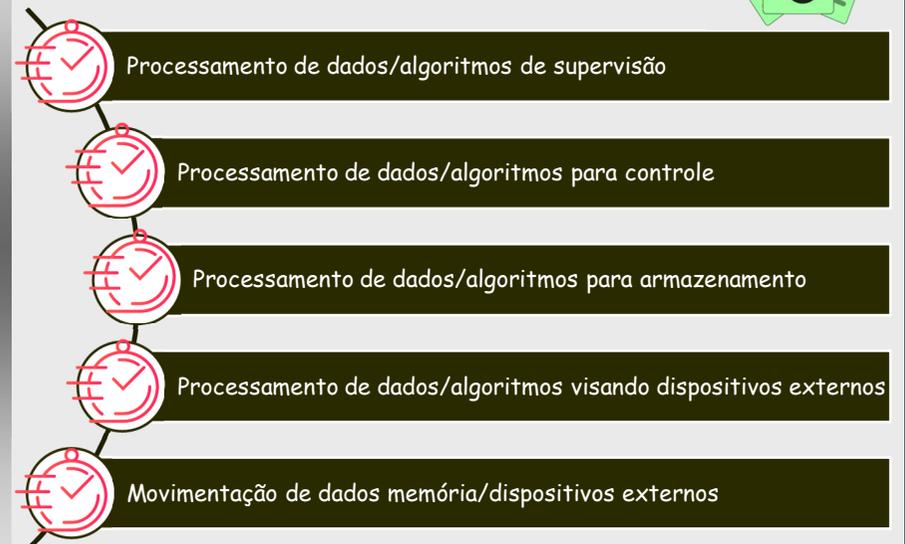
2

# HW para Tempo Real



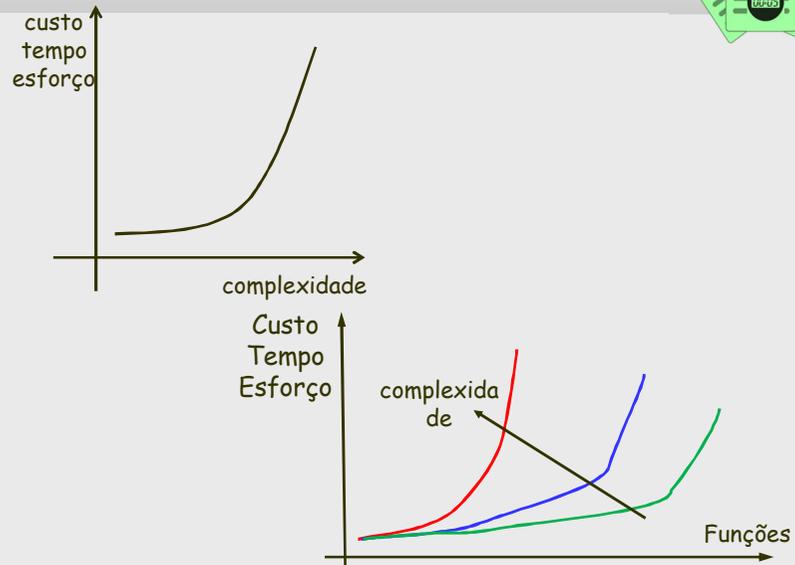
3

# Funções do Computador



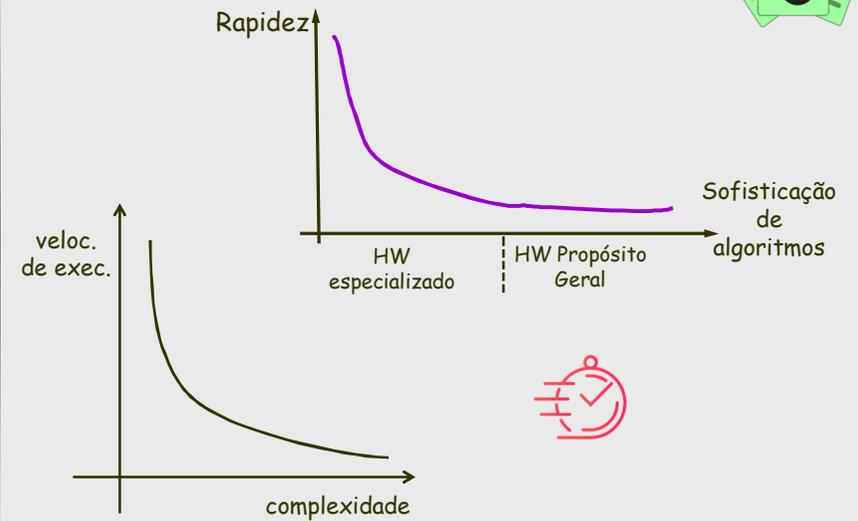
4

## Projeto de HW



5

## HW para Tempo Real



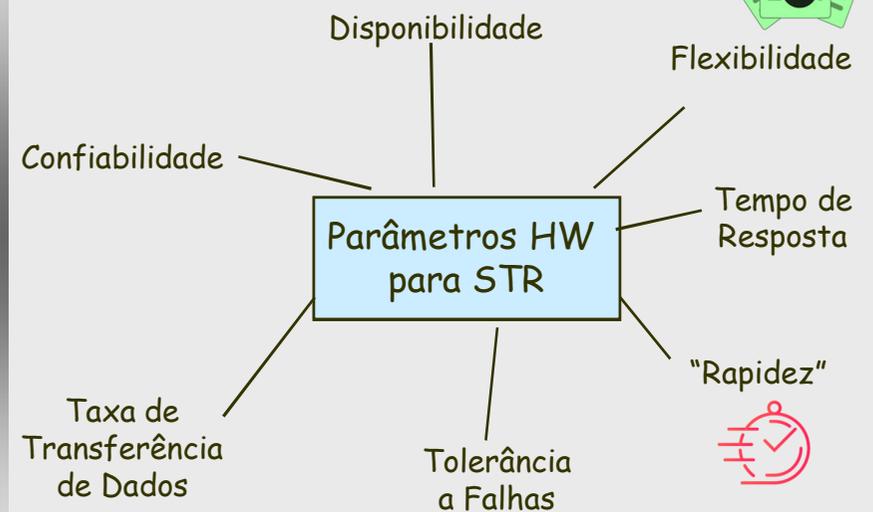
6

*HW para Tempo Real:  
alguns parâmetros importantes*



7

## Parâmetros de HW



8



# Tempos de Computação Importantes



Obter dados / sinais de entrada

Processar os dados/algoritmos

Armazenar dados processados

Acionar SW adequado de tratamento

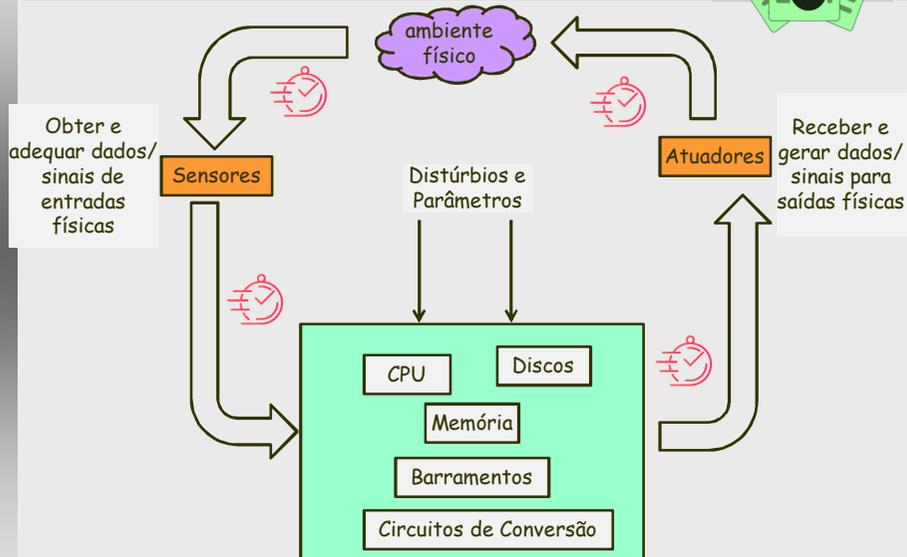
Gerar dados / resultados de saída

Armazenar dados / resultados de saída

# Dispositivos de HW



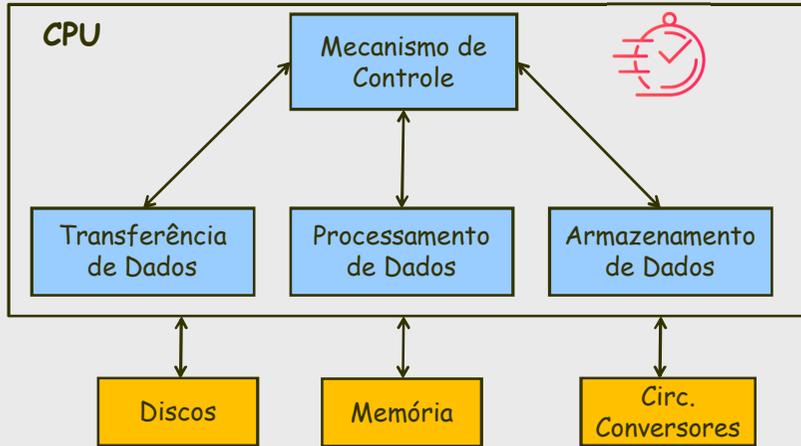
# HW para Tempo Real



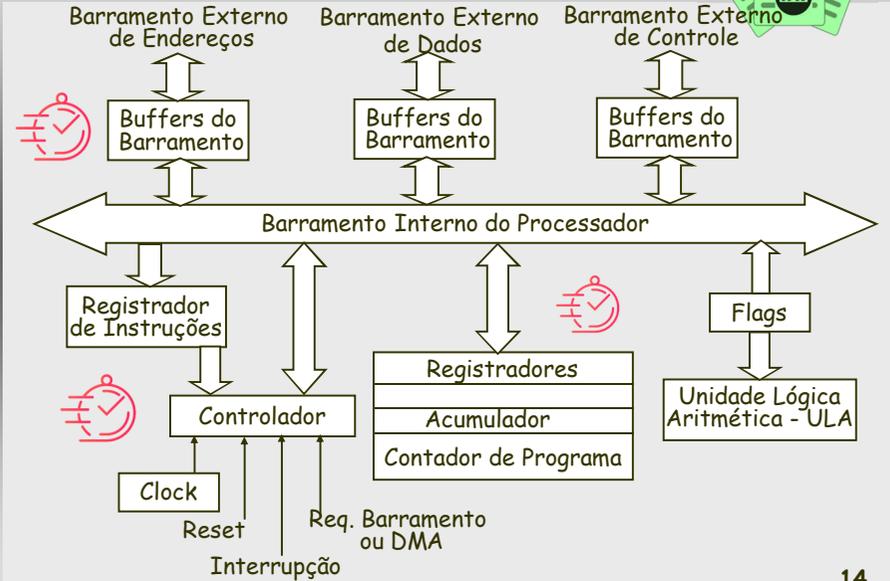
# Tempo Real Hardware Básico



# HW para Tempo Real



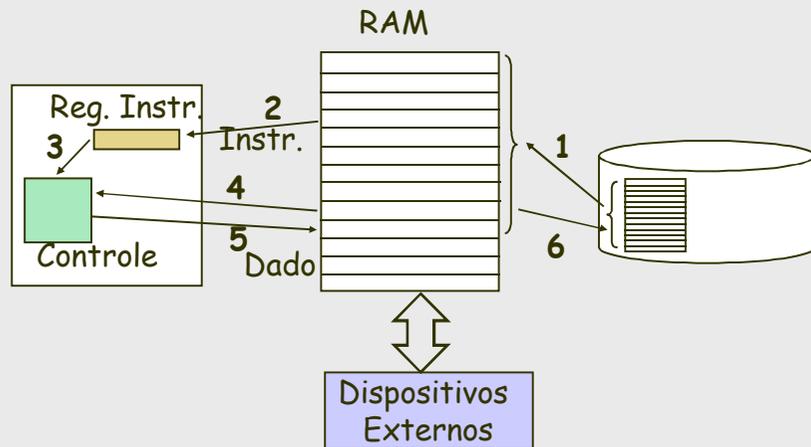
# Computador - Elementos Básicos



# Computador - Operações Básicas



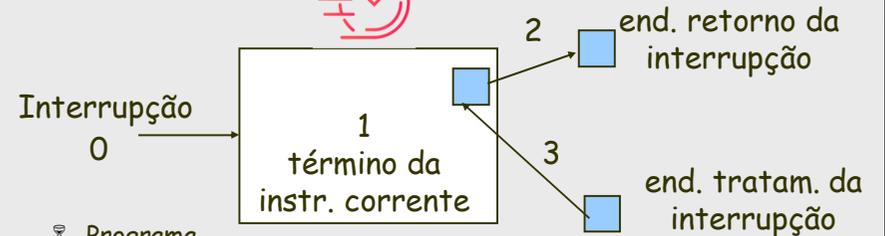
## ⌚ Ciclo Fetch-Executa



# Computador - Operações Básicas



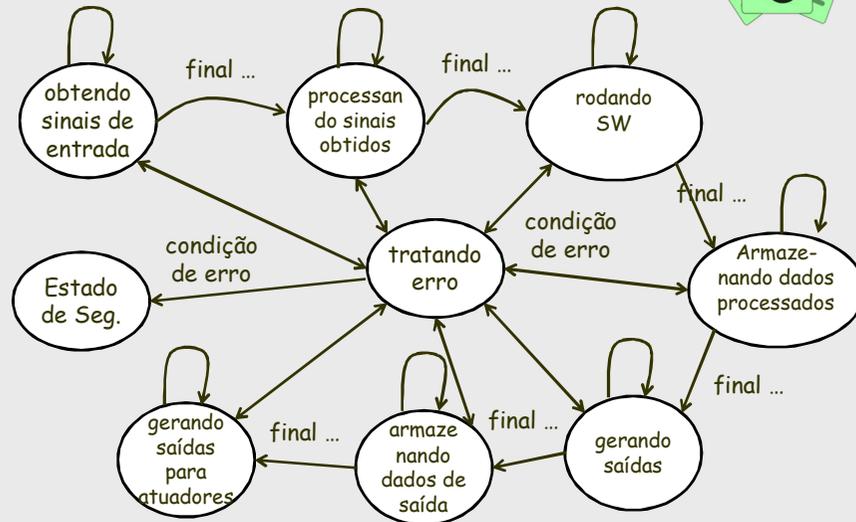
## ⌚ Tratamento de Interrupções



- ⌚ Programa
  - ⌚ Overflow, divisão por zero, ...
- ⌚ Temporização
- ⌚ Controlador de E/S
- ⌚ Falha de hardware
- ⌚ Alarmes



## Diagrama de Estados



17

## Organização e Arquitetura

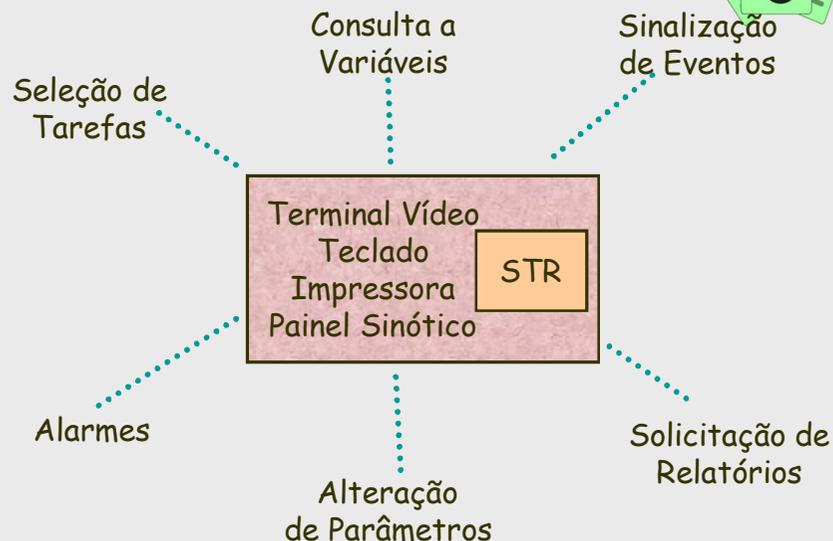


- ⌚ Aumento da velocidade dos processadores
- ⌚ Diminuição do tamanho físico
- ⌚ Redução no tempo de propagação de sinais
- ⌚ Incremento do uso de memória cache
- ⌚ Paralelismo
- ⌚ "Programação em Hardware" X Programação em Software



18

## Interface Homem-Máquina



19

## HW para Tempo Real



- ⌚ Arquiteturas com foco em falhas - redundância
- ⌚ Tipos e características dos componentes
- ⌚ Tipos e arranjos de memórias
- ⌚ Conjunto de instruções
- ⌚ Tipos de endereçamento
- ⌚ Determinação do desempenho
- ⌚ Objetivos: quais métricas empregar
- ⌚ Quais parâmetros medir/variar
- ⌚ Qual técnica de avaliação utilizar: Modelo Analítico, Simulação, Protótipo, Sistema Final



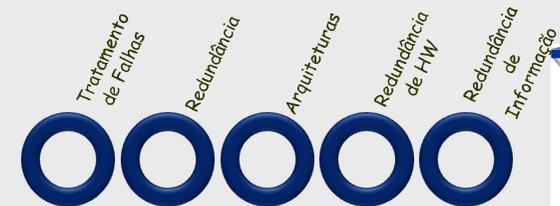
20

Importância da especificação de requisitos para o projeto de HW



21

## HW Tempo Real Redundâncias

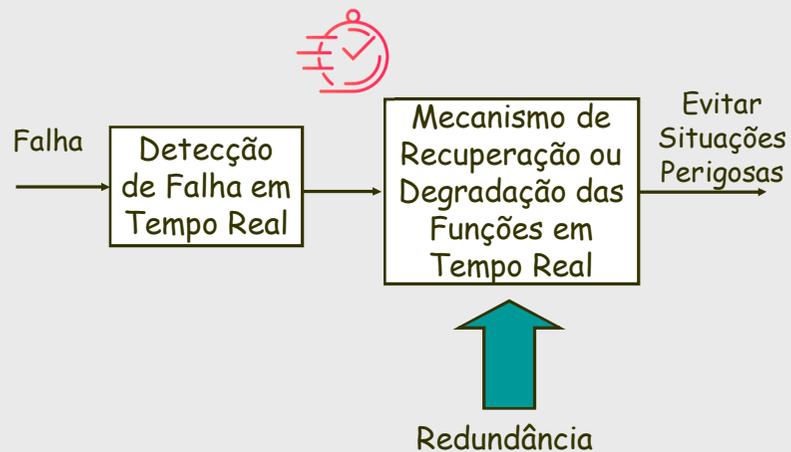


22

## Computadores para Tempo Real



⌚ Na ocorrência de uma falha



23

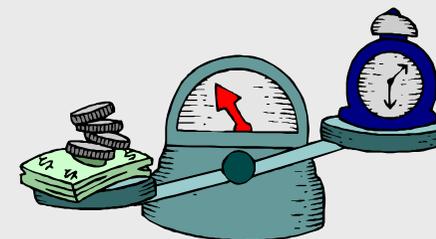
## Computadores para Tempo Real



⌚ Prevenção de falhas no HW

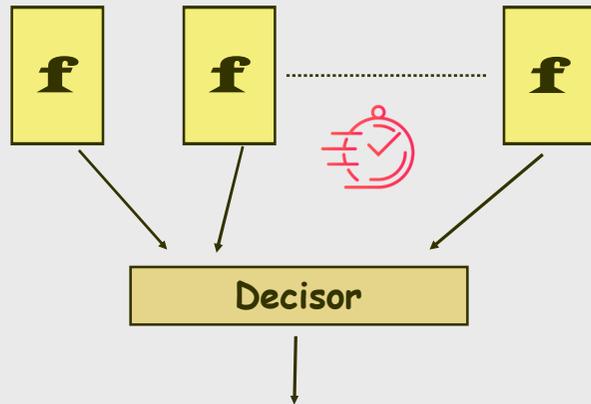


- ⌚ Componentes mais confiáveis
- ⌚ Técnicas de projeto
- ⌚ Documentação de desenvolvimento / implementação → Manutenção



24

## Redundância de Hardware

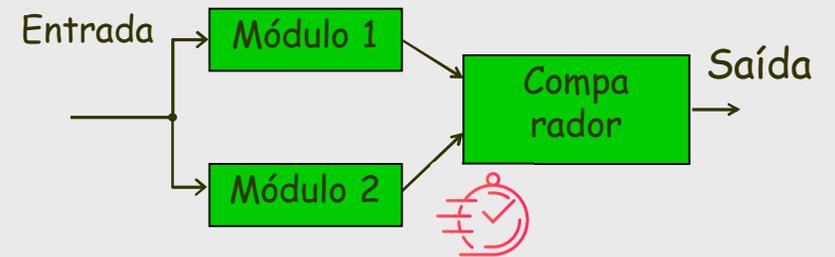


25

## Redundância de Hardware



- ⌚ Redundância Estática - Mascaramento de Falhas
- ⌚ Dois módulos com comparação

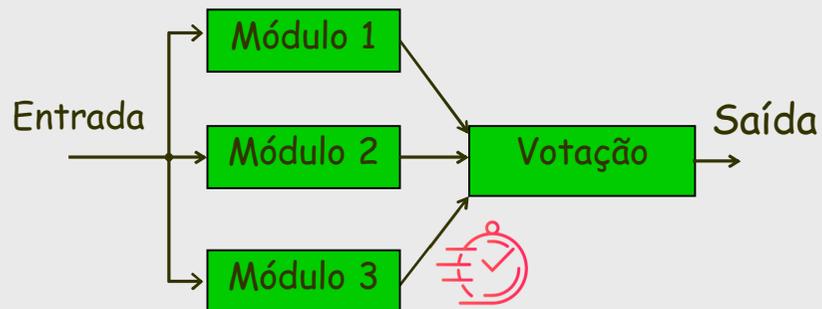


26

## Redundância de Hardware



- ⌚ Redundância Estática
- ⌚ Três módulos com votação

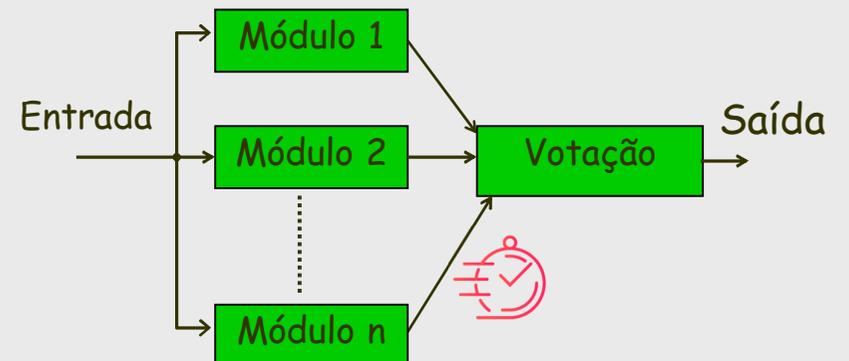


27

## Redundância de Hardware



- ⌚ Redundância Estática
- ⌚ Redundância N-Modular



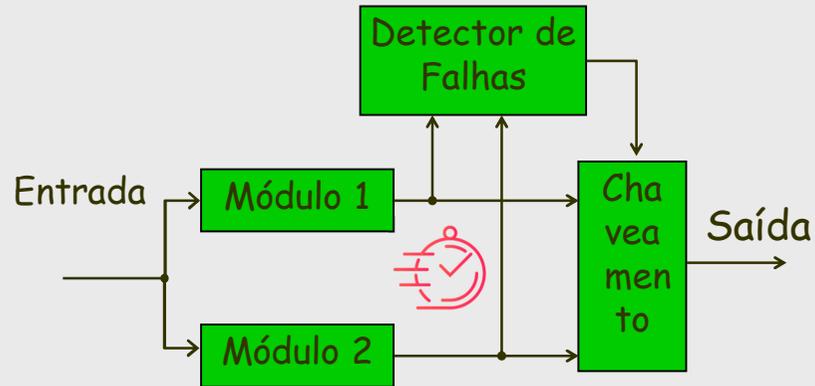
28

## Redundância de Hardware



⌚ Redundância Dinâmica - Detecção/Reconfiguração

⌚ Módulo em Espera



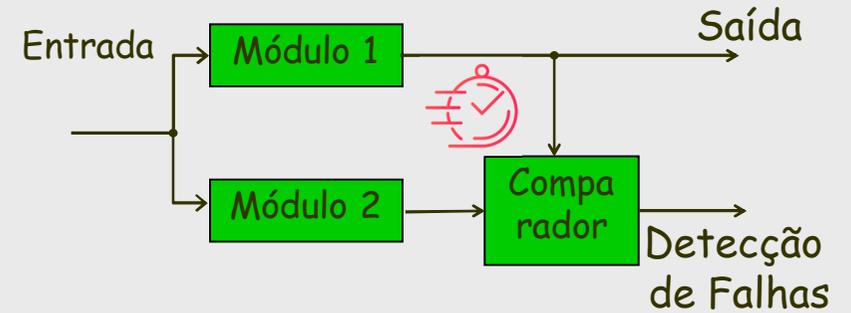
29

## Redundância de Hardware



⌚ Redundância Dinâmica

⌚ Self Checking

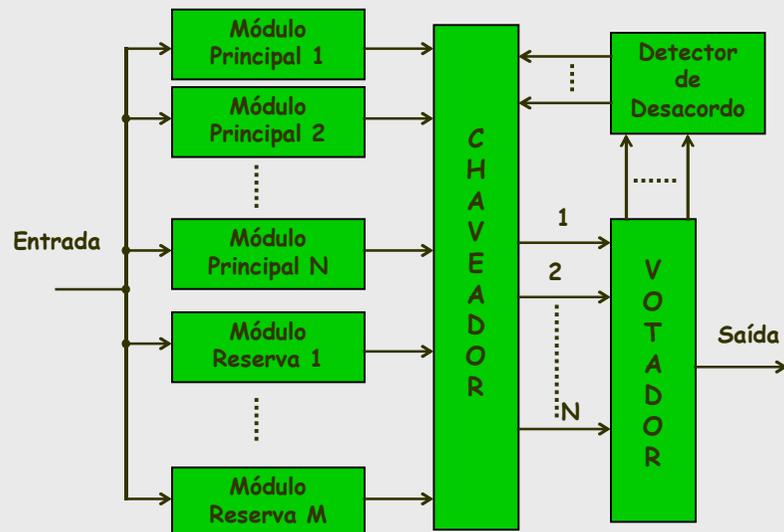


30

## Redundância de Hardware



⌚ Redundância Híbrida



31

## Redundância de Informação



⌚ Códigos de paridade

⌚ Códigos m de n (n bits no total, com m bits = 1)

⌚ Códigos duplicados

⌚ Checksum



32

Quais são aspectos positivos da redundância de hardware?  
E aspectos negativos?

33

# Tempo Real Barramentos

34

## Sistema de Tempo Real Processado

```

    graph LR
      S[Sensores] --> C1[Conversores]
      C1 --> CC1[Concentradores]
      CC1 --> P[Processador]
      P --> CC2[Concentradores]
      CC2 --> C2[Conversores]
      C2 --> A[Atuadores]
      IH[Interface Humana] --> P
  
```

35

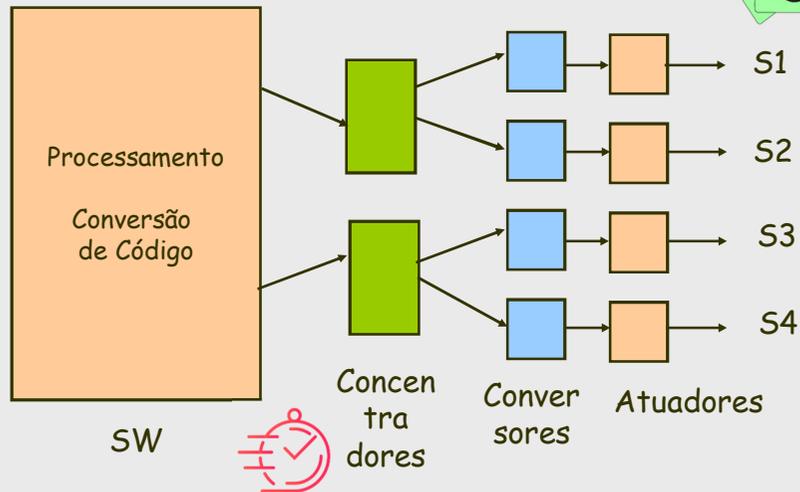
## Aquisição de Dados

```

    graph LR
      E1[E1] --> S1[Sensores / Medidores]
      E2[E2] --> S2[Sensores / Medidores]
      E3[E3] --> S3[Sensores / Medidores]
      E4[E4] --> S4[Sensores / Medidores]
      S1 --> C1[Conversores]
      S2 --> C2[Conversores]
      S3 --> C3[Conversores]
      S4 --> C4[Conversores]
      C1 --> CC1[Concentradores]
      C2 --> CC1
      C3 --> CC2[Concentradores]
      C4 --> CC2
      CC1 --> SW[SW]
      CC2 --> SW
      subgraph SW [SW]
        TR[Tratamento de Redundâncias]
        CC[Conversão de Código]
        SI[Separação de Informações]
        CA[Classificação / Armazenamento Dados]
      end
      SW --> Out[ ]
  
```

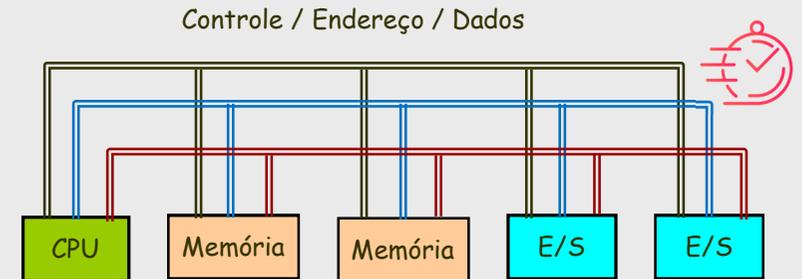
36

## Atuação



37

## Barramentos



38

## Barramento Único



⌚ Muitos dispositivos conectados

⌚ Atrasos de propagação

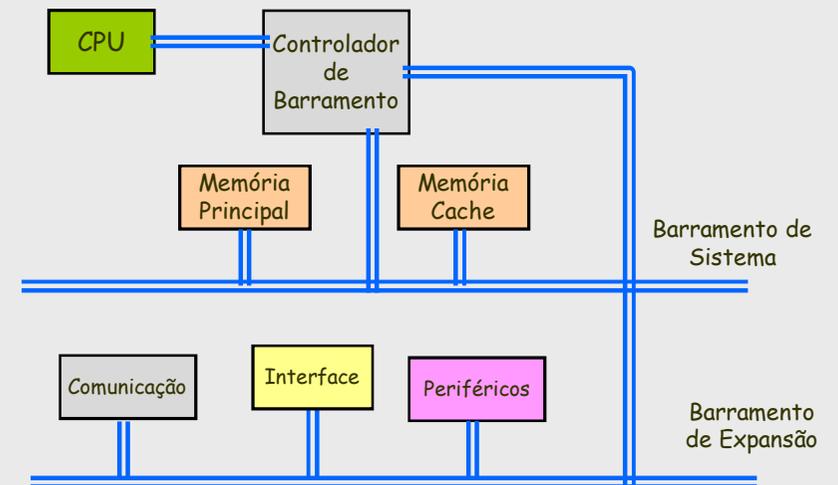
⌚ Caminhos longos a serem percorridos

⌚ Demanda de transferência de dados próxima da capacidade do barramento

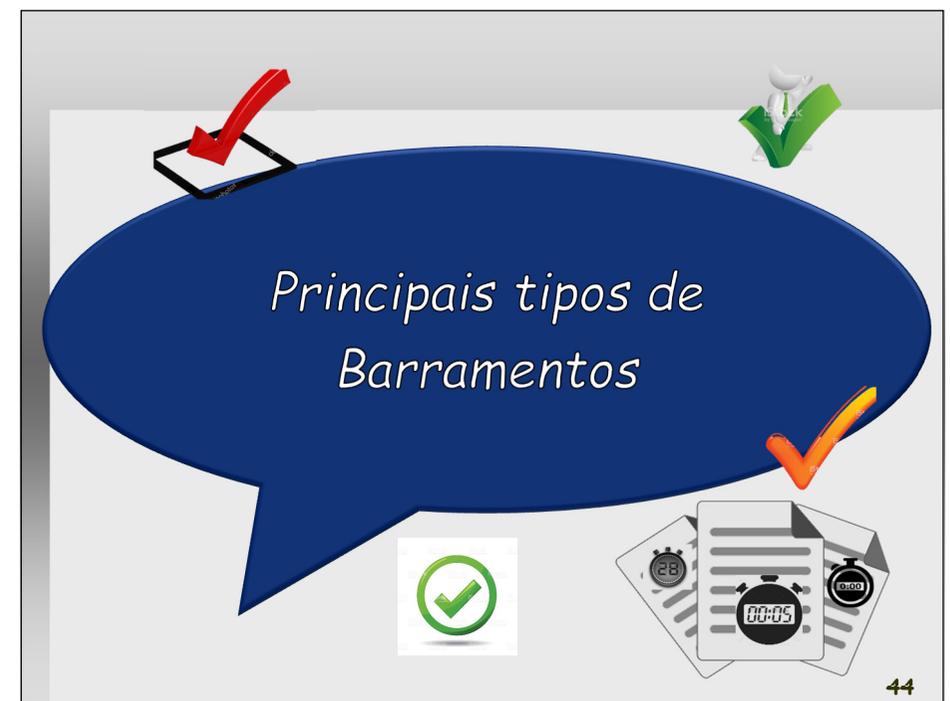
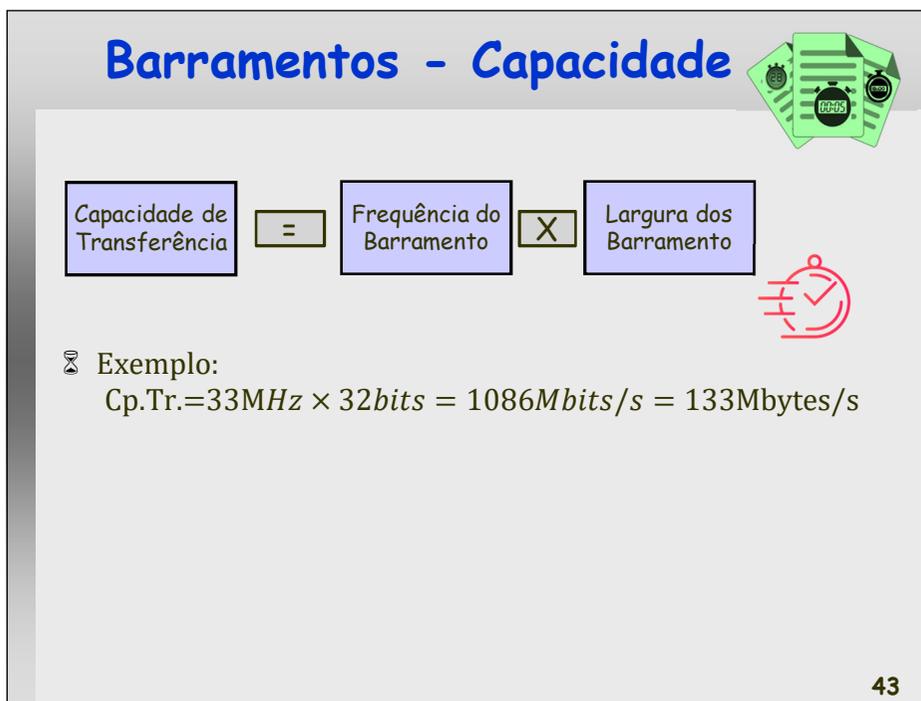
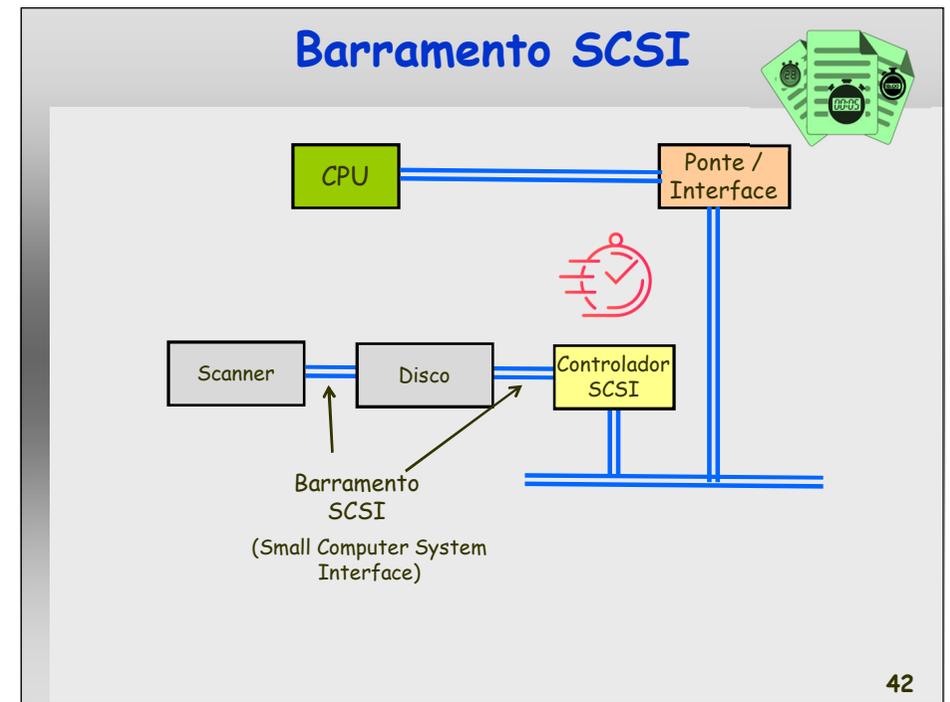
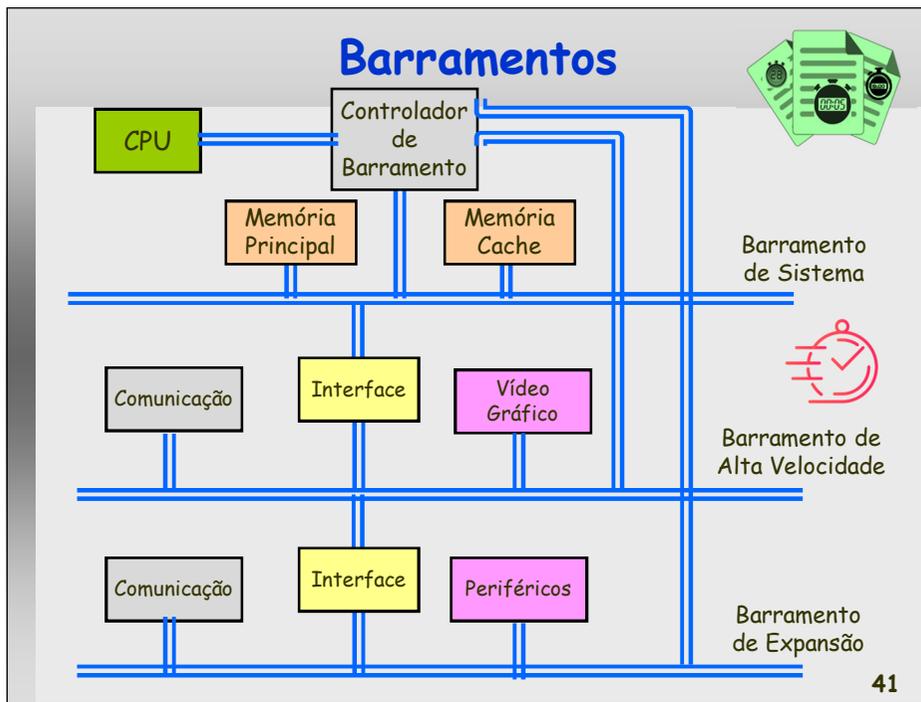


39

## Barramentos



40





## Barramentos



### ⌚ ISA: Industry Standard Architecture

- ⌚ Padrão mais antigo
- ⌚ 8 ou 16 bits de largura e clock de 8,33 MHz
- ⌚ 4 a 10 MB/s

### ⌚ PCI: Peripheral Component Interconnect

- ⌚ 1990 - Intel
- ⌚ 32 bits de largura e clock de 33 MHz
- ⌚ 132 MB/s

45



## Barramentos



### ⌚ PCI-X: Peripheral Component Interconnect Extended

- ⌚ 64 bits de largura
- ⌚ 100 a 533 MHz
- ⌚ 1064 MB/s

### ⌚ AGP: Accelerated Graphics Port

- ⌚ Voltado a placas de vídeo
- ⌚ 32 bits e clock de 66 MHz
- ⌚ 532 MB/s

46

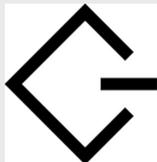


## Barramentos



### ⌚ SCSI: Small Computer System Interface

- ⌚ Comunicação entre dispositivos (HDs, scanners, impressoras, fitas, ...) - Identificadores
- ⌚ SCSI 1 - 1986, (8 bits, 5MB/s) 7 dispositivos
- ⌚ Wide SCSI (SCSI 2) (16 bits, 10 MB/s) 7 dispositivos
- ⌚ Ultra SCSI (16 bits, 80 MB/s) 15 dispositivos
- ⌚ Ultra 160 SCSI (16 bits, 160 MB/s) 15 dispositivos



47



## Barramentos



### ⌚ PCI Express: Peripheral Component Interconnect Express

- ⌚ 2004 - Intel
- ⌚ Substituto de PCI e AGP
- ⌚ 4 GB/s

48



## Barramentos



### ⌚ USB: Universal Serial Bus

- ⊕ USB 1.0 em 1996 // 2.0 em 2000// 3.0 em 2008 // 3.1 em 2013
- ⊕ Alimentação "direta"
- ⊕ Padrão de conexão
- ⊕ Plug and Play
- ⊕ Hot Swappable
- ⊕ Vários dispositivos em uma única porta
- ⊕ Cabos de até 5m
- ⊕ USB 2.0: até 60 MB/s
- ⊕ USB 3.0: até 4,8 GB/s

49



## Tempo Real Memória



50

## Memória



Barramento

Velocidade de Acesso à Memória



Tecnologia

Algoritmos de Acesso

51

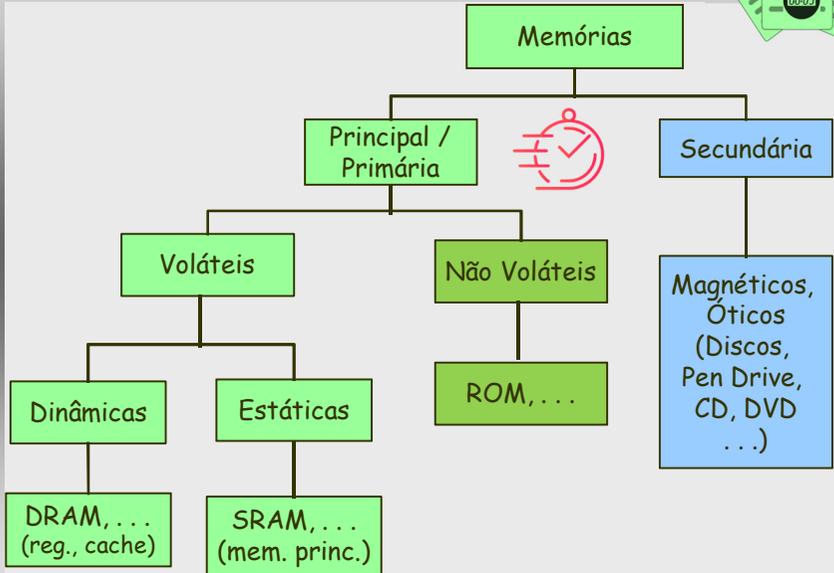
## Hierarquia de Memórias



Placa	Registradores	
	Cache	
	Principal	
Fora da Placa	Disco rígido	
	CD/DVD	
	Flash	
Off-line	Fitas	
	...	

52

# Hierarquia de Memórias



# RAM Dinâmica x Estática



## DRAM - RAM Dinâmica

- ⌚ Carga de capacitores
- ⌚ Refresh
- ⌚ Implementação mais simples
- ⌚ Menos \$
- ⌚ Mais lenta
- ⌚ Memória Principal

## SRAM - RAM Estática

- ⌚ Semicondutores - chaves eletrônicas
- ⌚ Sem refresh
- ⌚ Implementação mais complexa
- ⌚ Mais \$
- ⌚ Mais rápida
- ⌚ Memória Cache

# RAM - Magneto Resistive RAM



## Magneto Resistive RAM - MRAM

- ⌚ Tecnologia recente
- ⌚ Similar a DRAM
- ⌚ Células magnéticas para armazenamento
- ⌚ 1 transistor e 1 junção túnel magnética (material magnético)
- ⌚ Ainda \$\$\$

## Ferro Electric RAM - FeRAM

- ⌚ Tecnologia recente
- ⌚ 1 transistor e 1 capacitor com dielétrico magnético
- ⌚ Não volátil
- ⌚ Leitura destrutiva
- ⌚ Ainda \$\$\$

# Memória Flash

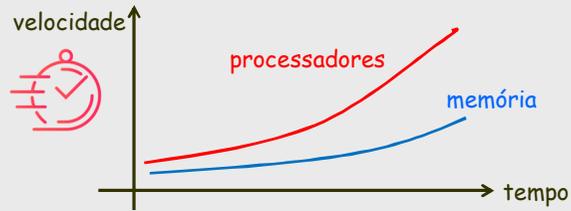


## Flash

- ⌚ Não volátil
- ⌚ Apaga dados em blocos de células
- ⌚ Consumo de energia reduzido

⌚ ---

## Memória

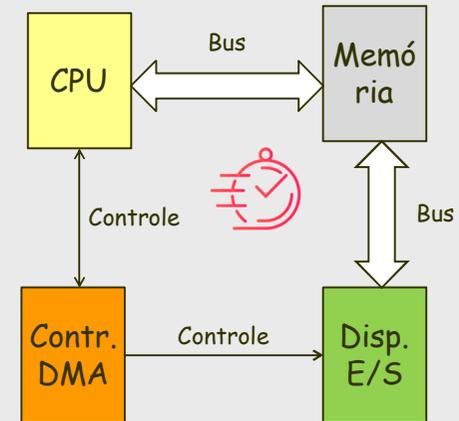


### Soluções

- ⌚ Aumentar largura da palavra/barramento
- ⌚ Memórias cache
- ⌚ Reduzir nº de acessos à memória
- ⌚ Aumentar velocidade do barramento

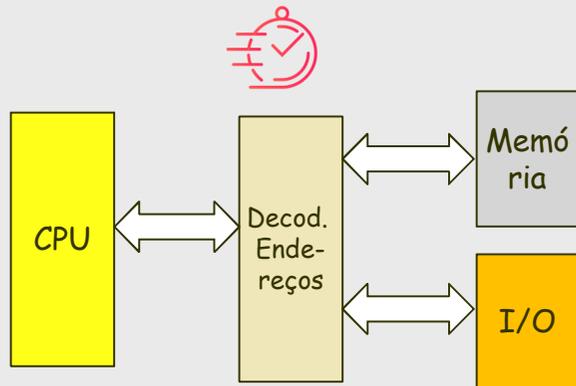
57

## DMA



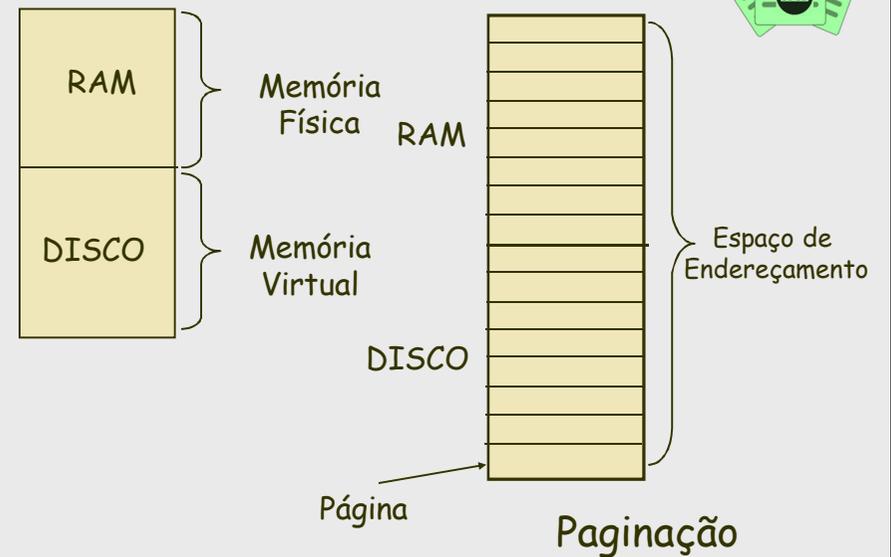
58

## Memória Mapeada em I/O



59

## Memória Virtual



60

# Memória Virtual



Espaço de Endereçamento Virtual

Página Virtual 0
Página Virtual 1
Página Virtual 2
...
Página Virtual 14
Página Virtual 15

Memória Principal

Página Física 0
Página Física 1
Página Física 2
...
Página Física 7

# Memória Virtual



Espaço de Endereçamento Virtual

0	1	Página Virtual 0	2
1	0	Página Virtual 1	0
2	1	Página Virtual 2	0
...	...	...	...
14	1	Página Virtual 14	7
15	0	Página Virtual 15	0

Memória Principal

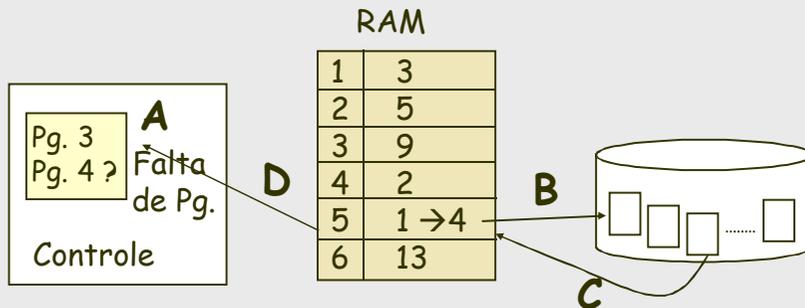
Página 2	0
Página 10	1
Página 0	2
...	...
Página 14	7

Tabela de Paginação

# Memória Virtual



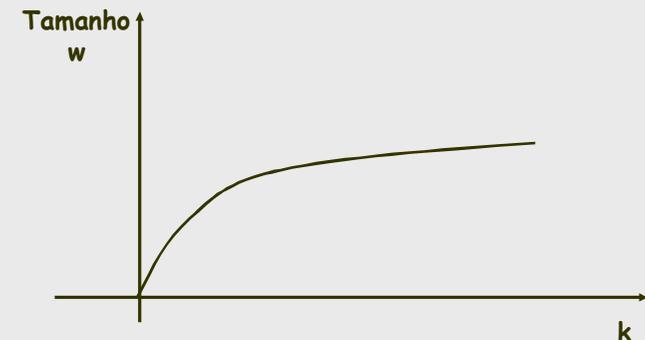
⌚ Paginação por Demanda de Trabalho: execução de uma instrução



# Memória Virtual



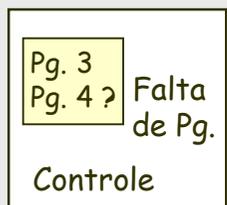
⌚ Conjunto de Trabalho (w): corresponde às páginas referenciadas pelas últimas k referências à memória



# Memória Virtual



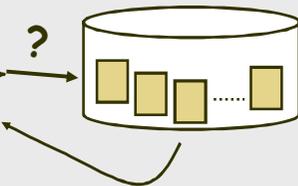
## Política de Substituição de Páginas



CPU

1	3
2	5
3	9
4	2
5	1
6	13

Memória Principal



### Algoritmos

- ⌚ LRU (MRU) - Last (Most) Recently Used
- ⌚ FIFO - First In First Out



# Memória Cache



## Cache Única



## Cache em 3 Níveis



Mais memória ou memória mais rápida?



# Tempo Real Armazenamento



## Discos Rígidos



### ⌚ Medidas de Desempenho

#### ❖ Tempo de Acesso

📄 Seek Time

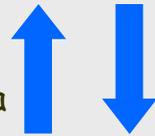
📄 Latência de Rotação

#### ❖ Taxa de Transferência de Dados

🕒 MTTF - Mean Time to Failure



Capacidade/  
Taxa de Transferência



Custo

69

## Meios Físicos de Armazenamento



### ⌚ RAID - Redundant Array of Inexpensive Disks

- ❖ Originalmente: discos mais baratos, em maior quantidade
- ❖ Atualmente: melhorar a disponibilidade
- ❖ Mudança I → Independent
- ❖ Redundância: informações extras → remontar dados perdidos por falhas (ex. espelhamento)
- ❖ Strip: bits divididos em diversos discos

70

## Meios Físicos de Armazenamento



### ⌚ RAID nível 1 - Discos espelhados



### ⌚ RAID nível 4 - Disco exclusivo para paridade



### ⌚ RAID nível 5 - Paridade é distribuída nos discos



### ⌚ RAID nível 6 - Redundância extra além da paridade distribuída



71



De fato, é necessária a redundância de discos?



72

## Comparação HDD x SSD

### HDD

- ▶ Hard Drive Disk
- ▶ Maior consumo
- ▶ Menos \$
- ▶ Maior capacidade
- ▶ 30/40 seg. para SO
- ▶ Partes mecânicas
- ▶ 1 falha/1,5 milhão de horas
- ▶ Taxa típica de 100MB/s
- ▶ Sujeito a interferências eletro-magnéticas

### SSD

- ▶ Solid State Drive
- ▶ Menor consumo
- ▶ Mais \$
- ▶ Menor capacidade
- ▶ 10/15 seg. para SO
- ▶ Sem partes mecânicas
- ▶ 1 falha/2,0 milhão de horas
- ▶ Taxa típica acima de 200 MB/s
- ▶ Livre de interferências eletro-magnéticas

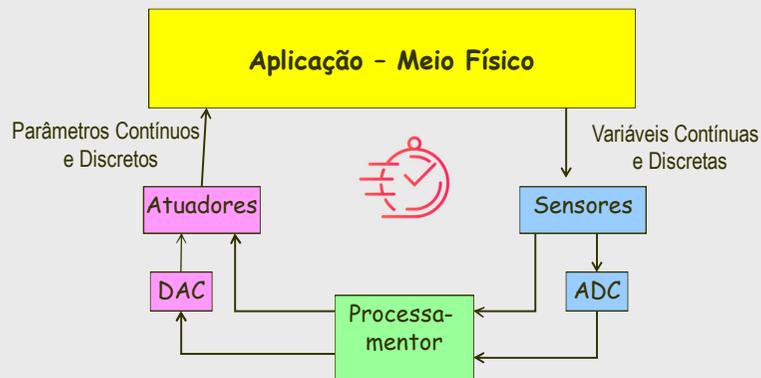
73

## Tempo Real Sensores e Atuadores



74

## Ciclo de Sensores / Atuadores



75

## Sensores

- ⌚ Transdutor que converte um estímulo físico em um sinal elétrico/digital, visando sua medição
- ⌚ Caract. particulares em função de grandezas físicas envolvidas.
  - ⌚ Não há um único tipo de sensor que atenda a todas aplic.
  - ⌚ Fatores: precisão, linearidade, estabilidade à temperatura, ...
- ⌚ Condicionar os sinais: amplificar, deslocar níveis, limitação ou filtragem de frequências, conversão (para pulsos, luz, ...), isolamento elétrico ...



76

## Sensores - Grandezas Medidas



- ⌚ Distância
- ⌚ Intensidade luminosa
- ⌚ Toque
- ⌚ Intensidade sonora
- ⌚ Temperatura
- ⌚ Proximidade
- ⌚ Aceleração
- ⌚ RFID
- ⌚ Código de barras

77



## Atuadores



- ✓ Dispositivo que converte um sinal elétrico/digital de comando em um movimento físico
- ✓ Normalmente necessita de um amplificador para o sinal de comando

78

## Atuadores- Tipos



- ⌚ Movimentação de alguma peça
- ⌚ Acionamento de um relé
- ⌚ Estímulo muscular
- ⌚ ...



- ⌚ Atuadores elétricos
  - ⊕ Motores DC, AC, de passo
- ⌚ Atuadores Hidráulicos
- ⌚ Atuadores Pneumáticos



79

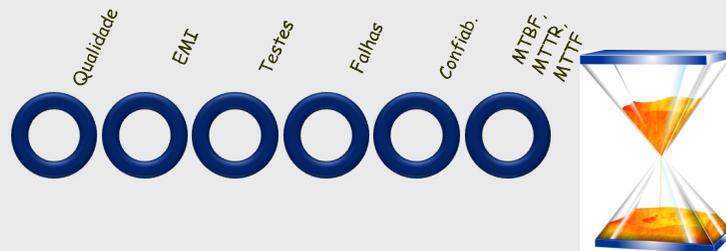


*Sensores / Atuadores  
mais rápidos ou com  
redundância?*



80

# Tempo Real Componentes



81

## Componentes



- ⌚ Qualidade
- ⌚ Interferências
- ⌚ Testes (elétricos, temperatura, umidade, ...)
- ⌚ Tipos de falhas
- ⌚ Confiabilidade, Disponibilidade, Manutenibilidade
- ⌚ Cobertura de falhas



82

## Componentes - Qualidade



- ⌚ Utilizar componentes de melhor qualidade
- ⌚ Evitar componentes "novíssimos"
- ⌚ Curva da "banheira" 
- ⌚ Burn-In
- ⌚ Tipos de falhas
  - ⌚ Desvio de parâmetros/tolerâncias
  - ⌚ Sobre-utilização (correntes, tensões)
  - ⌚ Curto circuito, circuito aberto, ...
  - ⌚ Interferências eletro-magnéticas

83

## Componentes - EMI



- ⌚ EMI - Interferências Eletro-Magnéticas
- ⌚ Chaveamento de cargas indutivas e capacitivas (motores, lâmpadas, relés)
- ⌚ Emissões de outros sistemas
- ⌚ Radiação de alta frequência de circuitos digitais
- ⌚ Rede de energia elétrica

84

## Componentes - Testes



### ⌚ Necessidade de testes

- ⌚ Elétrico / Eletrônicos / Digitais
- ⌚ Temperatura / Umidade
- ⌚ Vibração
- ⌚ EMI
- ⌚ ...

85

## Componentes - Tipos de Falhas



Tipo	Modo de Falha	%
CI Digital	Saída em nível alto ou baixo	80
	Não funcionamento	20
Transistor	Baixo ganho	20
	Curto circuito	20
	Circuito aberto	30
	Fuga de corrente	30
Diodo	Curto circuito	10
	Circuito aberto	20
	Corrente reversa alta	70
Resistor de Filme	Circuito aberto	30
	Mudança de parâmetros	70

86

## Tipos de Falhas



- ⌚ **Falhas Simples:** qualquer falha não deve conduzir o sistema a um estado inseguro, além de ser detectável
- ⌚ **Falhas Múltiplas:** qualquer combinação de falhas de grau 'n' não deve conduzir o sistema a um estado inseguro, além de ser detectável



87

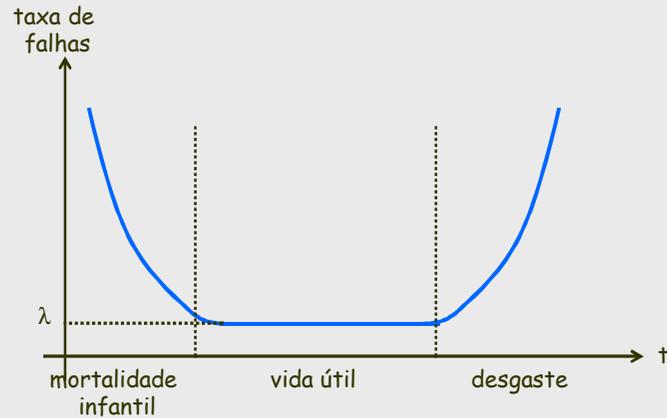
## Taxa de Falhas



- ⌚ Número esperado de falhas de um dispositivo ou sistema em um determinado período de tempo
- ⌚ Obtido pelo teste de um grande número de componentes iguais
- ⌚ Parâmetro usado no cálculo da confiabilidade  
→ símbolo  $\lambda$  → comparação entre dispositivos ou sistemas

88

## Taxa de Falhas



89

## Confiabilidade



- ⌚ Probabilidade do sistema funcionar de acordo com sua especificação em um período de tempo, dado que o mesmo estava funcionando corretamente no instante inicial do período considerado
- ⌚ Considerando um sistema com N componentes idênticos
  - ⌚  $S(t)$  = nº de componentes sobreviventes no tempo t
  - ⌚  $F(t)$  = nº de componentes falhos no tempo t
- ⌚  $R(t) = \frac{S(t)}{N} = \frac{S(t)}{S(t)+F(t)}$
- ⌚ Não Confiabilidade:  $Q(t) = \frac{F(t)}{N} = \frac{F(t)}{S(t)+F(t)}$

90

## Confiabilidade



- ⌚ Para um taxa de defeitos constante  $\lambda$  a confiabilidade  $R(t)$  varia exponencialmente em função do tempo
  - ⌚ Sistema na fase de vida útil: taxa de defeitos constante  $\lambda$
- ⌚  $R(t) = e^{-\lambda t}$ 
  - ⌚ Válida para componentes eletrônicos
  - ⌚ Lei de Falha Exponencial

91

## ⌚ MTTF - Mean Time To Failure

- ⌚ Tempo de operação antes da ocorrência da primeira falha

## ⌚ MTBF - Mean Time Between Failures

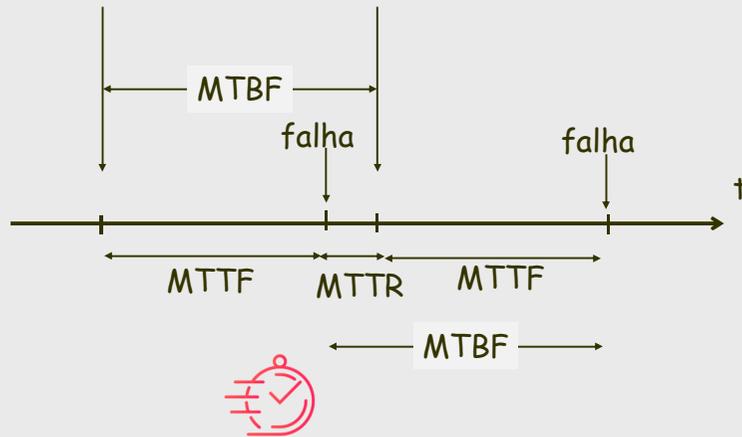
- ⌚ Tempo de operação entre duas falhas

## ⌚ MTTR - Mean Time To Repair

- ⌚ Tempo para se efetuar o reparo/conserto de uma falha

92

## Relação Entre os Tempos



93

## MTTR - Mean Time To Repair



- ⌚ Inclui
  - ⌚ Tempo identificando o erro (80% do total)
  - ⌚ Tempo resolvendo o erro
- ⌚ Difícil de estimar
  - ⌚ Geralmente usa-se injeção de falhas
  - ⌚ Injeta-se uma falha de cada vez e mede-se o tempo
- ⌚ Taxa de reparos  $\mu$ : nº médio de reparos que pode ser realizado, por unidade de tempo
- ⌚  $MTTR = \text{tempo médio requerido para realizar um único reparo: } MTTR = 1/\mu$



94

## Disponibilidade



- ⌚ Probabilidade do sistema estar operacional (funcionando de acordo com suas especificações) em um determinado instante de tempo  $t$ 
  - ⌚  $A(t) = R(t)$  quando reparo tende a zero
  - ⌚  $MTBF = MTTF + MTTR$
  - ⌚  $A(t) = MTTF / (MTTF + MTTR)$
  - ⌚  $A(t)$ : availability



95

## Segurança



- ⌚ Probabilidade de um sistema, em um determinado período de tempo  $\Delta t$ , desempenhar corretamente suas funções ou falhar sem comprometer a integridade de pessoas e do meio ambiente, dado que este sistema estava seguro no instante inicial

96

## Resumo



- ⌚ Com que frequência ocorrem defeitos? **Taxa de Falhas**
- ⌚ Qual o tempo entre um defeito e outro? **MTBF**
- ⌚ Qual o tempo até o primeiro defeito? **MTTF**
- ⌚ Qual o tempo gasto para reparar cada defeito? **MTTR**
- ⌚ Quais as chances do sistema funcionar sem defeitos durante um determinado período de tempo?  
**Confiabilidade**
- ⌚ Quais as chances dos sistema estar funcionando em um determinado instante? **Disponibilidade**

97



*Justifica-se o custo de utilizar componentes de melhor qualidade ?*



98

## Tempo Real Conjunto de Instruções



99

## Conjunto de Instruções



- ⌚ Afetam desempenho em TR
- ⌚ Quanto mais complexo esse conjunto
  - ⌚ Menos instruções são necessárias para gerar uma função
  - ⌚ Cada instrução individualmente levará mais tempo
  - ⌚ Arquiteturas RISC (Reduced Instruction Set Computer) baseiam-se em instruções mais curtas e simples

100



## CISC



- ⌚ Complex Instruction Set Computer
- ⌚ Instruções complexas ocupam ciclos diversos
- ⌚ Quase todas instruções podem referenciar a memória
- ⌚ Microprograma (complexo) para cada instrução
- ⌚ Instruções de formato variável
- ⌚ Vários modos de endereçamento
- ⌚ Pequeno conjunto de registradores

101



## RISC



- ⌚ Instruções simples com um ciclo de clock
- ⌚ Instruções executadas diretamente pelo HW
- ⌚ Instruções com formato fixo
- ⌚ Poucas instruções e modos de endereçamento
- ⌚ Grande conjunto de registradores
- ⌚ Complexidade no compilador

102

## Tipos de Comandos



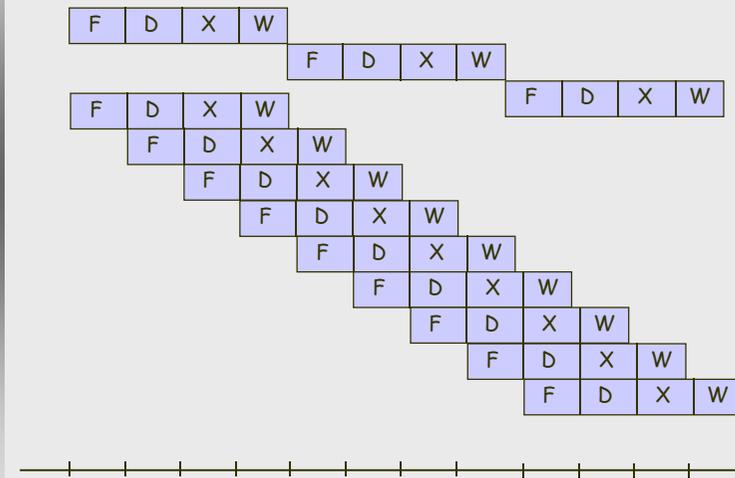
- ⌚ Transferência de dados
  - ⌚ Origem
  - ⌚ Destino
  - ⌚ Quantidade de dados
- ⌚ Lógica
  - ⌚ Bit a bit
  - ⌚ AND, OR, NOT
- ⌚ Conversões
- ⌚ E/S
  - ⌚ Movimentação de dados
- ⌚ Aritméticos
  - ⌚ Adição, ...
  - ⌚ Incremento, Decremento
- ⌚ Desvios
  - ⌚ Condicionais
  - ⌚ Incondicionais

103

## Pipeline



⌚ Fetch, Decode, Execute, Write



104





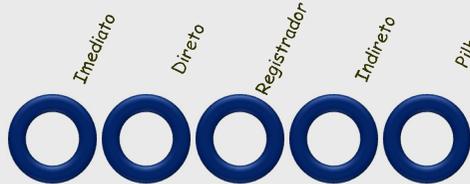

*Selecionar comandos conforme sua velocidade de execução ou com sua funcionalidade?*




105



# Endereçamento




106



## Endereçamento



- ⌚ Instruções de 1 endereço
  - ⌚ Acumulador = Acum. [operação] Registrador
  - ⌚ Acumulador = Acum. [operação] End
- ⌚ Instruções de 2 endereços
  - ⌚ Um endereço fonte e o outro destino
  - ⌚ End. 1 = End. 1 [operação] End. 2
- ⌚ Instruções de 3 endereços
  - ⌚ Dois endereços fonte e um destino
  - ⌚ Destino = End. 1 [operação] End. 2

107

## Modos de Endereçamento




- ⌚ Forma como os campos de endereço da instrução são interpretados para se encontrar um operando
- ⌚ Endereçamento
  - ⌚ Imediato
  - ⌚ Direto
  - ⌚ Registrador
  - ⌚ Indireto
  - ⌚ de Pilha
- ⌚ Influência direta no tempo de execução

108

## Modos de Endereçamento



⌚ Número relativo de ciclos



Modo	Nº Relativo Ciclos
Imediato	2
Direto	3
Registrador	2
Indireto	4
Pilha	2

109

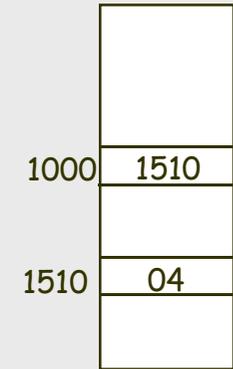
## Endereçamento Imediato



⌚ Instrução fornece o valor do operando

Cód. de Oper.	(*)	04
---------------	-----	----

\* campo que especifica Ender. Imediato



110

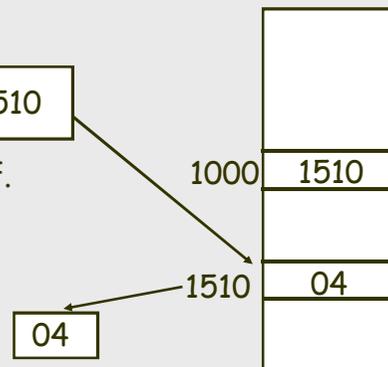
## Endereçamento Direto



⌚ Instrução fornece o endereço da palavra de memória que contém o operando

Cód. de Oper.	(*)	1510
---------------	-----	------

\* campo que especifica Ender. Direto



111

## Endereçamento de Registrador

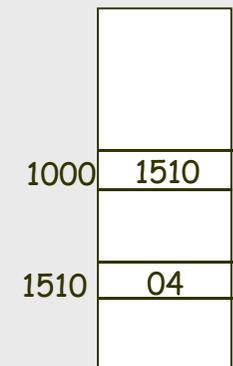


⌚ Instrução fornece o endereço do registrador que contém o operando

Cód. de Oper.	(*)	R1
---------------	-----	----

\* campo que especifica Ender. de Registr.

R1	04
R2	



112

## Endereçamento Indireto

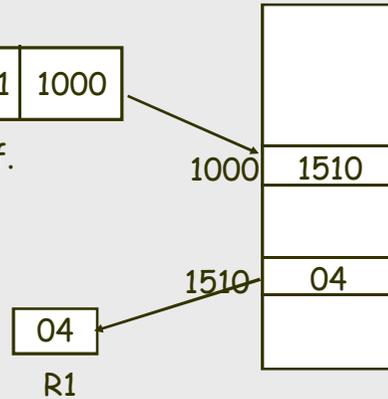


⌚ Campo de endereçamento da instrução especifica qual a palavra de memória ou registrador contém o endereço do operando

Cód. de Oper.	(*)	R1	1000
---------------	-----	----	------

\* campo que especifica Ender. Indireto

Ex.: Carga do reg. R1 de forma indireta a partir da posição de memória de end. 1000



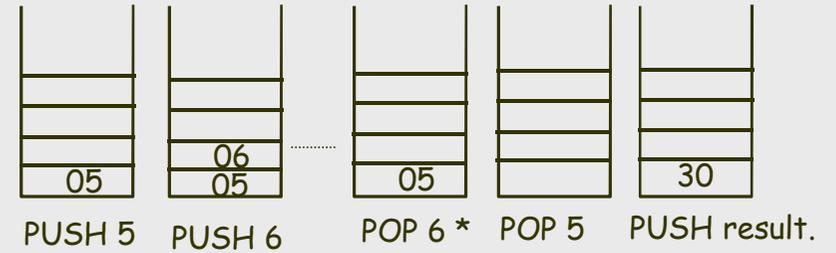
113

## Endereçamento de Pilha



Cód. de Oper.	(*)	05
---------------	-----	----

\* campo que especifica Ender. de Pilha



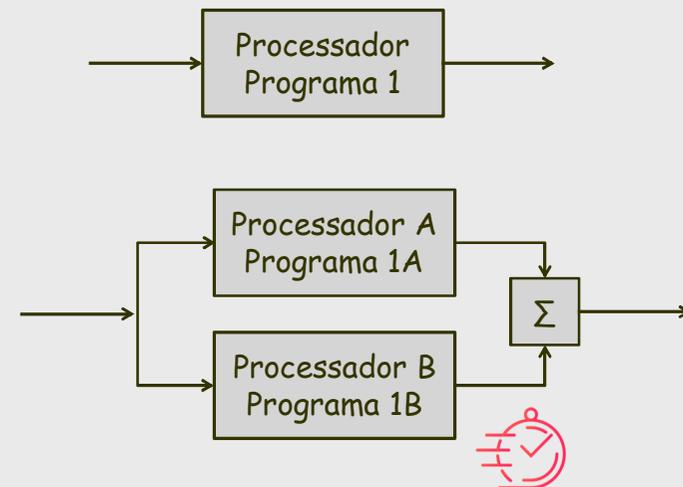
114

## Computação Paralela



115

## Computação Paralela



116

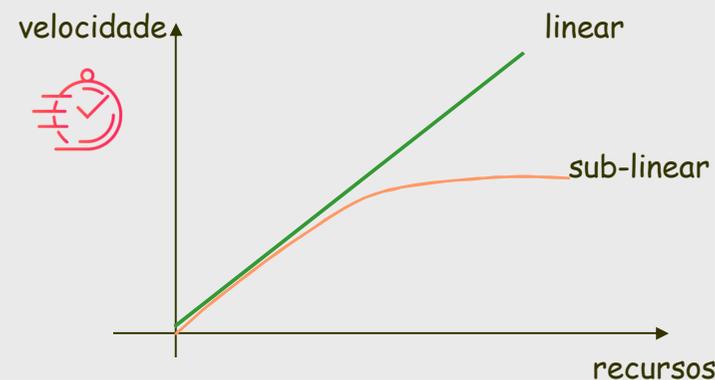
## Computação Paralela



- ⌚ Melhoria no processamento e E/S: múltiplas CPUs/discos em paralelo
- ⌚ Operações simultâneas
- ⌚ Duas medidas principais
  - ⌚ Throughput: nº de tarefas que podem ser completadas em um intervalo de tempo
  - ⌚ Tempo de Resposta: tempo para completar uma tarefa a partir do instante de sua submissão

117

## Computação Paralela



118



## Computação Paralela



- ⌚ Fatores contra
  - ⌚ Custo de processamento de startup de  $n$  de processadores
  - ⌚ Interferência de processos em paralelo → disputa por recursos compartilhados
  - ⌚ Skew → não exatidão na divisão de tarefas pelos processadores

119



## Lei de Amdahl



- ⌚ O speedup se aproxima de zero com o aumento do número de processadores
- ⌚  $N$  = nº de processadores,  $S$  = fração do programa que não pode ser paralelizável ( $0 \leq S \leq 1$ )
- ⌚  $Speedup = \frac{S+(1-S)}{S+\frac{(1-S)}{N}} = \frac{1}{\frac{1}{N} + (1-\frac{1}{N}) \cdot S}$
- ⌚  $S = 0 \rightarrow Speedup = N$
- ⌚  $S = 1 \rightarrow Speedup = 1$

120

# Propriedades de um Relógio de Tempo Real



121

# Propriedades de um Relógio TR



⌚ Corretude padrão

$$\oplus |C(t) - C_s(t)| \leq \epsilon$$

⌚ Flutuação Limitada

$$\oplus \text{Taxa de mudança do valor do relógio: } \frac{d(C(t))}{dt}$$

$$\oplus \text{Relógio perfeito: } C(t) = t \quad e \quad \frac{d(C(t))}{dt} = 1$$

$$\left| \frac{d(C(t))}{dt} - 1 \right| \leq \rho$$

122

# Propriedades de um Relógio TR



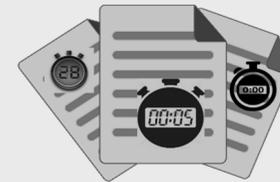
⌚ Monotonicidade

⌚ Para todo  $t_1$  e  $t_2$ , tal que  $t_1 > t_2$ ,  $C(t_1) > C(t_2)$

⌚ Regra violada apenas quando o relógio é atrasado por correções

123

# Avaliação de Desempenho



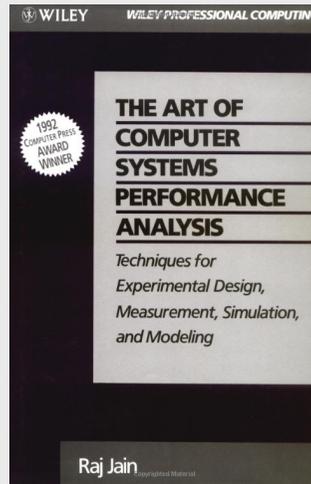
124

## Aval. Desempenho



⌚ The Art of Computer Systems Performance Analysis, Raj Jain

⌚ 2001



125

## Passos para Aval. Desempenho



- ⌚ Entender o problema: "Um problema bem entendido já tem metade da solução" 
- ⌚ Estabelecer os objetivos e definir o sistema com suas fronteiras
  - ⌚ Métricas, carga, metodologia, parte do sistema a analisar: dependem desses objetivos
- ⌚ Listar funções, entradas e saídas do sistema
- ⌚ Selecionar as métricas/critérios para comparação

126

## Passos para Aval. Desempenho



Listar/selecionar os parâmetros que podem afetar o desempenho

Variar os parâmetros possíveis e adequados

Selecionar a carga do sistema

Projetar a sequência de testes para minimizar o tempo e maximizar os resultados

Analisar, interpretar e apresentar os dados

Uso de linguagem e forma de apresentação adequadas

127

## Passos para Aval. Desempenho



- Uso de técnica correta de avaliação
  - Medição, simulação, modelagem analítica
- Usar o nível de detalhe mais adequado
- Tratar corretamente os outliers (valores fora do padrão)
- Detalhar suposições/limitações



128

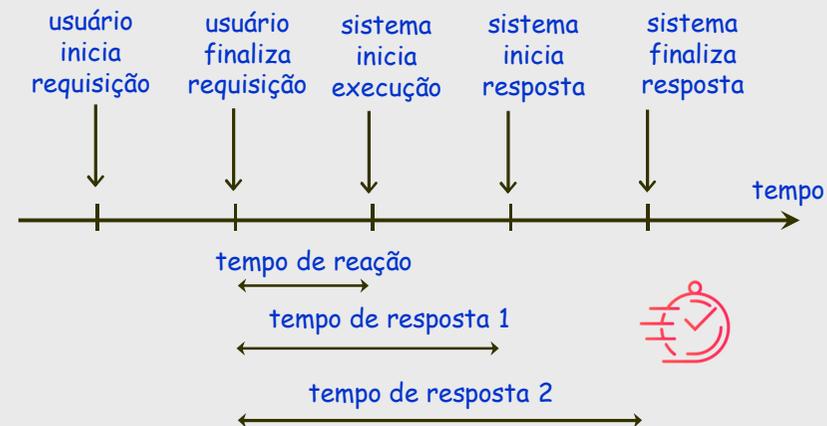
## Técnicas para Avaliar o Desempenho



	Modelo Analítico	Simulação	Medição
Estágio	qualquer	qualquer	pós-protótipo
Duração	curta	média	variável
Ferramentas	analistas	software	instrumentação
Acurácia	baixa	média	variável
Custo	pequeno	médio	grande
Escalabilidade	baixa	média	grande

129

## Tempo de Resposta



130

## Workload



Real

Sintético: não afeta a operação e pode ser reproduzido

131

## Monitores



- ⌚ Ferramenta usada para observar as atividades de um sistema → desempenho
- ⌚ Utilidades
  - ⌚ Descobrir os trechos de código usados com maior frequência
  - ⌚ Medir uso de recursos → obter gargalos
  - ⌚ Sintonizar o sistema por meio dos parâmetros
  - ⌚ Medir o workload real

132

## Monitores



### Classificação

Implementação:  
Hardware, Software,  
Firmware, Híbrido

Mecanismo de Disparo:  
por Evento, por Tempo

Exibição de Resultados:  
On-line, Batch

### Resultados

Quais programas devem  
ser otimizados

Quais programas  
provocam mais falta de  
páginas na memória



133

## Monitores



	Hardware	Software
Taxa de Monitoração	alta	depende da CPU
Resolução Temporal	pequena	média
Overhead	baixo	alto
Armazenamento	simples	tem overhead
Disponibilidade	total	não opera em crash
Custo	Alto	médio

134

## Simulação - Checklist



### ⌚ Antes do desenvolvimento

- ⌚ Especificação dos objetivos
- ⌚ Nível de detalhe do modelo apropriado aos objetivos
- ⌚ Tempo suficiente para o desenvolvimento

### ⌚ Durante o desenvolvimento

- ⌚ Gerador de números randômicos certificado
- ⌚ Modelo revisado e documentado periodicamente



135

## Simulação - Checklist



### ⌚ Após o início da operação

- ⌚ Tempo apropriado para simulações
- ⌚ Transientes iniciais removidos
- ⌚ Modelo validado com os resultados



136



Como melhorar o  
desempenho de um  
hardware já instalado e  
em uso em um Sistema  
de Tempo Real?

