

# TM-405 - Arquitetura de Computadores I

Aula 1 - INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

Raul Sena Ferreira

# Distribuição do custo em um computador

O processador é a parte mais custosa de um sistema computacional:

- Gabinete: 6% (folha metálica, plástico: 2%; fonte de alimentação, ventoinhas: 2%; cabos, porcas, parafusos: 1%; caixa de transporte, manuais: 1%)
- Placa do processador: 37% (**processador: 22%**; DRAM (128MB): 5%; Placa de vídeo: 5%; Placa-mãe (E/S e rede on-board): 5%)
- Dispositivos de E/S: 37% (teclado e mouse: 3%; monitor: 19%; disco rígido: 9%; unidade de DVD: 6%)
- SO + pacote office básico: 20%

\*Fonte: H&P, 2003

# Visão Geral da arquitetura da Máquina

Funções essenciais de um computador:

- Processamento de dados
- Armazenamento de dados
- Movimentação de dados
- Controle
  - Fornece instruções para o computador
  - Gerencia os recursos do computador

# Estrutura interna de um computador

Central processing unit (CPU): Controla o funcionamento do computador e executa suas funções de processamento

- Unidade de controle (UC): Controla o funcionamento da CPU
- Unidade Lógica e Aritmética (ULA): Realiza operações de comparação e alguns cálculos como adição e subtração
- Registradores: Armazena os dados internamente para a CPU

# Estrutura interna de um computador

Memória principal: Armazena dados

E/S: Move os dados entre o computador e o seu ambiente externo

Sistema de intercomunicação: Fornece comunicação entre CPU, memória principal e E/S

# Clock

Sinal elétrico usado para coordenar ações de dois ou mais circuitos eletrônicos:

- Usados na lógica seqüencial
- Mudança de estado (alto ou baixo)

Ciclos de clock: Relatar o desempenho em termos do número de ciclos gastos ao invés de tempo

$$\frac{\text{segundos}}{\text{programa}} = \frac{\text{ciclos}}{\text{programa}} \times \frac{\text{segundos}}{\text{ciclo}}$$

# Clock

Taxa de clock (frequência) = ciclos por segundo (1Hz = 1 ciclo / segundo)

Ex.: Processador de 2.4GHz tem um tempo de ciclo de:

$$\frac{1}{2.4 \times 10^9} = 0,42 \times 10^{-9} \text{s} = 0,42 \text{ns}$$

# Memória

É responsável pelo armazenamento das instruções e dados

Hierarquia de memória: RAM, caches, registradores, memória secundária e memória virtual

RAM e caches: compostas por um conjunto de células, onde cada célula possui um determinado número de bits

O bit é a unidade básica de memória e pode assumir apenas os valores 0 e 1



# Memória



# Dispositivos de E/S

Permitem a comunicação entre o computador e o mundo externo

- Barramento: São linhas de comunicação entre a CPU, a memória principal e os dispositivos de E/S
- É um conjunto de fios paralelos, onde trafegam informações, como dados, endereços ou sinais de controle
- Tipo: Unidirecional. Bidirecional.
- Forma: Paralela. Sequencial.

# Arquiteturas RISC × CISC

## RISC - Reduced Instruction Set Computer

- Executa menos instruções básicas com mais eficiência e rapidez
- Utiliza um conjunto reduzido de instruções de linguagem de máquina, o que na maioria das vezes pode ser executada num único ciclo do processador
- Exemplo: Sparc (SUN), RS-6000 (IBM)

## CISC - Complex Instruction Set Computer

- Possuem instruções complexas que são executadas por microprogramas
- Exemplo: Pentium (Intel), 68xxx (Motorola)

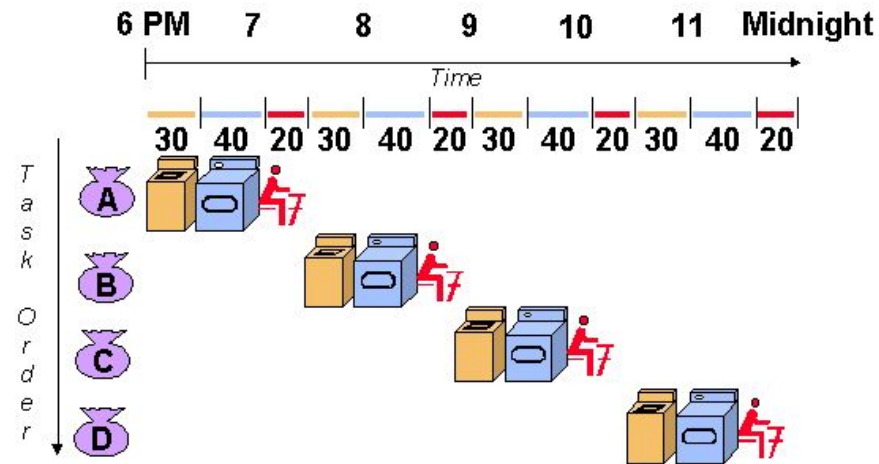
# Pipelining

Técnica utilizada em processadores para otimizar a execução de instruções

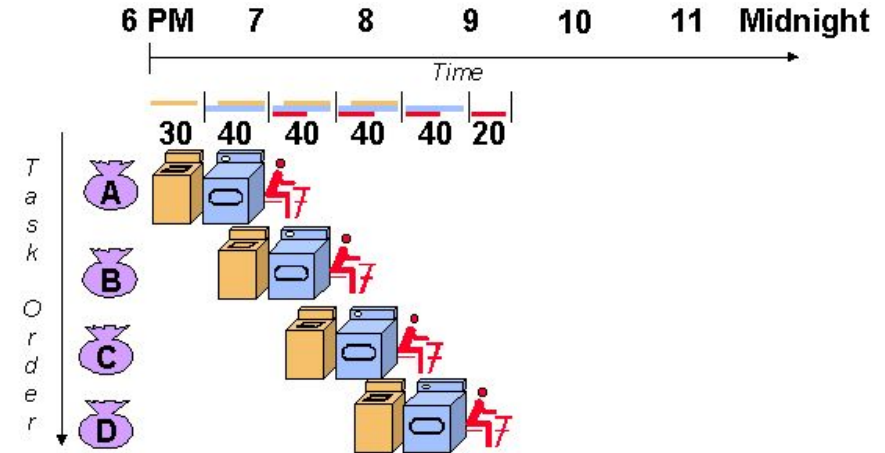
- Divisão da instrução em tarefas
- Sobreposição das tarefas
- Atributo presente em arquiteturas RISC

Diminui drasticamente o tempo de execução de tarefas **independentes entre si**

# Pipelining



Source: <http://www.ece.arizona.edu/~ece462/Lec03-pipe/>



Source: <http://www.ece.arizona.edu/~ece462/Lec03-pipe/>

# Pipelining

Desvantagens: Não funciona bem se existir dependência entre tarefas:

- add \$r3, \$r2, \$r1
- add \$r5, \$r4, \$r3

A segunda instrução é dependente da primeira

Compilador tenta resolver esse problema fazendo reordenação do código

# Sistemas de Numeração

Humanos - O sistema de numeração mais utilizado é o decimal (base 10)

Computador - O sistema de numeração é o binário (base 2)

- Corrente elétrica - ligado e desligado - 0,1
- Valores representados por combinações de símbolos 0 e 1s (bits)
  - 8 bits = 1 byte
  - 1024 bytes = 1 Kilobyte (KB)
  - 1024 KBs = 1 Megabyte (MB)
  - 1024 MBs = 1 Gigabyte (GB)
  - 1024 GBs = 1 Terabyte (TB)
  - 1024 TBs = 1 Petabyte (PB)
- O sistema binário é adequado às máquinas digitais

# Sistemas de Numeração

Conversão: decimal  $\rightarrow$  binário:

$$\begin{array}{r} \text{▶ } 3_{10} = 11_2: \quad 3 \overline{) 2} \\ \quad \quad \quad \underline{1} \quad 1 \overline{) 2} \\ \quad \quad \quad \quad \quad \underline{1} \quad 0 \end{array}$$

$$\begin{array}{r} \text{▶ } 8_{10} = 1000_2: \quad 8 \overline{) 2} \\ \quad \quad \quad \underline{0} \quad 4 \overline{) 2} \\ \quad \quad \quad \quad \quad \underline{0} \quad 2 \overline{) 2} \\ \quad \quad \quad \quad \quad \quad \underline{0} \quad 1 \overline{) 2} \\ \quad \quad \quad \quad \quad \quad \quad \quad \underline{1} \quad 0 \end{array}$$

Conversão: binário  $\rightarrow$  decimal:

$$\text{▶ } 11_2 = 3_{10}: 1 \times 2^1 + 1 \times 2^0$$

$$\text{▶ } 1000_2 = 8_{10}: 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$$



# Sistemas de Numeração

Representando números negativos através de **sinal e magnitude**:

- $-13_{10} = 10001101_2$
- $+13_{10} = 00001101_2$

Desvantagens:

- Perde-se 1 bit para representar o sinal do número
- Duas representações para o 0 (zero)

# Sistemas de Numeração

## Complemento a 1:

- Converter o módulo do número em decimal para binário
- Completar com 0 (zero) à esquerda até o número ter a quantidade de bits do problema em questão
- Calcular quanto falta em cada casa para alcançar o maior valor possível (base-1= 1)

Exemplo  $-13_{10}$  com 8 bits:

- Valor em binário (8 bits):  $13_{10} = 00001101_2$ .
- Quanto falta em cada casa para chegar a 1?:  $11110010_2$

Problema: Duas representações para zero -  $00000000_2$  e  $11111111_2$

# Sistemas de Numeração

Complemento a 2:

- Converter o módulo do número em decimal para binário
- Completar com 0 (zero) à esquerda até o número ter a quantidade de bits do problema em questão
- Calcular quanto falta em cada casa para alcançar o maior valor possível (base-1= 1)
- Soma 1. Se acontecer o “vai 1” no último bit (o mais significativo), então deve ser ignorado (apenas no bit mais significativo).

Não há duas forma de representar 0

"A smart terminal is **not** a smart\*ass\* terminal,  
but rather a terminal you can educate"

Rob Pike