

Capítulo 4 – Processadores funcionamento básico

Introdução

O processador, também chamado de microprocessador, CPU (Central Processor Unit) ou UCP (Unidade Central de Processamento), é um circuito integrado capaz de processar dados do computador.

CPU → CENTRAL PROCESSOR UNIT

UCP → UNIDADE CENTRAL DE PROCESSAMENTO



O modelo de Von Neumann

Praticamente todos os processadores modernos seguem um modelo criado por Von Neumann em 1945 quando estava desenvolvendo um computador chamado **EDVAC**.

Este modelo também pode ser chamado de computador de programa armazenado. A principal característica deste modelo é a programação do processador através de instruções dadas a ele através de programas.

Este modelo é constituído dos seguintes componentes:

1) Central Aritmética: É a unidade responsável pelos cálculos, também conhecida como unidade lógica e aritmética ou ULA, unidade de inteiros se for capaz somente de executar cálculos com valores inteiros e unidade de ponto flutuante se possuir a capacidade de executar cálculos com números reais.

ULA → UNIDADE LÓGICA E ARITMÉTICA

→ CÁLCULOS LÓGICOS (V E F)

→ CÁLCULOS ARITMÉTICOS (+ - * /)

UAL → UNIT ARITHMETICAL LOGICAL

PODE SER CHAMADA DE:

UNIDADE DE PONTO FLUTUANTE → SE EXECUTA CÁLCULOS COM CASAS DECIMAIS;

UNIDADE DE INTEIROS → SE EXECUTA CÁLCULOS SOMENTE COM NÚMEROS INTEIROS;

2) Central de controle (CC): Usada para buscar instruções na memória e interpretá-las (decodificá-las), isto é, converter instruções em sinais de controle que vão ativar /desativar outras partes do processador.

UC → UNIDADE DE CONTROLE

→ **BUSCAR** INSTRUÇÕES;

→ **INTERPRETAR** INSTRUÇÕES;

→ **INFORMAR DE ONDE VÊM E PARA ONDE VÃO AS INSTRUÇÕES;**

→ **PARTE MAIS COMPLEXA DA CPU;**

CU → CONTROL UNIT

3) Memória (M) : Possui duas funções:

→ **Armazenar dados necessários para a execução de instruções;**

→ **Armazenar a sequência de instruções (programa) a serem executadas e seus respectivos dados, se existentes.**

(USUÁRIO)

COMANDOS → **SISTEMA OPERACIONAL** → **INSTRUÇÕES** → **PROCESSADOR**

**(FACILMENTE
COMPREENDIDOS
PELO USUÁRIO)**

**(CÓDIGOS
COMPREENDIDOS
PELO PROCESSADOR)**

→ RAM (MEMÓRIA PRINCIPAL);

→ CACHE (MEMÓRIA AUXILIAR);

→ MEMÓRIAS VOLÁTEIS (MEMÓRIAS QUE PERDEM AS INFORMAÇÕES QUANDO CORTAMOS A ALIMENTAÇÃO ELÉTRICA);

4) Meio externo de gravação (R): É o que chamamos atualmente de dispositivos de entrada e saída.

→ **LOCAIS ONDE AS INFORMAÇÕES SERÃO ARMAZENADAS;**

→ **MEMÓRIAS NÃO VOLÁTEIS (MANTÉM AS INFORMAÇÕES ARMAZENADAS MESMO QUE A ENERGIA SEJA CORTADA);**

→ **HD / SSD / PEN DRIVE**

5) Entrada (I, Input) : Caminho para a conexão do meio externo de gravação à central de controle de memória.

→ **DISPOSITIVOS DE ENTRADA;**

→ **TECLADO, MOUSE, WEBCAM, SCANNER;**

6) Saída (O, Output) : Caminho para a conexão central de controle e da memória ao meio externo de gravação.

→ **DISPOSITIVOS DE SAÍDA;**

→ **IMPRESSORA, MONITOR, CAIXA DE SOM**

Instruções

São números que possuem um significado particular para o processador. Este significado varia de acordo com a linha de processadores.

→ INSTRUÇÕES SÃO NUMÉRICAS;

Todo processador possui um conjunto de instruções, que é justamente uma lista contendo as instruções que o processador possui e seu código. O conjunto de instruções utilizados pelos PCs é conhecido como X86 ou X87.

→ TODO PROCESSADOR TEM UM CONJUNTO FINITO DE INSTRUÇÕES;

→ CONJUNTO DE INSTRUÇÕES X86 E X87;

Linguagem de programação

Um programa ou código é uma sequência de instruções .

1 COMANDO → N INSTRUÇÕES

Linguagem de máquina ou linguagem de baixo nível

Linguagem que o processador é capaz de compreender . Como por exemplo a linguagem Assembly.

→ CÓDIGOS NUMÉRICOS (INSTRUÇÕES)

Linguagem de alto nível

Linguagens de programação mais fáceis de serem utilizadas, mas não entendidas pelo processador, linguagens de alto nível devem ser convertidas para linguagens de baixo nível para que o processador as compreenda.

→ COMANDOS (FACILMENTE COMPREENDIDOS PELOS USUÁRIOS)

CISC E RISC

Há duas categorias básicas de conjunto de instruções:

CISC (Complex Instruction Set Computing)

- Computação utilizando conjunto complexo de instruções.
- Instruções que não tem tamanho padronizado;
- **Instruções de tamanho variado;**
- Por serem complexas precisam ser divididas em diversas etapas internas dentro do processador, chamadas de microcódigo.
- As instruções X86 são do tipo CISC.
- PCs;
- MAIS BARATOS;

EXEMPLO :

→ LAVAR A CASA;

RISC (Reduced instruction Set Computing)

- Computação com conjunto de instruções de tamanho reduzido.
- **Possuem tamanho padronizado;**
- São sempre executadas na mesma quantidade de tempo;
- Por serem simples necessitam de mais instruções para terem o mesmo resultado final.
- APPLES;

EXEMPLO:

- LAVAR BANHEIRO;
- LAVAR A COZINHA;
- LAVAR O QUARTO;
- LAVAR A SALA;

Tempo de execução

O tempo de execução de uma instrução é medido em ciclos de máquina, isto é, pulsos de clock (clock interno do processador).

PROCESSADOR I9

2,7GHZ = 2.700.000.000 PULSOS DE CLOCK POR SEGUNDO

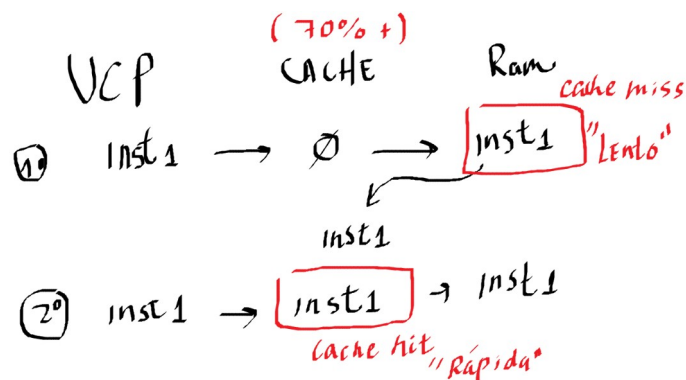
- **1 INSTRUÇÃO PODE LEVAR VÁRIOS CICLOS DE CLOCK PARA SER EXECUTADA;**

Memória

- Local onde o processador busca instruções a serem executadas.
- MEMÓRIA PRINCIPAL

Memória cache

- Memória de alta velocidade intermediária entre o processador e a RAM.
- ACELERAR PROCESSAMENTOS REPETIDOS;

**Registradores**

- São pequenas porções de memória dentro do processador usadas para armazenagem temporárias dos dados.
- **MEMÓRIAS INTERNAS DO PROCESSADOR;**
- **MEMÓRIAS MAIS RÁPIDAS DO COMPUTADOR;**
- **SÃO UM NÚMERO FINITO E LIMITADO DE DENTRO DO PROCESSADOR;**

* Registradores de uso geral

→ Também chamados de GPRS (General Purpose Registers), estão disponíveis para serem utilizados pelo programador como bem entender.

→ **REGISTRADORES QUE PODEM SER UTILIZADOS PELO PROGRAMADOR;**

* Registradores de Pilha

→ REGISTRADOR COM TAREFAS ESPECÍFICAS PARA PILHAS;

→ Estrutura do tipo LIFO (Last in First Out), o último a entrar é o primeiro a sair.

→ Parte da memória Ram que pode ser acessada como memória temporária para uso interno do processador.

→ Dados são armazenados em uma pilha com uma instrução chamada PUSH e extraídos com uma instrução chamada POP.

→ A posição atual da pilha é controlada por um registrador chamado ponteiro de pilha ou STACK POINTER (SP).

INSTRUÇÃO PUSH → ARMAZENA DADOS NA PILHA;

INSTRUÇÃO POP → RETIRAM DADOS DA PILHA;

REGISTRADOR QUE APONTA PARA O TOPO DA PILHA

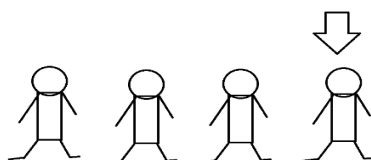
→ STACK POINTER (SP)

DOIS TIPOS BÁSICOS DE ESTRUTURAS DE ARMAZENAMENTO E RECUPERAÇÃO DE DADOS:

→ **FILA**

→ PEPS (PRIMEIRO A ENTRAR PRIMEIRO A SAIR)

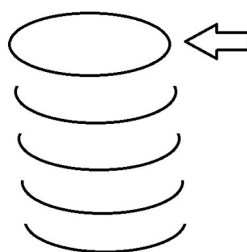
→ FIFO (FIRST IN FIRST OUT)



→ **PILHA**

→ UEPS (ÚLTIMO A ENTRAR PRIMEIRO A SAIR)

→ LIFO (LAST IN FIRST OUT)



Carry : Indica que houve um vai um ou empréstimo no último cálculo executado.

CARRY = 1 (ATIVO)

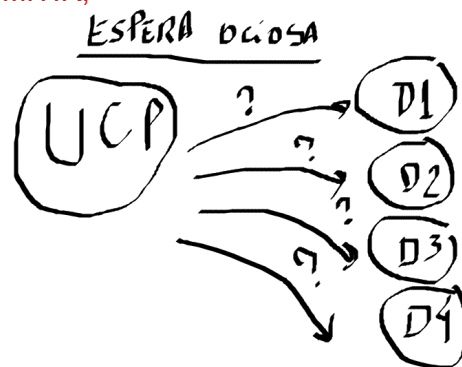
$$\begin{array}{r}
 \\
 \\
 \\
 \\
 \\
 \hline
 1 1 1 1 0
 \end{array}
 \quad (CARRY)$$

Interrupções

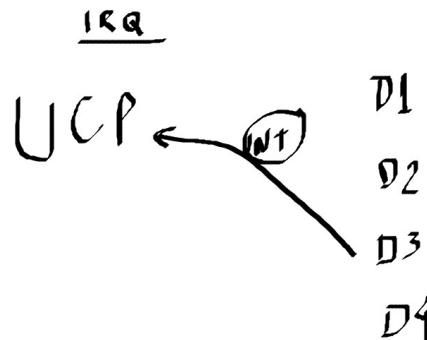
ANTES :

ESPERA OCIOSA:

→ O PROCESSADOR FICAVA EM LOOPING INFINITO PERGUNTANDO AO DISPOSITIVO SE O MESMO QUER TRANSMITIR;



Interrupções são pedidos ao processador para que ele para a execução do programa e vá atender a quem efetuou o pedido. As interrupções podem ser de hardware ou software.



IRQ – INTERRUPTION REQUEST

O DISPOSITIVO SÓ INTERROMPE O PROCESSADOR EM DUAS SITUAÇÕES:

- PARA INICIAR A ENTRADA E SAÍDA;
- PARA TERMINAR A ENTRADA E SAÍDA;

INTERRUPÇÕES DE HARDWARE → EXECUTADAS PELO DISPOSITIVO;
INTERRUPÇÕES DE SOFTWARE → EXECUTADAS PELOS PROGRAMAS;

As interrupções de hardware são executadas pelos dispositivos . E podem ser classificadas como mascaráveis e não mascaráveis;

AS INTERRUPTÕES DE HARDWARE PODEM SER CLASSIFICADAS COMO:

→ **Interrupções de hardware mascaráveis** → São as que são pedidas ao processador por um pino chamado INTR ou equivalente, podem ser desabilitadas por software.

→ **PEDIDAS POR UM PINO ESPECÍFICO – INTR;**

→ **Interrupções não mascaráveis** → são pedidas ao processador através de um pino chamado NMI, e não tem como serem ignoradas.

→ **PEDIDAS POR UM PINO ESPECÍFICO – NMI; (NÃO MASCARÁVEL INSTRUÇÃO)**

As interrupções de software são geradas por programas quando eles necessitam da atenção imediata do processador.

Execução de instruções CISC

As instruções CISC seguem a ordem do diagrama abaixo:

MEMÓRIA RAM
UNIDADE DE BUSCA
UNIDADE DE DECODIFICAÇÃO
UNIDADE DE EXECUÇÃO

ESSE SEQUÊNCIA É CHAMADA DE CAMINHO DE DADOS;

- | | |
|--------------------------|--|
| 1) PRÉ-BUSCA / PRÉ FETCH | → ESTÁGIOS ANTES DA BUSCAS DAS INTRUÇÕES |
| 2) BUSCA / FETCH | → BUSCAR AS INSTRUÇÕES |
| 3) DECODIFICAÇÃO | → DECODIFICAR |
| 4) DATA FETCH | → BUSCA DE DADOS |
| 5) EXECUTE | → EXECUTAR |
| 6) WRITE BACK | → RESPONDER |

ANTES:

PF	FETCH	DEC	DF	EX	WB
X	X	X	X	X	01

ATUALMENTE PIPELINE (NÃO DEIXAS AS UNIDADES OCIOSAS) (DEIXA O SISTEMA MAIS RÁPIDO)

PF	FETCH	DEC	DF	EX	WB
06	05	04	03	02	01

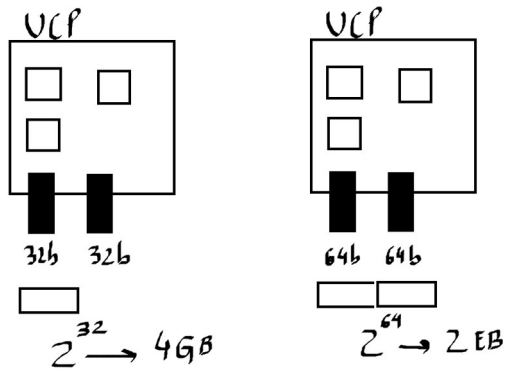
Uma técnica conhecida para aumentar o desempenho do processador é a chamada de **pipelining**, que permite que unidades que estejam ociosas no processador possam começar sua próxima etapa antes do processador ter terminado de executar a instrução anterior.

Processador de X bits

Um processador de 32 bits pode executar instruções de até 32 bits e um processador de 64 bits pode executar instruções de até 64 bits.

PROCESSADOR 32 BITS

PROCESSADOR 64 BITS



- PROCESSADORES COM MAIS BITS TEM MELHOR PERFORMANCE;
- PROCESSADORES COM MAIS BITS RECONHECEM MAIS MEMÓRIA;
- PROCESSADORES COM MAIS BITS PODEM TER INSTRUÇÕES MAIORES;