

UNIDADE CENTRAL DE PROCESSAMENTO

O processador, também conhecido como CPU (central processing unit, em inglês), ou UCP (unidade central de processamento, em português), é formado por chips responsáveis pela execução de cálculos, decisões lógicas e instruções que resultam em todas as tarefas que um computador pode fazer. Afinal de contas, todos os programas que estão em utilização em um computador, obrigatoriamente, devem ser executados pelo processador.

Atenção

Lembrando o que vimos na unidade anterior, um programa consiste em uma série de instruções que o processador deverá executar para que a tarefa solicitada seja realizada.

Antes de estudarmos como funciona um processador, você tem ideia de como ele opera no computador? Como ele se relaciona com os demais componentes do sistema computacional?

Assista a animação disponível em http://www.youtube.com/watch?v=oui_qEhe3P4, que trata o processador como um personagem, demonstrando-o em diversas situações de interação com os demais componentes da máquina com a responsabilidade de fazer com que tudo funcione. Após assisti-lo, comente no fórum suas impressões sobre o papel do processador.

Execução de programas

Para que um programa seja executado, é necessário que sejam transferidos todos os dados necessários, a partir de algum dispositivo de armazenamento para a memória RAM, de onde serão acessados pelo processador. Ao ser processado, ou seja, após o processador executar as instruções que compõem o programa, o resultado é entregue ao programa que será o responsável por determinar o que será feito com ele.

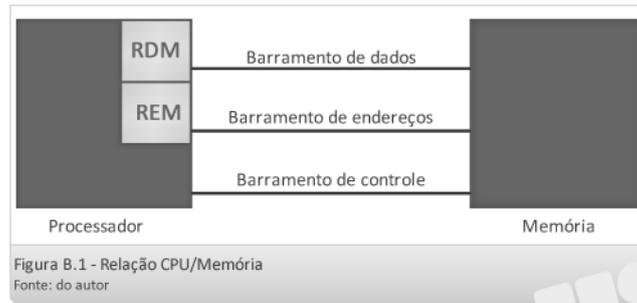
Relação CPU/Memória RAM

O processador trabalha apenas com valores armazenados em registradores na execução de suas atividades, ou seja, ele não acessa diretamente as informações da memória RAM.

Atenção

Os registradores são áreas de armazenamento temporário de valores presentes no processador. Neles são carregados os valores da memória necessários para a execução de uma instrução, sendo que também são utilizados como local de armazenamento para os resultados das execuções das instruções.

Portanto, para que uma instrução seja executada é necessário acessá-la em seu endereço na memória RAM e transferi-la para o registrador do processador.



A transferência de dados entre processador e memória é realizada através do barramento local, utilizando-se dos seguintes componentes:

- REM (Registrador de Endereço de Memória): armazena o endereço da memória onde será lido ou escrito o dado.
- RDM (Registrador de Dados da Memória): armazena o dado a ser escrito ou lido na memória.
- Barramento de dados: liga o RDM à memória, sendo o caminho por onde é feita a transferência de dados.
- Barramento de endereços: liga o REM à memória fornecendo o endereço a ser lido ou escrito.
- Barramento de controle: liga o processador à memória para enviar sinais de controle como leitura (READ), escrita (WRITE) ou espera (WAIT).

As operações, que são realizadas pelo processador na memória, envolvem a leitura de dados da memória para armazená-los nos registradores, já a escrita desses dados estão armazenados em registradores na memória.

A operação de leitura (READ), quando o conteúdo da posição de memória endereçada por REM é copiado em RDM, envolve a seguinte sequência de operações:

```
REM ← endereço
Comando READ
RDM ← Memória[REM]
```

A operação de escrita (WRITE), quando a posição de memória endereçada por REM recebe o conteúdo de RDM, envolve a seguinte sequência de operações:

```
REM ← endereço
RDM ← dado
Comando WRITE
Memória[REM] ← RDM
```

Clock

O processador, no desempenho de suas atividades, necessita realizar operações internamente e também comunicar-se com os demais componentes do micro (para ler e gravar informações na memória principal, por exemplo). Segundo Torres:

Os processadores atualmente utilizam um esquema de multiplicação de clock, onde o clock do barramento local é muito inferior ao clock interno do processador. Esse esquema de multiplicação de clock foi criado porque é difícil construir placas-mãe e circuitos de apoio que consigam operar em frequências de operação tão altas como aquelas que os processadores conseguem trabalhar internamente. (2001, p. 34)

Portanto, o clock interno do processador é a frequência com a qual o processador trabalha internamente na execução de suas atividades, sendo utilizado o clock externo para a comunicação com os demais componentes do computador através da placa-mãe.

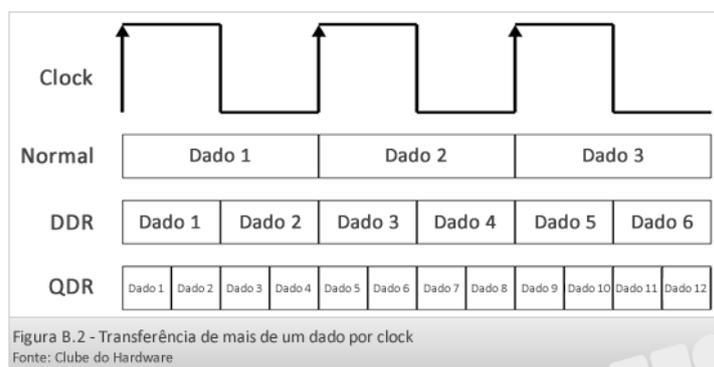
Atenção

Imagine um processador que possui uma frequência de operação (clock interno) de 1,8 GHz e que se comunica com uma placa-mãe que possui uma frequência de operação de 200 MHz. Como a frequência de operação entre ambos é completamente diferente, é necessário que o processador reduza sua frequência quando necessitar se comunicar com a placa-mãe, para isso, é utilizado um multiplicador que, neste caso, é 9. Sendo assim, dividindo o clock interno do processador, que é 1,8 GHz, por 9, obteremos o valor de clock de 200 MHz ($1,8 \text{ GHz} / 9 = 200 \text{ MHz}$), que é o necessário para que possa ocorrer comunicação com o restante do micro.

Técnicas para minimizar a diferença de clock

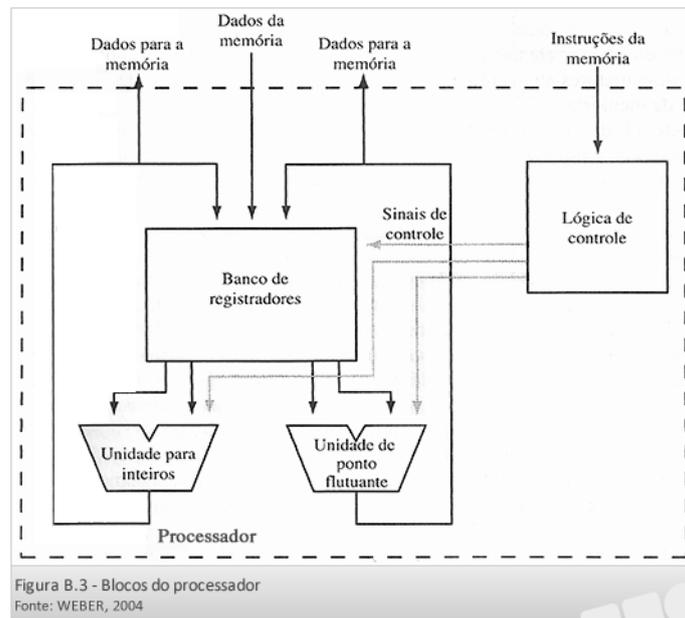
Visando minimizar a diferença existente entre o clock interno e o clock externo do processador são utilizadas algumas técnicas. Uma solução envolve a utilização de memória cache (que será abordada no próximo módulo, juntamente com o sistema de memórias), que se trata de uma memória de maior velocidade de acesso que a memória principal.

A transferência de mais de um dado por ciclo de clock também é uma técnica empregada para compensar essa diferença. Ela consiste em transferir dois dados por ciclo de clock, a chamada DDR (Dual Data Rate), assim como transferir quatro dados por ciclo de clock, a QDR (Quad Data Rate).



Arquitetura do Processador

Basicamente, um processador, em sua arquitetura, pode ser dividido em vários blocos:



Unidade operacional (execução)

Contém o hardware que executa as instruções, inclusive os responsáveis pela busca e decodificação de instruções e a unidade lógica e a aritmética que executam os cálculos.

Banco de registradores

Pequena área de armazenamento para os dados que o processador está usando, possibilitando acesso rápido.

Unidade de controle

Responsável pelo controle do restante do processador, determinando instruções a serem executadas e quais operações são necessárias para executar cada instrução.

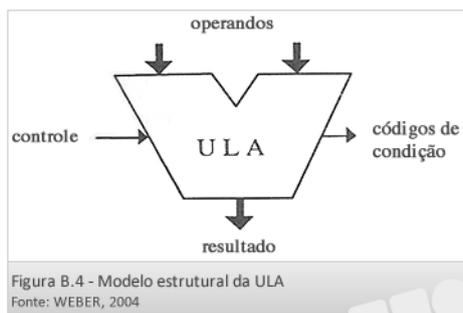
Unidade operacional

Ela é a responsável por executar as transformações sobre dados especificados pelas instruções do computador.

O número, tamanho e uso dos registradores e a quantidade e tipo de operações que a unidade lógica e aritmética realiza são alguns dos fatores que determinam o porte de um processador.

Unidade lógica e aritmética (ULA)

Nela são realizadas as operações aritméticas e operações lógicas sobre um ou mais operandos. Muitas vezes, as operações da ULA são indicadas por instruções simples. As funções mais complexas, exigidas pelas instruções de máquina, são realizadas pela ativação sequencial das várias operações básicas disponíveis.



A ULA apresenta as seguintes características:

- comprimento em bits dos operandos
- número e tipo de operações
- códigos de condição gerados

Os códigos de condição gerados servem como indicações sobre as operações realizadas, podendo ser, segundo Weber (2004):

- Overflow: resultado de uma operação não pode ser representada no espaço disponível (tamanho da palavra insuficiente).
- Sinal: indica se o resultado da operação é negativo ou positivo.
- Carry: dependendo da operação realizada (soma ou subtração) pode representar o bit de vai-um (carry-out) ou vem-um (borrow-out).
- Zero: indica se o resultado da operação realizada é nulo.

Acumulador (AC)

É um registrador que tem como função armazenar um operando que será utilizado, ou o resultado fornecido pela ULA. Nos computadores mais simples pode existir um acumulador e nos mais complexos podem ser encontrados vários registradores com a função de um acumulador. Dentre sua principal característica está o seu comprimento em bits, que indica a quantidade de bits que ele pode armazenar, sendo que em cada nova operação, um dado é copiado para o seu interior, fazendo com que o conteúdo antigo seja perdido (WEBER, 2004).

Unidade de controle (UC)

Sua principal finalidade é a de fornecer os sinais de controle que gerenciam o fluxo interno de dados no processador. É ele quem coordena o instante preciso em que ocorrem as transferências entre um componente do processador e outro na execução de uma instrução, conforme figura abaixo:



Cada sinal de controle gerado por ela comanda uma micro-operação a ser realizada. Ela recebe como entrada o valor do Registrador de Instrução e decodifica-o, juntamente com os sinais de saída da ULA (RST). Para cada código de instrução ela gera uma sequência de sinais diferentes, ativando os circuitos correspondentes para que cada uma das tarefas necessárias para a busca e execução da instrução seja completada.

As tarefas podem incluir a carga de valores em um registrador, seleção de um dado para entrada em componente, ativação da memória, seleção de uma operação da ULA, habilitação de um circuito lógico, etc. (WEBER, 2004)

Banco de registradores

No computador existem registradores que executam funções específicas e que, dependendo da arquitetura, podem ser encontrados em diferentes blocos.

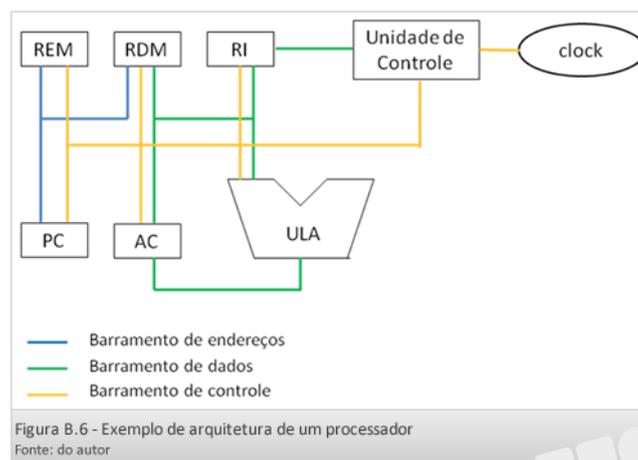
- Apontador de instruções (Contador de programa – PC)
 - Sua função é manter atualizado o endereço de memória da próxima instrução que deve ser executada após a atual.
- Registrador de instruções (RI)
 - Armazena o código da instrução que está sendo executada. É em função dele que a unidade de controle determina quais sinais de controle devem ser gerados para executar as operações determinadas pela instrução. O comprimento em bits do RI está ligado ao tamanho e à codificação das instruções do computador.
- Registrador de estado (RST)
 - Nele ficam armazenados os códigos de condição gerados pela unidade lógica e aritmética. Em função dele, em conjunto com o RI, que a UC toma decisões sobre a geração ou não de certos sinais de controle.

Demais elementos do processador

Além dos blocos que compõem o processador, também podem ser encontrados os seguintes elementos:

- Clock (relógio)
 - É usado para manter o sincronismo do funcionamento entre todos os componentes do processador.
- Decodificador de instruções
 - É um dispositivo utilizado para identificar as operações a serem realizadas com base na instrução a ser executada.
- Barramento Interno de Dados
 - É o caminho utilizado na transferência de dados entre os registradores e entre registradores e a ULA.
- Barramento Interno de Endereços
 - É o caminho utilizado para a transferência de endereços entre os registradores.
- Barramento Interno de Controle
 - É o caminho utilizado para transmitir os sinais da unidade de controle que comandam o funcionamento de cada circuito do processador.

O conjunto de todos os elementos que formam o processador será organizado de acordo com a arquitetura utilizada, como, por exemplo, pode ser visto na figura abaixo:



Execução de Instruções

Um conjunto de instruções é um conjunto de bits devidamente codificados que indica ao computador que sequência de micro-operações ele deve realizar. Elas podem ser classificadas, dentre outras formas, como (WEBER, 2004):

- Instruções de transferência de dados
- Instruções aritméticas e lógicas
- Instruções de teste e desvio

O conjunto de instruções envolve o conjunto de todas as instruções que um computador reconhece e pode executar. Todo programa é formado por uma sequência finita de instruções de um determinado conjunto de instruções. A maioria das instruções que formam um programa realiza operações que envolvem operandos (dados), que podem estar em qualquer posição de memória ou em algum registrador.

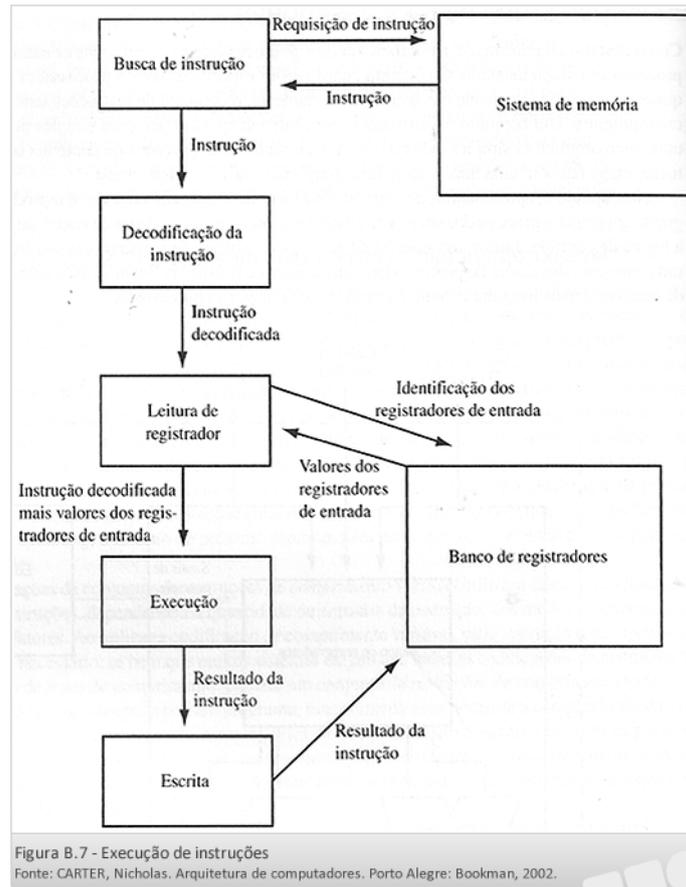
Para que a UC consiga localizar o operando é necessário que o endereço dele seja indicado junto com a instrução. Nas instruções que envolvem a operação de desvio é necessário indicar para qual posição ou endereço de programa será realizado o desvio.

Sequência de Funcionamento

A sequência para a realização de uma instrução pelo processador é conhecida como ciclo de “busca – decodificação – execução” de instruções.

Uma instrução é buscada por vez para ser executada, de acordo com os seguintes passos:

- a) Busca da próxima instrução na memória;
- b) Decodificação da instrução;
- c) Dados que servirão como operandos são buscados na memória, se for necessário;
- d) Cálculo do endereço da próxima instrução;
- e) Execução da instrução;
- f) Armazenamento do resultado.



No ciclo de instrução, as seguintes ações são executadas:

- Transferência do conteúdo do endereço de memória apontado pelo contador de programa (PC) como sendo da próxima instrução a ser executada para o RI.
- Atualizar o valor de PC para indicar o endereço da próxima instrução a ser executada.
- O decodificador de instruções recebe o código da operação e o decodifica, enviando-o para a UC.
- A UC gera os sinais necessários para que a instrução possa ser executada.

A sequência do ciclo de instruções é executada para cada nova instrução a ser executada repetidamente e se constitui nas seguintes etapas (WEBER, 2004):

- Busca
 - Copiar o PC para o registrador de endereços da memória (REM).
 - Ler uma instrução da memória.
 - Copiar o registrador de dados da memória (RDM) para o registrador de instruções (RI).
 - Atualizar o contador de programa (PC).
- Decodificação
 - Determina qual instrução deve ser executada.
- Execução
 - Cálculo de endereço dos operandos (se houver).
 - Busca dos operandos na memória (se houverem).
 - Seleção da operação a ser realizada pela ULA.
 - Carga de registradores.
 - Escrita de operandos na memória.
 - Atualização do PC (somente no caso das instruções serem desvios).

Pipelines

Nos computadores mais antigos as instruções eram executadas uma por uma, com o processador buscando uma instrução na memória, decodificando-a (determinar qual instrução era), lendo as entradas das instruções (no banco de registradores), executando os cálculos exigidos pela instrução e escrevendo os resultados de volta no banco de registradores. O problema desta abordagem é que o *hardware* necessário para executar cada um desses passos é diferente, de modo que a maior parte dele fica ocioso em um determinado momento, esperando que as outras partes do processador completem a sua parte de execução da instrução. (CARTER, 2002)

Com o objetivo de tornar ágil este processo, evita-se que elementos do processador fiquem parados, é utilizada a técnica de *pipelining* (*pipe* = tubo; *line* = sequência). Ela consiste em reduzir o tempo de execução de um conjunto de instruções, sendo que o tempo para executar uma instrução continua o mesmo, mas a quantidade de instruções executadas por um período de tempo aumenta. (CARTER, 2002)

Atenção

Ele utiliza a metodologia de uma linha de montagem, onde a fabricação de um produto qualquer é subdividida em partes (implementadas em setores) nas quais, tarefas independentes estão sendo desenvolvidas ao mesmo tempo. O tempo para a produção de um produto não diminui, mas sim o tempo entre eles.

É uma técnica de implementação de CPU onde múltiplas instruções podem estar em execução ao mesmo tempo em estágios de processamento diferentes, sendo que cada um deles é responsável pela execução de uma parte da instrução e possui o seu próprio bloco de controle. Assim que um estágio completa sua tarefa com uma instrução, passa ela para o estágio seguinte e começa a tratar da próxima instrução. Sendo que várias instruções são executadas ao mesmo tempo, ocorre um acréscimo no desempenho do processador.

A execução de uma instrução, por exemplo, pode ser dividida em 5 estágios básicos:

1. Busca de instruções (fetch)
2. Decodificação de instruções (decode)
3. Busca dos operandos e armazenamento em registradores (operand fetch)
4. Execução de instruções (execute)
5. Armazenamento do resultado em registradores (store)

Sem a utilização de pipeline a execução de uma instrução ocorreria da seguinte maneira:

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2		processo1			

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3			processo1		

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4				processo1	

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4					
5					processo1

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1					
2					
3					
4					
5					
6	processo2				

Dessa forma, quando um estágio estava ocupado executando sua função, os outros 4 estágios ficavam ociosos, assim como apenas um processo ocupa o processador, levando cinco unidades de tempo para executar e só então outro processo pode começar a ser processado.

Com a utilização de pipelines, o primeiro processo continua demorando cinco unidades de tempo para ser executado, mas logo a seguir, em cada unidade de tempo um novo processo é executado. Conforme segue:

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	
5	processo5	processo4	processo3	processo2	processo1

Tempo	Estágio1	Estágio2	Estágio3	Estágio4	Estágio5
1	processo1				
2	processo2	processo1			
3	processo3	processo2	processo1		
4	processo4	processo3	processo2	processo1	
5	processo5	processo4	processo3	processo2	processo1
6	processo6	processo5	processo4	processo3	processo2

Alguns estágios apresentam tempo de execução diferente dos outros, fazendo com que o tempo necessário para encher o pipeline e esvaziá-lo seja variável, com instruções ficando presas no pipeline, aguardando que as instruções que geram suas entradas sejam executadas. Também existe uma série de fatores que limitam a capacidade de um pipeline de executar instruções, como dependências entre instruções (leitura ou escrita de registradores que estão sendo utilizados por outras instruções), desvios (processador não sabe qual instrução deve buscar até que ocorra o desvio) e o tempo necessário para acessar a memória. (CARTER, 2002)

Devido à existência de fatores que podem comprometer o desempenho do uso de pipelines, existem técnicas que são utilizadas para amenizá-los:

- Pré-decodificação: o processador pode realizar a decodificação de instruções (paralelamente) antes do momento de elas serem executadas.
- Execução fora-de-sequência: o processador pode executar previamente um determinado número de instruções. Posteriormente, a ordem de execução é verificada e os resultados das operações são repassados na sua ordem correta.
- Previsão de desvio: caso exista uma instrução de desvio dentro do pipeline, ela pode ser calculada mais cedo (para determinar qual caminho será percorrido e descartar as instruções que não serão necessárias) no pipeline ou prever o destino de cada possibilidade de desvio, para que o processador possa procurar as instruções que serão necessárias anteriormente.

Medidas de Desempenho

Existem diversas formas de medir o desempenho de sistemas computacionais. Dentre elas podemos destacar:

- MIPS (Milhões de Instruções por Segundo): mede a execução de instruções, dividindo o número de instruções executadas de um programa pelo tempo necessário para executá-lo, não considerando que existem diferentes instruções, com tempos de execução diferentes.
- FLOPS (Operações de Ponto Flutuante por Segundo): mede basicamente o desempenho da ULA, analisando apenas quantas instruções complexas são executadas.
- Tempo de acesso: está relacionado à velocidade de cada componente e a do barramento que interliga o processador à memória, tratando o tempo gasto entre o instante em que foi realizada uma solicitação e o instante em que o sistema entregou a resposta.

Programação de processador

A única linguagem que o processador trabalha é a de máquina, baseada em linguagem binária, que é utilizada para a codificação do conjunto de instruções de um computador. Para facilitar a tarefa de programação e de depuração são utilizados mnemônicos, que são associados aos códigos das instruções, nomes aos operandos e rótulos (labels) às posições ocupadas pelo programa, a chamada linguagem simbólica. (WEBER, 2004)

Nesse processo, é utilizado o montador, pois o programa escrito em linguagem simbólica precisa ser traduzido em linguagem de máquina para que possa ser executado, passando por um processo conhecido como montagem.

Resumo

Ao final da presente unidade vimos que:

- Os programas a serem executados ficam armazenados na memória principal (RAM) de onde serão acessados pelo processador.
- Os dados necessários para a execução de um programa são armazenados, a partir da memória principal, em registradores (memória temporária interna ao processador), assim como o resultado do processamento.
- A transferência de informações envolvendo processador e memória, seja na leitura ou na escrita, ocorre através de elementos como registradores (REM e RDM) e barramentos (de dados, de endereços e de controle).
- O processador trabalha com dois clocks, um para realizar operações internamente e outro para se comunicar com demais componentes do computador (mais lentos), sendo que existem técnicas de compensação como o uso de memória cache e o envio de mais informações por ciclo de clock.
- Internamente, o processador é dividido nos blocos de unidade operacional (composta por ULA e acumulador), de banco de registradores (formada por PC, RI e RST) e da unidade de controle, bem como o clock, o decodificador de instruções e barramentos internos.
- O conjunto de instruções de um processador, que determina o que ele pode fazer (geralmente operações sobre operandos), pode ser dividido em três tipos de instruções: de transferência de dados, aritméticas e lógicas e as de teste e desvio.
- A sequência para a realização de uma instrução pelo processador é conhecida como ciclo de “busca – decodificação – execução” de instruções. Nela os elementos dos blocos básicos são ativados para que ocorra o processamento;
- A técnica de pipeline é praticada no processador visando dividir a execução de uma instrução em estágios de processamento diferentes, sendo que cada um deles é responsável pela execução de uma parte da instrução, ocorrendo ao mesmo tempo e possuem o seu próprio bloco de controle.
- As formas existentes de medir o desempenho de processadores, podem ser destacadas como MIPS, a FLOPS e o tempo de acesso, dentre outras.

Questões de revisão

- a) Qual é a relação existente entre o processador e os programas a serem executados? Que elementos são utilizados e como eles se relacionam na transferência de valores (na leitura e na escrita) entre a memória RAM e o processador?
- b) Comente a diferença entre clock interno e clock externo. Por que existem dois clocks no processador? Que técnicas são utilizadas para minimizar a diferença entre eles?
- c) Em que blocos pode ser dividida a arquitetura de um processador? Comente sobre as características de cada bloco.
- d) Conceitue conjunto de instruções, instrução, programa e operandos.
- e) A sequência da execução de uma instrução envolve que passos? Como funciona cada um dos passos da sequência do ciclo de instruções?
- f) Que medidas de desempenho podem ser utilizadas para um processador?
- g) Qual é a finalidade de uso e como funciona o pipeline em um processador?
- h) Que técnicas são utilizadas para compensar a diferença entre estágios de um pipeline?
- i) Como é realizada a programação de um processador?