

Capítulo 1

Introdução ao Estudo dos Computadores

A execução de instruções é um dos pontos centrais num sistema computacional. Vários aspectos na definição e implementação da arquitetura de um computador são influenciados pelas características de cada uma das instruções.

Este capítulo discute alguns conceitos básicos envolvidos no funcionamento de um computador eletrônico digital. Um dos pontos que dificulta a compreensão do seu funcionamento é justamente o fato deles serem eletrônicos. Grande parte do trabalho destes sistemas é realizado por sinais elétricos e por componentes eletrônicos. O usuário comum, em geral, não está familiarizado em este tipo de operação.

1.1 Introdução

Os computadores são máquinas que **podem** ser utilizadas eficientemente na solução de **alguns** tipos de problemas.

A decisão de usar ou não o computador na solução de um problema depende da necessidade do usuário. Em geral, essa decisão é baseada na complexidade do problema a ser resolvido. Por exemplo, suponha que desejamos calcular uma equação algébrica do vigésimo grau. Sua solução manual é quase impossível devido a grande quantidade de cálculos necessários à obtenção da solução numérica.

Outro aspecto relevante nesta decisão é o volume de dados a serem processados. As aplicações que exigem a manipulação de uma grande quantidade de informações se beneficiam dos sistemas computacionais.

É interessante observar que podemos identificar quatro tipos de problemas se considerarmos a sua complexidade e a quantidade de informações que devem ser manipuladas, que são os problemas de:

- a) natureza **simples** e com **baixo** volume de dados;
- b) natureza **simples** e com **grande** volume de dados;
- c) natureza **complexa** e com **baixo** volume de dados;
- d) natureza **complexa** e com **grande** volume de dados;

Os problemas de natureza simples e com pequena quantidade de informações não precisam necessariamente do uso do computador para serem solucionados. Entretanto, os demais devem necessariamente utilizá-lo na obtenção de uma solução em um intervalo de

tempo viável. Assim, os computadores são máquinas que podem ser utilizadas eficientemente na solução dos tipos de problemas que possuem uma grande complexidade ou um grande volume de dados.

Para que um problema possa ser resolvido por computador é necessário criar um **algoritmo computacional**. Um algoritmo computacional é composto por uma seqüência de passos (ações) que expressam a solução do problema. As principais fases de um algoritmo para a obtenção de uma solução computacional são:

- a) elaboração do **algoritmo computacional** referente ao problema;
- b) codificação do algoritmo computacional em uma linguagem de alto nível (**programa fonte**);
- c) tradução do programa fonte para o código correspondente em linguagem de máquina (**programa objeto**);
- d) **execução** do programa objeto pelo computador.

A tradução de um programa fonte para um programa objeto, pode ser realizada de duas maneiras: **interpretando** ou **compilando** o programa fonte. Na verdade, a interpretação não cria um programa objeto. Cada instrução expressa em linguagem de alto nível é interpretada e executada diretamente pelo hardware do computador. No processo de compilação um programa fonte, composto por instruções de alto nível, é transformado em **instruções básicas** de um processador. Essas instruções básicas é que são executadas diretamente pelos circuitos eletrônicos do computador.

Exercício 1 - Elabore, codifique, em uma linguagem de programação à sua escolha, e execute pelo menos um dos seguintes algoritmos:

- (a) Encontrar as raízes reais de uma equação do segundo grau. Executar o programa com três diferentes valores de entrada.
- (b) Listar os 200 primeiros números primos.
- (c) Mostrar a transposta uma matriz quadrada de dimensão N.

1.2 Conceitos Básicos

Organização Típica de um Computador

A função de um sistema de computador é atender as solicitações de serviço dos usuários para terem seu problema resolvido. Qualquer sistema de computador possui três componentes básicos: uma unidade central de processamento (CPU ou **processador**), um sub-sistema de **memória** e um sub-sistema de **entrada e saída** de dados. Estes componentes estão interligados através de um barramento.

A função do processador é controlar todo o sistema e realizar todas as operações sobre os dados, de acordo com o indicado pelas instruções no código do programa. A memória principal armazena as instruções que são executadas pelo processador, e os dados que serão manipulados. As interfaces de entrada e de saída são as portas de comunicação para o mundo externo, às quais estão conectados os dispositivos periféricos tais como vídeo, teclado, discos e impressora. É através do barramento que o processador se comunica com a memória principal e as interfaces de entrada/saída.

A Figura 1.1 mostra a organização típica de um computador pessoal ou estação de trabalho que fazem parte do seu cotidiano.

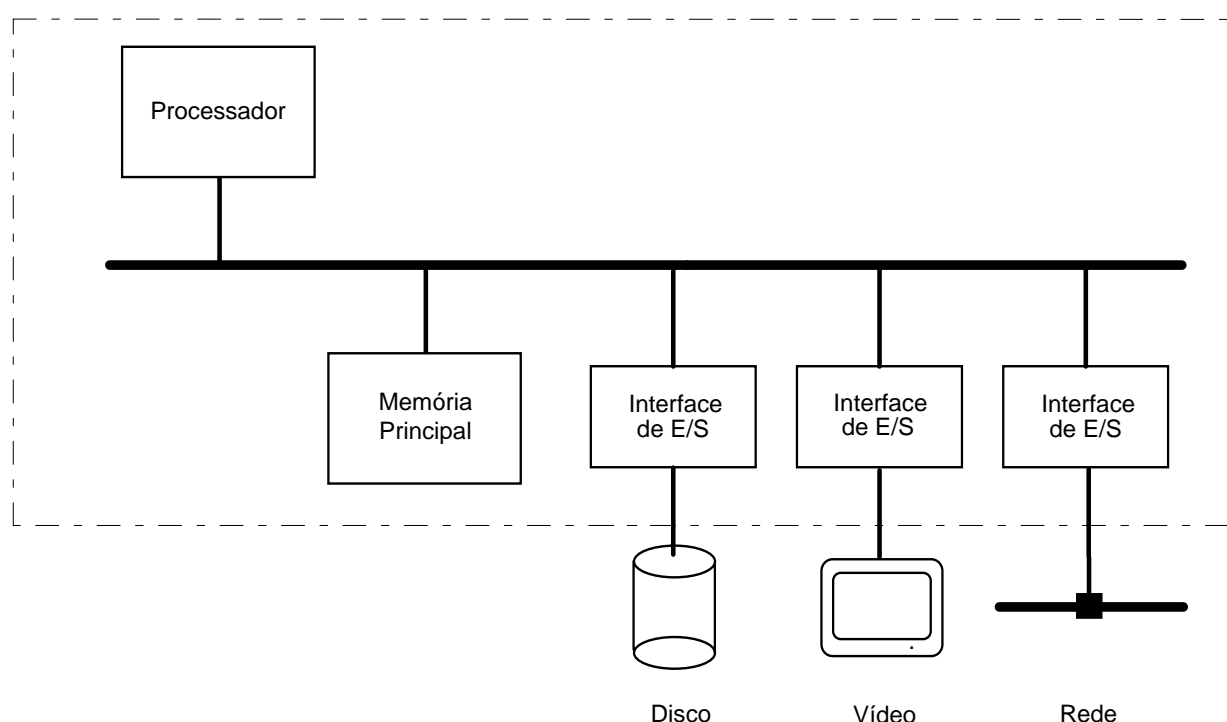


Figura 1.1. Organização típica de um computador.

Inicialmente é necessário definir como os problemas podem ser apresentados ao sistema. Isto é, qual a linguagem de comunicação entre os usuários e o sistema computacional. Definida uma linguagem comum, bastaria entregar na **entrada** do sistema a descrição do problema, e aguardar na sua **saída** o resultado do pedido.

Neste caso precisamos de um sistema capaz de manipular informações em todas as áreas de atuação dos usuários. O sistema receberá um determinado problema e deverá providenciar a sua solução. Note que o sistema deve estar preparado para solucionar uma diversidade enorme de problemas.

Contexto de Software

Em geral, é através das linguagens de alto nível que os usuários costumam especificar os problemas que devem ser solucionados. Como exemplo de linguagens de alto nível podemos citar a linguagem C, Pascal, Fortran entre outras. A Figura 1.2 situa o conjunto de instruções do processador dentro dos diversos níveis de *software* existentes em um sistema de computação.

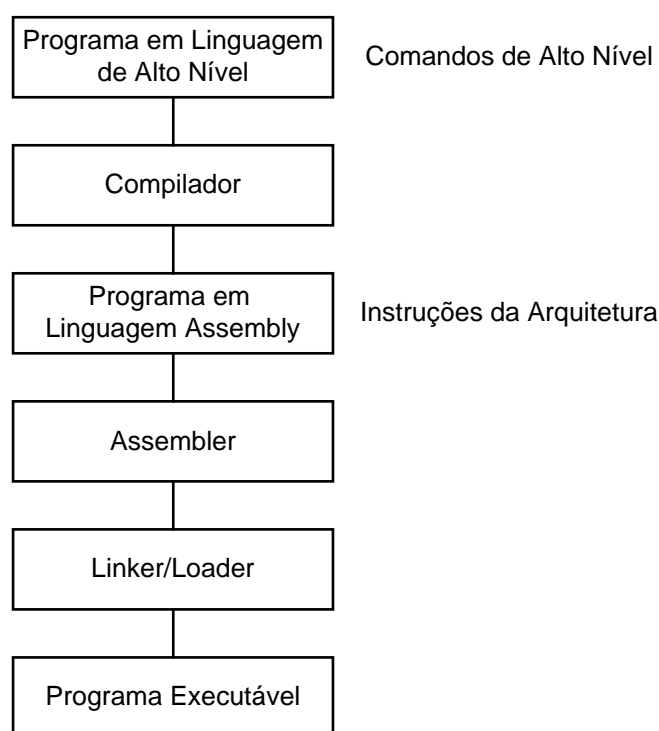


Figura 1.2. O conjunto de instruções dentro do contexto de software.

O compilador traduz o programa de alto nível em uma seqüência de **instruções de processador**. O resultado desta tradução é o programa em **linguagem de montagem** ou **linguagem de máquina** (*assembly language*). A linguagem de montagem é uma forma de representar textualmente as instruções oferecidas pela arquitetura do processador. Cada arquitetura possui uma linguagem de montagem particular. No programa em linguagem de montagem, as instruções são representadas através de mnemônicos, que associam o nome da instrução à sua função, como por exemplo, ADD ou SUB, isto é soma e subtração, respectivamente.

O programa em linguagem de montagem é convertido para um programa em **código objeto** pelo **montador** (*assembler*). O montador traduz diretamente uma instrução da forma

textual para a forma de código binário. É sob a forma binária que a instrução é carregada na memória e interpretada pelo processador.

Os programas complexos são normalmente estruturados em módulos. Cada módulo é compilado separadamente e submetido ao montador, gerando diversos módulos em código objeto. Estes módulos são reunidos pelo **ligador** (*linker*), resultando finalmente no programa executável que é carregado na memória.

A comunicação entre um sistema de computador e os usuários deve ser realizada através de uma linguagem simples. Isto é, a comunicação deve ser composta de pequenos passos, cada um deles bem fácil de ser executado.

Assim, o problema submetido ao sistema deve, então, conter uma série de passos. Dessa forma, quando os passos forem executados na seqüência eles podem produzir o resultado desejado. O programa do usuário é transformado para uma linguagem de baixo nível, adequada ao processador, para que possa ser executado e produzir os mesmos resultados do programa escrito em linguagem de alto nível.

Para que um computador possa executar um programa é necessário que esse programa esteja armazenado na memória. A memória é um dispositivo eletrônico capaz de armazenar instruções e dados. Ela pode receber e armazenar informações e fornecê-las quando solicitada.

Um programa escrito em linguagem de baixo nível, então, é composto de um conjunto de **instruções de processador** muito simples de serem executadas. Cada uma destas instruções é composta necessariamente por um **código de operação**. Esse código de operação identifica de maneira inequívoca a operação a ser realizada, e, dependendo da instrução pode necessitar de **operandos**. Em resumo, podemos dizer que o código de operação de uma instrução especifica a ação que deve ser realizada quando esta instrução for executada. Já o operando ou os operandos são elementos que complementam a ação.

No computador os operandos podem ser referenciados por seu **valor** ou por seu **endereço**. No caso da referência ao operando ser realizada por valor, a própria instrução contém um número que é interpretado como sendo o valor numérico do operando. No caso da referência ser realizada por endereço, a instrução contém um número que é interpretado como o endereço da locação de memória que o operando ocupa.

Tema para reflexão

Discuta as diferenças entre **endereço** e **conteúdo de endereço da memória**.

Passos de Execução de uma Instrução

Após o programa ser armazenado na memória, em uma série de endereços consecutivos, podemos iniciar o procedimento de solução do problema. O primeiro endereço de um programa contém necessariamente uma instrução de processador. Para realizar o processamento, propriamente dito, é necessário **buscar** a instrução que estiver armazenada no primeiro endereço de memória onde se encontra o programa. Em seguida, essa instrução é **decodificada**. Em outras palavras, o processador define o código de operação daquela instrução em particular. O passo seguinte é o de **execução** da instrução, seguido de um outro passo para armazenar o **resultado**, caso seja necessário.

Tema para reflexão

Porque existem diferentes instruções de soma ?

Resumindo podemos afirmar que a execução de uma instrução é realizada com quatro passos básicos: **busca**; **decodificação**; **execução** e **resultado**. Um programa é executado quando colocamos em *loop* esses quatro passos básicos.

Exercício 2 - Defina:

- (a) Instrução de processador;
- (b) Operando referenciado por valor;
- (c) Operando referenciado por endereço;
- (d) Programa armazenado;
- (e) Ciclo de execução de instruções.

1.3 Arquitetura de Von Neumann

A Figura 1.3 mostra o modelo de computador proposto em 1940 por **John von Neumann**, um matemático húngaro, naturalizado americano. Sua contribuição para o desenvolvimento da Ciência da Computação foi significativa, a ponto de tornar seu nome sinônimo da arquitetura de um computador uniprocessador (**máquina de von Neumann**).

O modelo de von Neumann ainda representa a base dos projetos das máquinas atuais. Na Figura 1.3 podemos identificar os componentes responsáveis pelo processamento, armazenamento e entrada e saída das informações. A diferença entre os computadores antigos e os atuais está na sua tecnologia de fabricação. Considerando o desenvolvimento da tecnologia de implementação dos circuitos eletrônicos podemos dividir os computadores em gerações. As máquinas da primeira geração eram fabricadas com válvulas, as da segunda com

transistores, as da terceira com circuitos integrados de densidade média e as da quarta com circuitos integrados de alta densidade.

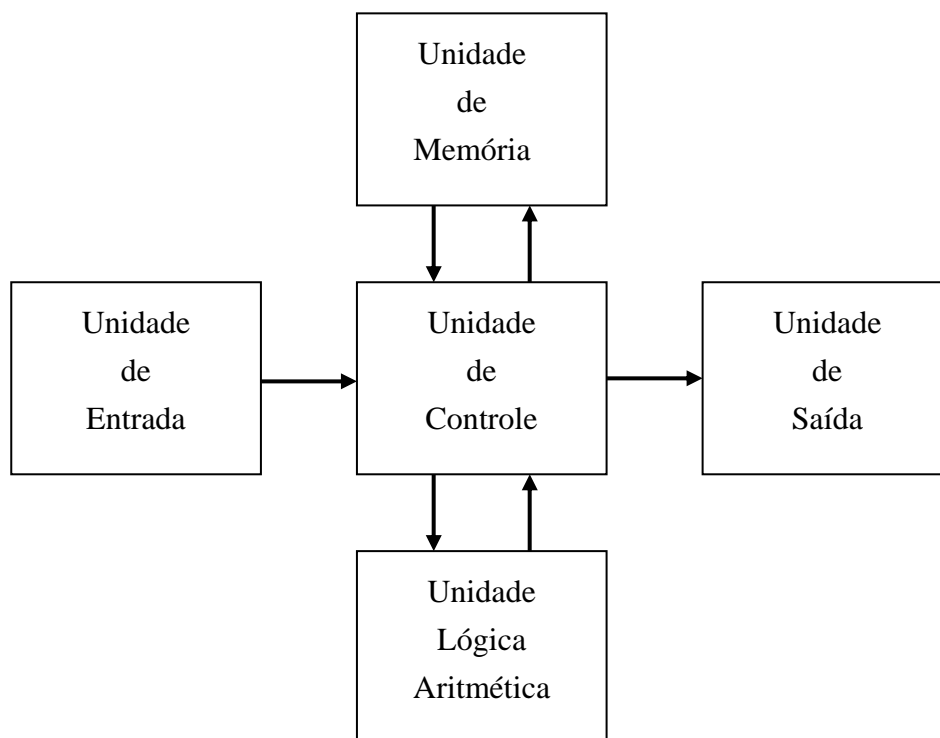


Figura 1.3. *O modelo de von Neumann.*

1.3 Formas de Representação da Informação

Os computadores eletrônicos digitais são assim denominados porque são equipamentos construídos a partir de circuitos eletrônicos. Esses computadores são capazes de realizar cálculos, operações lógicas e movimentação de dados entre seus dispositivos de armazenamento.

Os sistemas digitais, em seu nível mais baixo, representam as informações através de dígitos. Num nível mais alto, uma codificação com estes dígitos formam diferentes combinações que podem representar qualquer tipo de informação.

Os **sistemas digitais** são constituídos por uma combinação de dispositivos que manipulam informação representada por dígitos. Na maioria das vezes as informações são representadas internamente por sinais elétricos binários (5 Volts = 1 e 0 Volts = 0). Isto é, eles só assumem dois valores associados a 0 e 1. Incluído nesta categoria, o computador digital é um **sistema digital binário**, pois a informação é representada nele através dos dígitos binários.

Em resumo, um computador eletrônico digital processa informações que só podem assumir estados discretos. O termo digital significa que a informação processada neste tipo de computador só pode ser representada por um número fixo de valores. Adicionalmente, o fato do número de valores disponíveis para representar a informação ser igual a dois, faz com que o sistema digital seja denominado de sistema digital binário.

Exercício 3 - Mostre como se codifica os números decimais em números binários.

1.4 Outros Conceitos Básicos

Como citado anteriormente, podemos dizer que um sistema de computador é composto por três componentes básicos: o processador, o sub-sistema de memória e o sub-sistema de entrada e saída. Além disso, podemos dizer que os quatro passos básicos que compõem a execução de uma instrução são: busca, decodificação, execução e resultado. Esses quatro passos são repetidos, formando ciclos de execução de instruções, desde a primeira até a última instrução do programa. Todas as atividades realizadas pelo computador são executadas em tempos bem definidos.

A seguir definimos resumidamente alguns conceitos importantes para o nosso entendimento, e que são apresentados mais detalhadamente durante o curso.

Sinal de *clock*

Para coordenar as atividades entre os componentes básicos que compõem o sistema de um computador existe um componente eletrônico que gera um **sinal de *clock***. Mais especificamente gera um sinal periódico que alterna entre tensões altas e baixas (0s e 1s). A velocidade do *clock* é medida em **hertz ou ciclos por segundo**. Um sinal de 1 Hz alterna valores altos e baixos uma vez em cada segundo. Já um sinal de 1 MHz alterna esses valores um milhão de vezes por segundo. Diz-se, portanto, que a **frequência** destes sinais é igual a 1 Hz e 1 MHz, respectivamente. O tempo entre duas repetições sucessivas do *clock* é denominado **período do *clock***, sendo o período o inverso da frequência. A Figura 1.4 mostra um ciclo de *clock*.

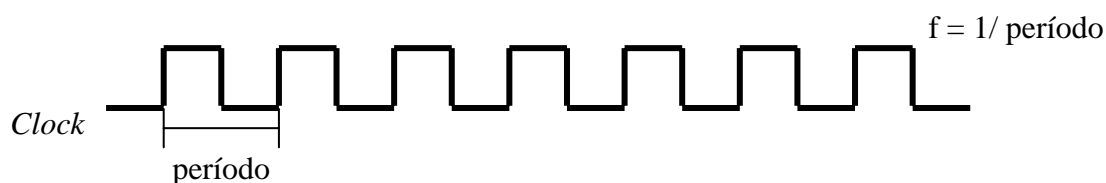


Figura 1.4. O Ciclo de *clock*.

O período do sinal de *clock* é a menor unidade de tempo perceptível por um sistema digital. Em outras palavras, em sistemas digitais, todas as ações ocorrem em intervalos de

tempo que são múltiplos inteiros do período do *clock* da máquina. Portanto, o ciclo de execução de instruções é realizado em um tempo que é um múltiplo inteiro do período do *clock* da máquina.

Os processadores modernos conseguem executar milhões de instruções em um segundo. A frequência do sinal de *clock* gira em torno de 800 MHz nas máquinas com um custo acessível à maioria dos usuários. Embora já estejam disponíveis processadores com mais de 1 GHz, porém com custo ainda elevado para a maioria dos usuários comuns.

Organização Genérica de um Processador

Uma organização seja de um processador, ou sistema de computador, deve ser definida através dos seus componentes, da funcionalidade de cada um desses componentes e como eles interagem entre si.

A execução de uma instrução típica, tal como a soma de dois valores, requerer um conjunto de ações separadas (busca na memória, decodificação, uso da unidade lógica aritmética, armazenamento), cada uma delas ocorrendo em um período do *clock* do sistema. A Figura 1.5 mostra uma organização genérica de um processador.

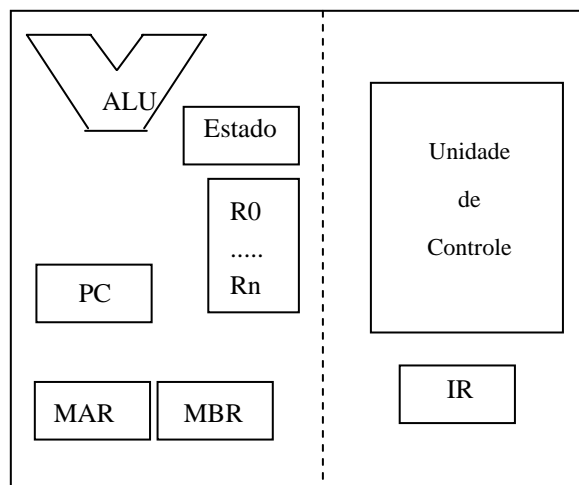


Figura 1.5. Organização genérica de um processador.

Um processador pode ser visto como possuindo duas seções: a **seção de controle** e a **seção de processamento**. É na seção de processamento que as informações são manipuladas. A seção de controle é responsável por enviar sinais à seção de processamento para controlar cada um de seus componentes.

A seção de controle é composta por uma **unidade de controle** e um **registrador de instruções**. A seção de processamento é composta por diversos **registradores** e uma **ALU** (*Arithmetic Logic Unit*).

O registradores podem ser classificados como registradores de **propósito geral** (R0 a Rn), **propósito específico** (estado, PC – *program counter*, IR – *instruction register*) e **temporários** ou **auxiliares** (MAR – *Memory Address Register*, MBR – *Memory Buffer Register*).

A ALU é responsável por manipular todas as informações, isto é, realizar as operações lógicas e aritméticas propriamente ditas. Associado a ela existe um **registrador de estado** para armazenar as informações sobre os resultados produzidos pela ALU. O registrador PC armazena o endereço de memória da próxima instrução a ser executada. Os registradores MAR e MBR são responsáveis por realizar a transferência de informações entre o processador entre a memória.

As instruções são armazenadas na memória, da mesma maneira que os dados, e recuperadas uma a uma no momento de sua execução. Vale observar que os dados necessários à execução de tais instruções também estão na memória, sendo recuperados exatamente da mesma maneira que as instruções. A única diferença é que a instrução quando vem da memória é armazenada no registrador MBR e depois transferida para o IR. O dado, que também é armazenado no MBR, depois é transferido para um registrador de propósito geral. As instruções permanecem no IR enquanto são executadas.

As instruções, os dados e os endereços são sempre formados por cadeias de 0's e 1's. Em última análise eles formam uma cadeia binária. Uma determinada cadeia binária armazenada na memória de um processador pode ser interpretada de várias formas diferentes. A Tabela 1.1 mostra como a cadeia de *bits* 10100000 pode ser interpretada de diferentes maneiras pelo processador 8086 da Intel.

Conteúdo da Memória	Significado	Código
10100000	A	Caracter EBCDIC
10100000	160	Número inteiro
10100000	á	Caracter ASCII-8
10100000	MOVE	Instrução Intel 8086

Tabela 1.1. Exemplos de significados de cadeias binárias.

O valor binário 10100000 pode ser interpretado pelo processador 8086 como um caracter EBCDIC, um número inteiro, um caracter ASCII, ou como uma instrução de processador.

O processador é capaz de interpretar a cadeia binária armazenada em determinado endereço de memória, a partir do endereço da primeira instrução do programa a ser executado. Todo processador é projetado de modo a carregar no PC (*Program Counter*) um determinado valor, sempre que a máquina for ligada. Este valor corresponde ao endereço de memória onde está armazenada a primeira instrução de um programa de inicialização do sistema. Tão logo este endereço tenha sido carregado no PC, o processador inicia a busca da cadeia de *bits* armazenada no endereço inicial, para em seguida transferi-la para o IR. Enquanto o processador estiver aguardando que a memória envie a cadeia de *bits* que representa a instrução para o MBR, ele atualiza o valor do PC. Esta atualização faz com que o PC aponte para a próxima instrução a ser executada.

Para se ter uma idéia de como uma cadeia binária pode ser aumentada para também aumentar a funcionalidade de um conjunto de instruções citamos, a seguir, a quantidade de instruções de alguns dos processadores antigos da família da Intel. Os processadores 8088 e 8086 interpretam em torno de 120 instruções. Isto significa que o decodificador associado ao IR reconhece 120 cadeias binárias diferentes, transformando cada uma delas em um conjunto de ações discretas. Cada cadeia de *bits* é responsável pela execução de uma instrução. O 80286 acrescenta 23 instruções ao conjunto de instruções do 8086, executando, portanto, 143 instruções. Já o 80386 executa em torno de 200 instruções diferentes.

Barramento

Define-se barramento como um caminho elétrico estabelecido entre dois ou mais componentes de um sistema de computador. Os sistemas possuem três barramentos: o de **dados**, o de **endereços** e o de **controle**. Cada um destes barramentos é formado por um número de fios diferentes. Além disso, os barramentos podem ser mantidos em três estados que correspondem ao nível lógico 1, ao nível lógico 0, e um estado neutro. O estado neutro não representa nenhum dos dois níveis lógicos mencionados. Ele pode ser interpretado como se o componente ligado a ele estivesse desconectado do sistema. A seguir apresentamos uma descrição de cada um dos três barramentos envolvidos na interligação dos componentes de um sistema de computador.

Barramento de dados: É um barramento bidirecional, onde circulam as informações que trafegam entre o processador e a memória principal. Neste barramento circulam tanto as instruções, sempre no sentido da memória para o processador, quanto os dados, estes em ambos os sentidos. O registrador MBR está diretamente ligado a este barramento. Isto é, as informações que são transferidas entre o processador e a memória passam necessariamente pelo MBR. O tamanho deste barramento indica o tamanho da palavra (a quantidade máxima de *bits*) num acesso à memória.

Barramento de endereços: É um barramento unidirecional, no sentido do processador para a memória ou para as interfaces de entrada e saída. Por ele circulam os endereços das informações nos acessos a esses componentes. O registrador MAR armazena o endereço das informações a serem acessadas e está diretamente ligado ao barramento de endereços. O tamanho do MAR está associado ao tamanho da memória. Para uma memória com 2^N endereços o MAR possui N bits.

Barramento de controle: É um barramento bidirecional, por onde trafegam os sinais necessários ao controle da troca de informações entre o processador e os demais componentes do sistema. Como exemplo de sinais de controle podemos citar os que controlam a leitura, a escrita ou a espera por um dispositivo.

Além das memórias, também estão ligados aos barramentos um conjunto de interfaces de entrada e saída. Cada interface de entrada e saída possui conectado a ela um dispositivo de entrada e saída. Essas interfaces tornam transparente para o processador os detalhes físicos dos dispositivos de entrada e saída do computador. Em um computador existem vários dispositivos periféricos como o teclado, o vídeo, o disco, a impressora entre outros.

Unidades Computacionais Básicas

Quando nos referimos à capacidade de processamento do processador, a capacidade de armazenamento da memória ou a transferência de informações entre dispositivos, costumamos utilizar diferentes unidades de medidas e seus múltiplos. A seguir estão exemplificadas, na Tabela 1.2, as unidades mais comuns usadas quando nos referimos a estes componentes.

Capacidade de Processamento (segundos)	Capacidade de Armazenamento (bytes)	Transferência de Informação (ordem de grandeza)
um Mili = 10^{-3} (ms)	um Kilo = 2^{10}	Tempo de acesso a disco: ms
um Micro = 10^{-6} (μ s)	um Mega = 2^{20}	Tempo de acesso a memória: ns
um Nano = 10^{-9} (ns)	um Giga = 2^{30}	Ciclo de máquina: μ s / ns
um Pico = 10^{-12} (ps)	um Tera = 2^{40}	Execução de instruções: μ s / ns

Tabela 1.2. Exemplos de unidades de medidas utilizadas em sistemas computacionais.

Bibliografia

- *Computer Organization and Design: The Hardware/Software Interface*
David Patterson and John Hennessy, Morgan Kaufmann Publishers, Inc.
- *Organização e Projeto de Computadores: A Interface de Hardware e Software*
David Patterson and John Hennessy – LTC Editora.