Unidade: Arquitetura de computadores e conversão entre bases numéricas



# Unidade: Arquitetura de computadores e conversão entre bases numéricas

## 1 - Conceitos básicos sobre arquitetura de computadores

O conceito de arquitetura de computador refere-se aos diversos componentes do computador a forma como se interligam, se comunicam e como se coordena toda esta organização.

Os computadores, cada vez mais, vêm fazendo parte de nossas vidas, em quase todas as atividades usamos computadores de forma implícita ou explicitamente. Em conjunto com a popularidade do computador percebemos que as necessidades dos usuários aumentam a cada dia. Muitas evoluções foram feitas e estão sendo feitas para melhorar o desempenho dos computadores, algumas delas são: o crescente desenvolvimento da CPU, a capacidade de memória, a capacidade do disco rígido, a existência de memória cache, a utilização de técnicas como DMA¹, são algumas evoluções feitas com o objetivo de satisfazer as necessidades dos usuários com respostas mais rápidas. A definição destes parâmetros e a forma como os diversos componentes do computador são organizados, definem o que é conhecido por arquitetura de computador e vai determinar aspectos relacionados à qualidade, desempenho e a aplicação para o qual o computador vai ser usado.

Lembre-se que um computador digital é uma máquina que pode resolver problemas executando um conjunto limitado de instruções e, estas instruções estão estritamente relacionadas ao microprocessador, um programa escrito em linguagem de alto nível como Java, C++, .NET, Delphi e outras, quando executadas são convertidas em instruções que o microprocessador pode entender. Isto significa que qualquer programa é convertido a uma dessas instruções limitadas do microprocessador, as instruções primitivas de um computador formam uma linguagem com a qual as pessoas, desde que tenham conhecimento, podem se comunicar, essa linguagem é denominada linguagem de máquina.

1

<sup>&</sup>lt;sup>1</sup> DMA: Acesso Direto a Memória é uma técnica utilizada onde o microprocessador passa a responsabilidade de uma determinada tarefa para um dispositivo específico, desta forma o microprocessador fica disponível para executar outras tarefas em paralelo.



Um diagrama básico de um computador pode ser composto de três elementos: dispositivos de entrada e saída, unidade central de processamento e memória.

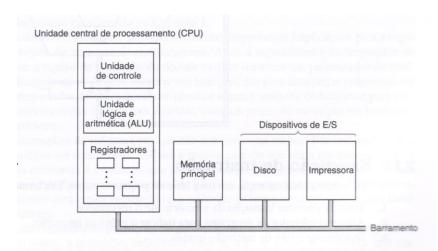


Figura 1: Diagrama básico de um computador: fonte Tanenbaum

A unidade central de processamento é o cérebro do computador. Sua função e executar programas armazenados na memória principal (RAM²) buscando suas instruções, examinando-as e executando-as uma após a outra.

Os componentes são conectados por um barramento, que é um conjunto de fios paralelos que transmitem endereços, dados e sinais de controles. Os barramentos podem ser externos à CPU, neste caso ele serve para conectar a memória e os dispositivos de entrada e saída, mas também podem ser internos a CPU, ou seja, dentro do chip da CPU há barramentos que são usados para o instruções tráfego de ou dados capturados na memória. Conforme pode ser observado na figura 1, a CPU é composta por várias partes distintas. Uma delas é a unidade de controle que é responsável por endereçar, ou seja, apontar para uma posição da memória onde estará a informação a ser carregada para o processador e controlar se estas informações serão lidas ou escritas na memória principal. A outra é a unidade lógica e aritmética que efetua operações básicas como adição, subtração, multiplicação, divisão e comparação dos dados.

Conforme pode ser verificado na figura 2 a unidade lógica e aritmética recebe duas informações, uma de cada registrador, estes dados serão

2

<sup>&</sup>lt;sup>2</sup> RAM – Memória de Acesso Aleatório é uma memória volátil, ou seja, quando desligamos o computador as informações contidas nelas são apagadas.



processados pela unidade lógica e aritmética que retornará um resultado e irá inseri-lo em um dos registradores, portanto você pode perceber que os dados vêm dos registradores, passam pela unidade lógica e aritmética e um resultado é obtido e enviado para um dos registradores. Basicamente é assim que funciona a unidade lógica e aritmética, os dados são enviados da memória principal para os registradores e o resultado da operação pode ser inserido em dos registradores ou então voltar direto para a memória principal.

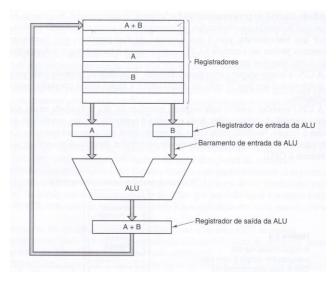


Figura 2: Unidade Lógica e Aritmética: fonte Tanenbaum

Os microprocessadores atuais contêm uma pequena memória de alta velocidade usada para armazenar resultados temporários e controlar informações. Essa memória, chamada registrador, é composta de uma pequena quantidade de espaço, normalmente, todos os registradores têm o mesmo tamanho. Cada registrador tem uma função específica e podem ser lidos e escritos em altas velocidades, pois são internos a CPU e usam o seu barramento interno.

Nota: Registradores são memórias, portanto sua função e armazenar dados temporariamente, por estar dentro do microprocessador e pelo material utilizado na sua fabricação, os registradores são memórias de altíssimas velocidades, eles têm uma função importante, pois o microprocessador usa as informações contidas nos registradores para processamento.

A memória principal é um grande vetor, onde cada endereço de memória é um índice desse vetor. Se seu computador possui 640 K de memória, então haverá disponível 65.536 posições (para se chegar a este número, basta



multiplicar 640 \* K<sup>3</sup>). É na memória principal que estão os dados e instruções que serão interpretados pela CPU, as informações contidas na memória poderão ser lidas de uma forma seqüencial ou alternadas tomando como referência o índice conhecido também como endereço

Os dados armazenados na memória estão no formato de bits, ou seja, armazenam apenas 0's e 1's, este procedimento é usado por ser mais confiável para codificar informações digitais e por propiciar maiores velocidades de processamento. Um endereço de memória, basicamente, contém oito bits, a este conjunto de 8 bits denominamos um byte.

A CPU, de alguma maneira, tem que obter os dados armazenados na memória para serem processados, esses dados são adquiridos pela informação do endereço de onde eles estão armazenados. Uma vez endereçado, os dados são disponibilizados pela memória e são direcionados para a CPU. É importante ressaltar, que estes dados são conjuntos de 8 bits contendo zeros e uns armazenados.

# 2 - Arquitetura de computador de Von Neumann

A máquina proposta por Von Neumann reúne os seguintes componentes: uma memória, uma unidade aritmética e lógica, uma central de processamento composta por diversos registradores e uma unidade de controle que tem a função de auxiliar na busca de um programa na memória instrução por instrução.

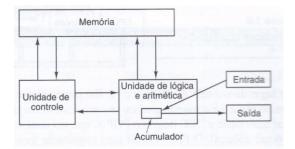


Figura 3: Arquitetura de Von Neumann: fonte Tanenbaum

1

 $<sup>^3</sup>$  Em informática, a unidade de medida K corresponde a 1024 e não a 1000, isto porque uma palavra em informática é representada em binário (0 ou 1) e a notação é feita na base 2, portanto  $2^{10} = 1024$ .



Todas as máquinas construídas até o início do projeto do EDVAC eram programadas através de meios externos como: cartões perfurados, fitas perfuradas, painéis, cabos de conexões etc, dispunham de pouquíssima memória para armazenar os dados e os resultados intermediários dos cálculos.

A construção do EDVAC, cuja capacidade de armazenamento era de vinte números, consumiu mais de 17.000 válvulas eletrônicas. Um fator decisivo para viabilizar o projeto de uma nova máquina foi a idéia de utilizar técnicas para implementar elementos de memória de custo mais baixo do que utilizando válvulas. Com essa idéia, tornou-se possível pensar numa máquina com dezenas ou centenas de milhares de bits. Von Neumann foi um dos responsáveis para que as pesquisas em memórias fossem desenvolvidas, pois sua idéia era que os programas fossem armazenados diretamente nos computadores e não mais utilizar os meios descritos acima para armazená-los. Com sua contribuição os computadores e os programas tornaram-se mais rápidos, mais flexíveis, e mais eficientes.

# 3 - Sistemas de numeração

O homem através dos tempos sentiu a necessidade da utilização de sistemas numéricos. Existem vários sistemas numéricos, dentre os quais se destacam o sistema decimal, o binário, o octal e o hexadecimal.

Os sistemas, binário, octal e hexadecimal são muito importantes na área de técnicas digitais e computação, pois são usados na representação dos sinais digitai, na programação a baixo nível e na representação de endereçamento de memória e dispositivos de entrada e saída.

O sistema binário de numeração é um sistema no qual existem apenas dois algarismos, o algarismo 1 e o algarismo zero. Podemos representar qualquer número decimal em binário usando apenas estes dois algarismos. Para representar o número zero em binário usamos o algarismo 0 e para representar o número 1 em binário usamos o algarismo 1. Pois bem, se temos apenas estes dois algarismos, como poderíamos representar o algarismo 2 em binário? Você já parou para pensar nisto? Não possuímos o algarismo 2 nesse sistema de numeração, como faríamos então já que em binário só temos dois



símbolos (0 e 1)?

Não é tão complicado quanto parece. No sistema decimal, nós não possuímos o algarismo dez e nós apresentamos a quantidade de uma dezena utilizando-nos do algarismo 1 seguido do algarismo zero, ou seja, passamos a repetir os algarismos que já existe neste sistema de numeração.

No sistema binário, com um bit, conseguimos representar 2 dados (o 0 e o 1) e para representar outros valores maiores que 0 e 1, usamos a mesma regra feita para o sistema decimal, portanto para representar outros valores temos que começar a agrupar outros bits.

A unidade central de processamento, mais especificamente a CPU, trabalha apenas com zeros e uns, ou seja, trabalha com sistema binário para processar as informações armazenadas na memória, esta forma de trabalhar aumenta a capacidade de processamento e, conseqüentemente, aumenta a velocidade do processamento das informações. As linguagens de baixo nível, linguagem de máquina e assembler são linguagens onde as representações das instruções são feitas por números binários e hexadecimais.

O número hexadecimal tem como características:

- São muito compactos.
- É fácil a conversão de hexadecimal para binário e de binário para hexadecimal.

Em computação o sistema de numeração hexadecimal é usado em várias situações, por exemplo, quando calculamos a posição da cabeça leitora e os setores de um disco rígido ou então quando usamos programas de editor do disco rígido para analisar algumas características diferentes ou algum problema ocorrido, para estes casos necessitaremos de um bom conhecimento do sistema Hexadecimal para ajudar na análise.

O sistema de numeração hexadecimal usa a base 16 e inclui somente os dígitos 0 a 9 e as letras A, B, C, D, E, e F. Nós costumamos usar a letra H depois do número para denotar que ele está no sistema hexadecimal.

#### 3.1 - Método Genérico de transformação de números.

Para qualquer conversão de uma base X para uma base Y podemos usar o método de decomposição dos números, ou seja, separamos cada





número em sua representatividade (unidade, dezena, centena etc.) e então multiplicamos pela base em que o número se encontra elevado a sua representatividade (unidade, dezena, centena etc...), lembrando que o expoente para a unidade será 0, para dezena será 1 e assim por diante.

#### Exemplo:

Decompor o decimal 5324.

$$5324:10 = 532$$
, fica **4**  $10^0$   
 $532:10 = 53$ , fica **2**  $10^1$   
 $53:10 = 5$ , fica **3**  $10^2$   
 $5:10 = 0$ , fica **5**  $10^3$ 

Exemplo para número real.

Decompor o número 0,8125 (decimal)

$$0.8125 * 10 =$$
**8**,  $125 10^{-1}$   
 $0.125 * 10 =$ **1**,  $25 10^{-2}$   
 $0.25 * 10 =$ **2**,  $5 10^{-3}$   
 $0.5 * 10 =$ **5**,  $0 10^{-4}$ 

0,0

# 3.2 - Conversão de decimal para binário

#### Exemplo

Converter 29 (decimal) para binário – após o cálculo devemos chegar ao resultado 11101<sub>2</sub>

Para converter de uma base decimal para qualquer outra base, basta dividir, sucessivamente, o número decimal pela base que se quer converter guardando o resto da divisão, o resultado é novamente dividido pela base e devemos guardar o resto, este processo irá se repetir até que o resultado seja menor que a base. Exemplo.

$$29: 2 = 14$$
, resto **1**  $2^0$   
14: 2 = 7, resto **0**  $2^1$ 





7:2 = 3, resto 1 
$$2^2$$
  
3:2 = 1, resto 1  $2^3$ 

$$1:2 = 0$$
, resto  $1 2^4$ 

O resultado final é composto por todos aqueles restos da divisão que guardamos durante o processo, a leitura é feita de baixo para cima, portanto temos como resultado **11101**<sub>2</sub>.

#### 3.3 - Conversão de números fracionários decimais em binário

Para converter números fracionários do sistema decimal em números fracionários do sistema binário, devemos multiplicar o número fracionário decimal por dois, pois é nesta base que estamos querendo converter.

Exemplo

Converter 0,8125 (decimal) para binário, o resultado, após o cálculo, deverá ser **0,1101**<sub>2</sub>, veja abaixo como é feito o cálculo

$$0.8125 * 2 = 1.625 2^{-1}$$
  
 $0.625 * 2 = 1.25 2^{-2}$   
 $0.25 * 2 = 0.5 2^{-3}$   
 $0.5 * 2 = 1.0 2^{-4}$ 

0,0

Veja abaixo outro exemplo

Converter 0,3 (decimal) para binário, o resultado deverá ser 0,01001 1001 1001...  $_{2}$ 

$$0,3 * 2 = 0, 6$$
  $2^{-1}$   $0,6 * 2 = 1, 2$   $2^{-2}$   $0,2 * 2 = 0, 4$   $2^{-3}$   $0,4 * 2 = 0, 8$   $2^{-4}$   $0,8 * 2 = 1, 6$   $2^{-5}$   $0,6 * 2 = 1, 2$   $2^{-6}$ 

•



Se for continuar com esta multiplicação veremos que não haverá um fim, note que a seqüência 1001 começa a repetir, isto caracteriza uma dízima, ou seja, esta mesma seqüência de números sempre irá aparecer.

# 3.4 - Conversão de Decimal para Hexadecimal

Usando a mesma regra da conversão de decimal para binário, no entanto temos que usar a base adequada, ou seja, ao invés de dividir o decimal por dois, iremos dividi-lo por 16, pois esta é a base para a qual queremos converter o número em questão.

#### Exemplo

Converter 2540,34 (decimal) para hexadecimal o resultado deverá ser, aproximadamente **9EC**, **570A3**<sub>16</sub>. A conversão será feita em duas partes conforme mostrado abaixo.

a) Primeiro, resolveremos o lado esquerdo da vírgula, deste lado temos o número inteiro, portanto devemos dividi-lo por 16.

Portanto do lado esquerdo da vírgula teremos => 9EC 16

b) Segundo, do lado direito da vírgula, temos o número fracionário, portanto devemos multiplicá-lo por 16, veja abaixo como fica o cálculo.

$$0.34 * 16 = 5.44 5 \Rightarrow 5 16^{-1}$$
 $0.44 * 16 = 7.04 7 \Rightarrow 7 16^{-2}$ 
 $0.04 * 16 = 0.64 0 \Rightarrow 0 16^{-3}$ 
 $0.64 * 16 = 10.24 10 \Rightarrow A 16^{-4}$ 
 $0.24 * 16 = 3.84 3 \Rightarrow 3 16^{-5}$ 
...

...

continua multiplicando e teremos => 0, 570A3... 16

Agora temos que agrupar os dois resultados, o que calculamos para o lado esquerdo da vírgula e o que calculamos do lado direito da vírgula e temos



como resultado final ~9EC,570A3<sub>16</sub>

#### 3.5 - Convertendo de Binário para Decimal

Para fazer conversão do sistema binário para o sistema decimal devemos proceder da seguinte forma. Multiplicamos o primeiro número binário da direita para a esquerda por dois elevado a zero, o segundo número da direita para a esquerda multiplicamos por dois elevado a um e assim sucessivamente até que todos os números sejam multiplicados por dois e seu respectivo expoente. Os resultados destas multiplicações devem ser somados para obtermos o número decimal. Veja abaixo um exemplo.

Converter 110101 (binário) para número decimal

$$110101_{2} = 1*2^{5} + 1*2^{4} + 0*2^{3} + 1*2^{2} + 0*2^{1} + 1*2^{0}$$

$$= 1*2^{5} + 1*2^{4} + 1*2^{2} + 1*2^{0}$$

$$= 1*32 + 1*16 + 1*4 + 1*1$$

$$= 53_{10}$$

Exemplo

Converter -11,101 (binário) para número decimal.

$$-11,101_{2} = -(1*2^{1} + 1*2^{0} + 1*2^{-1} + 0*2^{-2} + 1*2^{-3})$$

$$= -(1*2^{1} + 1*2^{0} + 1*2^{-1} + 1*2^{-3})$$

$$= -(1*2 + 1*1 + 1*0,5 + 1*0,125)$$

$$= -3,625$$

#### 3.6 - Conversão de Hexadecimal para Decimal

Para converter do sistema de numeração hexadecimal para o sistema de numeração decimal usamos a mesma regra de conversão do sistema de numeração binário para o sistema de numeração decimal, no entanto temos que trocar a base.





# Exemplo

Converter o número hexadecimal 9EC,570A3 para decimal

9EC, 570A3 
$$_{16}$$
 = 9\*16<sup>2</sup> + 14\*16<sup>1</sup> + 12\*16<sup>0</sup> + 5\*16<sup>-1</sup> + 7\*16<sup>-2</sup> + 0\*16<sup>-3</sup> + 10\*16<sup>-4</sup> + 3\*16<sup>-5</sup> = 2304 + 224 + 12 + 0,3125+ 0,02734375 + 0,0001525878... + 0,00000286102...

#### **= 2540 , 33999919891357421875**

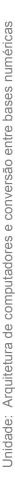
## 3.7 - Número binário e sua correspondência em hexadecimal e octal

Binario	Hex	Binário	Octal
0000	0	000	0
0001	1	001	1
0010	2	010	2
0011	3	011	3
0100	4	100	4
0101	5	101	5
0110	6	110	6
0111	7	111	7
1000	8		
1001	9		
1010	10		
1011	11		
1100	12		
1101	13		
1110	14		
1111	15		

Veja abaixo, outro exemplo.

Converter -6F,A<sub>16</sub> para binário.

Conforme a tabela acima, podemos perceber que precisamos de quatro dígitos binários para representar todos os símbolos hexadecimais. Para fazer





uma conversão rápida entre bases hexadecimais para binário podemos separar cada número hexadecimal e representá-lo em binário, depois basta agregar estes números binários para a representação final, veja o exemplo abaixo.

O contrário também é verdadeiro, para representar um número hexadecimal em binário devemos agrupar sempre em quatro dígitos, da direita para a esquerda do lado esquerdo da vírgula e da esquerda para a direita do lado direito da vírgula, e dar seu equivalente em hexadecimal. Exemplo

Converter 11,000011001 2 para hexadecimal

$$11,000011001_2 = 11,0000 1100 1_2$$
  
= 0011, 0000 1100 1000<sub>2</sub>  
= 3, 0 C 8<sub>16</sub>  
= 3,0C8 16

#### 4 - Função processamento

A CPU é encarregada de realizar as atividades relacionadas com a efetiva execução de uma operação, ou seja, tem a função de processar.

Processar o dado é executar com ele uma ação que produza algum tipo de resultado. Esta é a atividade fim do sistema de computação, ele existe para processar dados. Entre as tarefas comuns a esta função - processamento - podem ser citadas as que realizam: operações aritméticas, lógicas, movimentação de dados, desvios e operações de entrada e saída. O diagrama básico de um computador pode ser visto na figura abaixo.



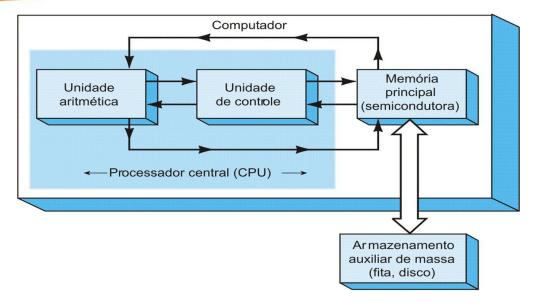


Figura 4: Diagrama básico do processador

Perceba que a unidade aritmética e a unidade de controle são implementadas dentro da Unidade Central de Processamento (CPU), pela figura acima pode-se notar que as setas da unidade aritmética tem acesso direto a memória principal tanto para escrita quanto para leitura dos dados.

A unidade de controle tem a função de receber informações da unidade aritmética, mais precisamente do registrador de dados, geralmente estas informações são endereçamento e são usados para posicionar em algum ponto da memória principal para que os dados possam ser lidos ou escritos.

A memória principal tem uma função bastante importante nos computadores digitais, pois é nela que fica armazenado todo e qualquer programa que esta sendo executado, portanto, tudo que nós vemos na tela, tudo que selecionamos com mouse, estão contidos na memória principal para serem processados pela CPU, desta forma podemos concluir que a CPU faz a comunicação com a memória principal para capturar os dados ou instruções que nela estão armazenados. Este é o conceito de programa armazenado ao qual Vou Neumann havia descrito em um de seus artigos e é usado como base de todos os computadores digitais desenvolvidos atualmente.

Pelo que foi descrito até agora, você já deve ter notado que a CPU faz o controle dos dados que devem ser lidos ou escritos na memória. Na memória principal são armazenados dados e instruções, os dados quando entram na



CPU são direcionadas para unidade lógica e aritmética, as instruções quando entram na CPU são direcionados para unidade de controle.

O diagrama abaixo apresenta alguns componentes da unidade central de processamento (CPU). Vamos, primeiramente, conceituar cada um dos componentes para depois entender a função de cada um deles de forma integrada.

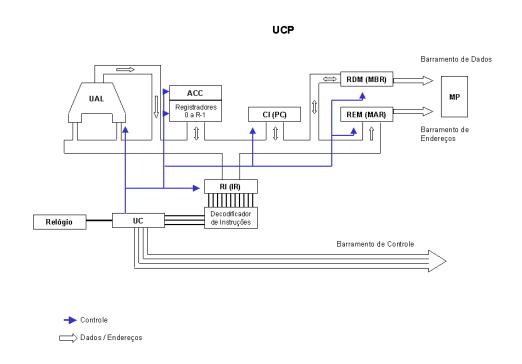


Figura 5: Diagrama detalhado do processador

UAL: A Unidade lógica e Aritmética tem como função receber dados para uma operação ou então comparação entre eles; observe na figura acima que em um dos lados há duas entradas de dados para a UAL e do outro lado uma saída, isto significa que serão recebidos dois dados para uma operação aritmética ou comparação entre eles e o resultado sairá pelo outro lado da unidade lógica e aritmética.

Registradores (ACC): os dados vindos da memória principal ou os resultados obtidos pelo processamento na unidade lógica e aritmética podem ser inseridos, temporariamente, nos registradores, o acumulador (ACC) é um dos principais registradoras dentro da CPU, pois toda e qualquer operação são feitas em relação a ele. Convém conceituar que estes registradores nada mais



são que memórias de acesso rápido e estão inseridos dentro da unidade central de processamento para que a velocidade de comunicação entre ele e a unidade lógica e aritmética tenha um melhor desempenho.

PC - significa contador de programas, sua função é gerar endereços de memória para acessar os dados ou instruções. Quando ligamos o computador o contador de programa inicia-se zerado para que posicione no primeiro endereço de memória, neste endereço encontra-se a memória FLASH conhecida como BIOS e é nela que se encontram as primeiras instruções para verificar o hardware e carregar o sistema operacional para a memória principal.

MBR : este registrador de barramento de memória tem a função de receber ou enviar os dados ou instruções que estão armazenados na memória. Todos os dados ou instruções passam por este registrador no momento de leitura ou escrita. Pelo conceito estabelecido podemos concluir que pelo MBR trafegam dados ou instruções que vem ou que vão para a memória principal.

MAR: o registrador de endereço de memória recebe endereçamento tanto do contador de programa (PC) como também dos registradores de instruções, sua função é de apontar para o endereço de memória indicando onde os dados ou instruções serão lidos ou escritos. Uma instrução pode indicar uma posição de memória, quando isto ocorre o decodificador de instruções envia para o registrador MAR este endereço, por sua vez o registrador MAR espelha este endereço para o contador de programa e posiciona a memória para que os dados possam lidos ou escritos.

Relógio: o relógio, como vimos anteriormente, tem a função de manter a sincronia de todos os componentes que estão dentro da unidade central de processamento. Ele gera sinais digitais síncronos para que os componentes mantenham, de forma, síncrona, a comunicação dos dados.

UC: a unidade de controle tem como função receber uma instrução e decodificá-las. O barramento de controle é ligado nos componentes externos que recebem informações do tipo leitura ou escrita e, em qual dispositivo estas ações devem ocorrer. Ele recebe estas instruções do registrador de instruções.

RI: registrador de instrução. Sempre que uma instrução é carregada para unidade central de processamento ela é direcionada para este registrador, sua função é decodificar estas instruções e passar parte dela para unidade de controle e outra parte para o registrador de endereçamento de memória.



Acima, você conheceu a função de cada um dos componentes da unidade central de processamento, esses componentes devem trabalhar de forma integrada, cada um executando as tarefas pelas quais são responsáveis, mas o que acontece com os dados ou instruções que circulam pela unidade central de processamento? Logo que o computador é ligado, o contador de programas (PC) é zerado, esta informação servirá como primeiro endereço para acessar uma instrução na memória, sendo assim, o contador de programas envia esta informação (endereço) para o registrador de endereço (MAR). O registrador de endereço, posiciona no endereço zero da memória, neste caso, para os computadores atuais, este endereço aponta para a BIOS.

Esta memória contem as primeiras instruções de verificação de periféricos e instruções para carregar o sistema operacional para memória principal. O registrador de endereço não lê as informações que estão na memória, ele apenas aponta para um determinado endereço, neste caso, informado pelo contador de programas (PC). Uma vez informado o endereço, o registrador de dados irá ler a informação da primeira posição da memória, esta instrução irá ser carregada e enviada através do barramento interno para o registrador de instrução (RI), pois na primeira posição de memória é conveniente que tenha uma instrução para iniciar todo o processo de boot. Deve ficar bem claro até este momento que todas as instruções irão seguir esse fluxo, ou seja, passará do registrador de dados para o registrador de instrução, isto ocorrerá caso a informação lida for uma instrução. O registrador de instrução irá analisar essa instrução enviada a ele, se essa instrução informar que um determinado dado deve ser lido em outra posição de memória, então parte destas informações, que neste caso é o endereço, vai para o registrador de endereço (MAR) e a outra parte da instrução, que seria instrução de leitura, irá para a unidade de controle. O registrador de endereço irá se posicionar neste novo endereço informado pela instrução e a unidade de controle irá acionar o circuito de leitura da memória, pois é exatamente isto que a instrução solicitou. Se já se tem o endereço e a unidade de controle solicitou a leitura dos dados, novamente o registrador de endereço irá pegar este dado posicionando-o em uma das entradas da unidade lógica e aritmética. Este procedimento ocorre milhões de vezes por segundo na unidade central de



processamento, uma hora é lida uma instrução outra é lido um dado, a instrução é enviada para o registrador de instruções (R I) para decodificá-las, os dados são enviados para unidade lógica e aritmética para serem processados executando uma operação aritmética ou comparação. Uma vez processados os dados da unidade lógica e aritmética será retornado um resultado, este resultado poderá ficar no acumulador ou em um dos registradores, isso dependerá da instrução sendo executada.

De qualquer forma o resultado irá ser armazenado na memória principal, dois fatos podem ocorrer aí: o resultado sai da unidade lógica e aritmética e é gravado direto na memória principal através do registrador de dados ou o resultado da unidade lógica e aritmética será enviado para um registrador para posteriormente ser enviado para a memória principal. Todo este conjunto tem que trabalhar de forma sincronizada, os microprocessadores, conforme já descrito em aulas anteriores, tem um conjunto de instruções reduzidas e estas instruções é que vão determinar as operações que devem ocorrer dentro da unidade central de processamento. Qualquer linguagem de programação de alto nível, tendo ela vários comandos para serem aplicados em um determinado programa, quando a baixo nível será convertido a uma dessas instruções que o microprocessador dispõe.





# Referências

STALLINGS, W. **Arquitetura e Organizacao de Computadores:** Projeto Para o Desempenho. 5. ed. Sao Paulo: Prentice Hall, 2004.

# Responsável pelo Conteúdo:

Prof. Vagner Silva

#### **Revisão Textual:**

Profa Ms. Rosemary Toffoli



www.cruzeirodosul.edu.br

Campus Liberdade

Rua Galvão Bueno, 868

01506-000

São Paulo SP Brasil

Tel: (55 11) 3385-3000

