Infraestrutura de Hardware

Juliana Regueira Basto Diniz
Abner Corrêa Barros

Volume 3

Recife, 2009

Universidade Federal Rural de Pernambuco

Reitor: Prof. Valmar Corrêa de Andrade
Vice-Reitor: Prof. Reginaldo Barros
Pró-Reitor de Administração: Prof. Francisco Fernando Ramos Carvalho
Pró-Reitor de Extensão: Prof. Paulo Donizeti Siepierski
Pró-Reitor de Pesquisa e Pós-Graduação: Prof. Fernando José Freire
Pró-Reitor de Planejamento: Prof. Rinaldo Luiz Caraciolo Ferreira
Pró-Reitora de Ensino de Graduação: Profª. Maria José de Sena
Coordenação Geral de Ensino a Distância: Profª Marizete Silva Santos

Produção Gráfica e Editorial
Capa e Editoração: Allyson Vila Nova, Rafael Lira, Italo Amorim, Glaucia Fagundes e Arlinda Torres
Revisão Ortográfica: Marcelo Melo
Ilustrações: Abner Barros e Allyson Vila Nova
Coordenação de Produção: Marizete Silva Santos

Sumário

Apresentação

Caro(a) Cursista,

Estamos, neste momento, iniciando o terceiro volume do livro da disciplina de **Infraestrutura de Hardware**. Neste volume iremos discutir o subsistema de memória que constituem os computadores modernos. As memórias são subdivididas de acordo com uma hierarquia e podem ser classificadas de acordo com uma série de critérios. Estudaremos as memórias semicondutoras, as memórias de superfície ótica (CDs e DVDs) e de superfície magnética (Discos rígidos).

Também abordaremos a troca de dados entre as memórias principal e a memória cache, bem como as políticas de substituição e de escrita de dados na memória principal e na cache.

Bons estudos!

Juliana Regueira Basto Diniz
Abner Barros

Professores Autores

Capítulo 1

O que vamos estudar?

Neste capítulo, vamos estudar os seguintes temas:

» Hierarquia de memória

» Classificação dos elementos de memória que compõem a hierarquia de memória entre memória primária e memória secundária

» Características técnicas e funcionais dos principais elementos de memória presentes na hierarquia de memória dos computadores atuais

Metas

Após o estudo deste capítulo, esperamos que você consiga:

» Entender como e porque foi criada a Hierarquia de Memória. Como esta hierarquia funciona e qual o seu objetivo na arquitetura dos computadores modernos

» Conheçer os princípios que nortearam a definição desta hierarquia, do ponto de vista do aproveitamento das características de cada um dos seus componentes

» Ter uma visão geral, sem muitos detalhes, das principais características técnicas e funcionais dos componentes que formam a hierarquia de memória

Capítulo 1 – Hierarquia de Memória

Vamos conversar sobre o assunto?

Considere as seguintes afirmações, feitas em épocas distintas, por alguns dos “papas“ da computação:

“Em termos ideais, desejaríamos dispor de uma capacidade de memória infinitamente grande e que pudesse disponibilizar imediatamente o conteúdo de qualquer de suas palavras...“

A.W.Burks, H.H.Goldstine e J. Von Neumann

“Desde o lançamento do primeiro computador, os programadores vêm exigindo capacidades ilimitadas de memória, de acesso quase instantâneo“

Andrew S. Tanenbaum

“...Somos forçados a reconhecer a possibilidade de construir um sistema de memória estruturado hierarquicamente, no qual cada um dos componentes da hierarquia tenha mais capacidade de armazenamento e um tempo de acesso maior do que aqueles que o precedem.”

A.W.Burks, H.H.Goldstine e J. Von Neumann

Como você deve ter percebido, a organização do sistema de memória (tamanho da memória x velocidade de acesso) é um dos problemas cruciais em sistemas computacionais.

Precisamos
De um lado, precisamos de um sistema de memória com uma grande capacidade de armazenamento e que
uma
espaço de armazenamento, que possa ser acessado a altíssima quase que instantâneamente.
velocidade. Infelizamente, entretanto,
Por outro lado, ainda não existe uma tecnologia de armazenamento que nos permita construir tais sistemas de memória a um custo acessível.

Lembre-se que tanto os programas quanto os dados ficarão armazenados neste sistema de memória, e que da velocidade de acesso a
estes
este depende, em última instância, o desempenho do computador como um todo.

Desta forma, como o próprio Von Neumann percebeu, ainda nos primórdios da história dos computadores, a única alternativa viável é a construção de sistemas de memória baseados em hierarquia, que lance mão de todas as tecnologias disponíveis, de forma a extrair o melhor de cada uma delas.

Este portanto será o foco do nosso estudo. Conhecer as tecnologias de armazenamento de dados atualmente disponíveis, e ver como estas tecnologias são aproveitadas na construção da hierarquia de memória.

Hierarquia de Memória

Quando pensamos em uma hierarquia qualquer, logo nos vem à mente uma estrutura que dispõe os seus elementos a partir de algum parâmetro que os ~~distingue~~ distingua em grau de importância dos demais. Na hierarquia de memória não é diferente, nela é a velocidade de acesso que determina o grau de importância de um elemento e, portanto, o seu grau de proximidade do processador. A Figura 1, a seguir, ~~nos~~ traz os diversos elementos presentes na hierarquia de memória dos computadores atuais.

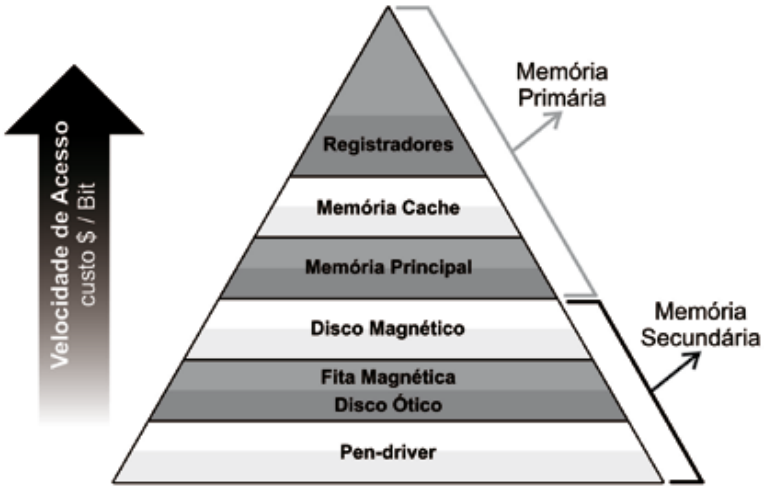

Figura 1 - Elementos da Hierarquia de Memória

Como podemos observar por esta figura, os elementos presentes na hierarquia de memória podem ainda ser organizados em dois grupos distintos: a Memória Primária e a Memória Secundária. Esta classificação está diretamente ligada à forma como cada um destes elementos se conecta ao sistema e, desta forma, como o processador tem acesso aos dados neles armazenados.

Fazem parte da Memória Primária os elementos aos quais o processador tem acesso direto e imediato, ou seja, aqueles elementos que estão conectados diretamente ao barramento de memória do processador. É na memória primária que normalmente ficam armazenados os dados e trechos de código dos programas em execução.

Como podemos ver pela Figura 1, ~~fazem parte da~~ estão presentes na memória primária o conjunto de registradores do processador, a memória cache e a memória principal, comumente chamada de memória RAM.

Por outro lado, fazem parte da memória secundária os elementos aos quais o processador tem acesso indireto, ou seja, aqueles que não estão conectados ~~ao~~ diretamente ao barramento de memória, mas sim a algum dos barramentos de entrada e saída do sistema.

Para que os dados armazenados nestes dispositivos se tornem acessíveis na hierarquia de memória, é necessária a utilização de device drivers, softwares específicos que intermediam as transações de leitura e/ou escrita na memória entre o programa em execução e o dispositivo em que estão armazenados os dados. São estes device drivers que permitem ao processador acessar os dados armazenados nestes dispositivos da mesma forma como acessaria os dados armazenados na Memória Primária. Ou seja, do ponto de vista do processador, é como se existisse apenas um único e grande espaço de memória, compreendendo os elementos presentes tanto na memória primária quanto secundária, ao qual ele pode acessar de maneira simples e direta.

Observe que, como falamos a princípio, o objetivo desta hierarquia é construir um sistema de memória com as principais características de cada um dos seus elementos, de tal forma que possa ser visto pelo processador com um espaço de armazenamento tão grande quanto o disponível nos elementos presentes na sua memória secundária e com um acesso tão rápido quando o possível nos elementos presentes na sua memória primária.

Em linhas gerais podemos considerar que, ao longo do tempo da execução de um programa os seus dados trafegam entre os diversos níveis desta hierarquia, vindo desde o local em que ficam armazenados quando não estão em uso, normalmente em algum ponto da memória secundária, passando pela memória principal e desta para a memória cache e para o banco de registradores, onde

por fim se tornam acessíveis ao processador.

No próximo capítulo estudaremos com detalhes como se dá este processo e quais as técnicas adotadas a fim de garantir o melhor desempenho do sistema de memória como um todo.

Por enquanto, que tal estudarmos com um pouco mais de detalhes alguns dos elementos presentes na hierarquia de memória que acabamos de conhecer? Para tanto, vamos começar estudando algumas características fundamentais dos elementos que compõem o sistemas de memória dos computadores atuais. A Figura 2, a seguir, nos traz um pequeno resumo destas características.

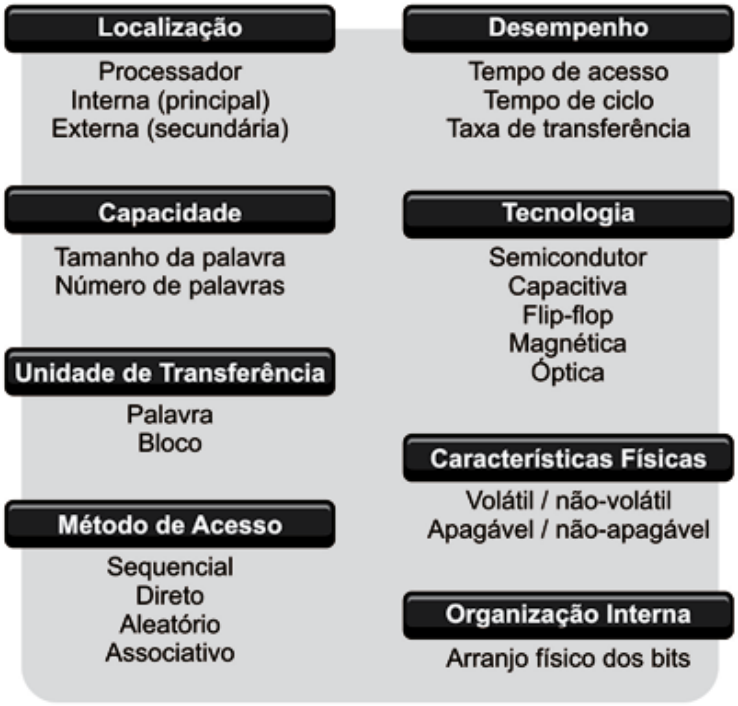

Figura 2 - Características fundamentais dos elementos do sistema de memória

» **Localização:** Esta característica está diretamente relacionada com a localização física do elemento no sistema computacional, ou seja, se este é interno ao processador como no caso dos registradores, externo ao processador, mas com conexão direta ao barramento de memória deste, sendo desta forma considerado como pertencente à memória interna do computador, ou ainda um elemento de armazenamento externo conectado ao sistema através de um dos barramentos de entrada e saída.

» **Capacidade:** Normalmente expressa em bytes, mas podendo

ser expressa também diretamente em bits ou em palavras de 16, 32 ou 64 bits. Esta característica revela a capacidade de armazenamento de informações do elemento de memória. Não é por acaso que esta é talvez a característica que mais interesse aos usuários comuns de computadores. Observe entretanto que possuir um computador com uma grande capacidade de memória não é, obrigatoriamente, sinônimo de possuir um computador de grande desempenho.

» **Unidade de transferência:** Esta característica está associada a como a informação é transferida de ou para o elemento de memória. Como já estudamos anteriormente, apesar de a unidade básica de informação nos computadores ser o bit, a unidade básica de manipulação da informação, é determinada a partir do tamanho da palavra do processador adotado. Por outro lado, por motivos de desempenho, conforme veremos mais a frente, os dados são normalmente transferidos em blocos contendo várias palavras formando assim uma unidade de transferência de dados.

» **Método de acesso:** Esta característica determina como se dará o acesso aos elementos de memória, ou às suas unidades endereçáveis. Existem, a princípio, quatro formas de acesso aos dados, são elas:

  ○ **Acesso sequencial:** no acesso sequencial uma determinada posição N de memória só pode ser acessada após ter-se acessado a posição N-1. Sistemas que adotam este tipo de acesso armazenam juntamente com o dado um registro de informação que contem o seu endereço de acesso. Desta forma, para se acessar uma determinada posição de memória deve-se ir acessando uma a uma todas as posições que a antecedem, lendo sempre o seu identificador, até alcançar o bloco desejado. Exemplo de dispositivo de armazenamento de acesso sequencial: Fita Magnética.

  ○ **Acesso direto:** o acesso direto pode ser considerado uma evolução do acesso sequencial. Nele os dados são organizados em blocos maiores, ou clusters, os quais possuem uma localização física conhecida no meio de armazenamento. Dentro do bloco os dados são acessados sequencialmente conforme acabamos de descrever.

Exemplo de dispositivo de armazenamento de acesso direto: Disco Magnético.

- **Acesso aleatório:** no acesso aleatório cada elemento de armazenamento possui um endereço único, definido a partir da posição física que este ocupa na estrutura do dispositivo que o implementa, podendo desta forma ser acessado diretamente, em qualquer sequencia, independente da posição de memória anteriormente acessada. Exemplo de dispositivo de armazenamento de acesso aleatório: Memória principal, também conhecida como memória RAM (*Random Access Memory*)[1].
- **Associativo:** o acesso associativo pode ser visto como uma junção do acesso direto com o acesso aleatório. Nos sistemas que adotam o acesso associativo cada posição de memória possui, além do espaço para o armazenamento da informação propriamente dita, um espaço para o endereço associado aquela informação. Desta forma, os espaços de armazenamento não tem mais um endereço fixo como acontece no acesso aleatório, podendo ser utilizados de uma maneira muito mais eficiente. O acesso aos dados é feito comparando-se o endereço da informação que se deseja acessar com os endereços associados aos elementos armazenados. Esta comparação pode ser feita simultaneamente com todos os elementos de memória, o que permite um acesso rápido e eficiente dos dados armazenados. Exemplo de dispositivo de armazenamento de acesso associativo: Memória cache. Mais a frente, no próximo capítulo, estudaremos em detalhes como o endereçamento associativo é importante para o funcionamento deste tipo de memória.

Saiba Mais

[1] Pesquise na Internet o significado da expressão: Random Access Memory

» **Desempenho:** Juntamente com a capacidade de armazenamento, o desempenho é uma das características mais importantes de um elemento de memória. Este desempenho pode ser medido a partir de três medidas básicas descritas a seguir:

- **Tempo de acesso:** Em uma memória de acesso aleatório, este é o tempo necessário para se efetuar um processo de escrita ou leitura em uma determinada posição de memória, ou seja, é o tempo necessário para que se

possa efetivamente gravar um dado em uma operação de escrita ou para que se possa disponibilizar um dado lido no barramento de dados da memória em uma operação de leitura de memória. Em elementos de memória de acesso direto ou em elementos de memória de acesso sequencial, este é o tempo necessário para acessar a primeira posição da região de memória desejada, por este motivo é também conhecido como tempo de posicionamento.

- **Tempo de ciclo de memória:** Em memórias de acesso aleatório, este tempo pode ser calculado como sendo o somatório do tempo de acesso mais o tempo necessário à liberação do barramento para um novo acesso. Em memórias de acesso direto e em memórias de acesso sequencial, este é o tempo necessário para acessar duas posições sucessivas em uma mesma região de memória.
- **Taxa de transferência:** É a taxa ou a velocidade com que os dados podem ser transferidos de ou para a memória. Em memórias de acesso aleatório, esta é igual ao inverso do tempo de ciclo de memória. Em memórias de acesso direto ou de acesso sequencial, vale a seguinte relação:

$$T_N = T_A + \frac{N}{R}$$

T_N = Tempo médio para ler ou escrever N bits
T_A = Tempo de acesso médio
N = Número de bits a ser transferidos
R = 1/tempo ciclo = taxa de transferência em bits por segundo

Figura 3 - Taxa de transferência

» **Tecnologia:** A tecnologia de construção de um elemento de memória tem impacto direto sobre todas as suas características. Normalmente uma tecnologia que apresenta excelentes resultados em determinados parâmetros não apresenta tão bons resultados em outros parâmetros. Por exemplo: os elementos de armazenamento baseados na tecnologia Magnética costumam ter um excelente custo de implementação, apresentando, assim, um baixo custo por bit, entretanto o seu tempo de acesso costuma ser milhares de vezes mais lento que os das memórias baseadas na tecnologia de semicondutores. As

principais tecnologias atualmente empregadas na construção de elementos de memória são:

- **Semicondutor:** Atualmente existe um grande número de dispositivos de armazenamento construídos a partir das tecnologias baseadas em semicondutores, entretanto as principais, do ponto de vista da construção da hierarquia de memória dos computadores são os dispositivos construídos a partir das seguintes tecnologias:
  - **Capacitiva:** Esta tecnologia tem dominado o mercado de fabricação das memórias tipo DRAM (Dynamic Random Access Memory), ou seja, das memórias de armazenamento dinâmico e acesso randômico, desde o seu lançamento no iníco da década da 70 do século passado. Nela a unidade básica de informação, o bit, é armazenado em capacitores construídos diretamente na pastilha de silício dos componentes de memória. A grande vantagem desta tecnologia é a sua alta densidade de armazenamento por área de silício, o que reduz em muito o preço por bit armazenado. Seu tempo de acesso e taxa de transferência, ainda que superior a grande maioria das outras tecnologias, não tem conseguido acompanhar a velocidade dos processadores atuais, sendo menor que os apresentados nas memórias construídas a partir de Flip-Flops, motivo pelo qual esta tecnologia ocupa o ponto mediano da hierarquia de memória.
  - **Flip-Flop:** Esta tecnologia é a que apresenta as melhores taxas de acesso e de transferência dentre todas as tecnologias atualmente disponíveis, entretanto devido a sua baixa densidade de armazenamento, cada bit de memória ocupa muita área na pastilha de silício, o seu custo por bit armazenado é também o mais caro entre todas as tecnologias atualmente empregadas. Desta forma, esta tecnologia é normalmente empregada apenas na construção dos dispositivos que ocupam o topo da hierarquia de memória, ou seja, dos registradores e da memória cache.
- **Magnética:** Esta tecnologia tem sido desde o seu

lançamento a principal tecnologia para a construção de elementos de armazenamento persistente da informação. Nela a informação é armazenada mudando-se a orientação do campo magnético de partículas de material ferri-magnético depositados nas mídias de armazenamento. A grande vantagem desta tecnologia é o seu baixo custo de construção e sua alta densidade de armazenamento, o que resulta em um baixíssimo custo por bit armazenado. Os principais dispositivos construídos com esta tecnologia são os discos rígidos, os discos flexíveis e as fitas magnéticas. A principal desvantagem desta tecnologia é o seu alto tempo de acesso e de ciclo de leitura e escrita, o que resulta em uma baixa taxa de transferência.

- **Ótica:** A tecnologia de armazenamento em mídias óticas surgiu como uma alternativa ao armazenamento de dados em mídias magnéticas, principalmente em substituição às mídias removíveis. Suas principais características são sua alta densidade de armazenamento e sua capacidade de manter os dados praticamente inalterados por tempo infinitamente longos se comparados às demais tecnologias, desde que armazenados de maneira adequada. Na tecnologia ótica a unidade básica de informação é armazenada a partir da mudança do índice de refração da luz da mídia no ponto em que a informação é gravada. Atualmente os principais dispositivos construídos com esta tecnologia são os CDs, DVDs e Blue-Rays.

» **Características Físicas:** As características físicas da tecnologia empregada na construção dos elementos de armazenamento determinam o grau de persistência da informação armazenada. Desta forma, as memórias podem ser classificadas em:

- **Voláteis ou não-voláteis:** Dizemos que uma memória é volátil quando esta depende de algum estímulo externo, para manter os dados nela armazenados. São, portanto, não-voláteis todos os demais tipos de memória que não dependem de tais estímulos para manter os dados armazenados. Atualmente apenas as memórias DRAM, as quais são construídas com tecnologia capacitiva são consideradas memórias voláteis.

- **Apagáveis ou não apagáveis:** Esta característica quase

dispensa comentário. Atualmente, apenas alguns tipos de mídias óticas e alguns tipos de memórias baseadas em semicondutores podem ser consideradas como memórias não apagáveis, uma vez que os dados nelas gravados não podem ser alterados sem que as mesmas sejam destruídas. Todas os demais tipos de memória podem ser consideradas como memórias apagáveis.

» **Organização:** A organização da memória é a característica que reflete como os seus bits estão dispostos, ou seja, qual o tamanho da palavra de dados com o qual a memória pode ser acessada. Observe a Figura 4 a seguir, nela podemos ver três formas distintas como uma memória com 96 bits de armazenamento pode ser organizada.

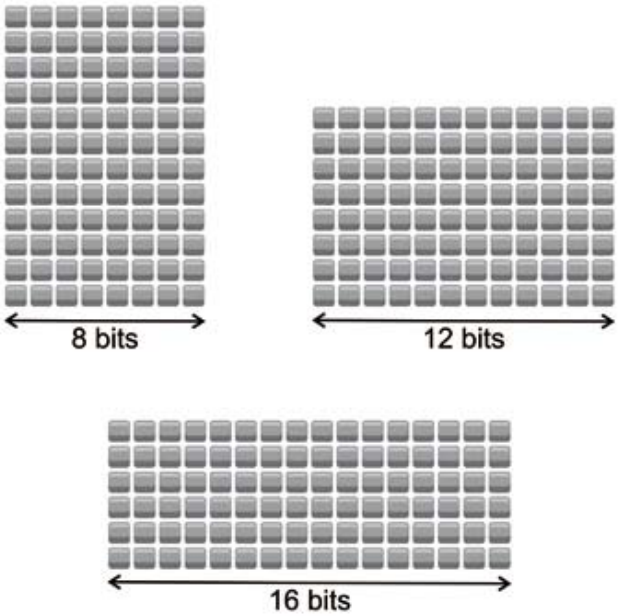

Figura 4 - Três formas distintas de organizar uma memória com 96 bits de armazenamento

Muito bem, agora que já conhecemos as principais características dos elementos que compõem um sistema de memória, estamos prontos para estudar os principais tipos de elementos de memória utilizados na construção da hierarquia de memória.

Vamos começar nosso estudo pelos elementos que compõem a memória primária da nossa hierarquia de memória:

Memória Primária

Conforme dissemos a princípio, fazem parte da Memória Primária

os elementos aos quais o processador tem acesso direto e imediato, ou seja, aqueles que estão conectados diretamente ao barramento de memória do processador. Por este motivo, a memória primária é também conhecida como memória interna. Tomando como referência a hierarquia disposta na Figura 1, vamos estudar primeiramente os registradores, em seguida a memória cache e, por último, a memória principal.

Registradores

Como podemos observar pela **Figura 1**, os registradores são os elementos mais próximos do processador, ocupando assim o topo da nossa hierarquia de memória. Os registradores nada mais são que palavras de memória construídas com tecnologia de semicondutores, Flip-Flops, diretamente na pastilha de silício do processador, sendo desta forma os elementos de memória de acesso mais rápido em toda a hierarquia de memória. A maioria dos registradores presentes em um processador são dispostos na forma de um banco de memória de acesso aleatório, organizado em palavras do tamanho exato da palavra do processador, as quais são acessíveis uma a uma em unidades de transferência também do tamanho exato da palavra do processador. Os registradores são considerados memórias apagáveis e não voláteis

Memória Cache

A memória cache é um espaço de memória, organizado na forma de bancos de memória, normalmente construída em dois ou três níveis, um diretamente na pastilha do processador, chamada de cache L1, e os demais, denominados de cache L2 e L3, construídos como componentes a parte do processador. Conforme dissemos anteriormente, da mesma forma que os registradores, a memória cache é construída com tecnologia de semicondutor, Flip-Flop, com acesso associativo, e organizado em palavras com tantos bits quantos necessários para armazenar uma ou mais palavras do processador e uma referência ao endereço de memória associado aos dados armazenados. Dada a sua importância para o desempenho da hierarquia de memória como um todo, a memória cache será estudada em maiores detalhes no próximo capítulo.

Memória Principal

A memória principal, como o próprio nome sugere é o elemento

central da hierarquia de memória. É na memória principal que normalmente ficam armazenados os trechos de código e os dados dos programas em execução.

Em linhas gerais, podemos dizer que a memória principal e o banco de registradores são os únicos espaços de memória visíveis ao processador. Tanto a memória cache quanto os elementos de armazenamento presentes na memória secundária existem a fim de melhorar as características de velocidade de acesso e de espaço de armazenamento da memória principal. O que se busca com a hierarquia de memória é que o processador veja a memória principal como se esta tivesse a velocidade de acesso da memória cache e o espaço de armazenamento da memória secundária.

Atualmente, a memória principal é construída a partir da utilização de vários chips de memória DRAM, organizados na forma de bancos de memória que podem variar de centenas de Mega-Bytes a dezenas de Giga-Bytes de memória. Conforme já foi dito, as memórias DRAM são memórias construídas com tecnologia capacitiva, voláteis, apagáveis, de acesso aleatório, organizadas e acessadas normalmente em palavras do tamanho da palavra do processador.

Entretanto, existe um outro tipo de memória que também faz parte da memória principal, na qual fica armazenado o Sistema Básico de Entrada e Saída da placa mãe, também conhecido como BIOS **(***Basic Input Output System***)**. O BIOS é o programa responsável pela verificação inicial da integridade dos componentes básicos da placa mãe e pela carga inicial do sistema operacional. Ele fica gravado em uma memória construída com tecnologia de semicondutor, não apagável, somente de leitura, denominada de memória ROM **(***Read Only Memory***)**. Assim como a memória DRAM, a memória ROM é uma memória de acesso aleatório, organizada e acessada em palavras do tamanho da palavra do processador.

Bem, agora que já conhecemos os elementos que compõem a memória primária, que tal conhecermos os elementos que compõem a memória secundária da nossa hierarquia de memória?

Memória Secundária

A memória secundária é formada pelos elementos de memória que não estão diretamente conectados ao barramento de memória do processador, sendo desta forma acessados de maneira indireta

a partir do barramento de entrada e saída. De um modo geral, podemos dizer que a memória secundária é formada por elemento de armazenamento de massa, tais como disco rígido e fita magnética, e pelos elementos de armazenamento removíveis, tais como discos flexíveis, discos óticos e pen-drivers. Por não estarem diretamente conectados ao barramento de memória do processador, estes também são conhecidos como memória externa.

Ainda seguindo a ordem apresentada na **Figura 1**, estudaremos primeiramente os Discos Magnéticos, em seguida os Discos Óticos, as Fitas Magnéticas e, por fim, os pen-drivers.

Disco Magnético

Conforme já dissemos, os discos magnéticos recebem esta designação por armazenarem a informação a partir da mudança da orientação magnética de minúsculas partículas Ferri-magnéticas depositadas sobre uma mídia suporte.

Saiba Mais

[2] Conheça um pouco da história, com fotos, dos primeiros discos Magnéticos. Visite o site http://www-03.ibm.com/ibm/history/exhibits/storage/storage_350.html

É realmente muito interessante!

Desde o seu lançamento, feito pela IBM em meados da década de 50 do século passado, os discos Magnéticos evoluíram[2] muito, tendo o seu tamanho sido reduzido a frações do tamanho original ao mesmo tempo em que sua capacidade de armazenamento e velocidade de acesso foram aumentadas de centenas a milhares de vezes a dos primeiros dispositivos. Os primeiros discos apresentados mediam 60x68x29 polegadas, aproximadamente 152 x 172 x 74 cm, e tinham uma capacidade de armazenamento de 50 MB de informação.

Ao longo do tempo foram surgindo diversas variantes para este tipo de dispositivo, sendo os mais importantes o disco flexível, também chamado de disquete, e o disco rígido, nosso conhecido HD. Dada a sua importância, tanto no contexto histórico quanto no contexto atual, aqui estudaremos apenas os discos rígidos.

A **Figura 5,** a seguir, nos traz um esquema simplificado da estrutura interna de um disco rígido. Como podemos perceber, seus principais componentes são a mídia magnética, o motor de tração da mídia, também conhecido como ***spinde motor***, a cabeça de leitura e escrita e o dispositivo de posicionamento da cabeça, também conhecido como ***voice coil***.

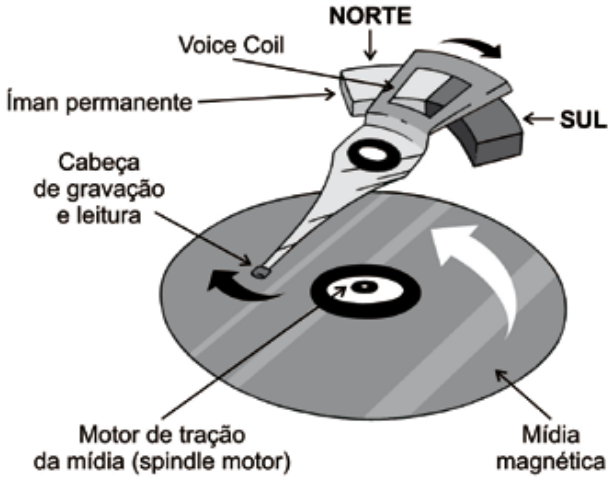

Figura 5 – Estrutura interna de um disco rigido

A fim de aumentar a capacidade de armazenamento do disco, os dois lados da mídia magnética são utilizados como discos independentes, cada qual sendo acessada por sua própria cabeça de leitura e escrita. É comum também utilizar-se um conjunto com várias mídias em conjunto, presas ao eixo de um único motor de tração, as quais são acessadas por um conjunto de cabeças de leitura e escrita controladas por um único dispositivo de posicionamento. A Figura 6, a seguir, pode nos dar uma ideia de como este conjunto fica disposto no interior do disco rígido.

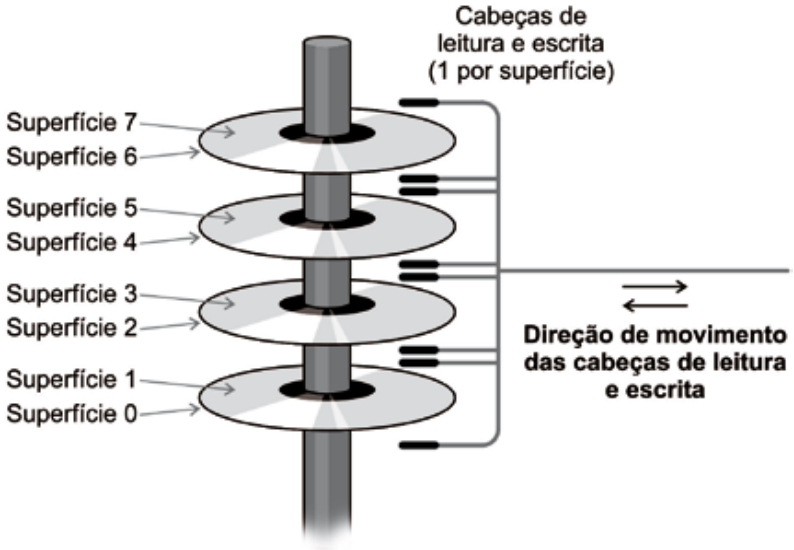

Figura 6 - Disposição das mídias Magnéticas e das cabeças de gravação e leitura de um disco rígido

O princípio de funcionamento de um disco rígido é de certa forma até bem simples. Conforme dissemos a princípio, os bits de dados

são gravados na mídia alterando-se o sentido do fluxo magnético do material Ferri-magnético que está depositado sobre esta, da mesma forma como se grava uma informação de áudio em uma fita magnética. A fim de permitir o acesso ordenado dos dados gravados, estes ficam organizados em blocos denominados de setores do disco, os quais são gravados em regiões distintas da mídia denominadas de trilhas. A Figura 7, a seguir, nos traz um diagrama simplificado de como fica a disposição das trilhas e setores do disco. Observe que cada trilha nada mais é que a região disposta sob a cabeça de leitura e escrita durante o processo de rotação que a mídia sofre ao ser tracionada pelo eixo do motor. Observe ainda que cada trilha, por sua vez, é dividida em pequenas regiões ou setores, em que são gravados os dados.

Desta forma, o processo de acesso aos dados gravados no disco exige primeiramente que a cabeça de leitura e escrita seja posicionada sobre a trilha onde se encontra o setor com os dados. Isto é feito variando-se a corrente elétrica aplicada ao **voice-coil**. Em seguida, simplesmente aguarda-se que o setor com os dados passe sob a cabeça de leitura e escrita, onde o fluxo magnético presente na mídia é convertido em corrente elétrica a qual é enviada à placa controladora do disco para, por fim, ser convertida em informação lógica. O processo de escrita é análogo a este, com exceção que em vez de ler a informação, a cabeça de leitura e escrita irá gravar a informação no setor desejado.

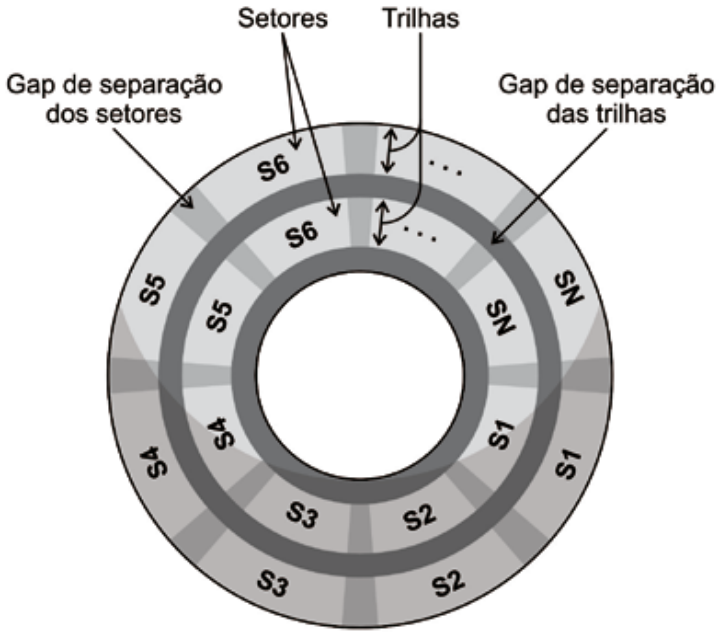

Figura 7 - Organização dos dados na mídia

Apenas para concluir, na Figura 8, a seguir, temos uma visão mais detalhada da organização dos dados nos setores do disco. Observe que antes dos dados propriamente ditos cada setor do disco tem uma região com um preâmbulo, ou seja, uma região com um padrão de gravação previamente estabelecido que permite identificar o ponto em que começa o setor e qual a sua identificação. E, após a região em que os dados estão gravados, temos uma região com um código de verificação de erro também conhecido como ECC. O ECC é utilizado tanto para a verificação da integridade dos dados gravados quanto para uma possível recuperação dos mesmos. Ainda como medida de segurança, a fim de melhor demarcar a região ocupada por cada setor, sempre entre dois setores existe uma região que é deixada sem gravação conhecida como gap de interseção.

Figura 8 - Organização dos dados nos setores do disco

Conforme citamos na introdução desta seção, com o passar do tempo a evolução dos discos rígidos não se deu apenas com vista à redução de suas dimensões físicas e no aumento da sua capacidade de armazenamento. Houve também avanços significativos na taxa de transferência dos dados e na redução das taxas de erro de leitura e escrita dos dados.

Sendo o disco rígido basicamente um elemento de memória, como é de se esperar, é de suma importância que os dados nele armazenados possam ser recuperados de maneira rápida e livre de falhas.

Observe que a princípio a taxa de transferência dos dados entre a mídia e o sistema é função direta da velocidade de rotação desta e da

área física ocupada pela unidade básica de informação gravada. Ou seja, quanto menor a área ocupada por um bit de informação e quanto mais rápido a mídia passar na frente da cabeça de leitura e gravação, maior é a taxa de transferência dos dados.

Atualmente a área ocupada por um bit de informação está na ordem do micrometro e a rotação da mídia na ordem das 10.000 rotações por minuto.

Por outro lado, quanto menor a área ocupada por um bit de informação na mídia e quanto maior a velocidade de rotação da mídia maior é o risco de ocorrer falhas de gravação e escrita.

Desta forma, historicamente, o mercado de discos rígidos teve sempre que buscar um ponto de equilíbrio entre a taxa de transferência e a segurança dos dados. Discos rápidos e seguros eram caros, discos baratos ou não eram rápidos ou não eram seguros.

Pensando nisto, no fim da década de 80 do século passado, um dos inventores dos processadores RISC, David Patterson, propôs uma nova arquitetura para os discos rígidos que permitiria aumentar a velocidade de transferência dos dados ao mesmo tempo em que aumentaria a segurança dos dados gravados, tudo isto a custa de inserir uma certa taxa de redundância nos dados gravados de tal forma que estes pudessem ser acessados em paralelo, e não mais sequencialmente como de costume, e que na ocorrência de uma falha os dados pudessem ser recuperados mais facilmente. Esta nova arquitetura recebeu o nome de RAID **(***Redundant Array of Independent Drives***).**

A proposta de Patterson na verdade apresentava não apenas uma, mas sete possibilidades diferentes de organizar um conjunto de discos rígidos a fim de explorar diferentes níveis de segurança e taxas de transferência dos dados. Veremos aqui apenas duas destas propostas que ficaram conhecidas como RAID 0 e RAID 1.

RAID Nível 0

A proposta da RAID 0 visa basicamente aumentar velocidade de acesso aos dados, ou seja, a taxa de transferência, sem entretanto apresentar nenhuma alteração no nível de segurança dos mesmos.

Conforme podemos ver pela Figura 9, a seguir, nesta proposta, os discos são internamente divididos em áreas, chamadas de **strip**.

Observe que no exemplo apresentado na figura nós temos quatro discos rígidos os quais foram divididos em 16 strip's. Veja que, se fosse possível dividir um arquivo em diversos arquivos menores, que estivessem distribuídos pelas strip's dos quatro discos, poderíamos acessar 4 partes do arquivo simultaneamente, uma de cada disco, aumentando assim em quatro vezes a nossa velocidade de leitura e/ou leitura.

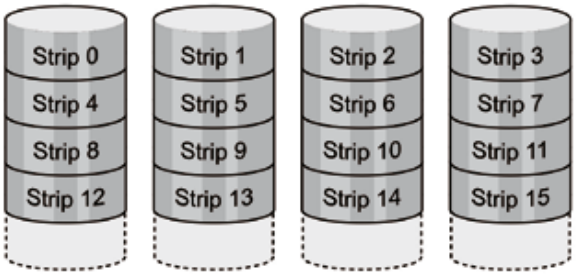

Figura 9 - Organização dos dados na RAID 0

RAID Nível 1

A proposta da RAID 1 é muito semelhante a anterior, e visa aumentar velocidade de acesso ao mesmo tempo em que busca aumentar o nível de segurança dos mesmos.

Conforme podemos ver pela **Figura 10,** a seguir, nesta proposta, os discos também são internamente divididos em áreas, só que além dos quatro discos iniciais temos mais quatro discos divididos exatamente como os quatro primeiros. O objetivo aqui é que este segundo conjunto de discos sirva como armazenamento de segurança para os dados gravados nos primeiros discos. Desta forma, sempre que um dado é gravado ou alterado no primeiro conjunto de discos o mesmo também é gravado ou alterado no segundo conjunto, de tal forma que caso seja identificada alguma falha nos dados gravados tanto no primeiro como no segundo conjunto de discos, a cópia existente no outro conjunto de discos possa ser utilizada para recuperar a falha identificada.

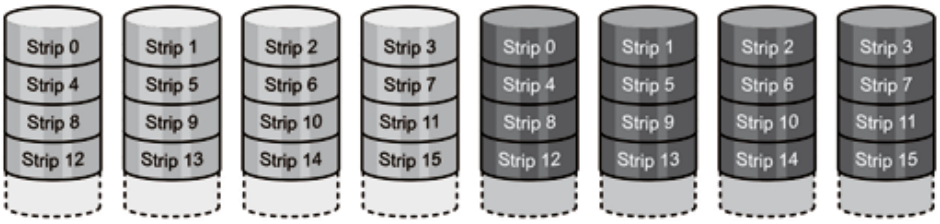

Figura 10 - Organização dos dados na RAID 1

Como você deve ter percebido, a utilização de uma arquitetura

RAID exige a utilização de um controlador especial, um hardware dedicado, responsável por prover tanto a leitura e/ou leitura simultânea das strip's nos discos, como por prover as verificações e correções de segurança necessárias nos arquivos gravados.

Disco Ótico

Os discos óticos recebem este nome por utilizarem a refração da luz como meio de gravação e leitura dos dados na mídia, ou seja, durante o processo de gravação a mídia recebe marcas, chamadas de Pits, que alteram o índice de refração da luz. As regiões que não recebem estas marcas recebem o nome de Lands. O mais interessante é que a informação não esta associada diretamente a presença ou não destas marcas, mas sim à transição entre elas. É utilizada a passagem de um Pit para um Land ou de um Land para um Pit indicar um bit em 1 e a ausência destas transições para indicar um bit em 0. A **Figura 11,** a seguir, pode nos dar uma ideia melhor do que estamos falando.

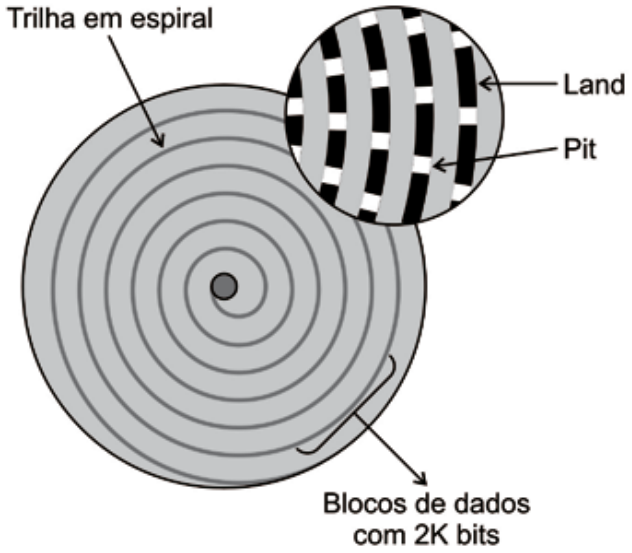

Figura 11 - Gravação dos dados na mídia ótica

Diferentemente dos discos magnéticos, onde os dados são gravados em trilhas concêntricas, nos discos óticos os dados são gravados sequencialmente em uma trilha espiral única, que nos CDs mede em torno de 5,6km, que se projeta no centro para a borda do disco. Devido a isto, a fim de evitar que a velocidade de leitura e/ou escrita do disco varie ao longo da trilha, a velocidade de rotação do disco varia, de forma a manter a taxa de gravação e/ou leitura aproximadamente constante.

O processo de leitura de uma mídia ótica se dá pela através de uma cabeça ótica de leitura e escrita, conforme podemos ver pela **Figura 12,** a seguir. Nesta figura temos um esquema simplificado, em corte dos elementos que formam esta cabeça ótica de leitura e escrita. Observe que temos uma fonte de luz lazer a qual é aplicada diretamente sobre a mídia e uma célula fotossensível que detecta se a luz projetada reflete ou não na mídia. Na verdade, para que a luz possa refletir na mídia, esta possui uma camada de material reflexivo colocada após a camada em que os dados estão gravados, no lado oposto em que incide o lazer. Dependendo, assim, do dado gravado, a luz aplicada pelo laser pode ou não transpassar a região em que o laser é aplicado, refletindo assim na camada reflexiva, e retornando em seguida para ser conduzida pelo espelho prismático à célula fotossensível.

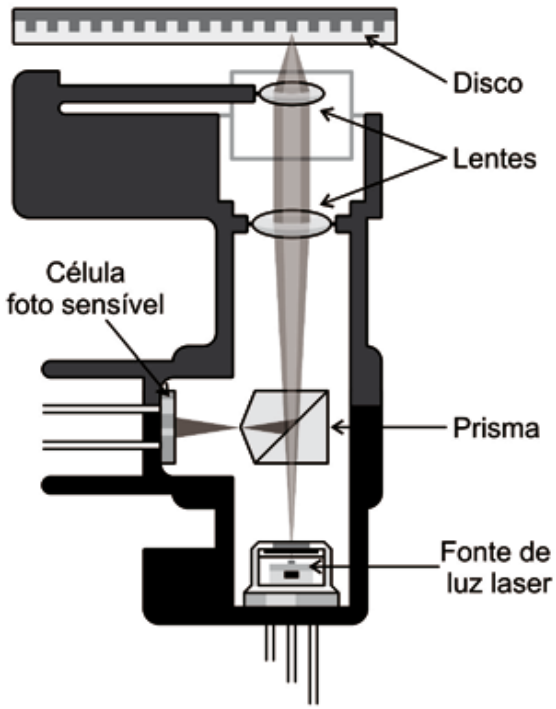

Figura 12 - Estrutura de uma cabeça ótica de leitura e escrita

Do ponto de vista do método de gravação, existem atualmente no mercado três tipos de mídias óticas. As mídias apenas para leitura, também conhecidas como mídias ROM, as mídias graváveis, conhecidas como mídias R, e as mídias regraváveis, conhecidas como mídias RW.

As mídias ROM tem o seu conteúdo gravado por processo de prensagem no seu substrato durante a sua fabricação. A estrutura de

uma mídia ROM com dupla face e dupla camada de gravação pode ser vista na Figura 13, a seguir. Como você pode observar, esta mídia é formada por várias camadas de policarbonato, o qual é moldado internamente com a informação a ser gravada, por este motivo diz-se que os dados gravados em uma mídia ótica deste tipo não podem ser afetados por arranhões ou riscos acidentais em sua superfície, uma vez que estão gravados no interior do disco.

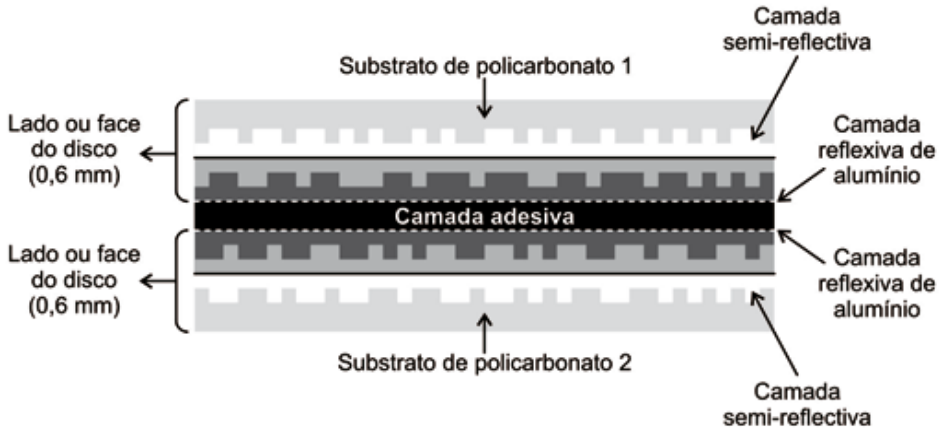

Figura 13 - Estrutura de uma mídia ROM

Tanto as mídias R como as RW possuem uma estrutura interna conforme podemos ver na Figura 14, a seguir. Nestes tipos de mídias tanto o processo de leitura quanto o de gravação é ótico, sendo feito a partir da aplicação de luz lazer. Como podemos ver pela figura, estas mídias possuem 4 camadas. A primeira, que serve de suporte às demais, é um substrato de policarbonato. A segunda camada é a camada fotossensível. A terceira camada é a camada reflexiva, e, por último, temos mais uma fina camada de verniz que serve de proteção à camada reflexiva.

Nas mídias tipo R, a camada fotossensível recebe uma aplicação de algum corante fotossensível, normalmente cianina ou ftalocianina. Desta forma, ao receber uma aplicação da luz lazer de alta intensidade, o que ocorre durante o processo de gravação, o corante modifica a sua cor, tornando-se opaco, permitindo assim a gravação da informação na mídia. Durante o processo de leitura aplica-se uma luz lazer de baixa intensidade, a qual não é suficiente para sensibilizar o corante presente na camada fotossensível.

Nas mídias RW, utiliza-se uma mistura de prata, índio, antimônio e telúrio no lugar do corante da camada fotossensível. Esta liga tem a propriedade de mudar o seu estado de cristalino para amorfo

quando exposto à luz laser intensa, o que ocorre durante o processo de gravação, e de retornar ao estado cristalino se exposto a luz laser moderada, o que ocorre durante o processo de apagamento do disco. O restante do processo de leitura e escrita é semelhante aos demais tipos de mídias óticas.

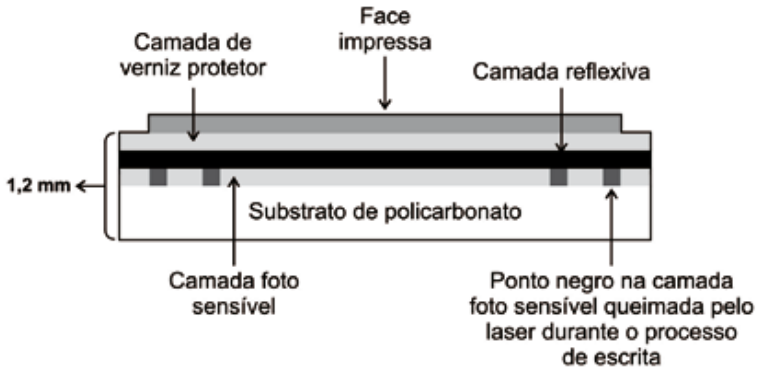

Figura 14 - Estrutura das mídias tipo R e RW

Por fim, vale um comentário a respeito das diferenças existentes entre os CDs, DVDs e Blue Rays.

A princípio, todos funcionam da mesma forma, conforme já foi descrito. A grande diferença está no comprimento de onda do lazer utilizado e no tamanho da região física utilizada para gravar um bit de informação. A Figura 15, a seguir, demonstra estas diferenças.

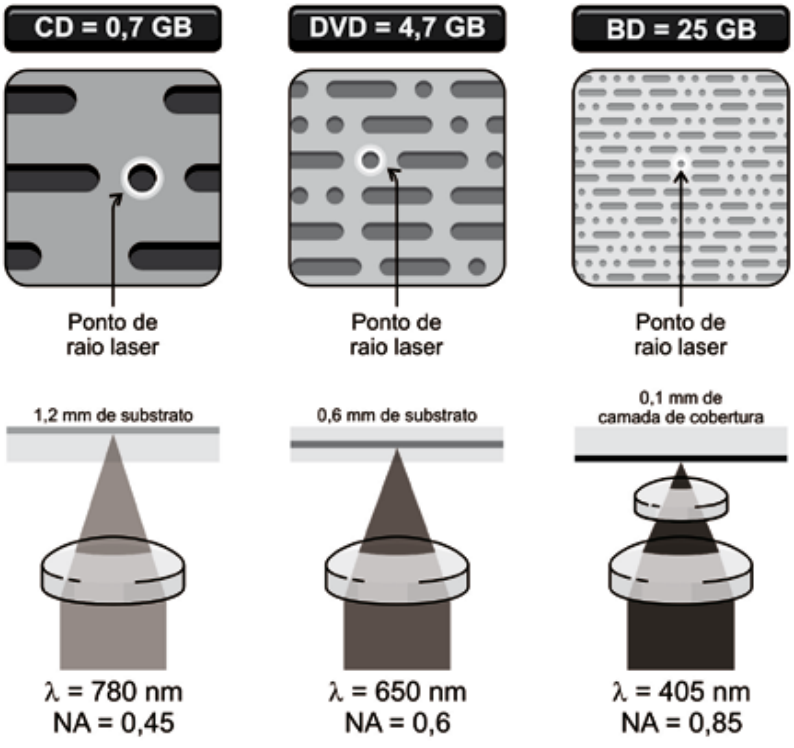

Figura 15 - Diferenças entre CDs, DVDs e Blue Rays

Fita Magnética

As fitas Magnéticas são a mais antiga forma de armazenamento de dados ainda em uso. Este meio de armazenamento foi inventado no início da década de 50 do século passado.

Assim com os discos magnéticos as fitas magnéticas utilizam a mudança da orientação magnética de algum tipo de material ferri-magnético como meio de gravar a informação na mídia suporte.

Apesar de toda evolução dos outros meios de armazenamento, as fitas Magnéticas ainda apresentam inúmeras vantagens, tais como: grande capacidade de armazenamento, baixo custo por unidade armazenada, longa expectativa de vida e a confiabilidade na retenção dos dados ao longo de sua vida útil. Por este motivo ainda são um dos meios mais utilizados como meio de armazenamento removível para grandes volumes de informação e onde se necessita garantir maior segurança aos dados armazenados.

Saiba Mais

[3] Conheça um pouco da história dos pen-drivers no link http://pt.wikipedia.org/wiki/USB_flash_drive

Pen-Driver

Conhecido pelos nomes de Pen-Driver, USB Flash Driver, Flash Driver, Thumb-driver entre outros nomes, este dispositivo faz parte da nova geração dos meios de armazenamento, construídos com o uso de memória não volátil baseada em semicondutor[3]. A Figura 16, a seguir, nos traz o formato de um pen-driver típico, ainda que hoje em dia não é incomum ver pen-drivers com as mais variadas formas.

Desde o seu lançamento no ano de 2000 pela IBM, os pen-drivers evoluíram substancialmente, passando de 8MB para atuais 128 GB de memória.

Devido a sua flexibilidade de uso, alta capacidade de armazenamento e à redução de custo pelo qual tem passado, os pen-drivers são a mídia removível preferida pela maioria dos usuários de computadores pessoais.

Figura 16 - Pen-Driver tipico

Conforme podemos ver pela **Figura 17,** a seguir, a estrutura interna de um pen-driver é bastante simples, composta basicamente de um controlador UBS e de um ou dois chips de memória FLASH[4].

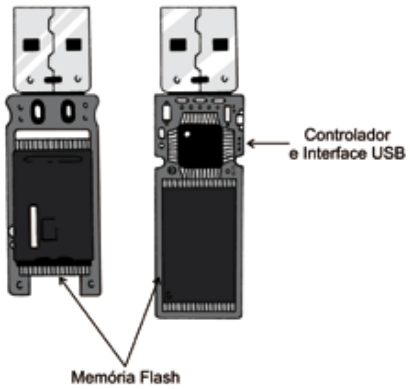

Figura 17 - Estrutura interna de um pen-driver

Saiba Mais

[4] Você sabe o que é uma memória FLASH? Conheça um pouco mais sobre este tipo de memória visitando a página

http://pt.wikipedia.org/wiki/Mem%C3%B3ria_flash

Aprenda Praticando

Chegou a hora de você por em prática o que aprendeu neste capítulo. A seguir temos alguns exercícios que visam consolidar o conteúdo que acabamos de apresentar.

1) Responda com suas próprias palavras:

a. A que conclusão as três afirmações apresentadas no início deste capítulo nos levam no tocante a quantidade e a velocidade de acesso dos elementos de memória de um computador ideal?

b. Qual o objetivo da hierarquia de memória? Como ela funciona?

c. Que princípio norteou a organização da hierarquia de memória? O que define o nível que um determinado elemento irá ocupar na hierarquia?

d. O que é memória primária e memória secundária? Como podemos identificar facilmente se um determinado elemento de memória pertence a memória primária ou a memória secundária? O que determina a qual destas classes um determinado elemento de memória pertence?

e. Como deveria estar organizada uma memória com 1024 bits considerando que a mesma deverá ser conectada a um processador que utiliza palavras de 64 bits?

f. Considere uma memória de acesso aleatório. Qual o tempo médio necessário para transferir 4096 bytes considerando um tempo médio de acesso de 10 ns e um tempo de ciclo de memória de 12 ns, considerando ainda que os dados estão agrupados em um único pacote com 4096 bytes.

   Resposta: T = 4096 * 8 * 12 ns = 393,22 μs

g. Considere uma memória de acesso seqüencial. Qual seria o tempo médio necessário para transferir 4096 bytes considerando um tempo médio de acesso de 10 ns e um tempo de ciclo de memória de 6 ns, considerando ainda que os dados estivessem agrupados em pacotes não consecutivos de 256 bytes.

   Resposta: Como queremos transferir 4096 bytes em pacotes de 256 bytes, teremos que efetuar 16 operações de transferência. Desta forma temos:

   $$T = 16 * (10ns + (256 * 8 * 6ns)) = 196,77\ \mu s$$

h. Considere a mesma memória do quesito anterior. Qual seria o tempo médio necessário para transferir 4096 bytes, considerando que os dados estivessem agrupados em pacotes não consecutivos de 16 bytes.

i. Considere agora um outro tipo de memória de acesso seqüencial, a qual tivesse um tempo de acesso de 15 ns e uma taxa de transferência de 200 MBits/Seg. Qual seria o tempo de necessário para transferir os mesmos 4096 bytes para esta memória se o mesmo tivesse que ser transferido em pacotes de 128 bytes?

j. Quais as técnicas mais empregadas para aumentar a capacidade e a velocidade de acesso dos discos rígidos?

k. O que é e para que serve o ECC?

l. Que diferenças você pode destacar entre os discos RAID 0 e RAID 1?

m. Qual é o meio de gravação e leitura dos dados em um disco ótico?

n. Qual a diferença entre a trilha de um disco ótico e a trilha de um disco magnético?

o. Por que a velocidade de rotação dos discos óticos varia durante um processo de leitura ou gravação?

p. De um modo geral, como são gravados os dados nos discos óticos tipo R e RW? Desconsidere as diferenças químicas entre os elementos foto sensíveis utilizados em cada tipo de mídia.

q. Na sua opinião, a que classe de memória, primária ou secundária, pertencem os discos virtuais? Justifique a sua resposta.

2) Associe as informações presentes na primeira coluna com as características dos elementos de armazenamento presentes na segunda coluna:

( ) Conexão direta ao barramento de memória

( ) Expressa em Bytes

( ) Determina como a informação é transferida (Tamanho do pacote de dados)

( ) Posição N só pode ser acessada após acessar a posição N-1

( ) Método de acesso utilizado nas memórias RAM

( ) Permite que se associe um endereço diferente a cada posição de memória

( ) Pode ser calculado pelo tempo de acesso ou pelo tempo de ciclo de memória ou pela taxa de transferência

( ) Pode ser utilizada para construir memórias capacitivas ou baseadas em Flip-Flops

( ) Tecnologia normalmente empregada na construção de elementos de memória secundária

( ) Os dados devem ser mantidos a partir de estímulos externos

( ) Palavras com 16, 32 ou 64 bits

(1) Localização

(2) Desempenho

(3) Capacidade

(4) Tecnologia de semicondutor

(5) Tecnologias ótica e magnética

(6) Unidade de Transferência

(7) Acesso Randômico

(8) Memórias voláteis

(9) Organização interna

(10) Acesso sequencial

(11) Endereçamento associativo

Conheça Mais

Em nossas referências bibliográficas você poderá encontrar inúmeras informações interessantes sobre os assuntos que acabamos de estudar. Se desejar ter uma visão tanto da perspectiva histórica quanto das possíveis tecnologias que deverão tem um grande impacto nesta áreas, leia também:

> "História das memórias de computador (em inglês)" *http://inventors.about.com/od/rstartinventions/a/Ram.htm*
>
> "História dos discos rígidos" *http://pt.wikipedia.org/wiki/Disco_r%C3%ADgido#Hist.C3.B3ria_do_disco_r.C3.ADgido*
>
> "Kingston lança primeiro pen-driver de 128G" *http://www.guiadohardware.net/noticias/2009-06/4a37cec2.html*
>
> "Memristor - Tecnologia inovadora que deverá revolucionar a hierarquia de memória (em inglês)" *http://www.embedded.com/210201673?cid=NL_embedded*

Atividades e Orientações de Estudo

Como você deve ter percebido, neste capítulo o nosso assunto foi muito mais teórico do que prático, o que vai exigir que você dedique um pouco mais de tempo para revisar o assunto a fim de fixar melhor o conhecimento aprendido.

Procure manter sempre um bom ritmo de estudo e você não terá dificuldades com esta parte do assunto.

Consulte sempre os tutores da disciplina, eles são as pessoas mais indicadas para tirar as suas dúvidas.

Vamos Revisar?

Neste capítulo, você conheceu a hierarquia de memória e seus principais componentes. Viu que o sistema de memória dos computadores atuais é formado conjuntamente pela memória primária

e pela memória secundária. E que a memória primária é formada pelos elementos de memória que estão conectados diretamente ao barramento de memória do processador, ao passo que a memória secundária é formada pelos elementos de memória que estão conectados ao processador indiretamente através do barramento de entrada e saída.

Neste capítulo, você também pode perceber que o objetivo da hierarquia de memória é apresentar ao processador um sistema de memória com uma velocidade de acesso próxima à da memória cache e com um espaço de armazenamento tão grande quanto o disponível nos elementos da memória secundária.

Você aprendeu também que a memória primária é formada pelo conjunto de registradores, pela memória cache e pela memória principal. E que a memória secundária, atualmente, é formada pelo disco rígido, pela unidades óticas (CD, DVD e Blue-Ray), pela fita magnética e pelos pen-drivers.

Por fim, você conheceu os detalhes de construção e a tecnologia empregada em cada um dos elementos de memória apresentados, indo desde as memória baseadas em semicondutores, passando pela memória de armazenamento em mídia magnética e concluindo com a memória de armazenamento ótico.

Capítulo 2

O que vamos estudar?

Neste capítulo, vamos estudar os seguintes temas:

» A memória cache e suas políticas de leitura e escrita de dados

» Troca de dados entre as estruturas de memória dos computadores

» Mapeamento de dados entre as memórias principal e cache

» Políticas de substituição de dados na memória cache

Metas

Após o estudo deste capítulo, esperamos que você consiga:

» Entender qual o objetivo da cache no subsistema de memória;

» Compreender como acontecem os diversos tipos de mapeamento de dados entre a memória principal e a memória cache;

» Conhecer as políticas de escrita e substituição de dados em cache.

Capítulo 2 – Memória Cache

Vamos conversar sobre o assunto?

Antes de começar a estudar a disciplina infraestrutura de hardware, você já tinha ouvido falar em memória cache? Possivelmente sim, mas o que talvez você não saiba é o porquê de seu uso e as suas principais funcionalidades. Isso é o que veremos nesse capítulo.

Vamos estudar a memória cache e como os dados são trocados entre as memórias cache e principal. Para iniciarmos a nossa conversa, podemos dizer que a cache é uma memória menor e mais rápida que a memória principal, estando localizada entre os registradores e a memória principal de acordo com a hierarquia de memória que estudamos no Capítulo 1. Por estar entre a CPU e a memória principal, normalmente, os dados são trocados da CPU com a cache e desta com a memória principal. Devido ao seu tamanho ser menor que a memória principal, a cache só contém cópia de porções da memória principal. Podemos observar a localização da cache no esquema apresentado na Figura 1.

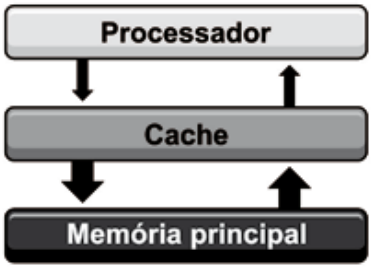

Figura 1 - Cache x Principal - Memória Cache: Elementos de memória intermediário entre o processador e a Memória Principal

Você deve estar se perguntando qual o objetivo de se criar uma memória menor que a principal, já que ela não poderia comportar todos os dados da principal. A principal motivação para a criação da cache foi acelerar a velocidade de transferência das informações entre CPU e a memória principal, aumentando o desempenho dos sistemas.

Pelo fato de ser menor, a memória cache armazena de instruções e dados de acordo com algum algoritmo de substituição. Por exemplo, pode-se utilizar um algoritmo que mantenha na cache apenas as informações mais frequentemente acessados do programa em execução. Esses algoritmos de substituição de dados serão abordados nesse capítulo em seções posteriores. O importante agora é que você entenda a filosofia da cache e o seu princípio básico de funcionamento.

Acesso a Dados na Memória Cache

Como a cache fica localizada entre a CPU e a memória principal, todo e qualquer informação acessada será incialmente procurada na cache. Somente no caso da informação não estar armazenada na cache é que a busca se dará na memória principal. Essa técnica naturalmente aumenta o desempenho dos sistemas computacionais, uma vez que a cache é menor que a principal. Sendo assim, procurar dados em um espaço menor tende a ser mais rápido que procurar dados num espaço maior.

Quando um dado não pôde ser encontrado na cache, significa que aquela informação não está lá, estando, portanto, apenas na principal ou no disco. Como esses dois últimos tipos de memória estão mais abaixo da cache na pirâmide de hierarquia, significa que são mais lentas, embora maiores. Sendo assim, o fato da busca por informações se dá inicialmente na cache reduz o número de acessos à principal e aos discos aumentando o desempenho dos sistemas computacionais.

Vamos observar na Figura 2, um fluxograma indicando como ocorre a leitura de dados na memória. No primeiro momento, a CPU solicita a leitura de um determinado endereço em memória. Como falamos

anteriormente, esse dado ou palavra será buscada inicialmente na cache. Por isso, o fluxograma apresenta o losango da estrutura de decisão onde é feita a pergunta se aquela palavra procurada está na cache. Caso positivo (o losango aponta a saída **Sim**), o conteúdo do endereço será buscado e entregue à CPU. Caso negativo (o losango aponta a saída **Não**) a informação a ser buscada deve ser acessada na memória principal. Porém, para trazer esses dados, a informação deve passar da MP para a cache e desta para a CPU. Assim, um espaço (aqui chamaremos de slot) será alocado na cache para comportar a informação, e em seguida, o dado será carregado na cache para posteriormente seguir para a CPU.

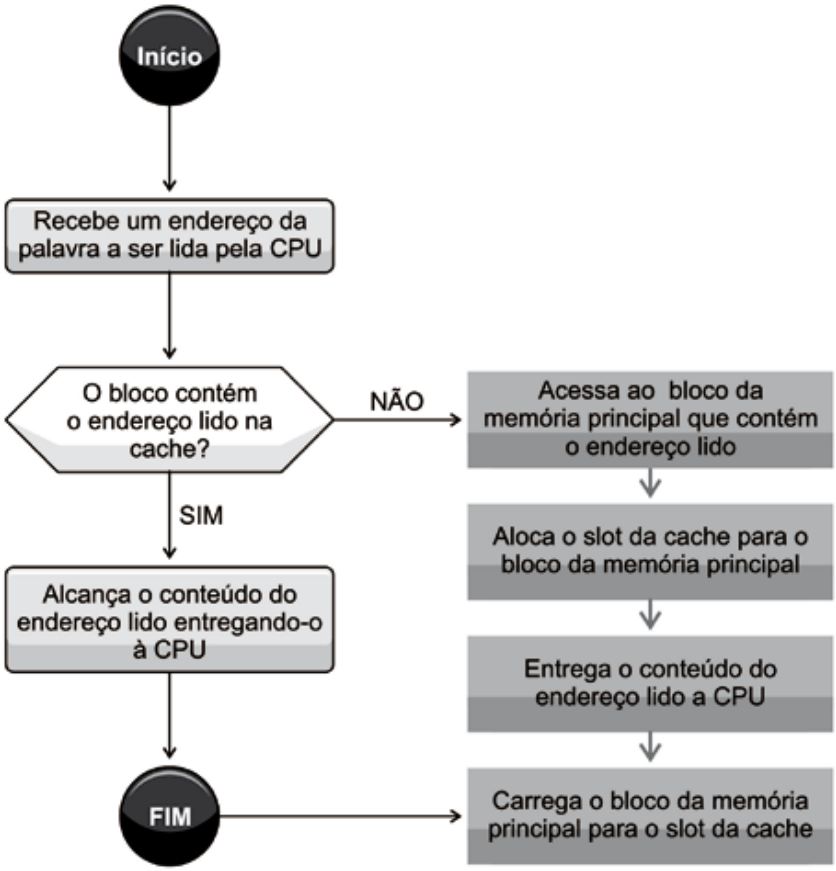

Figura 2 - Operação de Leitura na Cache

A memória principal apresenta a uma organização dividida em blocos. Um bloco é composto por um conjunto de palavras. Cada palavra possui um endereço para que a mesma possa ser alcançada e este endereço armazena um conteúdo, que pode ser dado ou instrução. Já para a memória cache, a organização se dá em linhas, também denominadas *slots* e cada *slot* (linha) armazena um conjunto de palavras. Você deve estar pensando que é coincidência

o fato de que tanto uma linha da cache quanto um bloco da memória principal armazenarem um conjunto de palavras. Entretanto, não é coincidência, e sim uma obrigatoriedade. Para que o subsistema de memória funcione eficientemente, o número de palavras de um bloco da memória principal deve ser igual ao número de palavras de uma linha da cache. Você pode observar na Figura 3 a divisão da memória principal e da memória cache.

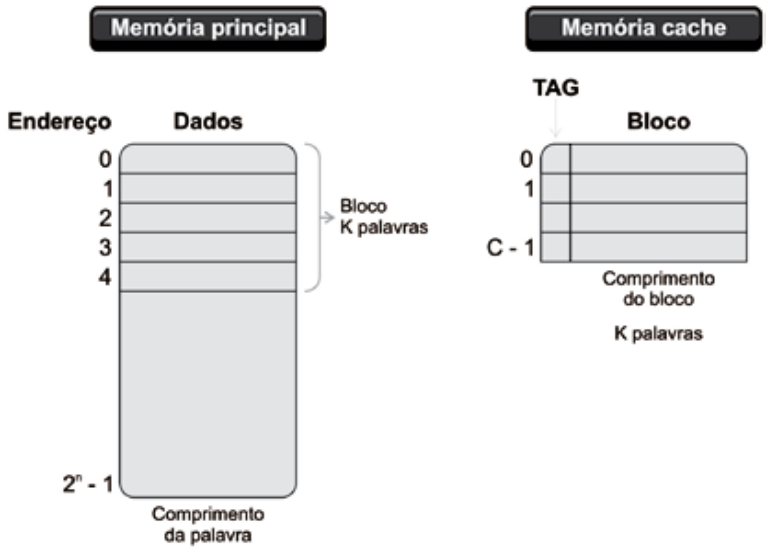

Figura 3 - Estrutura da MP e da Cache

Ao observar a Figura 3, notamos que a memória principal possui 2^n palavras endereçáveis. Cada palavra tem um endereço de n bits. Existe um número fixo de blocos com K palavras. A memória cache, por sua vez, apresenta C slots ou linhas, e cada um possui K palavras. Note que o número de palavras de uma linha da cache é igual ao número de palavras de um bloco da memória principal. Para que esse esquema funcione com eficiência, o número de linhas da cache deve ser bem menor que o de blocos da memória principal, já que a memória principal deve ser bem maior que a cache. Lembre-se que a a memória cache só pode ter parte dos dados da memória principal.

Sendo assim, se nem todos os dados da memória principal estão na cache e se inicialmente os dados são procurados na cache (vide Figura 2), como o processador poderá identificar se o dado procurado está na cache?

É importante salientar que o processador armazena em seus registradores, o endereço de uma palavra da memória principal e não da memória cache. Entretanto, o dado só é procurado na memória principal após não ser encontrado na cache. Mas se o dado está na

cache, como acessá-lo de forma rápida, uma vez que só se conhece o seu endereço da memória principal?

A solução para esse problema é o que chamamos de **mapeamento de memória**. Você já ouviu falar disso? Essa técnica existe para possibilitar a descoberta da localização de um bloco da memória principal entre as linhas da cache. Isso é necessário já que o número de linhas da cache é bem menor que o número de blocos da MP. Dessa forma, não existe uma linha da cache dedicada para cada bloco da MP. Caso isso acontecesse, o número de linhas da cache deveria ser igual ao número de blocos da MP e isso iria de encontro ao princípio básico que vimos no capítulo 1 , que pela hierarquia de memória, a cache fica acima da MP na pirâmide de hierarquia, sendo mais rápida e menor que a MP.

Agora que você já sabe o porquê da necessidade do mapeamento, vamos estudar três maneiras diferentes para se mapear de dados entre a MP e a cache. São eles: mapeamento direto, mapeamento associativo e mapeamento associativo por conjunto.

Mapeamento de Memória

A primeira técnica que vamos estudar é a mais simples e é denominada Mapeamento Direto. Através dele, cada bloco da memória principal é mapeado em uma linha específica da cache.

O nosso problema inicial é localizar uma determinada palavra na cache, conhecendo apenas o seu endereço na MP. Lembre-se que a troca de dados entre os registradores e a memória cache é feito palavra a palavra, enquanto que a troca de dados entre a MP e a cache é feito por bloco (conjunto de palavras). Sabendo qual o endereço da MP que contém a palavra procurada, é possível saber qual bloco da MP a contém. Conhecendo-se o bloco na MP, pode-se utilizar as técnicas de mapeamento direto para se calcular qual a linha da cache armazena aquele bloco e consequentemente, a palavra procurada.

Para se calcular qual é a linha específica da cache que um determinado bloco da MP se encontra, o subsistema de memória utiliza uma equação matemática, quando o mapeamento é direto:

i = j modulo m onde

i = linha da cache onde o bloco está armazenado

j = número do bloco da memória principal

m = número total de linhas na cache

Apenas para relembrar, a operação aritmética módulo retorna o resto da divisão inteira de **j** por **m**. Perceba que **j** é o número do bloco da memória principal que se deseja encontrar. Como ele está numa determinada linha na cache, precisa-se encontrar que linha é essa, no nosso caso, representado pela equação por **i**. A variável **m** refere-se ao total de linhas da cache.

Para ficar mais fácil de compreender, vamos observar o exemplo apresentado na Figura 4. Deseja-se descobrir onde está localizado na cache o bloco da MP de número 12. Sabe-se que a memória cache utiliza o mapeamento direto e que possui 8 linhas ao total.

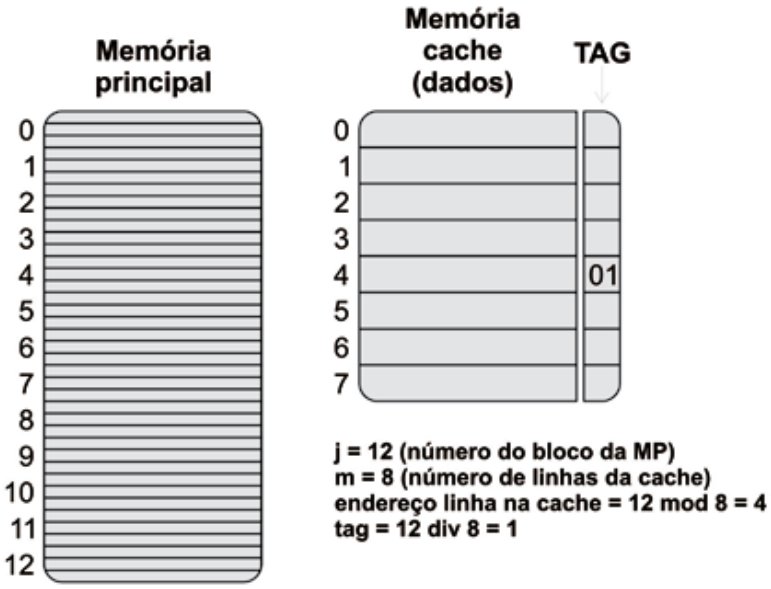

Figura 4 - Mapeamento Direto

Aplicando-se a equação **i = 12 modulo 8** , encontramos 4 como resultado, significando que o bloco 12, caso esteja na cache, deverá estar ocupando a linha 4, já que o mapeamento direto obriga que um determinado bloco da MP ocupe uma linha específica na cache. Vamos pensar agora que desejamos descobrir onde o bloco de número 20 da MP encontra-se na cache. Aplicamos novamente a equação **i = 20 modulo 8** e novamente encontramos 4 como resultado. Assim, o bloco 20 da MP também ocupará a linha 4 caso ele esteja na cache.

Nesse momento, você deve estar se perguntando, como os 2 blocos podem ocupar a mesma linha na cache. Eles podem sim, desde que em momentos diferentes. Nunca poderão estar os dois blocos ao mesmo tempo, já que a quantidade de palavras de um bloco é equivalente a quantidade de palavras comportadas por uma linha da cache. Nós consideramos o mapeamento como direto, porque sempre que o bloco 12 estiver na cache nessa organização de memória, o mesmo estará na linha 4. O mesmo dizemos do bloco 20 e também do 28... Isso comprova a nossa afirmação inicial, que cada bloco da memória principal é mapeado em uma linha específica da cache.

Neste momento, você deve estar com um novo questionamento: **como o subsistema de memória saberá se num momento, é o bloco 12, bloco 20 ou 28 que está na linha 4 da cache?**

Para isso, o esquema de mapeamento direto faz uso de uma outra informação, que é denominada **TAG** e está contida no registrador de endereço (MAR).

A **TAG** para o mapeamento direto é calculada pela seguinte equação:

TAG = j div m onde

TAG = número do bloco armazenado naquela linha

j = número do bloco da memória principal

m = número total de linhas na cache

Você consegue perceber qual a diferença de uma equação para a outra? A primeira utiliza a operação módulo que representa o resto da divisão inteira. A segunda utiliza a operação div que representa o quociente da divisão. Assim, a TAG será obtida através da obtenção do quociente entre o número do bloco da MP e o número total de linhas da cache.

Pensando no exemplo da Figura 4, para o bloco de número 12, seria calculada a TAG = 1, enquanto para o bloco de número 20, seria calculada a TAG = 2. A TAG = 3 equivaleria ao bloco de número 28. Quando se escreve esses números em binário, se obtém a representação na coluna na TAG, da linha 4. Essa TAG é o que define que no momento específico, quem está na linha 4 da cache é o bloco 12 e não o bloco 20.

Dessa forma, poderemos definir TAG como um campo para indicar

que bloco em particular da memória principal está sendo armazenado naquela linha da cache. Ele sempre será necessário, pois uma linha da cache não pode ser dedicada a um único bloco, já que o número de linhas da cache é bem inferior ao de blocos da MP.

Vamos agora resolver um exercício que consolidará esses conceitos que estudamos sobre mapeamento direto. Vamos considerar uma cache que trabalha com mapeamento direto e possui tamanho total de 64 Kbytes. A memória principal, por sua vez, tem capacidade total 16Mbytes e os seus endereços possuem 24 bits. O tamanho do bloco da memória principal é 4bytes. Descubra então qual é o formato do endereço da memória principal.

Para resolver esse exercício o que você precisa saber da teoria? Em primeiro lugar, você deve lembrar que o MAR é um registrador (memória interna à CPU) que armazena endereços de palavras da memória principal. Em segundo lugar, você precisa lembrar que o mapeamento direto transfere um dado bloco da memória principal para uma linha específica da cache. Também é importante lembrar que o tamanho do bloco da MP é o mesmo da linha da cache, ou seja, em outras palavras, o número de bytes que cabem no bloco da MP é o mesmo comportado por uma linha da cache.

Acrescentaremos ainda uma informação adicional com relação a divisão de campos do MAR para o mapeamento direto: O MAR precisará conter um conjunto de bits para representar a linha da cache ocupada pela palavra procurada; um outro conjunto de bits para representar a TAG (citado previamente utilizado para identificar qual é o bloco que ocupa aquela linha, naquele momento); e por fim, um último conjunto de bits para representar a própria palavra, dentro da linha, uma vez que uma linha da cache comporta um conjunto de palavras. Em síntese, o MAR, para o mapeamento direto, está subdividido em três agrupamentos de bits, como podemos observar na Figura 5.

Figura 5 - MAR para o mapeamento Direto

Voltando ao nosso exemplo original, já estamos cientes da base teórica para resolver o exercício. Dessa forma, precisamos descobrir

a quantidade de bits de cada um desses grupos (tag, linha e palavra) e a quantidade total dos bits do MAR. Essa última informação, foi nos passada no exemplo: observe a frase “A memória principal, por sua vez tem capacidade total 16Mbytes e os seus endereços possuem 24 bits”. Isso significa que o MAR possui 24 bits agrupados em três campos. Precisamos então descobrir a quantidade de linhas disponíveis na memória cache. Como faremos isso? Já sabemos o tamanho total da cache e também sabemos quantos bits cada linha comporta (lembre-se que uma linha tem a mesma capacidade de um bloco da MP). Dividindo-se o tamanho total da cache (64 kbytes) pela quantidade de bytes de 1 linha (4 bytes) chegamos a 64Kbytes/4 bytes = 16 K linhas.

Note que são 64 Kbytes e não 64 bytes. Por isso, você não poderá eliminar o **K, pois ele representa a ordem de grandeza 1000**. Entretanto, devemos utilizar a notação em binário o que significa que $1K = 1024 = 2^{10}$.

Ao colocarmos 16K em potência de 2, obtemos $16\ K = 2^4 \cdot 2^{10} = 2^{14}$. Isso significa que temos 2^{14} linhas possíveis na cache. Um bloco da MP pode ocupar alguma dessas linhas e para representá-las serão necessários 14 bits. Em outras palavras, você deverá observar o expoente da potência de 2 da quantidade de linhas. No nosso caso, já descobrimos que o campo slot/linha deverá ter 14 bits.

Vamos agora tentar descobrir quantas TAGs existem e consequentemente, quantos bits são necessários para representá-las. A TAG irá indicar qual bloco está naquela linha num dado momento. Para isso, deveremos dividir o número de blocos da MP pelo número de linhas da cache. Essa última informação, acabamos de descobrir (2^{14}). O número de blocos, ainda não temos, pois nos foi dado o tamanho da MP e a capacidade de um bloco. Ora, mas essas duas informações, já são suficientes para a descoberta do número de blocos. Basta dividirmos a capacidade toral da MP pela capacidade de 1 bloco, então teremos a quantidade de blocos.

Assim teremos 16Mbytes/4bytes = 4M. Entretanto, 1M equivale a 1K x 1K, ou seja $2^{10} \times 2^{10} = 2^{20}$. Assim, para esse caso, teremos $4M = 4 \times 2^{20}$ blocos.

Mas a nossa busca pelo número de bloco objetivou descobrimos o número de TAGs. Para descobrir o número de TAGs vamos dividir o número de blocos pelo número de linhas. Assim temos o seguinte

cálculo:

número de TAGs = 4 x 2^{20} blocos/2^{14} linhas

número de TAGs = 2^2 x 2^{20} blocos/2^{14} linhas

número de TAGs = 2^{22} blocos/2^{14} linhas

número de TAGs = 2^2 x 2^{20} blocos/2^{14} linhas

número de TAGs = 2^8 TAGs

Um bloco da MP deverá ter uma TAG para identifica-lo na cache e para representar essa TAG serão necessários 8 bits. Em outras palavras, você deverá observar o expoente da potência de 2 da quantidade de TAGs. No nosso caso, já descobrimos que o campo TAG deverá ter 6 bits.

Com isso poderemos descobrir o número de bits para representar a palavra, calculando o que falta para completar os 24 bits do MAR. Temos 14 para representar a linha e 8 para TAG, totalizando 22 bits. Como o MAR possui 24 bits, concluímos que os 2 bits restantes representam a palavra. O formato do MAR para a cache do exemplo, utilizando mapeamento direto é ilustrado na Figura 6.

Figura 6 - Formato do MAR

De acordo com o exemplo estudado, verificamos que quando uma palavra precisa ser procurada na memória, o subsistema de memória faz a leitura do endereço no barramento de endereços (conforme estudado no volume 2). Esse endereço está armazenado no MAR, registrador de endereço da CPU. Para interpretar a informação do MAR e fazer inicialmente a busca pela palavra na memória cache, caso o MAR contenha a informação :000000010000000000111011, o subsistema de memória interpretará esse endereço da seguinte maneira:

- Os 8 primeiros bits (00000001) indicam a TAG de número 0

- Os 14 bits seguintes (00000000001110) indicam a linha de número 14

- Os 2 últimos bits (11) indicam a palavra de número 3.

Assim a informação a ser buscada é a palavra 3 da linha 14, e identificado pela TAG 0.

Essa interpretação se dá devido ao tipo do mapeamento. Caso seja utilizado um outro tipo de mapeamento, a interpretação do MAR será diferente. O mapeamento direto foi o primeiro tipo estudado, mas estudaremos a seguir mais dois tipos de mapementos.

Para concluirmos o estudo deste primeiro tipo de mapeamento, vamos observar quais os pontos positivos e os pontos negativos. O mapeamento direto apresenta um custo em termos de hardware menor (circuitos mais simples e menos custoso), é uma técnica rápida, ganhando assim em velocidade na hora da busca por informações na cache, uma vez que os dados são procurados diretamente em uma linha específica, sem precisar varrer a cache completa.

Em se tratando dos pontos negativos, poderá se ter um mau aproveitamento do tamanho da cache, a depender dos endereços buscados. Para ficar mais claro de perceber isso, imagine naquele nosso exemplo anterior, se o bloco 12 fosse alocado para cache inicialmente. Ele iria para a linha 4, como vimos. Se imediatamente depois fosse acessado o bloco 20 e o mesmo não tivesse na cache, ele seria buscado na principal e trazido para a cache. O bloco 20 também seria alocado na linha 4, mesmo que a cache estivesse com outras linhas vazias. Isso comprova o mau aproveitamento da mesma. Além disso, o subsistema de memória reserva um espaço na cache para o controle de informações. Observe na Figura 7, uma síntese dos pontos positivos e negativos do mapeamento direto.

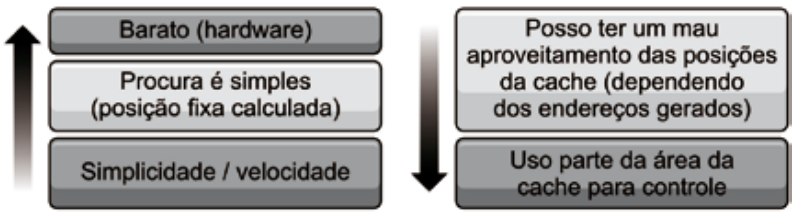

Figura 7 - Mapeamento Direto

A segunda técnica de mapeamento de memória que vamos estudar é denominada Mapeamento Associativo. Através dele, cada bloco da

memória principal pode ser mapeado em **qualquer** linha da cache.

O nosso problema inicial é localizar uma determinada palavra na cache, conhecendo apenas o seu endereço na MP. Lembre-se que a troca de dados entre os registradores e a memória cache é feito palavra a palavra, enquanto que a troca de dados entre a MP e a cache é feito por bloco (conjunto de palavras). Sabendo qual o endereço da MP que contém a palavra procurada, é possível saber qual bloco da MP a contém. Conhecendo-se o bloco na MP, pode-se utilizar as técnicas de mapeamento direto para se calcular qual a linha da cache armazena aquele bloco e consequentemente, a palavra procurada.

Para o mapeamento associativo, a lógica de controle da cache interpreta um endereço de memória simplesmente como um tag e uma palavra, conforme é possível observar na Figura 8. Como a linha a ser ocupada por um determinado bloco é aleatória, não há necessidade de um conjunto de bits no registrador de endereço (MAR) para representá-la.

Figura 8 - Mapeamento Associativo

Você deve estar se perguntando como o subsistema de memória descobre que um determinado bloco está numa linha. Normalmente, a TAG é o identificador que indica qual bloco está naquela linha. No mapeamento associativo isso também é verdade porém a TAG irá conter o número do bloco escrito na base binária. Observe na Figura 9, o mesmo exemplo da Figura 4 quando o mapeamento é o associativo. O bloco 12 teria como TAG, a sequência 01100.

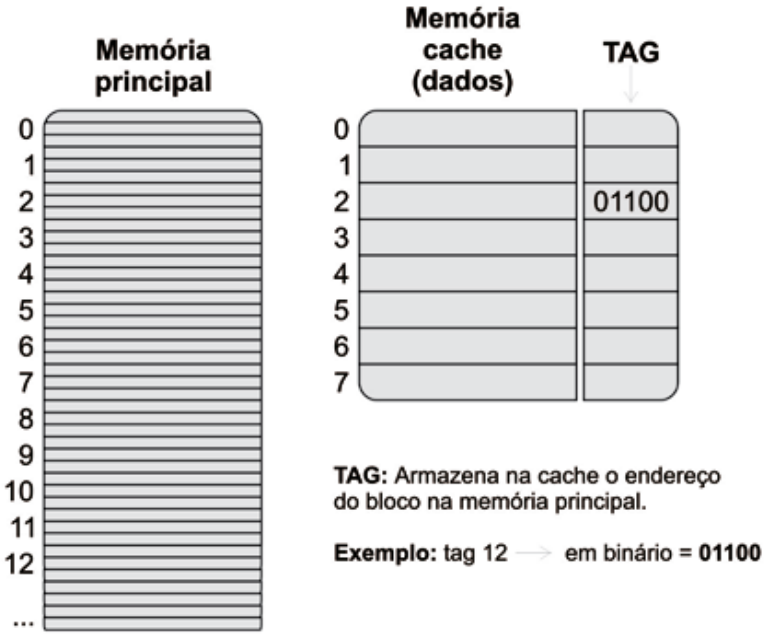

Figura 9 - Mapeamento Associativo

Na Figura 9, optamos por alocar o bloco 12 da MP na linha 2 da cache. Essa escolha foi aleatória. Poderíamos ter escolhido, por exemplo, a linha 7, 5 ou 0, já que o mapeamento é associativo.

Vamos agora resolver um exercício que consolidará esses conceitos que estudamos sobre mapeamento associativo. Como ficaria a divisão de bits do registrador de endereço (MAR) para uma cache associativa com 64 linhas (slots). Considere que o MAR possui 24 bits e que cada bloco da MP possui 64 palavras.

Aqui novamente recorreremos a teoria para resolver. Novamente, você deve lembrar que o MAR é um registrador (memória interna à CPU) que armazena endereços de palavras da memória principal. Em segundo lugar, você precisa lembrar que o mapeamento associativo transfere um dado bloco da memória principal para qualquer linha da cache. Também é importante lembrar que o tamanho do bloco da MP é o mesmo da linha da cache, ou seja, em outras palavras, o número de bytes que cabem no bloco da MP é o mesmo comportado por uma linha da cache. Por fim, lembre-se que a divisão de bits do MAR para o mapeamento associativo será em dois agrupamentos de bits. A TAG e a palavra.

Com relação aos dados do exemplo, o número de linhas da cache será irrelevante, já que esse dado não aparece no MAR. Já sabemos que o número total de bits do MAR é 24. Como cada bloco

da MP e consequentemente cada linha da cache possui 64 palavras, precisaremos de 6 bits para representar a palavra. Como chegamos nos 6 bits? Vamos escrever 64 palavras/bloco em potência de 2.

$64 = 2^6$ palavras em cada bloco ou em cada linha

Dessa forma, precisamos de 6 bits para representar as palavras de uma linha, restando 18 bits do MAR para armazenar informações de TAG, conforme observamos na Figura10.

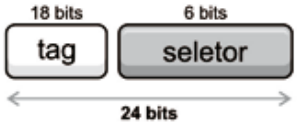

Figura 10 - Exemplo do MAR no Mapeamento Associativo

Vamos agora interpretar a mesma informação do MAR que fizemos no mapeamento associativo. Caso o MAR contenha a informação: 000000010000000000111011, o subsistema de memória interpretará esse endereço da seguinte maneira:

- Os 18 primeiros bits (000000010000000000) indicam a TAG de número 1024

- Os 6 bits seguintes (111011) indicam a palavra de número 59

Assim a informação a ser buscada é a palavra 59 do bloco 1024.

Vamos agora observar quais os pontos positivos e os pontos negativos deste tipo de mapeamento. O mapeamento associativo apresenta um custo adicional no momento em que se vai procurar uma informação na cache pois precisará comparar a TAG de cada linha com o número do bloco para tentar localizá-lo, já que o bloco procurado pode estar em qualquer linha. Isso poderia ser citado como um ponto negativo. Por outro lado, o que era um ponto negativo no mapeamento direto passa a ser um ponto positivo no associativo, pois a memória cache tende a ser melhor aproveitada. Como o bloco da MP pode ser alocado em qualquer linha da cache, o subsistema de memória irá procurar uma linha livre na cache para alocar um bloco da MP, no momento em que este for trazido para a cache.

Já quando a cache estiver lotada, uma linha precisará ser escolhida para alocar um novo bloco. Essa escolha é feita de acordo com algum

algoritmo de substituição de linhas na cache que veremos mais a diante. A depender do algoritmo escolhido, essa substituição poderá trazer benefícios ou não ao desempenho. Observe na Figura 11, uma síntese dos pontos positivos e negativos do mapeamento associativo.

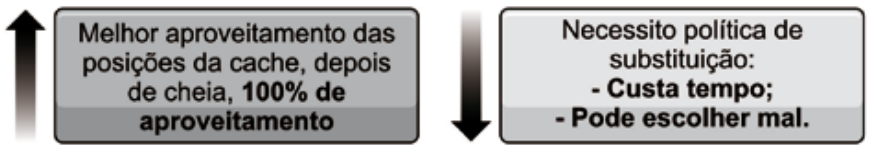

Figura 11 - Mapeamento Associativo

A terceira e última técnica de mapeamento de memória que estudaremos aqui é conhecida por Mapeamento Associativo por Conjunto. Através dele, a memória cache está subdividida em conjuntos cada bloco da memória principal pode ser mapeado em **qualquer** linha da cache.

Aprenda Praticando

Agora é a hora de você treinar o que foi apresentado no capítulo 2, por meio de exercícios. Serão apresentados 3 exercícios resolvidos, referentes aos três tipos de mapeamentos e em seguida, teremos os exercícios propostos. Inicialmente, tente resolver os exercícios resolvidos sem olhar as suas respostas. Após a resolução, confirme sua resposta com aquela apresentada nessa seção. Por fim, tente resolver os exercícios propostos. Não esqueça de resolver também aqueles que forem solicitados pelo seu professor formador como atividade somativa.

Exercícios Resolvidos

Exercício 1

Como ficaria a divisão de bits do registrador de endereço (MAR) para uma cache que utiliza mapeamento direto com 1024 linhas (slots). Considere que o MAR possui 32 bits e que cada bloco da MP possui 4 palavras.

Resolução: Como o mapeamento é direto, o MAR precisará conter um conjunto de bits para representar a linha da cache ocupada pela palavra procurada; um outro conjunto de bits para representar a TAG e por fim, um último conjunto de bits para representar a própria palavra, dentro da linha, uma vez que uma linha da cache comporta um conjunto de palavras.

Nos foi informado que o MAR possui 32 bits ao total. Vamos então calcular quantos desses 32 bits são necessários para representar as 1024 linhas da cache. Colocando 1024 como potência de 2, temos 2^{10} . Pegamos então o expoente da potência de 2 para representar o número de bits do MAR utilizados para endereçar as linhas da cache, ou seja 10. Em seguida vamos verificar quantos bits usaremos para representar as TAGs ou o número de bits para representar as palavras. Note que de posse de uma das duas informações, você automaticamente descobrirá a outra, uma vez que já conhece o número de bits total do MAR e das linhas.

Sendo assim, veremos no enunciado do exemplo o que ficará mais fácil para nós. Foi dito que **cada bloco da MP possui 4 palavras**. Dessa forma, colocando 4 em potência de 2, teremos 2^2 palavras. Pegando o expoente teremos 2 bits para representar as palavras. Com essa informação concluímos a questão, pois o MAR possui 32 bits, dos quais 10 representam a linha, 2 bits representam a palavra e os 20 bits restantes representam a TAG. A representação gráfica do MAR pode ser observada na Figura 12.

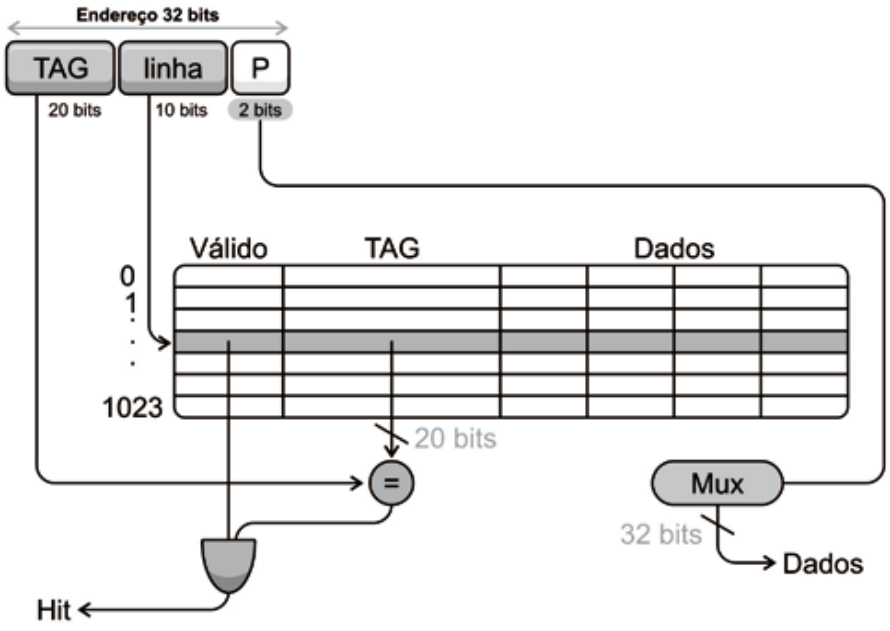

Figura 12 - Resolução do Exercício 1

Exercício 2

Como ficaria a divisão de bits do registrador de endereço (MAR) para uma cache associativa com 512 linhas (slots). Considere que o MAR possui 64 bits e que cada linha da cache possui 32 palavras.

Resolução: Para o mapeamento associativo, temos que o MAR está dividido em 2 conjuntos de bits: um para representar a TAG e outro para representar a palavra, já que não será necessário representar a linha, uma vez que no mapeamento associativo um bloco da MP qualquer ocupa aleatoriamente as linhas da cache.

O MAR possui 64 bits ao total. Se descobrirmos quantos bits serão utilizados para representar a TAG, subtrairemos dos 64 totais, obtendo, então o número de bits para representar a palavra. Se descobrimos primeiro o número de bits para representar a palavra, chegamos ao número de bits para TAG.

Verificamos pelo enunciado que **cada linha da cache possui 32 palavras**. Colocando 32 em potência de 2, temos 2^5 palavras. Assim, pegando o expoente da potência de 2, temos 5 bits para representar as palavras. Dos 64 bits do MAR, 5 são utilizados para a representação da palavra, sobrando 59 bits para as TAGs. Observe a representação do MAR na Figura 13.

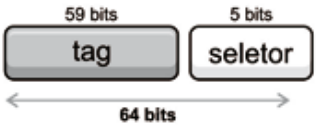

Figura 13 - Resolução do Exercício 2

Lista de Exercícios Propostos

1) Como ficaria a divisão de bits de endereço para uma cache mapeada diretamente com 2048 linhas que trabalhe com blocos de 8 palavras de 32 bits?

2) A memória cache possui 64 linhas (slots).A palavra contém 24 bits.Um bloco possui 64 palavras. Como ficaria a divisão de bits de endereço para esta cache considerando mapeamento direto?

3) Memória cache possui 512 linhas (slots). Cada slot possui 32 palavras e cada palavra possui 64 bits. Como ficaria a divisão de bits de endereço para esta cache, quando mapeada

diretamente?

4) Considere o mapeamento associativo e uma memória cache com 512 slots. Cada slot possui 32 palavras e cada palavra possui 64 bits. Como ficaria a divisão de bits de endereço para esta cache?

Atividades e Orientações de Estudo

Dedique, pelo menos, 6 horas de estudo para o Capítulo 2. Organize uma metodologia de estudo que inicie com a leitura dos conceitos e acompanhamento dos exercícios resolvidos.

Você poderá esclarecer suas dúvidas com o professor e os tutores utilizando os chats e os fóruns tira-dúvidas no ambiente virtual de seu curso.

Não esqueça de ler atentamente o guia de estudo da disciplina, pois nele você encontrará a divisão de conteúdo semanal, ajudando-o a dividir e administrar o seu tempo de estudo.

Observe os prazos estabelecidos pelo seu professor para essa atividades virtuais. Lembre-se que as atividades somativas propostas pelo professor no ambiente virtual são importantes para o aprendizado e para a composição da sua nota.

Referências

STALLINGS, William. **Arquitetura e Organização de Computadores.** 5. ed

PATTERSON, D. A. e Hennessy, John L. **Organização e Projeto de Computadores**. LTC, 2000.

TANENBAUM, Andrew S. **Organização Estruturada de Computadores.** 4. ed. Tradução Helio Sobrinho. Rio de Janeiro: Prentice-Hall, 2001.

Conheça os Autores

Juliana Regueira Basto Diniz possui graduação em engenharia eletrônica pela Universidade Federal de Pernambuco, mestrado e doutorado em Ciência da Computação pela Universidade Federal de Pernambuco. Atualmente é professora da Universidade Federal Rural de Pernambuco (UFRPE), desenvolvendo trabalhos no grupo de Educação a Distância desta universidade. Seus temas de interesse em pesquisa são: Sistemas Distribuídos, Computação Ubíqua e Ensino a Distância.

Abner Corrêa Barros é mestre em Ciência da Computação com foco em Engenharia de Hardware pelo Centro de Informática da Universidade Federal de Pernambuco. Possui graduação em Ciência da Computação pela mesma universidade. Atualmente é professor da disciplina de Organização e Arquitetura de Computadores da Faculdade Maurício de Nassau e Engenheiro de Hardware da Fundação de Apoio ao Desenvolvimento da UFPE (FADE), atuando em um projeto de convênio entre o Centro de Informática da UFPE e a Petrobrás. Suas áreas de interesse e pesquisa são: Hardware Reconfigurável, Arquitetura de Cores Aritméticas e Computação de Alto Desempenho em Field-Programmable Gate Array (FPGA).