Unidade III

5 ORGANIZAÇÃO DAS MEMÓRIAS

5.1 Características básicas da memória

Uma memória é definida como um componente pertencente a um sistema eletrônico e que tem como função o armazenamento de dados/instruções manipuladas por um sistema computacional (MONTEIRO, 2019). A figura a seguir mostra, em um esquema conceitual, uma memória fictícia simbolizada por um depósito para ser utilizada por uma ou mais entidades, como ocorre com um armário que possui várias gavetas contendo pastas em seu interior. Por causa do tipo de organização computacional atual, as memórias devem ser interligadas e integradas.

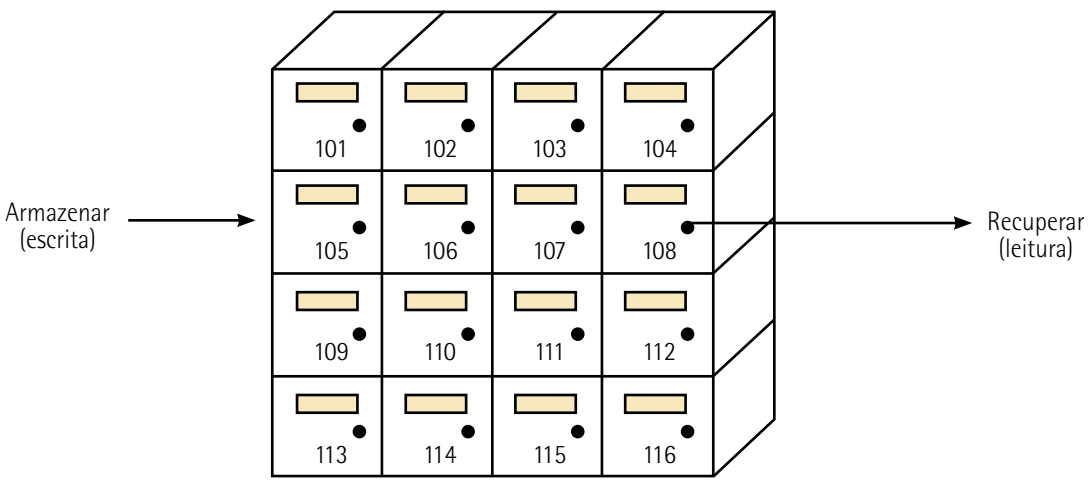


Figura 110 – Armário com gavetas para armazenar pastas

Lembrete

Devido à otimização na organização dos sistemas computacionais atuais, não é possível a utilização de somente um tipo de memória, mas sim de um conjunto de memórias interligadas e integradas.

São consideradas memórias primárias de um computador os seguintes dispositivos: memória *cache*, memória ROM (*read only memory* – memória somente de leitura) e memória RAM (*random access memory* – memória de acesso aleatório). Um dispositivo de memória deve ser capaz de armazenar no mínimo um dos dois estados fundamentais da lógica binária (0 ou 1). O modo pelo qual 1 bit será

identificado na memória pode variar como por meio de um campo magnético, presença ou ausência de uma marca óptica ou mais comumente por um sinal elétrico, esquematizado na figura:

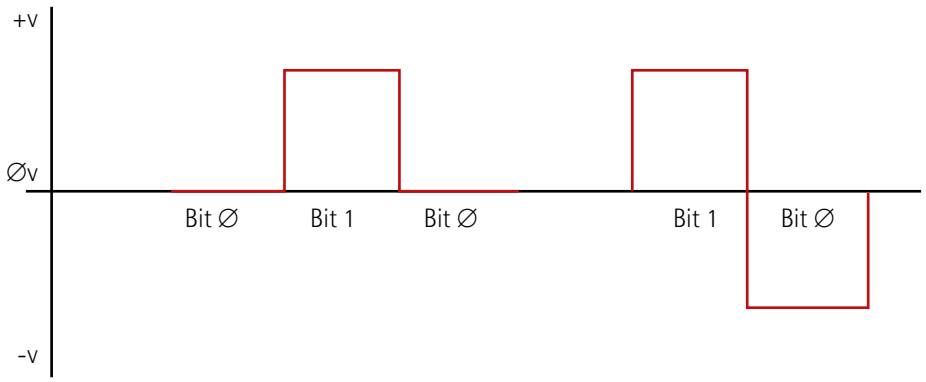


Figura 111 – Representação de *bits* em sinais elétricos

Como já é de conhecimento, 1 bit pode indicar dois valores distintos que podem ser utilizados para o armazenamento dos dados de forma simbólica, como, por exemplo, o alfabeto. O alfabeto em língua portuguesa contém 26 letras minúsculas (incluindo as letras k, w, x, y), 26 letras maiúsculas, 4 símbolos matemáticos (+, -, *, /) e 8 sinais de pontuação. Apenas nesse exemplo simples foi possível criar 64 possibilidades de representação de informações que necessitariam ser distinguidas internamente por um computador, utilizando apenas 2 bits para essa tarefa. Uma das possíveis soluções para esse problema de codificação é definir um código com 64 elementos, com cada elemento contendo 6 bits.

Além da codificação, os sistemas de memória também podem ser compreendidos de acordo com algumas classificações, como (STALLINGS, 2010):

- **Localização**: serve para indicar se a memória utilizada é interna ou externa ao computador.
- **Capacidade**: a memória deve ser fabricada para suportar a leitura/escrita em *bytes* (8 bits) de acordo com o tamanho da palavra em sistemas que compreendem 8, 16, 32 e 64 bits.
- **Unidade de transferência**: é constituída pelo número de linhas elétricas (barramento) para dentro ou fora do módulo de memória em operação, que podem ser lidas ou escritas de uma só vez. Em se tratando de memórias externas, os dados transferidos seguem ao seu destino em unidades maiores, como os blocos, que possuem centenas de palavras.
- **Palavra**: é a unidade natural de organização da memória, que possui um tamanho igual ao número de *bits* utilizados para representar o tamanho de uma instrução. A arquitetura x86, por exemplo, possui grande variedade de tamanhos de instruções, expressos como múltiplos de *bytes*.
- **Unidades endereçáveis**: a unidade endereçável é constituída pela palavra, que possui uma relação entre o tamanho em *bits* A de um endereço e o número N de unidades endereçáveis como $2^A = N$.

5.1.1 Acesso à memória

Outros parâmetros associados às memórias dizem respeito a como é realizado o acesso para leitura ou escrita dos dados e instruções, que podem ser subdivididos como:

- **Acesso sequencial**: as memórias são organizadas em unidades chamadas registros. O acesso em algumas memórias é realizado em uma sequência linear específica, assim a informação de endereçamento é utilizada no auxílio de separação de registros e no processo de recuperação de dados armazenados. Unidades de memória de fita magnética são o exemplo mais comum para o acesso sequencial.

Figura 112 – Rolos de fita magnética

- **Acesso direto**: de forma semelhante ao acesso sequencial, esse geralmente envolve um esquema compartilhado para realizar a leitura/escrita. Nesse tipo de acesso, os registros individuais ou blocos de registros possuem um endereço exclusivo em algum local físico da memória. Para que seja possível o acesso direto, primeiro é necessário alcançar uma vizinhança geral ao endereço, depois uma busca sequencial, uma contagem ou espera até que o local desejado seja alcançado. O tempo de acesso para esse tipo de sistema pode ser variável, pois depende de alguns fatores, como as rotações por minuto (RPM). Os discos rígidos magnéticos, como o da figura a seguir, são um exemplo de acesso direto à memória.

Figura 113 – Disco rígido magnético

- **Acesso aleatório**: nesse tipo de acesso à memória, cada local endereçável possuirá um sistema de endereçamento exclusivo, que está fisicamente interligado. O tempo necessário para o acesso à memória de forma aleatória é independente da sequência de acessos anteriores, dessa forma qualquer local desejado poderá ser selecionado de maneira aleatória e, também, endereçado e acessado de forma direta. A memória principal do computador (RAM), como mostrado na figura a seguir, é o principal exemplo de acesso aleatório à memória.

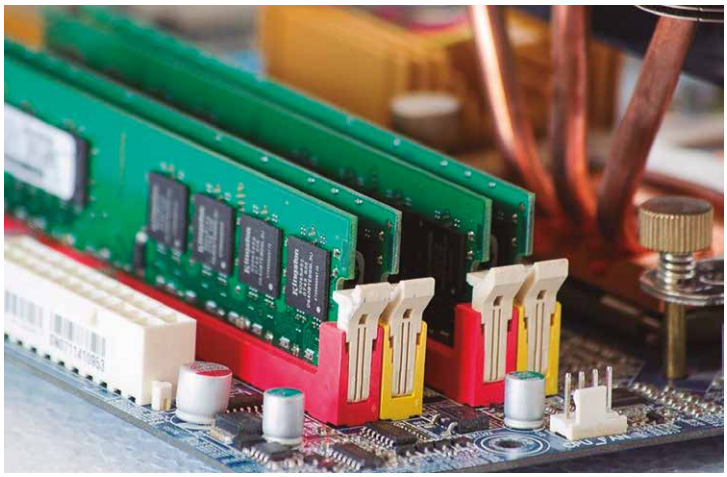


Figura 114 – Memória de acesso aleatório RAM

- **Acesso associativo**: é uma variação do acesso aleatório, porém nesse tipo de endereçamento uma palavra será recuperada da memória baseada em um pedaço de seu conteúdo, em vez do seu endereço. Assim, como no acesso aleatório comum, cada endereço terá seu modo de endereçamento, com um tempo de recuperação constante, independentemente do padrão de acesso anterior. Memórias *cache*, como as da figura a seguir, representam esse tipo de acesso.

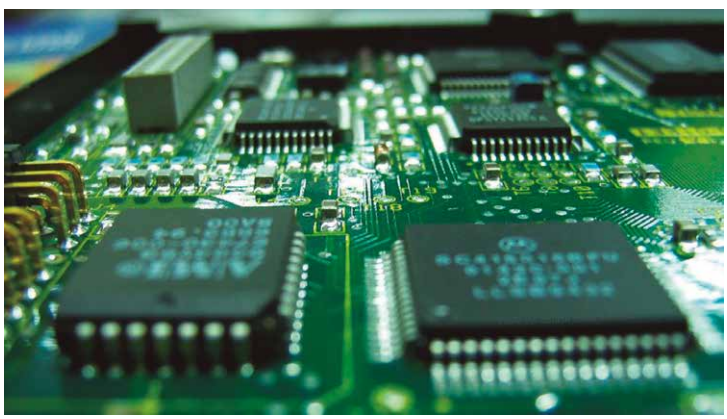


Figura 115 – Memórias *cache*

Para o usuário final de um sistema computacional, outras características também são importantes, como a capacidade de memória e sua velocidade (desempenho). Para que o desempenho seja avaliado, três parâmetros devem ser considerados:

- **Latência ou tempo de acesso à memória**: é o tempo gasto para que se realize uma operação de leitura/escrita. Esse tempo será contado desde que o endereço for apresentado à memória até o momento em que os dados forem armazenados ou se tornarem disponíveis para serem utilizados.

- **Tempo de ciclo de memória**: consiste no tempo para um acesso aleatório mais um tempo adicional antes do início de um segundo acesso.

- **Taxa de transferência**: consiste na taxa onde dados podem ser transferidos externamente ou internamente da unidade de memória. No caso de um acesso aleatório à memória, esse valor será igual a 1 sobre o tempo de ciclo, ou seja, $\frac{1}{\tau}$. Em situações em que o acesso não seja do tipo aleatório, a seguinte relação é utilizada:

$$T_N = T_{A+} \frac{n}{R}$$

onde T_N = tempo médio para ler/escrever **N** bits, T_A = tempo de acesso médio à memória, n é o número de *bits* e R a taxa de transferência em bps (*bits* por segundo).

5.1.2 Endereços de memória

As memórias são organizadas em células ou locais, onde é possível armazenar uma informação em forma de *bit*. Cada célula unitária possui um número que determina seu endereço, onde os programas podem ser referenciados durante a operação de leitura/escrita. Todas as células de memória possuem o mesmo número de *bits*, assim, se uma célula possuir k bits, ela poderá conter qualquer um dos 2^K diferentes tipos de combinações de *bits*. E se a memória possuir n células, seu endereçamento será de 0 a n – 1. A figura a seguir mostra três modos de organização de uma memória de 96 bits, em que as células adjacentes possuem endereços consecutivos. Todos os computadores atuais expressam o endereçamento de memória, como números binários, mesmo aqueles que também trabalham com as bases octal ou hexadecimal. De modo que se um endereço contiver m bits, o número máximo de células que poderão ser endereçáveis é de 2^m.

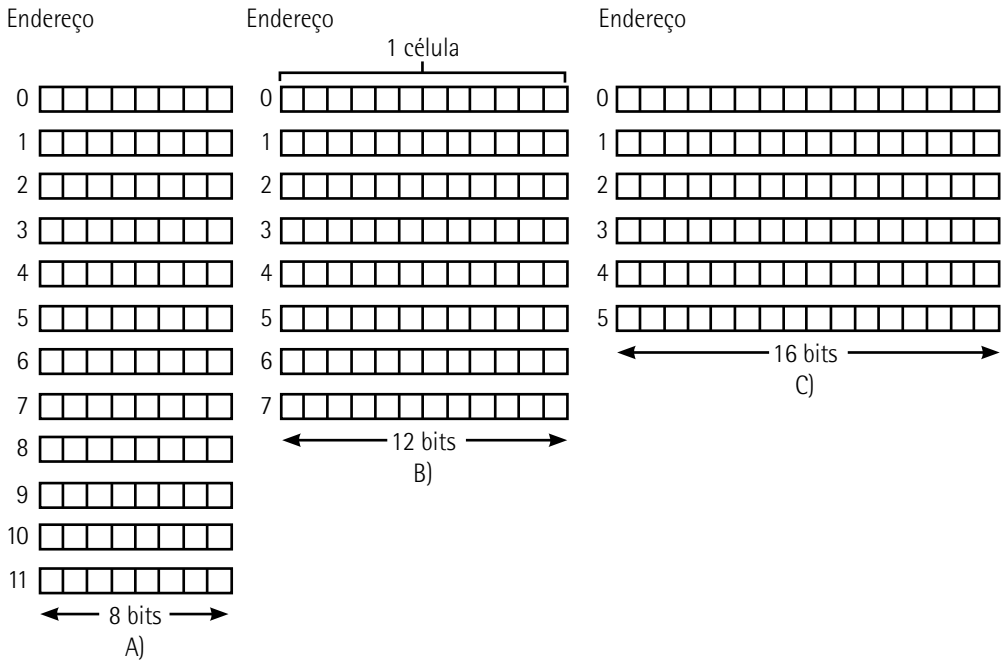


Figura 116 – A) célula de endereçamento de 8 bits; B) célula de endereçamento de 12 bits; e C) célula de endereçamento de 16 bits

5.1.3 Hierarquia de memória

Em um projeto de desenvolvimento de memórias existem três questões que devem ser levadas em consideração na sua fabricação: qual o seu custo, qual a sua capacidade e qual a sua velocidade.

Respondendo a essas perguntas, no que diz respeito à capacidade de armazenamento, as memórias devem possuir a maior possível, deixando livre para o uso de aplicações que necessitem de mais espaço. Em relação à velocidade, certamente que um maior desempenho é desejável para que o processador possa, em tempo síncrono, enviar e receber informações oriundas da memória, ou seja, é desejável que tanto o processador quanto a memória possuam a mesma velocidade de operação. No que tange à relação de custo da memória, ela não deve ser custosa em relação a outros dispositivos do computador. Além disso, a relação de custo de uma memória deve conter outras características, como:

- tempo de acesso rápido envolverá um custo maior por *bit*;
- quanto maior a capacidade de armazenamento, menor será o custo por *bit*;
- quanto maior a capacidade, mais lento será o tempo de acesso à memória.

Quando se projeta um sistema de memória, é necessário sempre atender aos requisitos de desempenho, se atentando para o custo por bit e a capacidade de armazenamento, além de buscar uma melhora no desempenho com tempos menores de acesso. Em um projeto de computador é necessário levar em conta não apenas um dispositivo de memória que fará parte do projeto, mas emprego de

uma hierarquia de memória, em que é possível estabelecer o tempo médio de acesso de vários tipos de memória, como demonstra a figura:

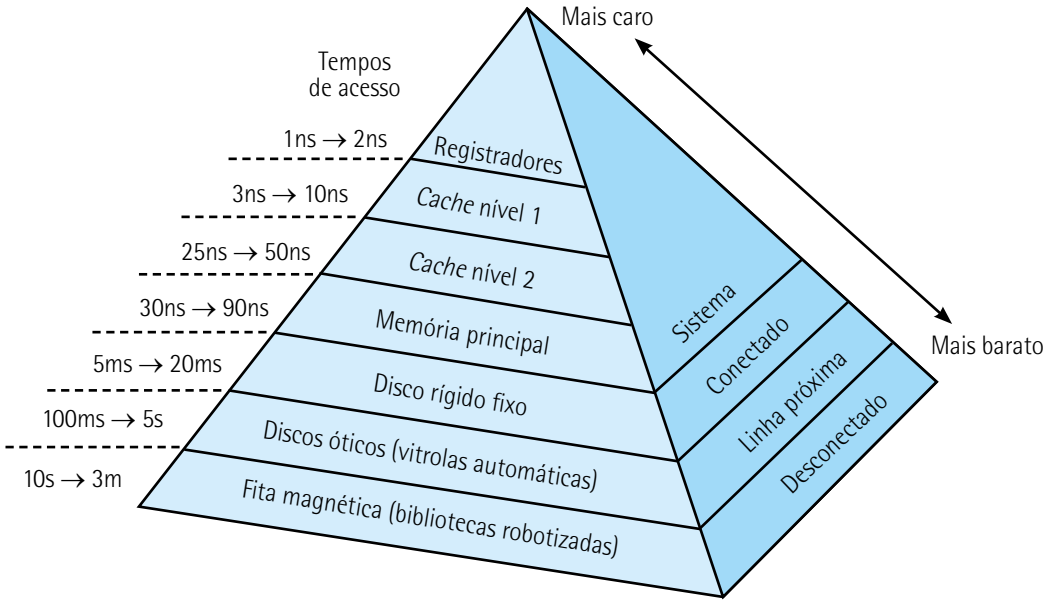


Figura 117 – Pirâmide estrutural da hierarquia de memória

A relação de descendência na hierarquia de memória obedece aos seguintes critérios:

- Diminuição do custo por *bit*, quanto menor for o nível na hierarquia.
- Aumento na capacidade de armazenamento, quanto menor for o nível na hierarquia.
- Aumento no tempo para acesso à memória, quanto menor for o nível na hierarquia.
- Aumento ou diminuição da frequência de acesso às memórias, de acordo com o nível de hierarquia.

Como observado na figura anterior, no topo da hierarquia de memória estão os registradores (MBR, IR, MAR, PC, AC etc.) do processador que, embora mais rápidos (por volta de 1 a 2 nanossegundos), são os mais caros e possuem menor capacidade de armazenamento. Memórias *caches* L1, L2 e L3, internas ao processador, possuem tempo de acesso estimado de 3 a 50 nanossegundos, respectivamente. A memória principal do computador (RAM) possui um tempo de acesso médio de 90 nanossegundos e capacidade de armazenamento que pode chegar a 32 gigabytes em um único pente de memória. Descendo mais no nível de hierarquia, os discos rígidos magnéticos, muito utilizados em computadores pessoais, possuem capacidade de armazenamento mediano (1 a 10 terabytes) a um custo de tempo de acesso na faixa dos milissegundos. Discos ópticos e fitas magnéticas, geralmente utilizadas para realizar *backup* em empresas, estão na base na hierarquia de memória e possuem tempo de acesso que pode levar segundos ou mesmo minutos, com a grande vantagem de serem as memórias com menor custo financeiro atualmente.

5.2 Memória *cache*

A memória *cache*, em um sistema computacional, tem como objetivo obter velocidade de memória similar à das memórias mais rápidas, como os registradores, porém disponibilizando uma maior capacidade (alguns *megabytes*) em comparação aos registradores.

Observação

O computador tem como tendência, durante sua execução, referenciar dados/instruções localizados na memória principal, de forma que essa memória intermediária tenta acessar esses endereços de maneira mais ágil e, ao mesmo tempo, armazenar parte desses dados ou endereços para possíveis acessos futuros.

O conjunto de memórias do computador consiste basicamente em uma memória principal (RAM) de maior capacidade, porém mais lenta, em conjunto com a memória *cache*, de menor capacidade, porém mais rápida, como esquematizado na figura a seguir.

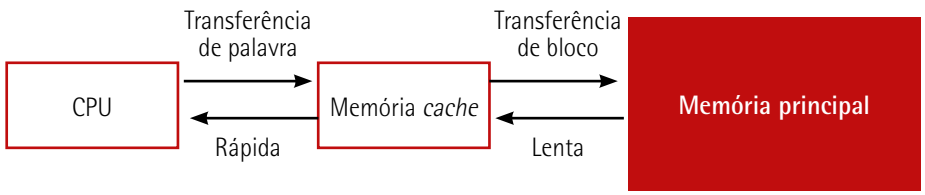


Figura 118 – Relação de comunicação entre a CPU, *cache* e memória principal

Na medida em que a CPU necessita realizar a leitura de uma palavra contida na memória, realiza-se a verificação para determinar se ela está contida na memória *cache*. Se estiver contida, a palavra será entregue à CPU, se não estiver, um bloco da memória principal, contendo algum número fixo de palavras, será lido na *cache* e o resultado será fornecido à CPU. Essas conexões são conhecidas como localidade de referência e ocorrem quando um bloco de dados é enviado para a memória *cache* a fim de realizar uma única referência à memória. As memórias do tipo *cache* ainda podem ser subdivididas em diferentes níveis (multiníveis) ou *layers* (L1, L2 e L3), em que a L3 possui maior capacidade de armazenamento entre todas as *caches*, porém sendo a mais lenta. A L2 é mais lenta que a L1 e, por consequência, a L1 é a mais rápida nessa hierarquia. A figura a seguir mostra uma relação de comunicação entre os diferentes níveis de *cache*, porém, nesse exemplo, todas elas estão externas à CPU, fato que não ocorre com as memórias *cache* atuais.

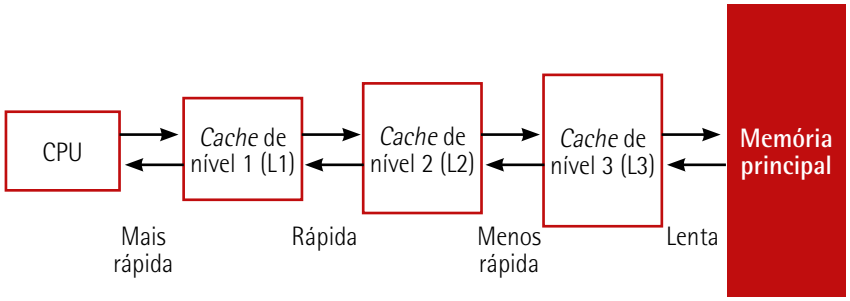


Figura 119 – Organização de memória *cache* em três níveis

De forma mais específica, a memória *cache* possui uma matriz de etiqueta e uma matriz de dados, como mostrado na figura a seguir. A matriz de etiquetas possui os endereços de todos os dados contidos na memória *cache*, e a matriz de dados contém, consequentemente, todos os dados.

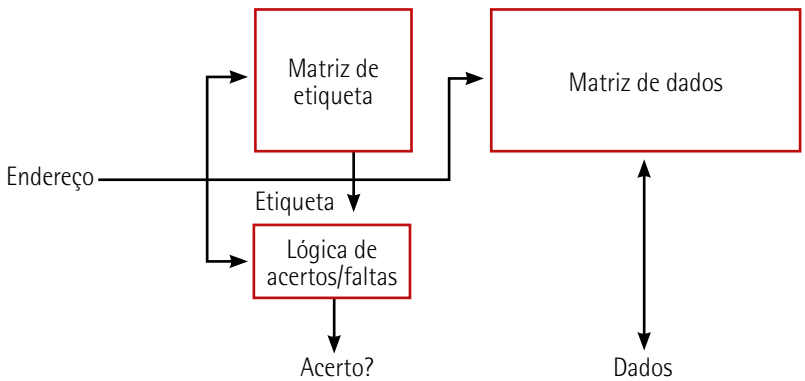


Figura 120 – Diagrama de blocos da memória *cache*

A proposta da divisão da memória *cache* em matrizes tem como objetivo reduzir seu tempo de acesso, pois a matriz de etiquetas geralmente possui menos *bits* se comparado à matriz de dados e, por consequência, poderá ser acessada de forma mais rápida se comparada à matriz de dados. Quando esta é acessada, a saída obtida precisa ser comparada com o endereço da memória de referência para que seja determinado se houve ou não um acerto de memória *cache*, também conhecido como *cache hit*. O acerto de *cache* significa que os *buffers* de dados e endereços são desativados e haverá uma comunicação apenas entre a CPU e a memória *cache*, sem outro tráfego no barramento interno do sistema. O processo inverso ocorre em caso de falha de *cache* ou *cache miss*, em que o endereço desejado é colocado no barramento e os dados são deslocados pelo *buffer* de dados para a memória *cache* e CPU.

5.2.1 *Cache* de dados e instruções

Em outros tipos de memória, como a RAM, as instruções e os dados geralmente compartilham o mesmo espaço dentro de cada nível da hierarquia de memória, entretanto o mesmo não ocorre na memória *cache* em que dados e instruções são armazenados em *caches* diferentes. Essa característica ficou conhecida como *cache* Harvard (figura a seguir), pois se trata de uma das características da arquitetura de Harvard.

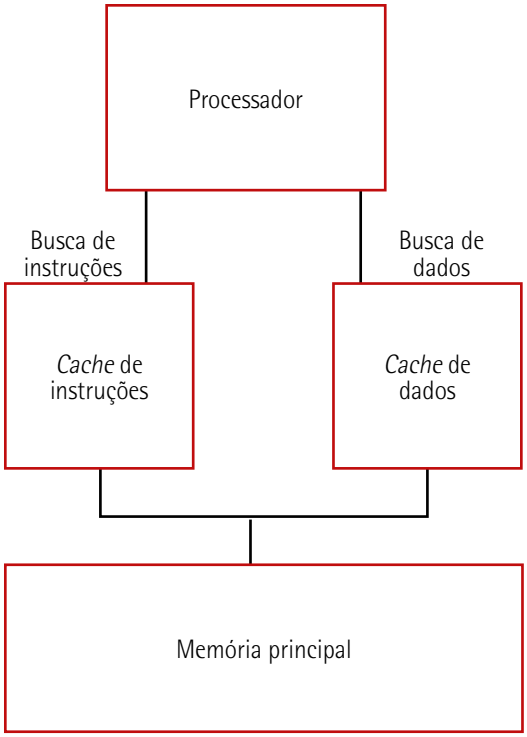


Figura 121 – Esquema de interligação de *cache* para processadores de arquitetura Harvard

Essa configuração é utilizada porque permite que a CPU busque de forma simultânea as instruções a partir da *cache* de instruções e dados pela *cache* de dados. Assim, quando a memória *cache* possui tanto instruções quanto dados, ela é denominada *cache* unificada. Uma desvantagem no uso de *cache* separadas é que a automodificação de certos programas se torna difícil, ou seja, quando um programa precisa modificar suas próprias instruções, tais instruções serão tratadas como dados e armazenados na *cache* de dados. Assim, para executar as instruções modificadas, o sistema operacional precisará realizar operações especiais de descarga. De forma geral, a memória *cache* de instruções pode ser de duas a quatro vezes menor do que a *cache* de dados, pois as instruções de um sistema operacional ocupam menor espaço físico na memória se comparado com os dados.

5.2.2 Endereço de *cache*

Um conceito intimamente relacionado com a memória *cache* diz respeito à memória virtual, nome dado à técnica que utiliza uma memória secundária como o disco rígido, ou mesmo a memória *cache* para o armazenamento temporário. Uma memória virtual, basicamente, permite que programas façam o endereçamento da memória de forma lógica, sem de fato considerar a quantidade de memória disponível fisicamente na RAM. Quando a memória virtual é utilizada, os locais de endereço nas instruções conterão endereços virtuais e não físicos, de forma que para realizar a leitura/escrita da memória RAM será necessário o uso de uma unidade gerenciadora de memória (MMU – *memory management unit*) para traduzir os endereços virtuais em endereços físicos na RAM, como mostra a figura:

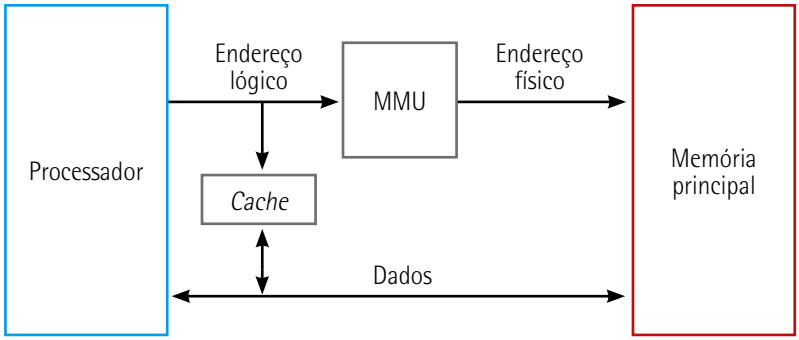


Figura 122 – Organização de memória *cache* lógica

Uma *cache* lógica ou virtual pode armazenar os dados utilizando endereços também virtuais, assim o processador irá acessar a memória *cache* de forma direta, sem a necessidade de ter que passar pela MMU, como mostra a organização feita na figura a seguir.

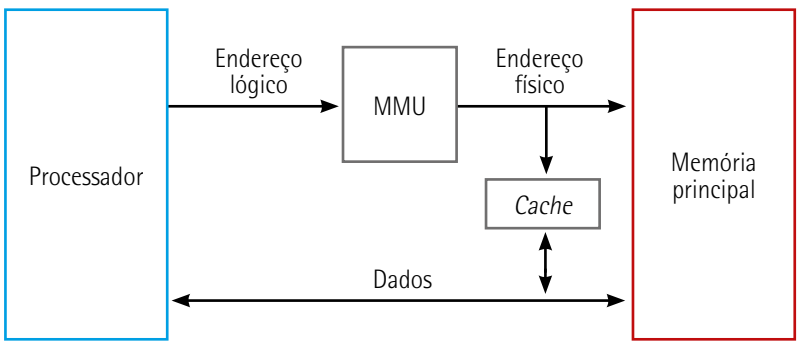


Figura 123 – Organização de memória *cache* física

Uma das vantagens nesse tipo de abordagem da *cache* lógica é que a velocidade de acesso será maior do que para uma *cache* do tipo física, pois, como os diagramas mostram, a *cache* pode responder antes que a MMU concretize a tradução do endereçamento.

5.2.3 *Cache*s associativas

A associatividade em memória *cache* significa quantas posições nela contêm um endereço de memória. As *caches* associativas permitem que os endereços sejam armazenados em diversas posições na *cache*, o que reduzirá as penalidades ocasionadas por conflitos no barramento de memória, devido a dados que precisem ser armazenados nas mesmas posições. Memórias *cache* que possuam uma baixa associatividade podem restringir o número de posições de endereços disponíveis, o que pode ocasionar o aumento nas falhas, e que, por consequência, reduzirá o espaço ocupado. As *cache*s podem ter um mapeamento associativo, como mostrado na figura:

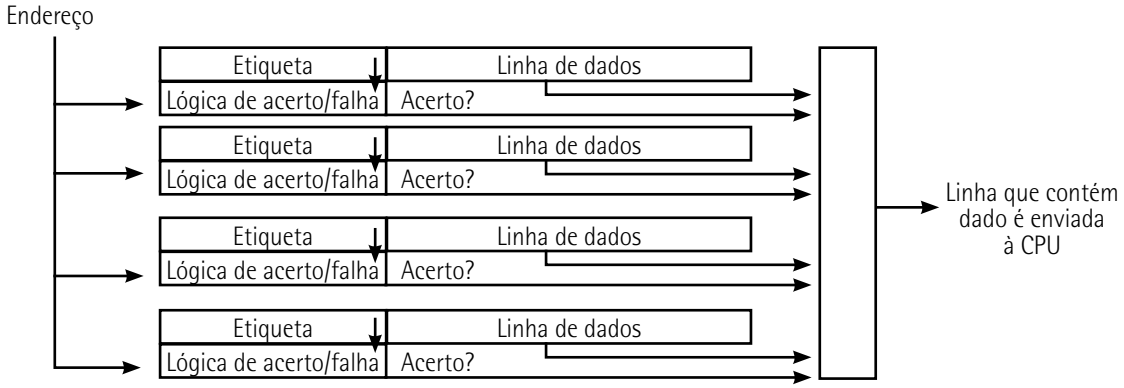


Figura 124 – Esquema de memória *cache* com mapeamento associativo

Nesse tipo de endereçamento de *cache* é permitido que qualquer endereço seja armazenado em qualquer uma das linhas de *cache*. Assim, quando uma operação de acesso à memória é solicitada à *cache*, tal acesso precisa ser comparado a cada uma das entradas na matriz de etiquetas, para que seja determinado se os dados referenciados na operação de fato estarão contidos nela. As *caches* associativas são, no geral, implementadas com matrizes distintas para dados e etiquetas.

5.2.4 *Cache*s com mapeamento direto

As *caches* que possuem mapeamento direto são exatamente o oposto das associativas. No mapeamento direto, cada endereço só poderá ser armazenado em uma posição da memória *cache*, como exemplificado na figura a seguir. Dessa forma, quando uma operação de acesso à memória é enviada a uma *cache* que foi mapeada de forma direta, um subconjunto de *bits* será utilizado para selecionar o *byte* que está dentro de uma linha da *cache* para onde aponta o endereço.

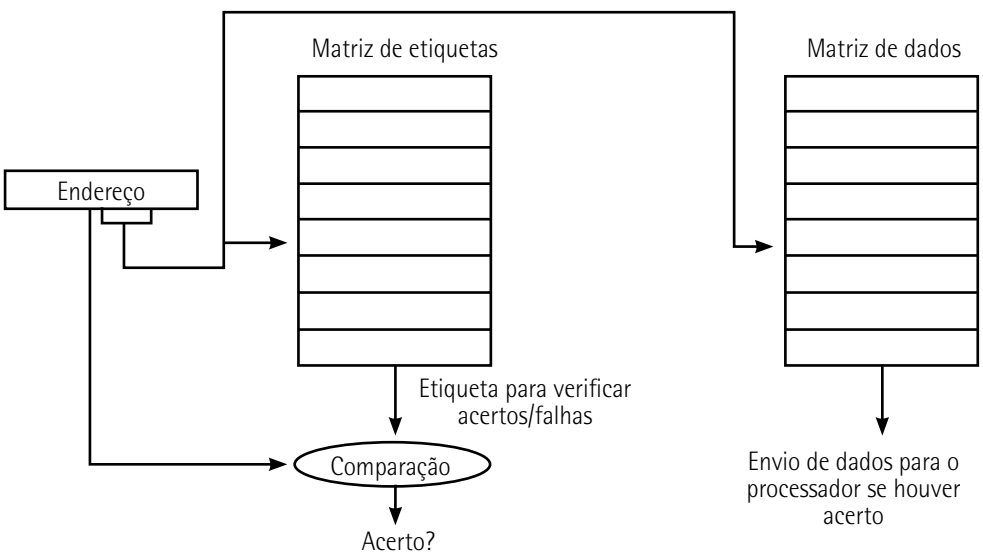


Figura 125 – Esquema de memória *cache* com mapeamento direto

De forma geral, haverá n bits menos significativos no endereçamento que serão utilizados para determinar a posição do endereço de dentro da linha de *cache*, onde n é dado por log na base 2 do número de bytes contidos na linha. Assim, os *m bits* mais significativos dados também por log na base 2 do número de linhas na *cache* serão utilizados para selecionar a linha que conterá o endereço armazenado.

Saiba mais

Aprenda mais sobre a importância das memórias do tipo *cache* nos computadores atuais em:

BEGGIORA, H. O que é memória *cache*? Entenda sua importância para o PC. *TechTudo*, 21 out. 2016. Disponível em: https://glo.bo/3qC9SAn. Acesso em: 25 fev. 2021.

5.3 Memória somente de leitura

As memórias somente de leitura (ROM – *read only memory*), como as da figura a seguir, são fabricadas a partir de materiais semicondutores (silício), não voláteis, ou seja, não perdem seus dados/instruções ao serem desligadas, possuem desempenho semelhante ao das memórias de leitura/escrita, além de serem seguras por permitir apenas a leitura de seu conteúdo através de determinados programas.

Observação

Todos os dispositivos computacionais modernos como computadores, celulares, *tablets*, consoles de jogos, câmeras digitais etc. utilizam uma parte do endereçamento da memória RAM baseada em memórias ROM.

Figura 126 – Memória somente de leitura ROM

As memórias do tipo ROM também são conhecidas por fazerem parte do sistema de inicialização dos computadores pessoais, e são popularmente conhecidas como BIOS (*basic input output system* – sistema básico de entrada e saída). Além de que, devido a sua versatilidade e baixo custo, a memória ROM também é responsável por armazenar sistemas de controle de processos em dispositivos embarcados como injeção eletrônica em automóveis, fornos micro-ondas, geladeiras, smart TVs, *drones*, entre outros. As memórias do tipo ROM vêm evoluindo ao passar das décadas, a fim de torná-las mais rápidas e práticas devido a sua grande aderência em dispositivos embarcados, de modo que vários tipos de diferentes configurações de ROM foram criados.

5.3.1 ROM programada por máscara

O tipo mais básico de memória ROM é a programada por máscara, nome dado devido ao processo de fabricação e escrita de *bits* nesse modelo. Os *bits* que fazem parte do programa de usuário são gravados no interior dos elementos da memória durante o processo de fabricação, conhecido como *hardwired*, onde cada um já é gravado no endereço de célula correspondente (MONTEIRO, 2019). Após sua fabricação, a pastilha ROM estará pronta para ser utilizada e nenhum tipo de *software* será capaz de modificar os *bits* nela gravados. A memória do tipo ROM também é mais barata de ser fabricada pois utiliza-se apenas uma máscara de um programa matriz para que milhares de cópias do *hardware* sejam produzidas. Porém também há desvantagens no processo de fabricação em larga escala dessas memórias como:

- Uma vez produzida, não há possibilidade de recuperar qualquer erro do programa, ou seja, se apenas 1 único bit estiver errado na pastilha, todo o lote deverá ser descartado.

- O custo de fabricação de uma máscara para escrita dos *bits* na memória é o mesmo, de modo que é possível fabricar milhares de módulos de memória.

Devido a essas características, a memória ROM é considerada um dispositivo mais simples, se comparada à memória RAM, pois necessita de apenas um decodificador e um barramento de saída, além de circuitos lógicos utilizando portas OR para realizar o endereçamento, como mostra a figura a seguir.

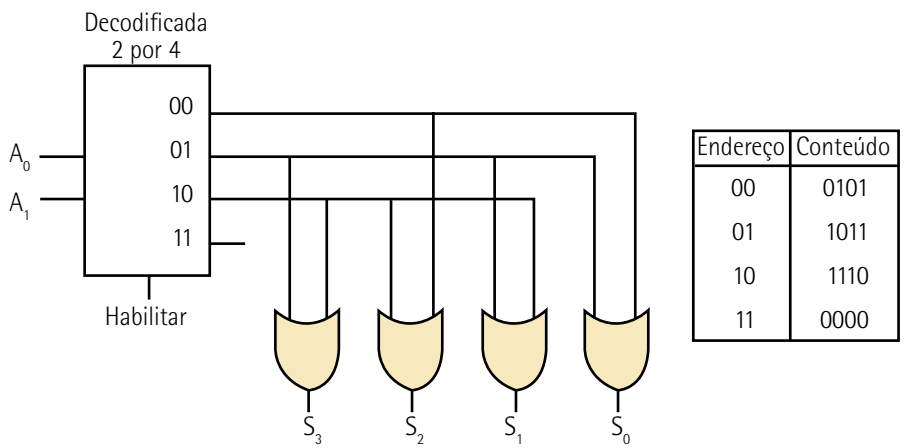


Figura 127 – Esquema de memória ROM com mapeamento direto

Nesse exemplo, é possível observar que a memória é decodificada no modo 2 por 4 e possui quatro células de 4 bits cada, podendo endereçar quatro endereços de 2 bits. Nota-se também que o decodificador de endereços possui uma entrada para um endereço de dois bits (A_0 e A_1) e linhas de saída do decodificador que se conectam em quatro portas OR (S_0, S_1, S_2 e S_3), em que cada porta será responsável pela geração de um dos 4 bits da célula de memória, de acordo com o endereço especificado.

5.3.2 PROM

As memórias ROM programáveis (*programable read only memory*) são fabricadas sem nenhuma informação gravada e, posteriormente, seja pelo fabricante ou pelo usuário, são gravados os *bits* contendo as informações que ficarão fixas nela. Isso quer dizer que, embora fabricadas sem nenhum dado (limpas), após serem gravados os dados não poderão mais ser apagados. Um outro diferencial entre a memória ROM e a PROM está no seu custo individual. Como a ROM pode ser fabricada em larga escala a custos fixos da matriz, dividido pela quantidade de cópias, no caso da PROM sua fabricação será facilitada devido ao menor custo individual de fabricação, que independe da quantidade, pois não haverá o custo da fabricação da máscara.

5.3.3 EPROM

Com a evolução dos computadores, também vieram as evoluções nas memórias ROMs. Uma variação da PROM foi a EPROM (*erasable* PROM) ou PROM apagável (figura a seguir), que também se trata de uma memória não volátil.

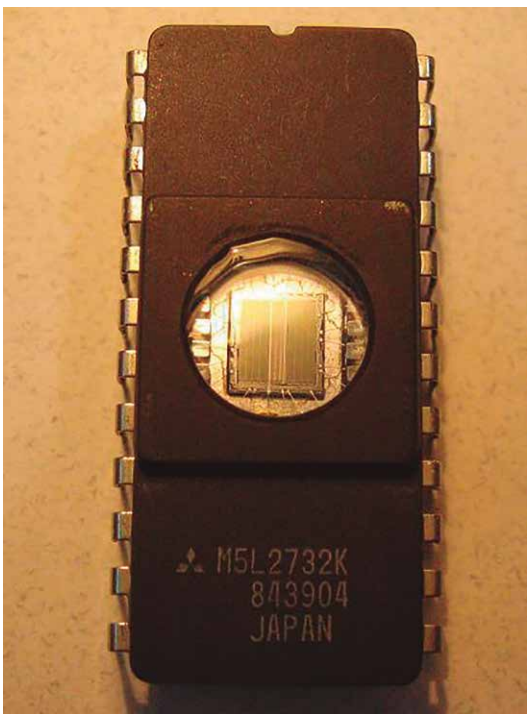


Figura 128 – Memória EPROM

Uma EPROM é programada por um dispositivo eletrônico que transforma algum programa computacional em corrente elétrica, injetada na memória. Uma janela de vidro (quartzo) situada acima da memória já programada é utilizada para apagar completamente todo seu conteúdo. Para que o processo de apagamento seja realizado, uma luz ultravioleta (~200-400 nanômetros) é incidida sobre a janela em um tempo estimado de 20 a 25 minutos. Uma vez apagada, a memória ainda pode ser regravada através do processo de "queima" de bits, que ocorre com a inserção de uma corrente elétrica mais forte do que a convencional utilizada para ler os dados.

5.3.4 EEPROM e *flash*

A memória EEPROM (*electronically* EPROM), ou EPROM eletronicamente gravável e apagável, é uma memória parecida com a RAM, pois permite que múltiplos endereçamentos sejam escritos/apagados em uma única operação. Com um grande diferencial da memória RAM, uma vez escrito algum dado, ele será preservado sem a necessidade de fonte de alimentação constante para mantê-lo. Uma variação da memória EEPROM é a memória *flash*, muito utilizada como cartões de memória, *pen drives*, MP3 *players*, armazenamento interno de câmeras digitais e celulares. Esse tipo de memória pode apagar total ou parcialmente seu conteúdo em uma única operação de escrita.

Figura 129 – Diversas memórias do tipo *flash*

Saiba mais

Sobre as memórias do tipo ROM, acesse:

GARRETT, F. O que é memória ROM? *TechTudo*, 14 out. 2015. Disponível em: https://glo.bo/3qH7YOJ. Acesso em: 26 fev. 2021.

5.4 Memória RAM

Uma célula unitária de uma memória principal é constituída basicamente de um material semicondutor, como o silício. Por definição, um material semicondutor é aquele que apresenta um nível intermediário (não é totalmente isolante e não é totalmente condutor) de portadores de carga ou elétrons livres.

As células unitárias de memória possuem algumas propriedades em comum, como (STALLINGS, 2010):

- Devem apresentar dois estados estáveis, que representam os números binários 0 e 1.
- Devem ser capazes de serem escritas, pelo menos uma vez, para que seu estado seja definido.
- Devem ser capazes de ser lidas, para que seu estado atual seja conhecido.

Lembrete

Por definição, um material semicondutor é aquele que apresenta um nível intermediário de portadores de carga ou elétrons livres.

Uma célula de memória geralmente possui três terminais funcionais, como mostra a figura a seguir, que são capazes de transportar o sinal elétrico que definirá seu estado (0 ou 1). Um dos terminais é o de seleção, que irá selecionar individualmente cada célula de memória, para que uma operação de leitura ou escrita seja realizada.

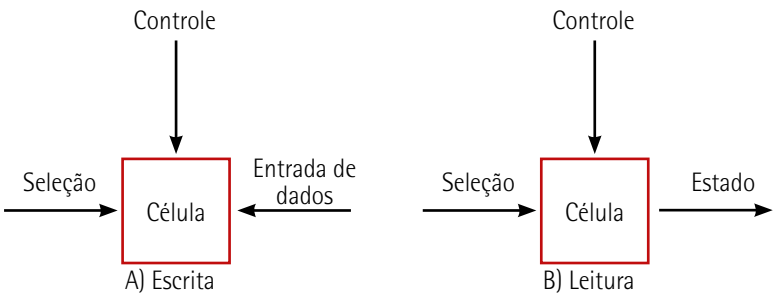


Figura 130 – A) operação de escrita; e B) operação de leitura

Observação

A célula possui um terminal de controle que define qual das duas operações serão realizadas naquele momento. Na operação de escrita, um terminal fornecerá um sinal elétrico que define qual estado (0 ou 1) deverá ser gravado na célula. Por fim, na operação de leitura, o terminal de saída é utilizado para obter o estado atual da célula.

5.4.1 Memória DRAM e SRAM

As memórias do tipo RAM (*random access memory*) acessam as palavras individuais de forma direta através de uma lógica de endereçamento interno. Além disso, em memórias RAM é possível ler os dados contidos na memória e escrever novos dados de forma muito rápida, pois necessitam apenas de sinais elétricos para essas tarefas. Outra característica muito importante nas memórias do tipo RAM é que elas são voláteis, ou seja, para que seu estado seja mantido por um certo tempo, elas devem receber alguma fonte de energia constante. Assim, se a energia for interrompida, os dados gravados serão perdidos, o que determina que a RAM é utilizada somente para armazenamento de dados/instruções de forma temporária.

Existem, de forma básica, dois tipos de tecnologias de memórias RAM: a RAM dinâmica e a RAM estática.

RAM dinâmica ou DRAM (*dynamic* RAM)

Nesse tipo de tecnologia, a célula unitária da memória é constituída por capacitores, que armazenam os dados como cargas elétricas em seu interior. Apesar de armazenarem um estado binário, as memórias RAM são basicamente um dispositivo analógico, pois armazenará uma carga elétrica com um valor dentro de um certo intervalo. Os capacitores são dispositivos eletrônicos que interpretam, nessa situação, a presença (*bit* 1) ou a ausência (*bit* 0) de uma carga elétrica para que seu estado seja definido. Entretanto, os capacitores são dispositivos dinâmicos, ou seja, variam em função do tempo e possuem como característica uma tendência natural para que sua energia seja descarregada. Dessa forma, para que os dados sejam mantidos até a próxima operação de leitura ou escrita, é necessário que um mecanismo de recarga elétrica periódica (*refresh*) ocorra, a fim de que os dados não sejam perdidos. A figura a seguir mostra uma estrutura esquemática de uma memória DRAM típica, possuindo uma célula unitária, capaz de armazenar 1 bit.

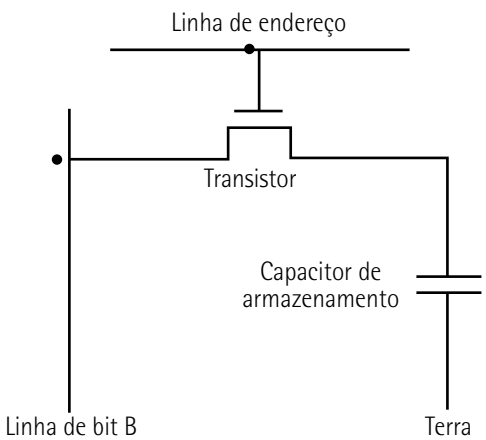


Figura 131 – Célula unitária de uma memória DRAM

Na figura, é possível observar que há uma linha de endereço, que é ativada toda vez que o valor do *bit* precisa ser alterado para o estado de lido ou escrito. Há também outro dispositivo eletrônico na

célula individual chamado de transistor. De forma sucinta, um transistor é um dispositivo semicondutor (silício ou germânio) que atua, de acordo com o tipo, ou como um amplificador de corrente elétrica ou como uma chave para atenuar ou barrar uma corrente elétrica. No caso das células unitárias de RAM, o transistor atuará como uma chave, aceitando ou não uma corrente elétrica. Assim, o funcionamento da célula de RAM se dá quando o transistor aceita a corrente aplicada na linha de endereço e armazena a carga no capacitor. Na operação de escrita, um sinal elétrico é aplicado na linha de endereço com destino ao capacitor, sendo que uma alta tensão (3 a 5 volts) representará um *bit* e uma baixa tensão (0-0,8 volts) representará o *bit* zero. Na operação de leitura, a linha de endereço é utilizada para acionar o transistor da célula unitária, a fim de que o estado atual do capacitor seja lido e sua carga transferida em uma linha de *bit*, para que um amplificador de voltagem possa comparar com um valor de referência, e assim determinar se a célula contém o *bit* zero ou um. É importante salientar que a cada operação de leitura da célula de memória o capacitor será descarregado, necessitando ser restaurado pelo estado anterior ou por um novo estado após a operação.

RAM estática ou SRAM (*static* RAM)

A principal aplicação de uma memória SRAM é como memória *cache* em um processador. Diferentemente da memória DRAM, a RAM estática é um dispositivo mais complexo, como pode ser observado na figura a seguir, e contém os mesmos elementos lógicos encontrados em um processador. O dispositivo básico de armazenamento dos estados nesse tipo de memória é o *flip-flop*, que se trata de um dispositivo estático, ou seja, os *bits* ficarão armazenados enquanto houver energia elétrica sendo fornecida ao dispositivo.

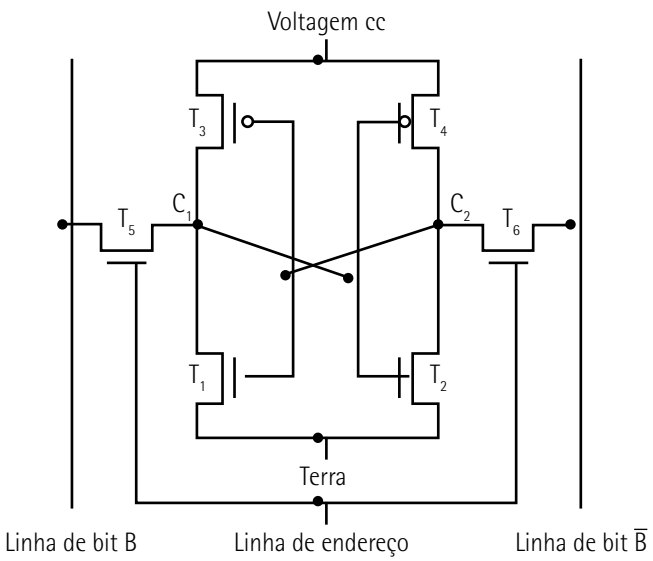


Figura 132 – Célula unitária de uma memória SRAM

No diagrama da figura, observa-se que existem quatro transistores (T_1, T_2, T_3 e T_4) que são cruzados, como se observa nos circuitos sequenciais, também conhecidos como *flip-flops*, de forma a produzir um estado lógico estável. Para o estado lógico, C_1 está em nível lógico alto e C_2 está em nível lógico baixo, de modo que T_1 e T_4 estarão desligados e T_2 e T_3 estarão ligados. Já para o estado

lógico zero, o ponto C_1 está em nível baixo e o ponto C_2 está em nível alto. Nessa situação, T_1 e T_4 estarão ligados e T_2 e T_3 estarão desligados. Os dois estados serão estáveis desde que uma corrente elétrica seja aplicada continuamente, e, diferentemente da memória DRAM, não será necessária nenhuma recarga para reter os dados. A linha de endereço na SRAM também será utilizada para abrir ou fechar uma chave, assim como na DRAM, além de controlar dois transistores (T_5 e T_6). Dessa forma, quando um sinal for aplicado à linha de endereço, esses dois transistores serão ligados, permitindo que uma operação de leitura ou escrita seja realizada. No caso de uma operação de leitura, o estado de *bit* desejado é aplicado à linha B e seu complemento à linha $\overline{B}$, de forma que forçará que os quatro transistores (T_1, T_2, T_3 e T_4) permaneçam no estado correto.

5.4.2 Endereço, conteúdo, armazenamento e posição na memória RAM

Em uma organização de memória composta por vários elementos, identificados e localizados individualmente, é necessário que algum código seja atribuído ao elemento a fim de que ele possa ser identificado e localizado. Esse conceito é o endereçamento ou posição da memória e é válido para qualquer tipo de memória. Na figura a seguir, é possível observar que existem dois endereços (257A e 257B), que alocam dois conteúdos, 1F e 2C, respectivamente. Os *Mbits* armazenados estão contidos em células de memória e são geralmente expressos em grupos de 8 bits ou 1 byte. Dessa forma, é comum expressar a capacidade da memória em termos como 64 Mbytes ou 64 MB.

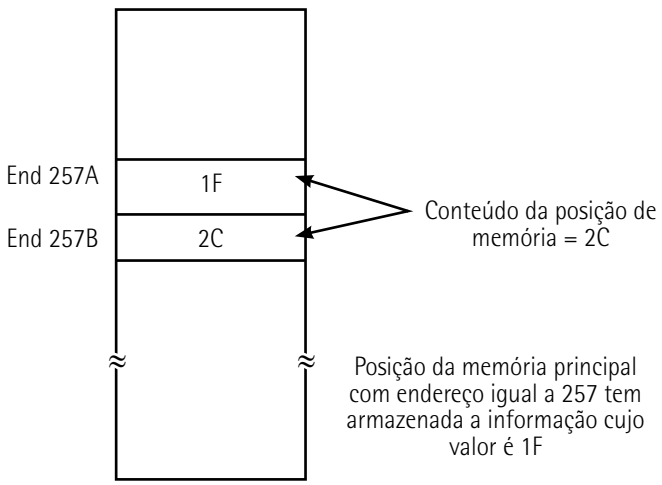


Figura 133 – Valores de endereço e conteúdo da memória principal

Lembrete

A capacidade da memória refere-se à quantidade de informações que podem ser armazenadas em um dado instante de tempo. Para computadores, onde a representação dos dados é através do *bit*, os valores para expressar a capacidade da memória podem ser de 512 bits, 16.384 bits ou 8.388.608 bits.

Porém, como pode ser observado, à medida que esse número de *bits* aumenta, fica muito difícil de se expressar, devido ao tamanho de algarismos necessários para essa tarefa. Entretanto é possível realizar a simplificação dos valores através da seguinte tabela:

Tabela 7 – Grandezas utilizadas em computação

Unidade	Valor em potência de 2	Valor em unidades
1 K (1 quilo)	2^{10}	1.024
1 M (1 mega)	1.024 K = 2^{20}	1.048.576
1 G (1 giga)	1.024 M = 2^{30}	1.073.741.824
1 T (1 tera)	2^{40}	1.099.511.627.776
1 P (1 peta)	2^{50}	1.125.899.906.843.624
1 E (1 exa)	2^{60}	1.152.921.504.607.870.976
1 Z (1 zeta)	2^{70}	1.180.591.620.718.458.879.424
1 Y (1 yota)	2^{80}	1.208.925.819.615.701.892.530.176

Adaptada de: Monteiro (2019).

Assim, os valores 16.384 e 8.388.608 podem ser representados de forma mais simples, como 16 Kbits e 8 Mbits, respectivamente. Embora facilite a interpretação, essa ainda não é a melhor maneira de quantificar a capacidade da memória. Como é de conhecimento, não é possível armazenar dois estados em uma única célula de memória, assim somente um único endereço poderá ser localizado e identificado, pois se dois valores fossem armazenados em um único endereço ou célula, o sistema operacional não seria capaz de identificar qual dos dois estados seria o desejado naquela operação (leitura ou escrita). Dessa forma, mais importante do que a capacidade de uma memória é a quantidade de endereçamento que ela pode manipular, visto que é possível armazenar um dado em cada endereço. Em se tratando de quantizar a capacidade da memória principal, não há padronização. Entretanto, o uso da quantidade de *bytes* em vez da quantidade de palavras é bem-aceito nos computadores modernos e outras denominações, como células, também são aceitas. Alguns exemplos também mostram que utilizar uma simbologia mais simplificada ajuda o leitor a identificar o volume de bytes envolvidos nos cálculos, como pode ser observado:

- 2.048 bytes ≡ 2 K bytes ≡ 2 x 2^{10} bytes
- 393.216 células ≡ 393 K células ≡ 393 x 2^{10} células
- 1.048.576 palavras ≡ 1 M palavras ≡ 1 x 2^{20} palavras

Como já aprendido, a memória RAM é constituída de um conjunto de N células, cada célula armazenando um valor (0 ou 1) designado como *Mbits*. Assim, a quantidade de endereços que ocupam o espaço endereçável da memória RAM também será igual a N, pois cada conteúdo da célula será associado a um endereço. De forma que o valor de N representará a capacidade da memória, que também está associada à quantidade de células ou endereços dessa memória. Como o *bit* é representado por dois estados possíveis, então é possível concluir que (MONTEIRO, 2019):

- É possível armazenar em cada célula unitária um valor entre 0 e 2^{M-1}, de forma que se tem 2^M combinações possíveis. Por exemplo, se M for igual a 8, então tem-se $2^8 = 256$ combinações, resultando em valores binários iniciados em 00000000 e 11111111, que podem ser convertidos em 0_{16} e FF_{16} na base 16 ou 0_{10} e 255_{10} na base decimal, respectivamente.

- A memória principal tem N endereços, e sendo E igual a quantidade de *bits* dos números que irão representar cada um dos N endereços, tem-se: $N = 2^E$. Assim, se N = 512, então $512 = 2^E$, logo E será igual a 9, pois $2^9 = 512$.

- O número total de *bits* que são armazenados na memória RAM é de 512 células, de forma que com 8 bits de tamanho resulta em: N = 512 (total de células), M = 8 bits (tamanho de cada célula), E = 9 bits (tamanho em *bits* do número que representa cada endereço) e T = 4.096 (total de *bits* da memória).

- Utilizando-se a equação T = N x M, tem-se que: 4.096 = 512 x 8. Esses valores podem ser mais facilmente operacionalizados se houver uma representação em potenciação, assim T = 4.096 bits ou 4 Kbits ou 2^{12} bits. É possível utilizar a potenciação para simplificar os demais termos no restante da equação, colocando-os em base 2, onde 512 é representado por 2^9 e 8 é representado por 2^3.

Exemplo de aplicação

Exemplo 1

Uma memória RAM possui um espaço de endereçamento de 2 K, onde cada célula pode armazenar até 16 bits. Qual será o valor total de *bits* que podem ser armazenados nessa memória e qual o tamanho de cada endereço?

Resolução

Se o espaço máximo endereçável é de 2 K, então N = 2 K (quantidade máxima de células também é 2 K). Cada célula tem 16 bits, então M = 16 bits (tamanho da célula) ou 2^4. Sendo N = 2 E, então N = 2 K, e convertendo 2 K em potência na base 2, tem-se: $2^1 \times 2^{10} = 2^{11}$ (mantém-se a base e somam-se os expoentes). Dessa forma, obtém-se que E = 11. Se E representa a quantidade de *bits* de cada número que irá expressar um endereço e E = 11, logo os endereços de cada célula são números que possuem 11 bits. Aplicando-se a equação: $T = N \times M \rightarrow 2^{11} \times 2^4 = 2^{15}$. Convertendo para múltiplos de $K = 2^{10}$, tem-se: $2^5 \times 2^{10} = 32K$.

Logo: T = 32 K (total de *bits* da memória principal) e E = 11 bits (tamanho de cada endereço).

Exemplo 2

Uma RAM foi fabricada com a possibilidade de armazenar 256 Kbits, onde cada célula unitária pode armazenar 8 bits. Qual é o tamanho de cada endereço e qual é total de células unitárias que podem ser armazenadas nessa RAM?

Resolução

Como o total de *bits* é dado por T = 256 K, e utilizando a potenciação em base 2, pode-se expressar 256 K como: $T = 2^8 \times 2^{10} \rightarrow T = 2^{18}$.

Se 1 célula contém 8 bits, então $M = 8 \rightarrow M = 2^3$. Utilizando-se a equação T = M x N, então N (quantidade de células) pode ser reescrito como: N = T/M. Assim, substituindo os valores na equação, tem-se: N = 256 K/8. Colocando tudo em potenciação para facilitar as contas: $N = 2^{18}/2^3 = 2^{15}$ (mantém a base e subtraem-se os expoentes).

Para facilitar a quantificação de N = 215, pode-se representar como N = 25 x 210 ou N = 32 K (células). Sabendo-se que N = 215 e que N = 2E, então o tamanho de cada endereço é E = 15.

5.4.3 Operação de leitura

A operação de leitura ocorre após a execução de instruções menores, também conhecidas como micro-operações, sendo que cada uma dessas instruções consiste em uma etapa do processo de leitura ou escrita da memória. Essas micro-operações geralmente gastam um certo tempo de acesso à memória para serem executadas. Assim, o intervalo de tempo gasto na realização de duas operações consecutivas (leitura-leitura, leitura-escrita ou escrita-leitura) é conhecido como ciclo de memória. Como exemplo, as operações de leitura de um dado armazenado (5C) em um endereço 1324 da memória principal para a CPU. As etapas que descrevem essa operação de leitura, que também pode ser observada no diagrama da figura a seguir, são:

- O registrador MAR (*memory address register*) recebe o dado de outro registrador da CPU.
- O endereço é então colocado no barramento de endereços.
- Insere-se um sinal de leitura no barramento de controle.
- Decodifica-se o endereço e a localização das células.
- O registrador MBR recebe os dados pelo barramento de dados.
- Enviam-se os dados contidos na MBR (*memory buffer register*) para outro registrador.

No passo inicial, a UC (unidade de controle) inicia a operação de leitura realizando a transferência do endereço 1324 de algum registrador específico, como o PC (*program counter*) para o registrador MBR, e insere o sinal de leitura (READ) no barramento de controle para indicar o que a memória principal deve fazer em seguida. A memória principal então decodifica o endereço recebido e transfere seu conteúdo para o registrador MBR pelo barramento de dados. A partir do registrador MBR, a informação desejada será transferida para o elemento do processador que será o destinatário final, que geralmente é algum outro registrador da CPU.

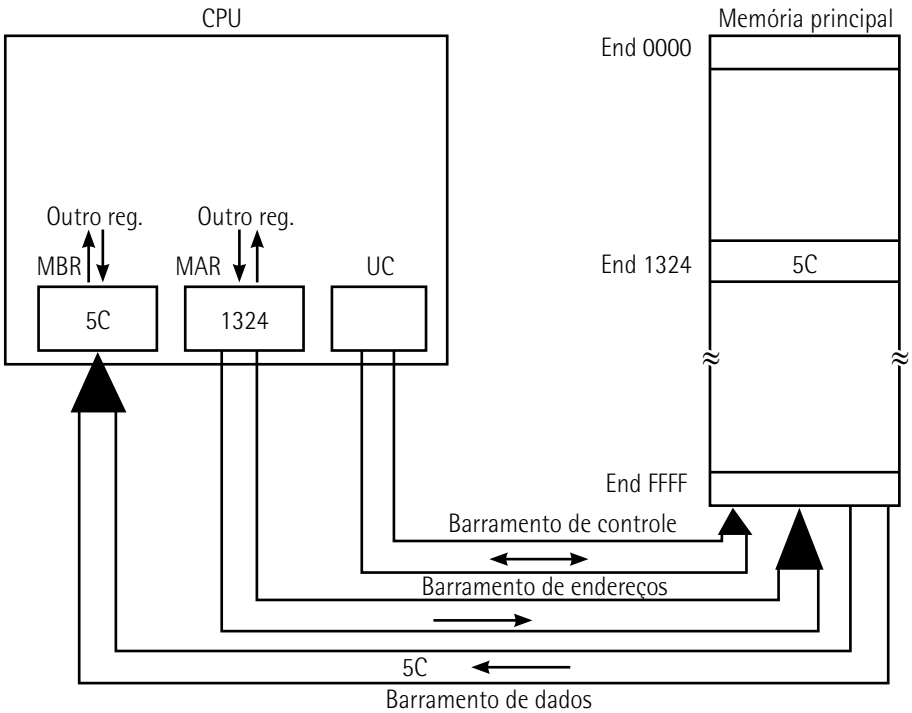


Figura 134 – Operação de leitura na memória principal

O tempo gasto no acesso para a realização dos seis passos descritos anteriormente não garante que a memória principal possa realizar uma nova operação logo em seguida. O que indica estar pronto ou não para uma nova operação está inteiramente relacionado com o modelo de memória RAM utilizada (DRAM ou SRAM). As memórias do tipo SRAM, por exemplo, permitem que outra operação de leitura ou escrita seja realizada de forma imediata logo após a conclusão de uma operação, enquanto memórias do tipo DRAM não possibilitam isso.

5.4.4 Operação de escrita

A operação de escrita possui um procedimento de funcionamento semelhante ao da operação de leitura, com exceção ao sentido da transferência, que é inverso ao da leitura, ou seja, o sentido é do processador para a memória principal. A figura a seguir mostra um exemplo de operação de escrita de um dado (F7) oriunda do processador para ser armazenado na memória principal no endereço 21C8. Os seguintes passos explicam essa operação:

- O registrador MAR recebe da CPU o dado oriundo de outro registrador.
- O endereço correto recebido é colocado no barramento de endereços.
- O registrador MBR recebe os dados da CPU oriundos de outro registrador.
- É realizado o sinal de escrita (WRITE) através do barramento de controle.
- O dado é transferido para a célula de memória referida através do barramento de dados.

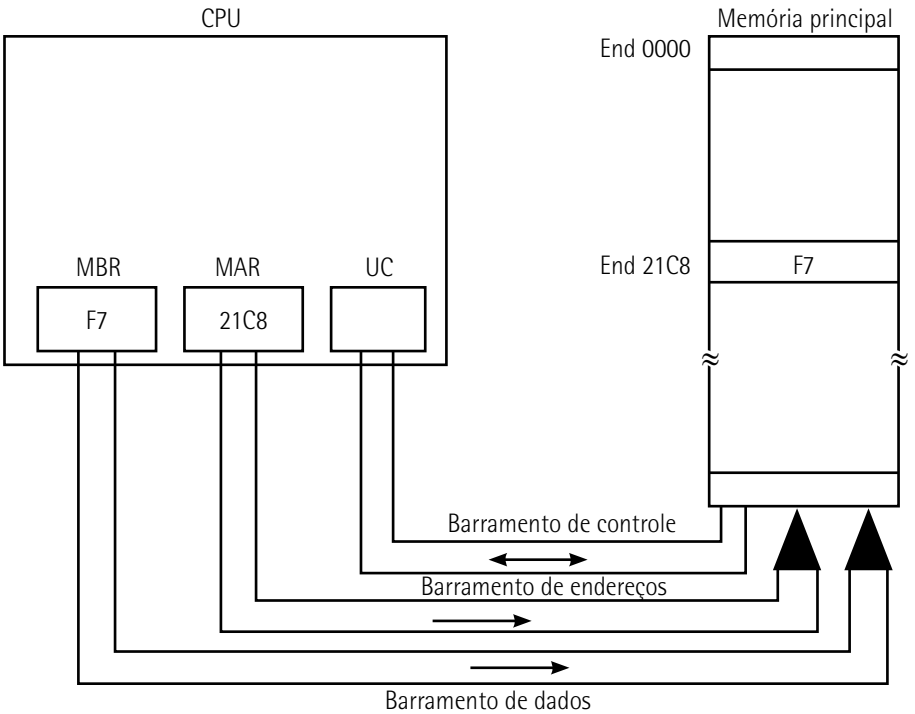


Figura 135 – Operação de escrita na memória principal

Inicialmente, a UC (unidade de controle) coloca o endereço desejado no registrador MAR e o dado que deve ser transferido no registrador MBR. O endereço é então colocado no barramento de endereços e, consequentemente, o dado é colocado no barramento de dados, além do sinal de escrita (WRITE), que é acionado no barramento de controle. O resultado da operação de decodificação do endereço pelos dispositivos de controle da memória será o valor F7, que será alocado na célula unitária com endereço 21C8.

5.4.5 DRAM síncrona

A SDRAM (*synchronous* DRAM) foi desenvolvida em 1994 para troca de dados com o processador de forma sincronizada através de um sinal de *clock* externo, na velocidade do barramento da CPU/memória, sem a necessidade de impor algum estado de espera na comunicação, como ocorria na DRAM tradicional assíncrona. Na DRAM básica, a CPU apresenta endereços e níveis de controle à memória, o que indica que um conjunto de dados que estão em um determinado local na memória deverá ser lido/escrito pela DRAM. Após um tempo de acesso, a DRAM irá escrever ou ler os dados, e durante o atraso de tempo de acesso, a DRAM poderá realizar diversas funções internas, como, por exemplo, verificar e rotear dados através de *buffers* de saída. Entretanto, a CPU deverá esperar por algum atraso, o que diminuirá o seu desempenho. O quadro a seguir mostra algumas das diversas pinagens encontradas nas SDRAM e suas diferentes funcionalidades.

Quadro 8 – Pinagem básica de uma DRAM síncrona

A0 a A13	Entradas de endereço
CLK	Entrada de *clock*
CKE	Habilitação de *clock*
$\overline{CS}$	Seleção de *chip*
$\overline{RAS}$	*Strobe* de endereço de linha
$\overline{CAS}$	*Strobe* de endereço de coluna
$\overline{WE}$	Habilitação de escrita
DQ0 a DQ7	Entrada/saída de dados
DQM	Máscara de dados

Fonte: Stallings (2010, p. 141).

Outra característica da SDRAM é o emprego do modo rajada (*burst*), que elimina o tempo de configuração de endereçamento e tempo de pré-carga de fileira de linha e coluna após um primeiro acesso à memória. No modo rajada, uma quantidade de dados pode ser enviada rapidamente após o primeiro *bit* ter sido acessado. Esse modo de operação é muito útil em situações em que todos os *bits* a serem acessados estiverem em sequência e na mesma linha de *array* do acesso inicial. Além disso, a SDRAM possui uma arquitetura interna (figura a seguir) com um banco múltiplo de memória, que melhora o desempenho devido a operações em paralelo no *chip*.

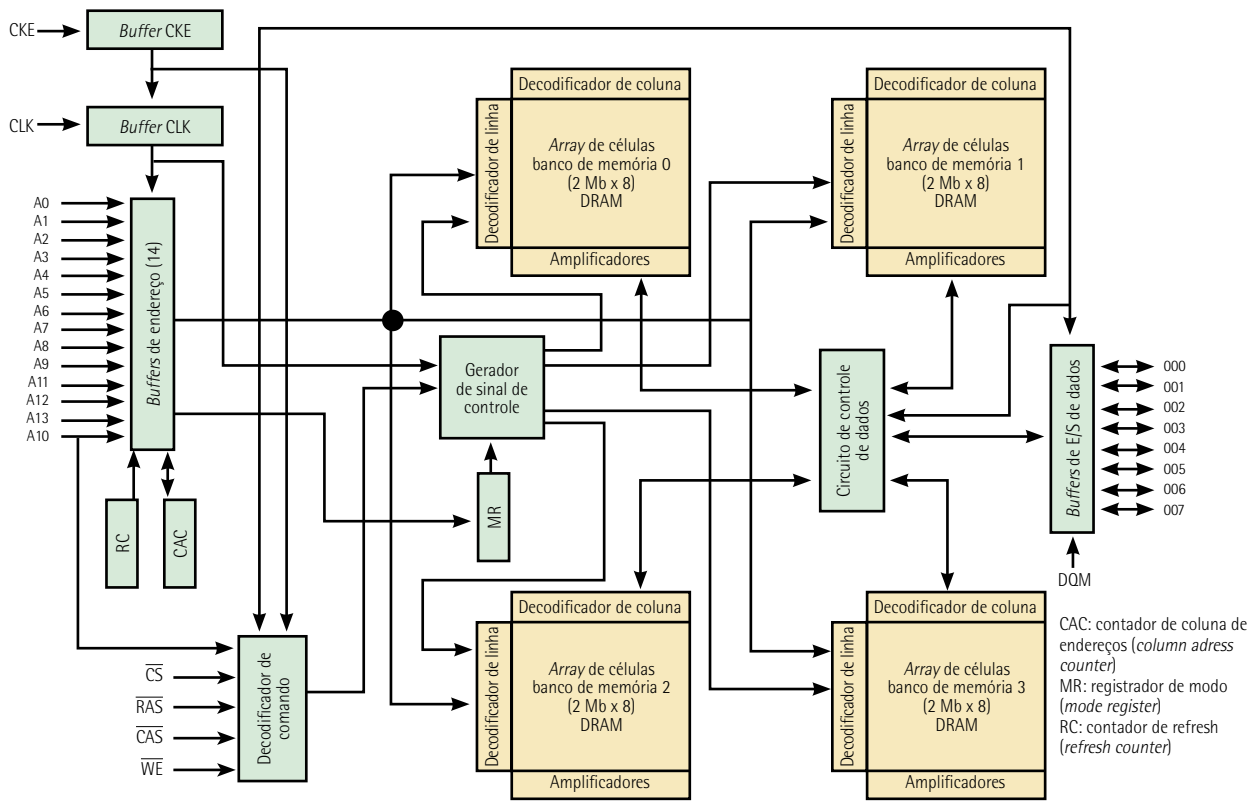


Figura 136 – Diagrama funcional de uma DRAM síncrona

O registrador de modo (MR ou *mode register*) e a lógica de controle também é considerado outro recurso, que diferencia as SDRAM das DRAMs convencionais. Esse mecanismo de lógica especifica o tamanho da rajada, que é baseada no número de unidades de dados alimentadas sincronamente pelo barramento. A SDRAM transfere, de modo eficiente, grandes blocos de dados em série, o que é muito utilizado para aplicações como processamento de textos ou planilhas.

5.4.6 DRAM Rambus

Desenvolvida em meados de 1997, esse tipo de memória se tornou a principal concorrente da memória SDRAM e foi adotada pela Intel em seus processadores Pentium e Itanium. Os *chips* de memória RDRAM são encapsulados verticalmente, com todos os pinos situados em um dos lados, como mostra a figura a seguir. A troca de dados entre memória e CPU ocorre por um barramento com 28 fios de cobre, que possui a capacidade de endereçar até 320 *chips* de RDRAM a uma taxa de 1,6 GBps (gygabits por segundo).

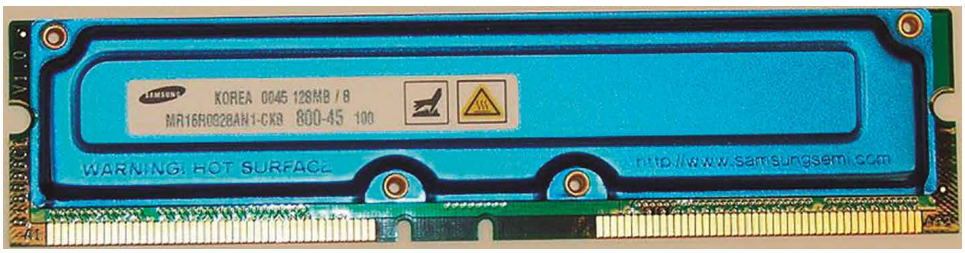


Figura 137 – Memória DRAM Rambus ou RDRAM

O barramento da RDRAM oferece também informações de endereço e controle utilizando um protocolo de comunicação assíncrono, orientado a bloco de palavras. Em vez de ser controlada por sinais RAS, CAS, R/W ou CE, como na memória SDRAM, a RDRAM recebe uma solicitação de memória através do barramento de alta velocidade, que contém o endereço solicitado, o tipo de operação e o número de *bytes* utilizados na operação atual. A configuração de uma RDRAM, como mostrado na figura a seguir, consiste em um controlador e uma série de módulos conectados pelo barramento em comum.

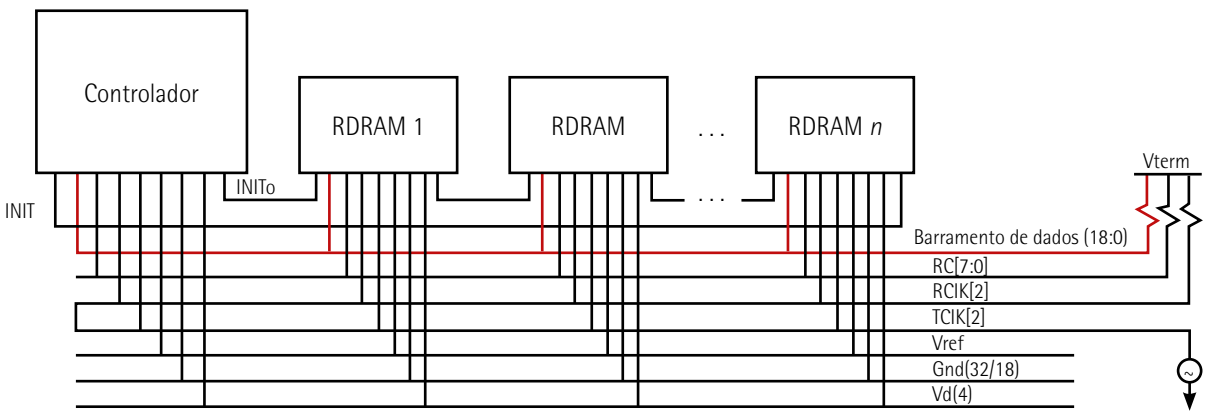


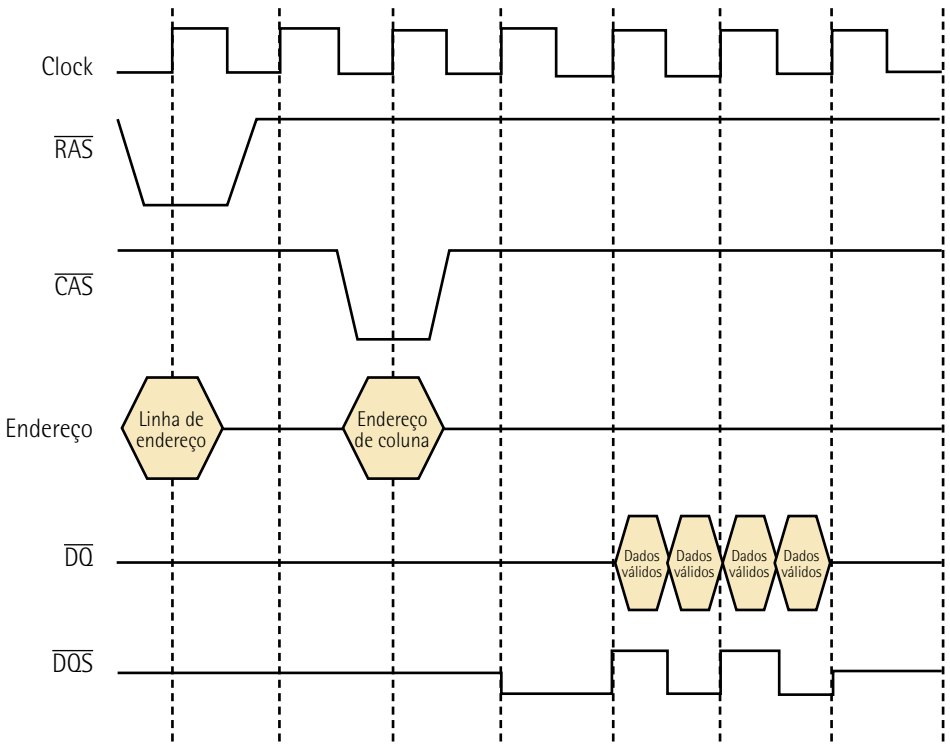
Figura 138 – Estrutura básica da memória RDRAM

O controlador está na extremidade da memória e conectado pelo barramento, que possui 18 linhas para dados, sendo 16 reais e 2 de paridade, pulsando a uma taxa dobrada do sinal do *clock*, ou seja, 1 bit

é enviado em cada transição de *clock*. Isso resultará em uma taxa de 800 Mbps em cada linha de sinal de dados. Na outra extremidade do barramento está localizada a entrada do sinal de *clock*, que se propaga até o extremo do controlador e depois retorna ao seu ponto original. As linhas adicionais do barramento incluem também uma tensão elétrica de referência, fio terra e fonte de alimentação.

5.4.7 DDR-SDRAM

A DRAM síncrona desenvolvida em 1994 era capaz apenas de enviar dados ao processador a cada ciclo de *clock*, o que tornava o processo de escrita e leitura um tanto quanto lento e limitado, pois cada um deles só podia ser realizado a cada ciclo. Uma nova versão de SDRAM, a DDR-SDRAM (*double data rate* SDRAM), desenvolvida pela JEDEC Solid State Technology Association, possui a capacidade de ler/escrever dados duas vezes por ciclo de *clock*. As operações de leitura ou escrita ocorrem, uma de cada vez, a cada transição de subida ou descida do pulso do sinal de *clock*. A figura a seguir mostra a temporização básica para o processo de leitura na DDR.



RAS: seleção de endereço de linha
CAS: seleção de endereço de coluna
DQ = dados (entrada e saída)
DQS: seleção DQ

Figura 139 – Temporização de leitura em uma DDR-SDRAM

A transferência de dados ocorre de forma síncrona, com a transição de subida e descida. Um sinal denominado *strobe* de dados bidirecional (DQS), fornecido pelo controlador da memória, ajuda a sincronizar a memória durante uma leitura ou escrita. Em algumas implementações de DRAM, o DQS pode ser ignorado durante o processo de leitura. Para facilitar a realização do processo de leitura e escrita nas células de memória, elas são organizadas como matrizes, construídas por linhas e colunas.

Dessa forma, para que um endereço na memória seja conhecido através de uma posição, o controlador de memória obtém o seu valor (RAS – *row address strobe*) de uma coluna e o seu valor de linha (CAS – *column address strobe*), como pode ser observado no diagrama a seguir.

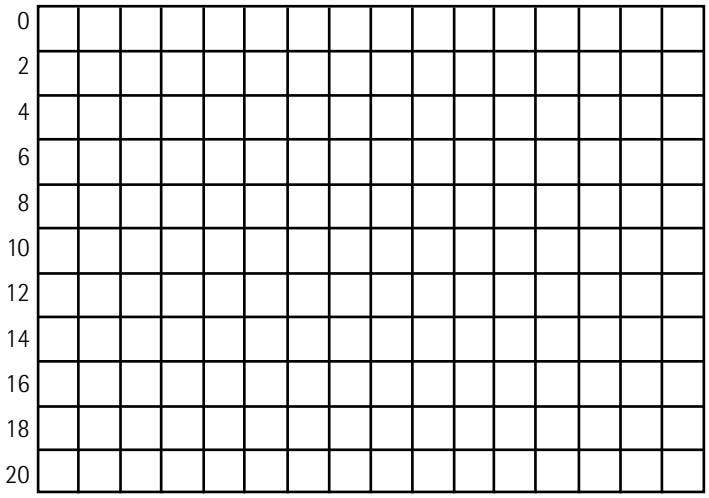


Figura 140 – Diagrama de linhas e colunas de um endereço de memória RAM

Existem algumas variações na tecnologia DDR. A DDR2, por exemplo, aumenta a taxa de transferência de dados, o que leva a um aumento da frequência de operação desse *chip* de memória, além de aumentar o *buffer* (*cache* localizada na RAM) que realiza a pré-busca, aumentando de 2 bits na versão anterior para 4 bits. Esse *buffer* permite que o *chip* RAM pré-posicione os *bits* que serão colocados na base de dados de forma rápida. Já a DDR3, desenvolvida em 2007, possui a capacidade de realizar a pré-busca utilizando um *buffer* de 8 bits. Um módulo DDR simples possui a capacidade de transferir dados a uma taxa de 200 MHz a 600 MHz. Já um módulo DDR2 tem a capacidade de transferir a uma taxa de 400 a 1.066 MHz, e um módulo de memória DDR3 consegue transferir a uma taxa de *clock* de 800 MHz a 1.600 MHz.

Saiba mais

Para um conhecimento mais aprofundado sobre qual tipo de memória RAM escolher para seu computador, recomenda-se a seguinte leitura:

SODERSTROM, T. How to choose the right memory: a 2020 guide to DRAM. *Tom's Hardware*, 5 jun. 2020. Disponível em: https://bit.ly/3bE9fSK. Acesso em: 1º mar. 2021.

Lembrete

Os sistemas básicos de memória são constituídos de vários dispositivos para realizar o armazenamento de dados e instruções, em que cada dispositivo possui sua própria característica em relação ao tempo de transferência, capacidade e custo.

6 MEMÓRIAS SECUNDÁRIAS

É difícil imaginar, mas os primeiros computadores não possuíam nenhum dispositivo de memória secundária como os discos rígidos, por exemplo. No período inicial da computação (1930-1940), os programas eram introduzidos manualmente todas as vezes que precisassem ser executados, mesmo que essa execução fosse repetida, ainda sim era necessário inserir manualmente o programa em cada nova utilização. Os primeiros dispositivos de armazenamento em massa foram as fitas magnéticas, semelhantes às utilizadas nos gravadores de som da época. Com o surgimento dos computadores pessoais (PCs) em 1981, surgiram também os primeiros dispositivos denominados discos rígidos magnéticos, desenvolvidos pela IBM. Antes disso, os discos magnéticos eram utilizados somente nas grandes corporações e eram muito grandes (quase o tamanho de duas geladeiras residenciais), se comparados com os discos rígidos atuais, como pode ser observado a seguir:

Figura 141 – Primeiro disco rígido

6.1 Disco rígido

O RAMAC (*random access method of accounting and control* ou método de acesso aleatório de contagem e controle) da IBM foi o primeiro disco rígido comercializado em 1956 (MONTEIRO, 2019).

Lembrete

Vale ressaltar que os dispositivos anteriores ao RAMAC não eram propriamente discos, e sim tambores cobertos com material magnético para o armazenamento dos dados no formato de minúsculos campos magnéticos.

Enquanto isso, a IBM trabalhava em um novo dispositivo, que era baseado em discos que giravam a taxas elevadas (1.000 a 1.200 rpm ou rotações por minuto), mas que tendia a danificar a cabeça de gravação. No início dos anos 1960, a IBM conseguiu lançar o aperfeiçoamento do RAMAC, facilitando assim o crescimento dessa nova tecnologia.

6.1.1 Organização e funcionamento dos discos rígidos

Um disco magnético é constituído de uma ou mais superfícies circulares e metálicas, denominadas pratos, onde cada superfície é coberta por um material magnetizável. A figura a seguir mostra um disco rígido magnético utilizado em computadores atuais. Note que ele está aberto somente para demonstração de seu interior, visto que, ao se abrir um disco rígido magnético, ele deixa de ter seu funcionamento correto.

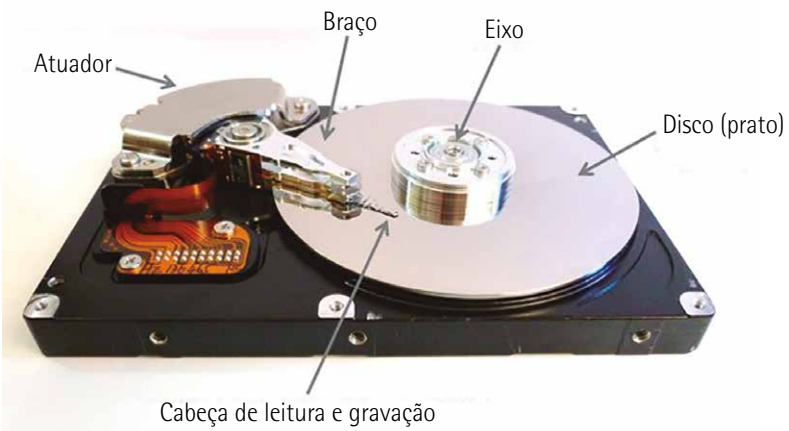


Figura 142 – Disco rígido magnético aberto e seus componentes

Cada superfície ou face do disco, também conhecido como prato, é organizado em áreas circulares concêntricas chamadas trilhas, e que são numeradas de 0 a T-1, endereço da última trilha do disco. Essas separações têm seu início a partir da 0 (mais externa do disco) até a mais interna T-1, como mostram as figuras a seguir. Todas as trilhas possuem a mesma capacidade de armazenar *bytes* e isso se deve ao fato de que existe uma diferença de densidade de gravação entre a trilha mais externa e a mais interna.

Dessa forma, para que se evitem erros devido à proximidade do final do prato nas extremidades, além de evitar que haja uma diferença muito grande em relação às densidades entre as trilhas externas, a sua área magnetizável fica posicionada na parte central do prato, de forma que não haverá trilhas próximo à borda do prato ou mesmo ao eixo central de rotação.

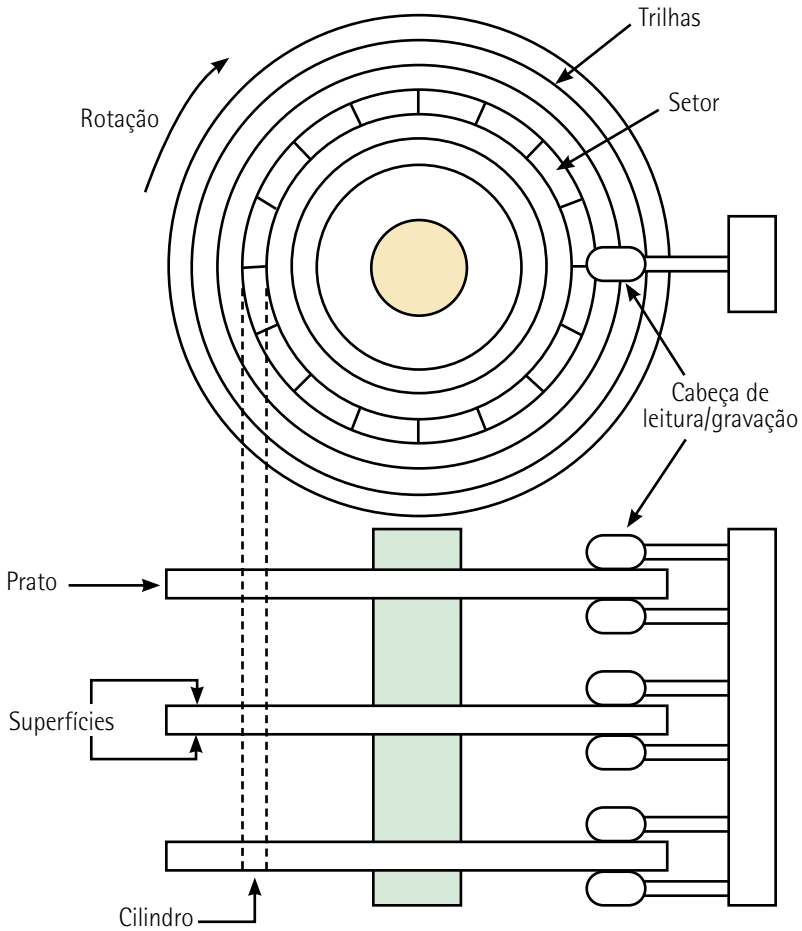


Figura 143 – Organização interna de um disco rígido magnético

As trilhas são constituídas de superfícies magnetizáveis e são divididas em partes menores de tamanho fixo, denominadas setores, e que servem de parâmetro de unidade de armazenamento. Cada setor do disco rígido é organizado para gravar os 512 bytes para dados, de forma que o sistema de leitura/gravação não seja confundido com outro setor.

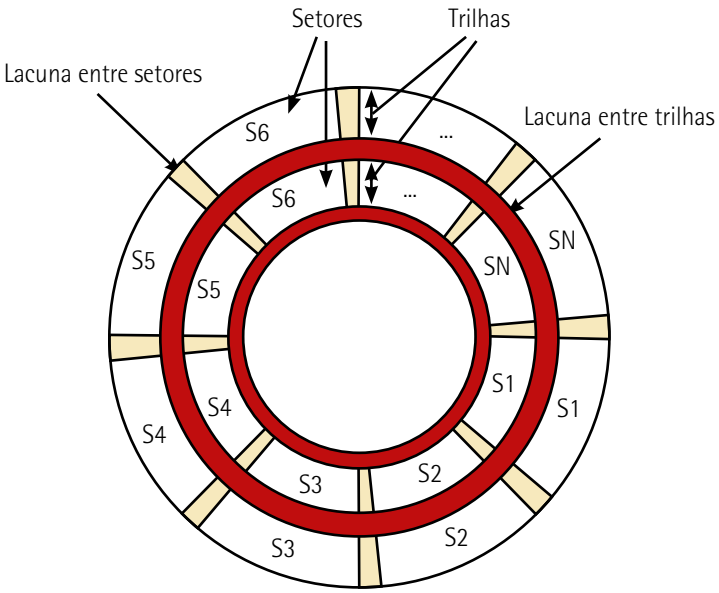


Figura 144 – *Layout* de separação interna do disco rígido

Os setores possuem um campo inicial, que se inicia antes dos *bytes* destinados para dados, e que é conhecido como preâmbulo. O preâmbulo contém elementos necessários para sincronizar a cabeça antes de cada leitura/gravação. Outro espaço existente entre cada par de setores é conhecido como espaço morto ou *gap* intersetorial e é utilizado para evitar a superposição de leitura/gravação nos casos em que os setores fiquem contíguos, devido à alta velocidade da rotação do disco no eixo. Os setores também possuem um campo denominado ECC (figura a seguir), que contém *bits* calculados durante a transmissão dos dados e possuem como finalidade proteger o bloco de 512 bytes de possíveis erros decorrentes da transmissão.

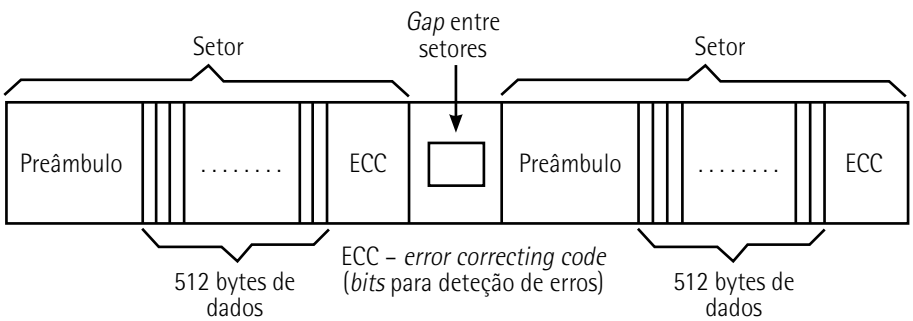


Figura 145 – Organização de trilhas e setores em um disco rígido

Um disco rígido completo é constituído não só por um, mas diversos pratos, que giram constantemente e na mesma velocidade em torno do seu eixo central. Um braço mecânico transporta a cabeça de gravação/leitura, efetuando um movimento na transversal, realizando as operações de gravação/leitura em cada uma das trilhas solicitadas na operação. A movimentação do braço é realizada através de um mecanismo, denominado atuador, movimentado pela atração de uma bobina. Existem diferentes métodos de acesso aos discos rígidos, além de diversas unidades de transferência que são utilizadas de

acordo com o sistema operacional escolhido. Alguns sistemas efetuam a transferência de dados do disco rígido para o processador e a memória principal de setor por setor.

Observação

Em outros sistemas, devido a sua maior capacidade e também ao volume de transferências, é realizada a movimentação de um grupo de setores ou *cluster* (vários setores) de cada vez.

6.1.2 Propriedades de funcionamento dos discos rígidos

O acesso a dados/instruções em um disco rígido ocorre em uma série de etapas, também conhecidas como micro-operações. De forma resumida, pode-se definir as seguintes etapas para que o acesso ao disco seja completado:

- Interpretação do comando que realiza a solicitação de E/S. Etapa em que um endereço físico que contém os dados desejados é localizado pelo número do cilindro ou setor e é conhecido como geometria do disco.
- Movimentação do braço em direção à trilha desejada ou cilindro, conhecido também como *seek*.
- Localização do setor, onde a cabeça de leitura/gravação passa por cima do setor desejado.
- Transferência de *bits* através de condutores com destino a um *buffer* e depois para uma área da memória principal especificada na operação de busca.

O tempo gasto entre o início da operação de leitura/escrita e o seu término é conhecido como tempo de acesso e é constituído pela soma dos quatro tempos distintos, correspondentes às seguintes etapas (MONTEIRO, 2019):

- **Tempo necessário para a interpretação do comando**: é todo o período gasto para que o sistema operacional seja capaz de interpretar o comando, transmitir a instrução para o controlador de disco e a conversão no endereço físico desejado.
- **Tempo para realização da busca (*seek*)**: é o tempo gasto para realizar a interpretação do endereço desejado pela unidade de controle além da movimentação mecânica do braço até as trilhas desejadas.
- **Tempo de latência**: é o período decorrido a partir da chegada da cabeça de leitura/gravação na trilha e a passagem pelo setor. O tempo médio para a latência será igual à metade do tempo gasto para que o disco efetue uma volta completa, ou seja, para que seja possível que o setor gire completamente ao redor da cabeça, o que é inversamente proporcional à velocidade de rotação do disco, estimada entre valores de 5.400, 7.200 e 10.000 rpm (rotações por minuto).

- **Tempo para realizar a transferência dos dados**: é o tempo gasto para que seja realizada a transmissão dos sinais elétricos em formato de *bits*, com valores típicos da ordem de 400 a 800 microssegundos.

6.1.3 Cálculo de espaçamento e armazenamento em discos rígidos

O cálculo de espaço necessário para o armazenamento em discos rígidos consiste na adoção do setor ou grupo de setores, como a unidade de transferência fixa, deixando de existir, portanto, o fator de bloco de dados variáveis. Dessa forma, será necessário o cálculo de quantidade de trilhas ou cilindros que serão consumidos por arquivo. Um cilindro é constituído por um conjunto de todas as trilhas que estão na mesma posição relativa, como exemplificado na figura:

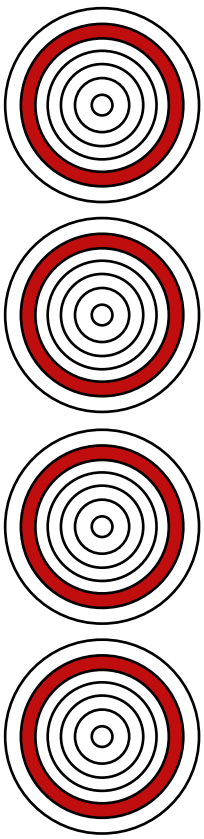


Figura 146 – Trilhas e cilindros internos

Exemplo de aplicação

Os arquivos de dados são constituídos de N registros lógicos e, considerando a divisão de trilhas em setores, cada um deles com uma quantidade fixa de *bytes* para o armazenamento, pode-se implementar o cálculo com o seguinte exemplo:

Deseja-se conhecer qual será o espaço necessário para armazenar em disco rígido um arquivo com 80.000 bytes. O disco para realizar o armazenamento possui 40 trilhas, além de 9 setores de 512 bytes cada. Como descobrir?

Resolução

A quantidade de setores necessária será 80.000/512 = 156,25. Mas como o número de setores deve ser composto por um número inteiro, então serão necessários 157 setores. O cálculo da quantidade de trilhas consiste em dividir o total de setores obtido (157) pelos setores disponíveis no disco (9), resultando em: 157/9 = 17,4 trilhas, que também precisa ser adequado para 18 trilhas, visto que esse número também não é fracionado.

6.1.4 Desempenho em discos rígidos

O desempenho funcional dos discos rígidos depende de vários fatores, como quantidade de operações de E/S, taxa de transferência, quantidade de *bytes* envolvidos na operação além do sistema operacional utilizado. Quando uma unidade de disco rígido está em operação, o disco estará girando de forma constante em altas velocidades, de modo que para realizar a leitura/gravação, a cabeça necessita ser posicionada no início do setor e na trilha desejada.

O processo de seleção da trilha envolverá a movimentação da cabeça em um sistema de cabeça móvel ou mesmo selecionando eletronicamente uma cabeça no sistema de cabeça fixa. Para sistemas de cabeça móvel, o tempo gasto para o posicionamento da cabeça na trilha é conhecido como tempo de busca ou *seek time* (STALLINGS, 2010). Após a trilha ser selecionada, o controlador de disco rígido aguarda até que o setor escolhido esteja alinhado com a cabeça. Assim, o tempo gasto até que o início do setor encontre a cabeça é conhecido como atraso rotacional ou latência rotacional. Além desses tempos gastos para o alinhamento da cabeça na trilha, existe a soma do tempo de busca, quando ocorre o atraso rotacional que será igual ao tempo de acesso, que é determinado pelo tempo gasto para o posicionamento da cabeça na operação de leitura/gravação. Dessa forma, quando a cabeça já está posicionada corretamente, a operação de leitura/gravação será realizada quando o setor se mover sob a cabeça, realizando de fato a transferência dos dados.

Em uma rede de computadores, onde existam servidores, existe uma técnica utilizada para a detecção da posição rotacional dos discos, conhecida como RPS (*rotational positional sensing* ou sensor de posição rotacional). Essa técnica é acionada quando o comando de busca é emitido, daí o canal é liberado para tratar outras solicitações de E/S. Assim, quando a busca termina, o dispositivo irá determinar quando os dados solicitados na operação estarão posicionados sob a cabeça do disco. À medida que o setor se aproxima da cabeça, o dispositivo procura o caminho solicitado para que ocorra a comunicação com o sistema. Se a unidade de controle estiver ocupada em outra operação de E/S, então uma nova tentativa para realizar a conexão é solicitada, e, se houver falha, ainda assim o disco necessitará girar por mais uma volta inteira antes de uma nova reconexão. Esse processo é conhecido como falha de RPS, e é um elemento que deverá ser somado à linha de tempo de transferência para o disco rígido, como mostra a figura a seguir.

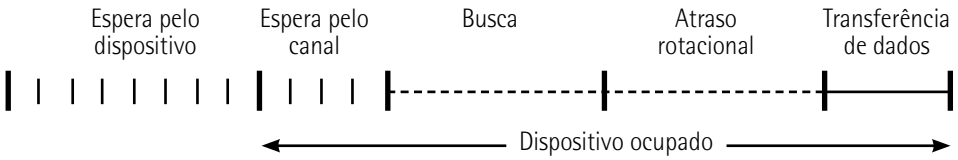


Figura 147 – Temporização de transferência em discos rígidos

O cálculo do tempo de transferência é dependente da velocidade de rotação dos discos rígidos e obedecem à expressão: T = b/rN, onde T é o tempo de transferência, b é o número de *bytes* a serem transferidos, N é o número de *bytes* em uma trilha e r a velocidade de rotação dada em rpm, ou rotações por minuto. Por exemplo, em uma operação de transferência de dados onde o número de *bytes* a serem transferidos é de 1.024 Mbytes, o número de *bytes* da trilha é de 512 bytes e a velocidade de rotação do disco é de 7.200 rpm. Calculando o tempo total T para a realização da transferência, a resposta seria: T = 1.024/7.200 * 512 → 277,77 μs (microssegundos).

6.2 RAID em discos rígidos

Em 1988, Patterson, Gibson e Katz sugeriram que seis tipos de organizações específicas de paralelismo poderiam ser utilizados para a melhoria de desempenho e a confiabilidade do armazenamento de dados nos discos rígidos magnéticos. As ideias desses autores foram empregadas pela indústria na fabricação dessa nova tecnologia para os dispositivos de E/S e ficaram conhecidas como RAID (*redundant array of inexpensive disks* ou arranjo redundante de discos baratos) (TANENBAUM; AUSTIN, 2013).

Lembrete

O processamento realizado de forma paralela nos computadores pode acelerar seu desempenho.

A ideia por detrás do sistema RAID envolve a instalação de uma caixa contendo vários discos rígidos, geralmente próximos a um servidor, e substituir a placa controladora de disco por uma placa controladora RAID. Dessa forma, de acordo com a configuração adotada, os discos rígidos, embora sendo muitos, aparecerão como apenas um único disco para o sistema operacional. Outra propriedade importante no sistema RAID se refere à distribuição dos dados pelos diferentes discos, a fim de permitir que a operação ocorra em paralelo. Com essa finalidade, Patterson, Gibson e Katz desenvolveram diferentes esquemas ou níveis que ficaram conhecidos como RAID 0 até o nível RAID 5.

6.2.1 RAID 0

Essa configuração (figura a seguir), embora faça parte do sistema de esquemas de redundância RAID, não possui uma redundância. Ele consiste em dividir os discos em tiras (*stripes*), distribuídos e intercalados (*striped*), o que auxilia no processo das solicitações de blocos de dados, que estarão distribuídos pelos vários discos. Isso tornará o processo mais veloz, visto que a resposta para essa solicitação será executada em paralelo por todos os discos envolvidos no processo.

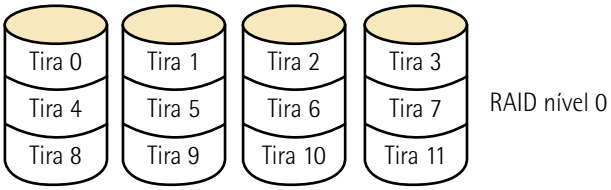


Figura 148 – RAID 0

6.2.2 RAID 1

Essa configuração de RAID consiste na implementação de redundância através da duplicação ou mesmo triplicação de um determinado volume de dados em todos os discos, criando uma espécie de "espelhamento", conforme se observa na figura a seguir. Dessa forma, cada transação de leitura/gravação de dados em um disco também ocorrerá nos outros definidos no espelhamento. É possível realizar a combinação dos RAID 0 e RAID 1, garantindo maior rapidez e confiabilidade. O uso de um sistema combinado de RAIDs geralmente é atribuído para servidores de arquivos ou grandes *datacenters*.

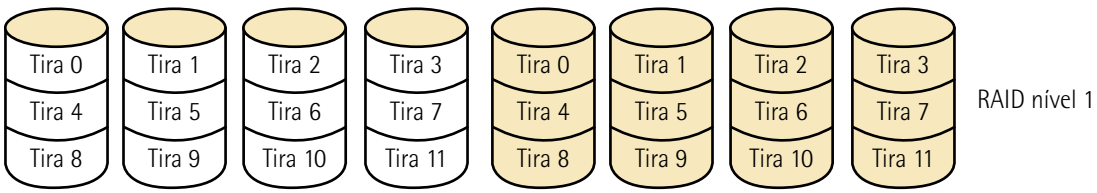


Figura 149 – RAID 1

6.2.3 RAID 2

Nesse sistema, o RAID incluirá um mecanismo para detecção de falhas como um código de correção de erros. No processo de leitura/gravação dos discos, os dados solicitados e o código de correção de erro associado serão entregues ao controlador do *array*. Se algum *bit* contiver erro, o controlador poderá reconhecê-lo e corrigi-lo de forma instantânea, utilizando algum processo de paridade de *bits* para a recuperação.

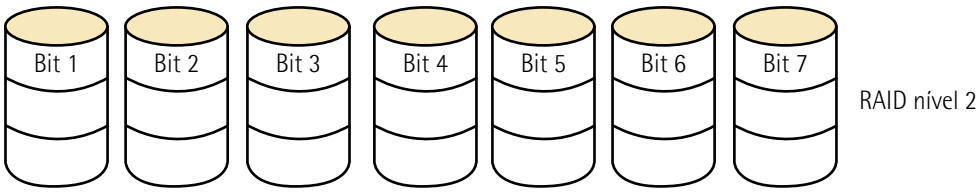


Figura 150 – RAID 2

6.2.4 RAID 3

Semelhante ao RAID 2, no RAID 3 (figura a seguir) os dados são divididos pelos vários discos rígidos, utilizando-se um disco adicional para a verificação de erros através do processo de paridade. O uso dessa técnica pode garantir uma maior integridade dos dados quando houver a necessidade de recuperação.

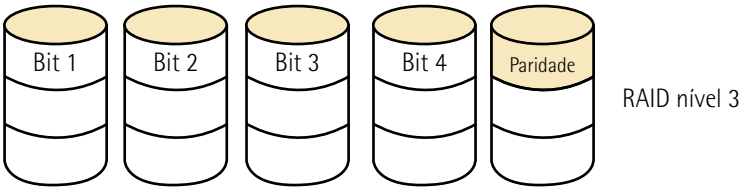


Figura 151 – RAID 3

6.2.5 RAID 4

No esquema de RAID 4 (figura a seguir), os dados são divididos igualmente entre todos os discos, com exceção de um, que é utilizado exclusivamente para inserir os dados necessários para realizar a paridade. A única diferença do RAID 4 com os anteriores consiste no tamanho dos blocos de armazenamento serem maiores do que do RAID 3, tornando o rendimento melhor no processo de leitura. Esse sistema é indicado para utilização com arquivos de tamanho grande, em que se requer uma maior integridade dos dados, pois a cada operação de leitura é realizada a paridade dos dados, resultando também em uma maior confiabilidade.

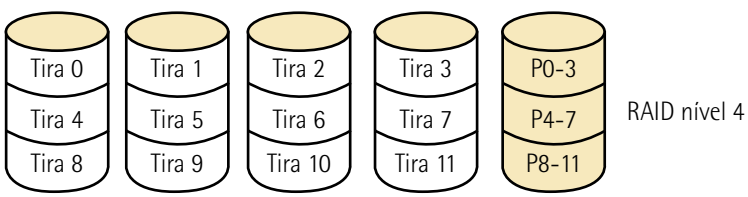


Figura 152 – RAID 4

6.2.6 RAID 5

No RAID 5 (figura a seguir), a paridade não se restringe a somente uma unidade de disco como foi observado no RAID 4, mas toda a matriz de discos possui a paridade distribuída. Nessa situação, o tempo de gravação será menor, pois não será necessário acessar um único disco de paridade em cada operação de leitura/escrita, mas sim acessar o disco que possui a paridade necessária para a recuperação de dados em um processo específico.

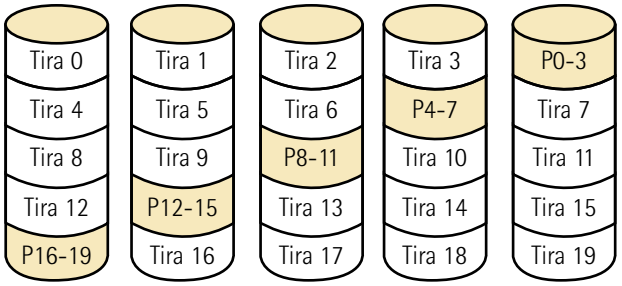


Figura 153 – RAID 5

Saiba mais

Aprenda mais sobre configurações de *software* e *hardware* dos sistemas RAID em:

VIANA, A. L. S. Um pouco sobre RAID. Configuração via *software* e via *hardware*. *DevMedia*, 2012. Disponível em: https://bit.ly/3rF35Hl. Acesso em: 2 mar. 2021.

6.2.7 Memória virtual

Outro conceito utilizado em discos rígidos magnéticos é conhecido como memória virtual.

Observação

A memória virtual tem como objetivo principal utilizar o disco como uma extensão da memória principal, aumentando, assim, o espaço para o endereçamento disponível da RAM.

Essa técnica é utilizada porque, no geral, os computadores pessoais não possuem uma quantidade elevada de memória RAM, tornando insuficiente a execução de algumas aplicações que envolvem, por exemplo, processamento gráfico, processamento de vídeo, programação de alto desempenho de dados, entre outros. Ao utilizar a memória virtual, a memória principal possuirá uma região muito maior do que a original para realizar o endereçamento. Essa área no disco rígido é reconhecida como um arquivo de páginas, pois contém algumas porções da memória principal. Uma forma simples de entender como funcionam as memórias virtuais é saber o conceito de posição imaginária da memória principal, endereçado no disco rígido pelo sistema operacional. A memória virtual também é conhecida como páginas, o que é uma maneira de dividir a memória em blocos com tamanhos fixos, dividindo também os programas com blocos do mesmo tamanho. Alguns conceitos de memória virtual envolvem os seguintes termos:

- **Endereço virtual**: é o endereço lógico no programa que o processo utilizará. Dessa forma, sempre que o processador gerar o endereço, ele estará relacionado ao espaço de endereçamento virtual.

- **Endereço físico**: é o endereço real na memória RAM.

- **Mapeamento**: é o mecanismo em que os endereços virtuais serão traduzidos para endereços físicos.

- **Quadro de páginas**: são pedaços ou blocos de mesmo tamanho em que a memória RAM é dividida.

- **Páginas**: são pedaços ou blocos onde a memória virtual é dividida, de modo que cada um desses blocos possuem o mesmo tamanho que um quadro da página.

- **Paginação**: é o processo de copiar uma página virtual do disco para um quadro de página que está na memória RAM.

- **Fragmentação**: é um pedaço da memória que se torna não utilizável.

- **Falha de página**: é um ou mais eventos que ocorrem quando a página que é solicitada não está na memória RAM e deve ser copiada do disco rígido para ela.

Como a memória RAM e a memória virtual são divididas em páginas de tamanhos iguais, algumas partes do espaço de endereçamento dos processos podem ser movidas para a memória RAM, e não precisam ser armazenados de forma contígua. Outro fator importante em termos de paginação são os registros utilizados para gravarem as informações das páginas, conhecido como tabela de páginas. Dessa forma, cada processo armazenado possui sua própria tabela, que irá armazenar sua posição física de cada página virtual do processo. Assim, uma dessas tabelas possui N linhas, onde N será o número de páginas virtuais do processo.

6.3 *Drive* de estado sólido (SSD)

Da mesma forma que os discos rígidos magnéticos, os SSD, do inglês *solid state drive* (figura a seguir), são memórias não voláteis e se apresentam como uma alternativa de memória de alta velocidade. Melhor que as memórias do tipo RAM, é constituído por várias células unitárias, cujo principal componente é o transistor. Os transistores são dispositivos eletrônicos que possuem como uma das suas principais características a comutação, ou seja, eles modificam o seu estado atual. Porém, ao comutarem, eles se desgastam chegando ao ponto de não funcionarem mais, conhecido como falha de um transistor. Essa falha também pode ser explicada como uma injeção de portadora, que ocorre quando uma carga elétrica é colocada dentro do transistor, alterando o seu estado anterior, tornando-o permanentemente ligado ou desligado.

Figura 154 – *Drive* de estado sólido

Como mencionado, embora tenha alterado um estado de um transistor para permanecer permanentemente constante, a empresa Toshiba, na década de 1980, viu que era possível aproveitar essa falha para criar um mecanismo de memória não volátil, a memória *flash* (TANENBAUM; AUSTIN, 2013).

Conforme pode ser observado na figura a seguir, uma célula de memória *flash* é composta de um transistor especial. Nele há uma espécie de porta flutuante que é carregada e descarregada através do uso de voltagens maiores do que os 5 volts convencionais, em se tratando de dispositivos para computadores. Logicamente, antes de a célula unitária *flash* ser programada, o que corresponde à carga elétrica embutida, essa porta flutuante não afetará a operação do transistor, de modo que ela continuará atuando como um isolador entre a porta de controle e o canal, tornando-o apenas um transistor simples. A referida alta tensão é baseada em 12 volts e aplicada na porta de controle, que irá acelerar o processo referido de injeção de portadora na porta flutuante. A carga elétrica inserida na porta flutuante tornará o transistor *flash* negativamente carregado, aumentando, assim, a tensão necessária para que ele seja ligado. Uma atenção mais baixa do que 12 volts será necessária para determinar se a porta flutuante está ou não carregada, o que resultará em um valor que será codificado como *bits* 0 ou 1. Dessa forma, uma das principais vantagens das memórias SSD é que, uma vez que a carga é inserida na célula unitária, ela irá permanecer estável, mesmo que não haja alimentação elétrica, tornando-a não volátil.

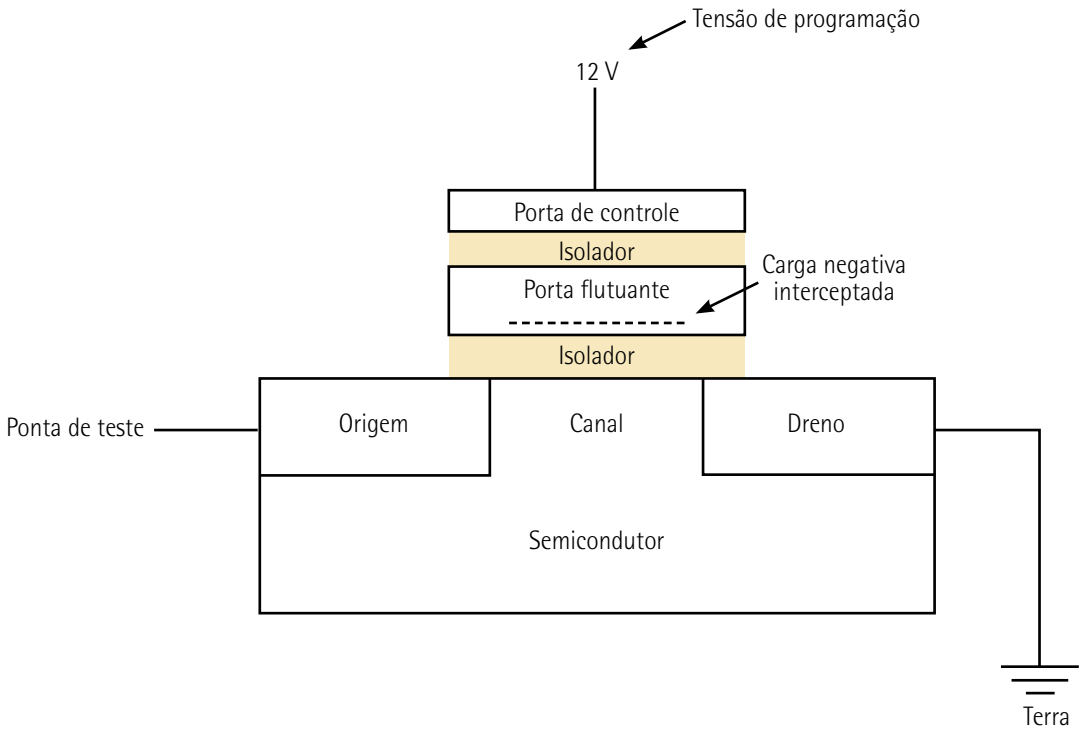


Figura 155 – Célula de memória do tipo *flash*

Diferentemente dos discos rígidos magnéticos, que possuem tanto parte elétrica quanto partes mecânicas, as memórias SSD possuem um desempenho superior, visto que são basicamente dispositivos inteiramente eletrônicos, sem peças mecânicas. Outra grande vantagem dos dispositivos SSD é a sua utilização em dispositivos móveis, como, por exemplo, *notebooks*, a fim de evitar trepidações ou movimentos bruscos, que dificilmente irão impactar no seu correto funcionamento.

Em relação à velocidade, os dispositivos SSD podem operar duas ou mesmo três vezes mais rápido do que um disco rígido convencional, que possui acesso a dados em até 100 MB/s. Logicamente, não há somente vantagens nesse tipo de dispositivo, podendo-se destacar a sua alta taxa de falha. Essas falhas ocorrem geralmente devido ao fato de que uma célula *flash* unitária tipicamente pode ser escrita aproximadamente cem mil vezes até que ela deixe de funcionar corretamente. Isso quer dizer que o processo de injetar elétrons danifica a célula aos poucos, assim como os isolantes que estão ao seu redor. Algumas técnicas para aumentar o tempo de vida dos dispositivos SSD são utilizadas atualmente. Uma das técnicas é conhecida como nivelamento de desgaste, e é empregada a fim de espalhar as escritas por todas as células *flash*, não tornando algum ou alguns pontos viciados. Dessa forma, todas as vezes que um bloco precisa ser escrito, há uma distribuição homogênea por ele, ou mesmo a busca por blocos que não foram escritos no processo anterior.

Ao utilizar o nivelamento de desgaste, um drive SSD poderá suportar uma quantidade de escritas que será igual ao número de escritas que uma célula poderia sustentar multiplicado pelo número de blocos no *drive*.

6.4 Discos ópticos

Desenvolvidos em 1980 pelas empresas Philips e Sony, os discos ópticos eram inicialmente utilizados para gravar em programas de televisão e, na sequência, devido à sua grande capacidade de armazenamento para a época, também passaram a serem utilizados no armazenamento de dados de computadores. A grande popularidade dos discos ópticos se deu, entretanto, na sua utilização para gravação de músicas, substituindo os discos de vinil. Entre as variedades desses discos ópticos podem ser citados os CD-ROM, CDs regraváveis, DVDs e *blu-rays*.

6.4.1 CD-ROM

Entre suas especificações, os CDs (*compact discs*) possuem 120 mm de diâmetro, 1,2 mm de espessura e um orifício de 15 mm em sua área central, como mostra a figura a seguir.

Figura 156 – Mídias de CD-ROM

No ato de sua fabricação, um CD é preparado com a utilização de um *laser* infravermelho de alta potência que produz orifícios com 0,8 micrômetros de diâmetro em um disco mestre, que é revestido de vidro. A partir desse disco mestre será fabricado um molde com saliências onde estavam originalmente os orifícios produzidos pelo *laser*. Na sequência, injeta-se o policarbonato, uma espécie de plástico fundido no molde, a fim de formar um CD com o mesmo padrão de orifícios do mestre. Logo após, é depositada uma camada fina de alumínio refletivo sobre a camada já depositada anteriormente de policarbonato, produzindo assim um verniz de proteção. Por fim, insere-se uma etiqueta com a nomenclatura da finalidade daquele CD, sendo as possibilidades diversas, como gravar músicas ou dados. Essas marcas inseridas no substrato de policarbonato são conhecidas tecnicamente como depressões ou *pits*, do inglês, e as áreas que estão posicionadas entre as depressões são conhecidas como planos, *lands* em inglês, como exemplificado na figura a seguir.

O processo de leitura de um CD-ROM ocorre quando um diodo *laser* de baixa potência emite uma luz infravermelha com comprimento de onda de 780 nm sobre as depressões e planos que estão no disco, realizando a leitura dos dados contidos naquele local por onde a luz passa. Devido às depressões possuírem uma altura de um quarto do comprimento de onda da luz do *laser*, a luz que é refletida na depressão terá uma defasagem de meio comprimento de onda em relação à que é refletida na superfície que a circunda. O resultado disso é que as duas partes irão interferir umas com as outras de forma destrutiva, o que envolverá a devolução de menos luz ao fotodetector do aparelho de reprodução de luz que reflete. Então é justamente dessa forma que um aparelho interpreta a diferença entre uma depressão e um plano, que envolverá justamente a leitura dos *bits* 0 e 1.

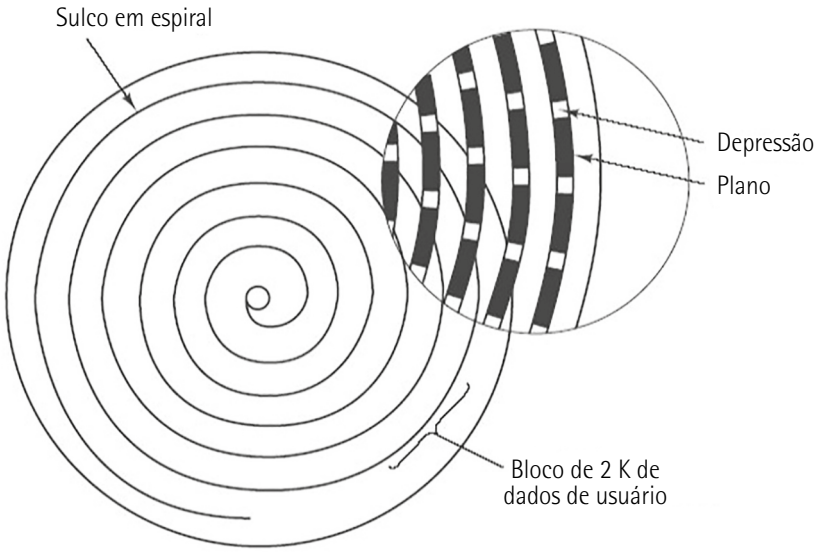


Figura 157 – Estrutura de gravação em CD-ROM

Embora possa parecer muito mais simples interpretar uma depressão como sendo um *bit* zero e o plano como sendo o *bit* 1, é muito mais confiável utilizar uma transição entre a depressão e o plano ou o plano, e a depressão para representar o *bit* 1 é a sua ausência para representar o *bit* zero. E é justamente dessa forma que os *bits* 0 e 1 são interpretados em um CD-ROM, como mostra o esquema da figura a seguir. As depressões em planos são escritas no CD-ROM em uma espiral única e contínua que começa próximo ao orifício central e se estende até uma distância de 32 mm em direção a sua borda. Uma espiral conterá 22.188 rotações ao redor do disco, o que daria cerca de 600 mm. Para que uma música seja tocada em um CD-ROM e haja uma taxa uniforme, é necessário que os planos e as depressões passem sob a luz a uma velocidade linear e constante. Como consequência, a taxa de rotação deve ser reduzida de forma contínua à medida que a cabeça de leitura do dispositivo se move, desde a parte interna até a parte externa do CD-ROM. A parte interna do disco geralmente possui uma taxa de rotação de 530 rpm, enquanto a taxa na parte mais externa é de 200 rpm. Essa variação na velocidade deve ser regulada da mesma maneira que ocorre nos discos rígidos magnéticos que possuem um sistema de velocidade angular constante.

Em relação a sua organização de dados, os dispositivos de CD-ROM operam com 75 setores/s, o que dá uma taxa de dados de 153.600 bps. Um CD-ROM padrão de áudio possui um espaço de 74 minutos de música, e se usado para dados, possui aproximadamente uma capacidade de 682.000 bits, o que geralmente é arredondado para 650 MB, pois um MB será igual a 2^{20} bytes, ou seja, 1.048.576 bytes.

6.4.2 DVD

Devido à necessidade de aumento da capacidade de gravação de dados em CDs, em meados da década de 1990, foi desenvolvida uma variação dessa mídia para que ela tivesse a capacidade de armazenar não somente músicas como também filmes. Com essa demanda, surgiu o DVD, do inglês *digital video disc*, e que atualmente também é conhecido como *digital versatile disc*. Os DVDs possuem o mesmo desenho dos CDs, porém com algumas novidades:

- Depressões com tamanhos reduzidos de 400 nm.
- Espirais mais estreitas, com uma separação entre trilhas de 740 nm.
- Leitura de gravação utilizando um *laser* de 650 nm.

Além dessas melhorias, que ocasionaram um aumento na capacidade em 7 vezes, chegando a um total de 4,7 GB, um dispositivo DVD funciona com diferentes velocidades. A velocidade de transferência do DVD 1x possui 1,4 KB/s e pode alcançar velocidades 4x (5.400 KB/s) ou maiores. Assim como os CDs de maiores capacidades, os DVDs podem gravar em ambas as faces de sua mídia, chegando à capacidade de armazenamento:

- Uma face, uma camada com 4,7 GB.
- Uma face, duas camadas totalizando 8,5 GB.
- Duas faces, uma camada totalizando 9,4 GB.
- Duas faces, duas camadas totalizando 17 GB.

6.4.3 *Blu-ray*

Em sucessão ao DVD, surgiu o *blu-ray*, assim chamado devido à frequência de luz na faixa do *laser* azul utilizado para leitura dos dados gravados. A frequência azul utilizada possui um comprimento de onda menor e uma frequência maior, o que obriga no ato da leitura que as depressões e os planos também sejam menores. Dessa forma, com mais depressões e planos na mídia é possível o armazenamento muito maior de dados. A figura a seguir mostra, de forma resumida, um comparativo entre as principais mídias óticas apresentadas, incluindo a variação das depressões e regiões planas, que influenciarão a quantidade de dados gravados.

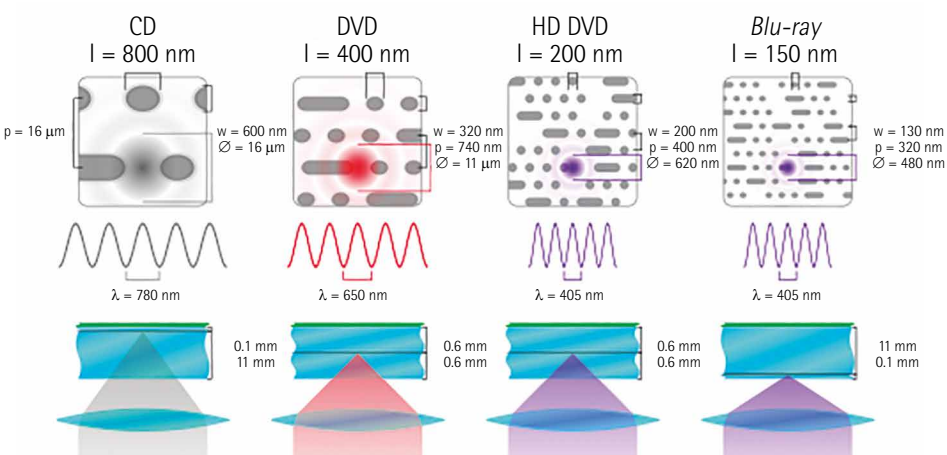


Figura 158 – Variações entre os tamanhos das depressões e planos em função da luz *laser* inserida

Saiba mais

Conheça mais sobre os últimos avanços em discos ópticos em:

CRIADO disco óptico que armazena dados em cinco dimensões. *Inovação Tecnológica*, 22 maio 2009. Disponível em: https://bit.ly/3t7g6d6. Acesso em: 3 mar. 2021.

6.5 Disquetes

Conhecidos também como *floppy disk drivers*, os disquetes (figura a seguir) possuem características semelhantes ao disco rígido, embora com menor capacidade de armazenamento. A grande diferença entre esses dispositivos está, além da capacidade, na velocidade de acesso, o tempo de transferência de dados e, principalmente, a sua portabilidade, ou seja, disquetes são muito mais simples de serem removidos do que os discos rígidos.

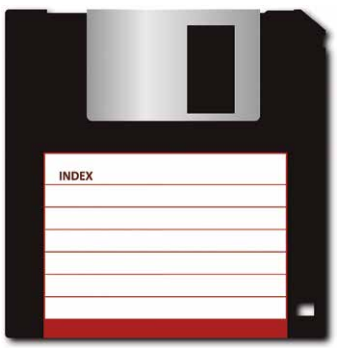


Figura 159 – Disquete

Observação

Os disquetes foram desenvolvidos na década de 1960 como sendo uma alternativa mais simples e barata do que os discos rígidos da época.

Entretanto, devido à sua facilidade e usabilidade, os disquetes se tornaram muito populares entre usuários tanto domésticos quanto empresariais; isso aconteceu juntamente à alavancagem dos computadores pessoais.

As informações são gravadas de forma idêntica como ocorre nos discos rígidos. Em trilhas esses setores estão posicionados na superfície do disco magnetizável. Para que seja acionado o disquete em um computador, há um mecanismo que realiza a leitura ou gravação para cada superfície.

Geralmente um computador possui um *drive* para acionar o disquete. Esse *drive* contém um sistema eletromecânico que irá acessar as trilhas entre setores dos disquetes. No geral, os usuários desse dispositivo dispunham de alguns modelos diferentes que eram oferecidos, conforme três fatores principais:

- capacidade total de armazenamento;
- taxa de transferência de *bits*;
- tempo médio para acesso ao dispositivo.

Em relação à taxa de transferência dos disquetes, logicamente não se compara com a dos discos rígidos magnéticos, que podem alcançar taxas de transferência na ordem de MB/s, enquanto em disquetes essa taxa não chega a centenas de KB/s.

6.6 Fitas magnéticas

As fitas magnéticas desenvolvidas em meados dos anos 1950 foram muito utilizadas como dispositivos para armazenamento de dados e instruções, visto que no início da computação os modos de realizar a leitura e gravação de dados eram um processo um tanto quanto lento, pois se baseavam no uso de cartões perfurados. O princípio de funcionamento das fitas magnéticas é bastante semelhante ao processo de leitura e gravação de fitas utilizadas em dispositivos de som. Eles consistem no uso de dois carretéis de fita que se desenrolam de um lado para o outro, de maneira que a fita passará por um par de leitura ou gravação em velocidades constantes. A figura a seguir mostra um dispositivo de fita magnética utilizada em computadores como *mainframes* na década de 1980.

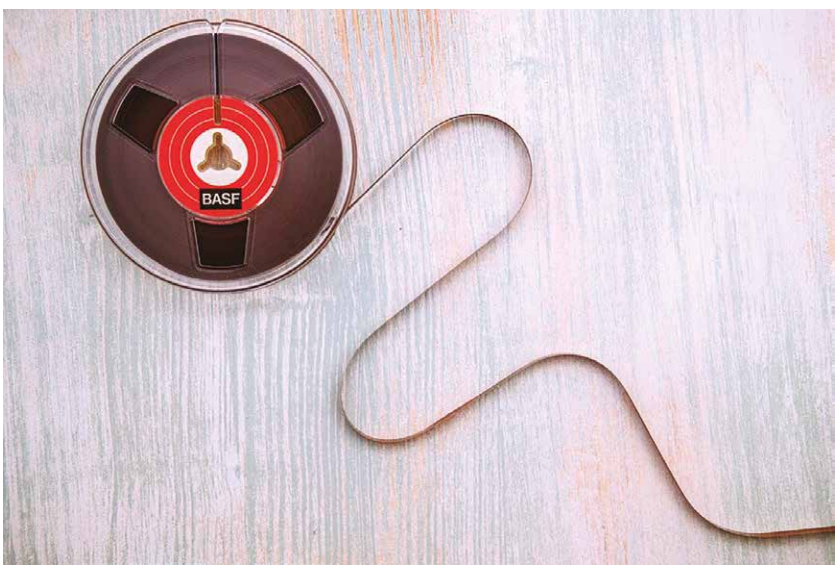


Figura 160 – Rolo de fita magnética

Diferentemente do tipo de acesso à memória que ocorre em dispositivos como discos rígidos magnéticos ou memória RAM, o acesso aos dados contidos na fita magnética é realizado apenas sequencialmente, ou seja, cada informação é armazenada uma após a outra e a sua leitura é realizada também num processo sequencial. Dessa forma, a localização do registro desejado na operação de leitura começará a partir do início da fita, variando registro a registro, até que se identifique o registro desejado, como corre numa fita cassete. De modo mais técnico, a fita magnética é constituída por uma tira contínua de um material plástico, que é coberto com alguns elementos magnetizáveis, em que os *bits* serão gravados. Assim, os *bits* gravados em um certo sentido de magnetização irão representar o *bit* 0, enquanto em outro sentido magnetizável irá representar o *bit* 1. As posições nas trilhas são magnetizáveis quando esses campos são preenchidos pela passagem de corrente elétrica em uma bobina que está presente na cabeça de gravação. A figura a seguir mostra como exemplo um trecho de uma fita magnética em que é possível observar os *bits* representados em linhas paralelas.

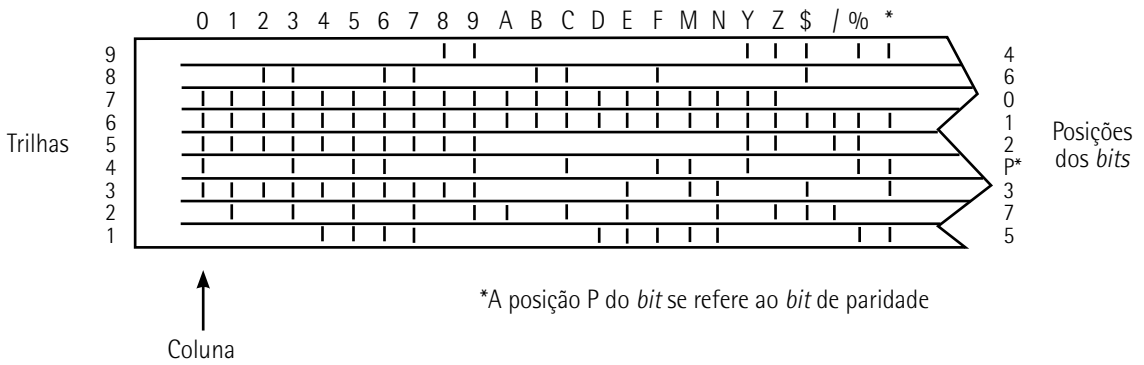


Figura 161 – Organização de linhas e colunas em diferentes trilhas de uma fita magnética

Os dados armazenados em canais paralelos, conhecidos como trilhas, percorrem toda a fita magnética. A quantidade de trilhas pode variar entre 7 ou 9, embora fitas com 7 trilhas já estejam obsoletas há muitas décadas, de modo que somente as com 9 trilhas continuam a ser fabricadas. Como mostrado na figura, cada caractere será armazenado verticalmente, 1 bit a cada trilha, adicionando-se 1 bit de verificação de paridade, o que irá completar as 9 trilhas necessárias.

Entre as características das fitas magnéticas, uma das mais interessantes é a respeito do seu sistema de transporte de unidades de fita que consiste no processo de parada e partida das rotações dos carretéis. Entre as colunas da fita magnética é obtida a gravação, que pode variar de acordo com a velocidade de passagem da fita, podendo demonstrar que uma das principais características referentes ao desempenho das unidades da fita está relacionada à sua densidade. Fitas magnéticas possuem em geral densidades de 800 caracteres por polegadas ou BPI (*bytes per inch*), chegando até mesmo em densidades de 6.250 caracteres por polegada. Devido à implementação de dispositivos com maior desempenho e velocidade para gravação de dados, como, por exemplo, DVD e *blu-ray*, as fitas magnéticas caíram em desuso principalmente para usuários domésticos, ficando restritas em grandes *datacenters*, como dispositivos de *backup*.

Além dessas características, as fitas magnéticas possuem algumas marcas de ações para o controle da gravação de dados, como, por exemplo, uma marca refletora no seu início e no final do carretel, para impedir que a fita ultrapasse os limites e ocorra algum dano a ela.

Saiba mais

Aprenda mais sobre as novas tendências de armazenamento de dados ultrarrápidos em:

ULTRAMAGNETRON: gravação magnética à velocidade da luz. *Inovação Tecnológica*, 18 jun. 2014. Disponível em: https://bit.ly/3v8jcPQ. Acesso em: 3 mar. 2021.

Resumo

Nesta unidade, aprendemos um pouco mais sobre os sistemas de memória. Vimos que elas podem ser subdivididas entre memórias internas, ou seja, aquelas que estão posicionadas ou dentro da CPU (registradores, memória *cache*) ou acopladas à placa-mãe, como as memórias RAM e ROM.

Vimos que as memórias do tipo externa são aquelas acopladas externamente às placas-mãe como os discos rígidos, *pen drive*, fitas magnéticas, mídias ópticas, disquetes etc.

Aprendemos também que existem regras para endereçamento e armazenamento dos dados e instruções nas memórias.

Foi possível apender como os discos-rígidos podem operar de forma segura e com melhor desempenho, utilizando-se o sistema de discos redundantes, RAID.

Exercícios

Questão 1. Uma empresa vem trabalhando em um projeto de um sistema de controle de um equipamento de ar-condicionado. Parte desse sistema corresponde a um programa que vai ser executado em um pequeno sistema computacional, interno ao equipamento. Esse sistema é composto por memória RAM, memória ROM, CPU e barramentos que interligam tais partes. Sabe-se que 10 bits são utilizados para endereçar cada uma das posições da memória, e que cada posição da memória pode armazenar 8 bits. Admita que o sistema tenha a quantidade máxima de memória RAM que pode ser endereçada, considerando-se o tamanho do barramento de endereços.

Com base no cenário descrito e nos seus conhecimentos, avalie as afirmativas a seguir.

I – O número máximo de *bits* que podem ser armazenados na memória RAM é de 213, supondo que todo o endereçamento seja utilizado na memória RAM (na prática, parte do endereçamento deve ser utilizado também para a memória ROM, mas devemos desconsiderar essa situação, de acordo com a simplificação proposta no enunciado).

II – Um programa que seja executado nessa máquina não pode utilizar mais de 18 bits de memória em dado instante de tempo.

III – Uma variável do programa pode ocupar no máximo 210 bits, que é igual ao tamanho do barramento de dados.

É correto o que se afirma apenas em:

A) I.

B) II.

C) III.

D) I e II.

E) I e III.

Resposta correta: alternativa A.

Análise das afirmativas

I – Afirmativa correta.

Justificativa: no cenário do enunciado, o valor máximo de *bits* é dado pelo total de posições de memória endereçáveis (no caso 210) multiplicado pelo número de *bits* em cada posição, 8 ou 23, o que resulta em 210 x 23 = 213 bits. Aqui, supõe-se que todo o endereçamento vai ser utilizado para posições da memória RAM.

II – Afirmativa incorreta.

Justificativa: um programa poderia utilizar, no máximo, o número total de *bits* armazenados na memória, ou 213 bits, como calculado na afirmativa I, e não apenas 18 bits.

III – Afirmativa incorreta.

Justificativa: uma variável do programa pode ocupar múltiplas posições de memória, ou seja, múltiplos de 8 bits. O valor máximo depende dos tipos de dados, mas esse valor não será necessariamente igual ao número de posições (endereços) de memória disponíveis (que é igual a 210). Na prática, esse valor (que vale para uma única variável) deve ser bem menor do que o máximo de *bits* que podem ser armazenados na memória RAM.

Questão 2. Uma empresa de tamanho médio do setor de varejo está tendo problemas com o gerenciamento das informações produzidas pelos seus funcionários. No formato atual, os funcionários gravam o resultado do seu trabalho nas suas próprias máquinas locais, o que ocasiona uma série de problemas, como a perda de informações e até mesmo o retrabalho quando outro funcionário não sabe da existência de um documento. Uma falha em uma máquina pode significar a perda de anos de trabalho, e essas máquinas estão frequentemente expostas a ambientes ruins (algumas ficam em um armazém) ou a vírus, além de não terem *backups* frequentes. Para resolver essa situação, a empresa quer centralizar o armazenamento dos documentos produzidos pelos seus funcionários, como arquivos de texto, planilhas e apresentações. Esses documentos devem ser gravados em um servidor de arquivos, que terá os *backups* frequentes e será administrado de forma adequada. Todas as máquinas serão ligadas em uma rede local de boa velocidade.

O servidor terá grande quantidade de armazenamento, já que todos os funcionários da empresa deverão gravar o resultado do seu trabalho de forma permanente nessa máquina. É fundamental que os arquivos sejam preservados de maneira segura. Além disso, o sistema deve ser tolerante a falhas de *hardware*, uma vez que a indisponibilidade do servidor vai impactar no funcionamento de toda a empresa. Adicionalmente, a *performance* de leitura e de gravação dos arquivos será outro fator relevante, especialmente porque afetará o ritmo de trabalho dos funcionários. Buscas por arquivos serão frequentes, assim como seu compartilhamento entre diferentes funcionários e departamentos.

Com base no cenário apresentado e nos seus conhecimentos, avalie as afirmativas a seguir.

I – Uma boa alternativa para garantir a confiabilidade, a segurança e a tolerância às falhas é que o armazenamento dos dados (no servidor) seja feito em vários discos em RAID 0.

II – Tanto com relação à segurança quanto em relação à tolerância a falhas, não há diferenças em utilizar um sistema de armazenamento com RAID 0 ou um sistema de armazenamento RAID 1.

III – Caso seja utilizado um sistema de armazenamento com RAID 4, será necessário usar um disco específico para o armazenamento da paridade.

É correto o que se afirma apenas em:

A) I.

B) II.

C) III.

D) I e II.

E) I e III.

Resposta correta: alternativa C.

Análise das afirmativas

I – Afirmativa incorreta.

Justificativa: um dos grandes problemas com relação ao RAID 0 é que não existe redundância na gravação dos dados, o que torna o sistema menos tolerante às falhas.

II – Afirmativa incorreta.

Justificativa: no caso do RAID 1, existe redundância na gravação dos dados: isso significa que mesmo que um disco venha a falhar, existe uma cópia das informações em outro disco. Assim, há aumento da confiabilidade em relação ao RAID 0, que não apresenta redundância alguma.

III – Afirmativa correta.

Justificativa: no caso do RAID 4, além de haver um cálculo de paridade para cada bloco, existe um disco dedicado especificamente a armazenar essa informação.