

Unidade 9: Memórias

Prof. Daniel Caetano

Objetivo: Compreender os tipos de memória e como elas são acionadas nos sistemas computacionais modernos.

INTRODUÇÃO

Nas aulas anteriores foram apresentadas diversas maneiras de interpretar os bits na memória; essa compreensão é de extrema importância, mas não responde às perguntas: o que é, como funciona e como é acessada a memória?

O objetivo desta aula é apresentar uma introdução sobre os diferentes tipos de memórias existentes no computador, além de apresentar a forma com que a memória - e outros dispositivos - são acessados, através do **barramento de sistema**, que será detalhado em aulas posteriores.

1. O QUE É A MEMÓRIA?

Em palavras simples, a memória é um dispositivo físico capaz de armazenar e recuperar uma configuração elétrica em um "conjunto de fios". Uma vez que essa configuração elétrica estabelece um padrão de bits, ligados ou desligados, é possível dizer que a memória armazena e recupera **dados**.

Olhando como uma caixa preta, a memória é bastante simples. Observe a Figura 1.

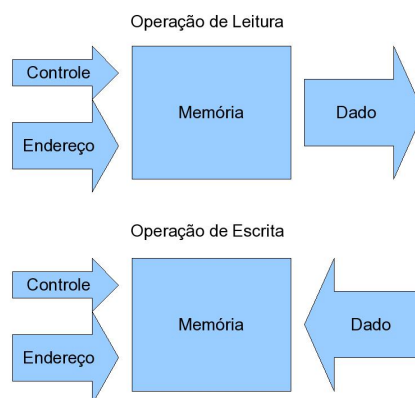


Figura 1: Operações de leitura e escrita na memória

Simplificadamente, o dispositivo memória recebe sinais de controle - que indicam se a operação é de leitura ou escrita na memória - e um endereço. Se a operação for de leitura, o dispositivo memória responde emitindo o dado armazenado no endereço em questão; se a

operação for de escrita, o dispositivo recebe o dado e o armazena na posição de memória indicada pelo endereço.

Apesar de ser um dispositivo de funcionamento aparentemente simples, as memórias são possivelmente os dispositivos com maior diversidade de implementações. Isso ocorre porque há diversas características que buscamos nas memórias como velocidade e capacidade, que não são atendidas plenamente por nenhum dos tipos de memória existente.

Há memórias que são rápidas, mas se forem desligadas perdem os dados armazenados e são muito caras; há memórias que são de velocidade média e possuem um preço razoável, mas se forem desligadas também perdem os dados. Há memórias muito baratas e que mantêm a informação quando são desligadas, mas que são muito lentas...

Além disso, nem todas as memórias fornecem dados do "tamanho" que o processador "quer". O tamanho dos dados lidos pelo processador é denominado **palavra** e pode ter diferentes tamanhos, como 8, 16, 32, 64, 128bits... dentre outros. Isso é o que determina, em geral, a expressão "processador de 64 bits": o tamanho do dado que ele manipula diretamente. Algumas memórias simplesmente fornecem os dados em blocos maiores do que uma palavra, exigindo algum "*malabarismo*" para permitir seu uso direto com um dado processador.

Adicionalmente, para que se possa tirar máximo proveito de um sistema computacional, a velocidade da memória deve ser compatível com a velocidade do processador, de maneira que esse último não precise ficar esperando por respostas da memória por muito tempo. Em tese, considerando os processadores atuais, isso exigiria que toda a memória fosse muito rápida e, como consequência, os equipamentos seriam muito caros e praticamente não poderiam ser desligados.

Certamente esse não era um caminho viável e, por essa razão, criou-se uma outra alternativa: usar diversos tipos de memória para obter o melhor desempenho ao menor custo.

2. HIERARQUIA DE MEMÓRIA

A quantidade de dados que um usuário médio armazena é gigantesca. Se considerarmos um servidor de uma grande empresa, essa quantidade de dados é ainda maior. Entretanto, a grande maioria desses dados **raramente** é usada pelo computador. Isso ocorre porque apenas um pequeno conjunto de programas e dados é usada rotineiramente.

Ainda assim, mesmo considerando os programas e dados que são processados com frequência, se medirmos a quantidade de tempo que o processador gasta com cada um destes bytes, veremos que a maior parte do tempo o computador está executando pequenos blocos de instruções e dados, realizando **tarefas repetidas**.

Observando este comportamento, os projetistas de *hardware* concluíram que poderiam equilibrar o custo de um equipamento se usassem um tipo de diferente de memória para cada tarefa:

a) Registradores e Memória Cache (Armazenamento Interno): Para armazenamento de curto prazo, de dados usados intensivamente pelo computador, adotam-se dispositivos de armazenamento volátil extremamente rápidos, mas de pequena capacidade devido ao **custo por bit ser muito alto**.

b) Memória Principal (Armazenamento Interno): Para armazenamento de médio prazo, de dados medianamente usados, adotam-se dispositivos de armazenamento volátil, cujo **custo por bit é médio**, proporcionando média capacidade com uma velocidade de acesso também média, já que estes dados são usados com alguma frequência.

c) Memória Secundária (Armazenamento Externo): Para armazenamento de longo prazo, de dados pouco usados, adotam-se dispositivos de armazenamento não volátil e cujo **custo por bit é baixo**, proporcionando grande capacidade, ainda que sejam lentos. A lentidão não é um problema, pois os dados aí contidos são pouco acessados.

d) Memória de Segurança (Armazenamento de Segurança): Para armazenamento de longuíssimo prazo, de dados que talvez nunca sejam necessários, adotam-se dispositivos de armazenamento não volátil de **custo por bit extremamente baixo**, com enorme capacidade, ainda que extremamente lentos.

Estes quatro níveis formam a hierarquia de memória, lembrando que todos os dados úteis de um computador precisam estar armazenados na memória secundária, sendo transferidos para a memória principal na medida em que são necessários. Da mesma forma, os dados da memória principal são transferidos para o cache e para os registradores também na medida em que são necessários. A comunicação ocorre obedecendo a hierarquia, como pode ser visto na Figura 2.

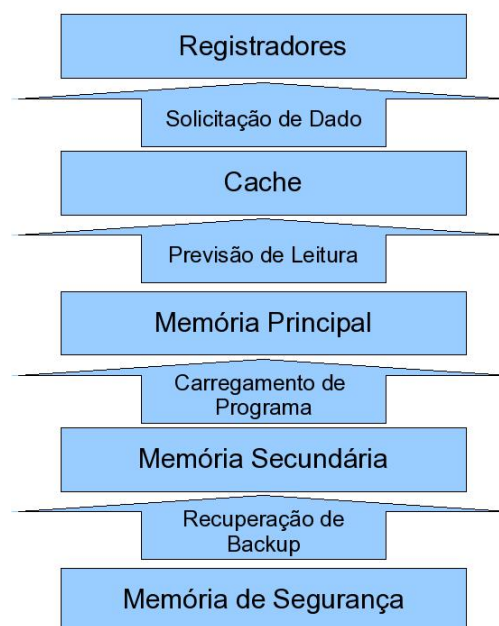


Figura 2: Situações de transferência de dados entre memórias (leitura)

3. TIPOS DE MEMÓRIA

As memórias utilizadas em cada uma destas camadas da hierarquia podem ser construídas com diferentes tecnologias. A diferenciação mais básica está entre os tipos RAM e ROM, mas existem diversos subtipos. Algumas delas estão descritas no quadro a seguir.

Tipo	Categoria	Apagamento	Escrita	Volatilidade	Palavra / Bloco	Interna / Externa	Velocidade	Usos	Custo por Bit
Memória de Acesso Aleatório Estática (SRAM)	Escrita e Leitura	Eletricamente	Eletricamente	Volátil	Bytes	Interna	Variada (pode ser tão rápida quanto o processador)	Registradores, cache, memória principal	De muito alto a alto
RAM Dinâmica (DRAM)	Escrita e Leitura	Eletricamente	Eletricamente	Volátil	Bytes	Interna	Média	Memória principal	Médio
Memória apenas de Leitura (ROM)	Apenas de leitura	Impossível	Máscaras	Não volátil	Bytes	Interna / Externa	Média para Rápida	Memória principal ou secundária	Baixo
ROM Programável (PROM)	Apenas de leitura	Impossível	Eletricamente	Não volátil	Bytes	Interna / Externa	Média para Rápida	Memória principal ou secundária	Baixo
PROM Apagável (EPROM)	Principalmente de leitura	Luz UV	Eletricamente	Não volátil	Bytes	Interna / Externa	Rápida para leitura	Memória principal ou secundária	Baixo
EPROM Eletricamente apagável (EEPROM)	Principalmente de leitura	Eletricamente	Eletricamente	Não volátil	Bytes ou Blocos	Interna / Externa	Rápida para leitura	Memória principal ou secundária	Médio
Memória Flash	Principalmente de leitura	Eletricamente	Eletricamente	Não volátil	Blocos	Externa	Média para Lenta	Memória secundária	Médio
Disco Magnético	Escrita e Leitura	Magneticamente	Magneticamente	Não volátil	Blocos	Externa	Lenta	Memória secundária	Baixo
Disco Óptico	Leitura (e, opcionalmente, Escrita)	Óptica	Óptica	Não volátil	Blocos	Externa	Muito lenta	Memória secundária e de segurança	Muito baixo
Fita Magnética	Escrita e Leitura	Magneticamente	Magneticamente	Não volátil	Bytes	Externa (no passado, interna)	Extremamente Lenta	Memória secundária e de segurança (no passado, principal)	Extremamente baixo

Das memórias voláteis, qual a diferença entre a memória SRAM e DRAM? Bem, a SRAM é um dispositivo que basta estar ligado para preservar seus dados; a DRAM, por outro lado, exige que de tempos em tempos seja feita uma "simulação de leitura" em cada região da memória, para garantir que ela não seja perdida, em um processo chamado *refresh*.

Como a maioria das memórias permite apenas **um acesso** por vez, isto é, ela só permite que um endereço de memória seja acessado de cada vez, durante o momento em que o *refresh* está sendo executado a memória DRAM fica **indisponível** para o processamento e, por essa razão, ela acaba sendo, no geral, mais lenta que a SRAM.

Projetar circuitos com SRAM é muito mais simples do que com DRAM; entretanto, a diferença de preço entre ambas faz com que, em geral, se a velocidade de uma SRAM não é necessária, os projetistas adotem o uso de DRAMs.

4. ACESSO A MEMÓRIA

Desde o início do curso é comentado que as partes do computador se comunicam por "fios". De fato, é isso que ocorre, embora esses "fios" muitas vezes sejam trilhas minúsculas em uma placa de circuito impresso.

Por exemplo, no caso da memória, apresentamos na Figura 1 "setas" que indicavam o fluxo de informações para a memória. Na prática, essas "setas" são fios ou trilhas, pelos quais trafegam sinais elétricos: a presença de sinal é interpretada como "1" e a ausência de sinal é interpretada como "0", formando os padrões desejados. Veja o exemplo da Figura 3.



Figura 3: Nomenclatura dos "fios" que ligam a memória ao sistema

Nesta figura, os "fios" do controle foram nomeados de W (Write), R (Read) e S (Select). Estes fios controlam o funcionamento da memória. Sempre que a memória estiver sendo usada, S (ou MREQ) deverá estar com sinal em 1, indicando que a memória está selecionada. Quando S (ou MREQ) estiver em 1, os valores de W e R indicarão qual é a operação solicitada.

Quando a memória for usada para escrita, o sinal de W deve estar em 1 e R em 0; quando a memória for usada para leitura, R deverá estar em 1 e W em 0. O comportamento se S (MREQ), R e W estiverem todos simultaneamente em 1 é indefinido.

Os "fios" de endereço foram nomeados de A0 a A2 (A vem de Address). Isso significa que essa memória tem 3 bits de endereçamento, permitindo acesso a 8 dados (2^3). Em outras palavras, **temos 8 posições de memória.**

Os "fios" para representar os dados foram nomeados de D0 a D7 (D vem de Data). Isso significa que cada uma das posições de memória tem 8 bits (1 byte), proporcionando o armazenamento de 256 valores (2^8) diferentes em cada posição de memória.

Assim, como temos 8 posições de 1 byte cada, esta é uma memória de 8 bytes.

Vamos analisar agora como funcionaria a memória no momento de uma leitura. Num primeiro momento, o circuito do computador irá indicar nos "fios" do controle que ele deseja ler a memória e, nos "fios" de endereço, vai indicar que deseja ler um determinado endereço. Suponhamos que o endereço a ser lido seja o endereço 6, ou seja, 110b. Observe a configuração na Figura 4.



Figura 4: Configuração para a leitura do endereço de memória 6 (110b)

O circuito do computador coloca essa configuração elétrica nos "fios" de entrada da memória e, alguns instantes depois, a memória configura eletricamente os "fios" de dados (D0 a D7) com a informação que nela está armazenada, conforme indicado na Figura 5.

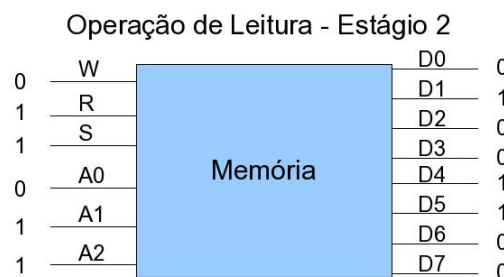


Figura 5: Resposta da memória à leitura do endereço de memória 6 (110b)

Ou seja: o dado armazenado na posição 6 da memória é: 00110010b, ou, interpretando como um decimal inteiro, 50.

Por terem funções **muito** distintas, cada um destes conjuntos de fios recebem nomes específicos. Os "fios" que controlam os dispositivos ligados ao computador são chamados de **barramento de controle**. Os "fios" que configuram endereços de memória e outros dispositivos são chamados de **barramento de endereços** e, finalmente, os "fios" que servem para a troca de dados entre os vários dispositivos são chamados de **barramento de dados**.

O exemplo desta aula foi feito com uma unidade de memória mas, de maneira geral, o procedimento é parecido para qualquer dispositivo que seja ligado no sistema computacional moderno.

Isso ocorre porque os computadores são projetados segundo um paradigma de arquitetura denominado "barramento de sistema", que será visto na aula que vem.

6. BIBLIOGRAFIA

MURDOCCA, M. J; HEURING, V.P. **Introdução à arquitetura de computadores**. S.I.: Ed. Campus, 2000.

STALLINGS, W. **Arquitetura e organização de computadores**. 5ed. São Paulo: Ed. Pearson Prentice Hall, 2003.