

DISPOSITIVOS DE MEMÓRIAS SEMICONDUTORAS

4.1	DISPOSITIVOS DE MEMÓRIAS SEMICONDUTORAS	51
4.1.1	Conceitos Importantes.....	52
4.1.2	Acesso.....	52
4.1.3	Volatilidade.....	52
4.1.4	Memórias de escrita/leitura ou somente leitura.....	52
4.1.5	Tipo de armazenamento.....	53
4.2	PRINCÍPIOS DE OPERAÇÃO DAS MEMÓRIAS	53
4.3	Entradas de Endereços	53
4.4	A Entrada R/W	54
4.4.1	Habilitação de Memória (CS).....	54
4.4.2	Conexões da Memória com a CPU.....	54
4.5	Memórias	55
4.6	Operação de leitura:	55
4.7	MEMÓRIA DE ACESSO ALEATÓRIO (RAM)	55
4.8	CIRCUITOS DE MEMÓRIA EM PARALELO	57
4.9	ORGANIZAÇÃO INTERNA DE MEMÓRIAS	59
4.10	RAM DINÂMICA (DRAM)	61
4.10.1	Estrutura e Operação da RAM Dinâmica.....	61
4.11	MEMÓRIA SOMENTE DE LEITURA (ROM)	61
4.11.1	IMPLEMENTAÇÃO DE UM CODIFICADOR DE ROM.....	63
4.11.2	HIERARQUIA E TIPOS DE MEMÓRIAS ROM.....	64
4.11.3	Memórias MROM (MASK ROM - ROM de MÁSCARA).....	64
4.11.4	Memórias PROM (<i>Programmable Read Only Memory</i> - Memória Programável Só de Leitura).....	65
4.11.5	Memória EPROM (<i>Erasable Programmable ROM</i> - ROM Apagável e Programável).....	66
4.11.6	Memória EEPROM (<i>Electrically Erasable Programmable ROM</i> - ROM Apagável e Programável Eletricamente).....	67
4.12	Memórias FLASH	68
4.13	TEMPOS DE CHAVEAMENTO DAS MEMÓRIAS	69
4.13.1	Ciclo de Leitura.....	69
4.13.2	Ciclo de Escrita.....	70
4.14	APLICAÇÕES DE MEMÓRIAS ROMS	70
4.14.1	<i>Firmware</i>	70
4.14.2	Memória de <i>Bootstrap</i>	71
4.14.3	Tabela de Dados.....	71
4.14.4	Conversores de Dados.....	71
4.14.5	Gerador de Funções.....	71

4.1 DISPOSITIVOS DE MEMÓRIAS SEMICONDUTORAS

As memórias são dispositivos capazes de armazenar um grande volume de dados importantes para o funcionamento dos computadores, e fundamentais no armazenamento confiável de informações. Quando aborda-se o tema “Memórias”, deve-se perceber claramente três restrições, que norteiam os projetistas de memórias para computador:

Qual o tamanho de armazenamento?

Qual a velocidade de acesso e os tempos de leitura/gravação do dispositivo?

Quanto custa?

A complexidade das aplicações utilizando microprocessadores estão intimamente ligadas ao tamanho das memórias; hoje se a capacidade de memória aumenta, com total certeza a aplicações em software se tornam mais complexas e incorporam o novo aumento. Quanto a velocidade, devemos lembrar que os microprocessadores estão constantemente trocando informações com os dispositivos de memória do sistema, e o fundamental para a boa operação do sistema global é minimizar o tempo de espera entre os dados trocados pelos microprocessadores e as memórias, portanto a velocidade e o desempenho das memórias devem acompanhar o ritmo de desenvolvimento dos microprocessadores para não se tornarem o gargalo no sistema. A questão final a ser considerada é o custo, pois o preço de produção envolvendo as memórias deve manter uma relação razoável com os outros componentes do sistema.

A Tabela 1 traz algumas características envolvendo os sistemas de memórias.

Tabela 1: características envolvendo os sistemas de memórias

Localização	Método de acesso	Tipos físicos
Processador	Seqüencial	Semicondutoras
Interna (principal)	Direto	Magnéticas
Externa (secundária)	Aleatório	Ópticas
Capacidades	Associativo	Magneto-ópticas
Tamanho da palavra	Performance	Características físicas
Número de palavras	Tempo de acesso	Volátil/não volátil
Unidade de transferência	Tempo de ciclo	Apagável/não apagável
Palavras	Taxa de transferência	
Blocos		

Como poderíamos esperar, há uma forte relação entre as três características principais sobre memórias, que definiram a evolução dos dispositivos; em outras palavras, capacidade, tempo de acesso, e custo estão intimamente ligados. As seguintes relações se mantêm até hoje:

- Maior velocidade de acesso acarreta maiores custos por bit
- Maior capacidade torna os custos por bit mais baratos
- Maior capacidade conseqüentemente aumenta o tempo de acesso

Os dilemas que norteiam os projetistas de memórias são claros, e nos próximos tópicos iremos discutir a hierarquia, o funcionamento, e os típicos desenvolvimentos históricos sobre o tema – Memórias Semicondutoras.

4.1.1 Conceitos Importantes

Célula de memória - dispositivo ou circuito capaz de armazenar um bit.

Por exemplo, um flip-flop, um capacitor, etc.

Palavra de memória – unidade natural de organização da memória. O tamanho da palavra é tipicamente igual ao número de bits usado para representar um número ou uma instrução.

Unidades de endereçamento – em muitos sistemas, a unidade de endereçamento é do tamanho da palavra de memória. Mas alguns sistemas fazem o endereçamento ao nível de bytes. Sendo em alguns, a relação entre o comprimento nos bits A de endereçamento e o número N de endereços é $2^A = N$.

Unidade de transferência – este é o número de bits lidos da memória, ou escritos na memória ao mesmo tempo. A unidade de transferência não precisa necessariamente ser igual ao comprimento da palavra, ou de uma unidade de endereço, pois muitas vezes os dados são transferidos em blocos.

4.1.2 Acesso

As memórias armazenam informações em lugares que se denominam “localidades de memória”. Cada localidade de memória guarda um conjunto de bits e tem um endereço. No acesso desses endereços podemos analisar:

O tempo de acesso - é o tempo que a memória necessita para que sejam escritos ou lidos os dados em suas localidades;

Tempo de ciclo da memória – este conceito é principalmente aplicado as memórias de acesso aleatório, e representa o intervalo mínimo entre dois acessos sucessivos à memória.

Taxa de transferência – representa a taxa, com que os dados são transferidos para dentro (gravação) ou fora (leitura) das unidades de memórias.

Acesso seqüencial - nas memórias que têm acesso seqüencial, para acessar um endereço de uma certa localidade, precisa-se passar por endereços intermediários (as memórias mais comuns desse tipo são as que utilizam fita magnética);

Acesso aleatório - as memórias que utilizam esse tipo de acesso são as que permitem que seja acessado qualquer dado em qualquer endereço sem a necessidade de ter que passar por outros endereços intermediários (ex.: memórias semicondutoras).

4.1.3 Volatilidade

Memórias voláteis - são aquelas que perdem as informações quando é cortada sua alimentação. São memórias que geralmente usam como elemento de memória o flip-flop.

Memórias não voláteis - são memórias que mesmo desligando-se sua alimentação, não perdem as informações armazenadas. Dentre essas se destacam as magnéticas e as eletrônicas ROM, PROM, EPROM, EEPROM, e outras.

4.1.4 Memórias de escrita/leitura ou somente leitura

Escrita/leitura - são memórias que podem ser acessadas pela CPU tanto para leitura quanto para escrita; elas são usadas para armazenar dados que serão utilizados durante a execução do programa (memórias RAM's, EEPROM's).

Somente leitura - são as memórias que armazenam o programa, ou seja são as memórias que só serão lidas pela CPU e que já vêm gravadas para o sistema (memórias ROM's, PROM's, etc).

4.1.5 Tipo de armazenamento

Estáticas - memórias estáticas são aquelas nas quais as informações permanecem armazenadas enquanto não houver escrita ou não faltar energia.

Dinâmicas - memórias dinâmicas são memórias que perdem informações armazenadas mesmo com alimentação. Na RAM dinâmica (ou DRAM) isso acontece porque cada célula tem um transistor MOSFET e um capacitor que armazena um dado (1bit).

A Tabela 2 lista os principais tipos de memórias semicondutoras. O termo RAM (*random-access-memory*) pode ser referido a todos os tipos de memórias vistos nesta tabela, pois todas são de acesso aleatório. Sendo uma diferença marcante das memórias RAM, a possibilidade de se escrever e ler dados da memória com muita facilidade e velocidade. A leitura e escrita são efetuadas através de sinais elétricos compatíveis com os encontrados no microprocessador. Outra característica marcante da RAM, se refere a volatilidade de seus dados, pois se a alimentação é interrompida, os dados se perdem.

Tabela 2: Tipos de Memórias Semicondutoras

Memory Type	Category	Erasure	Write Mechanism	Volatility
Random-access-memory (RAM)	Read-write memory	Electrically, byte level	Electrically	Volatile
Read-only-memory (ROM)	Read-only memory	Not possible	Masks	Nonvolatile
Programmable (PROM)	Read-only memory	Not possible	Electrically	Nonvolatile
Erasable PROM (EPROM)	Read-mostly memory	UV light, chip level	Electrically	Nonvolatile
Flash memory	Read-mostly memory	Electrically, block level	Electrically	Nonvolatile
Electrically Erasable PROM (EEPROM)	Read-mostly memory	Electrically, byte level	Electrically	Nonvolatile

4.2 PRINCÍPIOS DE OPERAÇÃO DAS MEMÓRIAS

Independente dos tipos de memórias existentes, o princípio básico de operação é o mesmo:

- Selecionar o endereço a ser acessado (leitura ou escrita);
- Selecionar o tipo de operação: leitura ou escrita;
- Se a operação for escrita, fornecer os dados de entrada;
- Se a operação for leitura, os dados estarão disponíveis na saída;
- Habilitar a memória para que a operação seja concluída e desabilitar a memória para que ela não responda às entradas de endereço e ao comando de leitura/escrita.

4.3 Entradas de Endereços

A figura 1 apresenta o diagrama de uma memória 32X4.

Usando a memória mostrada na figura 1, podemos notar que ela possui 5 entradas de endereço (A0 a A4). Logo existem 32 posições (2^N , onde N é o nº de bits) de armazenamento que podem ser ocupadas por palavras de 4 bits (D0 a D3).

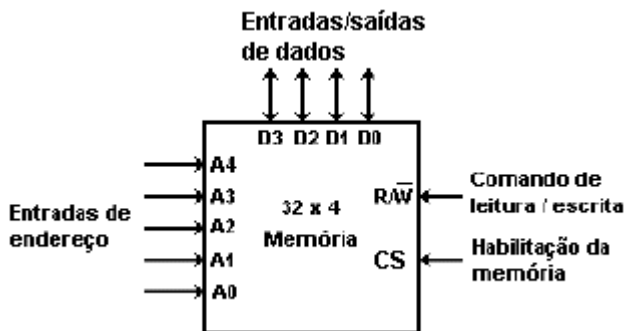


Figura 1. Diagrama de uma memória de 32 x 4

4.4 A Entrada R/W

A entrada R/W controla a operação que deve ser realizada na memória: leitura (R – Read) ou escrita (W – Write). Quando esta entrada estiver em “1” ocorre a operação de leitura, quando em “0” ocorre a operação de escrita. Algumas memórias usam os símbolos W (*write* - escrita) ou WE (*write enable* - habilitação de escrita) para identificar esta entrada.

4.4.1 Habilitação de Memória (CS)

A todo o momento a memória possui níveis lógicos em suas entradas e o pino de habilitação de memória impede que ela responda a estes níveis lógicos, evitando assim um acesso errôneo nesta memória. Esta entrada pode ser identificada como CE (chip enable) ou CS (chip select).

Exemplo: Uma determinada memória tem uma capacidade de 4K x 8.

Quantas linhas de entrada/saída de dados ela tem?

R: Oito, pois o tamanho da palavra é oito bits.

Quantas linhas de endereço ela tem?

R: A memória armazena $4K = 4 \times 1024 = 4096$ palavras. Tendo em vista que $4096 = 2^{12}$, logo a memória tem 12 bits (linhas) de endereço. Qual é a sua capacidade em bytes?

R: Um byte tem oito bits, então esta memória tem uma capacidade de 4096 bytes.

4.4.2 Conexões da Memória com a CPU

As memórias RAM e ROM são interfaceadas com a CPU através de três grupos de linhas de sinais ou barramentos: barramentos de endereço, barramento de dados e barramento de controle.

A figura 2 traz as conexões entre a CPU e as memórias.

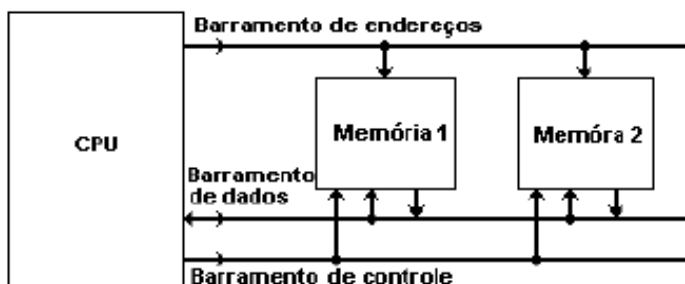


Figura 2. Conexões entre a CPU e as memórias.

- Barramento de endereços – barramento unidirecional que leva o endereço proveniente da CPU para a memória para selecionar uma posição de memória.
- Barramento de dados – barramento bidirecional que transfere dados entre a CPU e a memória.
- Barramento de controle – barramento que leva sinais de controle da CPU para a memória.

4.5 Memórias

Exemplo: Operação de escrita:

- A CPU coloca o endereço binário da posição de memória onde o dado deve ser armazenado no barramento de endereços.
- A CPU coloca o dado a ser armazenado no barramento de dados.
- A CPU ativa as linhas de controle para a operação de escrita.
- A memória decodifica o endereço e localiza a posição onde o dado deve ser armazenado.
- O barramento de dados é transferido para a posição de memória.

4.6 Operação de leitura:

- A CPU coloca o endereço binário da posição a ser lida da memória no barramento de endereços.
- A CPU ativa as linhas de controle para operação de leitura.
- A memória decodifica o endereço e localiza a posição a ser lida.
- A memória transfere o dado armazenado nesta posição para o barramento de dados, onde a CPU vai buscá-los.

4.7 MEMÓRIA DE ACESSO ALEATÓRIO (RAM)

A menor unidade de memória é chamada uma célula e pode ser usada para armazenar um bit de informação, isto é, 0 lógico ou 1 lógico. Um número determinado de células juntas forma uma palavra e, as células de uma palavra são lidas ou escritas ao mesmo tempo. Para formar uma célula de memória podemos utilizar um *latch*, um *flip-flop*, ou capacitores. Na figura 3 está apresentada uma memória, onde cada palavra é composta por duas células. Então, a memória tem 4-palavras e cada palavra tem 2-bits. Assim, a capacidade da memória, é igual a 8-bits, e a organização da memória é 4x2 (palavrasxbits).

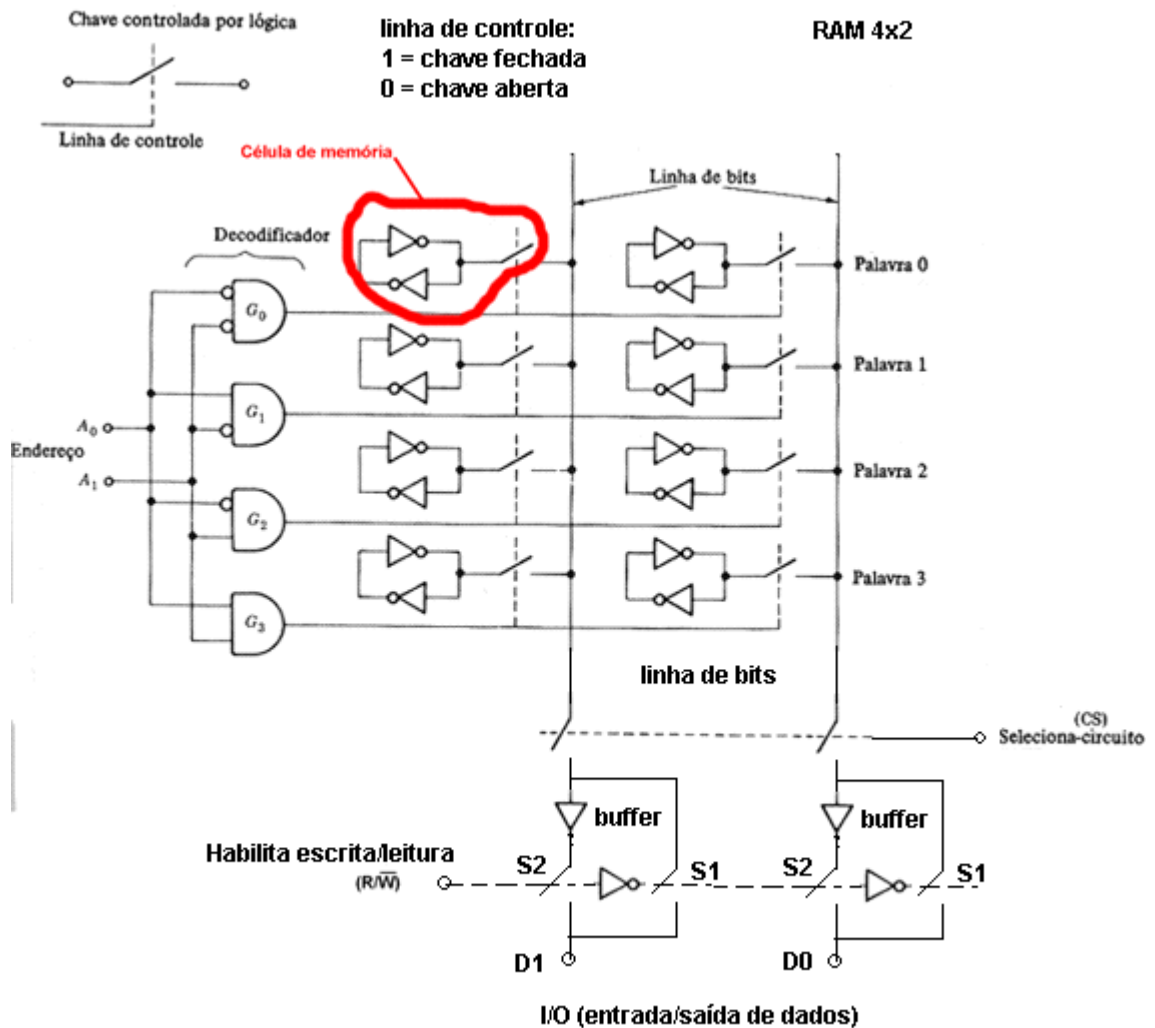


Figura 3. Memória de 8 bits (4x2), onde cada palavra é composta por duas células

Cada palavra da memória tem uma locação associada a um endereço definido pelos bits de endereço A1A0.

Endereço A1A0	Posição	Dados
00	Palavra	0
01	Palavra	1
10	Palavra	2
11	Palavra	3

Os endereços são decodificados pelo decodificador de endereços formado pelas portas AND G₀, G₁, G₂ e G₃, cujas saídas quando ativas conectam as saídas das células às linhas de bits, preparando a posição para uma operação de leitura ou escrita. As memórias sempre incorporam o decodificador de endereços para limitar o número de pinos da memória.

A memória é habilitada pelo sinal Seleciona-Circuito (CS – *chip select*). Quando CS está inativo, nível baixo, então as chaves operadas por lógica, que ligam as linhas de bits às saídas/entradas da memória, estão abertas, colocando-as em *three-state*, desconectando-as do barramento.

As operações de leitura e escrita são controladas pelo sinal Habilita Escrita/Leitura. Quando está no nível baixo as chaves S1 operadas por lógica dos I/O (D1 e D0) estão fechadas, e as S2 estão abertas, conectando os dispositivos externos às linhas de bits, e

deixando a memória em posição de escrita. No nível alto, estão fechadas as chaves S1 e abertas as S2, colocando as linhas de bits em ligação com os dispositivos externos em operação de leitura. As operações de leitura/escrita só podem ser realizadas com a memória habilitada, o pino Selecciona-Circuito no nível ativo.

4.8 CIRCUITOS DE MEMÓRIA EM PARALELO

Freqüentemente, ocorre que o número de palavras disponível em um CI não é adequado, ou o número de bits por palavras não é adequado, ou ambos. A solução é ligar memórias em paralelo.

A ligação de CI's de memória em paralelo para aumentar o número de bits por palavra (mas não para aumentar o número de palavras) é ilustrada na figura 6. Onde ligamos em paralelo dois CI's de oito palavras e quatro bits/palavra para construir uma memória em que o número de palavras permanece em oito, mas o número de bits aumenta de quatro para oito. O endereço de entrada de três bits é aplicado aos pinos de endereçamento de ambas as memórias. Os terminais CS dos dois CI's são interligados, bem como os terminais *Read/Write*. O CI1 aceita e armazena quatro bits (0, 1, 2 e 3), e o CI2 aceita e armazena mais quatro bits (4, 5, 6 e 7). O paralelismo pode ser estendido a circuitos adicionais para aumentar mais o número de bits por palavra. Três CI's de oito palavras e quatro bits/palavra produzem uma memória de oito palavras e doze bits/palavra, e assim por diante.

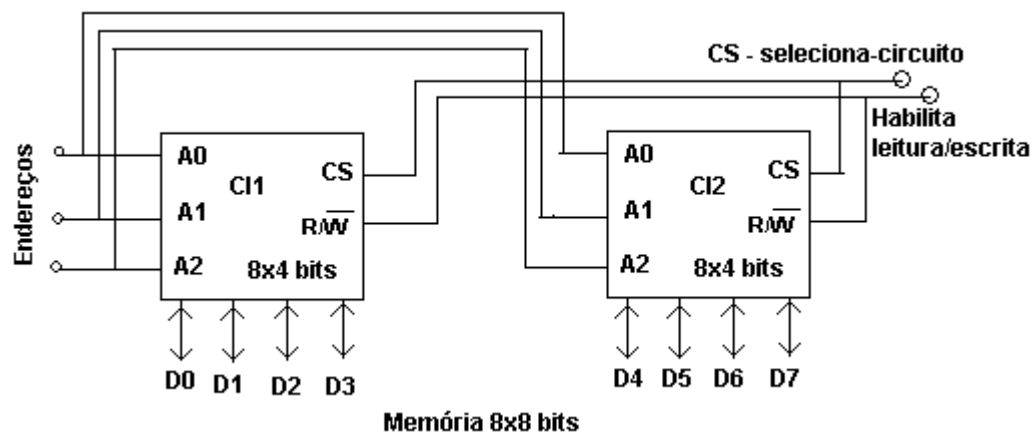


Figura 4. Dois CI's de oito palavras e quatro bits/palavra ligados em paralelo, para formar uma memória de oito palavras e oito bits/palavra.

Na figura 4 é mostrado como dois CI's de oito palavras e quatro bits/palavra podem ser ligados em paralelo para obter uma memória de dezesseis palavras e quatro bits/palavra. Como antes, os três bits de endereço são aplicados a ambos os circuitos, mas, em vez de termos um bit CS comum aplicado a ambos os circuitos, quando a entrada CS de um circuito é ativada a entrada CS do outro circuito é desativada. A entrada CS do sistema de memória pode ser considerada como um bit adicional de endereço, que chamamos de A4. Quando A4= 1, o CI endereçado é o CI2, e, quando A4= 0, o CI endereçado é o CI1. O bit de endereço de seleção de circuito A4 seleciona um ou o outro CI; os bits de endereço A0, A1 e A2 selecionam uma localização particular no circuito selecionado. Os barramentos de dados (entrada/saída) são ligados em comum aos dois circuitos, bem como à entrada Read/Write. Assim, um circuito que não for selecionado tem suas linhas de bits desligadas de seus pinos de dados.

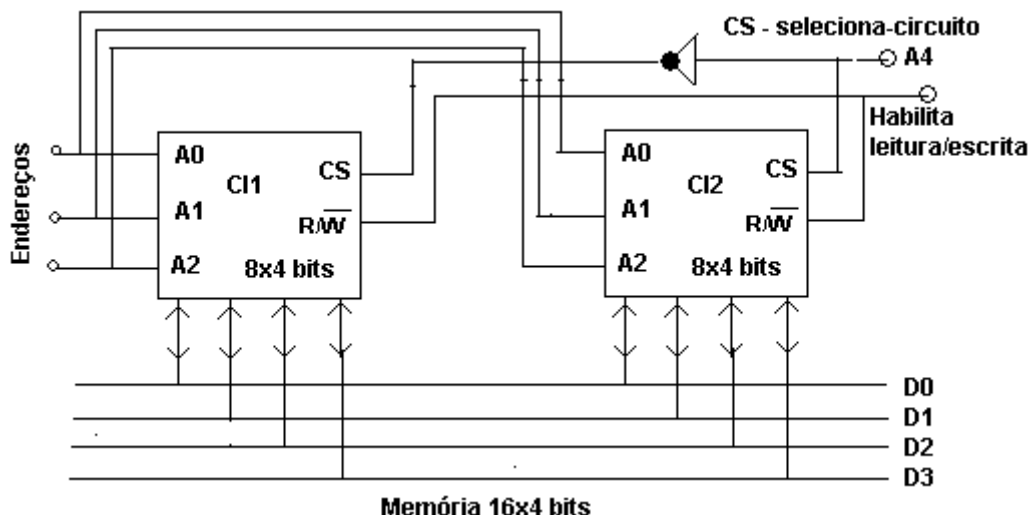


Figura 5. Ligação de circuito de memória em paralelo para aumentar o número de palavras mantendo fixo o número de bits por palavra.

O método de ligação em paralelo da figura 5 pode ser estendido a mais circuitos. Suponha que desejássemos usar aquele método para ligar quatro circuitos em paralelo formando uma memória de 32 palavras e quatro bits/palavra. Introduziríamos um quinto bit de endereço A5. Os três bits de endereço A0, A1 e A2 seriam, como anteriormente, aplicados em comum aos pinos de entrada de endereço dos CI's. Os bits de endereço A4 e A5 seriam aplicados a um decodificador, como mostrado na figura 6. As saídas do decodificador seriam usadas para ativar as entradas de seleção de circuito dos quatro CI's. Em muitos casos os fabricantes incluem na memória uma porta AND de entradas múltiplas e fazem com que a saída desta porta seja o nível lógico de entrada CS para o CI. Quando tais portas são incluídas, o decodificador externo de CS não é necessário e a decodificação de seleção de circuito pode ser feita no próprio CI. Em um caso típico, um CI de memória pode ter três entradas de seleção de circuito: CS1, CS2 e CS3, e o CI só será selecionado quando CS1=CS2=CS3= 1. Neste caso, até 8 CIs podem ser ligados em paralelo sem decodificação externa.

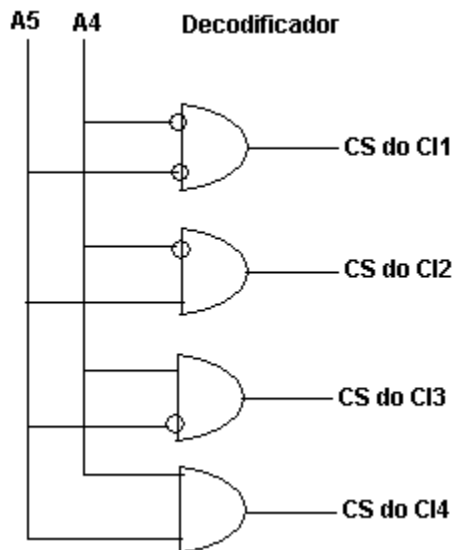


Figura 6. Os bits de endereço A4 e A5 aplicados a um decodificador.

4.9 ORGANIZAÇÃO INTERNA DE MEMÓRIAS

Para exemplificar como funcionam as memórias com estrutura unidimensional e bidimensional, iremos supor uma memória de dezesseis palavras e um bit/palavra, como mostra a figura 7.

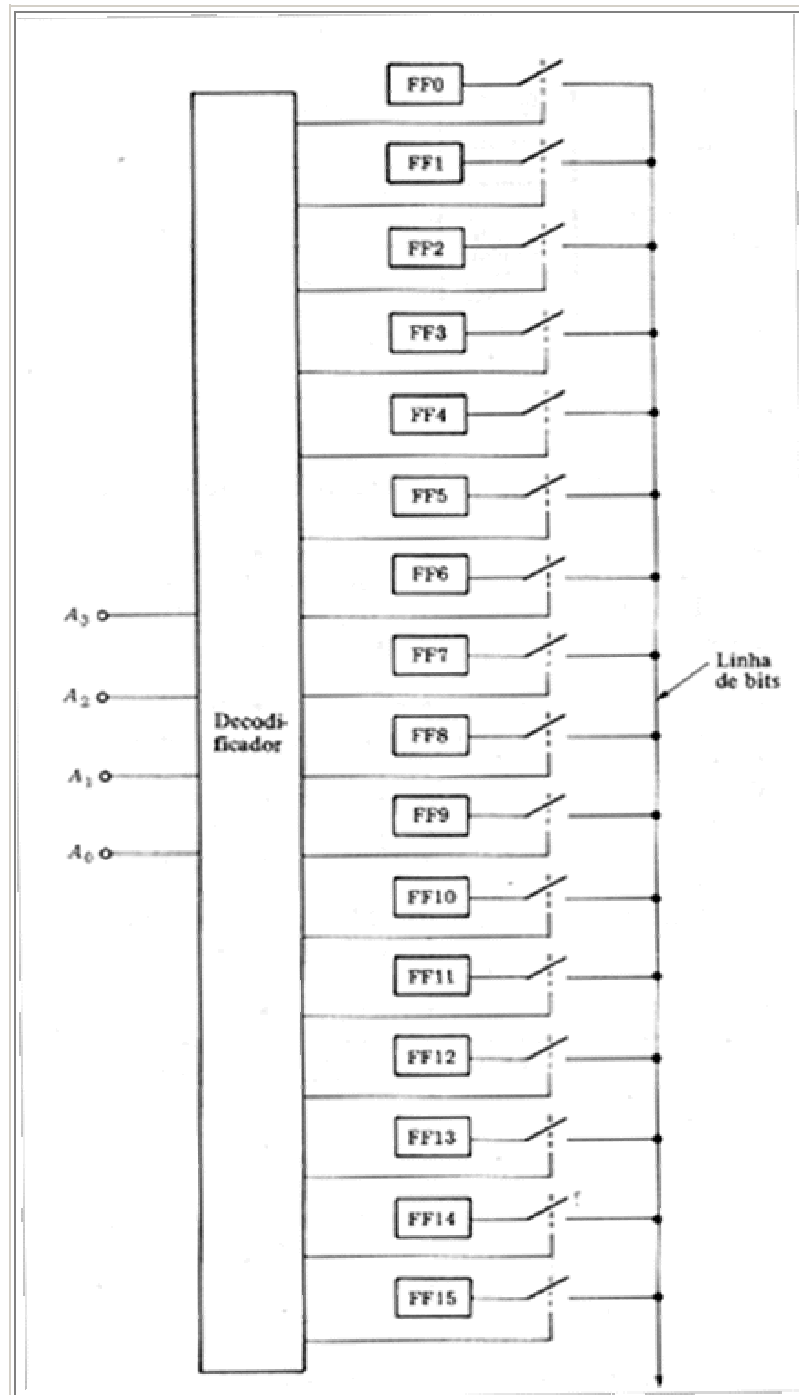


Figura 7. Estrutura de memória unidimensional ou linear

A memória de 16x1 tem um decodificador de endereços com 16 portas AND de 4 entradas, desde que existam 4-bits de endereços $A_3A_2A_1A_0$. À medida que a memória aumenta a capacidade, a quantidade de bits de endereços cresce, então, o decodificador deve aumentar de tamanho, acrescentando mais portas AND com maior número de entradas. Isto se torna um problema para memórias com muitos endereços.

A solução encontrada é organizar a memória internamente em linhas e colunas, com um decodificador de linhas e outro de colunas, como na figura 8.

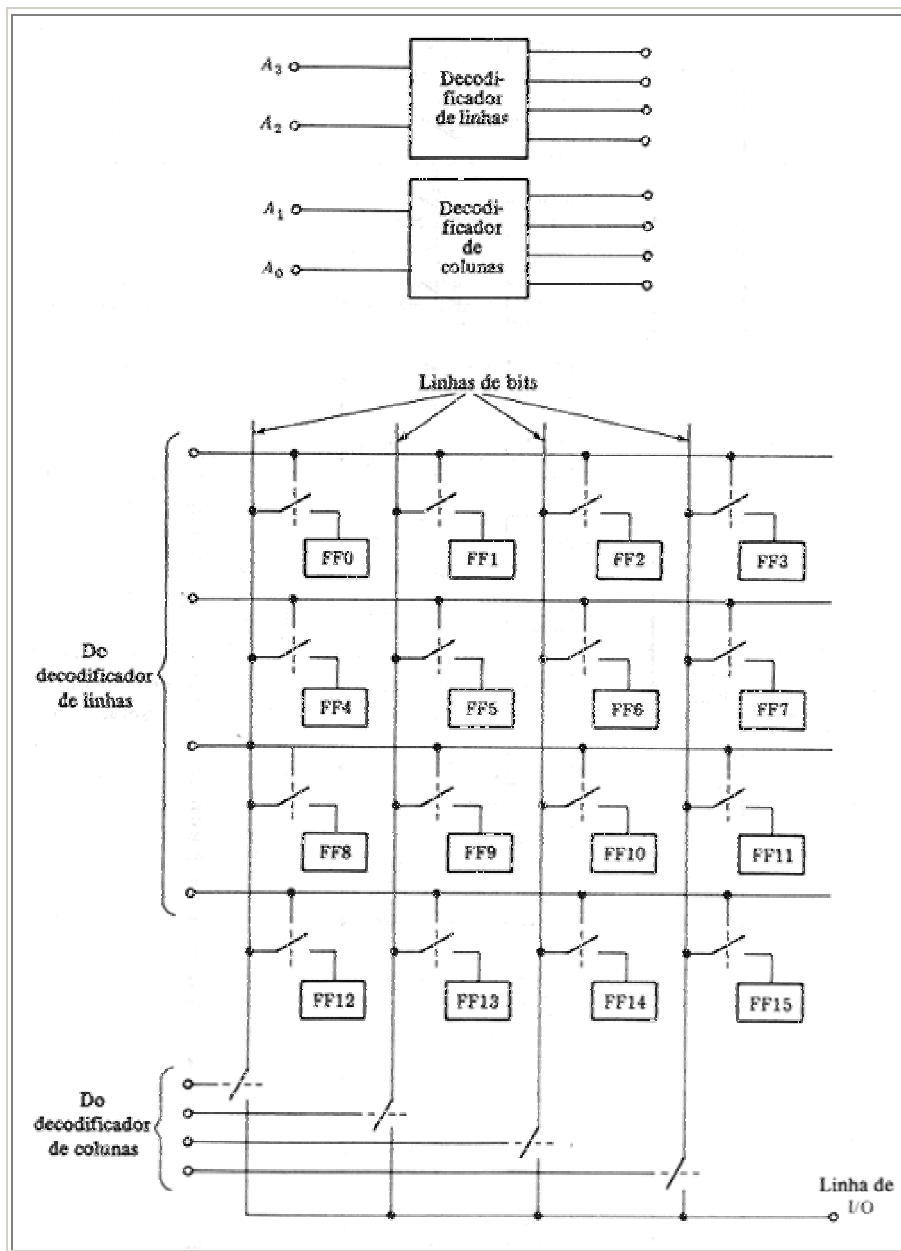


Figura 8. Estrutura de memória bidimensional.

Os decodificadores de linhas e de colunas são de 2-entradas e 4-saídas, cada um tem quatro portas AND de 2-entradas, reduzindo pela metade a quantidade de portas e de entradas nas portas.

O arranjo bidimensional usa quatro chaves a mais que o unidimensional, mas usa oito portas AND a menos. A vantagem torna-se aparente em memórias maiores. Consideremos, por exemplo, uma memória com 4096 (64^2) palavras. O arranjo unidimensional requer 4096 portas AND e 4096 chaves. O arranjo bidimensional requer apenas $64 + 64 (=128)$ portas AND e $4096 + 64 (=4160)$ chaves.

4.10 RAM DINÂMICA (DRAM)

As células de memória RAM vistas anteriormente se enquadram na classe RAM-Estática, pois utilizamos como elemento de armazenamento da informação flip-flop e latch, mas existem RAM's com capacitores como elemento de armazenamento, o que contribui para um aumento na densidade de informação armazenada.

Nesse tipo de memória a informação é armazenada em pequenos capacitores. Como existe uma tendência normal dos capacitores se descarregarem, há a necessidade de uma recarga (refresh – refrescamento) desses capacitores. De tempos em tempos (tipicamente de 2 a 8 ms) o bit armazenado no capacitor é atualizado. Esse tipo de memória constitui a memória principal de um computador por causa de sua alta capacidade e baixo consumo. Em relação à velocidade, as RAM estáticas são mais rápidas.

A principal vantagem deste tipo de memória é a simplicidade na sua construção, alcançando, assim, alto grau de integração.

4.10.1 Estrutura e Operação da RAM Dinâmica

Simbolicamente uma célula de memória dinâmica é constituída conforme o esquema da figura 10.

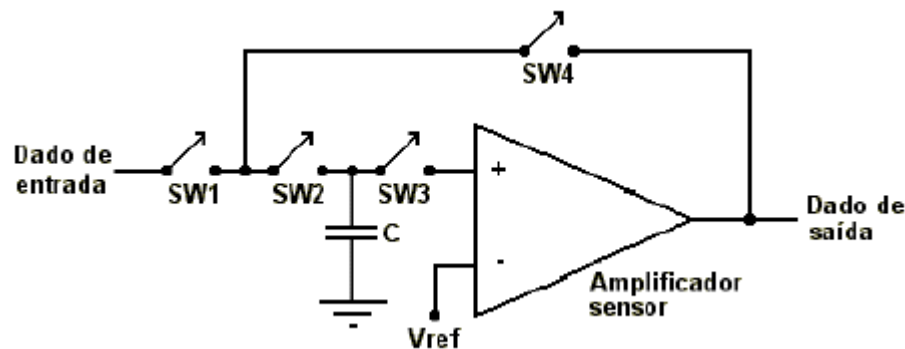


Figura 9. Esquema da célula de memória dinâmica.

Para escrever em uma célula, os sinais provenientes da decodificação de endereços e da lógica de leitura/escrita fecham as chaves SW1e SW2, mantendo SW3 e SW4 abertas. Isto conecta a entrada de dados a C. Um nível lógico 1 carrega C e um nível lógico 0 o descarrega. Depois disso todas as chaves são abertas, de modo que C fica desconectado do resto do circuito. Mesmo desconectado, existe uma corrente de fuga que causa a descarga gradual do capacitor e por isso o capacitor deve ser “refrescado”; o refresh é feito de tempos em tempos através das chaves SW2, SW3 e SW4, que transferem a saída do Amplificador Sensor (comparador) para o capacitor.

Para ler uma célula, as chaves SW2, SW3 e SW4 são fechadas, e SW1 permanece aberta. Isso conecta a tensão armazenada no capacitor ao amplificador sensor. Assim, na saída do amplificador teremos o valor armazenado no capacitor, que é atualizado via SW4 no momento da leitura.

4.11 MEMÓRIA SOMENTE DE LEITURA (ROM)

As Memórias Só de Leitura (*Read Only Memories - ROM*) são uma classe de memórias semicondutoras projetadas para aplicações onde a relação de operações de leitura para operações de escrita é muito alta (em termos de velocidade de acesso), ao contrário das memórias de leitura/escrita (*Read/Write Memories - RWM*) vistas anteriormente, onde a memória podia ser lida ou escrita com a mesma facilidade. O uso corrente emprega os

termos RAM para designar as memórias RWM e ROM para qualquer memória que pode ser apenas lida.

Em uma memória tipo ROM os bits não são armazenados em *latches* ou *flip-flops* como nas memórias RWM. Os bits em uma ROM são fixos e inalterados, programados na fabricação da ROM, ou em laboratório, e as células são circuitos combinacionais. Alguns tipos de ROM permitem programação do seu conteúdo sem retirá-lo do circuito digital.

Uma ROM nada mais é que um conversor de código e, como os conversores de código, consiste em um decodificador e um codificador. Como exemplo, consideremos a organização de uma memória com oito palavras e quatro bits/palavra.

Tabela 1: tabela verdade.

Posição de Memória	Endereço			Palavra de Dados			
	A ₂	A ₁	A ₀	D ₃	D ₂	D ₁	D ₀
m ₀	0	0	0	0	1	0	0
m ₁	0	0	1	0	1	1	1
m ₂	0	1	0	1	0	1	0
m ₃	0	1	1	1	1	0	1
m ₄	1	0	0	0	0	1	0
m ₅	1	0	1	1	0	1	1
m ₆	1	1	0	0	1	1	1
m ₇	1	1	1	0	1	0	0

A Tabela Verdade para memória indica que a memória é um conversor de código de entrada (endereços A₂A₁A₀) para outro código de saída (dados D₃D₂D₁D₀), isto é, um decodificador em cascata com um codificador.

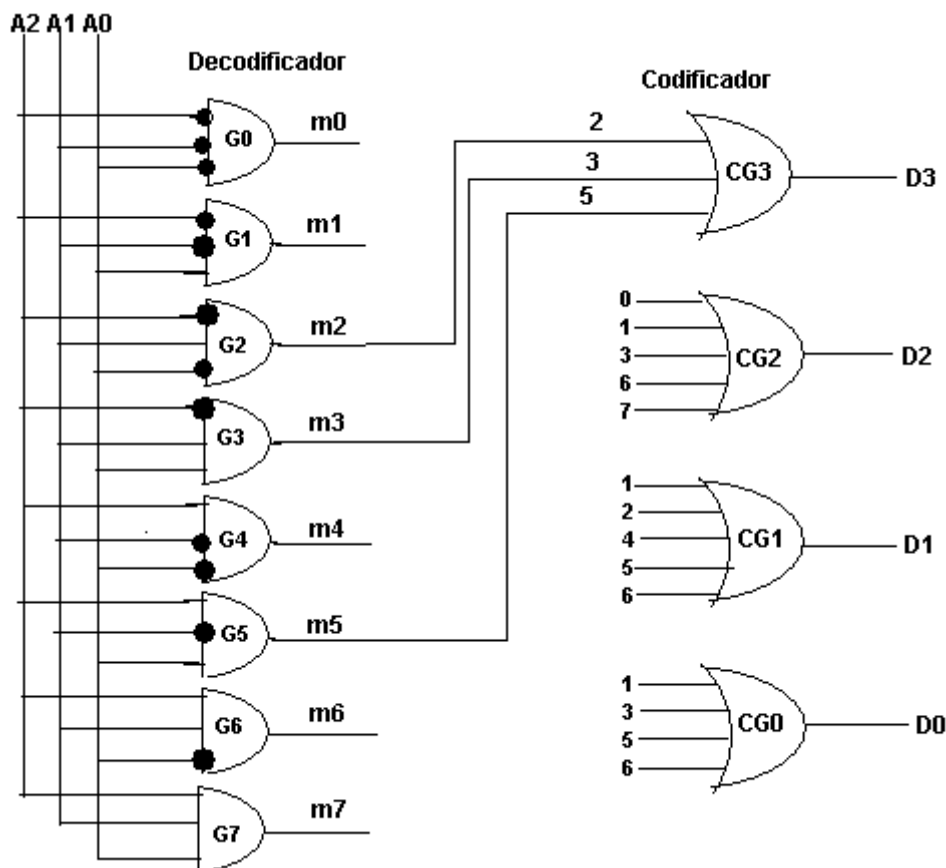


Figura 10. Esquema do circuito de uma ROM.

Para verificar se as ligações estão corretas, examinamos a tabela verdade (tabela 1). Notamos que $D_3 = 1$ somente quando $m_2 = 1$, ou $m_3 = 1$, ou $m_5 = 1$. Ligando a porta CG3 a m_2 , m_3 e m_5 , teremos $D_3 = 1$ somente quando necessário. As demais ligações podem ser verificadas de modo similar. Consideremos, por exemplo, que o endereço de entrada seja $A_2A_1A_0 = 011$. Devemos ter $D_3D_2D_1D_0 = 1101$. Observamos que, com esse endereço de entrada, só $m_3 = 1$ e m_3 é ligado as portas CG3, CG2 e CG0, como desejado.

4.11.1 IMPLEMENTAÇÃO DE UM CODIFICADOR DE ROM

O codificador de uma ROM pode ser implementado empregando uma matriz de diodos, em lugar de portas OR, desde que uma matriz de diodos realiza uma função OR. A figura 11 mostra o circuito de duas portas OR com diodos: a primeira realiza $Z = A + B$ e a segunda, a operação OR para qualquer número de variáveis de entrada.

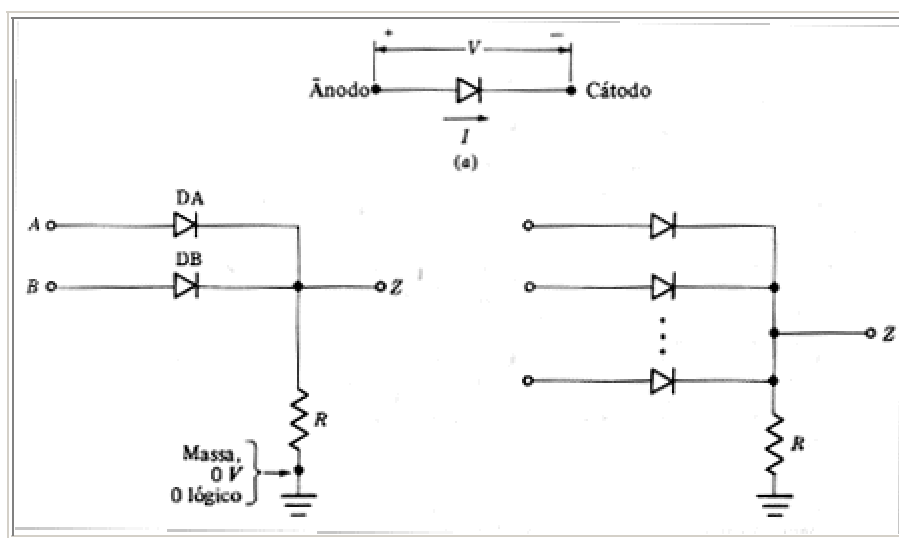


Figura 11. Codificador de ROM implementado com diodos.

Na figura 12 está apresentado o codificador implementado com diodos no lugar das portas OR vista no circuito ROM 8x4, anteriormente.

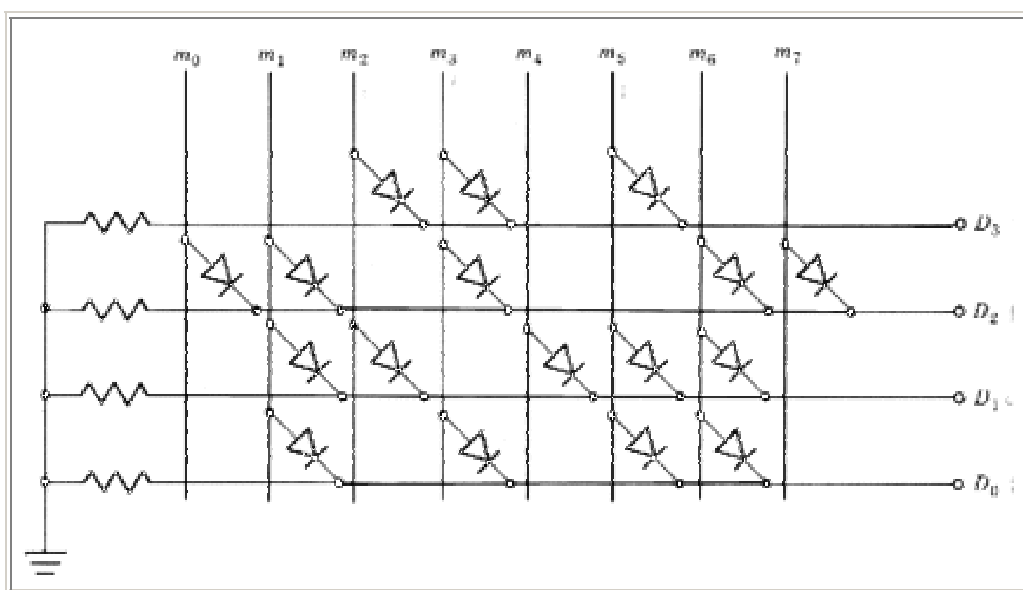


Figura 12. Rede de diodos como codificador de ROM

Veja que para cada 1 no cruzamento de uma linha mj com uma coluna Dk na tabela verdade existe um diodo na matriz do codificador na interseção, ligando a linha de endereço mj com a linha de dados Dk. Há tantos diodos no codificador quanto 1s nas colunas de dados da tabela da ROM.

4.11.2 HIERARQUIA E TIPOS DE MEMÓRIAS ROM

A figura 13 apresenta a hierarquia da memória ROM.

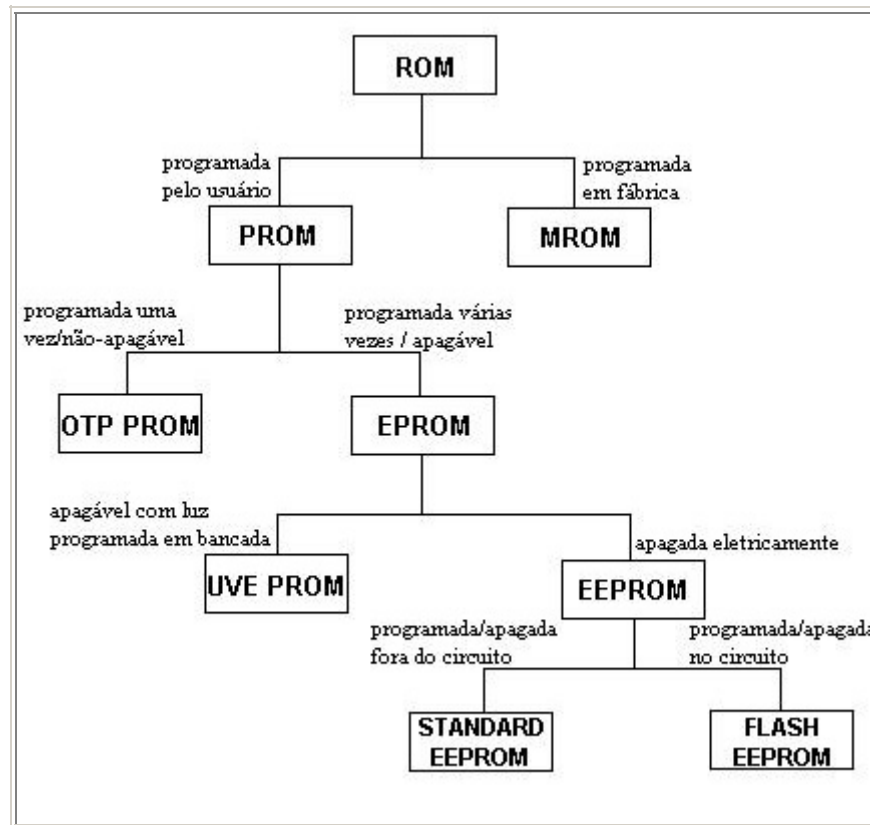


Figura 13. Tipos de memórias ROM

4.11.3 Memórias MROM (MASK ROM - ROM de MÁSCARA)

As memórias MROM têm suas locações de armazenamento programadas pelo fabricante de acordo com as especificações do cliente. As maiores desvantagens da MROM é que ela não pode ser reprogramada e só é econômica em grandes quantidades.

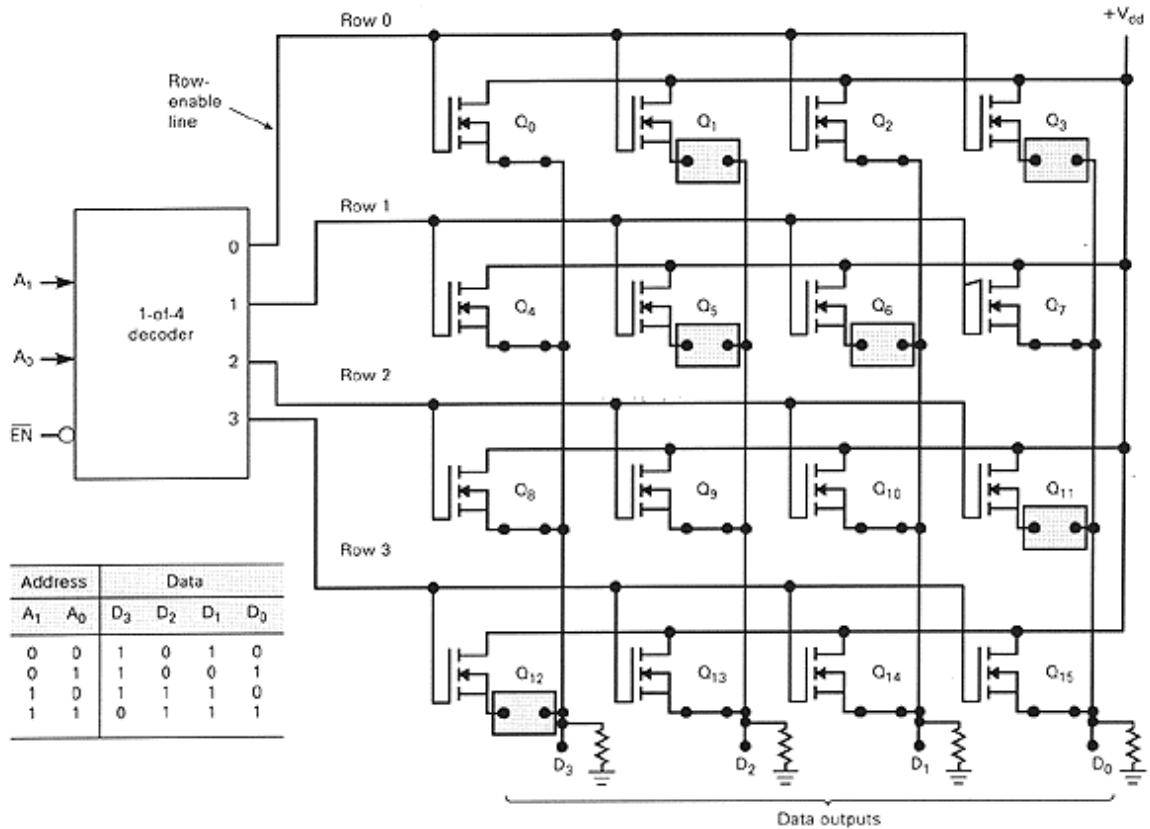


Figura 14. Estrutura de uma memória MROM.

A figura14 mostra a estrutura de uma MROM MOS com 16-células arranjadas em 4-linhas e 4-colunas.A conexão da fonte(source) dos transistores MOSFET às linhas de dados na saída de dados determina se a célula armazena 1 ou 0: conexão de fonte aberta armazena 0 e fechada armazena 1.Todas as células de uma coluna estão ligadas à linha de saída da coluna, porém apenas uma linha de células é habilitada por vez através das saídas do decodificador de endereços aplicadas nos gates.

O decodificador de endereços seleciona que linha terá seus dados lidos através do endereço A1A0.Quando o decodificador estiver desabilitado - entrada de habilitação no nível alto, então todos os transistores na matriz estão em corte, pois as saídas do decodificador estão baixas.

4.11.4 Memórias PROM (Programmable Read Only Memory - Memória Programável Só de Leitura)

Para aplicações de pequenas quantidades, os fabricantes desenvolveram PROMs com elos-fusíveis que podem ser programadas pelo usuário.Uma vez programada, a PROM não pode ser apagada e reprogramada. A figura 15 mostra como os fusíveis podem ser queimados para armazenar 0 na célula correspondente.

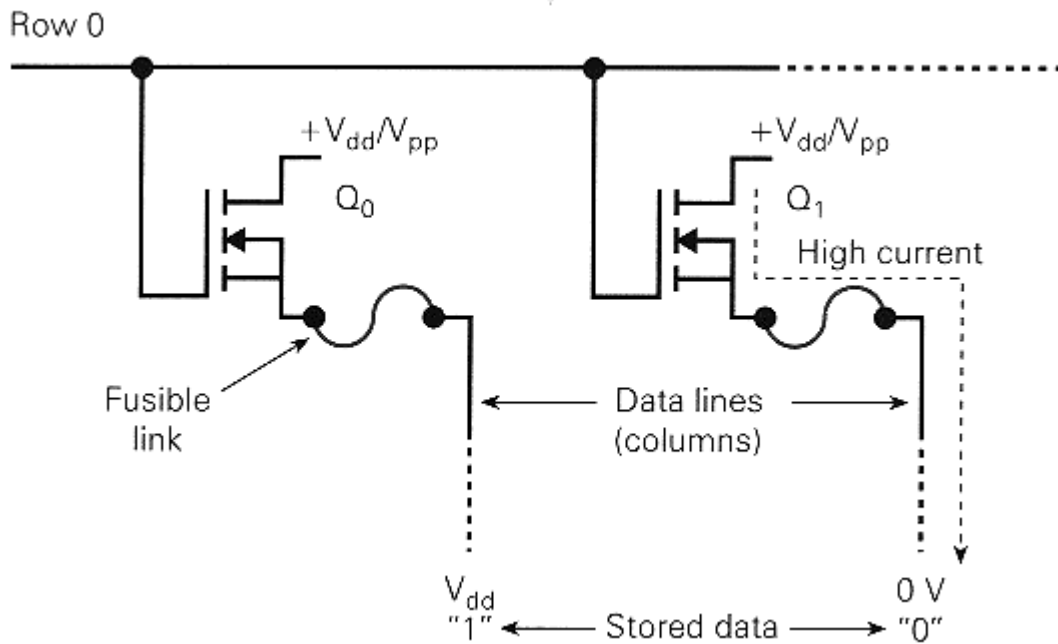


Figura 15. Esquema de uma memória PROM.

4.11.5 Memória EPROM (Erasable Programmable ROM - ROM Apagável e Programável)

As células de armazenamento em uma EPROM são transistor MOS com porta de silício sem conexão (flutuante). No estado normal, o transistor está desligado e a célula armazena 1 lógico. O transistor pode ser ligado aplicando um pulso de tensão que injeta elétrons de alta energia na região da porta. Estes elétrons ficam presos, pois não há caminho de fuga. A figura 16 mostra uma célula de uma EPROM.

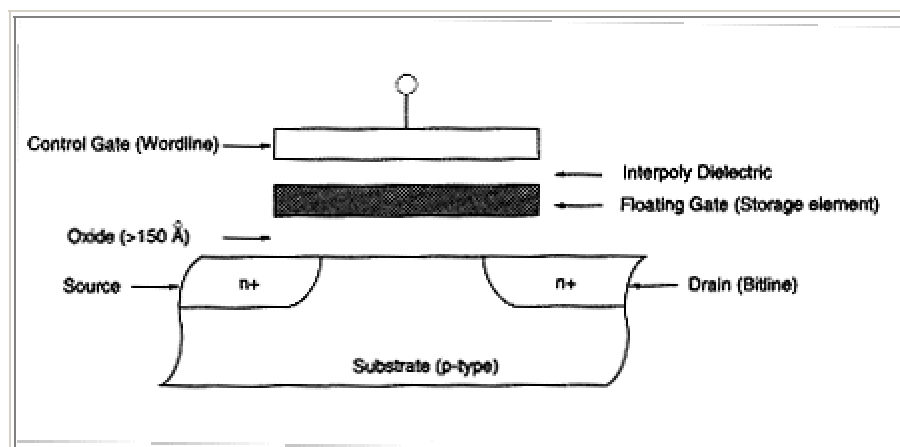


Figura 16. Célula de uma EPROM constituída de transistor MOS

Uma EPROM pode ser programada pelo usuário e também pode ser apagada e reprogramada quantas vezes desejar. O processo de programar uma EPROM consiste em aplicar tensões especiais (10 a 24V) nos pinos apropriados (entre a base e o coletor), durante um certo tempo (50ms por locação). Uma EPROM pode ser apagada se for exposta a luz ultravioleta, que produzirá uma corrente da porta flutuante para o substrato, removendo as cargas e desligando o transistor, e restaurando o 1 lógico. Este

processo deve gastar entre 15 e 20 minutos, dependendo da memória e intensidade de luz, e apagará toda a memória.



(a) EPROM, onde pode-se ver a janela para irradiação ultravioleta. (b) Apagador de EPROM.

Figura 17.

4.11.6 Memória EEPROM (Electrically Erasable Programmable ROM - ROM Apagável e Programável Eletricamente)

As memórias EEPROMs(E2PROMs) têm a mesma estrutura de porta flutuante que as EPROMs, mas com o acréscimo de uma região com uma fina camada de óxido sobre o dreno do transistor MOSFET da célula de memória. Esta modificação produz a característica mais importante das EEPROMs: são apagáveis eletricamente. Uma alta tensão(21 Volts) entre o dreno e a porta do MOSFET produz uma carga induzida na porta flutuante, que permanece mesmo retirando a tensão de alimentação. Uma tensão reversa de mesmo valor remove as cargas da porta flutuante e apaga a célula de memória. Uma vez que o transporte de cargas precisa de pequenas correntes, apagar e programar uma EEPROM podem ser realizados sem retirar a memória do circuito.

Outra característica da EEPROM é a possibilidade de programar e apagar eletricamente bytes individuais da matriz da memória. Isto torna muito mais fácil modificar os dados de uma EEPROM. É muito mais rápida a programação de uma EEPROM, pois o tempo de programar uma posição de memória é 5ms.

As EEPROMs mais modernas têm internamente os circuitos de suporte para gerar a tensão de programação, controlar a temporização e seqüência das operações de programação da memória, o que torna mais simples usar as EEPROMs. A possibilidade de apagar bytes e o alto nível de integração das EEPROMs apresentam duas desvantagens: densidade e custo. A complexidade da célula de memória e o circuito de suporte de uma EEPROM reduzem a capacidade de bit por milímetro quadrado da pastilha de silício, aproximadamente duas vezes a área de uma EPROM.

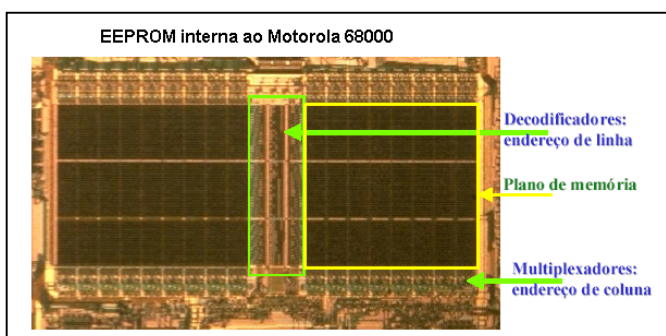


Figura 18. Figura da EEPROM interna ao Motorola 68000.

4.12 Memórias FLASH

A memória FLASH é um dispositivo de armazenamento confiável, não volátil, de boa relação custo/ benefício e que possui características de leitura da EPROM, EEPROM e SRAM , porém quando aplica-se 12V sobre o dispositivo, este pode ser gravado com base em bytes. No caso da memória FLASH - 5V estes dispositivos foram projetados para serem programados dentro do sistema com o fornecimento padrão de 5V. Em programadores de EPROM convencionais não há necessidade de 12Vpp, nem para programação, nem para apagamento. É composta de uma arquitetura de apagamento de setor (qualquer combinação pode ser apagada simultaneamente) e 100.000 ciclos de apagamento/ programação. As figuras 19a, 19b e 19c mostram o dispositivo básico da memória flash e os princípios de programação de um transistor de *gate* flutuante.

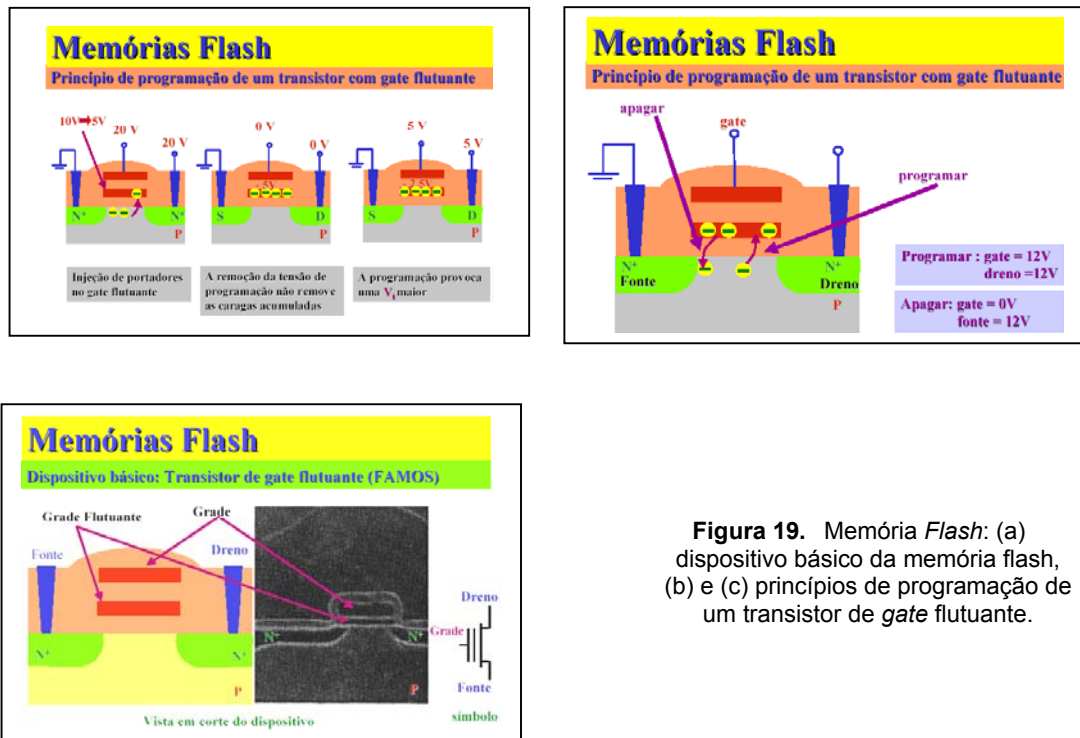


Figura 19. Memória Flash: (a) dispositivo básico da memória flash, (b) e (c) princípios de programação de um transistor de *gate* flutuante.

A memória Flash é largamente usada para conter códigos de controle ou sistemas operacionais dedicados, como a BIOS de um computador, o programa de controle de um telefone celular, câmeras digitais, receptores de satélite domésticos, controladores embutidos, adaptadores de vídeo e outros dispositivos. O uso da memória Flash nestes dispositivos permite a atualização dos programas de controle de maneira fácil, o que traz extrema vantagem de custo e tecnologia. Todo um programa de controle de um telefone celular pode ser atualizado e funções novas serem adicionadas, sem que a parte física precise ser modificada, o mesmo acontecendo com as BIOS de computadores. Quando estes códigos precisam ser regravados, a memória Flash pode ser reescrita em blocos, facilitando a atualização. Esta característica torna-a inadequada como memória de acesso randômico (RAM), porque a RAM precisa poder ser endereçada byte a byte.

A figura 20 mostra uma memória Flash de 20Kbytes.

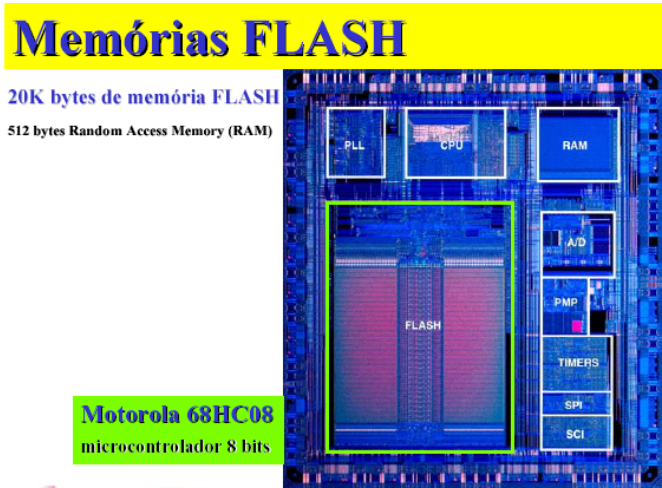


Figura 20. Memória Flash de 20Kbytes.

4.13 TEMPOS DE CHAVEAMENTO DAS MEMÓRIAS

Para exemplificar os tempos de leitura e escrita das memórias, tomaremos como base a memória RAM-Estática, pois todos os conceitos aplicados para este modelo podem ser estendidos para os demais.

4.13.1 Ciclo de Leitura

O ciclo de leitura é mostrado na figura 21.

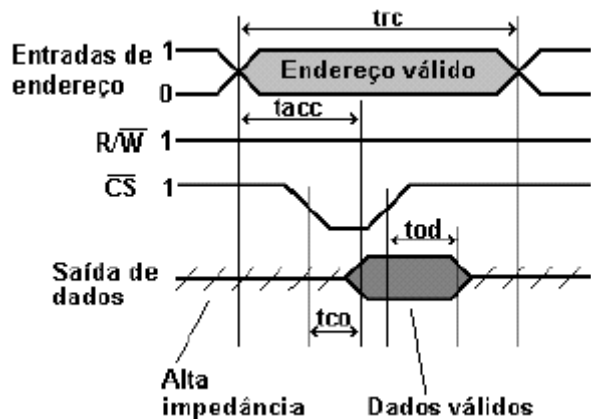


Figura 21. Ciclo de leitura de uma RAM estática

Onde:

trc = intervalo de duração do ciclo de leitura;

tacc = tempo de acesso à RAM;

tco = tempo que a saída da RAM leva para sair de alta impedância e ter um dado válido;

tod = tempo decorrido entre a desabilitação da RAM e o instante que as saídas da RAM vão para alta impedância.

Durante o ciclo de leitura, a CPU coloca um endereço no barramento de endereços ligado à RAM. Este endereço corresponde à posição de memória que deve ser lida.

A CPU sinaliza a operação de leitura através de um nível ALTO na entrada W/R. A CPU habilita a memória colocando a entrada CS em nível BAIXO. A RAM responde a isso colocando o conteúdo da posição endereçada nas saídas de dados. Depois disso a memória é desabilitada (CS é ALTO) e suas saídas ficam em tri-state (alta impedância).

4.13.2 Ciclo de Escrita

O ciclo de escrita é mostrado na figura 21.

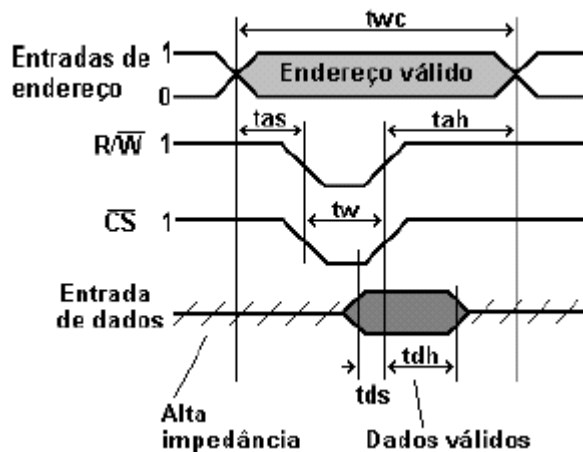


Figura 22. Ciclo de escrita de uma RAM estática.

Onde:

twc = intervalo de duração do ciclo de escrita;

tas = tempo para estabilização do barramento de endereços, antes de habilitar a RAM;

tah = intervalo necessário para que o barramento de endereços permaneça estável;

tw = tempo de escrita, onde CS e W / R ficam em BAIXO;

tds = tempo que os dados devem ser mantidos na entrada antes da desabilitação de CS e W / R ;

tdh = tempo que os dados devem ser mantidos na entrada depois da desabilitação de CS e W / R .

No ciclo de escrita, a CPU coloca um endereço estável no barramento de endereços. Nesse endereço o dado deve ser escrito. Ela coloca o dado a ser armazenado no barramento de dados.

A CPU sinaliza a operação de escrita, colocando nível BAIXO na entrada W/R . A CPU habilita a memória colocando a entrada CS em nível BAIXO. O dado é transferido para a posição de memória indicada pelo endereço. A memória é, então, desabilitada (CS é ALTO) e suas saídas ficam em tri-state.

4.14 APLICAÇÕES DE MEMÓRIAS ROMS

Tradicionalmente as ROMs são empregadas como memórias de armazenamento de programas e dados de sistemas de computadores. O desenvolvimento de ROMs, PLAs e PALs proporciona uma substituição de circuitos combinacionais e seqüenciais complexos, reduzindo o tamanho dos circuitos, consumo de energia, e preços.

4.14.1 Firmware

Firmware são dados e códigos de programas que devem ser disponíveis ao se ligar sistemas de microprocessadores. Equipamentos como automóveis, VCRs, CD players, e outros empregam *firmware* em memórias ROM.

4.14.2 Memória de *Bootstrap*

Programas de *bootstrap* são pequenos programas armazenados em ROM que são executados ao ligar o microcomputador e se encarregam de carregar o sistema operacional. Este programa inicializa a CPU e o hardware.

4.14.3 Tabela de Dados

As memórias tipo ROM podem armazenar tabelas de dados que não mudam. Como exemplo, a ROM MM4220BM, que armazena as funções seno de 0 a 900. A memória é organizada em 128 palavras de 8-bits cada, as entradas de endereços representam os ângulos em incrementos de 0,70.

4.14.4 Conversores de Dados

As memórias ROMs são programadas de modo que a aplicação de um endereço particular (código de entrada) produz dados na saída que representam o equivalente a um novo código. A ROM TTL 74185, mostrada na Figura 23, implementa a conversão de código binário-para-BCD para uma palavra de entrada de 6-bits.

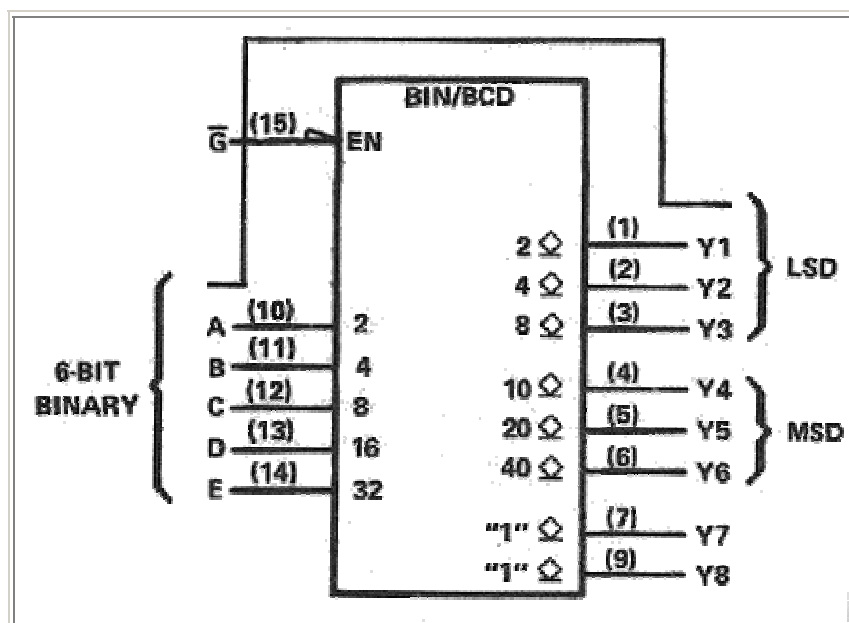


Figura 23. ROM TTL 74185.

4.14.5 Gerador de Funções

Como gerador de funções analógicas, a memória ROM é empregada junto com um chip Conversor Digital/Analogico(DAC) e um Contador(Counter), conforme mostrado na figura 24.

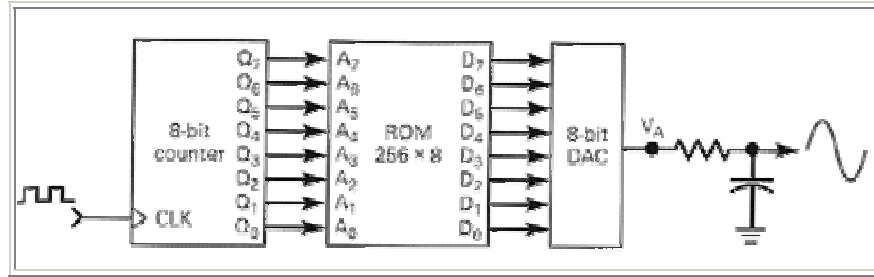


Figura 24. Gerador de funções analógicas.

A ROM armazena 256 valores de 8-bits, cada um correspondendo a um ponto da forma-de-onda. Quando o contador varre um ciclo de contagem (256 endereços diferentes), as saídas da ROM produzem 256 pontos de saída para o DAC. A saída do DAC é uma forma-de-onda em passos de 256 valores. O filtro passa-baixas retira as imperfeições da forma-de-onda, isto é, suaviza a saída do DAC.