

Memórias

Henrique Bernardes

24 de outubro de 2025

1 Introdução - Registradores

De início, temos os registradores, compostos por flip-flops e multiplexadores 2 para 1 controlados a partir de um sinal de "load":

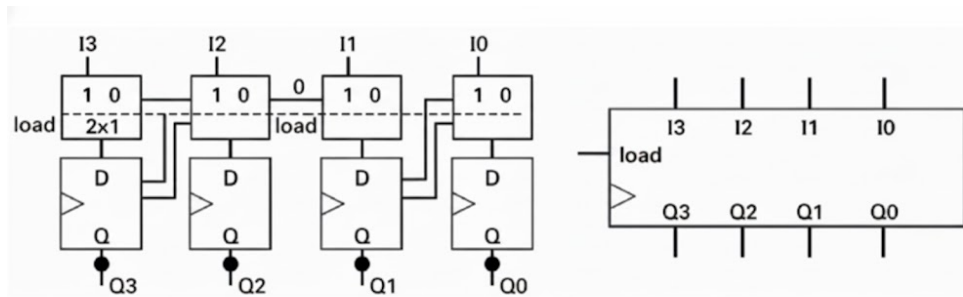


Figura 1: Registradores

Existem, aqui, duas situações:

- Quando $load = 1$ e clk estiver na borda de subida, o flip-flop é carregado com a entrada de dados I (I_i)
- Quando $load = 0$ e clk estiver na borda de subida, o flip-flop é carregado com sua saída Q (Q_i)

Essas estruturas simples podem ser utilizadas para criar os chamados "bancos de registradores". Os bancos de registradores utilizam endereços para acessar cada registrador. No exemplo da imagem 2, um decodificador 3:8 é utilizado para endereçar, a partir de 3 bits, 8 endereços e, então, direcionar o dado *input* para o registrador adequado.

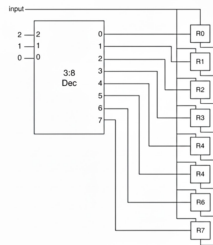


Figura 2: Banco de Registradores

2 Memórias

Em um projeto RTL, são utilizados componentes adicionais além dos blocos operacional e de controle, como por exemplo memórias. Um dos tipos de memória é denominada *RAM - Random Access Memory*.

Como visto na imagem 3, um bloco de memória *RAM* é descrito por dois números, como, na imagem, 1024×32 . O primeiro número, 1024, indica o número de palavras (profundidade) que compõem o bloco de memória. Já o segundo número, 32, indica o tamanho de cada palavra (largura). Cada palavra pode ser entendida como um registrador, ou seja, cada palavra pode ser lida e escrita individualmente a partir de entradas de endereço e dados. Detalhando os demais sinais:

- *addr* - 10 bits ($2^{10} = 1024$ palavras)
- *rw* = 0 - leitura
- *rw* = 1 - escrita
- *en* = 0 - desabilita a memória

Para realizar uma leitura, lê-se primeiro uma linha inteira de palavras e, então, seleciona-se a palavra apropriada nessa linha.

Finalmente, para armazenar uma maior quantidade de bits, uma *RAM* típica, ao invés de utilizar flip-flops, utiliza estruturas mais compactas.

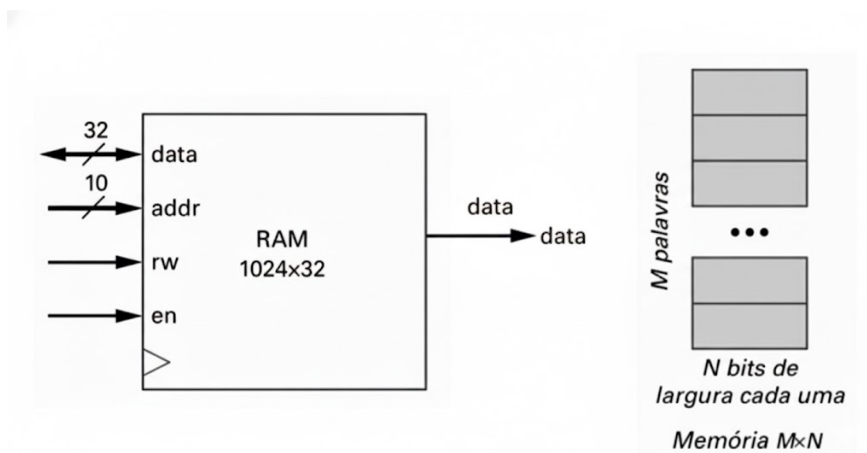


Figura 3: Memória RAM

2.1 Estrutura da RAM

Uma *RAM* se baseia em uma grade que contém blocos de armazenamento, ou as chamadas **células**. Cada célula é alimentada com um certo dado; contudo, apenas ao ser ativada pelo decodificado (de dimensão $A \times M$), alimentado por um endereço, que as células dessa palavra específica são ativadas, como mostra a imagem 4.

A entrada *rw* se conecta a todas as células e determina dois cenários para cada célula:

- Se o valor da célula será lido e disponibilizado na saída $rdata(N - i)$
- Se o valor da entrada $wdata(N - i)$ será gravado na célula.

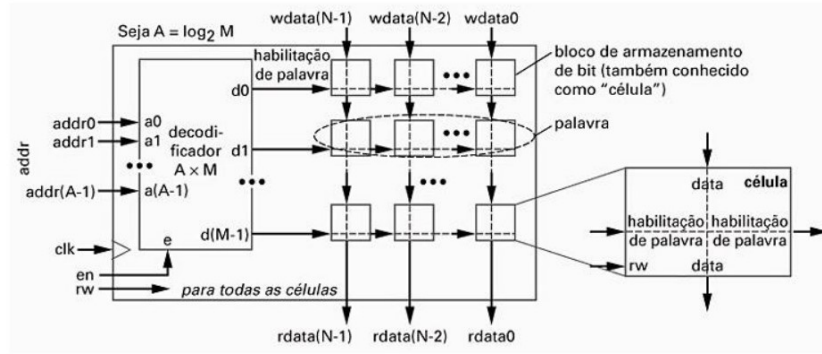


Figura 4: Estrutura da RAM, organizada em células que formam palavras na horizontal.

3 Memória RAM estática(SRAM)

No caso da SRAM, o bit fica armazenado no laço criado pelos inversores.

Para as operações de escrita:

- Escrita de 1: Faz-se $data = 1$ e $data' = 0$ e $enable = 1$, ativando os transistores.
- Escrita de 0: Faz-se $data = 0$ e $data' = 1$ e $enable = 1$, ativando os transistores.

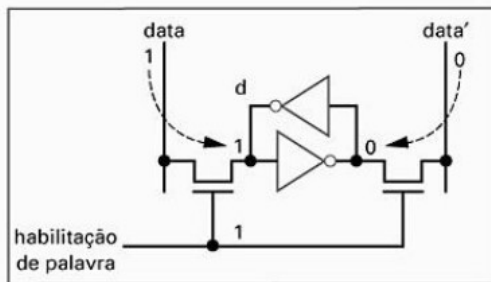


Figura 5: Operação de escrita SRAM

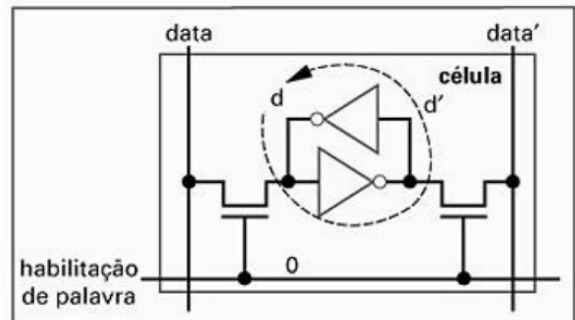


Figura 6: Estrutura SRAM, composta por seis transistores

Para as operações de leitura, primeiro se realiza o carregamento de 1 em ambas as linhas $data$ e $data'$ e também no sinal de habilitação.

Em um dos lados, terá um 0 armazenado, provocando uma queda de tensão. Um circuito amplificador e comparador de tensão detecta em qual lado houve a maior queda.

- Se a tensão em $data > data'$ então a célula tem 1 armazenado. Houve queda de tensão em $data'$
- Se a tensão em $data < data'$ então a célula tem 0 armazenado. Houve queda de tensão em $data$

Enquanto houver energia alimentando os transistores, a SRAM mantém o bit armazenado(volátil) sem que ele mude(estático).

Por ser composta de apenas seis transistores, a memória SRAM utiliza menos transistores que um flip-flop.

Por necessitar de um circuito amplificador e comparador, o acesso aos bits armazenados(leitura) requer mais tempo.

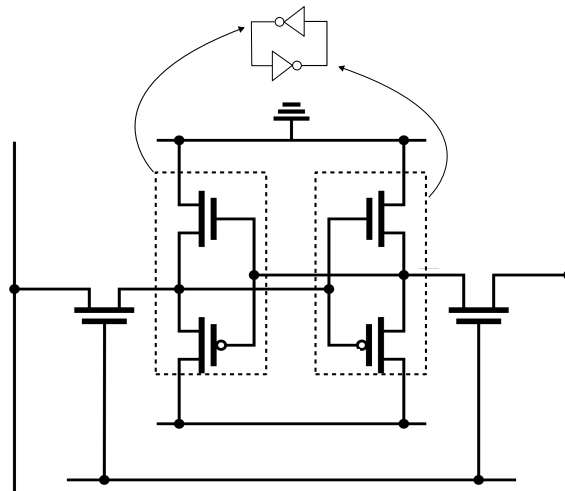


Figura 7: Estrutura interna da SRAM

4 Memória RAM Dinâmica(DRAM)

A memória DRAM é composta por um capacitor e um transistor.

Para as operações de **escrita**, primeiro se escreve o valor a ser armazenado na linha *data* e então *enable* = 1, carregando o capacitor (de capacitância elevada) que, então, sofre uma lenta queda, como mostra a imagem 8.

Para a operação de leitura, alimenta-se a linha de dados com uma tensão intermediária entre 0 e 1. Depois, faz-se *enable* = 1, fazendo com que a tensão armazenada no capacitor cause uma queda de tensão na linha de dados. Da mesma forma que na *SRAM*, aqui um circuito amplificador e comparador é utilizado para definir qual era o valor armazenado.

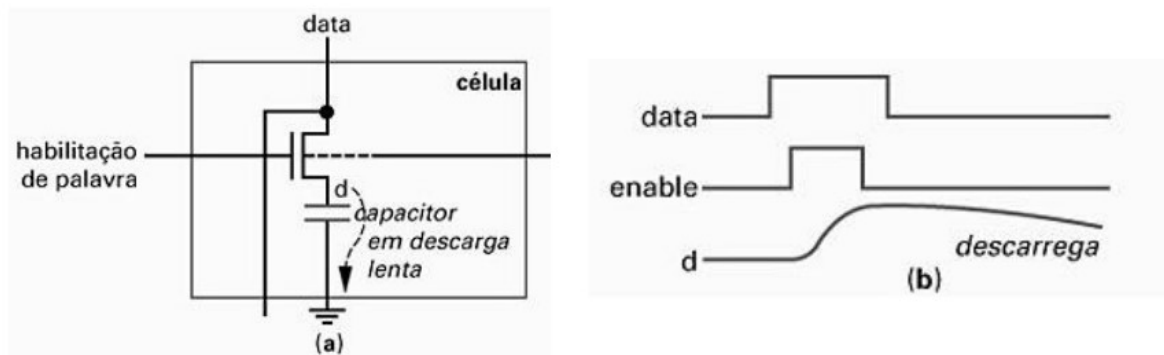


Figura 8: Estrutura da *DRAM*, composta de um transistor e um capacitor.

As DRAMs tem mais capacidade de armazenamento do que as SRAMs, contudo, são mais lentas que as SRAMs pois cada leitura deve ser reescrita novamente, aumentando o tempo de acesso.

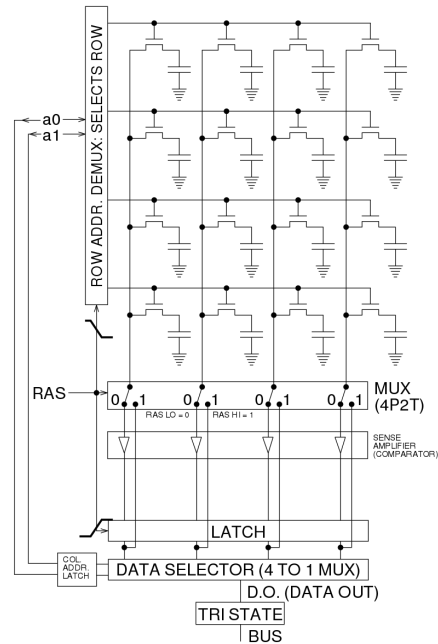


Figura 9: Array 4x4 DRAM

5 ROM (READ ONLY MEMORY) - PROM, EPROM e EEPROM

Começamos com as vantagens da ROM em relação à RAM:

- Maior capacidade de armazenamento ;
- Não volatilidade: conteúdo é preservado mesmo após o desligamento do fornecimento energético;
- Mais rápidas;
- Baixo consumo energético → a ROM não consome energia para manter o seu conteúdo.

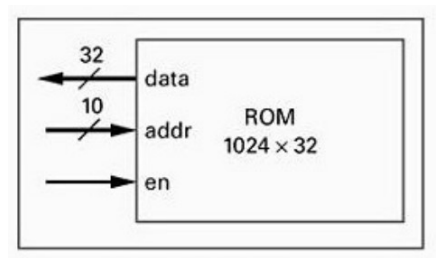


Figura 10: Símbulo memória ROM

É importante notar, aqui, que não há mais entrada de dados, apenas saída, como indicado pelo símbolo e, também, não há mais clock: ROM pode ser entendida como uma unidade puramente combinacional, embora existam variações.

5.1.2 OTP-ROM - ROM programável por fusível - PROM

Este tipo de ROM utiliza um fusível em cada célula, que, ao ser fabricada, possui todos os fusíveis intactos.

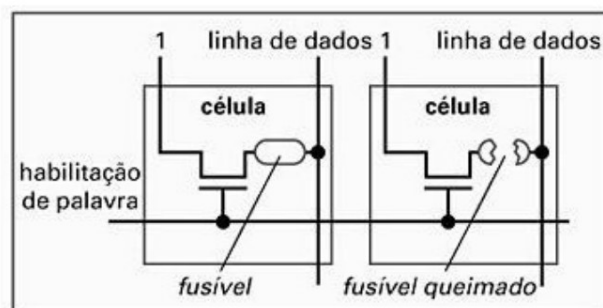


Figura 13: ROM programável por fusível - OTP-ROM/PROM

Para programá-la, utiliza-se um dispositivo que fornece uma corrente acima do normal àquelas células que se deseja programar com 0. Para a detecção desta alta impedância como um 0 lógico, é utilizado um circuito extra. Esse tipo de ROM também não é reprogramável.

5.1.3 PROM apagável - Erasable PROM - EPROM

Este tipo de ROM utiliza um tipo de transistor com "porta flutuante," na qual os elétrons ficam aprisionados. Um transistor que possui elétrons aprisionados em sua porta permanece em estado de não condução, ou seja, está programado para armazenar um 0. Inicialmente, os transistores não possuem elétrons aprisionados, logo, estão todos em estado lógico 1.

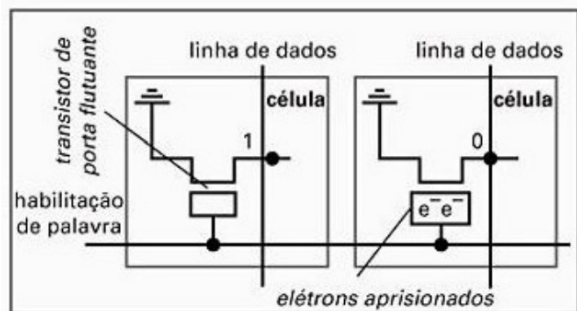


Figura 14: PROM apagável - Erasable PROM - EPROM

EPROM Transistor/Cell

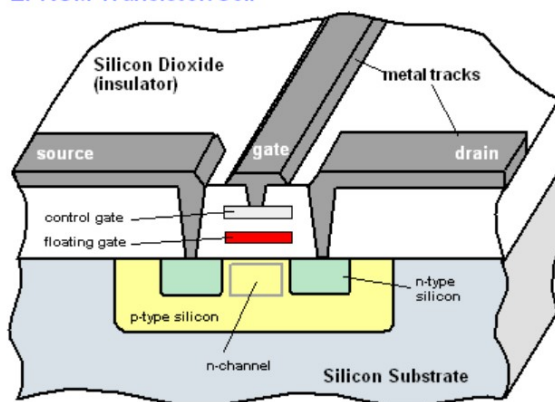


Figura 15: Floating-gate MOSFET - Transistor utilizado nos EPROMs

Para a programação, um dispositivo aplica uma alta tensão àquelas transistores daquelas células que devem armazenar 0. Essa alta tensão, produz um efeito de tunelamento através do pequeno material isolante até a porta flutuante, fazendo os elétrons acumularem. Quando a tensão é removida, os elétrons não têm energia suficiente para criar o túnel de volta, permanecendo aprisionados na célula. Como consequência, há o aumento da tensão de threshold para o transistor fechar o canal.

Para que o conteúdo seja apagado, o chip da placa de circuito da EPROM precisa ser retirado e exposto à luz ultravioleta para energizar os elétrons aprisionados previamente, de forma que possam vencer a barreira do material isolante da porta flutuante, apagando, assim, todos os 0's. É importante mencionar que **todo** o conteúdo é apagado: não é possível apagar somente uma parte da memória. Em média, podem ser apagadas e reprogramadas mais de 10 mil vezes, mantendo o conteúdo por 10 anos ou mais.



Figura 16: Chips EPROMs expostos.

5.1.4 PROM eletricamente apagável - EEPROM

A diferença base entre uma EPROM e uma EEPROM é que esta não precisa de luz UV para ser apagada e sim de uma alta tensão na linha de dados. Também, aqui, essa tensão pode ser aplicada em células específicas, ou seja, é possível apagar e reprogramar apenas certas palavras.

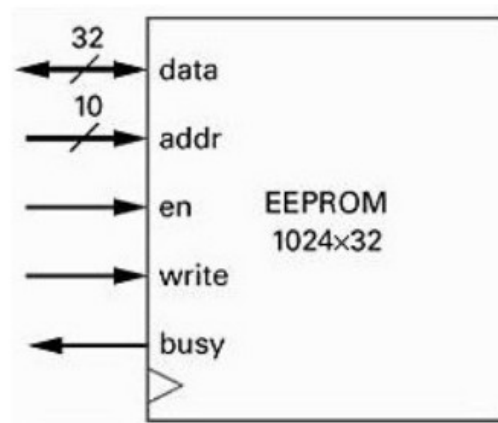


Figura 17: Símbulo memória EEPROM

A estrutura base de uma memória EEPROM é vista na imagem 18. Aqui, são utilizados dois transistores, sendo um deles um *Floating-gate MOSFET*, do mesmo tipo utilizado nas EPROMs. O MOSFET tradicional é utilizado para selecionar qual célula deseja escrever; já o FG MOS contém o valor guardado na célula.

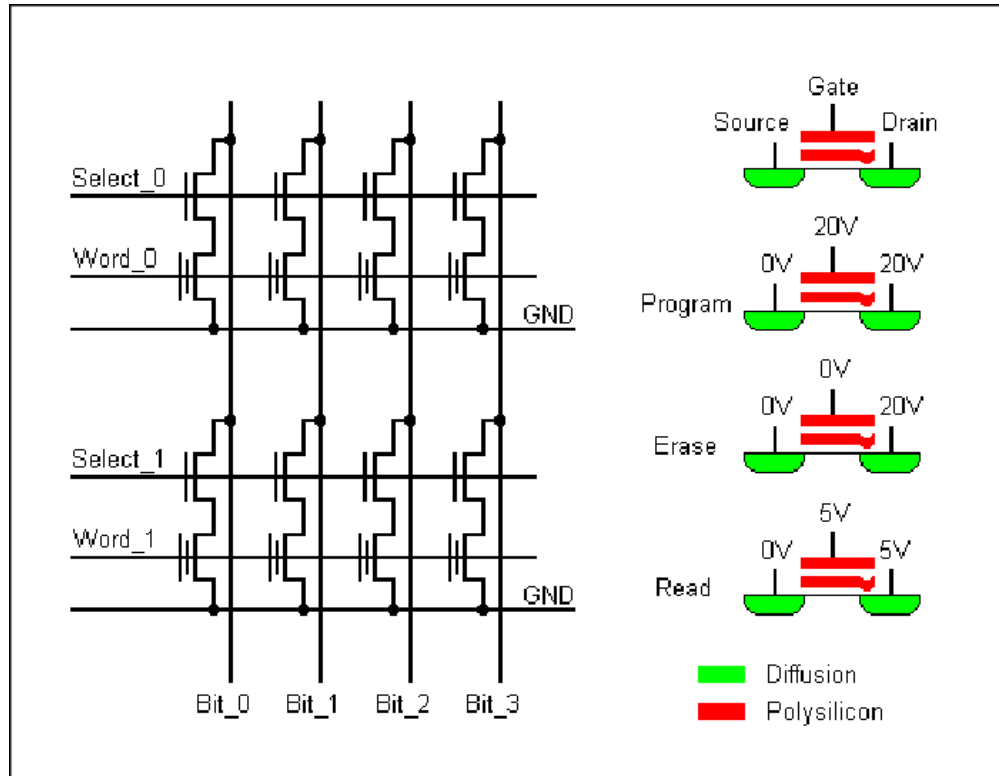


Figura 18: Estrutura de um array de memorias EEPROM.

Para se escrever em uma célula, aplica-se tensões altas no gate e drain do FGMOS, causando assim o tunelamento e, assim, o aprisionamento dos elétrons na porta flutuante.

Para apagar o valor em uma célula, aplica-se uma alta tensão no drain do FGMOS, concedendo energia suficiente para os elétrons presos na porta flutuante para saírem pela porta drain e assim apagar o valor guardado no transistor.

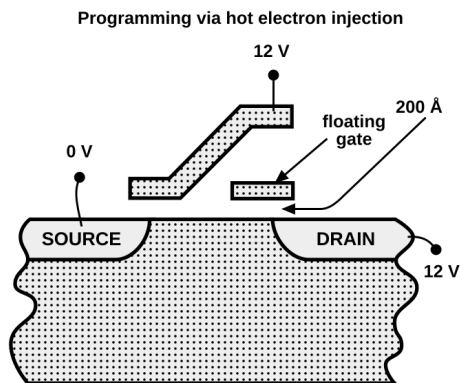


Figura 19: Escrevendo em uma unidade EEPROM

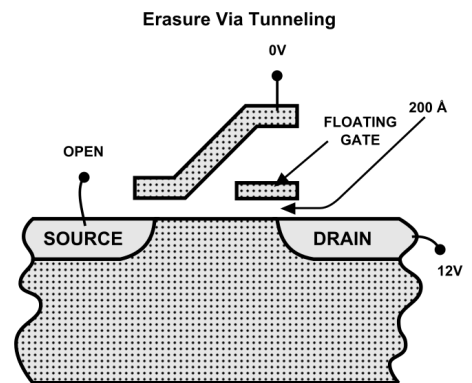


Figura 20: Apagando em uma unidade EEPROM

Memórias EEPROM modernas (*in system programmable (ISP)*) possuem controladores internos para programação, ou seja, é possível programa o conteúdo de uma EEPROM sem retirar o chip da placa de circuito.

Ou seja, funcionam de forma similar a uma RAM. Contudo, a programação de uma EEPROM leva muito mais tempo do que a leitura, uma sendo da ordem de dezenas de microssegundos e a outra da ordem de nanossegundos, respectivamente. EEPROMs podem ser programadas milhões de vezes e preservar seu conteúdo por cem anos ou mais sem energia.

Embora as EEPROMs permitam apagar células individuais, existem aplicações nas quais é necessário apagar grandes blocos de memória. Neste caso, são utilizadas as memórias Flash, que implementam justamente essa funcionalidade. Ou seja, memórias Flash não são apagáveis por bit(ou *bit erasable*), mas sim apagáveis globalmente. Também, as células podem ser de apenas um transistor, aumentando a densidade de componentes se comparado com memórias EEPROM.