5 Circuitos de Armazenamento

Nos sistemas digitais, e em particular nos computadores, as informações estão representadas por conjuntos de dígitos binários denominados "palavras". Nos computadores atuais o tamanho da palavra é de 32, 64 ou 128 bits. Porém, até há pouco tempo atrás, os computadores pessoais usavam apenas 8 e 16 bits.

Naturalmente, um sistema digital é projetado para trabalhar com um determinado tamanho de palavra, devendo portanto conter recursos de hardware que lhe permitam processar e armazenar simultaneamente conjuntos de n bits, onde n é o tamanho da palavra.

A seguir estudaremos os circuitos digitais responsáveis pelo armazenamento de informação. Alguns destes circuitos são construídos de modo a também poder manipular a informação armazenada. Dentre as operações possíveis estão os deslocamentos (à direita e à esquerda), o incremento e o decremento.

5.1 Registradores

Um **registrador** é um circuito digital formado por *n* flip-flops, de modo a poder armazenar simultaneamente (e de maneira independente) *n* bits. Trata-se de um tipo de elemento de armazenamento básico: um processador possui um conjunto de registradores que pode variar de três a algumas dezenas. A existência de registradores dentro do processador acelera o processamento, pois os dados que estão sendo manipulados ficam armazenados próximo dos recursos de processamento (ULA, por exemplo), o que reduz os acessos feitos à memória.. A figura 5.1 mostra um registrador de 4 bits feito com flip-flops D (disparados pela borda ascendente).

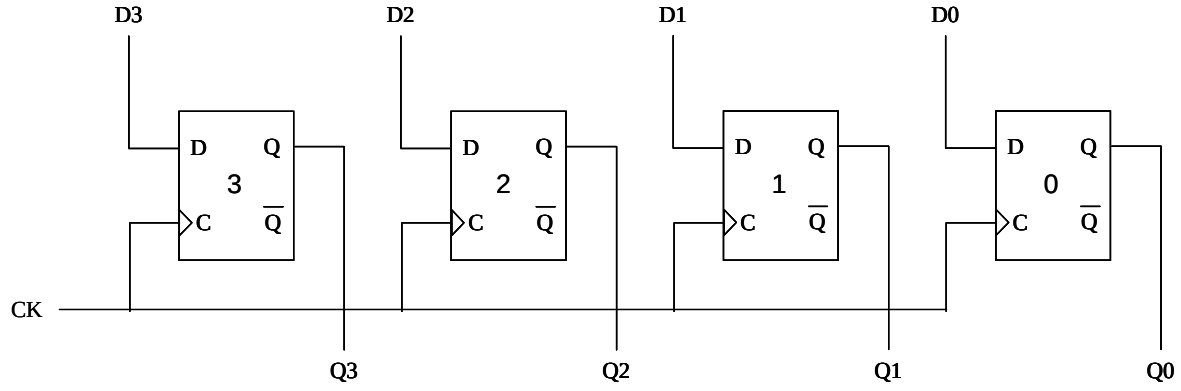

Figura 5.1 - Um registrador de 4 bits, com carga paralela.

Note que cada flip-flop é responsável pelo armazenamento de um bit, seguindo a notação posicional, e que cada bit possui um caminho independente dos demais, tanto para entrada como para a saída. Por isso, o registrador é dito "de carga paralela". Note também que o flip-flop de índice 0 armazena o bit menos significativo e o flip-flop de índice 3 armazena o bit mais significativo de uma palavra de 4 bits. Um registrador funciona como uma barreira:

os dados disponíveis nas entradas D0, D1, D2 e D3 somente serão copiados quando o sinal de relógio (CK, no caso) passar por uma borda ascendente. Os valores copiados quando da passagem de uma borda ascendente permanecerão armazenados pelos flip-flops até a ocorrência da próxima borda ascendente. Isto deixa o registrador imune a eventuais mudanças indesejadas dos sinais representados por D0, D1, D2 e D3. O valor armazenado num flip-flop qualquer está sempre presente na sua saída Q. Isto quer dizer que o dado armazenado no registrador pode ser consultado por outro recurso de hardware a qualquer tempo, desde que haja um caminho físico (i.e., um conjunto de fios) entre a saída do registrador e a entrada do outro elemento. O outro elemento pode ser, por exemplo, um somador/subtrator como no capítulo 3.

O registrador da figura 5.1 apresenta, porém, uma deficiência grave: toda a vez que o sinal de relógio CK passar por uma borda ascendente, os valores das entradas D0, D1, D2 e D3 serão copiados, mesmo que isso não seja explicitamente desejado. Entretanto, os circuitos de um computador devem seguir as ordens dos sinais provenientes da unidade de controle. O sinal de relógio serve apenas para determinar o momento no qual uma ordem deverá ser cumprida. Logo, um registrador de um computador deve possuir recursos capazes de realizar a carga do dado (i.e., a carga paralela dos sinais conectados as suas entradas) quanto o relógio passar pela borda ativa somente se o sinal de "carga" (também conhecido por "load") estiver ativado. A figura 5.2 mostra um registrador de 4 bits com carga paralela e sinal de carga.

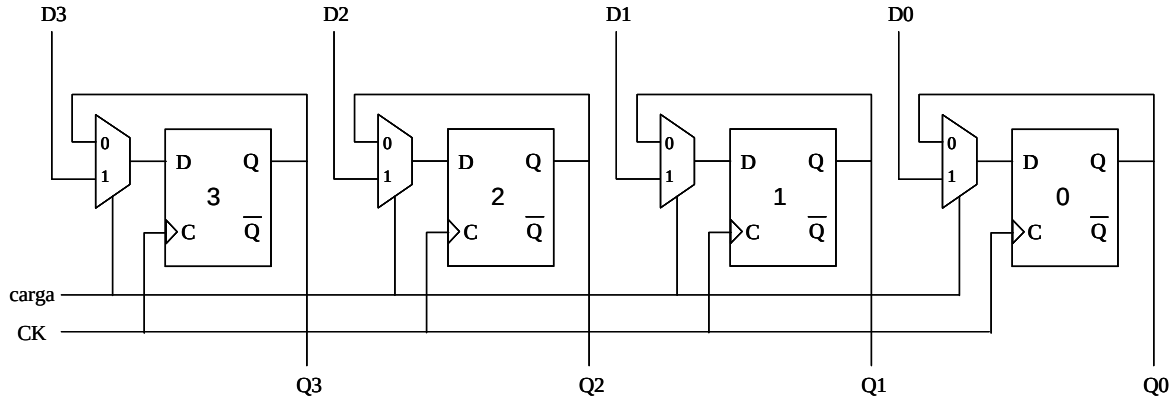

Figura 5.2: - Um registrador de 4 bits, com carga paralela e sinal de carga.

Nos desenhos de circuitos digitais mais complexos, será utilizado preferencialmente o símbolo mostrado na figura 5.3, ao invés do esquema completo do circuito, a fim de facilitar a compreensão.

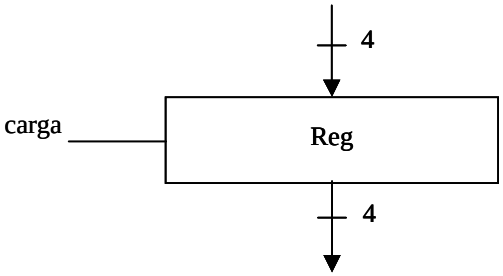

Figura 5.3 - Símbolo para um registrador de 4 bits, com carga paralela e sinal de carga. "Reg" é o nome do registrador.

5.1.1 Registrador com Carga Paralela (e Sinal de Carga)

Exemplo 5.1: desenhar a forma de onda da saída Q para o circuito que segue.

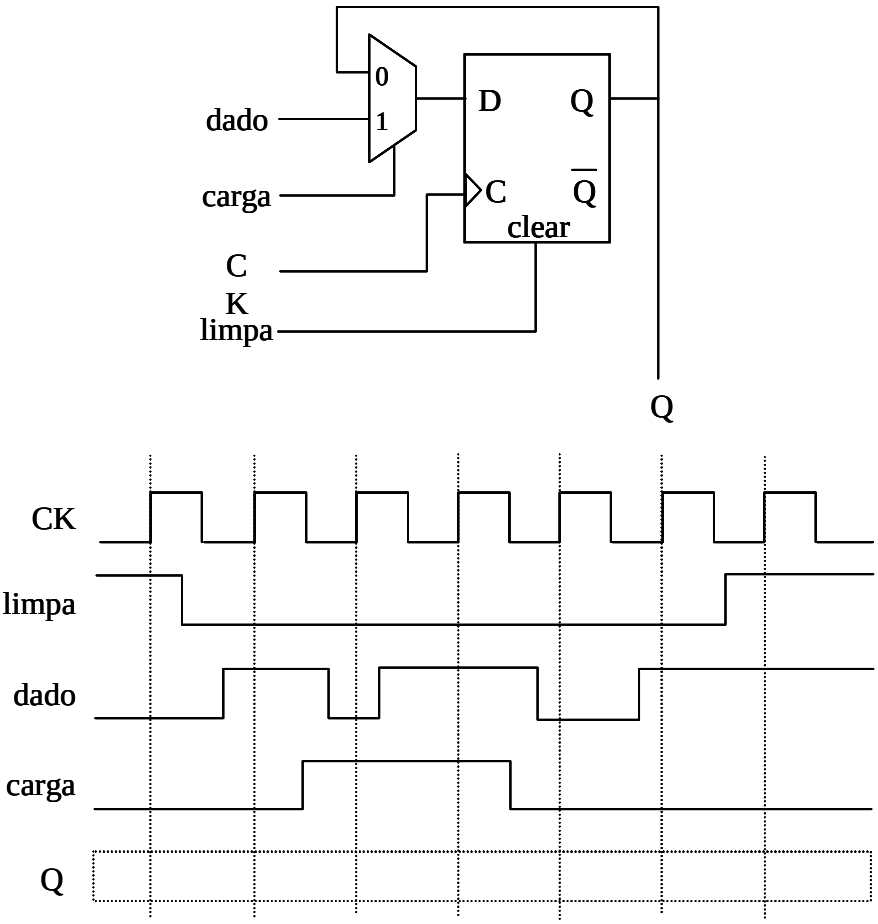

Figura 5.4 - Um bit do registrador com carga paralela.

Este exemplo ilustra o funcionamento do circuito associado a um bit de um registrador de carga paralela com sinal de carga.

5.1.2 Registradores de Deslocamento (*shift registers*)

Uma operação muito importante na aritmética binária é o deslocamento de bits. Essa operação consiste em deslocar o conteúdo de um flip-flop para o seu adjacente. A operação pode se dar da esquerda para a direita (deslocamento à direita) ou da direita para a esquerda (deslocamento à esquerda).

No primeiro caso, cada flip-flop recebe o conteúdo do seu vizinho imediatamente à esquerda. O flip-flop mais à esquerda recebe o dado de uma "fonte" externa pela "entrada serial". Já o conteúdo do flip-flop mais à direita é descartado.

No segundo caso, cada flip-flop recebe o conteúdo do seu vizinho imediatamente à direita. O flip-flop mais à direita recebe o dado de uma "fonte" externa pela "entrada serial". Já o conteúdo do flip-flop mais à esquerda é descartado.

Exemplo 5.2: traçar as formas de onda dos bits armazenados no registrador-deslocador mostrado a seguir, a partir das formas de onde fornecidas.

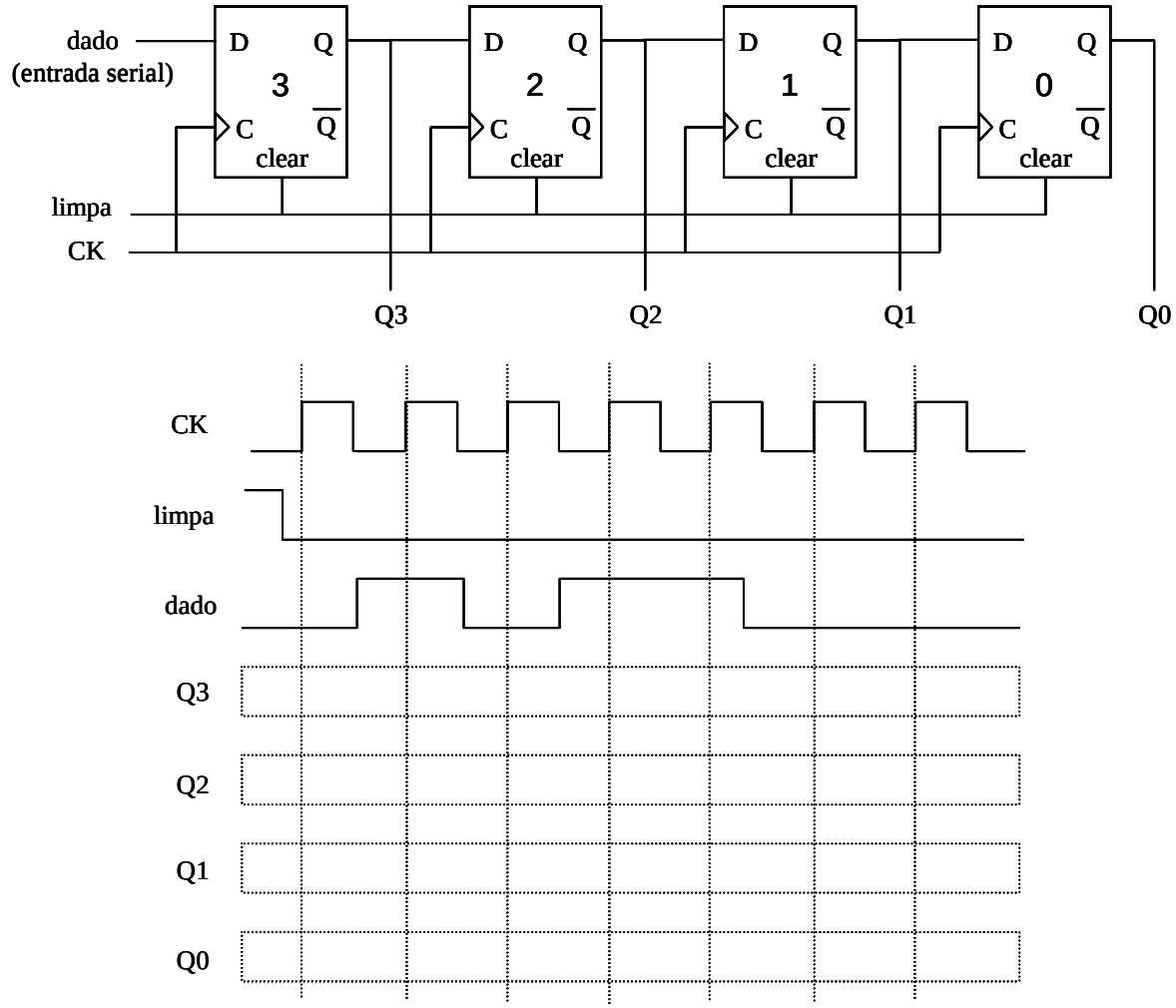

Figura 5.5 - Registrador de deslocamento à direita de 4 bits (com reset assíncrono): exemplo de funcionamento.

Repare que há uma ligação entre a saída de cada flip-flop e a entrada do seu vizinho imediatamente à direita (adjacente a direita). O registrador de deslocamento do exemplo 5.2 não possui sinal de carga. Porém, tal sinal normalmente existe, como será visto mais adiante.

Um registrador de deslocamento à esquerda deve apresentar uma ligação entre a saída de cada flip-flop e a entrada do flip-flop imediatamente à esquerda. Um tal registrador é mostrado na figura 5.6. Note que a entrada serial está conectada ao flip-flop mais à direita (flip-flop que armazena o bit menos significativo).

Exemplo 5.3: traçar as formas de onda dos bits armazenados no registrador-deslocador mostrado a seguir, a partir das formas de onde fornecidas.

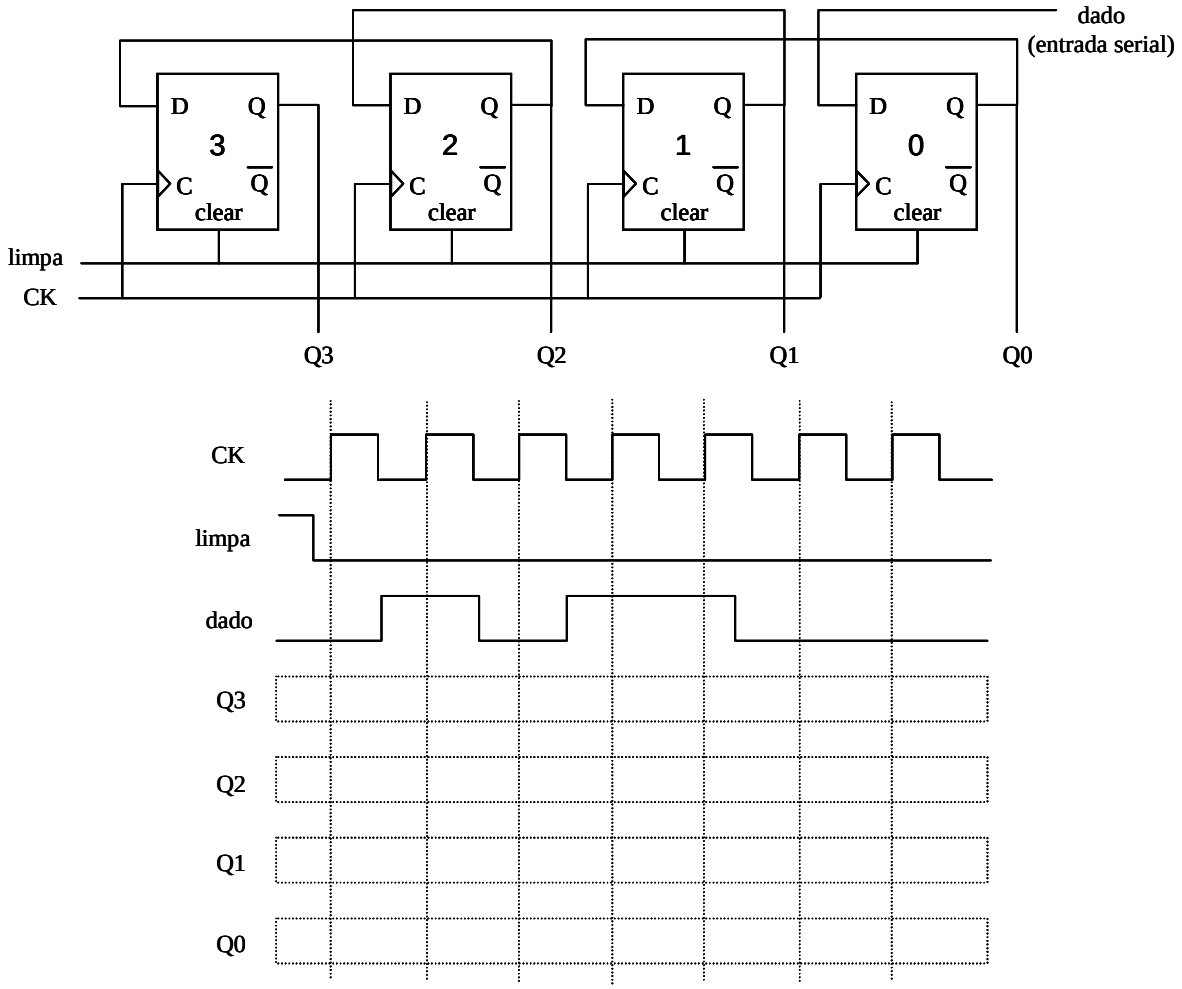

Figura 5.6 - Registrador de deslocamento à esquerda de 4 bits (com reset assíncrono): exemplo de funcionamento.

5.1.3 Registrador de deslocamento com sinal de carga paralela

Um registrador muito útil é aquele que, além de permitir a carga paralela por meio de sinal de carga, ainda permite deslocamentos à direita e à esquerda. Isso é possível se, na entrada de cada flip-flop houver um seletor capaz de escolher de onde vem o dado a ser armazenado no flip-flop corrente: de uma fonte externa (no caso de uma carga paralela), da direita, da esquerda (no caso de deslocamento) ou do próprio flip-flop (no caso de simplesmente se querer manter o conteúdo inalterado). Um tal registrador é mostrado na figura 5.7.

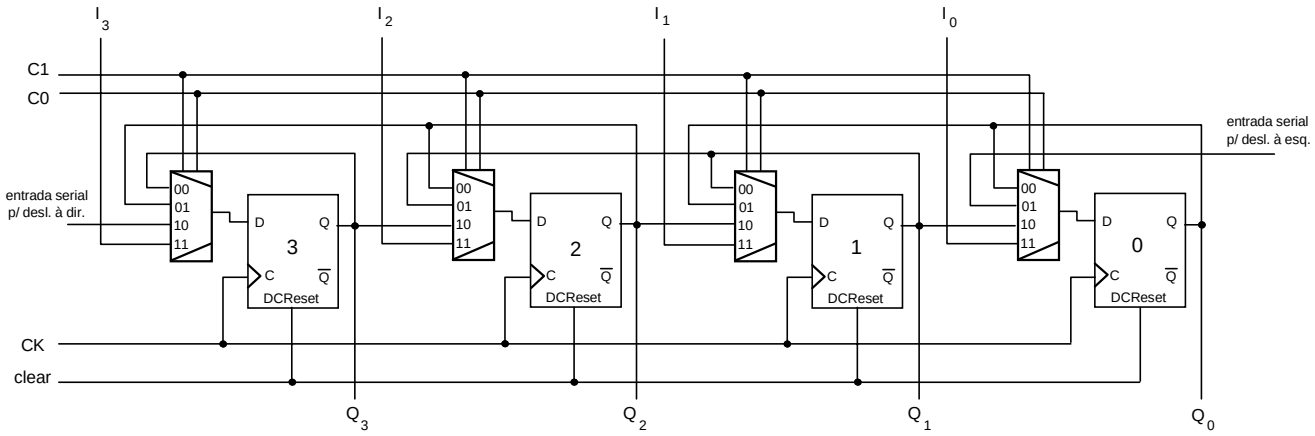

Figura 5.7 - Um registrador-deslocador de 4 bits com sinal de carga e reset assíncrono.

As operações possíveis para o registrador-deslocador (também conhecido como shift-register) da figura 5.7 são:

1. Carga paralela;
2. Mantém conteúdo;
3. Zera o conteúdo (fazendo-se clear=1);
4. Desloca à direita e desloca à esquerda.

E seu funcionamento se dá como segue:

Se o sinal clear=1, Q3=Q2=Q1=Q0=0;

Caso contrário, vale a tabela verdade a seguir

CK	C1	C0	**operação**
$\neq\uparrow$	X	X	mantém conteúdo
$\uparrow$	0	0	mantém conteúdo
$\uparrow$	0	1	desloca à esquerda (shift left)
$\uparrow$	1	0	desloca à direita (shift right)
$\uparrow$	1	1	carga paralela

5.1.4 Registrador Contador Assíncrono

Um contador (ou incrementador) é um registrador que "conta" em binário. Ou seja, a cada sinal de relógio, o conteúdo do registrador é incrementado de uma unidade. Logo, um registrador contador de 4 bits é capaz de contar de 0 (0000) até 15 (1111).

Primeiramente, vejamos como funciona um contador de um bit.

Exemplo 5.4: traçar a forma de onda de Q para o circuito a seguir.

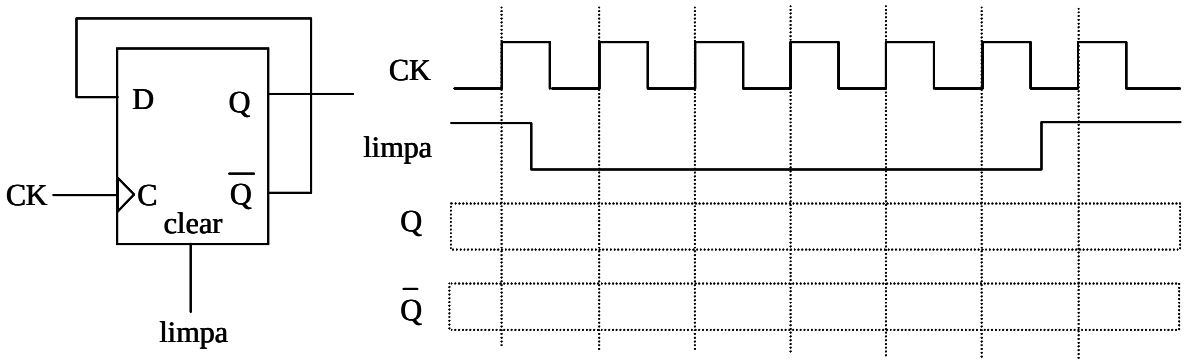

Figura 5.8 - Contador de um bit (com reset assíncrono).

Um circuito contador de mais bits possui uma conexão entre cada flip-flop vizinho, de modo que cada flip-flop de maior ordem é responsável pela ordem de incremento de seu vizinho de menor ordem.

Exemplo 5.5: encontrar as formas de onda para as saídas dos flip-flops do registrador abaixo.

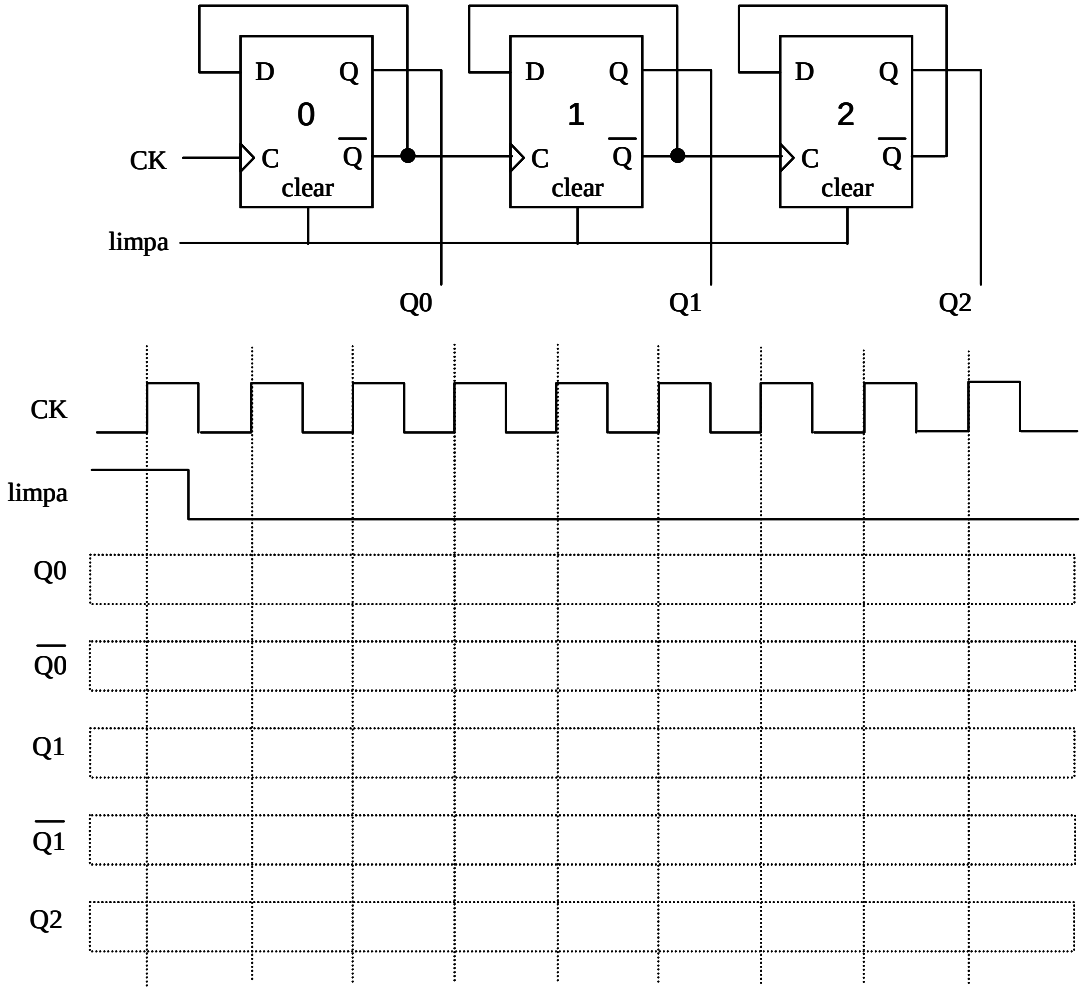

Figura 5.9 - Contador assíncrono de 3 bits (com reset assíncrono).

5.2 Memórias

Na seção anterior, vimos como são construídos diversos tipos de registradores. Apesar de serem muito rápidos, os registradores têm capacidade de armazenamento reduzidíssima: cada registrador é capaz de armazenar somente uma palavra por vez. Porém, nos sistemas digitais em geral, e particularmente nos computadores, grandes quantidades de informação (palavras) devem poder ser armazenadas. Para tanto, é necessário que o sistema digital possua um conjunto específico de circuitos que sejam mais apropriados ao armazenamento simultâneo de um grande número de palavras. Tais circuitos efetivamente existem e são genericamente denominados de **memórias**.

Do ponto de vista do funcionamento, as memórias podem permitir apenas a consulta (leitura) ao seu conteúdo ou podem permitir a consulta (leitura) e a modificação (escrita) de seu conteúdo. As memórias que só permitem a leitura são chamadas de **ROMs** (*Read-Only Memories*, que significa memórias de leitura apenas), enquanto que as memórias que permitem a leitura e a escrita são genericamente denominadas **RAMs** (*Random-Access Memories*, que significa memórias de acesso randômico).

O conteúdo das ROMs pode ser escrito (gravado) quando da fabricação ou mesmo após, por um usuário, que no caso pode ser o fabricante do computador, por exemplo. A característica principal é que uma vez gravadas as informações na ROM, estas não poderão ser modificados, mas somente consultadas (lidas). Já as memórias RAM possuem circuitos capazes de armazenar as informações binárias, as quais podem ser modificadas um número indeterminado de vezes.

Tanto as memórias ROM como as memórias RAM são fabricadas com a tecnologia CMOS, que aliás é a mesma tecnologia de fabricação dos microprocessadores. Porém, o que origina a diferença de funcionamento entre ROMs e RAMs é o tipo de circuitos utilizados. Na seção que segue, veremos detalhes de implementação das memórias RAM, as quais são utilizadas na implementação da **memória principal** dos computadores.

5.2.1 Memória de Acesso Randômico (RAM - Random-Access Memory)

Nesta seção iremos estudar a estrutura física das memórias RAM, isto é, os circuitos que as compõem.

Uma memória RAM é organizada como uma matriz de **2^n linhas** com **m bits** armazenados em cada linha, perfazendo um total de **2^n x m** bits. Em geral, n está entre 16 e 32, enquanto que m pode ser 1, 4, 8, 16 ou 32. A figura 5.10 ilustra essa organização matricial. Note que existem 2^n linhas, também chamadas posições. A cada posição é associado um endereço, iniciando pelo endereço 0 (zero). Portanto, são necessários n bits para decodificar os 2^n endereços existentes.

Suponha que se deseje fabricar num único **chip** (circuito integrado) uma memória RAM capaz de armazenar 2^n x m bits, seguindo a organização descrita anteriormente. A figura 5.11 mostra duas representações gráficas possíveis para um tal chip. Este chip deverá possuir n entradas de endereço (A_{n-1}, …, A_1,A_0), de modo a se poder selecionar uma (e somente uma) dentre todas as 2^n posições existentes na matriz. Este chip também deverá conter m entradas (I_{n-1}, …, I_1,I_0), e m saídas (O_{n-1}, …, O_1,O_0), de modo a permitir a leitura (=consulta) do conteúdo de uma das 2^n linhas ou a escrita (=gravação) de uma nova informação numa das 2^n linhas da matriz. Como existem duas operações possíveis sobre o conteúdo da matriz (leitura

e escrita), é natural que deva existir uma entrada de seleção de operação. Esta entrada será denominada RWS (*Read/Write Select*). Quando RWS=0, a operação a ser realizada será a **leitura** do conteúdo da posição cujo endereço está presente na entrada de endereços. O valor lido aparecerá na saída do chip. Quando RWS=1, a operação a ser realizada será a escrita da informação binária presente na entrada do chip na linha cujo endereço está presente na entrada de endereço. Por fim, deve existir um sinal de habilitação do chip como um todo (CS - *Chip Select*). Caso CS=0, o chip está desativado. Caso CS=1, o chip estará realizando a operação especificada pelo valor da entrada RWS.

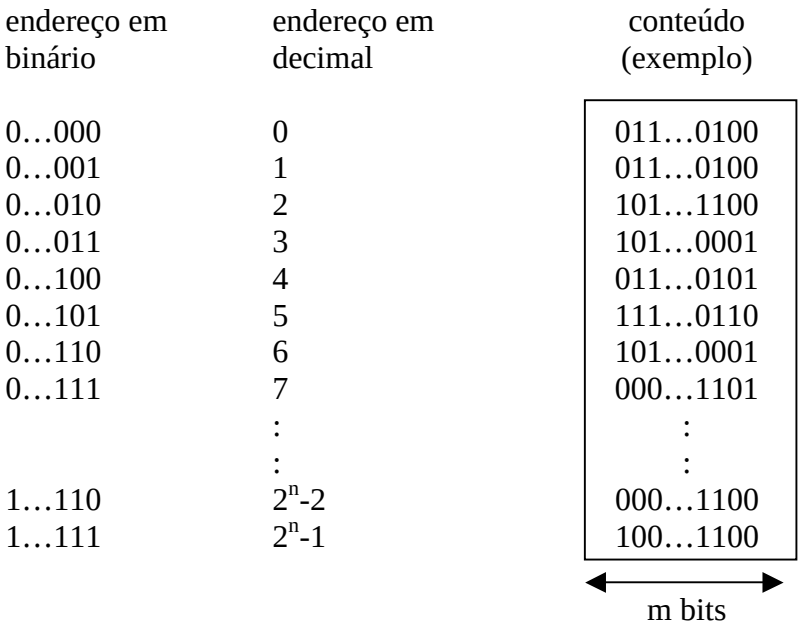

Figura 5.10 - Organização de uma memória RAM: linhas e endereços.

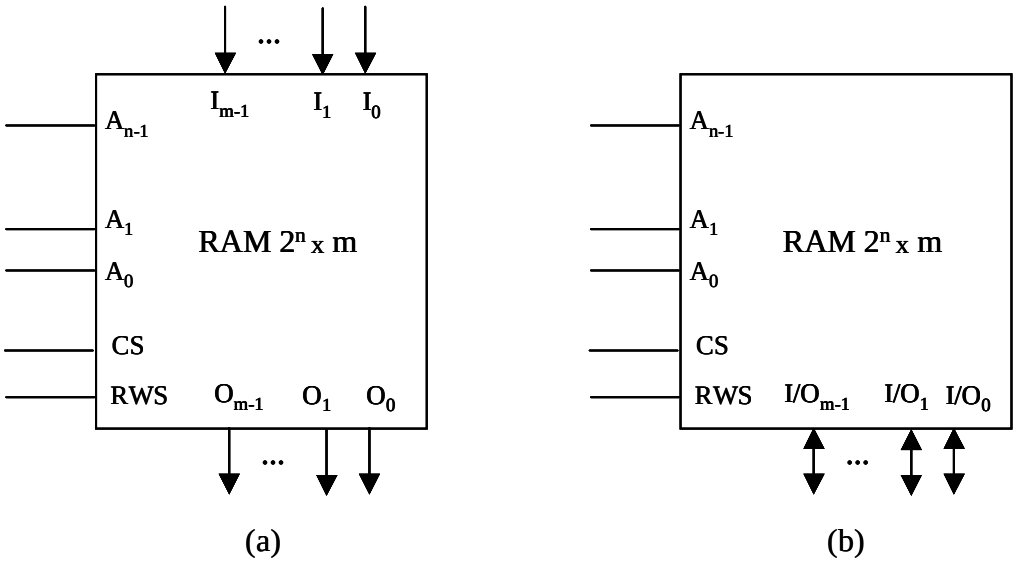

Figura 5.11 - Representações gráficas possíveis para um chip de memória RAM.

O sinal de habilitação do chip, CS, serve para o caso em que seja necessário mais de um chip para implementar a memória do computador.

Quando o número de bits a serem armazenados em cada posição de memória m for pequeno, o chip RAM poderá ter as entradas separadas das saídas, conforme representado pelo símbolo da figura 5.11a. Entretanto, o mais comum é que um mesmo conjunto de pinos do chip sirvam como entradas e como saídas, situação representada pelo símbolo da figura 5.11b. A função irá depender da operação, selecionada por meio do pino RWS. Esse uso compartilhado reduz o número de pinos do chip, tornando-o mais barato.

Do ponto de vista estrutural, uma memória RAM é organizada como uma matriz de elementos básicos de memória, *buffers* de entrada e saída e um decodificador de endereços. Como mostrado na figura 5.12, uma **célula de memória** (CM) pode ser representada simbolicamente como sendo constituída por um latch D controlado cuja entrada está conectada a saída de uma porta **E** e cuja saída se conecta à entrada de um *buffer*.[1] Quando o sinal de **seleção de linha** é igual a 1, o bit armazenado no latch passará pelo *buffer*, ficando disponível na **saída** da célula. Se o sinal de **habilitação de escrita** também valer 1, o valor presente na entrada será armazenado no latch. O sinal de **habilitação de escrita** serve como controle para o latch.

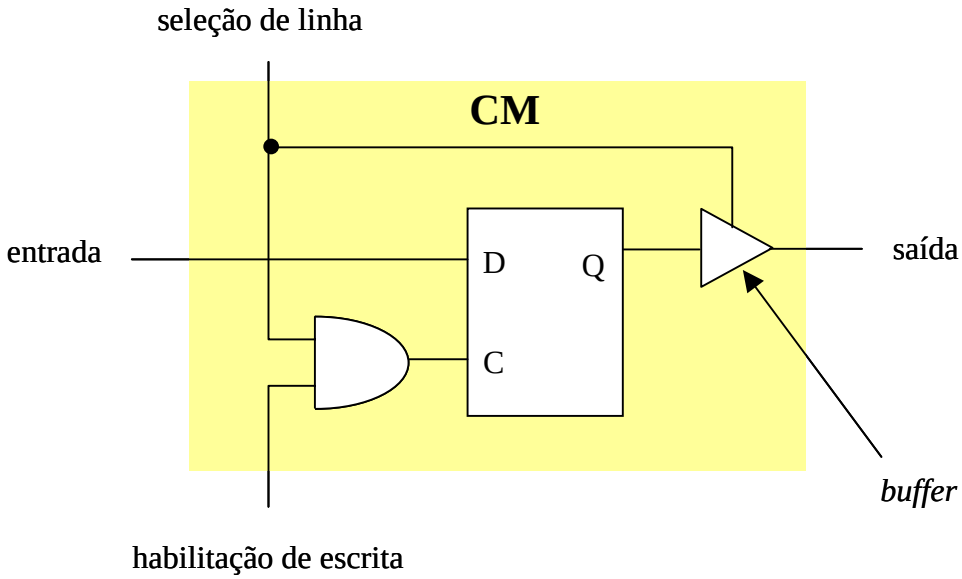

Figura 5.12 - Célula de memória (CM).

A figura 5.13 mostra um exemplo de memória RAM 4 x 4, constituída por 16 CMs. Para cada acesso à memória, o decodificador de endereços ativa o sinal de **seleção de linha** associado ao endereço aplicado as suas entradas, o que ativa todos os CMs da linha selecionada. Neste momento, se RWS=1 e CS=1, o novo conjunto de bits será armazenado nas CMs da linha selecionada. Por outro lado, se RWS=0 e CS=1, os bits que estão armazenados na linha selecionada passarão pelos *buffers* de saída das respectivas CMs e pelos *buffers* de entrada/saída do chip.

A organização da memória RAM implica em restrições de tempo nas operações de escrita e leitura. Por exemplo, como o caminho crítico da entrada a saída passa pelo decodificador, as entradas de endereço devem estar estáveis antes de quaisquer outros sinais. Isto significa que durante o ciclo de leitura mostrado pela figura 5.14 as entradas de endereço

[1] Um *buffer* (as vezes chamado *driver*) pode ter duas funções: reforçar o sinal lógico (o que é necessário quando existem muitas portas conectadas a uma mesma saída ou quando o sinal deve percorrrer um fio muito longo) ou servir como uma chave eletrônica que isola fisicamente a saída da entrada (o que serve para disciplinar o acesso de vários sinais a um mesmo fio ou barramento).

deverão ser fornecidas em t_0, seguidas por CS em t_1. Assim, os dados da memória estarão disponíveis somente em t_2. O atraso t_2-t_0 é denominado **tempo de acesso à memória** (*memory-access time*), enquanto que o tempo t_2-t_1 é denominado **tempo de habilitação da saída** (*output-enable time*). Note que após os valores das entradas de endereço terem sido modificadas em t_3, os dados ainda estarão disponíveis até t_5. O intervalo t_5-t_3 é denominado **tempo de manutenção da saída** (*output-hold time*). Já o intervalo t_5-t_4 é denominado **tempo de desabilitação da saída** (*output-disable time*). Como o caminho entre as entradas de endereço e as saídas é maior do que o caminho entre CS até as saídas, o **tempo de acesso** determina a validade dos dados sempre que o endereço e CS forem aplicados ao mesmo tempo. Por outro lado, se o endereço e CS deixarem de ser válidos (CS=0, no caso), o **tempo de desabilitação** determinará a validade dos dados.

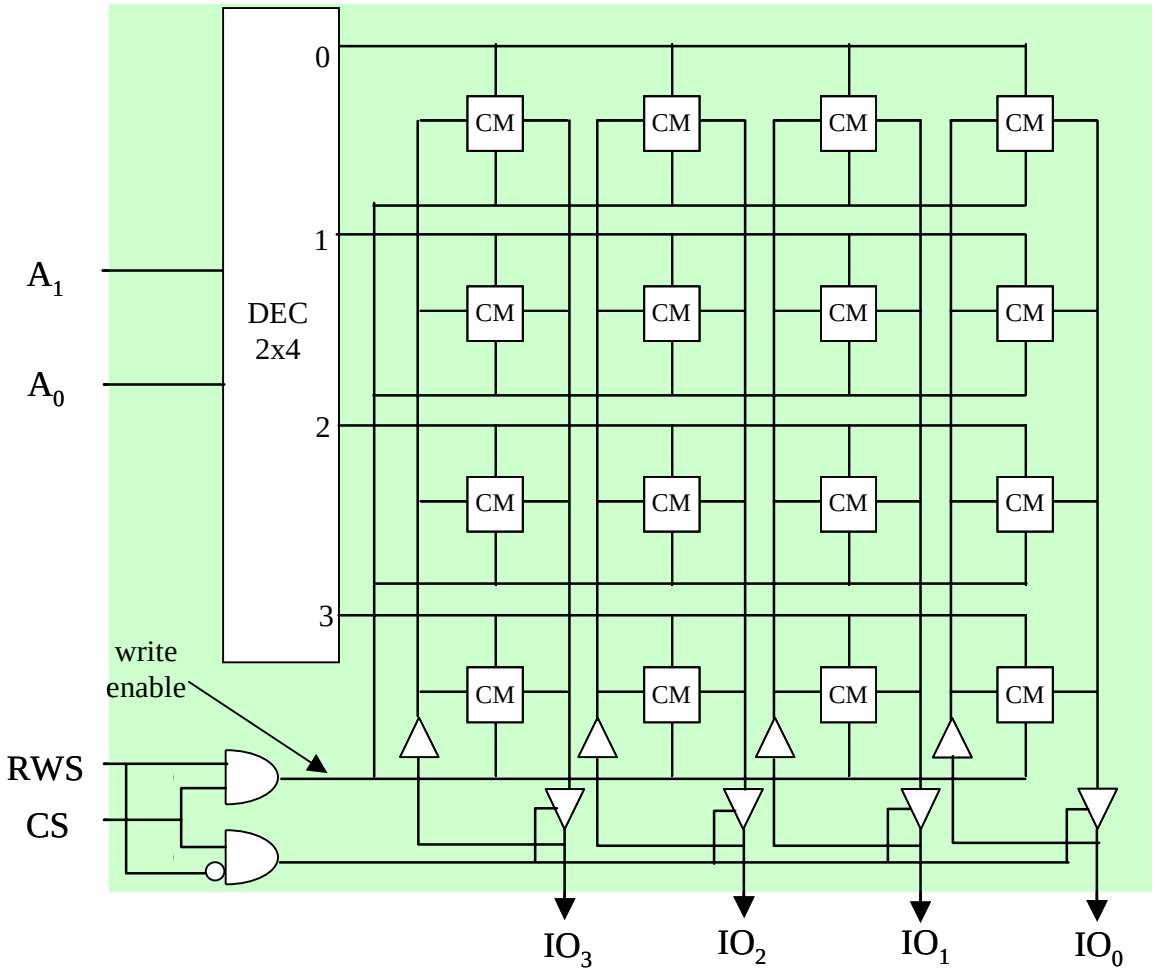

Figura 5.13 - Diagrama de blocos de uma memória RAM

A figura 5.15 mostra as restrições temporais para o caso de um ciclo de escrita numa memória RAM. No exemplo, foi assumido que CS e RWS foram aplicados simultaneamente, no instante t_1. Como o atraso entre o endereço e a saída é maior do que o atraso entre CS ou RWS e a saída, o endereço deve ser aplicado algum tempo antes, como por exemplo em t_0. O atraso t_1-t_0 é denominado **tempo de preparação do endereço** (*address setup time*). Como cada CM é feita a partir de um latch D controlado com CS fazendo o papel de controle, cada bit do dado na borda de descida de CS (t_3) ficará armazenado no respectivo latch. Entretanto, é necessário que o dado esteja estável por algum tempo antes e depois da borda de descida de CS para garantir a escrita. Na figura 5.15 esses tempos são anotados como **tempo de**

preparação do dado (*data setup time*) e **tempo de manutenção do dado** (*data hold time*), sendo definitos respectivamente como t_3-t_2 e t_4-t_3.

Também é importante ressaltar que CS ou RWS devem ter uma duração igual ou maior que o intervalo de tempo t_3-t_1, o qual é denominado **duração do pulso de escrita** (*write-pulse width*). Além disso, o endereço deve estar válido por algum tempo após a borda de descida de RWS ou CS. Este tempo é chamado **tempo de manutenção do endereço** (*address-hold time*), e é definido como t_5-t_3.

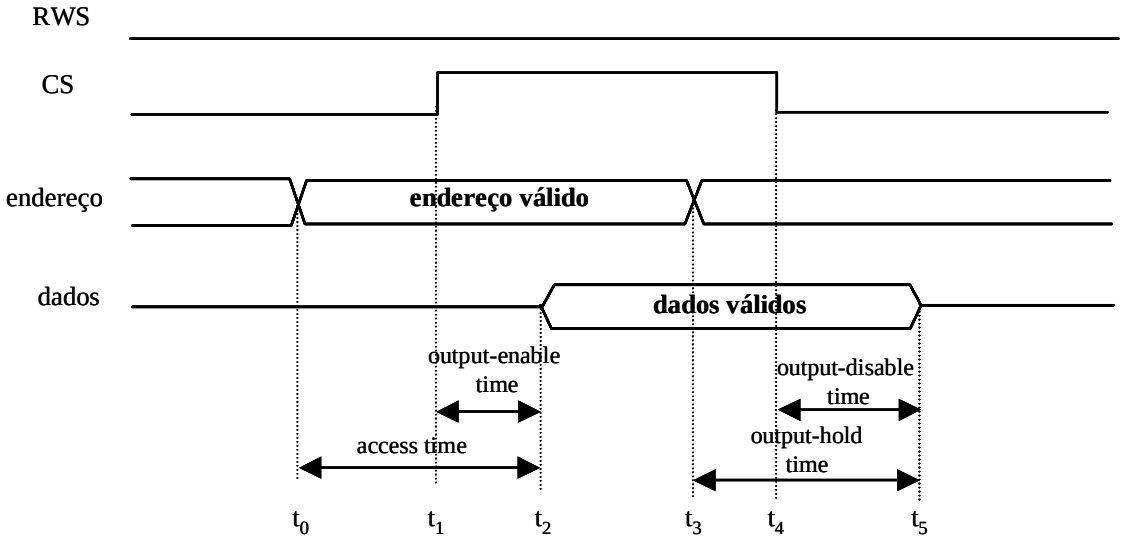

Figura 5.14 - Ciclo de leitura em uma memória RAM.

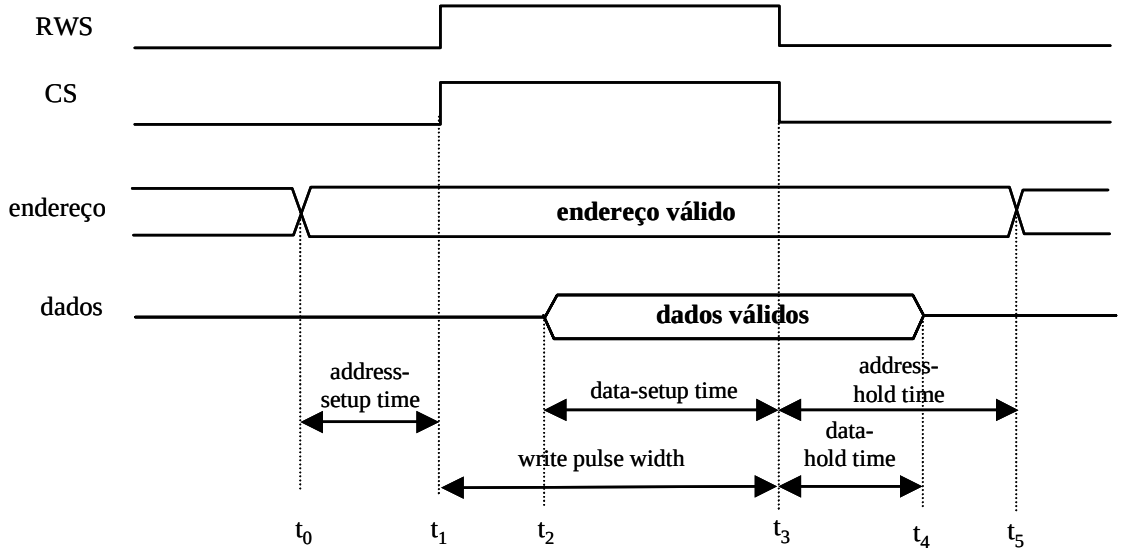

Figura 5.15 - Ciclo de escrita em uma memória RAM.

Apesar da CM ter sido representada como sendo constituída por um latch D e duas portas, na prática sua fabricação pode ser levada a cabo com estruturas mais simples, que utilizam menos transistores. A forma de implementação de CMs leva a classificação das memórias RAM em **estáticas** e **dinâmicas**. No caso da RAM estática (conhecida por **SRAM**, *static RAM*), a CM é feita com 6 transistores, onde quatro deles formam dois inversores

conectados em laço de realimentação, fazendo o papel do latch D. No lugar da porta **E** e do *buffer* de saída há um transistor (para cada um, no caso), o qual serve como chave de liga-desliga. A memória SRAM é capaz de manter seu conteúdo por tempo indeterminado, desde que este a alimentação não seja interrompida. No caso da memória dinâmica (conhecida por **DRAM**, *dynamic RAM*), cada CM é implementada com somente um transistor. A desvantagem deste tipo de RAM é que o conteúdo da CM é perdido após a operação de leitura, devendo ser reescrito. Para piorar, devido às imperfeições do processo de fabricação, o conteúdo da CM só se mantém por um curto período de tempo. Estes dois problemas são contornados pela utilização de um mecanismo de *reflash* construído dentro da memória, o qual periodicamente reforça o conteúdo de cada linha de CMs. Durante a operação de *reflash*, as operações de leitura e escrita são suspensas, o que reduz o desempenho deste tipo de memória.

As memórias DRAM apresentam uma densidade muito grande, o que se traduz em maiores capacidades de armazenamento. Elas também apresentam um custo bem reduzido. Devido a estas duas características, as DRAMs são muito utilizadas no projeto de produtos eletrônicos.

Por outro lado, as memórias SRAM são mais caras e apresentam menores capacidades de armazenamento. Porém, são mais velozes do que as memórias DRAM, sendo portanto apropriadas para os casos em que a quantidade de dados a serem armazenados não é grande e uma velocidade maior de operação é necessária.

As memórias SRAM e DRAM são ditas memórias **voláteis**, pois, uma vez interrompido o fornecimento de energia, elas perdem seu conteúdo. Já as memórias ROM são ditas **não-voláteis**, pois não perdem seu conteúdo quando o fornecimento de energia é interrompido. As memórias ROM tem uma organização semelhante às memórias RAM (matriz de CMs com decodificador de endereço e buffers de saída). Porém, a CM de uma ROM é bem mais simples, normalmente constituída por um único transistor ou diodo, o qual pode ser configurado uma vez para permitir o acesso ao 0 lógico (=0V) ou ao 1 lógico (de 5V a 1,5 V, conforme a tecnologia). Nos computadores, as memórias ROM servem para armazenar todas as configurações básicas que jamais serão alteradas, como por exemplo, as rotinas de entrada e saída (denominadas de ROM-BIOS, nos computadores tipo PC).

Bibliografia Suplementar

[1] GAJSKI, Daniel D. **Principles of Digital Design**, New Jersey: Prentice Hall, 1997 (ISBN 0-13-301144-5)

[2] MANO, M. Morris; **Computer Engineering: Hardware Design.** New Jersey: Prentice Hall, 1988 (ISBN 0-13-162926-3)

[3] BROWN, Stephen; VRANESIC, Zvonko. **Fundamentals of Digital Logic with VHDL Design**. McGraw-Hill Higher Education (a McGraw-Hill Company), 2000 (http://www.mhhe.com/engcs/electrical/brownvranesic)