

Memória RAM

Vamos agora desenvolver uma célula de memória completa, chamada célula binária (*binary cell*), com base no flip-flop. Quando uma célula é selecionada e no modo de "ler" (*read*), o valor atual do seu flip-flop subjacente vai ser transferido para a linha de saída da célula. Quando a célula é selecionada e no modo "escrever" (*write*), um sinal de dados de entrada irá determinar o valor lembrado pelo flip-flop. Um circuito completo é mostrado na Figura 12.34. O design fundamental desta célula binária é baseado no flip-flop RS, embora existam algumas diferenças significativas. Para começar, a célula tem três entradas e uma única saída. As entradas são rotuladas "Select", "Read/Write" e "Input". A linha de saída é rotulada "Output".

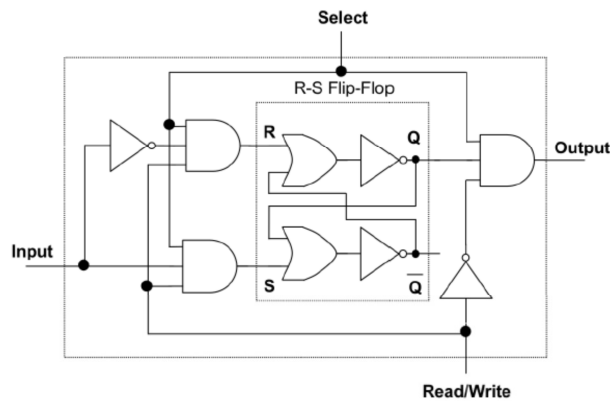


Figure 12.34: A Binary cell (BC) for RAM memory

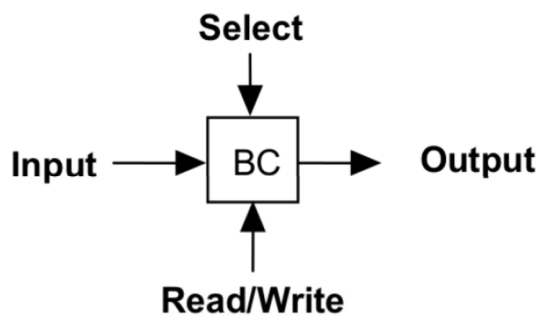


Figure 12.35: A Binary Cell – encapsulated view

A entrada "select" é usada para acessar o celular, seja para leitura ou escrita. Quando a linha "select" é alta, "1", uma operação de memória pode ser realizada nesta célula. Quando a linha "select" da célula binária é baixa, "0", o conteúdo da célula não são atualmente de interesse - isto é, no momento presente, a célula não está sendo lida ou gravada. Podemos ver como é dado este poder ao "select", observando que as entradas e a saída do flip-flop RS subjacente são encaminhadas às portas AND e que "select" é uma das entradas para cada uma dessas portas. Assim, se "select" é baixo, as entradas para o RS flip-flop ficarão baixas (o que significa que o seu valor armazenado não vai mudar) e a saída produzida pela célula vai ser baixa (independentemente de se o bit real armazenado nos flip-flops é "0" ou "1").

A próxima entrada a examinar é "Read/Write". Um relógio do sistema (*clock*) irá conduzir esta entrada. Como no caso com os flip-flop RS sincronizados, um baixo, "0", significará "ler" (*read*), enquanto um alto, "1", significará "escrever" (*write*). Durante a fase de leitura não será possível escrever na célula. Da mesma forma, durante a fase de escrita que não será possível ler o conteúdo da célula.

Assuma que a célula tenha sido selecionada (isto é, "select" é "1", significando que uma operação de acesso à memória é para ser realizada nesta célula.) Além disso, assumo que o valor de relógio (*clock*) na linha de "Read/Write" é baixo (forçando a negação de "Read/Write" para valor alto "1") indicando que o conteúdo da célula é para ser lido. Neste caso, o valor de

saída da célula binária irá depender apenas do valor Q do flip-flop. Se Q é baixo, a célula emite um "0", se Q é alta, a célula produz um "1". Isto ocorre porque a porta AND ligada à linha de saída da célula tem três entradas: "select", negação de "read/write", e Q , e tanto "select" e negação de "read/write" são atualmente altas, "1".

Como mencionado anteriormente, quando a célula está sendo lida o seu conteúdo não pode ser alterado. A razão para isto é que o mesmo valor baixo "0" na linha "read/write" que permite que a célula seja lida, é alimentada para as portas AND guardando as entradas do flip-flop. Assim, durante as leituras, as entradas para R e S são mantidas em valor baixo, "0", impedindo que o valor do flip-flop seja modificado. Quando a célula é selecionada e a linha "read/write" é definida como alta, "1", significando uma operação de "write", o valor colocado na célula dependerá unicamente do estado da linha "Input". A razão para isso é que as portas que guardam as entradas R e S do flip-flop vão ter suas entradas selecionadas em alta, "1": as entradas "select" e "read/write". Assim, se "Input" é alto, S (Set) receberá um valor alto, "1", e o flip-flop irá armazenar o valor "1". Se, por outro lado, "input" é baixo, "0", então R (reset) que recebe a versão de negação de "input" será alto "1" e o flip-flop irá resetar a "0". Note que ter uma versão de negação da linha de "input" sobre R é uma idéia inteligente, uma vez que impede que o flip-flop RS entre em seu estado inválido. (Lembre-se que no flip-flop RS se R e S são definidos como "1", ao mesmo tempo, o flip-flop entra em um estado estável, mas inválido.)

Vale ressaltar que durante as operações de escrita, a leitura é proibida. Isso é fácil de ver, uma vez que a porta AND que guarda a linha "Output" recebe uma de suas entradas a partir da negação do "read/write" que é mantida baixa, "0", durante operações de gravação. Dessa forma, a saída a partir da célula será sempre baixa, "0", durante as gravações, independentemente do valor real na linha Q .

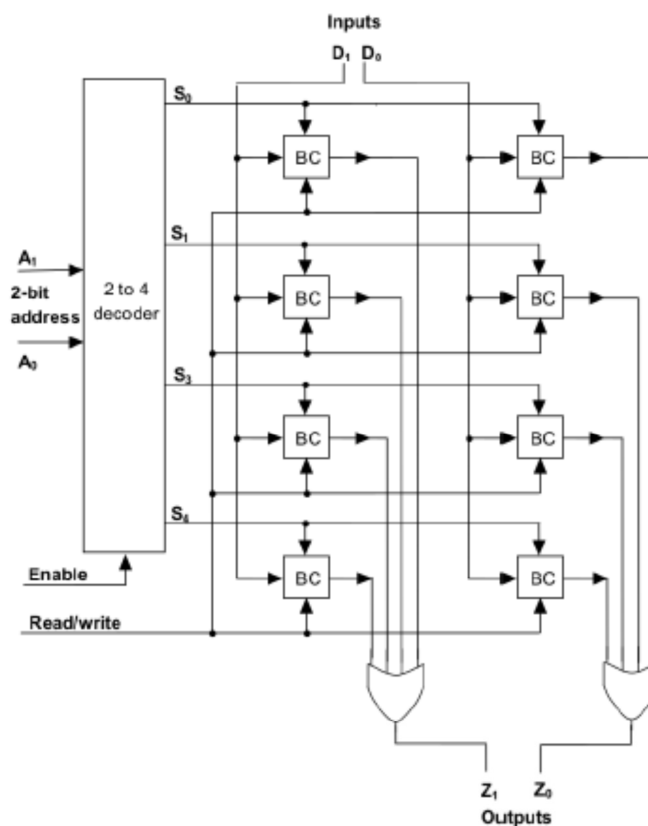
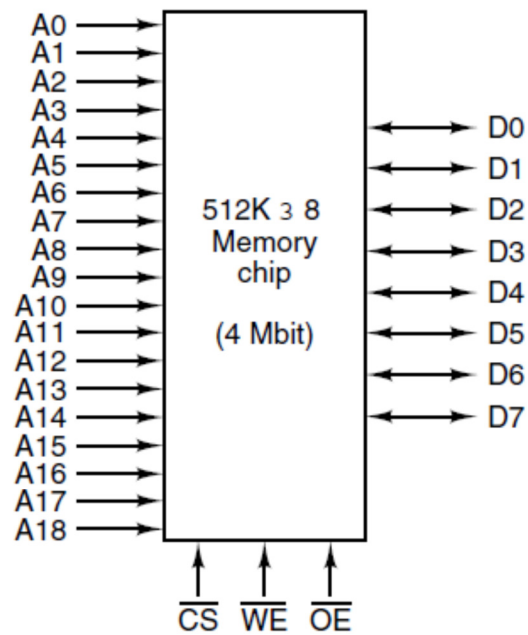


Figure 12.36: A 4 x 2 RAM memory

Exemplo de chip de memória



Memória Estática e Dinâmica

Um conceito importante sobre memória é a diferença entre memória estática e dinâmica. A memória dinâmica é implementada usando um capacitor e um transistor, e assim é simples e barato. No entanto, no capacitor há fuga de corrente, por isso é necessário recarregá-lo em todo ciclo de relógio (*clock*), fazendo da memória dinâmica uma memória mais lenta. A memória estática não tem de ser recarregada, de modo que é mais rápida, mas requer, pelo menos, 5 portas para ser implementada. Isso faz com que a memória estática seja mais rápida mas mais cara. Ambos os tipos de memória existem em um computador.

<http://watson.latech.edu/book/circuits/circuitsMicrocomputer3.html>
Structured Computer Organization, Fifth Edition, Andrew S. Tanenbaum
Digital circuit projects - Charles W. Kann III