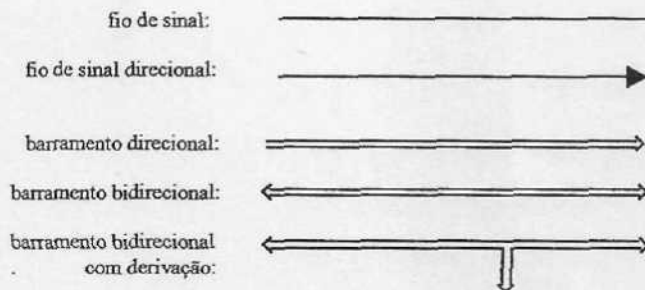


O barramento de dados interno pode ser constituído de mais de um trecho de barramento de dados. Para se comunicar com os dispositivos externos, os dados são colocados em um registrador de onde vão para o barramento de dados externo, ou que recebe dados do barramento externo. De certa forma, o barramento de dados externo é uma continuação do interno. A conexão é feita através de pinos do microprocessador.

O endereço de memória ou de dispositivo entrada/saída é gerado no microprocessador (para isso há necessidade de interligar registradores e outras unidades internas relacionadas com a geração de endereço) e colocado num registrador de endereço, de onde sai para o barramento de endereço externo através de pinos do microprocessador.

As linhas de controle internas saem da Unidade de Controle e vão para cada parte do microprocessador onde devem controlar alguma ação (são especializadas); as que devem controlar dispositivos externos (por exemplo, informar se o dispositivo deve receber algum dado do microprocessador ou enviar um dado para ele são ligados a pinos do microprocessador, para daí darem origem às linhas de controle externas.

Nos esquemas de ligações de circuitos eletrônicos, um fio é representado por uma linha; quando é necessário informar o sentido em que o fio é percorrido pelo sinal, é utilizada uma seta, que pode vir no extremo; um conjunto de fios que sirva para transportar um conjunto de sinais (por exemplo, 32 bits) é representado por uma seta grossa. O sentido da seta indica a direção de transmissão do sinal; quando bidirecional, as setas aparecem em extremos opostos (sempre indicando saída):



Note que no barramento bidirecional a indicação de direção de sinal só aparece na saída; isso não significa que, em cada um dos extremos, o sinal só circule em uma direção.

Na arquitetura a três barramentos, os três barramentos externos interligam o microprocessador com cada um dos periféricos. Cada periférico tem um controlador próprio, de modo que a comunicação, na verdade, é feita entre o microprocessador e cada controlador, e cada controlador se comunica com seu periférico (dispositivo de entrada/saída ou memória).

Isto se deve ao fato de que o microprocessador se comunica sempre utilizando os mesmos sinais digitais (grupos de bits), enquanto que cada periférico tem seus sinais típicos. Cada controlador faz a "tradução" entre os sinais digitais padronizados do microprocessador e os sinais típicos do dispositivo controlado (que podem ser de diversos tipos).

Além disso, o controlador efetua as ações localizadas de controle do dispositivo, de modo que o microprocessador só precisa enviar a informação da ação que deve ser executada e o controlador se encarrega de controlar os detalhes da execução da ação. Dessa forma, o controlador é um auxiliar do microprocessador, tanto para "tradução" dos sinais como para efetuar o controle do periférico.

Com essa forma de ligação, o microprocessador controla os dispositivos periféricos (memórias, entrada/saída), troca informações com eles e intermedia a comunicação entre os periféricos, quando esta for necessária. Ou seja, se for o caso de ser necessário fazer com que um dispositivo periférico envie dados para outro, o microprocessador, que é quem vai controlar isso, faz com que o primeiro lhe envie os dados e em seguida ele envia os dados para o segundo dispositivo.

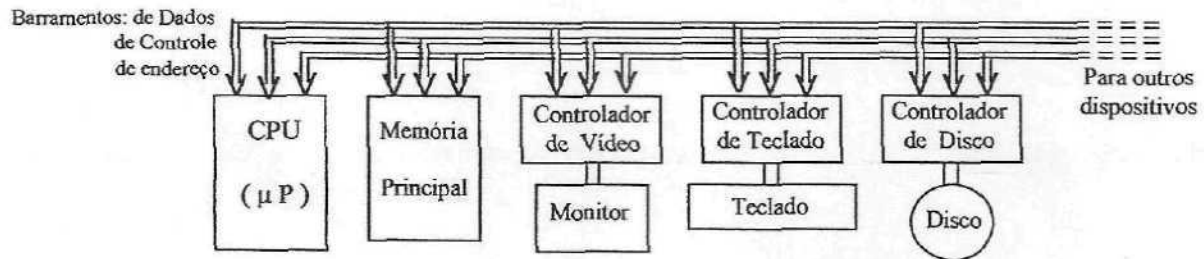
Arquitetura a três barramentos (continuação)

R - 1/8

UFF - TET - Microprocessadores - Prof Luiz Pinto de Carvalho

Uma exceção a esse esquema será vista mais adiante; é quando o microprocessador passa o controle dessa operação de transferência de dados entre memória e dispositivo de entrada/saída para o Controlador de DMA (Direct Memory Access, ou Acesso Direto à Memória). Nesse caso, o controlador de DMA passa a substituir o microprocessador em sua função de controlar a transferência de dados, enquanto o microprocessador faz outras coisas.

A interligação do microprocessador com os periféricos por meio dos barramentos de dados, endereço e controle costuma ser representada assim:



ORIGINALS MIT

ORIGINALS MIT COPY

Porta AND (E):

Símbolo:

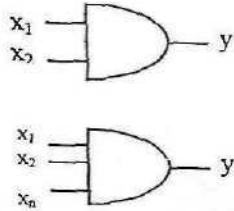


Tabela verdade:

x_1	x_2	y
0	0	0
0	1	0
1	0	0
1	1	1

Porta OR (OU):

Símbolo:

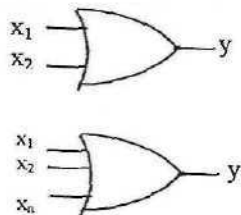


Tabela verdade:

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

Inversor:

Símbolo:

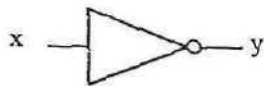


Tabela verdade:

x	y
0	1
1	0

Porta NAND (Não-E):

Símbolo:

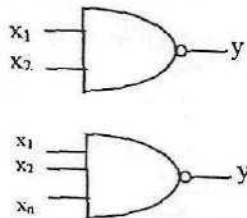


Tabela verdade:

x_1	x_2	y
0	0	1
0	1	1
1	0	1
1	1	0

Porta NOR (Não-OU):

Símbolo:

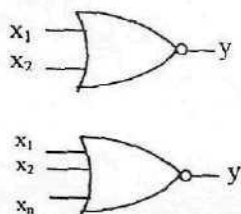


Tabela verdade:

x_1	x_2	y
0	0	1
0	1	0
1	0	0
1	1	0

Portas lógicas (continuação)

Exclusive-OR (X-OR, OU Exclusivo):

Símbolo:

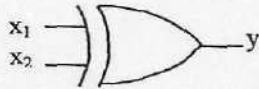


Tabela verdade:

x ₁	x ₂	y
0	0	0
0	1	1
1	0	1
1	1	0

Exclusive-NOR (X-NOR, Não-OU Exclusivo)

Símbolo:

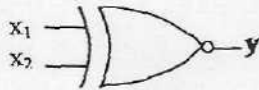
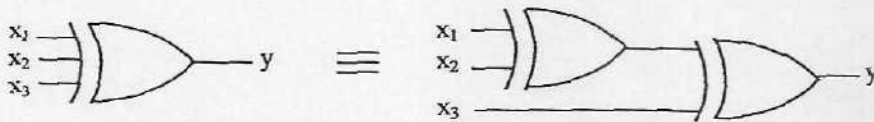


Tabela verdade:

x ₁	x ₂	y
0	0	1
0	1	0
1	0	0
1	1	1

Generalização do Exclusive OR (OU Exclusivo) para mais de duas entradas:



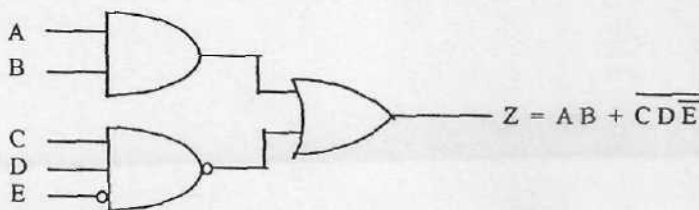
Portas com mais de duas entradas (por exemplo, 3)

Tabela verdade:

x ₁	x ₂	x ₃	y (generalização)				
			AND	OR	NAND	NOR	Ex-OR
0	0	0	0	0	1	1	0
0	0	1	0	1	1	0	1
0	1	0	0	1	1	0	1
0	1	1	0	1	1	0	0
1	0	0	0	1	1	0	1
1	0	1	0	1	1	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	0	0	1

As portas podem ser combinadas (de várias formas) para formar qualquer função lógica.

Exemplo:



ORIGINALS NIT COPY
 XEROX
 NIT. COPY LTDA. ME
 ORIGINALS NIT COPY

Portas lógicas e outros circuitos lógicos ("flip-flop", combinações de portas lógicas, circuitos mais complexos, como uma ALU completa ou outros sistemas ainda mais complexo) são construídos na pastilha de semicondutor usando um dos tipos de transistores vistos, com alguns aperfeiçoamentos em alguns casos.

Conforme o tipo de dispositivo (transistor) utilizado, diz-se que o Circuito Integrado pertence a uma família de circuitos lógicos. Principais famílias:

- TTL - Transistor-Transistor-Logic (lógica transistor-transistor - transistores bipolares);
- IIL - Integrated Injection Lógica (lógica de injeção integrada - transistores bipolares);
- ECL - Emitter Coupled Logic (lógica com acoplamento pelo emissor - transistores bipolares);
- PMOS - P-channel MOS (MOS canal P - transistores MOS, canal p);
- NMOS - N-channel MOS (MOS canal N - transistores MOS, canal n);
- CMOS - Complementary MOS (MOS Complementar - par complementar de transistores MOS, canal n e p).

Desses, os mais utilizados atualmente são TTL e CMOS.

Características básicas:

TTL: circuitos mais rápidos, consumo maior; usado em circuitos não muito grandes que precisam ser rápidos;

CMOS: menor consumo, porém mais lento; usado em circuitos muito grandes (microprocessadores) para evitar consumo excessivo; paga-se o preço de serem mais lentos.

Família TTL

A família TTL tem algumas variantes, resultando em "séries" com diferentes características, algumas com circuitos mais rápidos, outras com circuitos mais lentos; os mais rápidos tendem a consumir maior potência porque, para serem mais rápidos, devem movimentar as cargas mais rapidamente; maior movimento de carga significa maior corrente.

Alguns "truques" procuram melhorar a rapidez sem aumentar tanto o consumo, como o TTL Schottky. No Schottky, cada transistor inclui um "diodo Schottky" que serve para limitar a tensão entre base e coletor. Quando a tensão de base cresce o suficiente para a corrente de coletor atingir o máximo ("saturação" do transistor), há um acúmulo de cargas na junção; quando a tensão de base voltar a diminuir, a carga se escoar, mas isso toma tempo (atraso), devido ao acúmulo (o que torna o circuito mais lento). Evitando-se tensão excessiva entre base e coletor, evita-se a saturação e, portanto, o acúmulo de carga.

O TTL Schottky é mais rápido sem que isso implique maior consumo; pode ser feito ainda mais rápido com maior consumo, ou menos rápido, com menor consumo.

Deseja-se menor atraso e menor potência dissipada, mas quando um diminui, o outro tende a aumentar. Uma forma de se estimar a qualidade do circuito é considerando simultaneamente velocidade e consumo. Isso pode ser feito considerando-se o produto atraso X potência dissipada.

Na tabela abaixo, são mostrados exemplos de séries TTL (apenas como exemplo; novos dispositivos surgem constantemente, com tempos de retardo menores); para cada exemplo, a tabela mostra o atraso de propagação, em ns (nanosegundos, 10^{-9} s) e a dissipação de potência, em mW (miliwatts). É mostrado também o produto do atraso pela dissipação. Quanto menor o produto, melhor; o ideal seria mínimo atraso e mínima dissipação; para um mesmo produto, pode-se escolher entre uma série com menor dissipação ou menor atraso.

Séries	Tipo de transistor e potência	Atraso de propagação (ns)	Dissipação de potência (mW)	Produto atraso X potência (pJ)
54LS/74LS	Schottky, baixa potência	9,5	2	19
54L/74L	Comum, baixa potência	33	1	33
54S/74S	Schottky, potência normal	3	19	57
54/74	Comum, potência normal	10	10	100
54H/74H	Comum, alta potência	6	22	132

2025
ORIGINALS MIT CC
XEROX
NIT. COPY LTDA. - HE
ORIGINALS MIT COPY

Na coluna "Séries", os números e letras são a parte inicial (prefixo) da designação de cada tipo de Circuito Integrado, podendo ser 74 ou 54, seguido ou não das letras mostradas; cada circuito pode ser uma coleção de portas ou um circuito mais complexo, e é designado por um número formado pelo prefixo seguido de um número seqüencial.

Exemplos: 7400: 4 portas NAND de 2 entradas cada; 7404: 6 inversores; 7411: 3 portas AND de 3 entradas cada; 7432: 4 portas OR de 2 entradas cada; 7486: 4 portas Ex-OR; 74153: 2 multiplexadores 4/1; 74154: decodificador 4/16; 74181: ULA.

No prefixo, 74 corresponde à versão comercial e 54 à versão militar de cada circuito. Existem CI equivalentes (que são do mesmo tipo de porta) nas duas versões; fazem a mesma coisa; a diferença é que a versão militar é projetada para funcionar em condições ambientais mais extremas. Por isso, são mais confiáveis e custam mais caro.

As letras designam a série: sem letra, é o circuito mais simples e de potência normal; L, de "Low", é baixa potência; H, de "High", é alta potência; S é "Schottky".

A tabela está ordenada pelo produto atraso X potência, em ordem decrescente; nota-se que os tipos Schottky de baixa potência são os que têm o menor produto atraso X potência; têm, também, o menor consumo de potência, mas perdem em atraso tanto para o Schottky normal como, até mesmo, para o comum de alta potência. O mais rápido é o Schottky de potência normal, mas perde em consumo tanto para o comum de baixa potência como para o Schottky de baixa potência. Qual utilizar depende da finalidade.

Esta escolha não será feita pelo usuário do computador, mas sim, apenas, pelo projetista do "hardware" de um sistema (como um computador, um periférico, uma placa controladora). Para o usuário, essas noções podem ser úteis como base para tomadas de decisões, com conhecimento de causa, sobre equipamentos a comprar ou especificar para compra, ou para montar sistemas mais complexos (compostos de vários sistemas menores).

Fan-Out

A tensão de saída de uma porta (diferença de potencial entre a saída da porta e a terra) é fornecida à entrada de outro circuito (aplicada entre a entrada e a terra); essa tensão, conforme o sinal de saída corresponda a bit 0 ou 1, pode ser alta (5 V ou 3,3 V) ou baixa (0 V); em ambos os casos, a aplicação dessa tensão à entrada de um circuito faz com que circule uma corrente, que é fornecida pela saída da porta.

No caso de se precisar fornecer o mesmo sinal a vários circuitos (ligando-se as entradas desses vários circuitos, em paralelo, à saída da porta), a porta deverá fornecer corrente aos diversos circuitos; portanto, a porta deve ser capaz de fornecer uma corrente total maior.

Cada porta tem um limite de corrente que ela consegue fornecer; como cada circuito da mesma família consome aproximadamente a mesma corrente na entrada, isso corresponde a um limite na quantidade de outras portas da mesma família a que ela consegue fornecer sinal. Esse limite se chama "fan-out". Para cada circuito, o fabricante fornece o fan-out. Para a família TTL, o fan-out é da ordem de 10 para séries padrão e de alta potência, e da ordem de 20 para séries de baixa potência.

Margem de ruído

Ao ligarem-se um ou vários circuitos à saída de um certo circuito, a queda de tensão faz com que a tensão de saída se aproxime mais do valor do outro bit, podendo causar erro. Para evitar erros, cada circuito deve garantir que a tensão de saída para bit 0 e para bit 1 estão dentro de limites que permitem reconhecê-las sem dubiedade.

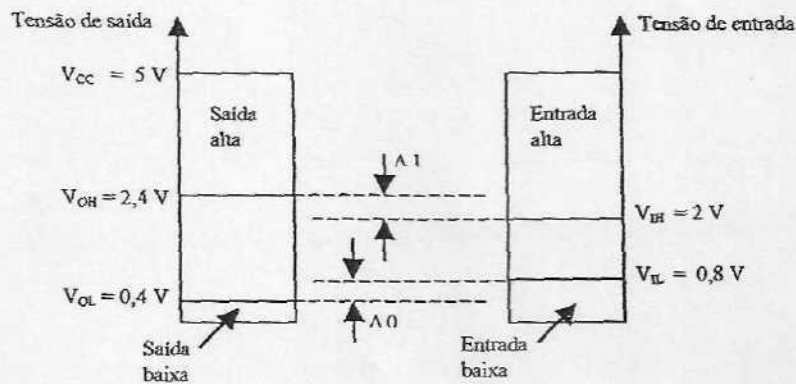
Cada circuito deve ser capaz de reconhecer corretamente, em sua entrada, um sinal cuja tensão esteja entre esses limites. Para evitar o efeito de imprecisões nas tensões de saída, os limites para reconhecimento de tensões à entrada de cada circuito deve ter uma folga em relação aos limites de saída.

Todo circuito está sujeito ao efeito de "ruídos" elétricos (interferências causadas por outros circuitos, outros sistemas ou fenômenos naturais); o ruído induzido em uma parte de um circuito é uma tensão aleatória que se soma ao valor do sinal.

Todo sistema é construído procurando-se minimizar esse efeito. Com isso, uma tensão resultante de ruído geralmente é bastante pequena, não perturbando o funcionamento de um circuito digital (circuitos digitais são menos susceptíveis a ruídos exatamente por trabalharem com apenas dois níveis de tensão; quando esses níveis são reconhecidos corretamente, informam corretamente qual é o bit, mesmo que não tenham o valor exato que se esperava).

Ainda assim, como sempre há combinação de alguma queda de tensão e algum ruído, a tensão pode ser reconhecida com erro. Este é um outro motivo para ter-se uma folga entre os limites de tensão à saída e à entrada de um circuito digital. Por esse motivo, essa folga é chamada "margem de ruído".

Cada família de CIs tem valores limites de tensão de saída e de entrada e, portanto, de margem de ruído. A figura abaixo mostra, como exemplo, valores para circuitos TTL padrão (54 ou 74) com alimentação de 5 V. Outros valores existem para outras família e outros valores de alimentação.



Na figura,

$V_{CC} = 5,0 \text{ V}$: tensão de alimentação

$V_{OH} = 2,4 \text{ V}$: tensão mínima de saída para bit 1

$V_{OL} = 0,4 \text{ V}$: tensão máxima de saída para bit 0

$V_{IH} = 2,0 \text{ V}$: tensão mínima que será reconhecida, à entrada, como bit 1

$V_{IL} = 0,8 \text{ V}$: tensão máxima que será reconhecida, à entrada, como bit 0

$$\Delta 1 = V_{OH} - V_{IH} = 2,4 \text{ V} - 2,0 \text{ V} = 0,4 \text{ V}$$

$$\Delta 0 = V_{IL} - V_{OL} = 0,8 \text{ V} - 0,4 \text{ V} = 0,4 \text{ V}$$

$$\Delta 1 = \Delta 0 : \text{margem de ruído}$$

Quando a tensão de entrada ficar acima de V_{IH} , o circuito a reconhece como bit 1; quando abaixo de V_{IL} , o circuito a reconhece como bit 0. Em ambos os casos, o circuito reage de acordo com o significado do bit, não importando o valor exato da tensão de entrada. Com isso, a saída desse circuito produzirá a tensão de saída adequada, que corresponderá a bit 1 ou 0 dependendo das entradas e da função do circuito. Essa característica faz com que cada circuito "regene" o valor do bit, de modo que não há acúmulo de erro. Essa é uma das razões pelas quais circuitos digitais reproduzem sinais com maior fidelidade e maior segurança.

Família CMOS

Fan-out

Corrente de entrada de uma porta: da ordem de 1 pA (10^{-12} A).

Capacidade de fornecer corrente à saída: da ordem de 1 mA (10^{-3} A).

Como a capacidade de fornecer corrente é 10^9 (1.000.000.000) de vezes maior que a corrente necessária à entrada, seria possível à saída de um circuito fornecer sinal a uma grande quantidade de portas (entradas - sendo todas da mesma família CMOS).

A limitação na quantidade de portas que se pode ligar à saída de um circuito digital, porém, provém de outro problema: quanto maior o número, maior o retardo de propagação introduzido. Isto porque, pela própria construção, em que a porta do transistor MOS funciona como se fosse uma das placas de um capacitor (a outra placa é o próprio canal de semicondutor), a entrada de um circuito possui uma certa capacitância; e uma capacitância, em paralelo com um circuito, introduz retardo porque armazena carga, e isso leva um certo tempo (tanto para armazenar como para desarmazenar). Quanto maior o número de entradas ligadas a uma saída, maior o retardo introduzido.

Procura-se conseguir fazer com que a capacitância introduzida por cada entrada seja feita cada vez menor, nos novos dispositivos desenvolvidos. Como exemplo, os valores de capacitância de entrada de circuitos CMOS, em uma certa época, foram da ordem de 5 a 10 pF (10^{-12} Farad); na mesma época, uma capacitância de cerca de 50 pF introduzia um retardo de 50 a 100 ns. Conseqüentemente, a ligação simultânea de 5 a 10 entradas em uma mesma saída era capaz de causar retardo de 50 a 100 ns, grande comparado ao que se consegue com circuitos TTL.

Esses valores mostram duas coisas:

- 1 - a capacitância de porta dos transistores MOS torna os circuitos CMOS mais lentos que os TTL;
- 2 - deve-se evitar ligar um número excessivo de entradas a uma mesma saída, em circuitos CMOS, para evitar que o circuito seja muito lento.

Outro fator a considerar é que a carga/descarga das capacitâncias faz com que passe uma corrente na porta que só circula quando há transição de valor de tensão. Com isto, essa corrente aumenta para maiores frequências de funcionamento do sistema. Por este motivo, um microprocessador CMOS consome maior potência ao funcionar com relógio de maior frequência.

É bom lembrar que estas são preocupações para o projetista de hardware; para o usuário de computadores, esses fatos interessam apenas para entender o que se passa (ou seja, para que o profissional possa fazer julgamentos e tomar opções com conhecimento de causa).

Margem de ruído

Para os circuitos CMOS com alimentação de 5 V, a margem de ruído é de cerca de 1 V. É menor para tensão de alimentação menor (como 3,3 V).

Combinação de CMOS e TTL em um mesmo circuito

Algumas características elétricas são diferentes nos circuitos CMOS e TTL, como visto. Isto faz com que seja mais fácil projetar um circuito totalmente em CMOS ou em TTL. No entanto, como, atualmente, ambos os tipos de circuito podem funcionar com as mesmas tensões de entrada e saída, é possível interligar CIs de tecnologia diferente; é também possível fabricar CIs construídos com ambas as tecnologias.

Características de circuitos digitais

Circuitos de saída

Uma característica importante é o circuito de saída. É quem fornece a tensão e, portanto, a corrente à entrada do circuito seguinte. Deve fornecer tensão com um de dois valores (alto ou baixo). Esses valores devem ser bem definidos, entre os limites determinados para bit 1 e bit 0, como visto acima, de acordo com cada família.

Uma forma de fazer isso é utilizar transistores que funcionem em corte ou saturação, ou utilizar um par de transistores, um ligando a saída à fonte de alimentação e o outro ligando a saída à terra. Existem também outras variantes, cada qual com suas vantagens e aplicações.

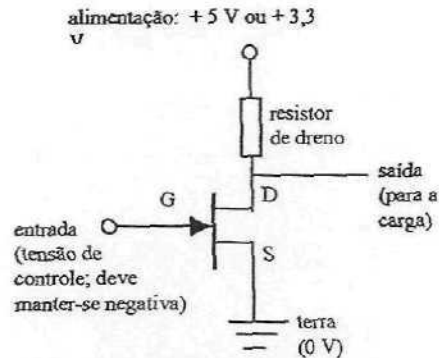
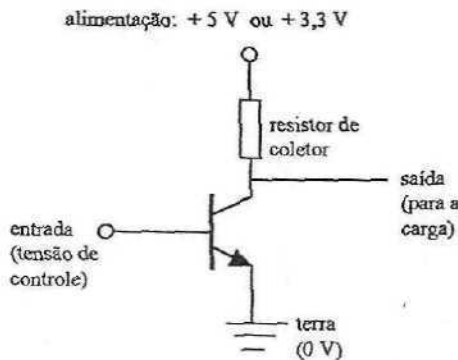
No primeiro caso, um resistor completa o circuito do coletor ou do dreno para a fonte, como será visto a seguir. No segundo caso, temos o circuito "totem pole" (totem, ou poste de totem, que será visto mais adiante).

Outras variantes também serão vistas adiante

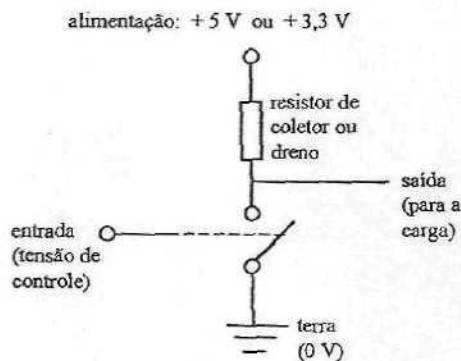
ORIGINALS NIT

XEROX
NIT COPY LTDA. ME
ORIGINALS NIT COPY

Circuito de saída com resistor no coletor ou no dreno



Neste tipo de circuito, o transistor pode ser entendido como uma chave de um polo, duas posições (chave "liga-desliga") comandada pela tensão de entrada, conectada a um resistor; na figura abaixo, a linha tracejada simboliza ação de abrir e fechar a chave causada pela tensão de controle:



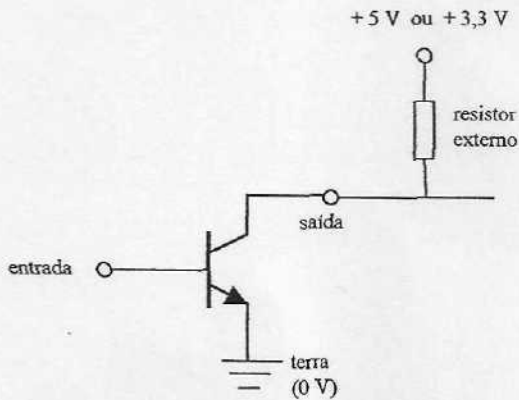
Com o transistor conduzindo (chave fechada), a saída está ligada diretamente à terra. A menos de resistências internas do próprio circuito, seria uma resistência zero para terra, o que garantiria tensão zero; nos circuitos reais, a tensão de saída fica próxima a zero, porém um pouco positiva, devido à passagem de corrente proveniente da carga pela resistência interna do circuito.

Quando a tensão de entrada causa a situação de não condução do transistor (chave aberta), a saída fica ligada à fonte pelo resistor, de modo que a tensão é igual à da fonte a menos da queda de tensão no resistor.

Não é possível substituir o resistor por uma ligação direta (um fio), pois, nesse caso, haveria curto circuito entre a fonte de alimentação e a terra quando o transistor conduzir. A presença do resistor e da conseqüente queda de tensão é um inconveniente que pode ser superado pelo uso de um par de transistores, característico do tipo de circuito de saída chamado "totem pole", visto mais abaixo.

Circuito de saída com coletor aberto

Em um circuito com resistor de coletor ou dreno, o transistor fornece a corrente de saída quando a tensão de saída é baixa (transistor conduzindo), mas é o resistor que deve fornecer a corrente adequada para a entrada do circuito (ou circuitos) seguinte(s) quando a tensão de saída é alta (transistor cortado), sendo ideal que o resistor tenha o valor exato para fornecer a corrente necessária sem causar queda de tensão excessiva. Como esses circuitos podem ser fabricados na forma de circuito integrado par uso em diversos sistemas diferentes, é difícil para o projetista do CI prever que corrente será solicitada da saída. O resistor colocado lá deve, então, ser uma "média", uma espécie de adivinhação. Uma forma de contornar isso é deixar que quem for utilizar o CI coloque o resistor externamente, no valor mais conveniente para o sistema. Isto é o que se chama circuito com "coletor aberto":

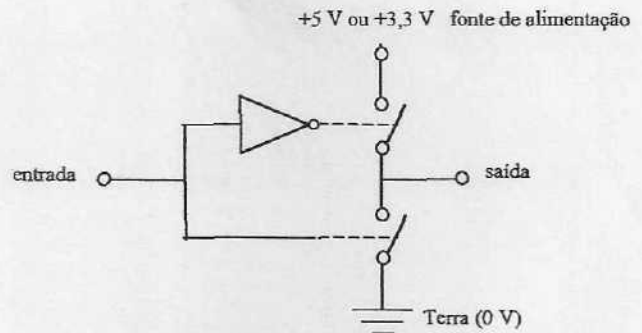
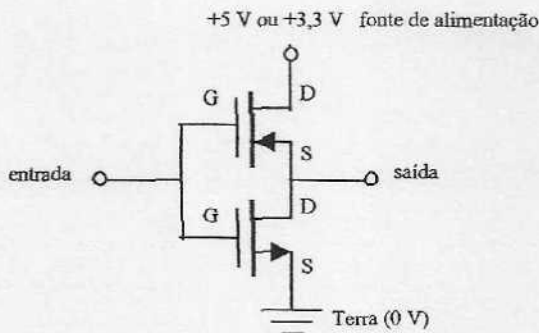


Circuito de saída totem pole

O nome vem da forma do circuito, com dois transistores empilhado, como se fossem duas carrancas formando um totem (em inglês, um totem é chamado "totem pole", sendo "pole" poste).

Utiliza um par de transistores (bipolares ou MOS):

circuito equivalente:



Neste caso, a tensão de entrada faz com que um transistor conduza e o outro "corte"; no circuito equivalente, o inversor simboliza inversão do efeito da tensão de controle sobre a chave de cima, em relação à de baixo.

Conforme a tensão de entrada, o transistor a conduzir será o que liga a saída à fonte ou à terra, fazendo com que essa ligação sempre se dê através de uma pequena resistência (o transistor conduzindo), e não através de um resistor, de modo que a queda de tensão é minimizada. Isto faz com que a tensão de saída, alta ou baixa, sempre seja aplicada de forma bem definida, com capacidade de fornecer uma corrente considerável à carga (equivalente a um "fan-out" elevado).

Entradas dos circuitos lógicos

As entradas devem receber a tensão de entrada consumindo pouca corrente, para que a família a que pertence o circuito tenha um "fan-out" elevado (pode-se falar em "fan-in" para essa característica de uma entrada "carregar" mais ou menos as saídas a ela ligadas). Portanto, devem ter impedância tanto quanto possível alta, o que significa alta resistência e baixa capacitância (alta reatância capacitiva).

O valor da impedância de entrada depende da família do circuito, como já vimos; a resistência de entrada não é tão alta como seria desejável nos circuitos TTL e é bastante alta nos circuitos CMOS. Por outro lado, a capacitância de entrada é pequena, porém não desprezível em ambas as famílias, sendo mais significativa nos circuitos CMOS (o que os torna mais lentos que os circuitos TTL).

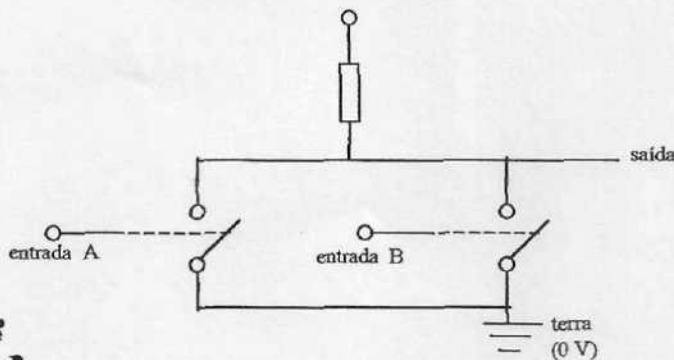
A função de cada entrada está relacionada com o tipo de circuito, ou seja, com a função do circuito. A parte intermediária do circuito (entre entrada e saída) também depende da função.

Circuitos equivalentes de algumas portas

Para simplicidade das figuras (e porque o que interessa aqui é a parte intermediária das portas), serão representados simbolicamente circuitos equivalentes com saída simples (nem "totem pole" nem "tri-state", mas saída com resistor de coletor ou de dreno).

Porta Não-OU (NOR) de duas entradas

alimentação: +5 V ou +3,3 V

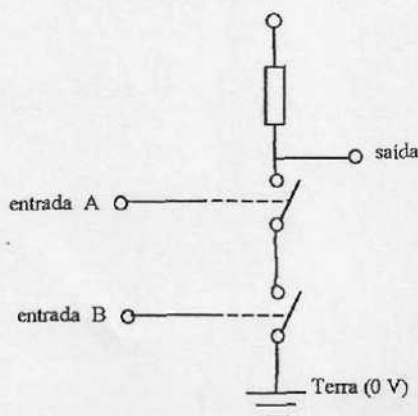


Basta uma das entradas altas para a saída estar ligada à terra pela chave respectiva fechada. Só com ambas entradas baixas e, portanto, as duas chaves abertas, a saída apresenta uma tensão alta proveniente da fonte de alimentação através do resistor.

Porta OU (OR): mesma coisa, com estágio adicional de inversão antes da saída.

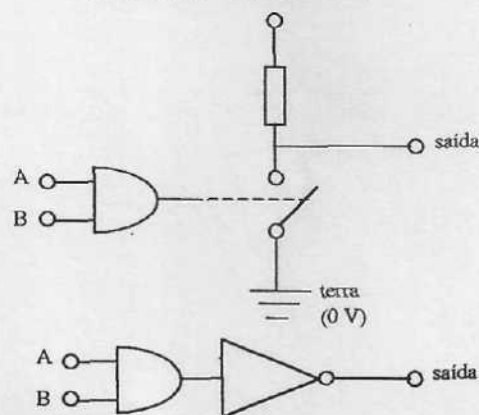
Porta Não-E (NAND)

alimentação: +5 V ou +3,3 V



Formas alternativas de se simbolizar a NAND com base no símbolo da AND:

alimentação: +5 V ou +3,3 V



Portas E (AND): mesma coisa, com estágio adicional de inversão antes da saída.

Circuito "wired-and"

Se houver necessidade de uma combinação de portas em que se tenha, por exemplo, duas portas Não-OU seguidas de uma porta E, é possível obter essa combinação utilizando apenas duas portas Não-OU com saída que não seja totem pole nem três estados (ou seja, com resistor de coletor ou dreno, ou senão tipo "coletor aberto"), simplesmente ligando suas saídas em paralelo; a ligação em paralelo das saídas faz o efeito de uma porta E, como se pode entender pela representação simbólica:

Nessa representação simbólica, nota-se que, à saída de cada porta Não-OU, tem-se, para cada uma (considerando-se a respectiva chave):

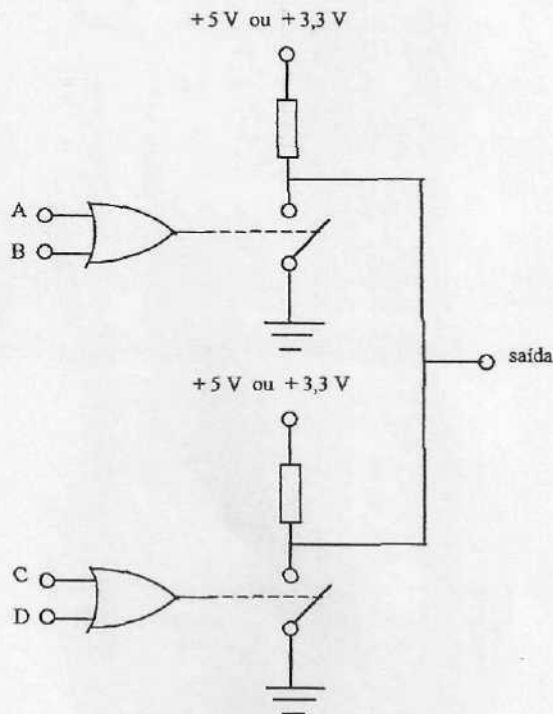
$$\overline{A + B}$$

$$\overline{C + D}$$

Quando $\overline{A + B} = 0$ ou $\overline{C + D} = 0$, a respectiva chave coloca a saída final do circuito à terra; é necessário que ambas as portas Não-OU tenham saída alta para que a saída final do circuito seja alta. Isso faz o mesmo efeito que ligar uma porta AND à saída das duas Não-OU. A função do conjunto fica:

$$\overline{(\overline{A + B}) (\overline{C + D})}$$

No caso de se utilizar portas com saída em coletor aberto, o circuito seria o mesmo, com a diferença de que haveria apenas um resistor para a fonte de alimentação ligado externamente.



Circuitos lógicos combinacionais são os circuitos lógicos que não têm memória. Em oposição, temos os de memórias, como os "flip-flops" e os seqüenciais (que podem incluir flip-flops).

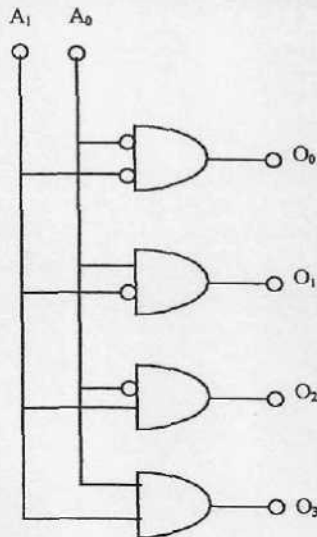
Os circuitos combinacionais geralmente são formados por combinações de portas lógicas. O motivo é simples: a função dos circuitos combinacionais é executar uma função lógica (sem uso de memória); e toda função lógica (que não envolva memória) pode ser implementada por meio de uma porta lógica.

Pode-se pensar em projetar um circuito combinacional sem se utilizar portas lógicas, simplesmente considerando as propriedades elétricas que correspondam à realização da função desejada. O circuito resultante, porém, será equivalente a uma combinação de portas lógicas, mesmo que com alguma simplificação por não ser necessário implementar todos os aspectos de cada porta. Além disso, os recursos de simplificação de funções lógicas (mapa de Karnaugh) lidam com funções lógicas básicas, que correspondem a portas lógicas.

Decodificadores

Decodificadores têm entrada com n fios e saída com 2^n fios.

A finalidade é identificar o número binário de n bits colocado a sua entrada; isso é feito pela colocação de uma tensão em um dos 2^n fios de saída. Com essa "identificação", ele está informando qual foi o número, ou seja, está "decodificando" o número binário. Exemplo com $n = 2$, $2^n = 4$:



Funcionamento:

Números binários de entrada possíveis: 00, 01, 10, 11

(em decimal: 0, 1, 2, 3)

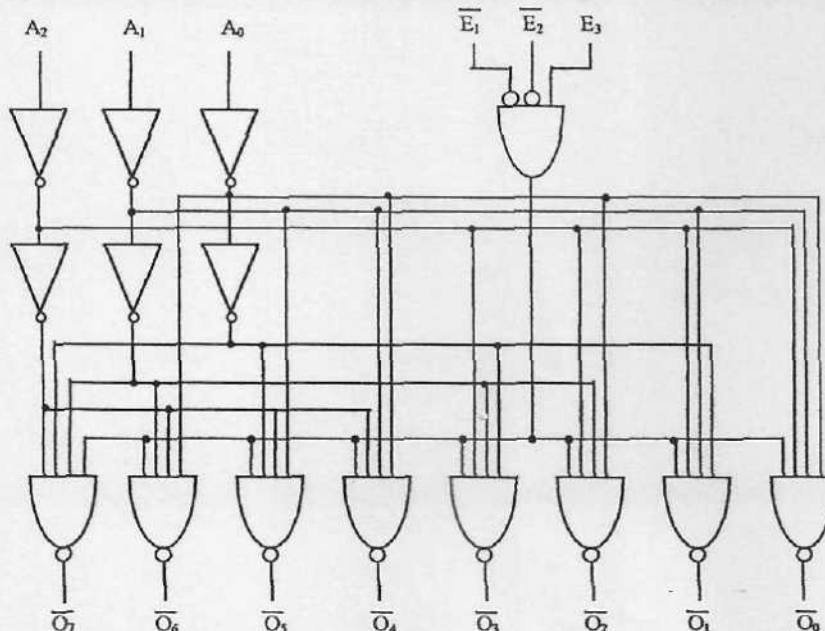
Por exemplo, para $A_1 = 1$ e $A_0 = 0$

(número 10, em binário; 2 em decimal),

apenas a porta cuja saída é O_2 fica com tensão alta.

Todas as outras ficam com tensão baixa.

Um circuito integrado decodificador produzido comercialmente é o de tipo 74138, ou 54138 (versão militar), ou 74LS138, etc. O esquema que mostra o funcionamento desse tipo de circuito é mostrado abaixo:



Exemplos de uso de decodificadores:

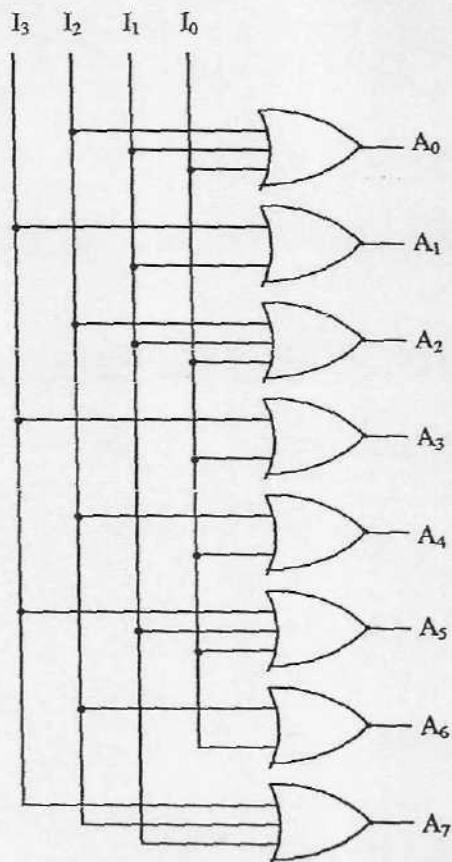
- para interpretar endereço de memória (achar fisicamente a posição da palavra)
- para reconhecer endereço de dispositivo de entrada/saída
- para interpretar instrução de programa

Codificadores

Codificadores recebem na entrada um número de fios, cada qual representando um evento que deve ser informado a outra unidade por meio de código digital; se houver, digamos, m eventos possíveis, dos quais só um poderá ocorrer de cada vez, serão m fios; enquanto nenhum dos eventos ocorrer, todos os m fios permanecem com a mesma tensão (por exemplo, bit 0); ao ocorrer um dos m eventos, o fio correspondente recebe uma tensão que informa que aquele evento ocorreu (por exemplo, bit 1).

Se for conveniente transmitir o número binário correspondente ao número de ordem do fio que receber tensão informando a ocorrência do respectivo evento, é necessário que m seja potência de 2, para que a saída do codificador seja um número binário de n bits, transmitido por n fios, sendo $m = 2^n$.

No entanto, nada obriga, em princípio, que se utilize código representando numeração binária seqüencial; pode-se utilizar um código arbitrário, definido pelo projetista, inclusive com mais fios de saída que o necessário para transmitir a informação de forma numérica seqüencial; como exemplo, o circuito abaixo faz isso, tendo quatro entradas (quatro eventos possíveis) e oito saídas (código arbitrário utilizando oito bits para informar qual dos quatro eventos ocorreu); o código improvisado é mostrado na tabela:

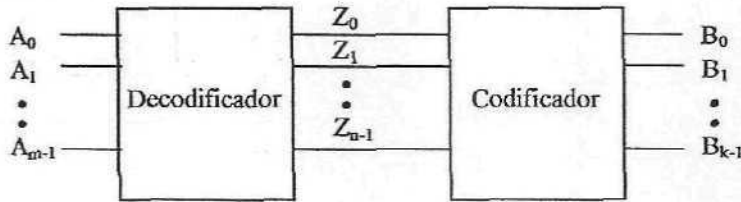


I_3	I_2	I_1	I_0	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1				1		1		1		1	
	1			1	1		1		1		1
		1		1		1			1	1	1
			1		1	1	1	1	1	1	1

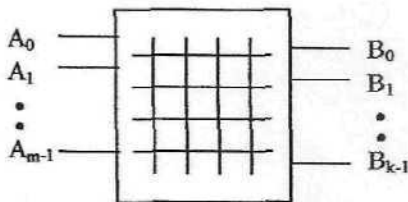
ORIGINALS NI
XERO
ORIGINALS NI COPY
A ME

Conversores de código

Pode-se interligar um decodificador e um codificador ou fazer uma matriz de conversão de código; decodificador/codificador:



Matriz:



Exemplos de uso de **conversores de código**:

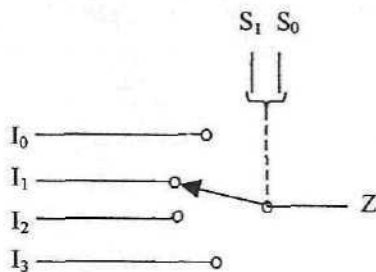
- conversor de binário para código de 7 segmentos
- conversor de um código binário simples para outro (binário para BCD, ASCII, etc, ou vice-versa)
- converter "op-code" de uma instrução de programa em endereço de micro-instrução

Multiplexadores

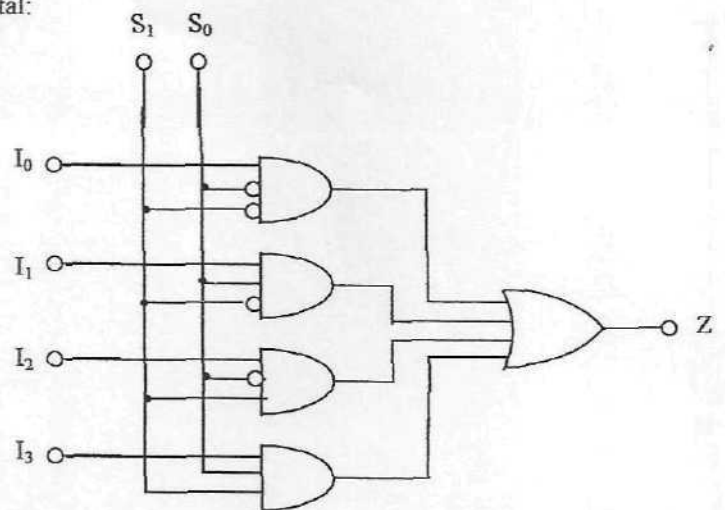
Multiplexadores permitem escolher que saída (entrada do multiplexador) enviará sinal a uma entrada (pela saída do multiplexador), como se fosse uma chave (eletrônica); a decisão sobre qual entrada do multiplexador enviará sinal para a saída deve ser informada ao multiplexador por uma palavra digital com k bits;

para n entradas, $n = 2^k$.

Símbolo (chave mecânica):



Circuito digital:

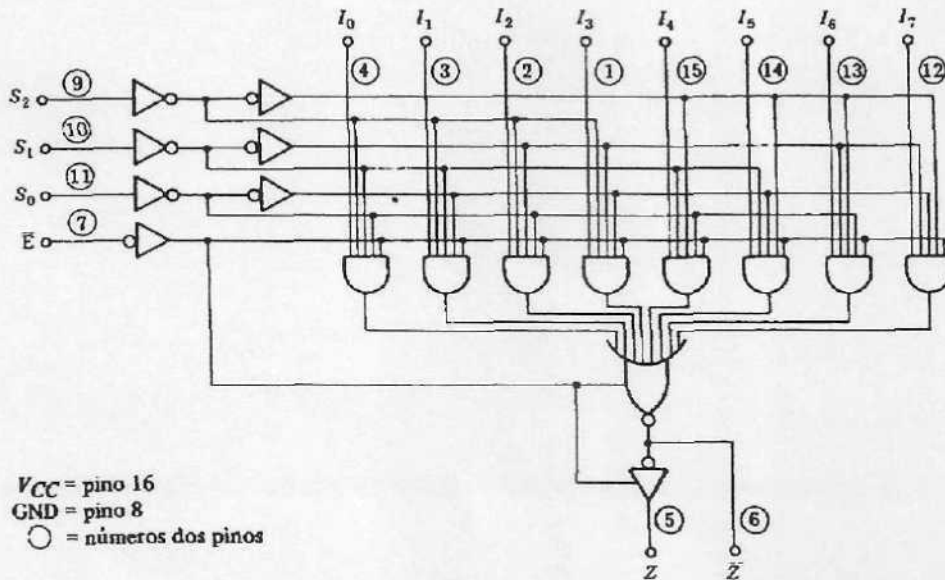


I_0, I_1, I_2, I_3 : entradas de sinal (a escolher uma); Z: saída (à qual a entrada escolhida estará conectada);

S_0, S_1 : entradas de controle;

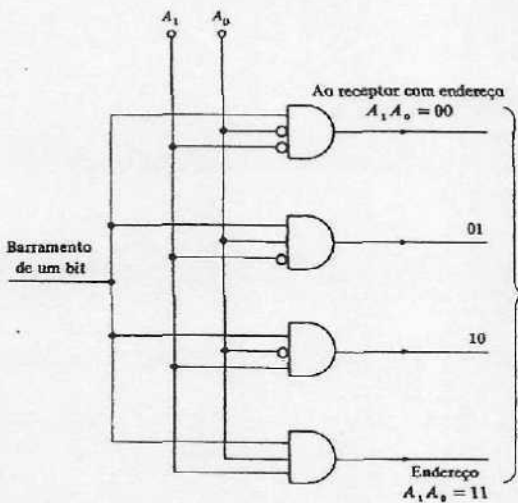
para $S_0, S_1 = 0,0; 0,1; 1,0$ ou $1,1$, será conectada à saída a entrada $I_0; I_1; I_2$ ou I_3 , respectivamente.

Exemplo de circuito multiplexador em integrado comercial: 74151

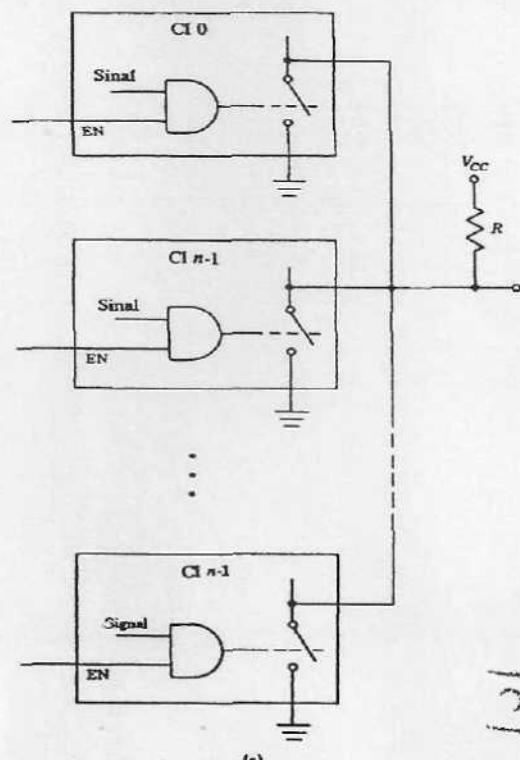


Demultiplexadores

Permitem enviar o sinal que vem por uma linha para um dentre n linhas separadas (ou seja, faz o inverso do que o multiplexador faz, ou ainda, complementa o que o multiplexador faz, pois enquanto este junta vários caminhos de sinal em um único caminho, o demultiplexador separa novamente):



É possível selecionar uma saída de circuito dentre várias utilizando a entrada de habilitação ("enable") de um circuito de saída três estados ("tri-state"):

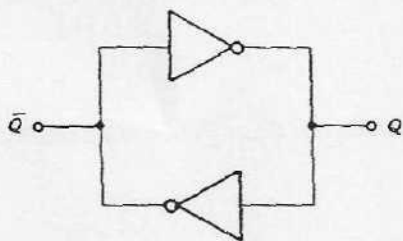


ORIGINALS N
ORIGINALS MIT'CC

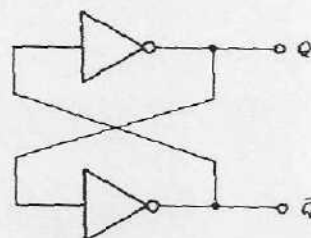
"Flip-flops" são circuitos que mantêm um de dois estados possíveis; servem como células de memória. Também chamados "multivibradores bi-estáveis (há os mono-estáveis e os a-estáveis; apenas os bi-estáveis nos interessam aqui). As formas mais simples são também chamadas "latches". Os flip-flops são "latches" mais elaborados, com dois estágios, chamados "flip-flop mestre-escravo". Vejamos primeiro latches, para, daí, chegar ao flip-flop.

Latches são circuitos capazes de armazenar uma informação binária simples: "0" ou "1", "tensão baixa" ou "tensão alta";

forma mais simples: interligação de dois inversores:

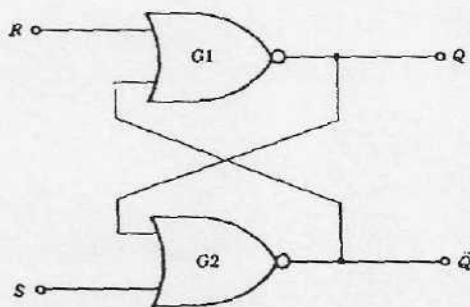


desenhando de forma conveniente:



tensão de saída alta em Q produz tensão baixa em \bar{Q} , o que mantém tensão alta em Q (armazenamento), inconveniente: como provocar mudança? solução: mais de uma entrada em cada porta:

Usando porta "NOR":



Símbolo:

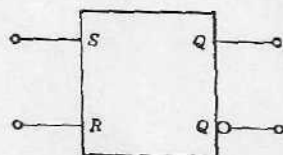


Tabela verdade:

S	R	Q	\bar{Q}
L	L	L ou H	H ou L
L	H	L	H
H	L	H	L
H	H	Não usada	

S: "Set" - "colocar", tornar positivo

L: "Low" - tensão baixa

R: "Reset" - desfazer o "set", tornar zero

H: "High" - tensão alta

Tensão baixa em ambas as entradas: tratando-se de portas NOR, a tensão baixa não interfere; o circuito age como se fossem dois inversores (como antes).

Tensão alta em uma das portas (apenas uma): força tensão baixa na saída, aplicada à entrada da outra porta, que fica com as duas entradas com tensão baixa, causando tensão alta em sua saída; esta tensão alta está aplicada a uma entrada da outra porta, o que mantém a situação nessa outra porta (tensão baixa na saída), mesmo depois de se retirar a tensão aplicada.

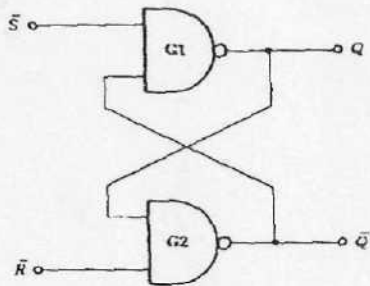
Forma alternativa da tabela verdade:

S	R	Q_{n+1}	\bar{Q}_{n+1}
L	L	Q_n	\bar{Q}_n
L	H	L	H
H	L	H	L
H	H	Não usada	

Tensão na entrada S ou R produz alteração de estado; portanto, envolve tempo: há um instante de tempo no qual o flip-flop tem um estado, outro em que há outro estado; chamamos um instante qualquer de "n"; o instante seguinte é n+1; na forma alternativa da tabela verdade, é mostrado qual o estado no instante Q_{n+1} ; este instante é igual a Q_n se não houve alteração; se houve alteração, pode ter-se tornado L ou H.

Latches, Flip-flops (continuação)

Latches usando portas "NAND":



Símbolo

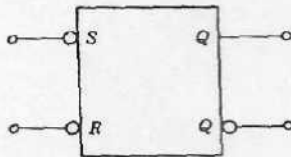


Tabela verdade:

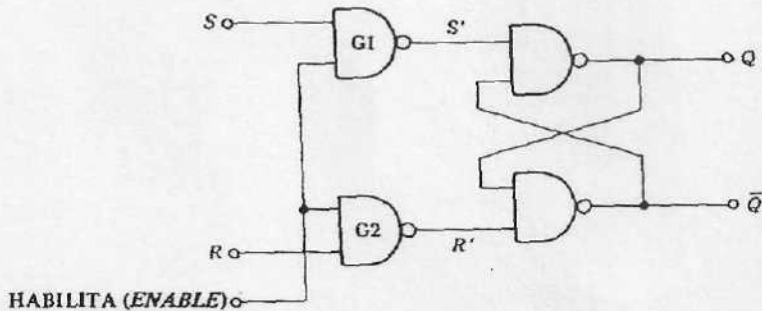
S	R	Q	Q̄
L	L	Não usada	
L	H	H	L
H	L	L	H
H	H	L ou H	H ou L

Neste caso, é preciso manter as duas entradas em tensão alta para não alterar o estado do latch; tensão baixa na entrada de uma das portas NAND faz com que sua saída seja alta, o que causa tensão baixa na outra saída, o que mantém a situação provocada na primeira porta.

Forma alternativa:

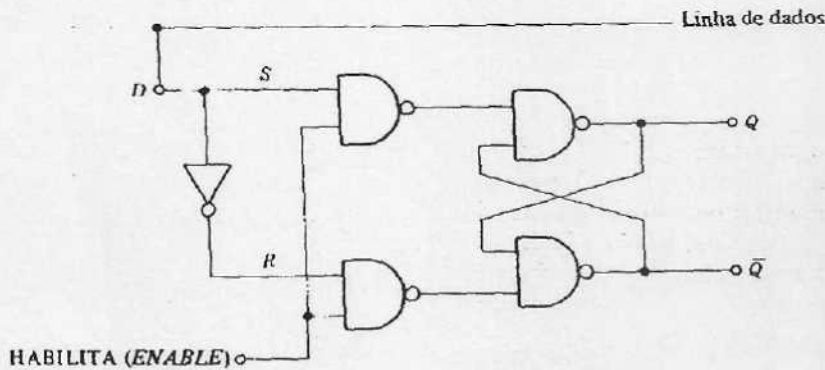
S	R	Q _{n+1}	Q̄ _{n+1}
L	L	Não usada	
L	H	H	L
H	L	L	H
H	H	Q _n	Q̄ _{n+1}

Latch controlado - só aceita entrada quando tensão de controle (ou de "relógio", ou "clock") estiver alta:



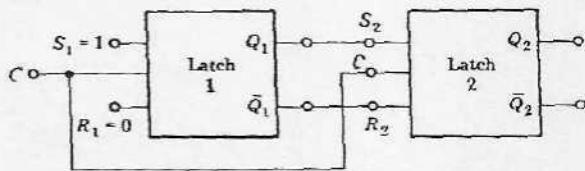
Só há alteração (tensão baixa) em S' ou R' ao aplicar-se tensão alta em S ou R se houver também tensão alta em Habilita (Enable); portanto, Habilita controla a possibilidade de alteração do estado do latch.

Latch controlado tipo "D":



Basta aplicar uma única tensão (o "dado") à única entrada D para "setar" o latch (Q=1) se D=1 ou resetar (Q=0) se D=0, porque essa tensão é aplicada a S e seu inverso é aplicado a R. Por isso, este latch é mais útil como célula de memória, para guardar (armazenar) um bit.

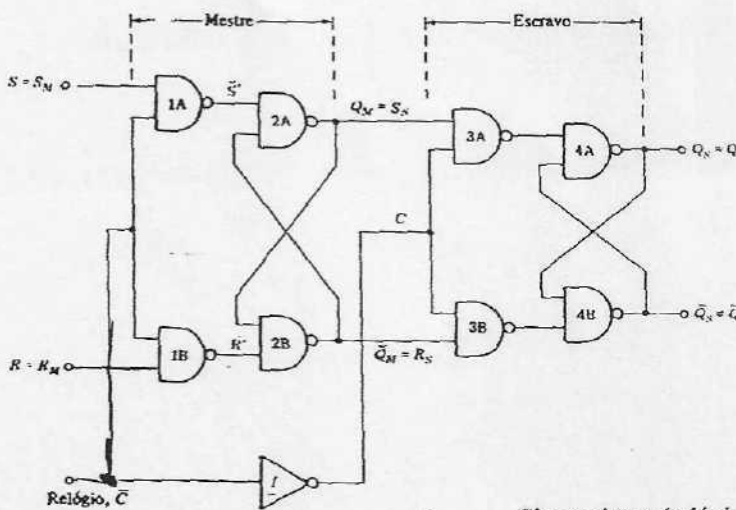
Problema com o "latch":



Tentamos fazer um registrador de deslocamento usando apenas latches controlados. Que acontece se a intenção for entrar com dado no latch 1 enquanto passa o dado que está no latch 1 para o latch 2? O que entrar no latch 1 aparece de imediato à entrada do latch 2 e é repassado para este, perdendo-se o que estava antes no latch 1 (ou seja, não funciona como registrador de deslocamento).

"Flip-flop R-S Mestre-escravo"

dois estágios, para evitar que o dado de entrada do latch 1 passe também para o latch 2:

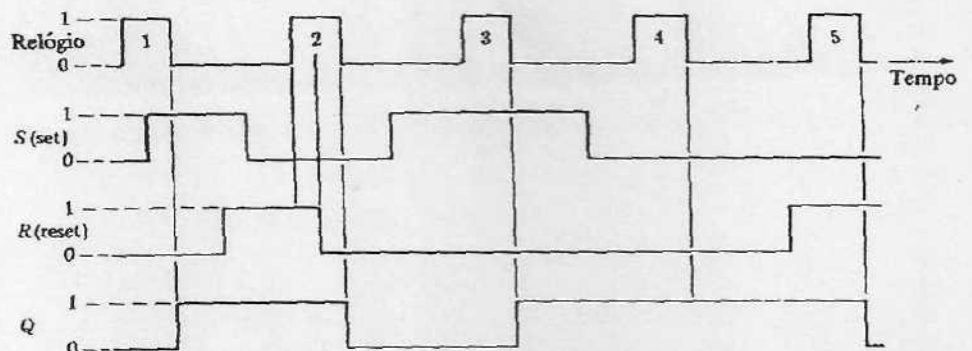


Seqüência de eventos:



Sincronismo (relógio, "clock"):

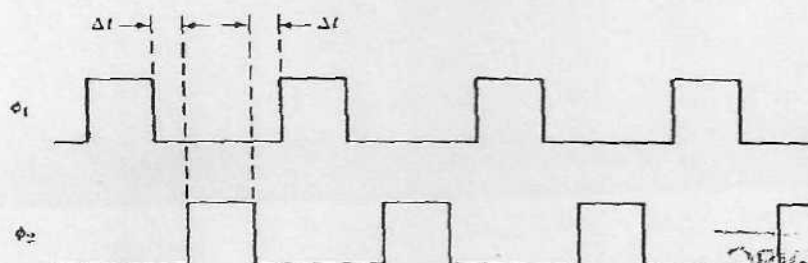
As entradas R e S atuam sobre o latch "mestre"; as entradas do latch "escravo" vêm das saídas do latch "mestre", que repassa seu estado para o "escravo". A entrada de controle (habilita, enable, relógio, clock) atua no "mestre" e no "escravo" em ocasiões diferentes devido ao inversor; portanto, a tensão aplicada à entrada altera apenas o estado do "mestre" enquanto a tensão do relógio for alta; o novo estado é repassado para o escravo (para os dois ficarem iguais) apenas quando a tensão do relógio voltar a baixar.



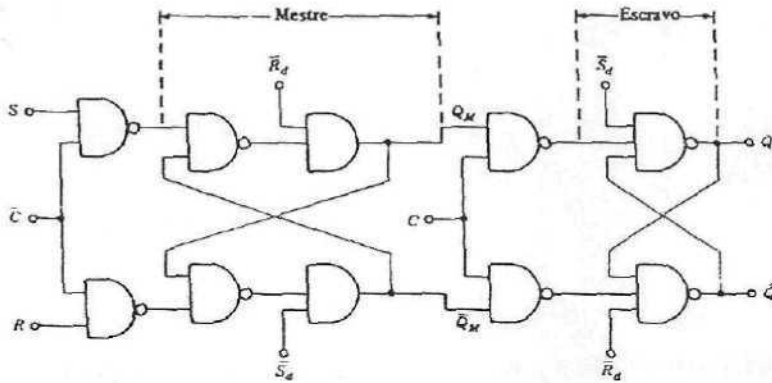
Isto faz com que seja possível entrar com um dado e repassar o dado anteriormente armazenado para um outro flip-flop, como no registrador de deslocamento.

Por isso, esse flip-flop é dito "gatilhado pela descida do pulso" (só na descida do pulso sua saída é atualizada).

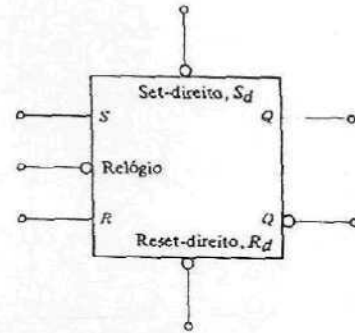
Sincronismo de duas fases (com duas entradas de relógio, ϕ_1 e ϕ_2 , defasadas):



Flip-flop R-S com entradas de "set" e "reset" diretas:



Simbolo:



Cada entrada "direta" (\bar{S}_d ou \bar{R}_d) atua sobre a porta à qual é aplicada sem depender do relógio; por isso, sua atuação é imediata, não precisando esperar pelo próximo pulso de relógio para atuar. Pode estar aplicada a uma das duas posições indicadas (ou existir em ambas).

Outros tipos de Flip-flop:

Flip-flop tipo J-K:

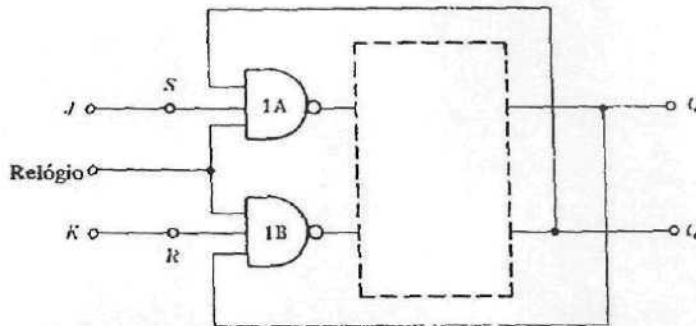
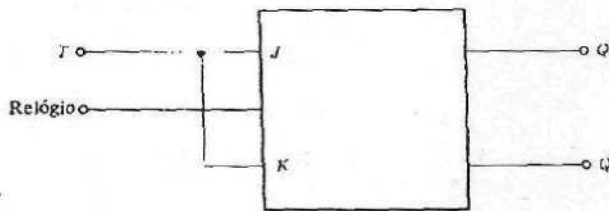


Tabela verdade:

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	\bar{Q}_n

J-K: é possível aplicar $J=1$ e $K=1$ ao mesmo tempo porque a saída que estiver com tensão '0' bloqueia a entrada do lado oposto (que não faria mesmo nenhum efeito). Com isso, aplicando-se 1 em ambas as portas, o flip-flop J-K inverte o estado.

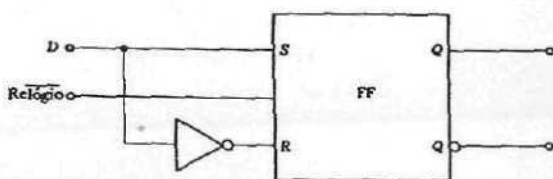
Flip-flop tipo T (de "Toggle", "chave"):



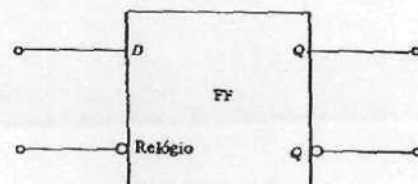
T: aplica tensão de entrada simultaneamente em J e K, de modo que o efeito de 1 na entrada T é inverter o estado.

D: o dado aplicado em D é aplicado em S; seu inverso, em R, de modo que o dado é armazenado.

Flip-flop tipo D (de "Dado"):



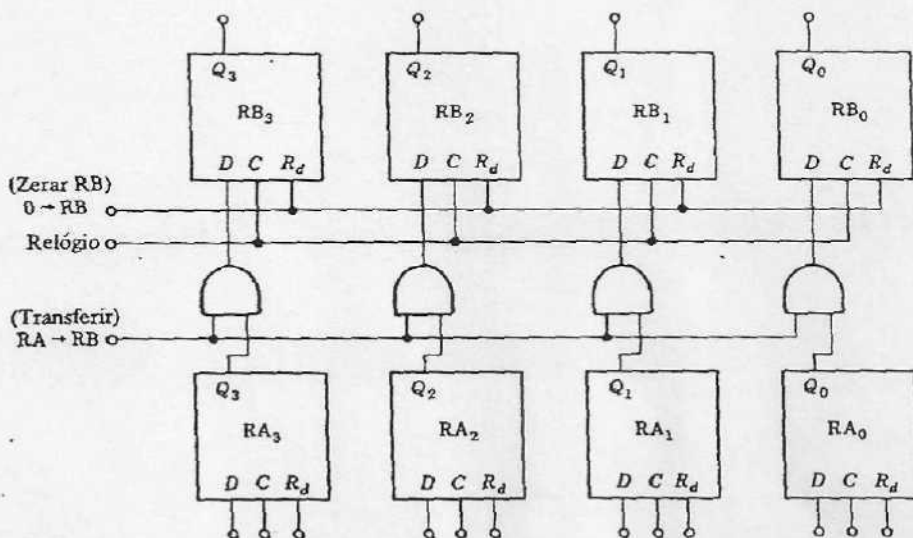
Simbolo:



XEROX ORIGINALS NIT
NIT COPY TIDA
ORIGINALS NIT COPY

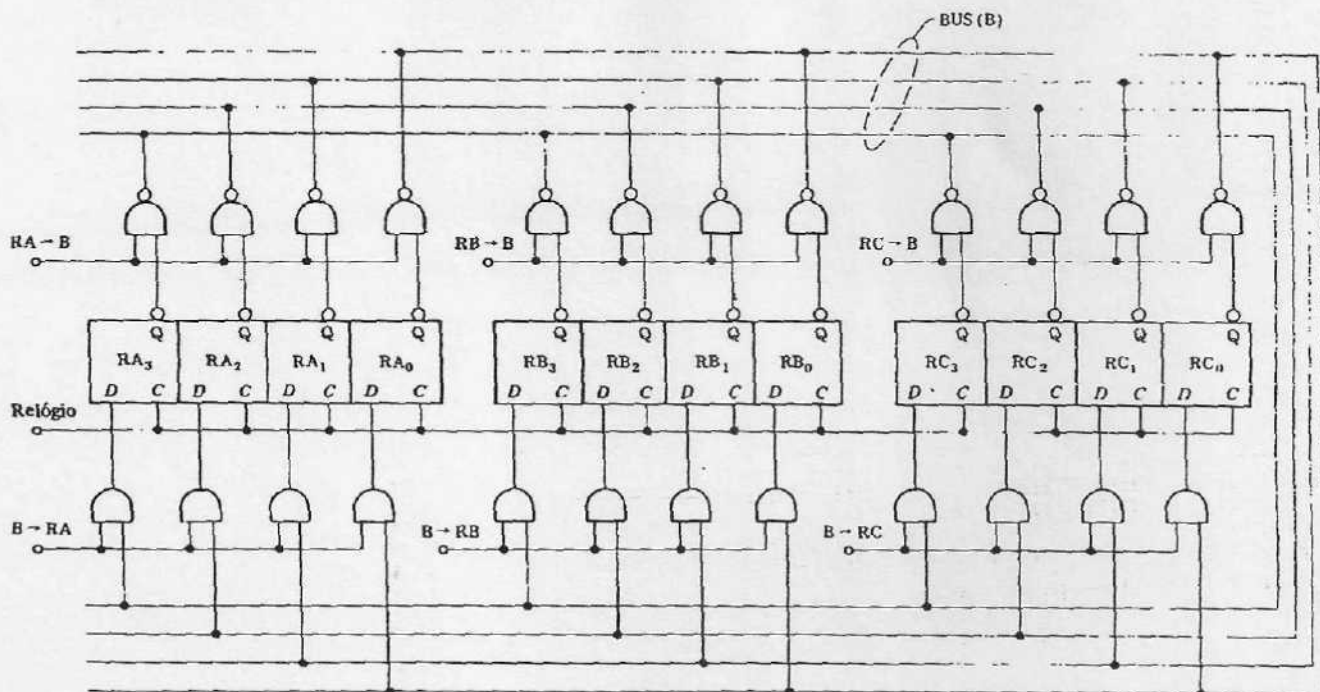
Registadores são grupos de "latches" ou de "flip-flops" que servem para armazenar grupos de bits; um conjunto de "n" flip-flops (ou "n" latches) armazena uma palavra de "n" bits.

Além de armazenar, um registrador deve ser capaz de transferir seu conteúdo para outro registrador; transferência de palavra de um registrador para outro (diretamente):



O mais comum, porém, é transferência entre registradores através de barramento.

Transferência de registrador para registrador através de barramento:

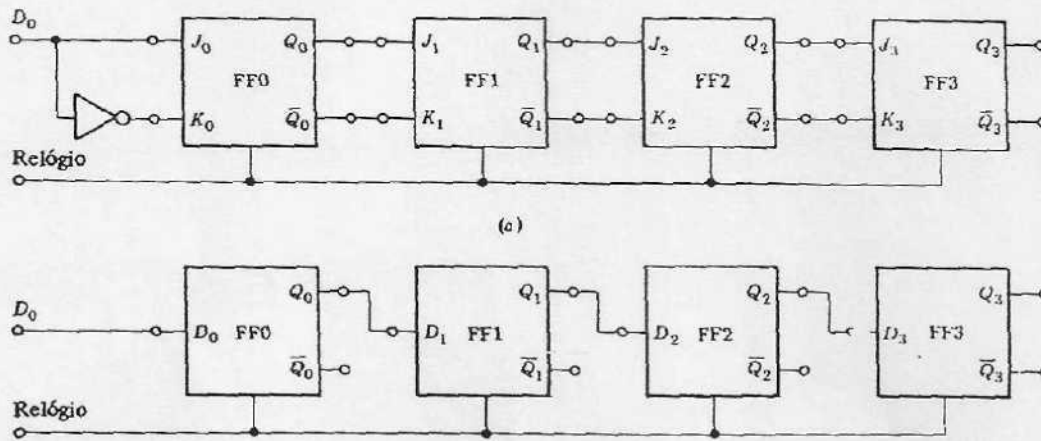


Para não zerar os registradores a cada pulso de relógio quando $B \rightarrow RA = 0$, é preciso modificações: habilitar a entrada "C" ("Clock", relógio, habilitação, "enable") dos flip-flops de cada registrador separadamente apenas quando houver, simultaneamente, tensão alta no fio " $B \rightarrow RA$ " (ordem para armazenar) e no fio "Relógio" (que vem do gerador de relógio geral do sistema que utilize esses registradores).

Registadores de deslocamento (utilizando flip-flops mestre-escravo):

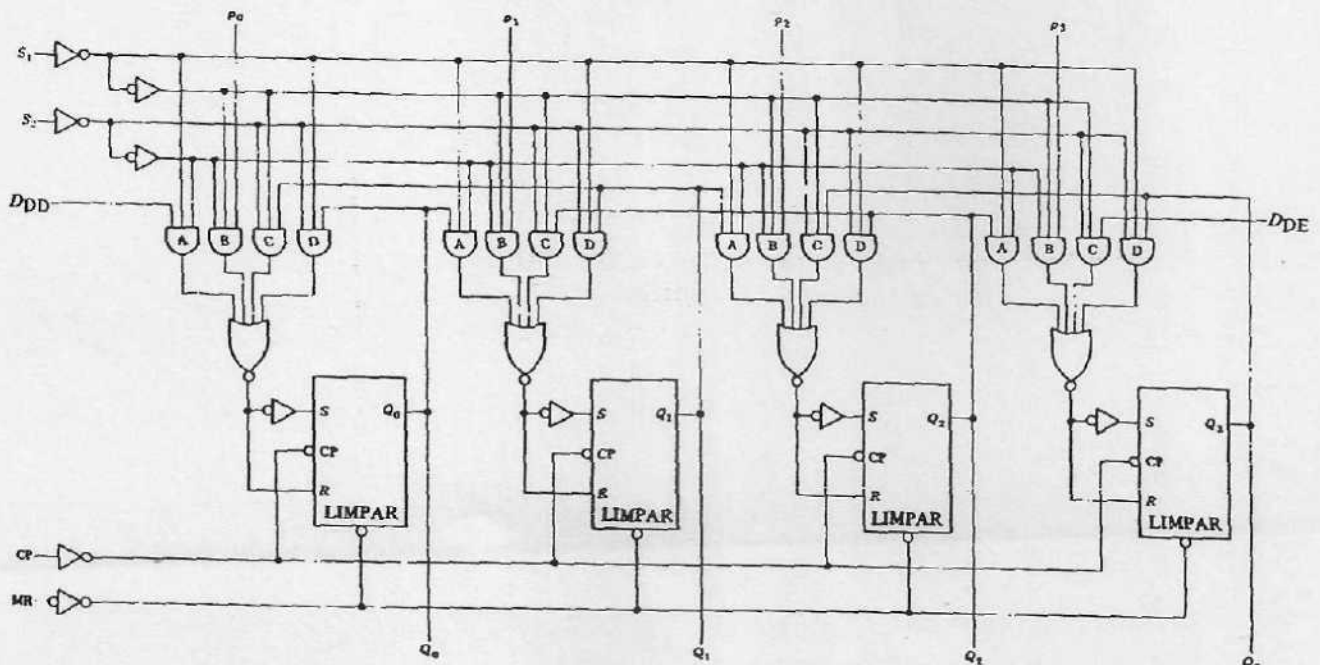
Utilizando-se flip-flops "mestre-escravos" fica fácil fazer um registrador de deslocamento, utilizando flip-flop tipo J-K ou D: basta ligar a saída de cada um com a entrada do seguinte (levando-se em conta o sentido em que se quer fazer o deslocamento); a entrada do primeiro estágio (mais à esquerda ou mais à direita, dependendo do sentido) fica sendo uma "entrada serial" para o registrador.

Exemplos com deslocamento para a direita (entrada à esquerda) com flip-flops J-K e D:



Registrador de deslocamento universal bidirecional de quatro bits tipo 74194:

É possível fazer um circuito com todas as interligações para funcionar como registrador de deslocamento para a direita (entrada serial à esquerda) ou para a esquerda (entrada serial à direita) e ainda entrada em paralelo (para se for desejado entrar com alguma palavra de forma não serial). É necessário incluir linhas de controle e portas lógicas para fazer com que o circuito assuma a forma adequada para cada uma dessas funções; o circuito abaixo é um exemplo de circuito comercial que faz essas coisas; ele tem as linhas de controle s_1 e s_2 para determinar, através das combinações 00, 01, 10, 11, qual das operações ele deve executar (analisando o circuito, é fácil descobrir a função de cada combinação de s_1 e s_2). Além de 4 operações (armazena palavra em paralelo, desloca para a direita com entrada serial pela esquerda, desloca para a esquerda com entrada serial pela direita, ou mantém o que já estava armazenado), é possível comandar a limpeza do flip-flop inteiro por meio da linha MR. A ação comandada se dá efetivamente quando o pulso de relógio retorna a zero.



Memória (revisão de Técnicas Digitais; novo para Ciência da Computação)

R - 7/1

UFF - TET - Microprocessadores - Prof Luiz Pinto de Carvalho

Memória guarda informação sob a forma de palavras digitais (números binários representando informações); geralmente é um conjunto de recursos para armazenar um número de palavras de "n" bits; são de vários tipos, conforme a utilização; os tipos se classificam de várias formas, segundo o ponto de vista.

Pode-se considerar tipos quanto a uso, forma de acesso, permanência da palavra armazenada, tecnologia, etc.

Caso particular de memória: registrador (uma só palavra; geralmente não é chamado "memória").

Tipos, quanto ao uso:

- memória principal
- memória secundária, ou de massa
- memórias auxiliares

Tipos, quanto ao acesso:

- acesso aleatório
- acesso seqüencial

Tipos, quanto à permanência do armazenamento:

- permanente
- volátil

Tipos, quanto à tecnologia:

- memórias a semicondutor
- memórias magnéticas
- memórias óticas
- outras

Há memórias que são, por exemplo, secundárias, permanentes e magnéticas (exemplo: disco);

Sub-tipos e características de cada tipo:

Memória principal: é a memória que guarda os dados e programas que estão sendo utilizados pelo computador; deve permitir leitura e escrita imediata e não precisa manter os dados após a máquina desligada; deve ser grande e, por isso, econômica; deve possibilitar acesso a qualquer palavra de dados diretamente.

Com esses requisitos, a memória principal, atualmente, tem as seguintes características:

- acesso aleatório;
- volátil;
- semicondutor (tipo RAM, DRAM ou SDRAM);

Memórias secundárias ou de massa:

- discos magnéticos
- discos óticos
- fitas magnéticas

Discos magnéticos são memórias auxiliares (ou de massa, porque guardam uma grande quantidade de dados), de acesso seqüencial (não é acessado tão constantemente quanto a memória principal, por isso não precisa ser de acesso aleatório), magnética.

Discos óticos (CD, DVD) são também auxiliares (ou de massa, ou entrada/saída), seqüenciais, porém óticos (e não magnéticos).

Fitas magnéticas são menos usadas atualmente, mas já foram dos meios mais utilizados em computadores de grande porte; são memórias de massa (secundárias), de acesso seqüencial, magnéticas.

Memórias auxiliares: cache e outras utilizadas para inicialização e para guardar parâmetros de configuração do computador.

Quanto ao acesso:

Memórias de acesso aleatório: o dado que está armazenado é obtido (recuperado) de imediato quando é fornecido seu endereço; o acesso não é aleatório no sentido de ser obtido "qualquer" dado, mas no sentido de não ser seqüencial.

Memórias de acesso seqüencial: para obter-se um dado, é necessário ler-se todos os dados armazenados desde o ponto em que a fita está, no momento, até chegar-se ao dado que se quer obter; por isso, demora.

Quanto à permanência:

Memória permanente: dados permanecem armazenados mesmo que seja desligada a alimentação.

Memória volátil: dados desaparecem quando a alimentação é desligada.

Quanto à tecnologia:

Memórias a semicondutor podem ser de vários tipos:

RAM: Random Access Memory - é de acesso aleatório e volátil; é chamada "de acesso aleatório" porque foi das primeiras memórias de acesso aleatório a semicondutor a surgirem; permite escrita e leitura; cada célula pode ser de vários tipos (por exemplo, um flip-flop);

ROM: Read Only Memory - de acesso aleatório e permanente; permite apenas leitura; cada célula pode ser de vários tipos, mais simples que as de uma RAM; por exemplo, pode ser uma matriz a diodos; já é fabricada com o programa ou os dados incluídos (não é "programável");

PROM: Programmable ROM: uma ROM que recebe os dados a serem armazenados depois de fabricada ("programável");

EPROM: Erasable Programmable ROM: além de programável, pode ser apagada e reprogramada; o apagamento é feito pela exposição de uma "janela" à radiação ultravioleta intensa;

EEPROM: Electrically Erasable PROM: o apagamento e reprogramação é feito pela aplicação de tensões.

Todas essas memórias funcionam como se fossem "pilhas" de registradores, que são chamados "palavra" ou "posição" de memória; cada posição de memória tem "n" bits; cada posição de memória tem um endereço; o acesso ao conteúdo de uma posição de memória é feito informando-se o endereço.

O acesso pode ocorrer para armazenar ("escrever") ou para obter, recuperar ("ler") o dado;

"leitura": fornecido o endereço e a ordem para ler, o conteúdo da posição de memória correspondente aparece no barramento de saída (ou de entrada/saída);

"escrita": fornecido o endereço e a ordem para armazenar, a palavra que estiver no barramento de entrada (ou de entrada/saída) é armazenada na posição correspondente ao endereço.

A memória principal de um computador é do tipo RAM ou tipo derivado deste, como DRAM, SDRAM, etc.

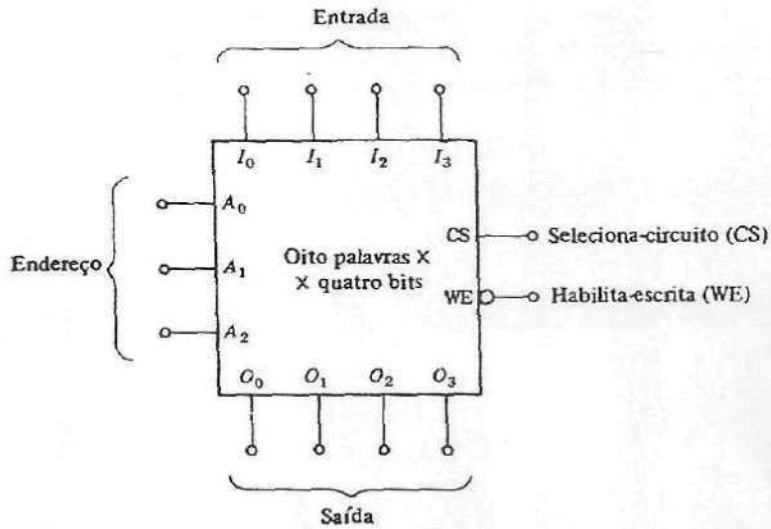
Memórias **DRAM** ("dynamic RAM", RAM dinâmica) são memórias mais baratas e consomem menos potência para funcionar que uma RAM; por isso, podem ter grande quantidade de palavras; baseiam-se em guardar o valor da tensão por meio da carga de um pequeno capacitor formado no próprio semicondutor; devido ao escoamento das cargas, precisam ser renovadas constantemente.

Memórias **SDRAM** ("synchronous DRAM", DRAM síncrona) são, basicamente, memórias DRAM que operam com rigoroso sincronismo de todas as operações básicas envolvidas na leitura e escrita, de modo a obter o menor tempo de acesso possível.

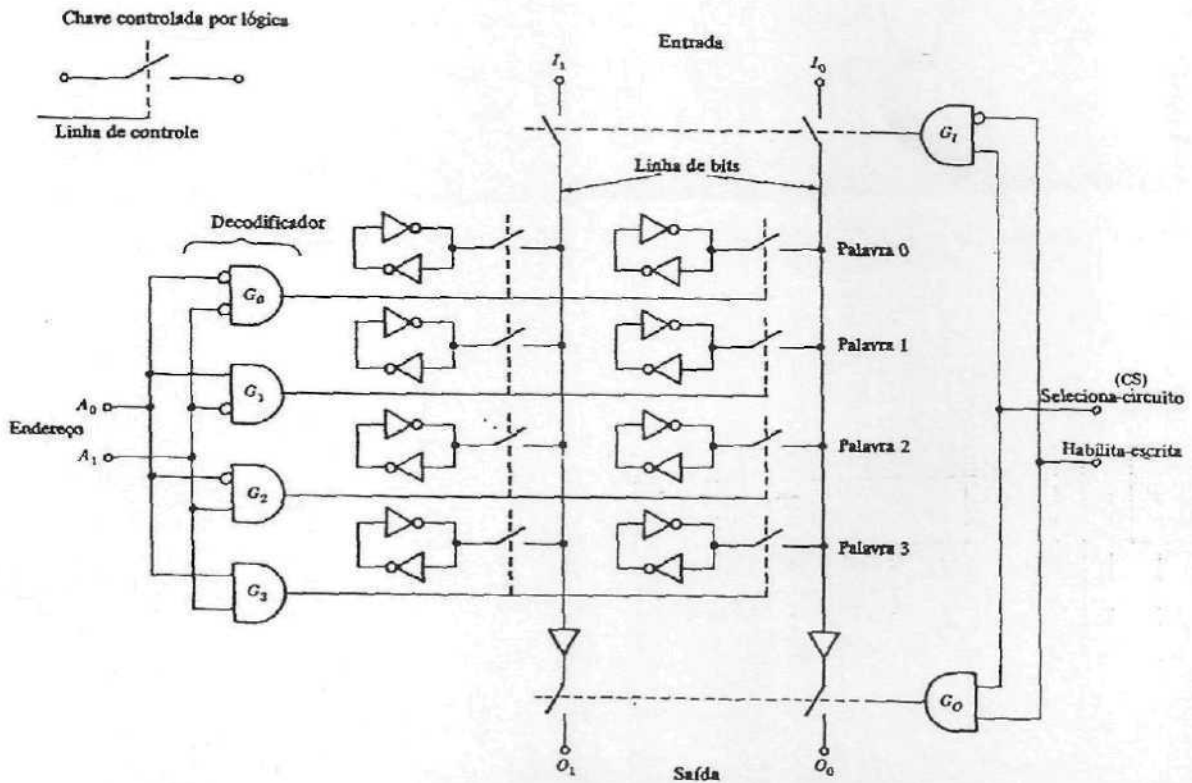
Estrutura das memórias de acesso aleatório

Um circuito integrado com memória de acesso aleatório tem pinos (fios) para entrada de dados, saída de dados, endereço e sinais de controle (ler, escrever).

Um exemplo de memória com 8 palavras (endereços de 3 bits) de 4 bits cada pode ser representada assim:



sendo uma RAM, pode ser representada simplificada assim:



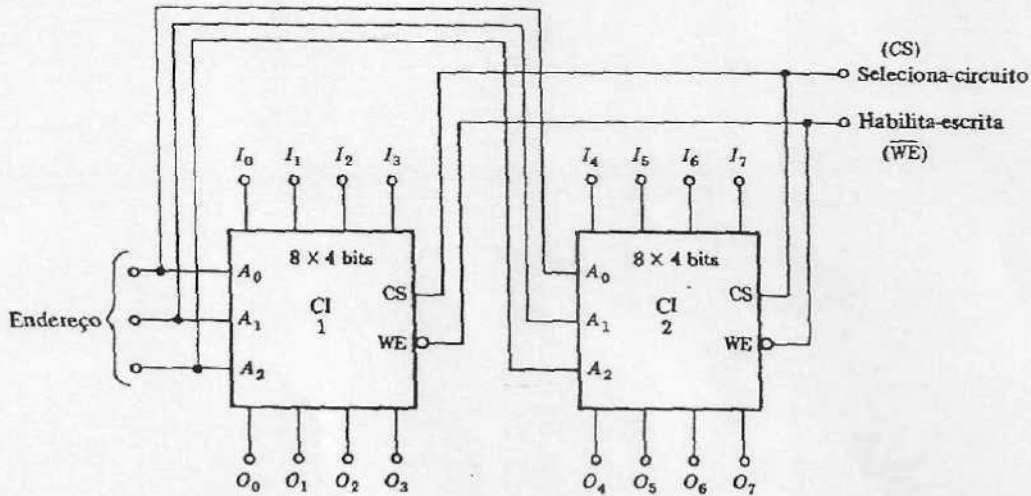
Memória (continuação)

R - 7/4
UFF - TET - Microprocessadores - Prof Luiz Pinto de Carvalho

Ampliação de memória

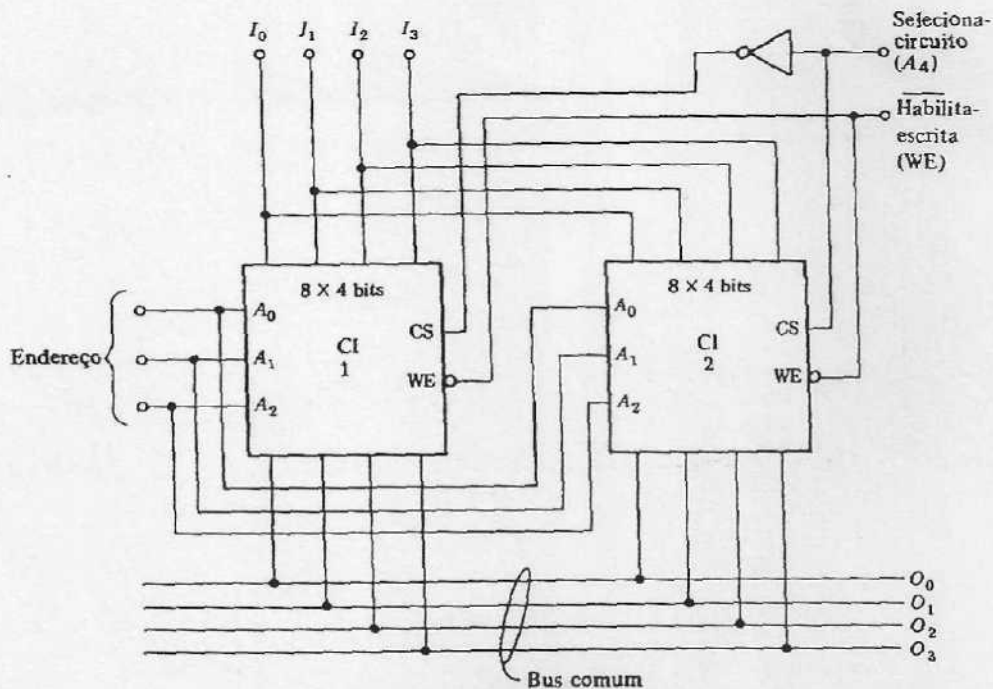
Para atingir-se a capacidade de memória desejada, pode ser necessário ligar mais de um CI juntos; se o módulo tiver palavras menores do que o desejado, pode-se querer a mesma quantidade de palavras, cada uma com mais bits.

Exemplo de ampliação de memória de 8 palavras de 4 bits (8×4) para ter palavras de 8 bits (8×8):



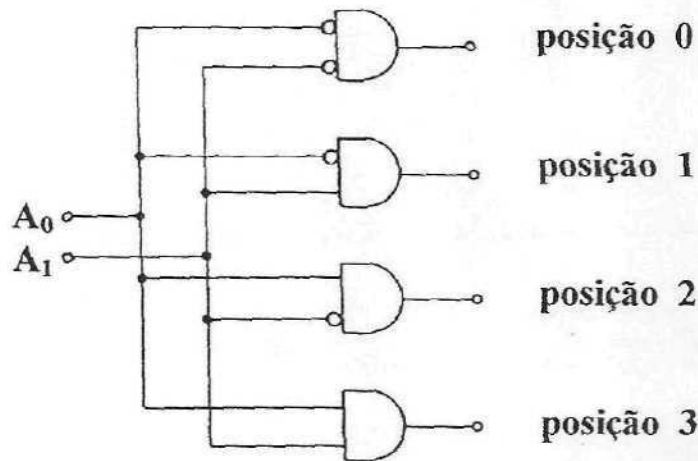
Pode ser necessário aumentar o número de palavras com o mesmo número de bits.

Exemplo de ampliação de um módulo de 8 palavras de 4 bits (8×4) para dezesseis palavras (16×4):

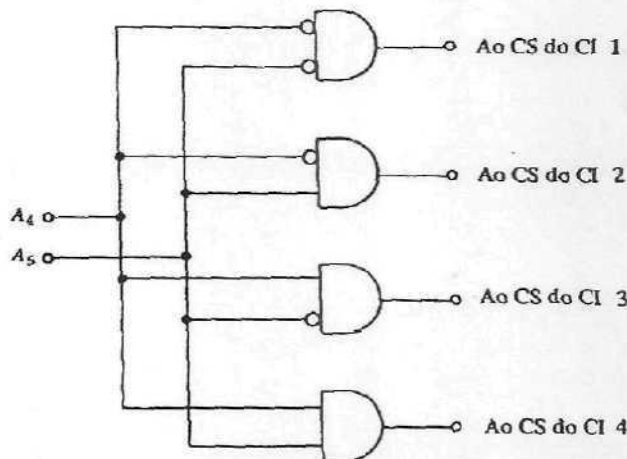


Endereço e decodificadores

Dentro da memória, o endereço deve "encontrar" fisicamente o grupo de células que formam a palavra; para isso, um decodificador "traduz" a combinação de 0s e 1s na aplicação de tensão alta em um fio; para cada endereço, há um fio que vai a cada posição de memória:



O mesmo "truque" pode ser utilizado para localizar um módulo de memória dentre outros; por exemplo, localizar um módulo em quatro:



as figuras têm mostrado CIs com linhas de entrada e saída de dados separadas; no entanto, com o controle de leitura/escrita e utilizando habilitação de saída, é possível fazer a comutação dentro do CI de modo que o CI tenha linhas de dados (entrada/saída juntas).

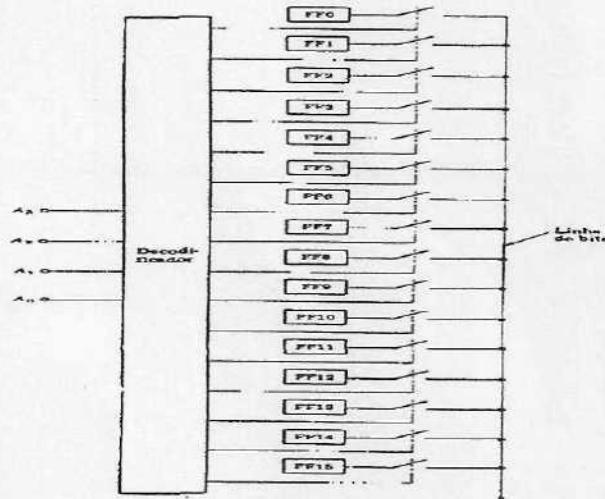
Memória (continuação)

R - 7/6

UFF - TET - Microprocessadores - Prof Luiz Pinto de Carvalho

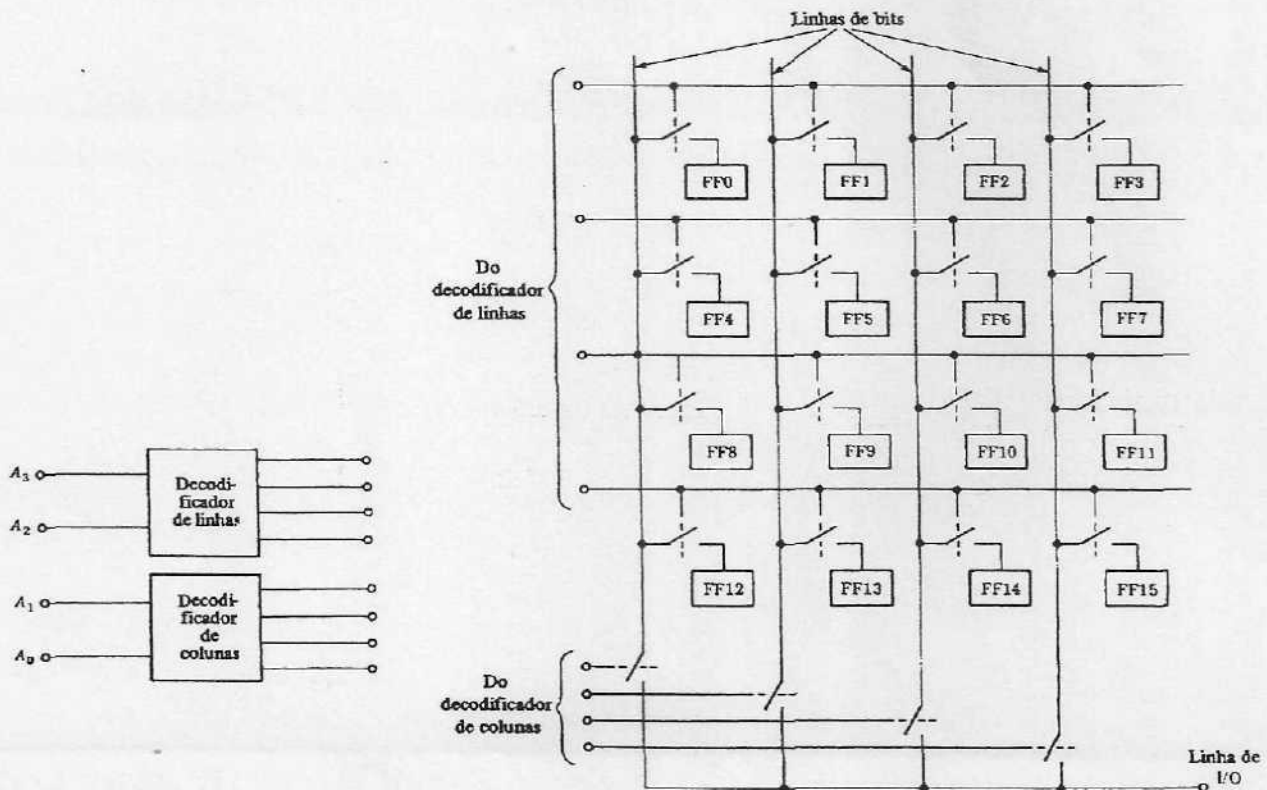
Para evitar quantidade excessiva de condutores no interior do CI, o decodificador pode ser substituído por dois decodificadores menores; as linhas de saída são combinadas em uma matriz; cada cruzamento da matriz corresponde a uma posição de memória; exemplo: memória de 64 MB formada por duas placas, com 2 CIs cada: cada CI tem 16 MB; com um só decodificador, seriam 16 milhões de condutores dentro de cada CI; as duas figuras abaixo comparam o que se teria no caso de uma memória de 16 palavras organizada com um ou dois decodificadores.

Memória de 16 palavras (4 bits de endereço) com um só codificador (16 linhas):



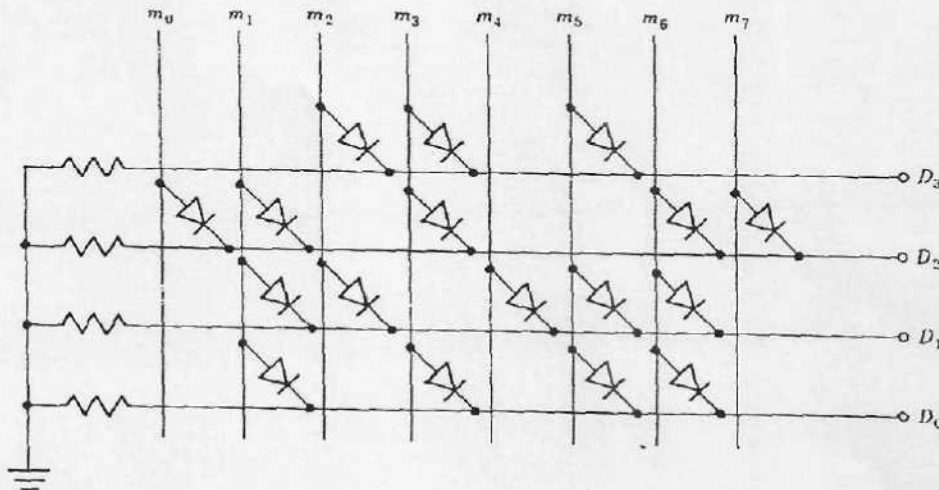
No de CI de 16 MB com 2 decodificadores, cada CI teria que ter, internamente, $2 \times \sqrt{16\ 000\ 000} = 2 \times 4000 = 8\ 000$ condutores (em vez de 16 000 000 de condutores).

Memória de 16 palavras (4 bits de endereço) com dois codificadores (4 + 4 linhas):



Célula de memória: se for RAM, pode ser um latch ou flip-flop; se DRAM ou SDRAM, pequeno capacitor no semicondutor, se ROM, diodos.

Exemplo de organização de uma memória ROM com 8 palavras de 4 bits cada:



O decodificador que recebe o endereço aplica tensão alta em uma das linhas m_0 a m_7 , localizando assim a palavra; onde houver diodo, a tensão alta passa para a linha de bit (D_0 a D_3), indicando bit 1; onde não houver diodo, a tensão permanece baixa, o que é causado pelos resistores ligados à terra. Portanto, a presença de diodo corresponde a bit 1 armazenado e a ausência, a bit 0.

A ROM já é fabricada com seu conteúdo gravado.

Uma PROM (Programmable ROM) é fabricada com todos os diodos e com "fusíveis" nas conexões dos diodos. A aplicação de tensões apropriadas nas linhas de dados e de endereço (externamente à memória) causa a fusão desses fusíveis, alterando o valor do bit armazenado; portanto, são programáveis (mas não reprogramáveis, pois não são "apagáveis": o fusível "queimado" permanece assim para sempre).

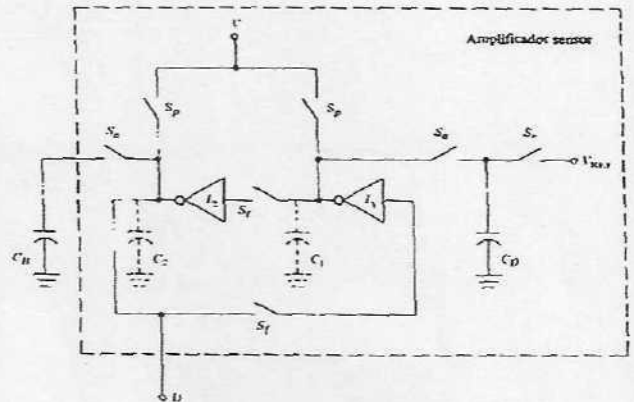
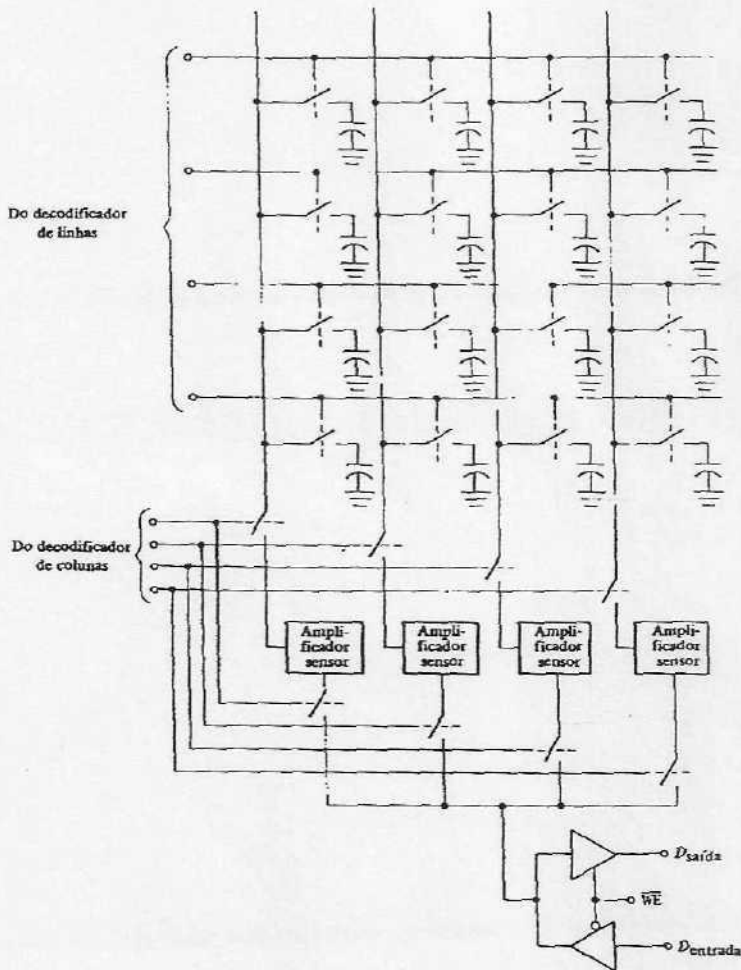
Uma EPROM (Erasable PROM, ou seja, PROM apagável) pode ser apagada porque, em vez de diodos e "fusíveis", tem transistores a efeito de campo com difusão de cargas que controlam a condução ou não condução do transistor, determinando se o bit é 1 ou 0. Por dependerem da presença de carga para ter diferentes bits, pode ser apagada por meio da incidência de forte luz ultra-violeta (e pode ser programada eletricamente, pela aplicação de tensões que vão colocar as cargas nos transistores adequados, para distinguir bit 0 de bit 1). Para isso ser possível, o envólucro do circuito integrado tem uma "janela" transparente por onde se pode ver a pastilha de semicondutor, que fica, normalmente, coberta por uma fita adesiva opaca; para se apagar a memória, retira-se a fita, expõe-se a memória à luz ultra-violeta e recoloca-se a fita.

Uma EEPROM (Electrically Erasable PROM) pode ser apagada por meio da aplicação de um pulso de tensão de valor suficientemente alto a um pino do circuito integrado (e pode ser reprogramada também eletricamente). Sobre uma RAM, tem a vantagem de também ter palavras armazenadas, lidas e apagadas de forma puramente elétrica e, além disso, é também memória permanente (enquanto a RAM é volátil). No entanto, embora toda operada eletricamente, não pode ser utilizada como memória principal (e muito menos como "cache") porque tem operações de escrita e de apagamento mais lentas que as da RAM (embora leitura mais rápida) e, o que é mais complicado, na operação de apagamento todas suas palavras são apagadas simultaneamente. Além disso, requer pulsos de tensões mais alta para esse apagamento; por isso, precisam de um circuito especial (com tensões especiais) para ser reprogramadas. Existem, porém, equipamentos que incluem circuito para reprogramar internamente a EEPROM utilizada para conter programa e dados de uso permanente (mas que podem ser atualizados quando conveniente).

DRAM - (RAM Dinâmica)

elemento de memória: pequeno capacitor feito no próprio semicondutor:

amplificador sensor:



Dois decodificadores endereçam matricialmente cada célula (na figura acima à esquerda só é mostrada uma célula de cada posição de memória); uma célula é formada por um capacitor construído no próprio semicondutor; a informação é guardada sob a forma de uma tensão (portanto, uma carga) no capacitor; o capacitor endereçado fica em contato com o "amplificador sensor" da respectiva coluna, que repassa a tensão para a saída (se for "leitura") ou repassa a tensão de entrada para o capacitor (se for "escrita", armazenagem).

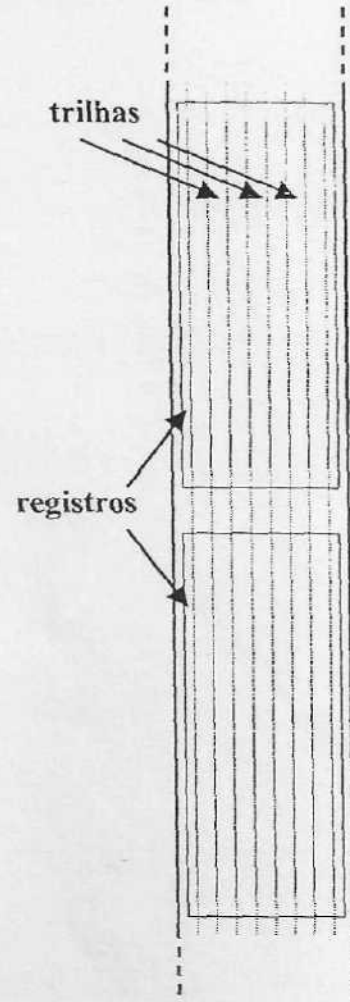
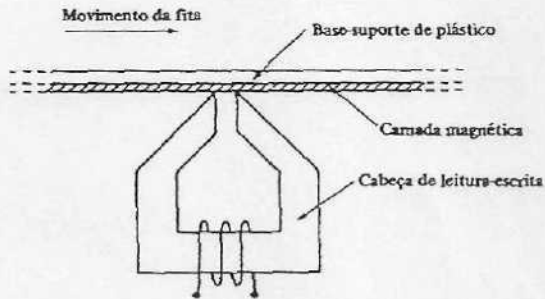
Como um capacitor assim construído tende a se descarregar rapidamente, a carga precisa ser repostada freqüentemente; isso é feito pelo "amplificador sensor", que, de tempo em tempo, verifica se a tensão está acima ou abaixo de um limiar e aplica a tensão total correspondente (por exemplo, 5 V ou 0 V).

Funcionamento do "amplificador sensor" da DRAM: Na figura acima à direita é mostrado o funcionamento de cada "amplificador sensor"; as chaves (na verdade, transistores que podem ser postos a conduzir ou cortados) fecham e abrem de modo a fazer o seguinte, sucessivamente: equalizar tensões nas saídas dos inversores (S_p), carregar C_D com a tensão de referência (limiar entre tensão de bit 0 e de bit 1) (S_e) e completar o "latch" e conectar o capacitor C_B ao circuito (S_a e S_r). Feito isso, o estado do "latch" tende para um dos extremos, determinado por quem tinha tensão (portanto, carga) maior: C_B ou C_D .

Fitas magnéticas

Fitas magnéticas "tradicionais" de computadores: 9 "trilhas" em paralelo (antigas: 7 trilhas) - palavras de 9 bits; palavras uma em seguida à outra, formando "registro", cada registro lido de uma vez:

gravação na superfície da fita é feita por cabeça magnética com bobina por onde circula corrente elétrica para produzir campo magnético, que magnetiza superfície da fita, revestida por material magnético (ferrite, etc); leitura: campo magnético variando (movimento da fita) causa indução de tensão na bobina da cabeça de gravação e leitura:



Discos magnéticos

Superfície circular revestida de material magnético; cabeça de gravação pode se posicionar em vários círculos (trilhas), discos empilhados: pilhas formam "cilindros"; informação é gravada em "setores", nas trilhas dos cilindros; física da gravação semelhante à da fita:

