



SISTEMAS DIGITAIS E ARQUITETURA DE COMPUTADORES

CIRCUITOS COMBINATÓRIOS/Combinacionais

Módulo 3



CIRCUITOS COMBINATÓRIOS

Introdução

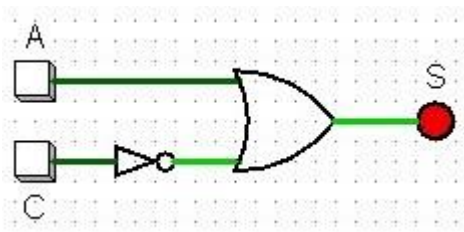
Informações básicas

O que é um circuito combinatório

Um circuito combinatório é um circuito digital que possui uma ou mais saídas que dependem apenas do estado ATUAL da(s) entrada(s).

Isto é, as entradas do circuito produzem um resultado “instantâneo” na saída. Só não é instantâneo por conta do atraso das portas lógicas, como foi discutido na aula 4.2.

Na aula anterior, os circuitos que foram mostrados eram circuitos combinatórios. Veja um deles na imagem abaixo:



Observe que a saída S depende apenas do estado atual da saída A e da saída C . Sendo mais específico, S será 1 apenas se A for 1 e C for 0. Para qualquer outra combinação destas entradas, a saída S é 0. E o estado atual de A ou C não afeta o estado futuro da saída.

Se você lembrar bem, o circuito acima foi resultado de uma simplificação de uma expressão que também continha uma entrada B . Entretanto, a entrada B não apareceu no circuito final.

Isto quer dizer que B é desnecessário para o circuito final. Portanto, apesar da entrada B existir no contexto do circuito, ela pode ser ignorada e inutilizada. No tópico abaixo estas coisas farão mais sentido.

Significado real das entradas

As entradas são, na verdade, indicadores de condições. Por exemplo, se são 17h; se a caixa d'água está cheia; se a porta do carro está fechada; se a temperatura é maior que 30°C; se um botão foi pressionado...

Ou seja, as entradas indicam o estado de alguma coisa do mundo real. No momento, não precisa se preocupar em saber qual é o circuito que gera o sinal digital dizendo que “a caixa d'água está cheia” por exemplo.

Pois, para montar estes circuitos, é necessário utilizar a eletrônica analógica, sistemas embutidos ou até a própria eletrônica digital. E, não é objetivo deste curso falar sobre isso no



momento. Portanto, trataremos estes circuitos como caixas pretas que informam o estado das coisas com sinais digitais (0 ou 1).

Lógica da entrada: positiva e negativa

As entradas de um circuito podem ter dois tipos de comportamento: lógica positiva e lógica negativa. A entrada possui lógica positiva se ela for 1 quando a condição foi cumprida, e 0 senão. E, lógica negativa, quando for o contrário.

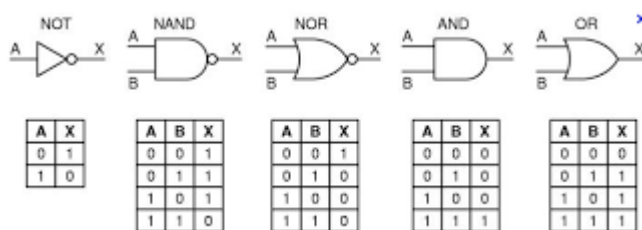
Exemplo: uma entrada que informa se a porta do carro está fechada. Se ela for lógica positiva, a entrada será 1 se a porta está fechada e 0 se ela está aberta. E o contrário para a lógica negativa: 0 se a porta está fechada e 1 se a porta está aberta.

A entrada poderia ser “porta do carro aberta”, aí a lógica seria o contrário da analisada acima. Tudo depende de como a entrada é definida.

Os circuitos combinatórios, por definição, são circuitos cujo valor lógico das suas saídas, depende exclusivamente do valor lógico das suas entradas.

Com o aumento da complexidade exigida por alguns projetos, houve necessidade de integrar em circuitos únicos, alguns circuitos combinatórios cujas funções por eles realizadas são muito úteis na simplificação desses projetos. Esses circuitos permitem implementar funções típicas em projetos de sistemas digitais.

Os circuitos mais utilizados e conhecidos são o **comparador**, o **descodificador BCD-7 segmentos**, o **multiplexer**, o **demultiplexer**, o **decoder**, e o **encoder**.



O Comparador

Comparadores são circuitos que detetam se dois números de n bits são iguais e, sendo diferentes, qual é o maior.

Este circuito, tal como o nome indica, faz a comparação entre dois valores binários. A versão que vamos analisar compara dois valores de quatro bits cada um, como se mostra na figura seguinte:



	Entradas		Saída 1	Saída 2	Saída 3	Saída 4
Posição	A	B	$A < B$	$A = B$	$A > B$	$A < > B$
0	0	0				
1	0	1				
2	1	0				
3	1	1				
Expressão						

Comparadores XOR: A porta XOR (ou exclusivo) pode ser considerada como um comparador elementar de 1 bit. Na realidade, na sua saída teremos o valor lógico 0 se as duas entradas forem iguais e o valor lógico 1 se as entradas forem diferentes.

	Entradas		Saída 1	Saída 2	Saída 3	Saída 4
Posição	A	B	$A < B$	$A = B$	$A > B$	$A < > B$
0	0	0				
1	0	1				
2	1	0				
3	1	1				
Expressão						

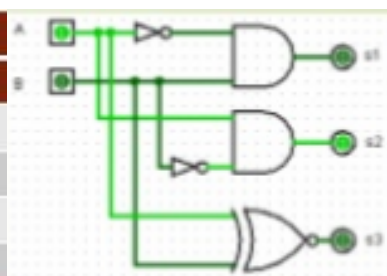


Comparadores XNOR: A porta XNOR (ou coincidência) pode ser considerada como um comparador elementar de 1 bit. Na realidade, na sua saída teremos o valor lógico 1 se as duas entradas forem iguais e o valor lógico 0 se as entradas forem diferentes.

	Entrada		Saída 1	Saída 2	Saída 3	Saída 4
Posição	A	B	A<B	A=B	A>B	A<>B
0	0	0				
1	0	1				
2	1	0				
3	1	1				
Expressão						

Comparadores para n°s de 1 Bit A figura representa um exemplo para dois números de 1 bit.

	Entradas		Saída 1	Saída 2	Saída 3
Posição	A	B	A<B	A=B	A>B
0	0	0	0	1	0
1	0	1	1	0	0
2	1	0	0	0	1
3	1	1	0	1	0
Expressão			$\overline{A}B$	$\overline{A}\overline{B}+AB$ $A \oplus B$ XNOR	$A\overline{B}$





Comparadores nºs 2 Bits. A figura representa um exemplo para dois números de 2 bits. O LSD (Low Significant Digit) e o MSD (Most Significant Digit) assumem importância na medida em que, a posição dos bits é importante para a comparação.

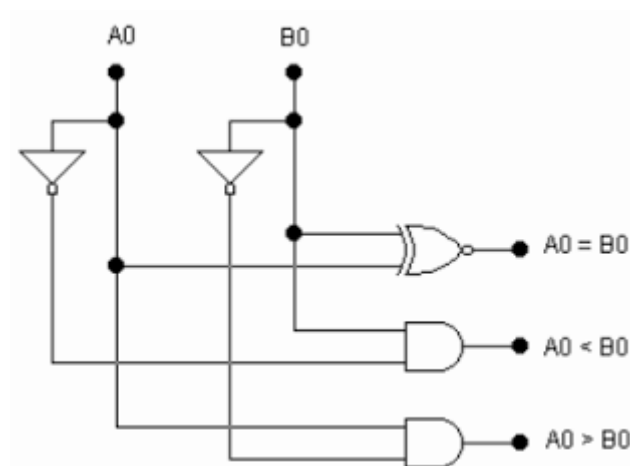
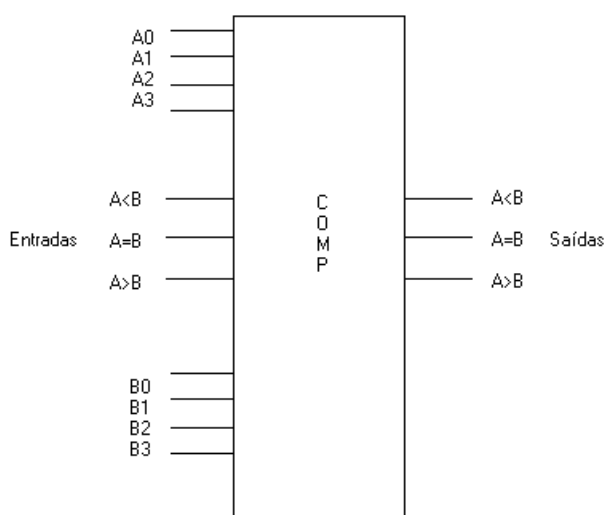
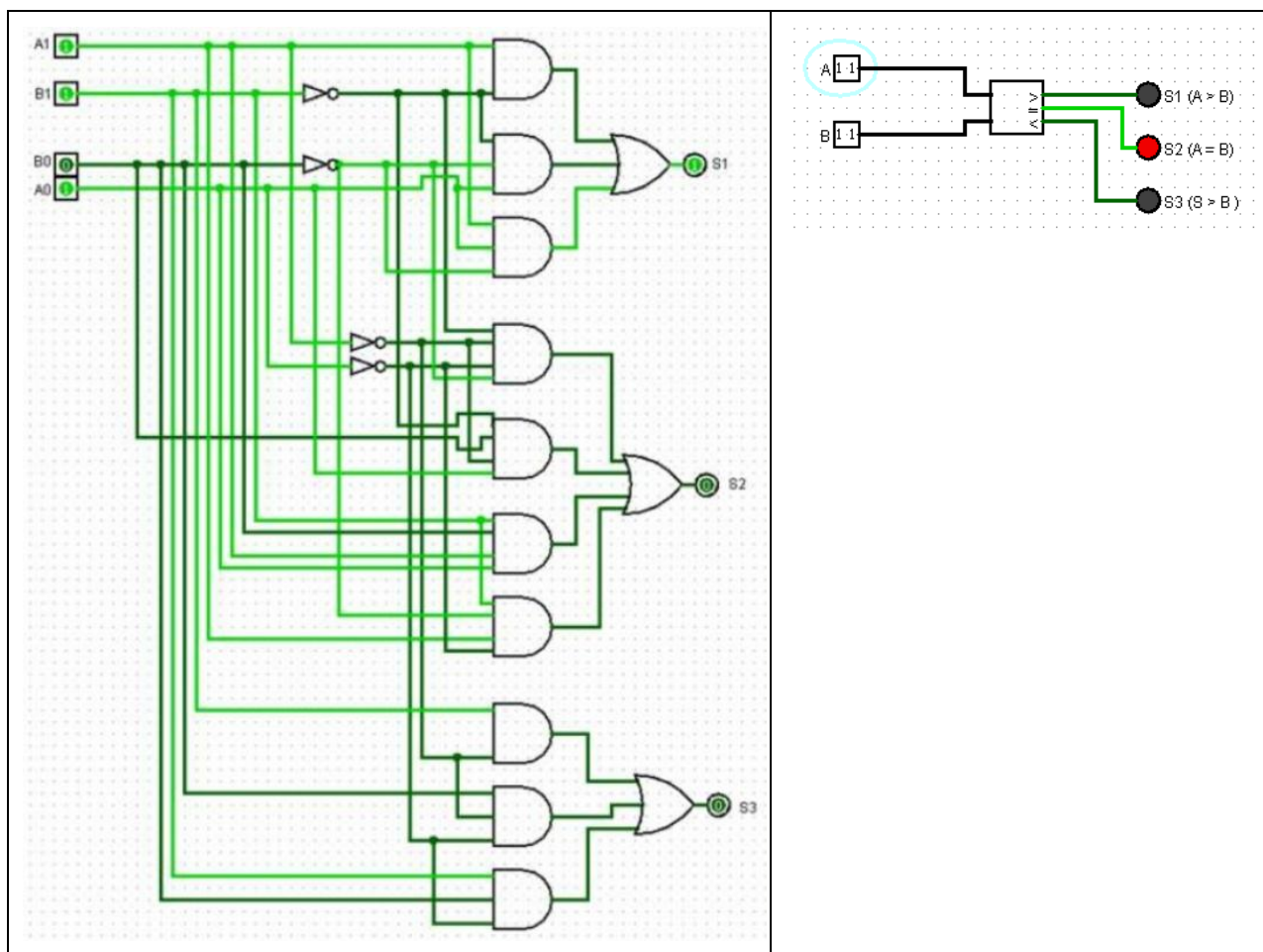
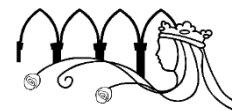
Pós.	Entradas				Saídas			Expressão Booleana Soma de produtos (SDP)
	A		B		S1	S2	S3	
	A1	A0	B1	B0	A > B	A = B	A < B	
0	0	0	0	0	0	1	0	
1	0	0	0	1	0	0	1	
2	0	0	1	0	0	0	1	
3	0	0	1	1	0	0	1	
4	0	1	0	0	1	0	0	
5	0	1	0	1	0	1	0	
6	0	1	1	0	0	0	1	
7	0	1	1	1	0	0	1	
8	1	0	0	0	1	0	0	
9	1	0	0	1	1	0	0	
10	1	0	1	0	0	1	0	
11	1	0	1	1	0	0	1	
12	1	1	0	0	1	0	0	
13	1	1	0	1	1	0	0	
14	1	1	1	0	1	0	0	
15	1	1	1	1	0	1	0	



	$\overline{B1}.\overline{B0}$	$\overline{B1}.B0$	$B1.B0$	$B1.\overline{B0}$
$\overline{A1}.\overline{A0}$	0	0	0	0
$\overline{A1}.A0$	1	0	0	0
$A1.A0$	1	1	0	1
$A1.\overline{A0}$	1	1	0	0
S1 =				

	$\overline{B1}.\overline{B0}$	$\overline{B1}.B0$	$B1.B0$	$B1.\overline{B0}$
$\overline{A1}.\overline{A0}$	1	0	0	0
$\overline{A1}.A0$	0	1	0	0
$A1.A0$	0	0	1	0
$A1.\overline{A0}$	0	0	0	1
S2 =				

	$\overline{B1}.\overline{B0}$	$\overline{B1}.B0$	$B1.B0$	$B1.\overline{B0}$
$\overline{A1}.\overline{A0}$	0	1	1	1
$\overline{A1}.A0$	0		1	1
$A1.A0$	0	0	0	0
$A1.\overline{A0}$	0	0	1	0
S3 =				



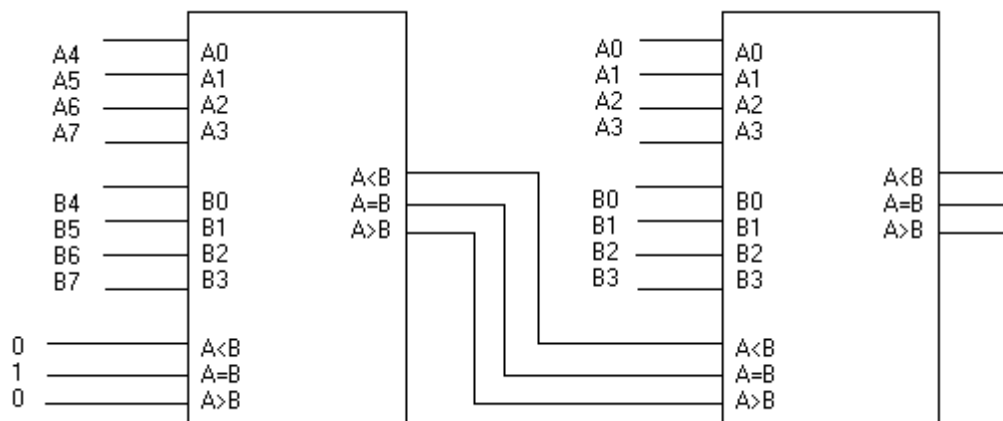


Como se pode observar, no circuito anterior, as entradas $A_0 \dots A_3$ e $B_0 \dots B_3$, que são os valores binários a comparar. Coloca três saídas disponíveis, $A < B$, $A = B$ e $A > B$, ativando uma e uma só destas saídas consoante o valor colocado em A seja menor, igual ou maior que o valor colocado em B.

As entradas $A < B$, $A = B$ e $A > B$, servem para efetuar a concatenação de circuitos destes, para se realizar módulos comparadores de valores com mais bits.

Se se pretender usar um único circuito destes (caso de se comparar dois números de 4 bits cada), deve-se colocar as entradas $A < B$, $A = B$, $A > B$ com 0, 1 e 0 respetivamente, pois o circuito comparador tem estas entradas para ser informado do resultado da comparação de um módulo anterior, que neste caso não existiria, e por isso, deve-se colocar nestas entradas estes valores para indicarmos ao circuito que o resultado da comparação anterior (inexistente) foi $A=B$, pois só assim ele fará a comparação.

Para se entender o atrás referido, tem que se perceber que se comparar dois valores, de 8 bits, por exemplo, compara-se a parte mais alta desses valores, e se a comparação der diferente, é esse resultado $A < B$ ou $A > B$, que é o resultado final. A comparação das partes baixas, só se efetua se o resultado da comparação das partes altas for $A = B$. O comparador foi desenhado para poder ser concatenado, e por isso tem essas entradas. Se não se ligar com mais nenhum, deve-se colocar nas entradas os valores referidos anteriormente, pois só assim ele fará a comparação, senão dará o resultado colocado nas entradas. A figura seguinte mostra a ligação de dois comparadores de 4 bits para se obter um módulo de comparação de 8 bits.



O resultado da comparação da parte alta informa o comparador da parte baixa se deve ou não comparar essa parte, pois só se a comparação anterior der igual é que a parte baixa deve ser comparada para se obter dos valores de 8 bits.



Multiplexer

Um circuito multiplexer é um circuito combinatório que coloca como valor lógico da sua saída o valor lógico que está presente na entrada selecionada pelos seus bit's de seleção. Este circuito tem várias entradas, uma só saída e bit's que consoante a valor assumido definem a valor da saída. São designados como bit's de seleção. A figura seguinte apresenta um multiplexer de 4 entradas, como tal, este circuito necessitado dois bits de seleção (S0 e S1).

O Mux (Multiplexer) básico de 2 entradas (2X1) • A partir de uma entrada de Seleção, direciona a informação I0 ou I1 para a saída (pode ser um sinal de clock, bit de informação de dados etc.).

Note-se que existem circuitos multiplexer com mais entradas, aumentando também nesses circuitos o número de bits de seleção.

Em relação à figura anterior, tomemos o seguinte exemplo: se na entrada "0", estiver o valor lógico "1" será necessário colocar em S0 e S1 os valores lógicos 0 e 0 para que a saída tome o valor lógico da entrada "0" que será 1. Se em S0 e S1 estiver outra combinação que não a anterior, a saída do multiplexer tomará o valor lógico da entrada correspondente à combinação formada por S0 e S1. Vejamos a seguinte tabela de verdades que exemplifica o funcionamento deste circuito multiplexer:

Bit's de seleção		Entradas				Saída
S0	S1	"0"	"1"	"2"	"3"	S
0	0	0	-	-	-	0
0	0	1	-	-	-	1
1	0	-	0	-	-	0
1	0	-	1	-	-	1
0	1	-	-	0	-	0
0	1	-	-	1	-	1
1	1	-	-	-	0	0
1	1	-	-	-	1	1

Como se pode verificar pela tabela:

- ☞ Quando $S0 = 0$ e $S1 = 0$, independentemente dos valores das outras entradas, a saída S tomará o valor lógico da entrada "0".
- ☞ Quando $S0 = 1$ e $S1 = 0$, Independentemente das outras entradas, a saída S assume o valor da entrada "1".
- ☞ Quando $S0 = 0$ e $S1 = 1$, Independentemente das outras entradas, a saída S assume o valor da entrada "2".



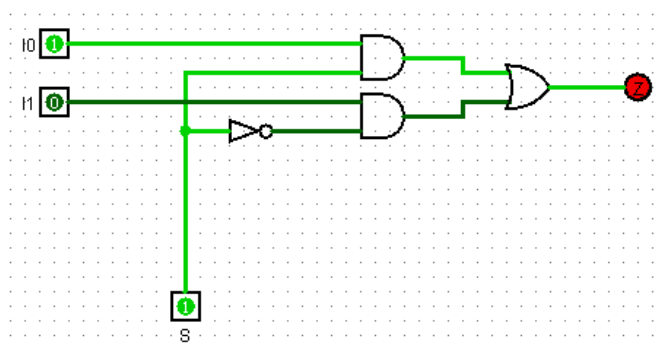
- Quando $S_0 = 1$ e $S_1 = 1$, Independentemente das outras entradas, a saída S assume valor da entrada "3".

Dois assuntos importantes a estudar sobre este circuito:

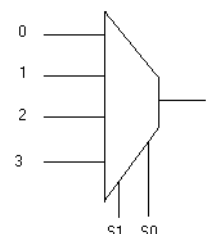
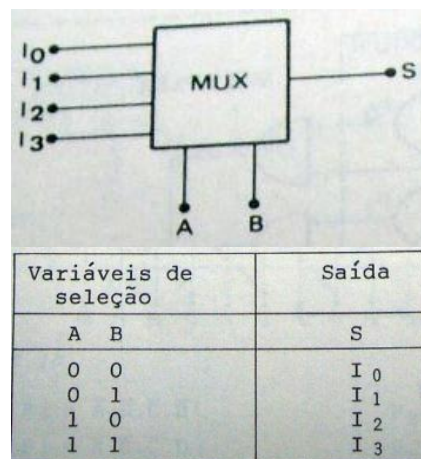
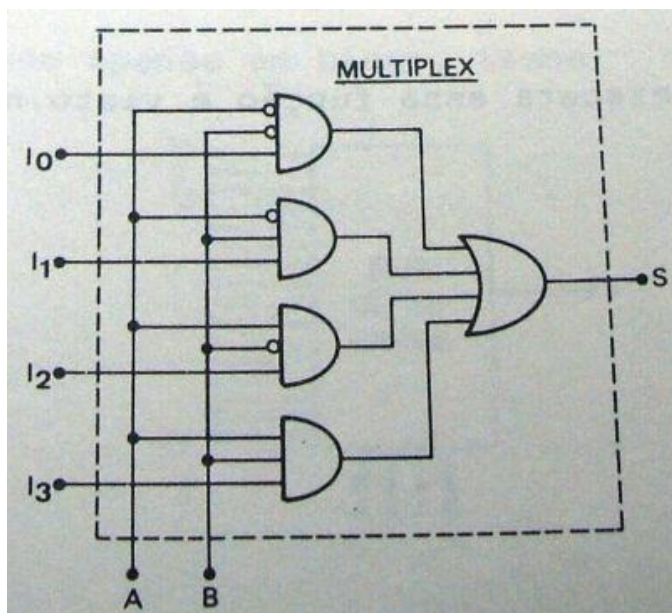
- Como concatenar vários multiplexers para obter um multiplexer com mais entradas
- Como implementar funções lógicas a partir de um multiplexer.

Exercício:

Implemente o seguinte circuito no logisim e elabora a tabela de verdade de forma a obter a expressão Z .

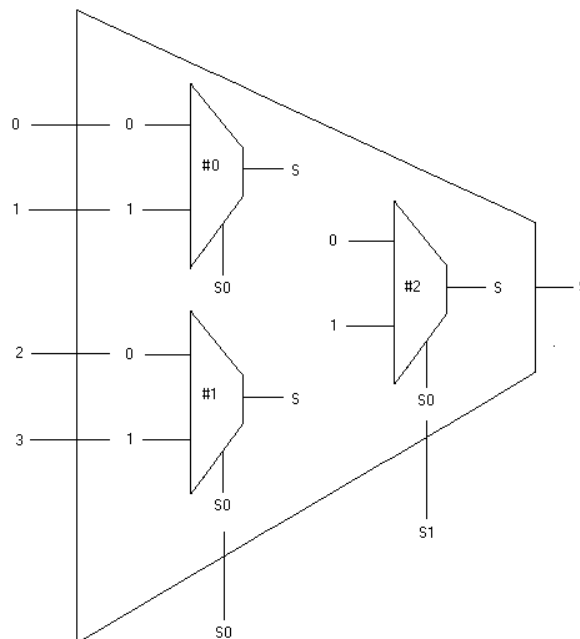


Mux de 4 entradas (Utiliza-se da mesma ideia básica de um mux de 2 entradas)





A figura seguinte mostra como se pode construir um multiplexer de 4 entradas usando três multiplexers de duas entradas. Esta concatenação é feita de uma forma lógica e simples.



O S0 do multiplexer #0 e o S0 do multiplexer #1 estão ligados efetuando o S0 do conjunto.

O S0 do multiplexer #2 fica disponível para fazer o S1 do conjunto. O S0 do conjunto atua nos multiplexer #0 e #1 escolhendo em cada um deles a entrada a colocar nas suas saídas, e o S1 do conjunto (S0 do #2) escolhe qual dos multiplexers anteriores coloca na saída do conjunto o valor lógico. Assim ficamos no final com um multiplexer de 4 entradas, usando a concatenação de três multiplexers de duas entradas.



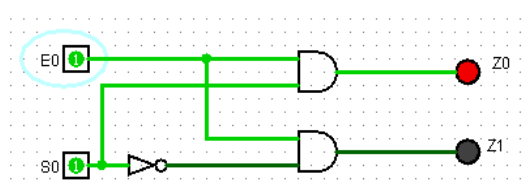
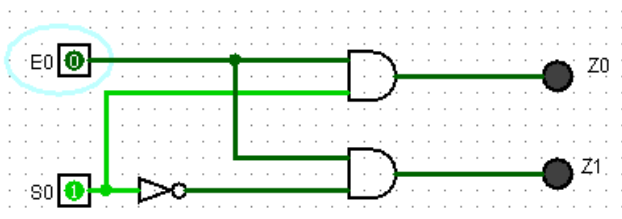
Demultiplexer

Este circuito faz a operação inversa do multiplexer. Coloca na saída selecionada pelos bits de seleção o valor lógico da entrada. Tem, portanto, várias saídas, os bits de seleção e uma só entrada. A figura abaixo representa um demultiplexer de 4 saídas. Para se escolher em qual das saídas se coloca o valor da entrada necessita de dois bits de seleção (S_0 e S_1) tal como seria de esperar. As saídas são designadas por "0", "1", "2" e "3" e a cada combinação de S_0 e S_1 corresponde a colocação do valor lógico da entrada nessa saída.

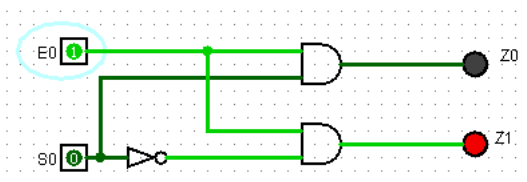
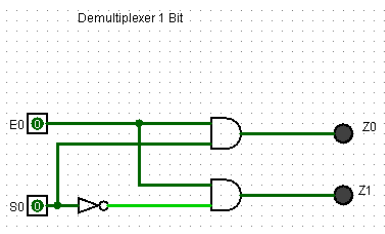
Se por exemplo, na entrada do demultiplexer estiver o valor lógico 1, se em S_0 e S_1 estiverem os valores 0 e 1 respetivamente, na saída 2 aparecerá o valor lógico 1, pois é esta a saída selecionada. O estado das outras saídas permanecerá a 0.

Demultiplexer de 1 bit de seleção

Se $S_0 = 1$ a saída Z_0 vai assumir o valor de E_0

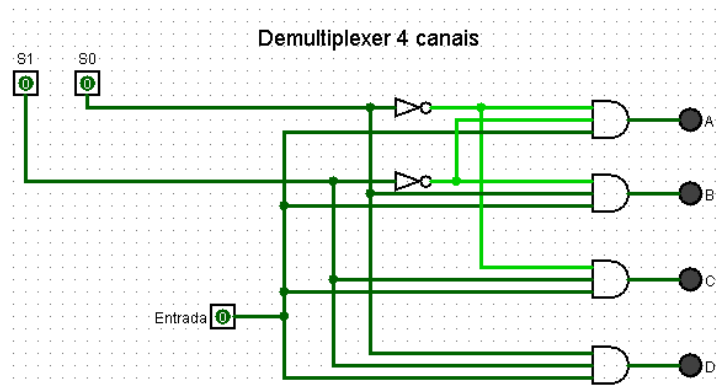
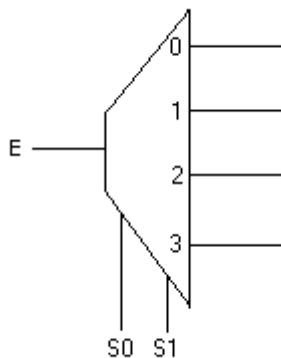


Se $S_0 = 0$ a saída Z_1 vai assumir o valor de E_0





Analise o seguinte circuito e preencha a tabela de verdade que explica o funcionamento de um circuito demultiplexers com 2 bit's de seleção:

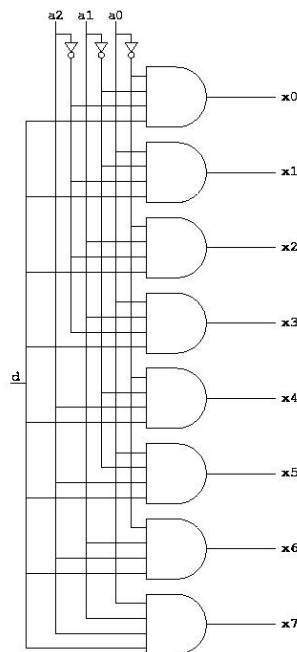


Bit's seleção		Entrada	Saídas			
S0	S1		A	B	C	D
0	0	E	E	x	x	X
0	1		x	E	x	x
1	0		x	x	E	x
1	1		x	x	x	E

X significa que as saídas permanecem inalteradas.

Demultiplexer 3 bits seleção (8 canais)

Bit's seleção			Entrada	Saídas							
S2	S1	S0		X0	X1	X2	X3	X4	X5	X6	X7
0	0	0	E	E	x	x	x	x	x	x	x
0	0	1		x	E	x	x	x	x	x	x
0	1	0		x	x	E	x	x	x	x	x
0	1	1		x	x	x	E	x	x	x	x
1	0	0		x	x	x	x	E	x	x	x
1	0	1		x	x	x	x	x	E	x	x
1	1	0		x	x	x	x	x	x	E	x
1	1	1		x	x	x	x	x	x	x	E

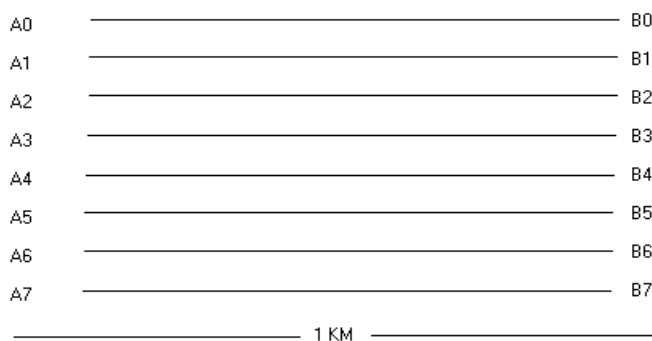




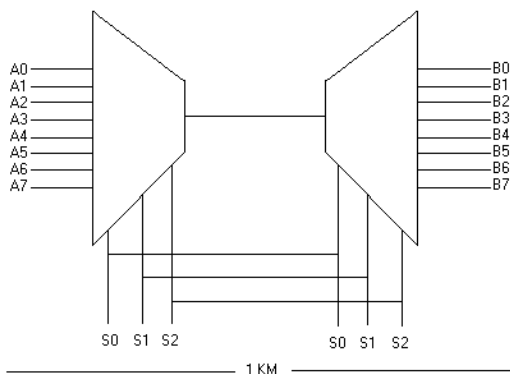
Aplicação Prática

O uso do multiplexer e do demultiplexer em conjunto pode facilitar e simplificar a comunicação entre dois pontos, senão vejamos:

Imagine que pretende efetuar uma comunicação entre dois pontos com 8 bits de dados. Esses dois pontos distam 1000 metros um do outro. Como teríamos de usar 8 fios (um para cada bit), teríamos 8000 metros de fio para a comunicação.



Podemos reduzir o comprimento de fio gasto e em consequência os custos usando um circuito multiplexer e outro demultiplexer como se mostra abaixo:

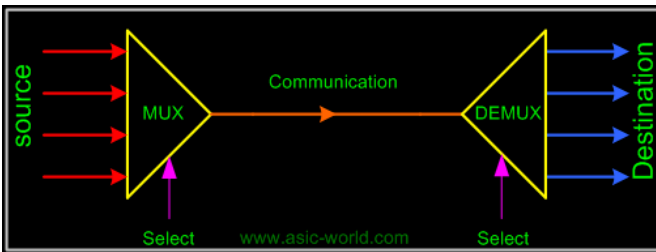
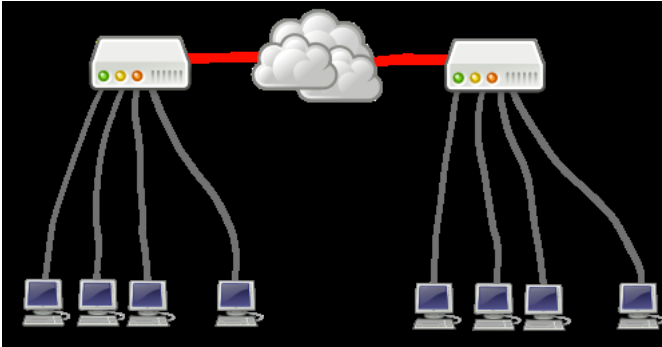
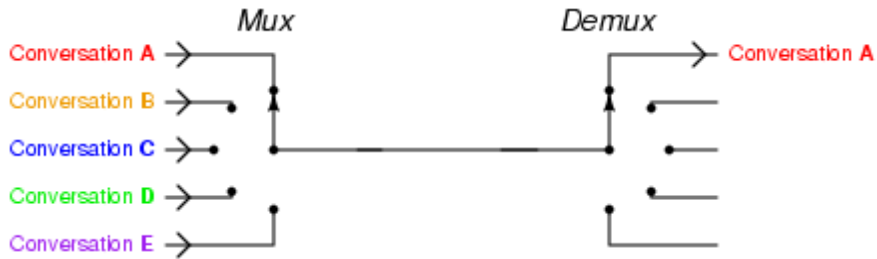


Este circuito usa só 4 fios para a comunicação o que reduz para metade os custos desta comunicação.

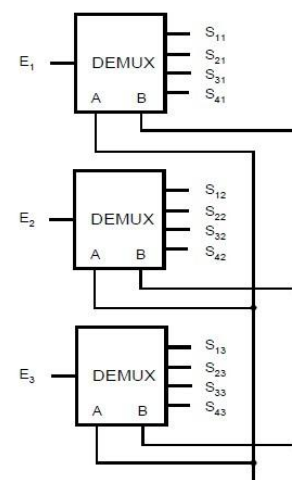
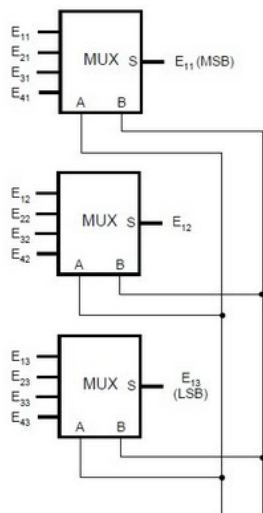
Gerando ciclicamente os valores de 0 a 7 em binário em S0, S1 e S2, o multiplexer passará para a sua saída os valores de A0 ... A7 e o demultiplexer colocará os valores recebidos na sua entrada em B0..B7 na mesma ordem pois S0, S1 e S2 são comuns aos dois circuitos. A este tipo de comunicação chama-se comunicação série porque enviam-se os bits um de cada vez pela mesma linha de dados.

Os circuitos multiplexer e demultiplexer têm três variáveis de seleção (S0, S1 e S2) porque têm 8 entradas e 8 saídas respetivamente. Como $2^3 = 8$, logo temos de ter 3 bits de seleção em cada um deles.

Se usarmos uma comunicação de 16 bits (16 Km de fio) em modo paralelo, podemos usar apenas 5 fios (5 Km) na mesma comunicação em modo série pois usamos multiplexer e demultiplexer de 4 bits de seleção ($2^4 = 16$). Neste caso a poupança ainda era maior.



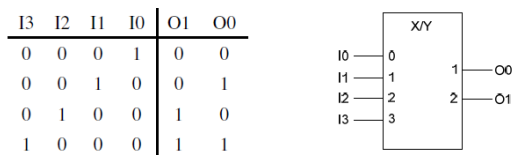
Multiplexador/Demultiplexador: Associação





Encoder (Codificador)

O CODIFICADOR binário é um circuito combinatório que indica qual das entradas possíveis é que está ativa.

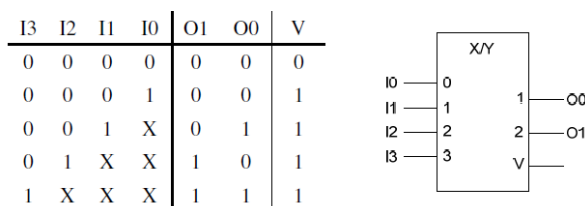


Nesta versão simples, o codificador só considera 4 das 16 combinações possíveis de entrada.

- O circuito não distingue a situação de todas as entradas estarem a “0”.
- O circuito não distingue as situações em que estão a “1” mais do que uma entrada.

CODIFICADOR DE PRIORIDADE

As entradas deste codificador têm uma ordem de prioridades: em caso de haver mais de uma entrada ativa, o circuito considera ativa a entrada de maior prioridade.



Codificadores 8 para 3

Entradas ativas	S2	S1	S0
E0	0	0	0
E1	0	0	1
E2	0	1	0
E3	0	1	1
E4	1	0	0
E5	1	0	1
E6	1	1	0
E7	1	1	1

Com base na tabela anterior calcule as expressões das saídas S0, S1, S2

S2=

S1=

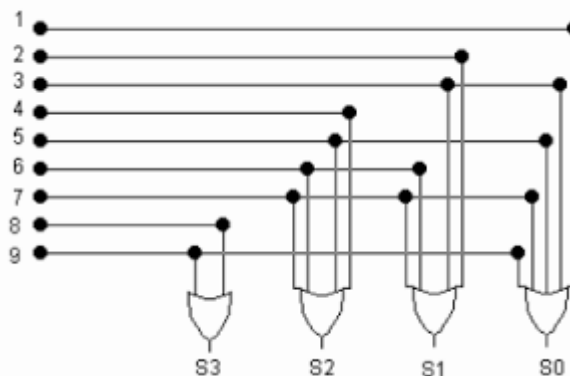
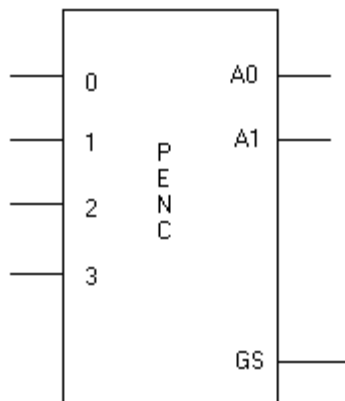
S0=

Se pretendermos construir um circuito codificador com prioridade temos que adicionar uma saída que possa definir a prioridade.

$$IDLE = \overline{E0} \cdot \overline{E1} \cdot \overline{E2} \cdot \overline{E3} \cdot \overline{E4} \cdot \overline{E5} \cdot \overline{E6} \cdot \overline{E7}$$



Este circuito faz a operação inversa do decoder. Por isso, coloca nas suas saídas a palavra em binário correspondente à entrada ativa. No caso da figura seguinte, existem quatro entradas e, logicamente, a palavra de saída terá dois bits, A0 e A1.



Duas questões tiveram de ser resolvidas aquando de implementação deste circuito.

A primeira tem a ver com o facto de se ativar duas ou mais entradas. Esta é resolvida com uma codificação com prioridades, ou seja, estabelecendo um critério de prioridades na codificação das entradas (por isso estes circuitos são mais designados por Priority Encoders, ou PENC como abreviatura). Um dos critérios que se podem estabelecer, é o de se dar prioridade às entradas de menor peso. Assim se a entrada de peso 0 é a mais prioritária, e por isso sempre que ela esteja ativa, o valor na saída é 00, independentemente se mais alguma das outras entradas estiver ativa. A entrada seguinte em termos de prioridade é a 1, e assim, se ela estiver ativa, juntamente com a 2 ou 3, o valor da saída é 01, e assim sucessivamente. É este o critério usado normalmente por este circuito.

Entradas	3	2	1	0
Prioridade	–		+	

A outra questão prende-se com a circunstância de não haver nenhuma entrada ativa no circuito. Qual o valor escolhido para a saída, quando não há nenhuma entrada ativa no circuito? Será bom escolher o 00? Assim, como distinguir entre não haver entrada ativa e estar ativa a entrada 0? É por isso que existe a saída GS (groupselect) que se ativa quando pelo menos uma das entradas está ativa. Assim, se não se ativar nenhuma entrada, A0 = 0 e A1 = 0, e GS = 0. Se se ativar qualquer entrada, A0 e A1 têm a codificação dessa entrada e GS = 1. Se se ativar a entrada 0 A0 = 0, A1 = 0 e GS = 1. O GS serve para se fazer a distinção entre a ativação da entrada 0 e a não ativação de qualquer entrada.



Codificador Decimal

E ₉	E ₈	E ₇	E ₆	E ₅	E ₄	E ₃	E ₂	E ₁	E ₀	S ₃	S ₂	S ₁	S ₀
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

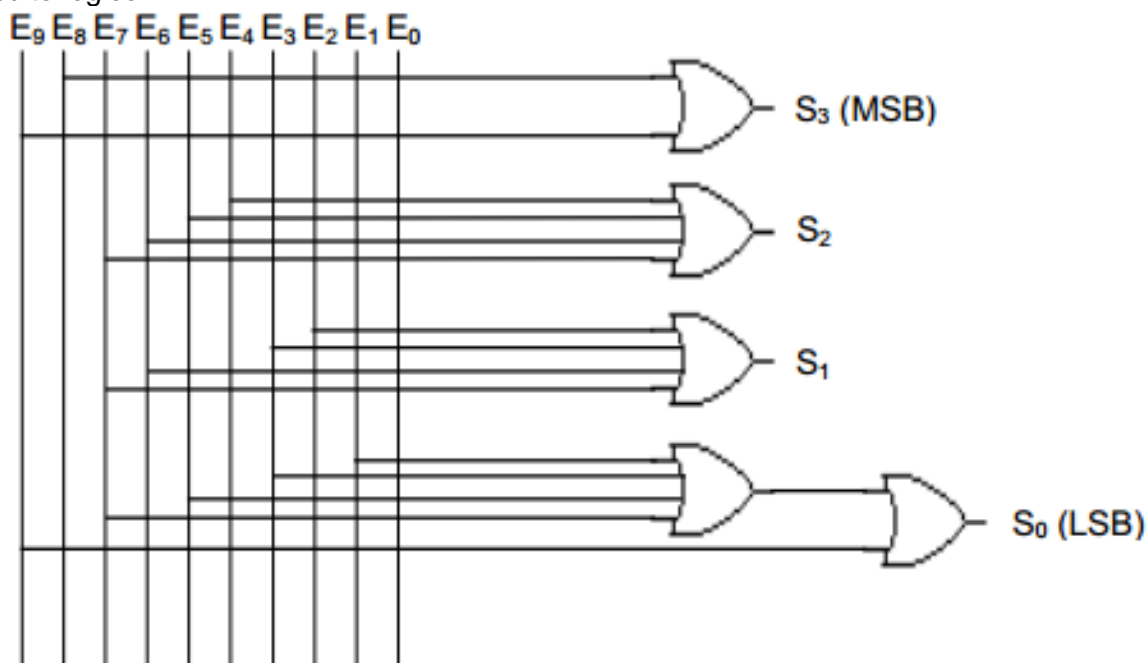
$$S_3 = E_8 + E_9$$

$$S_2 = E_4 + E_5 + E_6 + E_7$$

$$S_1 = E_2 + E_3 + E_6 + E_7$$

$$S_0 = E_1 + E_3 + E_5 + E_7 + E_9$$

Circuito lógico

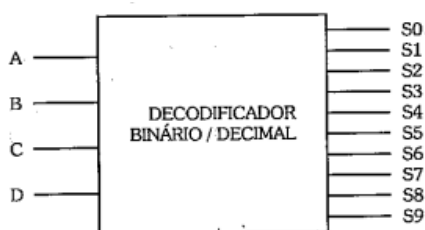




Decoder(Decodificador)

- É um dispositivo que faz o inverso do codificador, desfazendo a codificação. Assim, a informação original que foi codificada pode ser recuperada.
- É um circuito lógico que converte um código binário de N bits que lhe é apresentado como entrada, em M linhas de saída.
- Cada linha de saída será ativada por uma, e somente uma, das possíveis combinações dos bits de entrada.
- São ferramentas importantes nos projetos digitais pois são amplamente utilizados para selecionar memórias e realizar conversões de códigos e roteamento de dados.

A estrutura geral deste decodificador se ilustra na figura

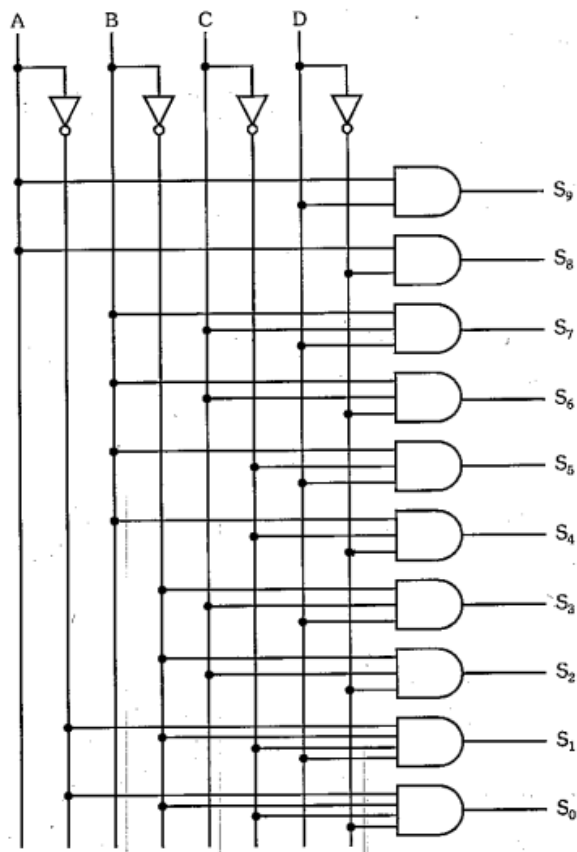


Decodificador binário/decimal

As entradas deste decodificador são o código BCD8421 e as saídas são o código 9876543210

BCD 8421				Código 9876543210									
A	B	C	D	S9	S8	S7	S6	S5	S4	S3	S2	S1	S0
0	0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0	0	0	0	0

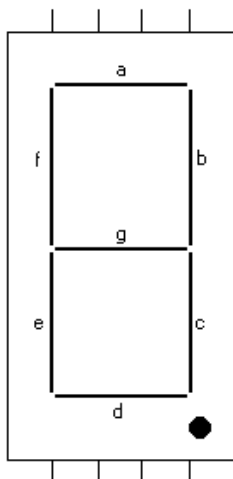
- Cada expressão de saída S0 até S9 deve ser simplificada
- Para construir o circuito simplificado deve-se ter em conta que o código BCD não possui valores maiores que 9.
- Portanto, os valores maiores a 9 são termos don't care.



1.7 - O Descodificador BCD-7 Segmentos

Este tipo de circuitos atua num display de 7 segmentos. Um valor binário a quatro bits é colocado nas suas entradas e os segmentos (leds) do display acendem formando o dígito em decimal correspondente. Sendo assim, um descodificador deste tipo “ataca” um só display, sendo portanto necessários quantos circuitos destes quantos os displays ou dígitos se queiram controlar. As entradas deste circuito são a 4 bits mas só os valores de 0 a 9 se podem introduzir nas entradas binárias caso contrário o descodificador não responde bem.

Na figura seguinte, está um esboço de um display de 7 segmentos, onde há um pino para cada segmento incluindo também um pino para o ponto e outro para o comum aos leds.



As saídas do decodificador devem atuar os pinos correspondentes do display. O decodificador é construído através de uma tabela de verdades com as quatro entradas (A0...A3) e as 7 saídas (a..g). Consoante o valor de entrada, consoante as saídas se devem ativar ou não de modo a desenharem no display o número correspondente ao valor colocado nas entradas. Assim se tiram as expressões de cada saída (segmento) e integram-se esses circuitos ficando assim o circuito final do decodificador BCD-7 segmentos.

Os displays existem em duas formas no mercado, cátodo comum e ânodo comum. Se o display for de ânodo comum, para se ativar um segmento deve-se colocar no pino correspondente um 0. Se for de cátodo comum, deve-se colocar um 1 para se ativar o segmento correspondente. Assim existem decodificadores com a ativação das suas saídas a 1 e outros que ao ativarem as suas saídas colocam-lhes 0.

Na figura seguinte, está representado um decodificador ânodo comum, pois as suas saídas estão negadas, ou seja entradas para se escrever o valor a representar, existem outras que permitem as seguintes operações:

LT - LampTest: quando ativo promove a ativação de todos os segmentos do display.

RBI - RippleBlanking Input: quando ativo, se o valor da entrada for 0 promove a desativação de todos os segmentos.

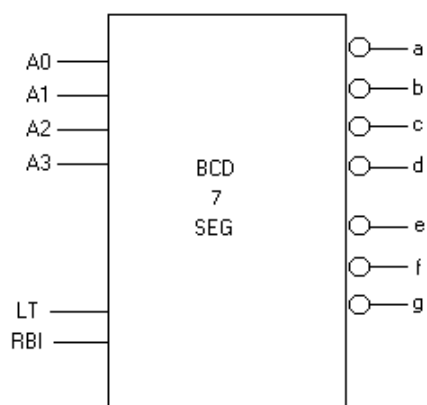
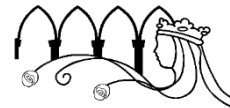


Tabela de ativação dos segmentos:

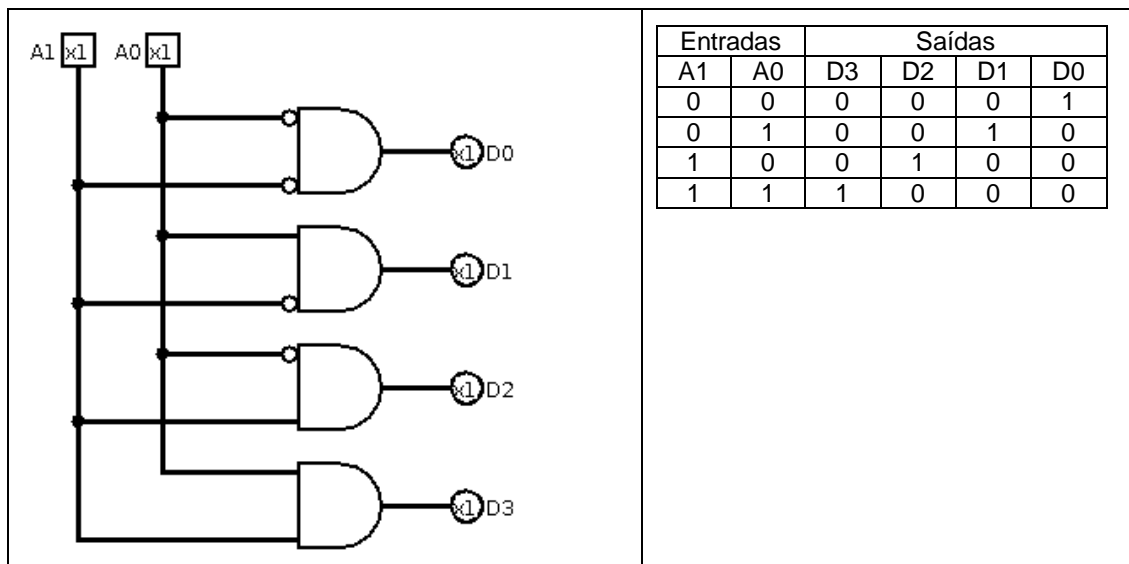
Não esquecer que as saídas se ativam a 0.

A3	A2	A1	A0	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	1	0	0	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0
0	0	1	1	0	0	0	0	1	1	0
0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	1	0	0	1	0	0
0	1	1	0	0	1	0	0	0	0	0
0	1	1	1	0	0	0	0	1	1	1
0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	1	0	0

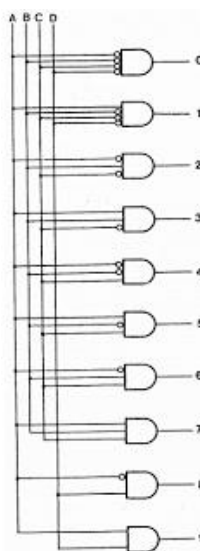
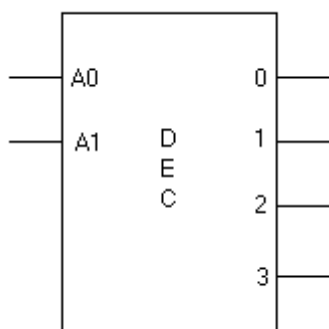


Circuito decodificador

O circuito abaixo é um decodificador de 2 linhas de entrada para 1 de 4 linhas de saída. Para cada combinação de entrada (2^2 possibilidades), apenas uma saída é selecionada.



1. Considere o seguinte circuito. Monte o **circuito** e escreva sua **expressão lógica** e **tabela verdade**



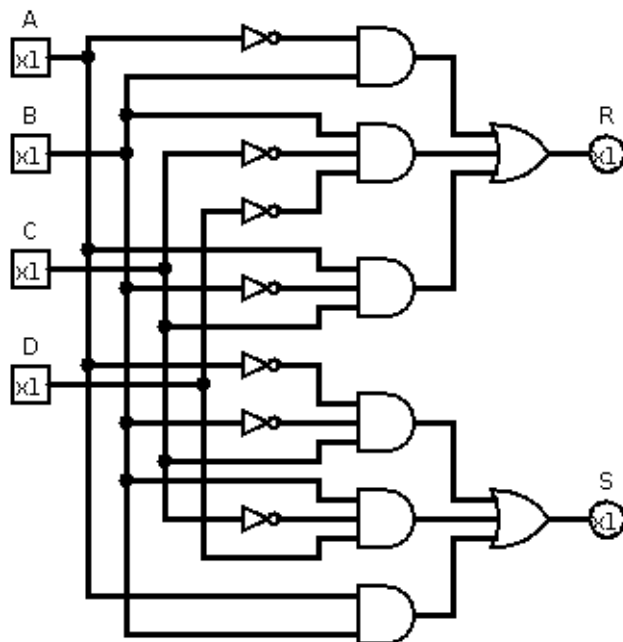
A seguinte tabela de verdades ilustra o funcionamento deste circuito:

A0	A1	"0"	"1"	"2"	"3"
0	0	1	0	0	0
1	0	0	1	0	0
0	1	0	1	1	0
1	1	0	1	0	1

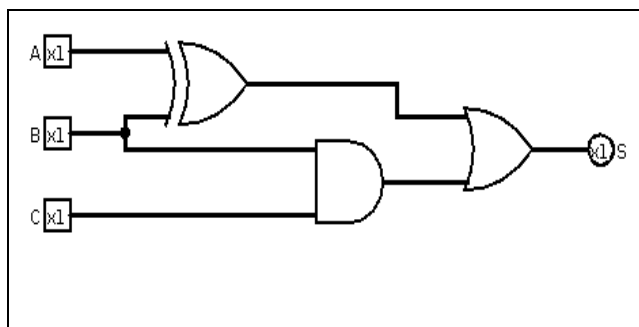
- ☞ Quando $A0 = 0$ e $A1 = 0$ ativa-se a saída "0" ficando as restantes desativas.
- ☞ Quando $A0 = 1$ e $A1 = 0$ ativa-se a saída "1" ficando as restantes desativas.
- ☞ Quando $A0 = 0$ e $A1 = 1$ ativa-se a saída "2" ficando as restantes desativas.
- ☞ Quando $A0 = 1$ e $A1 = 1$ ativa-se a saída "3" ficando as restantes desativas.



Construir os circuitos lógicos, simular e determinar a tabela verdade dos circuitos representados pelas



Entradas				Saídas	
A	B	C	D	R	S
0	0	0	0		
0	0	0	1		
0	0	1	0		
0	0	1	1		
0	1	0	0		
0	1	0	1		
0	1	1	0		
0	1	1	1		
1	0	0	0		
1	0	0	1		
1	0	1	0		
1	0	1	1		
1	1	0	0		
1	1	0	1		
1	1	1	0		
1	1	1	1		



Entradas			Saída
A	B	C	S
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

Problema de controlo

Um sistema de controlo de uma máquina monitoriza os valores de **temperatura** (T), **pressão** (P) e **rotações do motor** (R), da seguinte forma:

- Temperatura normal (Nível lógico 0) / Temperatura alta (Nível lógico 1).
- Pressão normal (Nível lógico 0) / Pressão alta (Nível lógico 1).
- Rotação normal (Nível lógico 0) / Rotação baixa (Nível lógico 1).

O sistema de controlo deve acionar uma lâmpada de advertência (LA) (Nível lógico 1), de forma a avisar o operador quando ocorrer uma OU outra das seguintes condições:

- rotação(R) normal **E**(And) temperatura(T) alta
- pressão(P) alta **E**(And) temperatura(T) alta

Monte um **circuito lógico** que atenda a estas condições, e escreva a **expressão lógica** e **tabela verdade**.



T	P	R	LA
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

$$S = T\bar{P}\bar{R} + TPR + TP\bar{R}$$
 | | | \bar{R} | R | |-------------------|-----------|---| | $\bar{T}.\bar{P}$ | 0 | 0 | | $\bar{T}.P$ | 0 | 0 | | $T.P$ | 1 | 1 | | $T.\bar{P}$ | 1 | 0 | $$S = T\bar{R} + TP$$ |

1.8 - Notas Finais

Os circuitos foram aqui apresentados de uma forma teórica. No final deste capítulo, encontram-se alguns circuitos existentes no mercado. Todos apresentam mais entradas do que aquelas a que me referi, será necessário referir-me agora a elas.

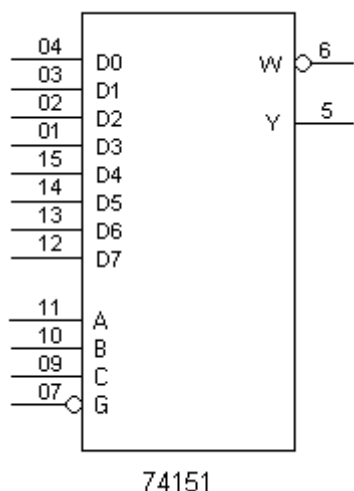
As entradas de "enable", ou "gate", abreviadas por "E", ou "G" são entradas que quando ativas, deixam o circuito funcionar normalmente, e quando não ativas inibem o funcionamento do circuito.

Também será necessário referir-me à natureza dos sinais tanto de entrada como de saída. Os sinais que têm uma negação são sinais que quando ativos têm presente o valor lógico 0, e os sinais que não têm essa negação, são sinais que quando ativos têm o valor lógico 1, tanto nas entradas como na saída. Os primeiros designam-se por sinais de natureza "active-low", e os segundos designam-se por sinais de natureza "active-high".



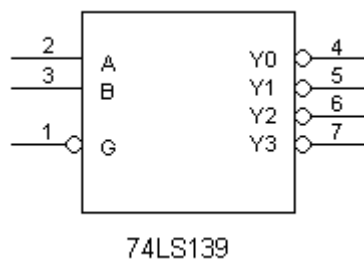
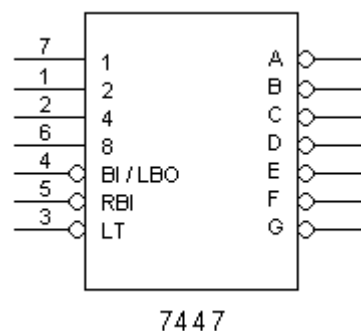
1.9 - Exemplos de Circuitos existentes no Mercado

Apresentam-se de seguida alguns circuitos combinatórios existentes no mercado.

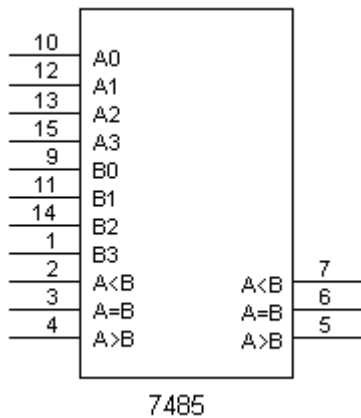


Um multiplexer de 8 entradas, o 74151, apresenta os três bits de seleção A, B e C, correspondendo a S0, S1, e S2, respetivamente. A entrada G é uma entrada "gate", "active-low", que quando ativa (a zero) Apresenta finalmente, a saída Y e a negação dessa saída W.

Um decodificador BCD-7 Segmentos 7447 que é apresentado com as saídas (A..G) "active-low", ou seja ativam-se quando têm zero. Este circuito interliga-se a displays de ânodo comum.

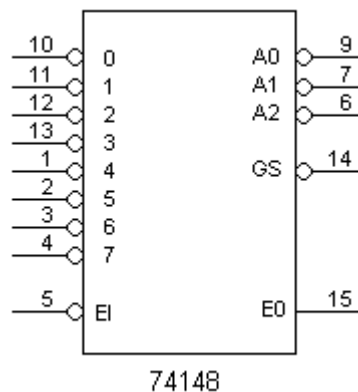


Um decoder de dois bits de entrada, 74139, apresenta-se com as suas entradas A e B que correspondem a A0 e A1 respetivamente. As suas saídas também aparecem "active-low" e possui uma entrada "gate", "G", também "active-low" que coloca o decoder em funcionamento quando está a zero (ativa).



Temos um comparador de dois números de 4 bit's cada, 7485, que apresenta as entradas do primeiro número (A0..A3) e do segundo (B0..B3). Tem as entradas normais e as de concatenação (A<B, A=B, A>B) e as saídas (A<B, A=B, A>B).

Temos, por último um Priority Encoder com 8 entradas, 74148, que coloca nas saídas A0, A1 e A2 a codificação da entrada ativa. Note que tanto as entradas como as saídas são "active-low", pelo que tudo ficará negado em relação aos valores normais. Tem uma entrada EI, que inibe ou não o funcionamento do circuito.



Resumo de Lógica Combinacional

Então, para resumir, os **Circuitos Lógicos Combinatórios ou combinacionais** consistem em entradas, duas ou mais portas lógicas e saídas básicas. As portas lógicas são combinadas de tal forma que o estado de saída depende inteiramente dos estados de entrada. Circuitos lógicos combinacionais têm "sem memória", "timing" ou "loops de feedback", a operação é instantânea. Um circuito lógico Combinacional executa uma operação atribuída logicamente por uma expressão booleana ou tabela verdade.

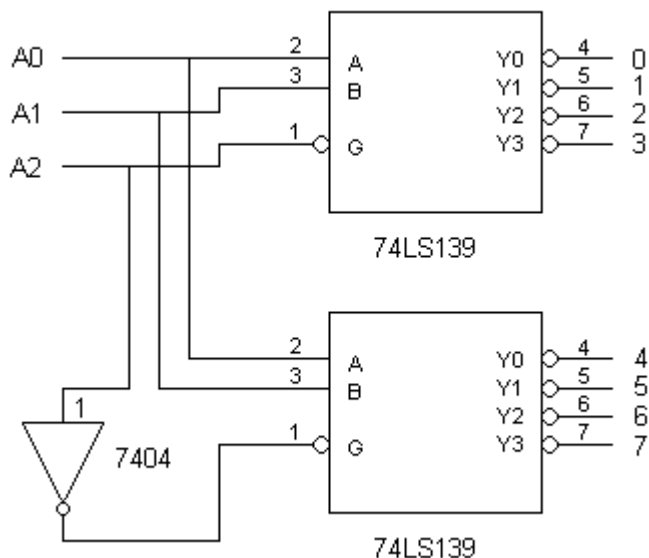
Exemplos de circuitos lógicos combinacionais comuns incluem: meio somadores, somadores completos, multiplexadores, demultiplexadores, codificadores e decodificadores,



Exercícios Resolvidos

- Desenhe, a partir de circuitos decoders de 2 bits um circuito decoder de 3 bits.

Resolução:

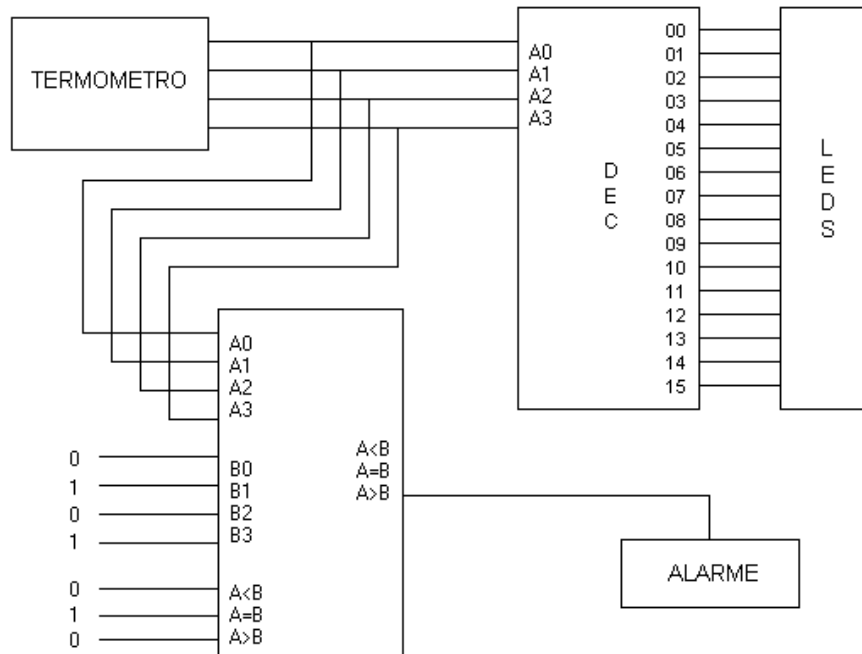


Os bits A0 e A1 do conjunto atacam os dois decoders da forma normal que se estudou. O bit A2 do conjunto, faz a comutação entre os dois decoders, de forma que, quando a 0, coloque em funcionamento o primeiro decoder e iniba o funcionamento do segundo, pois, quando A2 a 0, as palavras a descodificar estão no intervalo 0..3, logo o primeiro decoder tem de as descodificar. Quando A2 está a 1, o primeiro decoder fica inibido e o segundo fica a funcionar normalmente, e assim as palavras a descodificar estão no intervalo 4..7.



2. Um termómetro digital fornece a 4 bits temperaturas entre 0 e 15 graus. Projete um circuito que active um led para cada valor de temperatura e active um alarme sempre que a temperatura seja maior que 10 graus.

Resolução:

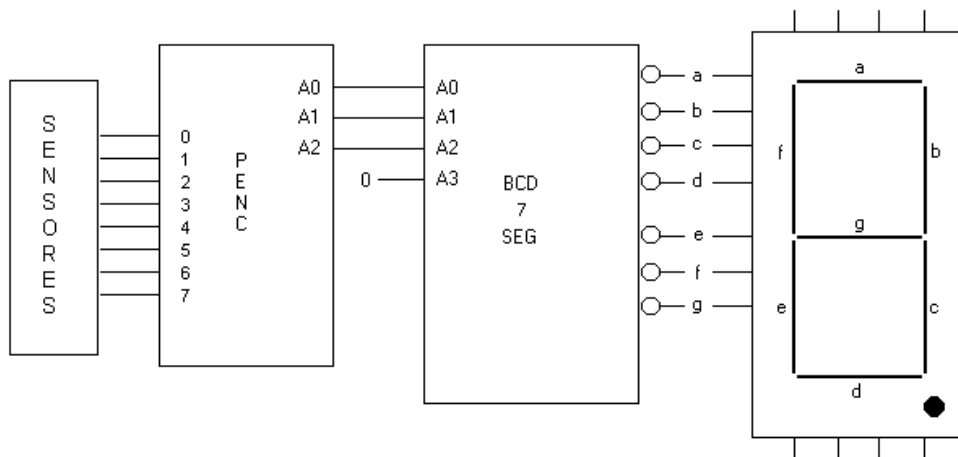


Se o termómetro dá a temperatura em 4 bits, podemos aplicar um decoder de 4 bits que ativa a saída correspondente ao valor da entrada em binário e aplicar uma barra de leds, cada um ligado a uma saída do decoder, ficando assim com uma barra que indica quantos graus leu o termómetro. Depois, aplicamos B do comparador, ligamos o valor 10 (1010) e quando a saída A>B se ativar, é porque a temperatura do termómetro ultrapassou os 10 graus, e assim ativa-se o alarme. De notar, por último, que as entradas do comparador devem ser ligadas como no esquema, para que o comparador faça a comparação.



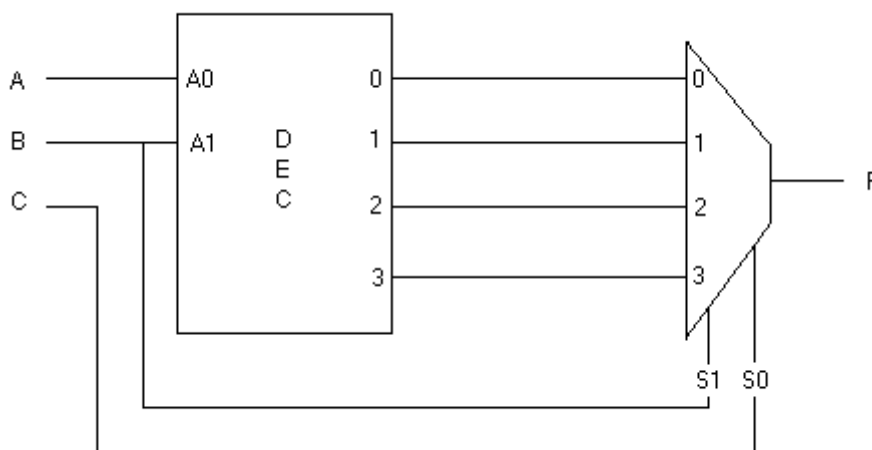
3. Projete um sistema de controlo para um parque de estacionamento de 8 lugares.

Este sistema deve colocar à entrada do parque um display, informando o número do lugar vago mais próximo da entrada.



Este circuito recebe a informação dos sensores que estão colocados nos respetivos lugares de estacionamento no parque. Assim se um lugar estiver vago, ativa-se o respetivo sensor. Basta usar um PENC com 8 entradas, codificá-las e aplicar a palavra em binário a um decodificador BCD-7 segmentos, que atacará um display. Note que na entrada do decodificador BCD-7 segmentos, o bit A3 está colocado a 0, pois se só temos uma palavra de três bits, o quarto bit deve ficar a 0 para não haver problemas.

a. Extraia a expressão da saída F





Resolução:

Para resolver este tipo de problemas, devemos usar uma tabela de verdade:

A	B	C	A0	A1	S0	S1	"0"	"1"	"2"	"3"	F
0	0	0	0	0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1	0	0	0	0
0	1	0	0	1	1	0	0	0	1	0	0
0	1	1	0	1	1	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1	0	0	0
1	0	1	1	0	0	1	0	1	0	0	0
1	1	0	1	1	1	0	0	0	0	1	0
1	1	1	1	1	1	1	0	0	0	1	1

Assim a função F é $\bar{A} \bar{B} \bar{C} + ABC$.

4. Implemente a seguinte função usando um multiplexer: $F = AB + AC$

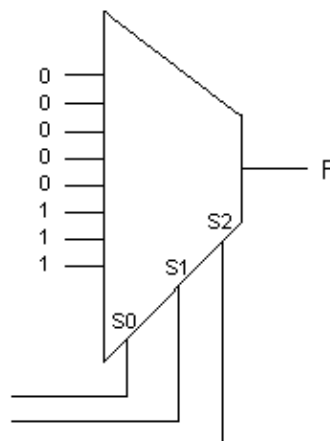
Resolução:

Este tipo de problemas resolve-se usando uma tabela de verdades da função:

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Agora, basta pensar que se a função depende de três variáveis, deve-se usar um multiplexer de 3 bits de seleção para termos as oito combinações possíveis.

Devemos colocar como S0 a variável C pois esta é a de menor peso. A resolução fica então na forma da figura seguinte:





Exercícios Propostos

1. Multiplexador 2 x 1

Após a construção da tabela verdade, é obtida a expressão simplificada para gerar uma equação ótima.

a. Considere o multiplexador $Z = \bar{S}.A + S.B$

i. Elabora a tabela de verdade e o circuito lógico.

Para ver solução alterar tipo de cor da tabela

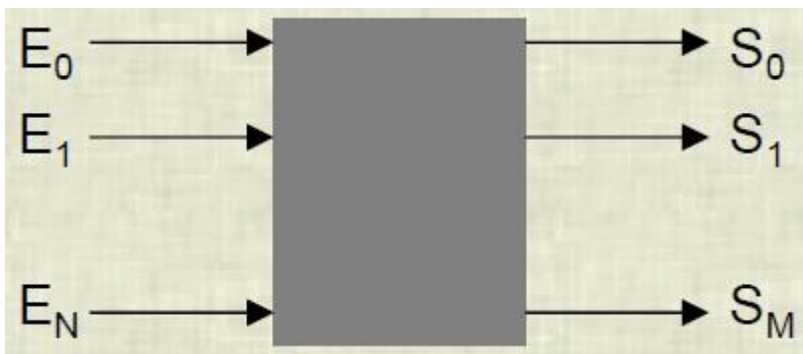
Codificador

Codificador é um circuito que mapeia um conjunto de entradas num conjunto de saídas segundo uma função de codificação

Por outras palavras, é um circuito que transforma uma informação de um formato para outro

Um codificador é normalmente implementado de forma combinacional—Implementação deste em VHDL pode ser realizada com o comando with-select

Representação gráfica de codificador genérico



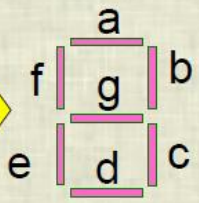


O exemplo abaixo ilustra um codificador BCD (binary-coded decimal) para sete segmentos

–Entrada em 4 bits (vetor Ent(3 downto 0))

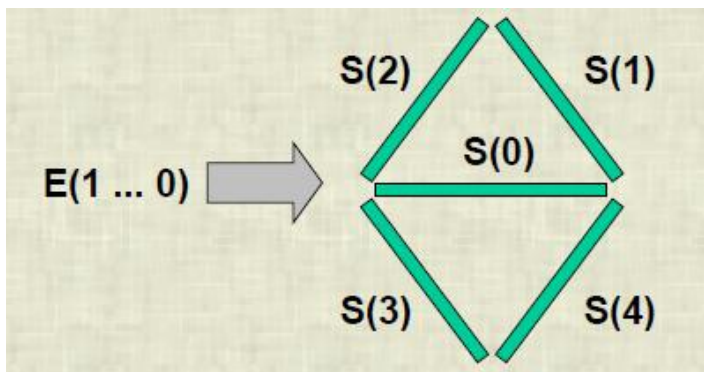
–Saída em 7 bits (vetor S(6 downto 0))

```
with Ent select
  S <= "0000001" when "0000",
       "1001111" when "0001",
       "0010010" when "0010",
       "0000110" when "0011",
       "1001100" when "0100",
       "0100100" when "0101",
       "0100000" when "0110",
       "0001111" when "0111",
       "0000000" when "1000",
       "0001100" when "1001",
       "1111110" when others;
```

BINÁRIO → 

EXERCÍCIOS:

- Descubra qual é a codificação do display de 7 segmentos, que relaciona as posições de S com os segmentos a,b,c,d,e,f,g
- Diga se os segmentos acendem com 0 ou 1
- Faça agora um codificador para mostrar todos os números hexadecimais
- Faça a codificação do display de elevador ilustrado abaixo



Este tem como entrada um vetor de 2 bits que recebe a seguinte codificação S

parado	00	-
subindo	01	Λ
descendo	10	V
estragado	11	todos segmentos acesos



Codificador com Prioridade

Codificador com prioridade é um circuito que determina entradas preferenciais. Conforme a ordem de preferência das entradas, no caso de haver mais de uma entrada ativa (por exemplo em 1), a entrada de maior prioridade é que determinará a codificação.

Exemplo:

A prioridade máxima é de S(2) e a mínima de S(0)

```
Y <= "11" when S(2) = '1' else
    "10" when S(1) = '1' else
    "01" when S(0) = '1' else
    "00";
```

Importante haver condição *default* em atribuições e estruturas similares:
NÃO HAVENDO ESTA CONDIÇÃO IMPLICA EM HAVER MEMORIZAÇÃO DO SINAL - diferente de software! (warning latch inferred)

Exercício:

Desenhe o diagrama de blocos deste circuito

Decodificador

Descodificador como um codificador, um circuito que mapeia entradas em saídas segundo uma função de codificação.

Genericamente, um circuito é chamado de decodificador, quando a função de transformação que ele realiza é inversa à de um circuito codificador.

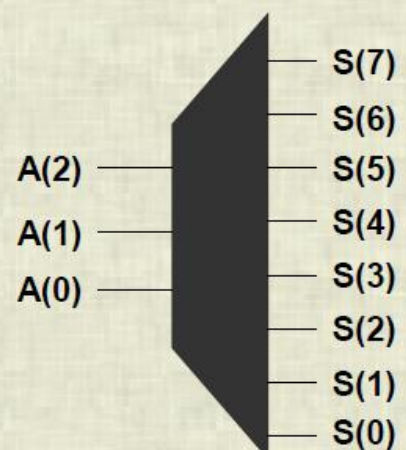
Um caso especial de decodificador é o binário. Este relaciona n entradas em 2^n saídas. Normalmente utilizado para endereçamento de memórias RAM.

– Exemplo:

– Decodificador de 3 entradas (A(2...0)) e 2^3 (8) saídas (S(7...0))

with A select

```
S <= "00000001" when "000",
    "00000010" when "001",
    "00000100" when "010",
    "00001000" when "011",
    "00010000" when "100",
    "00100000" when "101",
    "01000000" when "110",
    "10000000" when "111";
```

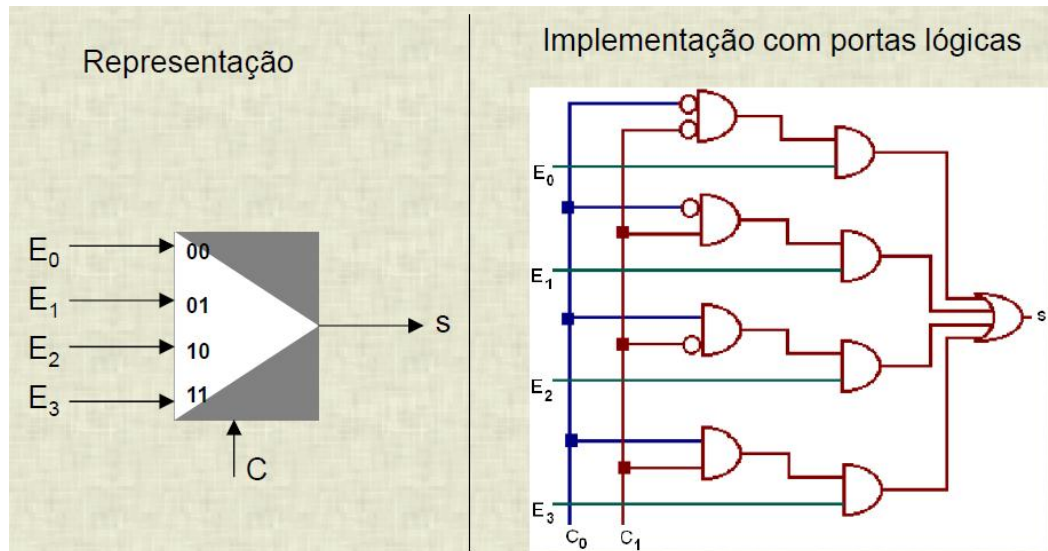




Multiplexador (Mux)

É um circuito que permite seleccionar mais de uma entrada em uma mesma saída, conforme um sinal de seleção

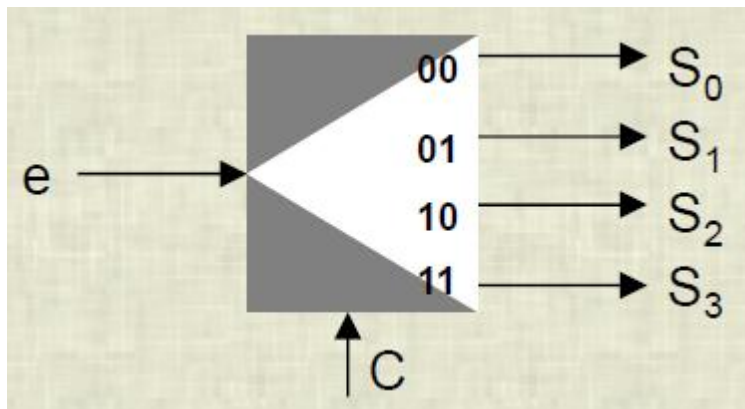
– Exemplo de um multiplexador 4x1



Demultiplexador (Demux)

É um circuito que opera de forma inversa ao multiplexador. Ou seja, recebe uma entrada e distribui esta em uma de várias saídas conforme um sinal de seleção

– Exemplo de um demultiplexador 1x4



Igualmente ao multiplexador, um demultiplexador pode ser implementado com diversos comandos. O mais comum é o que segue:

$S(0) \leftarrow e$ when $C = "00"$ else 'Z';

$S(1) \leftarrow e$ when $C = "01"$ else 'Z';

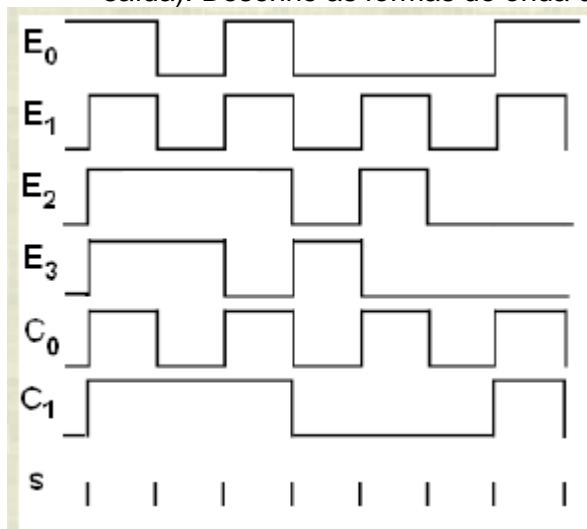
$S(2) \leftarrow e$ when $C = "10"$ else 'Z';

$S(3) \leftarrow e$ when $C = "11"$ else 'Z';

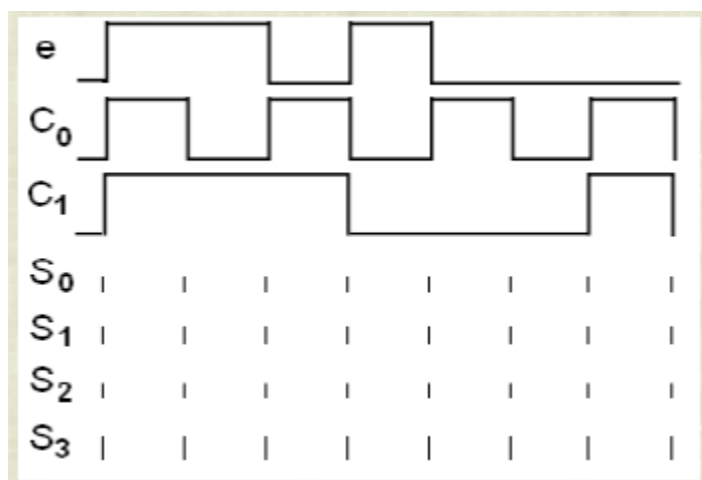


Exercícios

- Uma fábrica produz porcas e parafusos. O presidente da fábrica necessita de um sistema automático para separar as peças em lotes. Sempre que na esteira houver somente porcas, as mesmas devem ser enviadas para um caixa A. Se na esteira houver somente parafusos, os mesmos devem ser enviados para uma caixa B. Caso existam na esteira porcas e parafusos, os mesmos devem ser enviados para uma outra caixa C.
- Projete este sistema utilizando um circuito adequado (codificador, decodificador, multiplexador ou demultiplexador)
- A figura abaixo mostra as formas de ondas um mux 4x1 (E: entrada, C: controle, s: saída). Desenhe as formas de onda de s



- A figura abaixo mostra as formas de ondas um demux 1x4 (E: entrada, C: controle, S: saídas). Desenhe as formas de onda de S para o caso do demux ser implementado com tri-states e para o caso de ser implementado com portas lógicas



- Implemente um multiplexador 4x1, utilizando apenas multiplexadores 2x1
- Implemente um demultiplexador 1x4 utilizando um demultiplexador 1x8
- Implemente a expressão booleana a seguir usando um mux 8x1
 $S \leq (A \text{ and not } B) \text{ or } (A \text{ and not } (B \text{ and } C))$;
 Implemente a expressão booleana a seguir utilizando um mux 4x1 e uma porta NAND



$S \Leftarrow (A \text{ and not } B \text{ and not } C) \text{ or } (A \text{ and not } B \text{ and } C) \text{ or } (A \text{ and } B \text{ and not } C);$

ResoluçãoAula05

<https://www.youtube.com/watch?v=HoYOHmchgvc>

https://www.youtube.com/watch?v=Vly_pWedQGo

Projetos práticos: <https://www.youtube.com/watch?v=febM8Nc45i4>

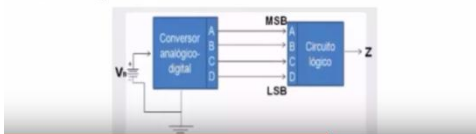
Projete um circuito lógico com 3 entradas, cuja saída será nível alto apenas quando a maioria das entradas seja alto.

Procedimento completo de implementação do projeto:

1. Interprete o problema e construa a tabela de verdade para descrever o seu funcionamento;
2. Escreva o termo AND (produto) para cada linha em que a saída seja 1;
3. Escreva a expressão da soma de produtos;
4. Simplifique a expressão se possível;
5. Implemente o circuito para a expressão final simplificada

EXEMPLO 1:

Análise a figura abaixo, em que um conversor analógico-digital está monitorando a tensão DC de uma bateria de 12V de uma espaçonave em órbita. A saída do conversor é um número binário de quatro bits, ABCD, que corresponde à tensão da bateria em degraus de 1V, sendo a variável A o MSB. As saídas binárias do conversor são entradas de um circuito que gerará uma saída em nível ALTO sempre que o valor binário for maior que $0110_2 = 6_{10}$, ou seja, quando a tensão da bateria for maior que 6V. Projete esse circuito lógico:



2. Explique o que é um circuito multiplexer e implemente à custa de um multiplexer a função: $F = A \cdot B + A \cdot C + B \cdot C$.
3. Explique o funcionamento de um demultiplexer e implemente através de demultiplexers de 4 saídas um demultiplexer de 8 saídas.



4. Qual a utilidade de um decodificador! BCD-7 segmentos?
5. Construa, usando portas lógicas um circuito encoder de 2 entradas, que dê prioridade às entradas de maior peso.
6. Desenhe um módulo comparador de dois números de 16 bits cada.

Ver: <http://www.estgv.ipv.pt/paginaspessoais/ffrancisco/sd0506/sumarios.htm>

<http://www.daelt.ct.utfpr.edu.br/elisanm/Digital/>

<https://www.electronics-tutorials.ws/category/combination>

exercícios:

<http://iris.sel.eesc.usp.br/sel414m/>