

Circuitos MSI e LSI e suas aplicações

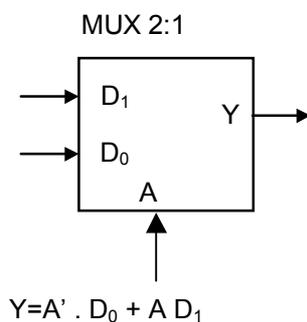
De acordo com a classificação dos CI's quanto ao nível de integração, directamente relacionado com o número de portas lógicas, são considerados circuitos MSI e LSI os "multiplexers"/seletores de dados, decodificadores/"demultiplexers", codificadores de prioridade, comparadores e circuitos aritméticos.

Multiplexer/Selector de dados

- ◆ Um multiplexer ou selector de dados é um circuito que permite seleccionar uma de entre várias entradas (entradas de dados) como fonte de informação para uma única saída.
- ◆ A selecção de uma das entradas de dados é controlada por um conjunto de entradas adicionais (entradas de selecção). Uma das 2^N entradas de dados é seleccionada à custa de N entradas de selecção (MUX $2^N:1$).
- ◆ Os multiplexers podem ter uma entrada adicional de "enable" (EN) ou "strobe" (G) que permite controlar (activar/desactivar) a função de multiplexagem.

Multiplexer de 2 para 1

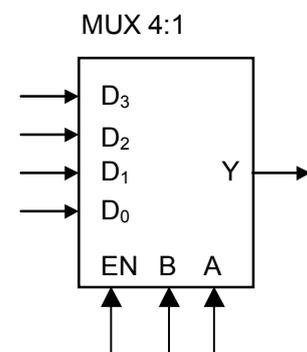
Um multiplexer de 2 para 1 (2 entradas de dados), designado por MUX 2:1, tem associada uma única entrada de selecção.



Entradas de dados: D_0, D_1
 Entrada de selecção: A (activa a um)
 Saída de dados: Y

A	Y
0	D_0
1	D_1

Multiplexer de 4 para 1 com entrada adicional de enable/strobe

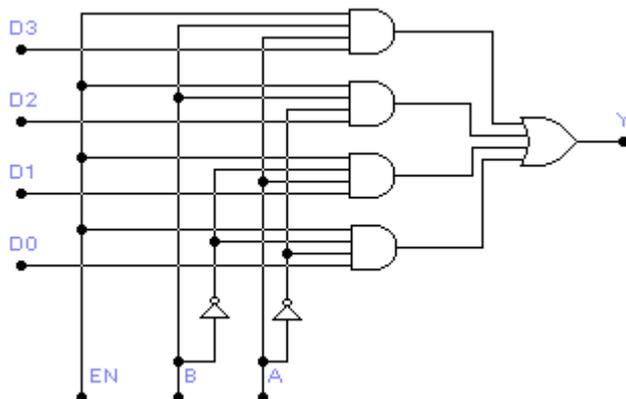


Entradas de dados: D_0, D_1, D_2, D_3
 Entradas de selecção: B, A (activas a um)
 Entrada de enable = EN (activa a um)
 Saída de dados: Y

EN	B	A	Y
0	X	X	0
1	0	0	D_0
1	0	1	D_1
1	1	0	D_2
1	1	1	D_3

$$Y = EN \cdot B' \cdot A' \cdot D_0 + EN \cdot B' \cdot A \cdot D_1 + EN \cdot B \cdot A' \cdot D_2 + EN \cdot B \cdot A \cdot D_3$$

Estrutura interna de um MUX 4:1 com entrada de enable activa a um.



Multiplexers disponíveis sob a forma de CI

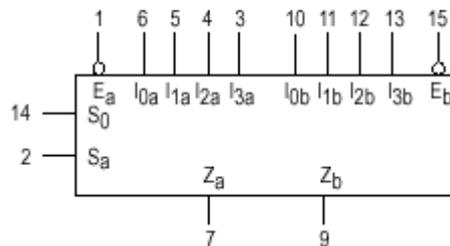
Exemplos:

◆ 54/74/XXX153 – Dual 4-Line to 1-Line Data Selectors/Multiplexers

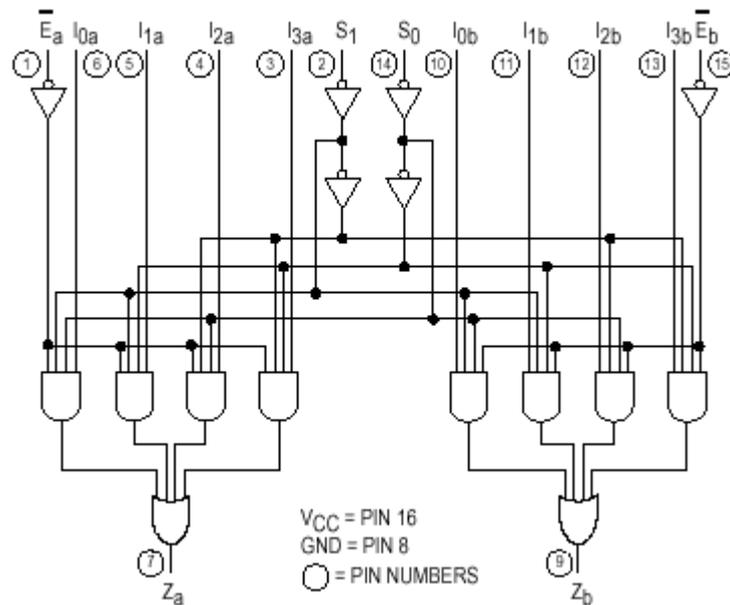
Principais funcionalidades:

- dois multiplexers de 4 para 1 (MUX 4:1);
- entradas de dados ($I_{3a} \dots I_{0a}$, $I_{3b} \dots I_{0b}$);
- entradas de selecção comuns (S_1 , S_0);
- entradas de enable independentes (E'_a , E'_b activas a zero);
- saídas não complementadas (Z_a , Z_b)

LOGIC SYMBOL



LOGIC DIAGRAM



TRUTH TABLE

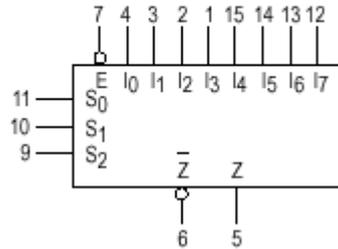
SELECT INPUTS		INPUTS (a or b)					OUTPUT
S_0	S_1	E	I_0	I_1	I_2	I_3	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

◆ 54/74/XXX151 – Dual 8-Line to 1-Line Data Selector/Multiplexer

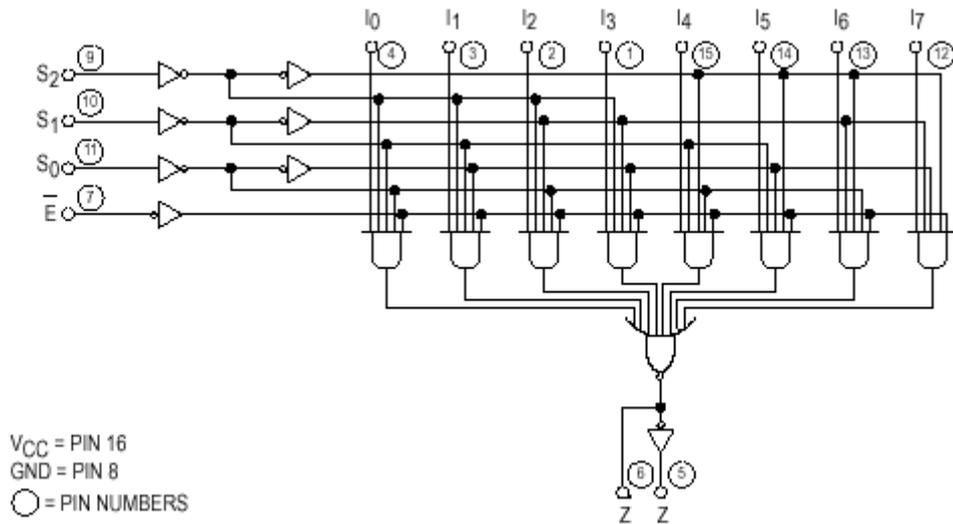
Principais funcionalidades:

- multiplexer de 8 para 1 (MUX 8:1);
- entradas de dados ($I_7..I_0$);
- entrada de enable (E' activa a zero);
- entradas de selecção ($S_2..S_0$)
- saída complementada (Z') e não complementada (Z).

LOGIC SYMBOL



LOGIC DIAGRAM



TRUTH TABLE

E	S ₂	S ₁	S ₀	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	Z'	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

Expansão de multiplexers

A expansão de multiplexers pode ser realizada através das entradas de enable. Na figura seguinte apresenta-se o diagrama lógico de um MUX 16:1 construído à custa de dois MUX 8:1.

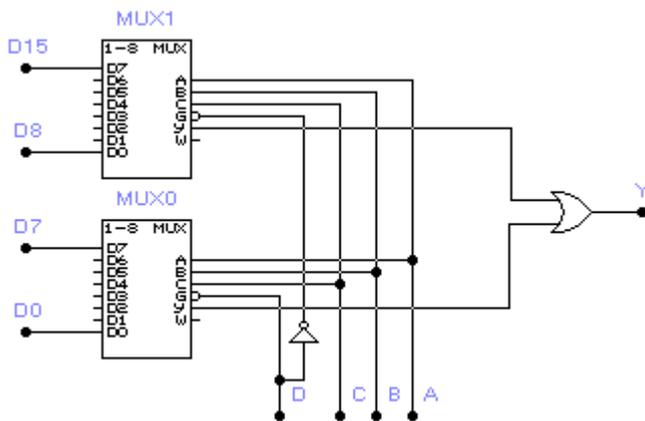
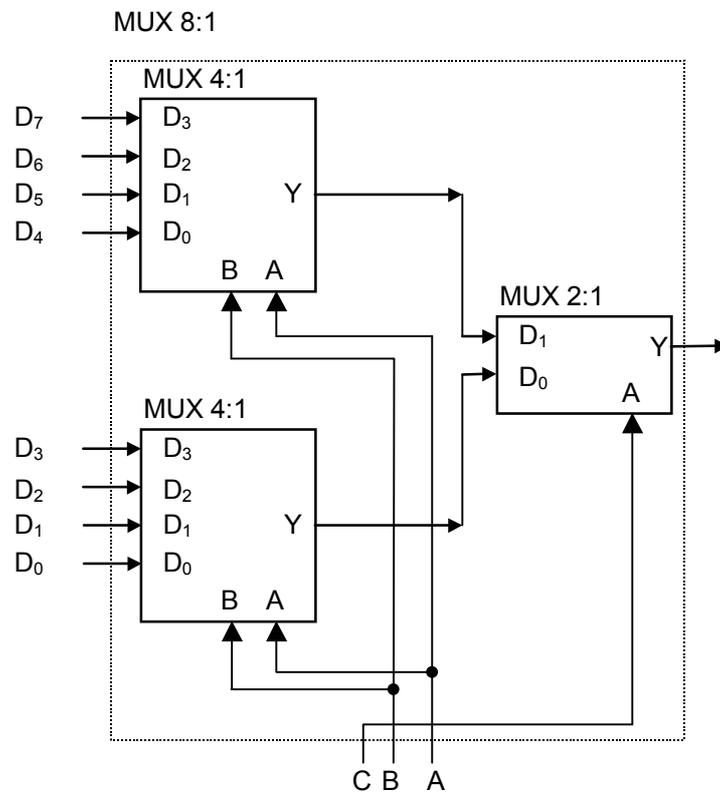


Tabela funcional

D	MUX0	MUX1	Saída Y do MUX0	Saída Y do MUX1	Y
0	Activo	Inactivo	Y= uma das entradas D7..D0 em função de C,B,A	0	Y= uma das entradas D7..D0 do MUX0 em função de C,B,A
1	Inactivo	Activo	0	Y= uma das entradas D7..D0 em função de C,B,A	Y= uma das entradas D7..D0 do MUX1 em função de C,B,A

No caso de não estarem disponíveis entradas de enable, pode colocar-se multiplexers em cascata, conforme se exemplifica.

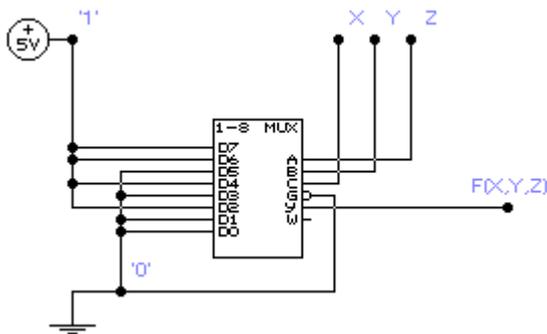


Implementação de funções combinacionais com multiplexers

Um multiplexer pode ser utilizado na implementação de uma função combinacional directamente a partir da tabela de verdade. Os mintermos de uma função são gerados por um multiplexer através das entradas de selecção, restando apenas ligar as entradas de dados a 0 ou a 1 em conformidade com a respectiva tabela de verdade. Uma função de N variáveis pode ser implementada com um MUX $2^N:1$ (N entradas de selecção e 2^N entradas de dados).

Exemplo:

Considerando $F(X,Y,Z)=\Sigma m(2,4,6,7)=\Pi M(0,1,3,5)$, fazendo corresponder as variáveis independentes X,Y,Z às entradas de selecção, as entradas de dados D_2, D_4, D_6 e D_7 devem ser ligadas a 1 e as entradas de dados D_0, D_1, D_3 e D_5 devem ser ligadas a 0.



X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

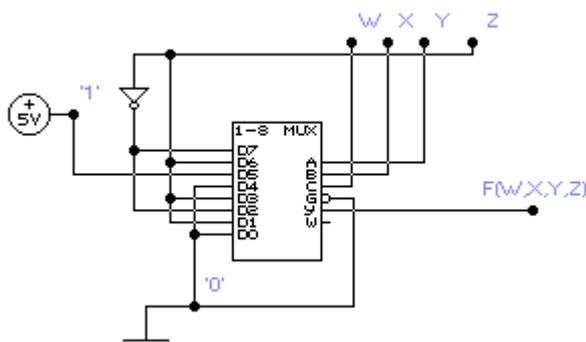
A implementação de funções de N variáveis com base em MUX $2^{N-1}:1$ ($N-1$ entradas de selecção) é possível desde que se considere a tabela reduzida e alguma lógica adicional.

Exemplo:

Considerando $F(W,X,Y,Z)=\Sigma m(3,4,7,10,11,13,14)$ pode se obter uma tabela de verdade reduzida, exprimindo a saída em função de uma das variáveis de entrada.

Nº	W	X	Y	Z	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Nº	W	X	Y	F
0	0	0	0	0
1	0	0	1	Z
2	0	1	0	Z'
3	0	1	1	Z
4	1	0	0	0
5	1	0	1	1
6	1	1	0	Z
7	1	1	1	Z'



Entradas de dados:

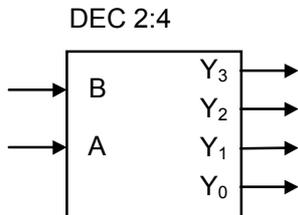
- D0 e D4 ligadas a 0
- D5 ligada a 1
- D1, D3 e D6 ligadas a Z
- D2 e D7 ligada a Z'

Descodificadores/Demultiplexers

- Um descodificador é um circuito cujas saídas são activadas selectivamente por combinações de valores lógicos aplicados nas entradas.
- Um descodificador com 2^N saídas tem N variáveis de entrada.
- Os descodificadores podem ter uma entrada adicional de “enable” (EN) ou “strobe” (G) que permite controlar (activar/desactivar) a função de descodificação.

Descodificador de 2 para 4

Um descodificador de 2 para 4, designado por DEC 2:4, tem 2 entradas e 4 saídas.



$$Y_0 = B'A'; Y_1 = B'A; Y_2 = BA'; Y_3 = BA$$

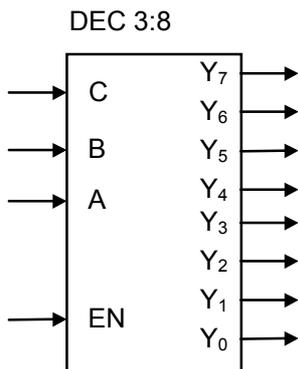
Entradas: B,A (activas a um)
Saídas: Y_3, Y_2, Y_1, Y_0 (activas a um)

Tabela funcional

B	A	Y_0	Y_1	Y_2	Y_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Descodificador de 3 para 8 com entrada de enable

Um descodificador de 3 para 8, designado por DEC 3:8, tem 3 entradas e 8 saídas. Se a entrada de enable estiver inactiva, qualquer que seja o estado das entradas, todas as saídas estão inactivas.



$$Y_0 = EN C'B'A'; Y_1 = EN C'B'A; Y_2 = EN C'BA';$$

$$Y_3 = EN C'BA; Y_4 = EN CB'A'; Y_5 = EN CB'A;$$

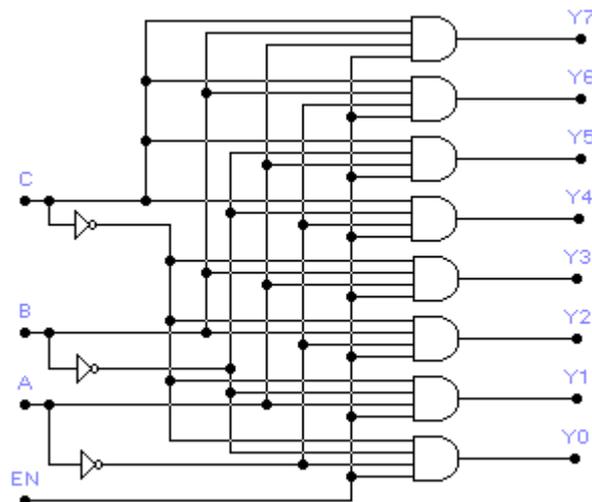
$$Y_6 = EN CBA'; Y_7 = EN CBA$$

Estrutura interna

Entradas: C, B, A (activas a um)
Entrada de enable : EN (activa a um)
Saídas: $Y_7, Y_6, Y_5, Y_4, Y_3, Y_2, Y_1, Y_0$ (activas a um)

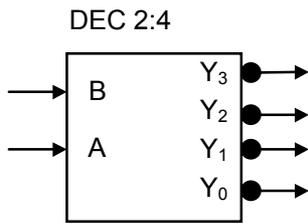
Tabela funcional

EN	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
1	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	1
0	X	X	X	0	0	0	0	0	0	0	0



Descodificador de 2:4 com saídas activas a zero

Neste tipo de descodificador as saídas são obtidas a partir de portas NAND.



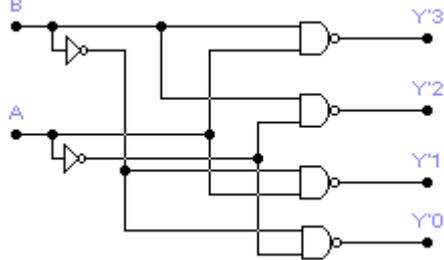
Entradas: B,A (activas a um)
Saídas: Y₃, Y₂, Y₁, Y₀ (activas a zero)

Tabela funcional

B	A	Y ₀	Y ₁	Y ₂	Y ₃
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

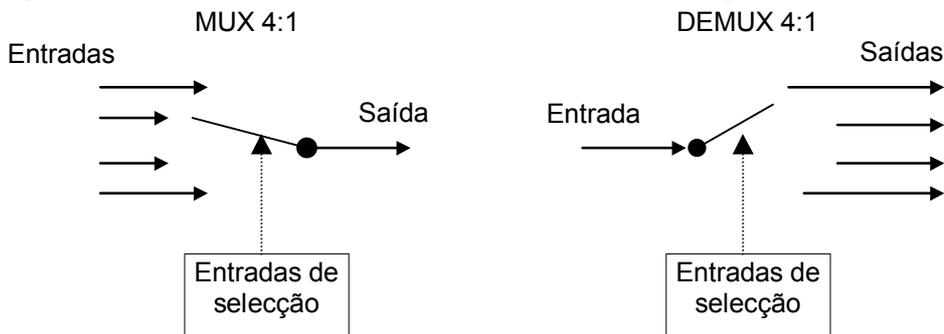
$Y'_0=(B'A)'$; $Y'_1=(B'A)'$; $Y'_2=(BA)'$; $Y'_3=(BA)'$

Estrutura interna



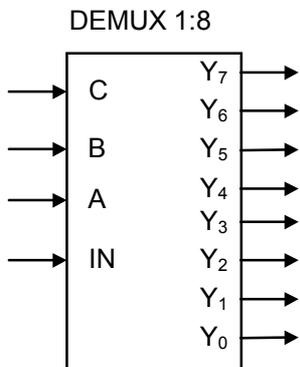
Demultiplexers

- Um demultiplexer é um circuito que permite encaminhar selectivamente a informação aplicada numa única entrada para uma de várias saídas.
- Um demultiplexer de 1 para 2^N tem uma entrada, 2^N saídas e N entradas de selecção.
- Realiza a função inversa de um multiplexer, conforme se evidencia na figura.



Demultiplexer de 1 para 8

Um demultiplexer de 1 para 8, designado por DEMUX 1:8, permite o encaminhamento selectivo da informação aplicada na entrada para uma de entre as 8 saídas. A selecção da saída é feita por intermédio de 3 entradas de selecção.



Entrada = IN
Entradas de selecção: C, B, A (activas a um)
Saídas: Y₇, Y₆, Y₅, Y₄, Y₃, Y₂, Y₁, Y₀

Tabela funcional

C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	IN	0	0	0	0	0	0	0
0	0	1	0	IN	0	0	0	0	0	0
0	1	0	0	0	IN	0	0	0	0	0
0	1	1	0	0	0	IN	0	0	0	0
1	0	0	0	0	0	0	IN	0	0	0
1	0	1	0	0	0	0	0	IN	0	0
1	1	0	0	0	0	0	0	0	IN	0
1	1	1	0	0	0	0	0	0	0	IN

Se compararmos a tabela funcional de um DEMUX 1:8 com a tabela funcional de um DEC 3:8 com entrada de enable, verifica-se que estas são idênticas se se fizer coincidir a entrada de enable (EN) do DEC 3:8 com a entrada (IN) do DEMUX 1:8. Esta é a solução adoptada nos dispositivos disponíveis no mercado, justamente designados por descodificadores/demultiplexers.

Descodificadores/Demultiplexers disponíveis sob a forma de CI

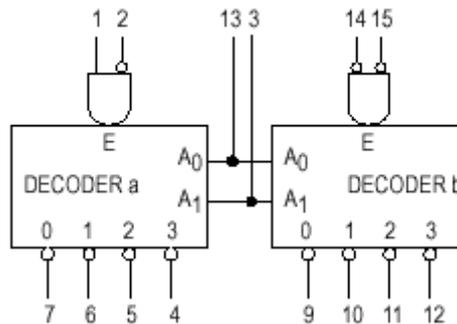
Exemplos:

◆ 54/74/XXX155 – Dual 1 of 4 Decoder/Demultiplexer

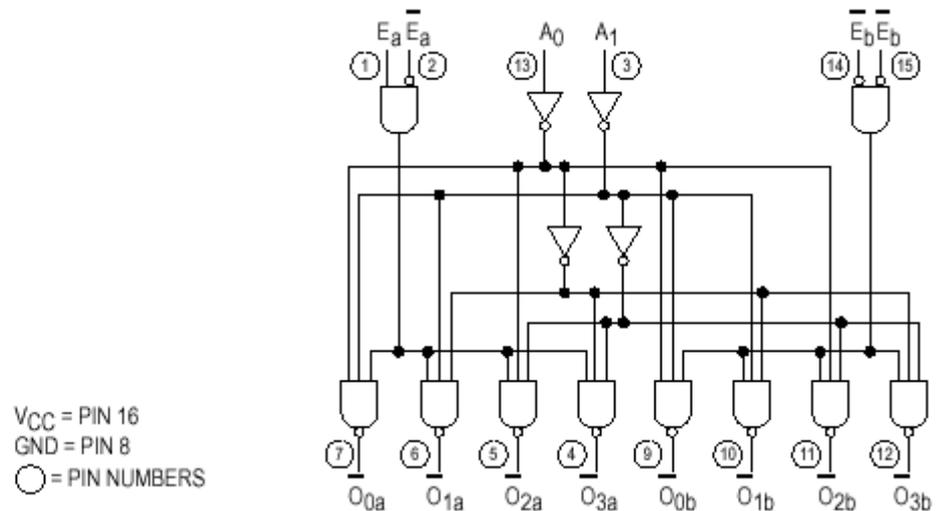
Principais funcionalidades:

- dois descodificadores de 2 para 4 (DEC 2:4)/ demultiplexers 1 para 4 (DEMUX 1:4);
- entradas de selecção comuns (A_1 , A_0 activas a um);
- saídas activas a zero ($O'_{3a}..O'_{0a}$, $O'_{3b}..O'_{0b}$);
- entradas de enable independentes: o descodificador –a- com duas entradas de enable, uma activa a um (E_a) e a outra activa a zero (E'_a); o descodificador –b- com duas entradas de enable ambas activas a zero (E'_b).

LOGIC SYMBOL



LOGIC DIAGRAM



TRUTH TABLE

ADDRESS		ENABLE "a"		OUTPUT "a"				ENABLE "b"		OUTPUT "b"			
A_0	A_1	E_a	E'_a	O_0	O_1	O_2	O_3	E'_b	E'_b	O_0	O_1	O_2	O_3
X	X	L	X	H	H	H	H	H	X	H	H	H	H
X	X	X	H	H	H	H	H	X	H	H	H	H	H
L	L	H	L	L	H	H	H	L	L	L	H	H	H
H	L	H	L	H	L	H	H	L	L	H	L	H	H
L	H	H	L	H	H	L	H	L	L	H	H	L	H
H	H	H	L	H	H	H	L	L	L	H	H	H	L

- ◆ 54/74/XXX156 – Dual 1 of 4 Decoder/Demultiplexer – Idêntico ao 155 mas apresenta saídas em colectador aberto.

Expansão de descodificadores

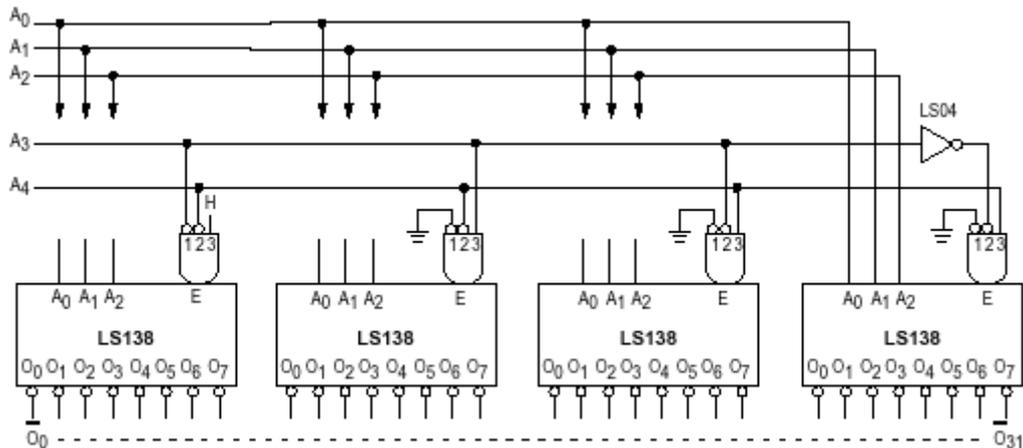
A expansão de descodificadores é normalmente realizada através das entradas de enable.

Exemplo:

Descodificador de 5 para 32 com base em 4 descodificadores 74LS138.

Entradas: $A_4..A_0$ (activas a um)

Saídas: $O'_{31}..O'_0$ (activas a zero)



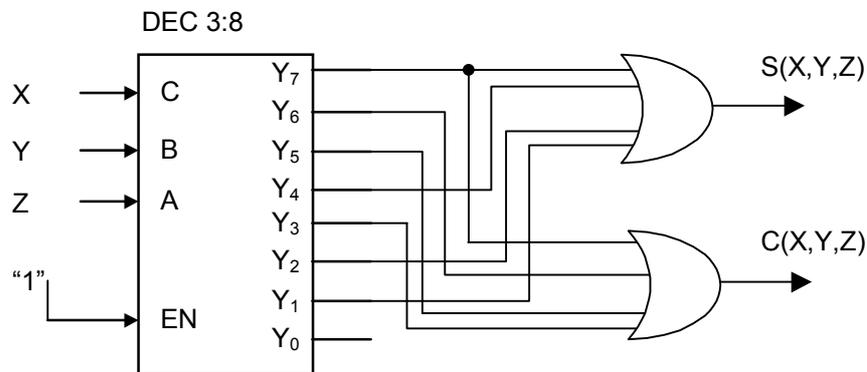
Implementação de funções combinacionais com descodificadores

Um descodificador disponibiliza nas suas saídas os 2^N mintermos de uma função de N variáveis. Cada saída do descodificador corresponde a um mintermo. Uma função combinacional pode ser implementada com base numa porta OR com as entradas ligadas às saídas do descodificador que correspondem aos mintermos da função.

Exemplo:

$$S(X,Y,Z) = \sum m(1,2,4,7)$$

$$C(X,Y,Z) = \sum m(3,5,6,7)$$



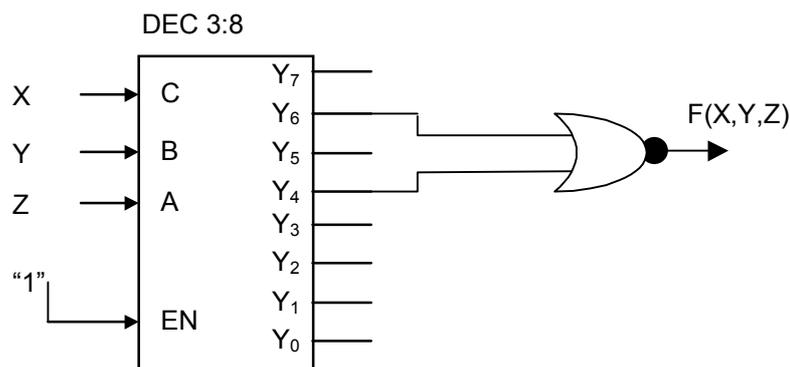
Se o número de mintermos da função a implementar é superior a $2^{N/2}$, então a função complementar é expressa por um menor número de mintermos, sendo vantajosa a utilização de uma porta NOR.

Exemplo:

$$F(X,Y,Z) = \sum m(0,1,2,3,5,7) = \prod M(4,6)$$

$$F'(X,Y,Z) = \sum m(4,6)$$

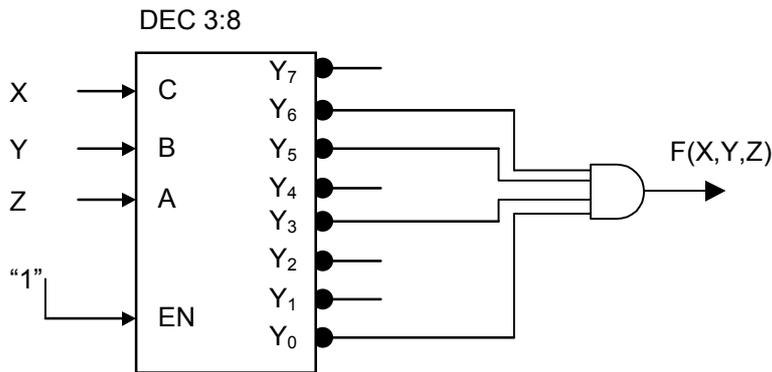
$$F''(X,Y,Z) = (\sum m(4,6))'$$



Para decodificadores construídos com portas NAND, ou seja, que apresentam as saídas activas a zero, deve ser utilizada uma porta AND com entradas ligadas às saídas do decodificador que correspondem aos maxtermos da função.

Exemplo:

$$F(X,Y,Z) = \sum m(1,2,4,7) = \prod M(0,3,5,6)$$



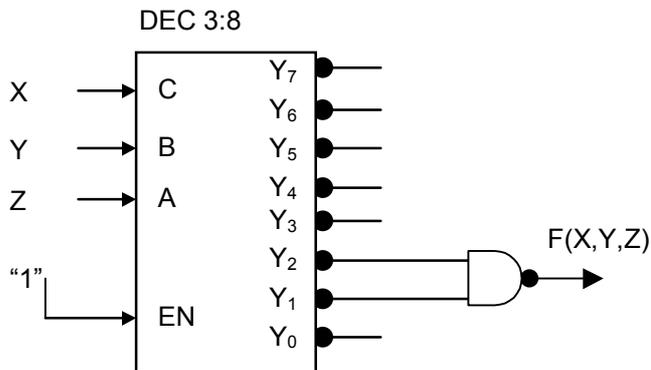
Se o número de maxtermos da função for superior a $2^N/2$, então a função complementar é expressa por um menor número de maxtermos, sendo vantajosa a utilização de uma porta NAND.

Exemplo:

$$F(X,Y,Z) = \prod M(0,3,4,5,6,7) = \sum m(1,2)$$

$$F'(X,Y,Z) = \prod M(1,2)$$

$$F''(X,Y,Z) = (\prod M(1,2))'$$

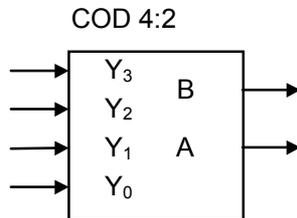


Codificadores

- ◆ Um codificador é um circuito que efectua a operação inversa de um descodificador, ou seja, representa nas suas saídas o código binário do número da entrada que está activa.
- ◆ Um codificador com 2^N entradas tem N variáveis de saída.

Codificador de 4 para 2

Um codificador hipotético de 4 para 2, designado por COD 4:2, tem 4 entradas e 2 saídas.



Saídas: B,A (activas a um)
Entradas: Y_3, Y_2, Y_1, Y_0 (activas a um)

Tabela funcional

Y_0	Y_1	Y_2	Y_3	B	A
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

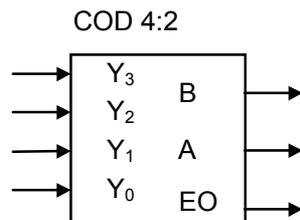
No entanto, se se observar a tabela funcional, verifica-se que não estão representadas todas as combinações possíveis para as entradas Y_3, Y_2, Y_1, Y_0 . Qual o valor das saídas B e A para, por exemplo, $Y_3=1, Y_2=0, Y_1=1, Y_0=0$? Nestas situações, deve ser considerado um sistema de prioridades. Quando mais do que uma entrada está activa, as saídas representam o código binário do número da entrada com maior prioridade. A este tipo de circuito dá-se o nome de codificador de prioridade.

Codificadores de prioridade

- ◆ Um codificador de prioridade é um circuito que representa nas saídas o código binário do número da entrada activa com maior prioridade.
- ◆ Um codificador de prioridade com 2^N entradas tem N variáveis de saída.
- ◆ Os codificadores de prioridade podem ter uma entrada adicional de “enable” (EN) ou “strobe” (G) que permite controlar (activar/desactivar) a função de codificação.
- ◆ Os codificadores de prioridade têm, normalmente, uma saída adicional que permite detectar a situação em que nenhuma das entradas está activa.

Codificador de prioridade de 4 para 2

O codificador de prioridade de 4 para 2 da figura, designado por COD 4:2, tem 4 entradas (Y_3, Y_2, Y_1, Y_0), sendo a mais prioritária a entrada Y_3 e a menos prioritária a entrada Y_0 . Para além das saídas B e A, existe uma saída adicional EO que indica se pelo menos uma das entradas está activa.



Saídas: B,A (activas a um); EO (activa a um)
Entradas: Y_0, Y_1, Y_2, Y_3 (activas a um)

Tabela funcional

Y_0	Y_1	Y_2	Y_3	B	A	EO
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

Estrutura interna

Recorrendo aos mapas de Karnaugh para as funções de saída B, A e EO, obtém-se:

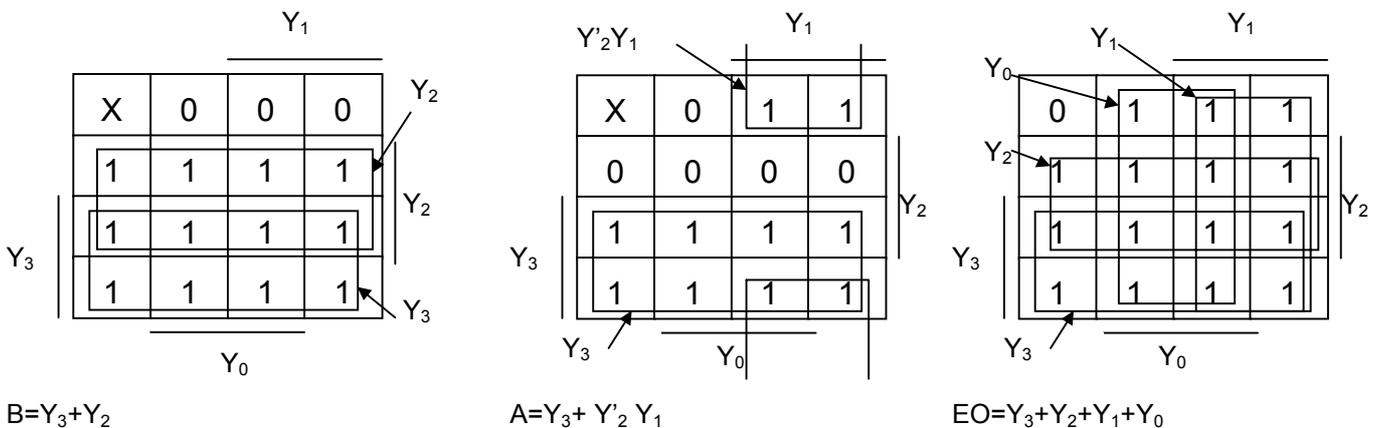


Diagrama lógico

