

Sistemas Digitais

Aula 10

Multiplexers

Tecnologia 3-state

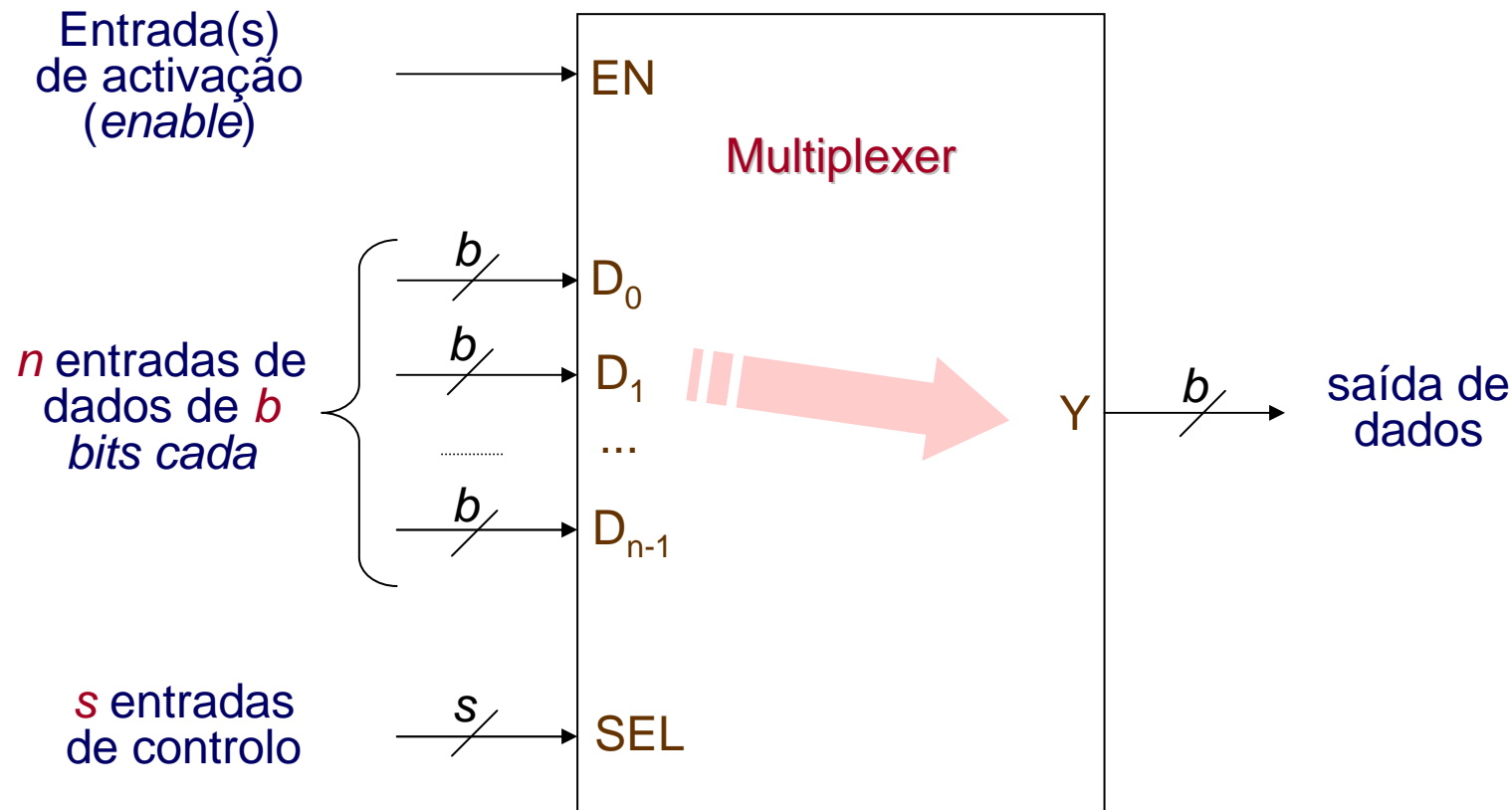
Funções com multiplexers



Multiplexers

Um **multiplexer** encaminha o valor lógico de uma das n entradas de dados (com n potência de 2) para uma única saída.

A fonte é seleccionada com base em $s = \lceil \log_2(n) \rceil$ entradas de controlo.



Multiplexers (cont.)

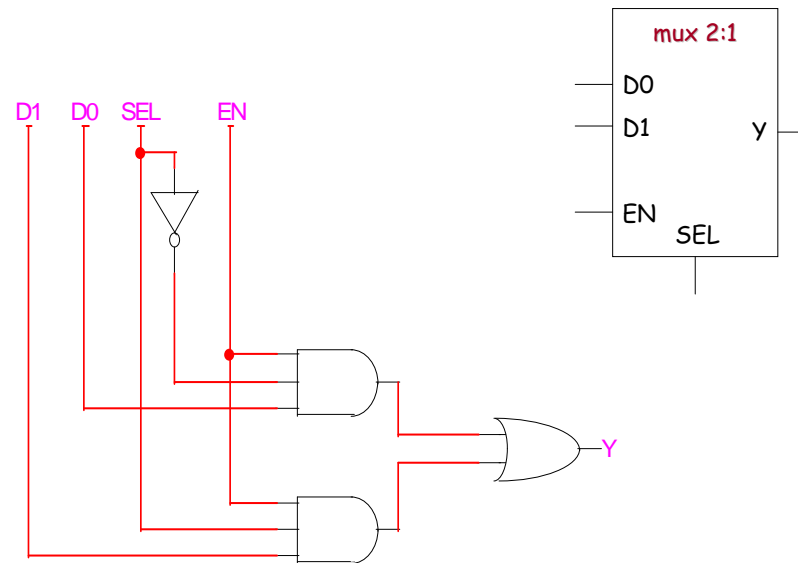
Equação de saída de um multiplexer n:1
(m_j – termo mínimo j das entradas de controlo)

$$Y = \sum_{j=0}^{n-1} EN \cdot m_j \cdot D_j$$

Exemplo: Multiplexer 2:1:

EN	SEL	D1	D ₀	Y
0	x	x	x	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

EN	SEL	Y
0	x	0
1	0	D ₀
1	1	D ₁



$$Y = EN \cdot \overline{SEL} \cdot D0 + EN \cdot SEL \cdot D1$$



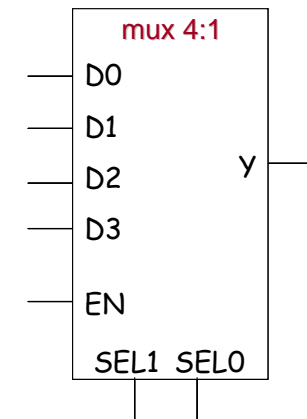
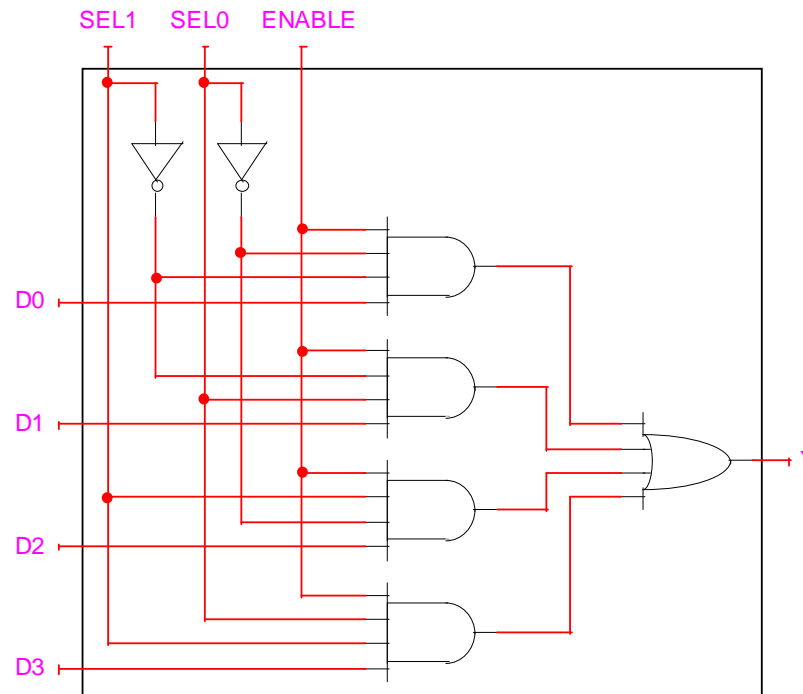
Multiplexers (cont.)

Equação de saída de um multiplexer n:1:
(m_j – termo mínimo j das entradas de controlo)

$$Y = \sum_{j=0}^{n-1} EN \cdot m_j \cdot D_j$$

Exemplo: Multiplexer 4:1:

EN	SEL1	SEL0	Y
0	x	x	0
1	0	0	D0
1	0	1	D1
1	1	0	D2
1	1	1	D3

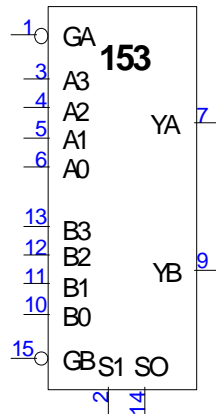


$$Y = EN \cdot \overline{SEL1} \cdot \overline{SEL0} \cdot D0 + EN \cdot \overline{SEL1} \cdot SEL0 \cdot D1 + EN \cdot SEL1 \cdot \overline{SEL0} \cdot D2 + EN \cdot SEL1 \cdot SEL0 \cdot D3$$



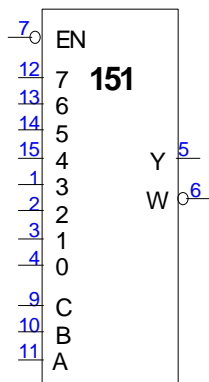
Multiplexers comerciais

74x153 – dual multiplexer 4:1



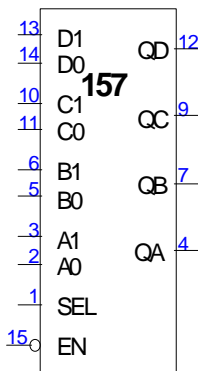
GA_L	GB_L	S1	S0	YA	YB
0	0	0	0	A0	B0
0	0	0	1	A1	B1
0	0	1	0	A2	B2
0	0	1	1	A3	B3
0	1	0	0	A0	0
0	1	0	1	A1	0
0	1	1	0	A2	0
0	1	1	1	A3	0
1	0	0	0	0	B0
1	0	0	1	0	B1
1	0	1	0	0	B2
1	0	1	1	0	B3
1	1	x	x	0	0

74x151 - multiplexer 8:1 de 1 bit



EN_L	C	B	A	Y	W_L
1	x	x	x	0	1
0	0	0	0	D0	$\overline{D0}$
0	0	0	1	D1	$\overline{D1}$
0	0	1	0	D2	$\overline{D2}$
0	0	1	1	D3	$\overline{D3}$
0	1	0	0	D4	$\overline{D4}$
0	1	0	1	D5	$\overline{D5}$
0	1	1	0	D6	$\overline{D6}$
0	1	1	1	D7	$\overline{D7}$

74x157 - multiplexer 2:1 de 4 bits



EN_L	SEL	QD	QC	QB	QA
1	x	0	0	0	0
0	0	D0	C0	B0	A0
0	1	D1	C1	B1	A1

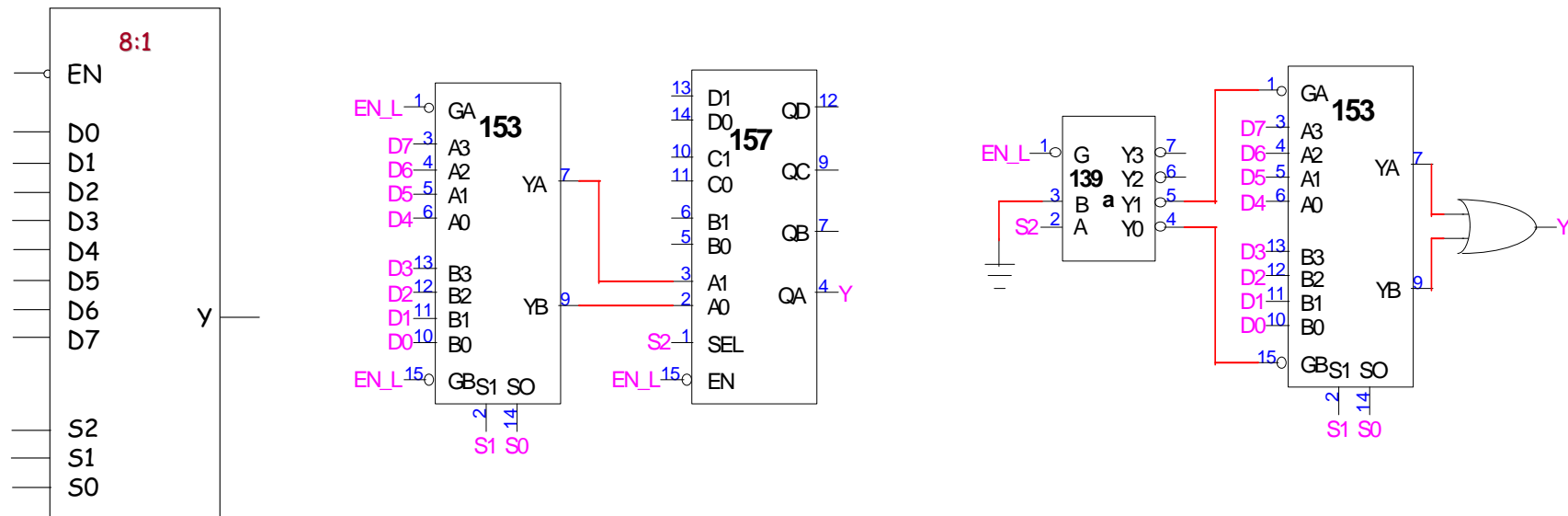


Multiplexers em cascata

Multiplexers maiores podem ser implementados colocando multiplexers mais pequenos em cascata.

Exemplo:

Construir um multiplexer 8:1:

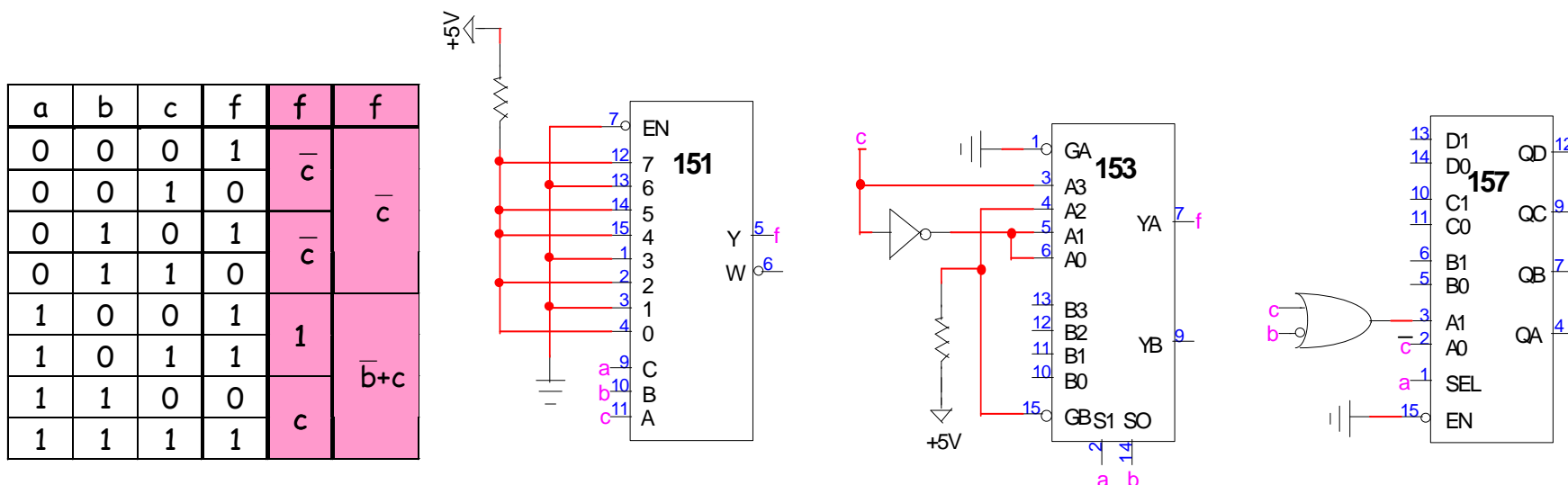


Multiplexers e funções lógicas

Com um multiplexer $2^n:1$ e constantes 0 e 1 pode-se implementar qualquer função lógica de n variáveis.

Com um multiplexer $2^{n-1}:1$, constantes 0 e 1 e portas NOT pode-se implementar qualquer função lógica de n variáveis.

Exemplos: $f(a,b,c) = a \cdot \bar{b} + a \cdot c + \bar{a} \cdot \bar{c}$



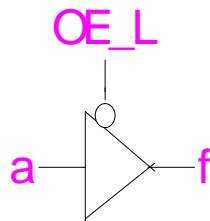
Buffers 3-state

Para além das representações eléctricas **LOW** e **HIGH** dos estados lógicos, existe um terceiro “estado” eléctrico - **alta impedância (Z)** que representa uma resistência infinita.

Uma saída com 3 estados possíveis chama-se **saída three-state** (ou **tri-state**).

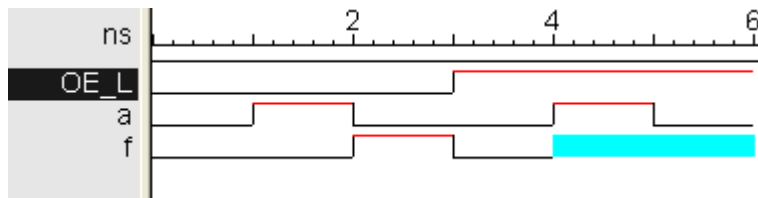
Dispositivos **3-state** têm uma entrada adicional de activação (**enable**).

Buffer 3-state



OE_L	a	f
0	0	0
0	1	1
1	x	Z

OE_L	f
0	a
1	Z



O estado Z permite ligar as saídas de mais do que um dispositivo **3-state** ao mesmo ponto, desde que **apenas um** tenha a sua saída activa em cada instante.

Para tal os dispositivos **3-state** são projectados por forma a entrarem no estado Z mais rapidamente do que a saírem do mesmo:

$$t_{pLZ} < t_{pZL}$$

$$t_{pLZ} < t_{pZH}$$

$$t_{pHZ} < t_{pZH}$$

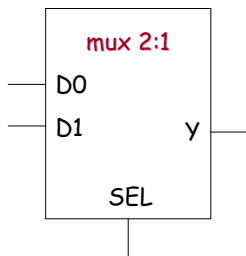
$$t_{pHZ} < t_{pZL}$$



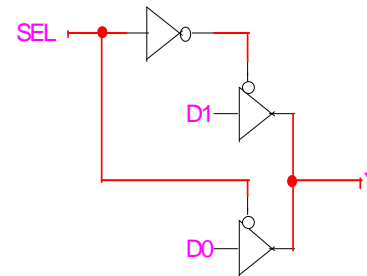
Multiplexagem com buffers 3-state

Com portas 3-state pode-se construir multiplexers “baratos”.

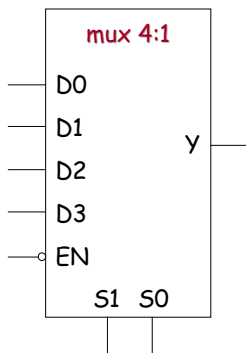
Exemplos: Multiplexer 2:1:



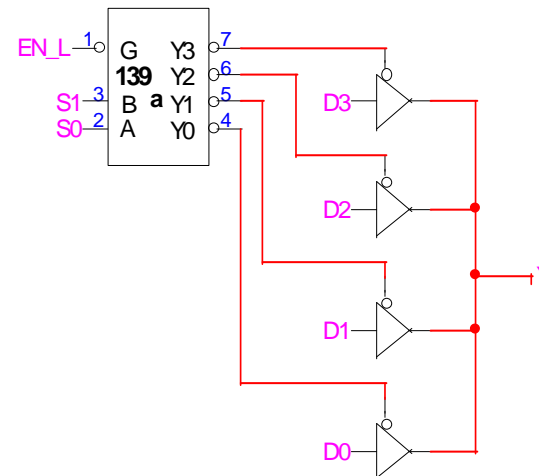
SEL	Y
0	D0
1	D1



Multiplexer 4:1:



EN_L	S1	S0	Y
1	x	x	Z
0	0	0	D0
0	0	1	D1
0	1	0	D2
0	1	1	D3



Exercícios

Implemente a função $f(a,b,c) = a + \bar{b} + c$ com

a) um decodificador 3-to-8;

b) um multiplexer 4:1.

Construa um multiplexer 8:1 com um decodificador 3-to-8 e *buffers* 3-state.



Exercícios

Considere a função $f(x_3, x_2, x_1, x_0)$ dada pelo seguinte mapa de Karnaugh:

- a) Determine a função mínima na forma de uma soma de produtos a dois níveis
- b) Implemente a mesma função recorrendo a um multiplexer de 16:1 e as constantes '0' e '1'
- c) Implemente a mesma função recorrendo a um multiplexer de 8:1 e 1 operador not e constantes '0' e '1'
- d) Implemente a mesma função recorrendo a um multiplexer de 4:1, às constantes '0' e '1' e ao menor número possível de operadores lógicos booleanos

		X3X2			
		00	01	11	10
X1X0	00			1	1
	01	1		1	
	11	1		1	
	10			1	1

