

# Lógica e Sistemas Digitais

Circuitos combinatórios

João Pedro Patriarca ([jpatri@cc.isel.ipl.pt](mailto:jpatri@cc.isel.ipl.pt))

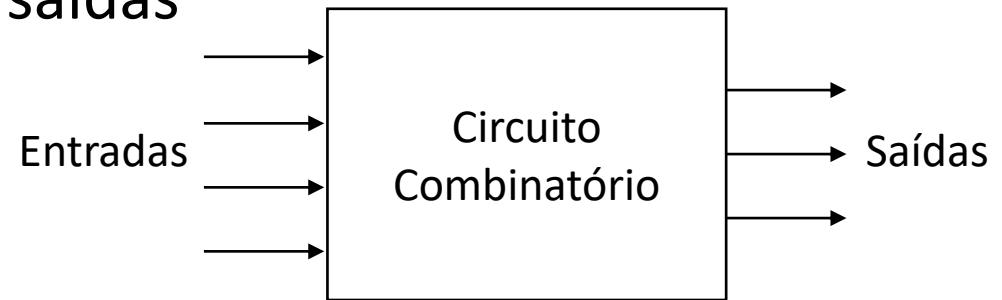
Slides inspirados nos slides do prof. Mário Véstias



# Circuito combinatório

---

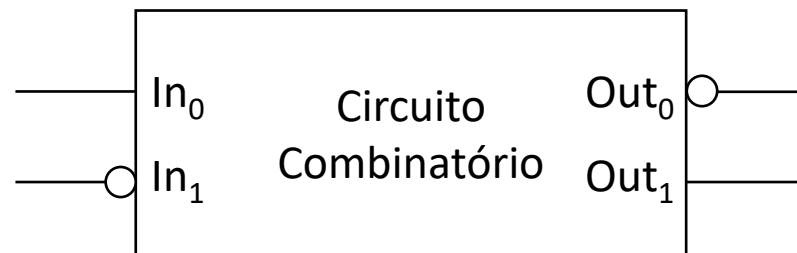
- Por definição, um circuito combinatório é caracterizado pelas saídas dependerem, num dado instante, exclusivamente do valor lógico presente nas entradas
- Ao contrário, um circuito sequencial (tema estudado em futuras aulas) é caracterizado pelas saídas dependerem não só do valor lógico presente nas entradas mas também dependerem das condições a que o circuito esteve sujeito em momentos anteriores (noção de estado anterior ou memória)
- A interface de um circuito combinatório com o exterior é constituída por uma ou mais entradas e uma ou mais saídas
- Cada saída é representada por uma função booleana, função de variáveis de entrada



# Ativação de entradas/saídas de um circuito combinatório

---

- A ativação das entradas e saídas de um circuito combinatório podem ter diferentes níveis lógicos, ou seja, quando se diz que uma entrada ou uma saída está ativa não significa necessariamente que apresente o valor lógico 1
- Entradas e saídas ativas a 0 são representadas na interface do bloco do circuito com um círculo
- No exemplo, a entrada  $In_0$  e a saída  $Out_1$  são ativas com o valor lógico 1 (*active high*) e a entrada  $In_1$  e a saída  $Out_0$  são ativas com o valor lógico 0 (*active low*)



# Síntese de circuitos combinatórios simples

---

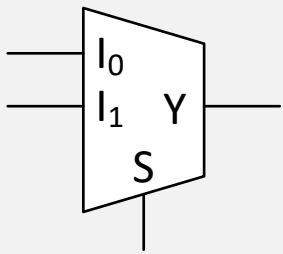
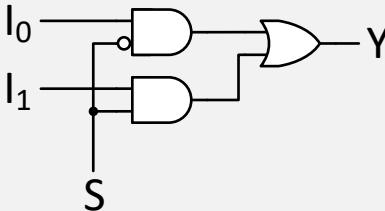
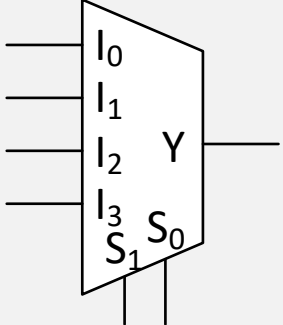
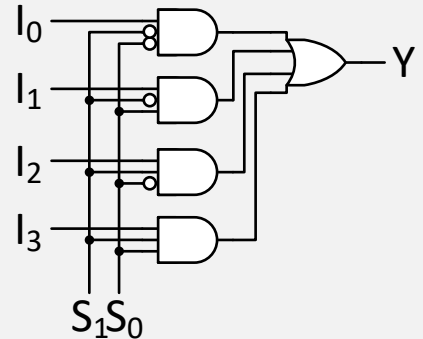
- Identificar entradas e saídas do circuito
- Determinar a tabela de verdade ou a expressão lógica de cada uma das saídas
- Simplificar as expressões de saída
- Descrever o circuito numa linguagem de descrição de hardware (VHDL)
- Verificar a correção do circuito por simulação
- Implementar e testar o circuito

# Circuitos combinatórios típicos

Circuitos combinatórios MSI (*Medium Scale Integration*)

# Multiplexer

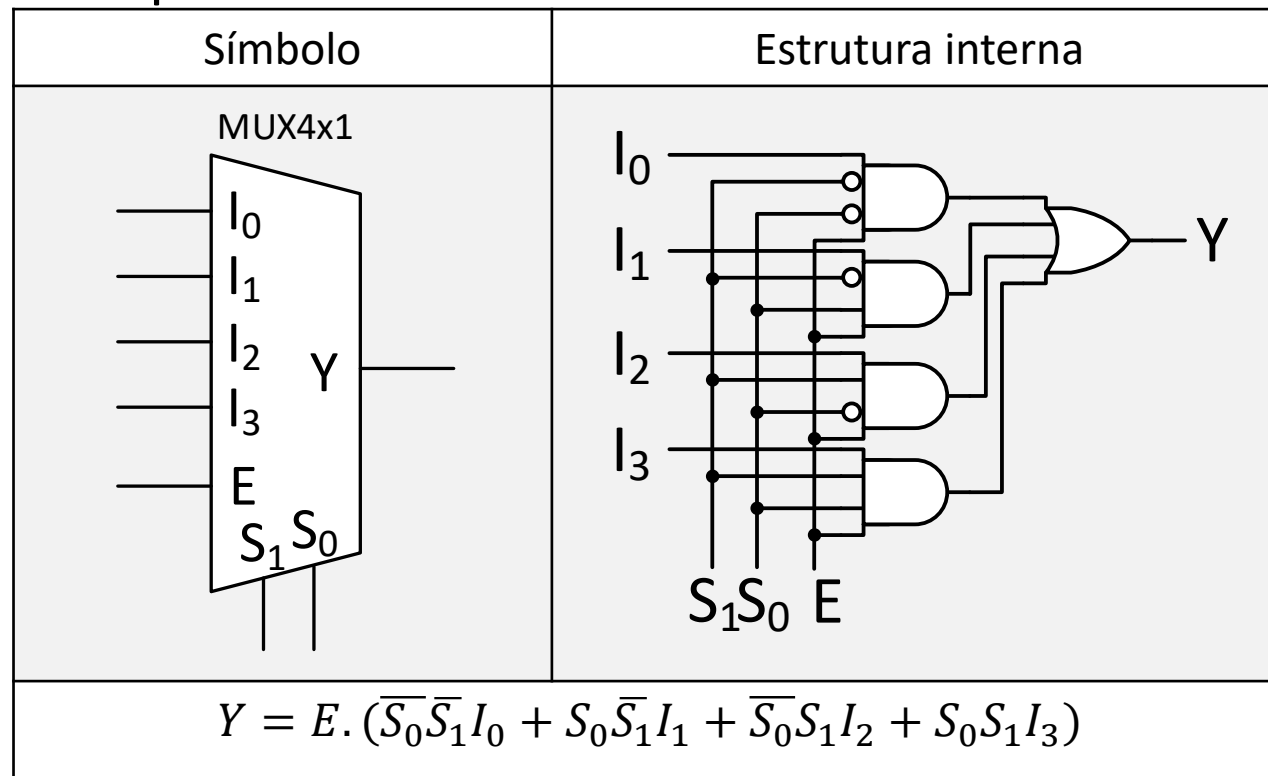
- Circuito que multiplexa num canal de dados (Y) múltiplos canais de dados ( $I_n$ ) baseado numa palavra de controlo ( $S_x$ )
- Exemplos de *multiplexers*

Mux 2x1		Mux 4x1	
Símbolo	Estrutura interna	Símbolo	Estrutura interna
			
$Y = \bar{S}.I_0 + S.I_1$		$Y = \bar{S_0}\bar{S_1}I_0 + S_0\bar{S_1}I_1 + \bar{S_0}S_1I_2 + S_0S_1I_3$	

Mux 2x1			
S	I1	I0	$Y = \bar{S}.I_0 + S.I_1$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

# Multiplexer com controlo de *Enable*

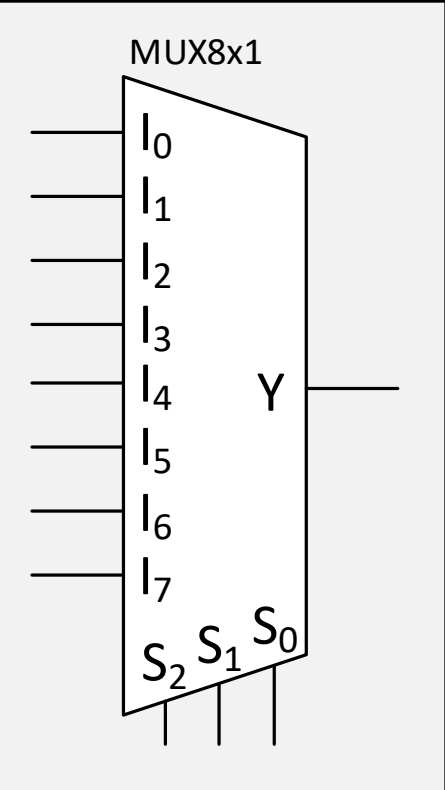
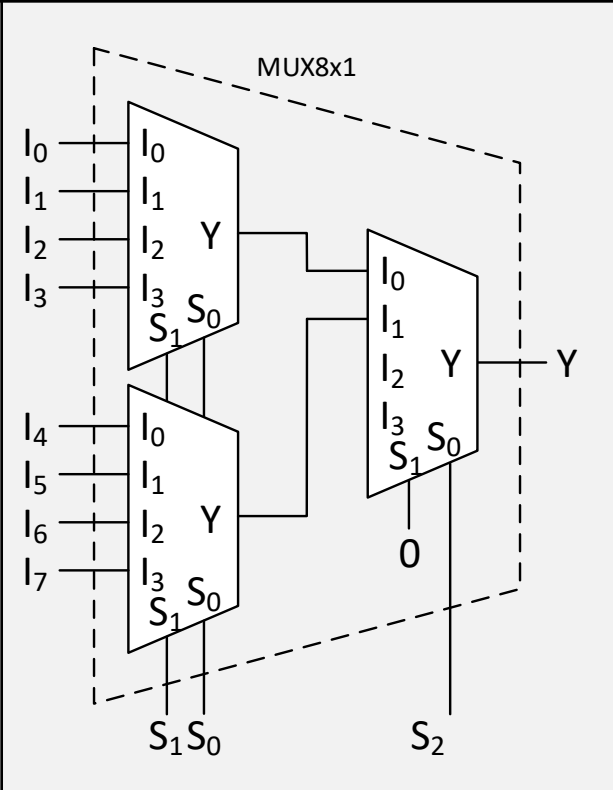
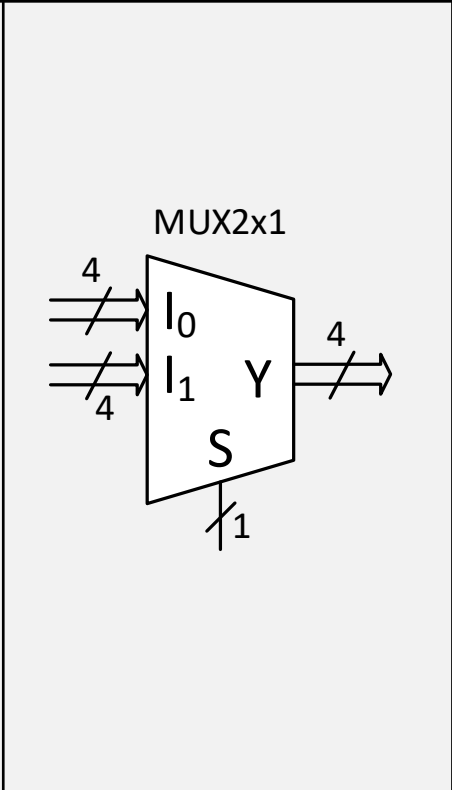
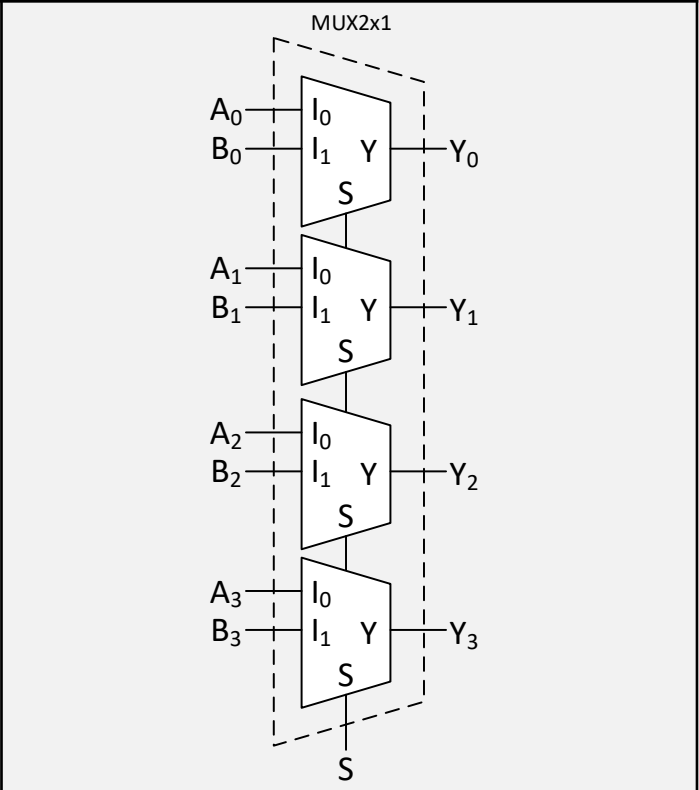
- Um *multiplexer* pode incluir uma entrada de *Enable* que controla o estado de ativação da saída
- Enquanto *Enable* desativo, a saída permanece desativa independentemente dos valores presentes nas restantes entradas



Mux 4x1 com controlo de <i>Enable</i> (active high)			
E	S <sub>1</sub>	S <sub>0</sub>	Y
0	-	-	0
1	0	0	I <sub>0</sub>
1	0	1	I <sub>1</sub>
1	1	0	I <sub>2</sub>
1	1	1	I <sub>3</sub>

# Expansão de *multiplexers*

- Exemplo de Mux 8x1 baseado em Mux 4x1 e de Mux 2x1 de palavras a 4 bits baseado em Mux 2x1 de palavras de 1 bit

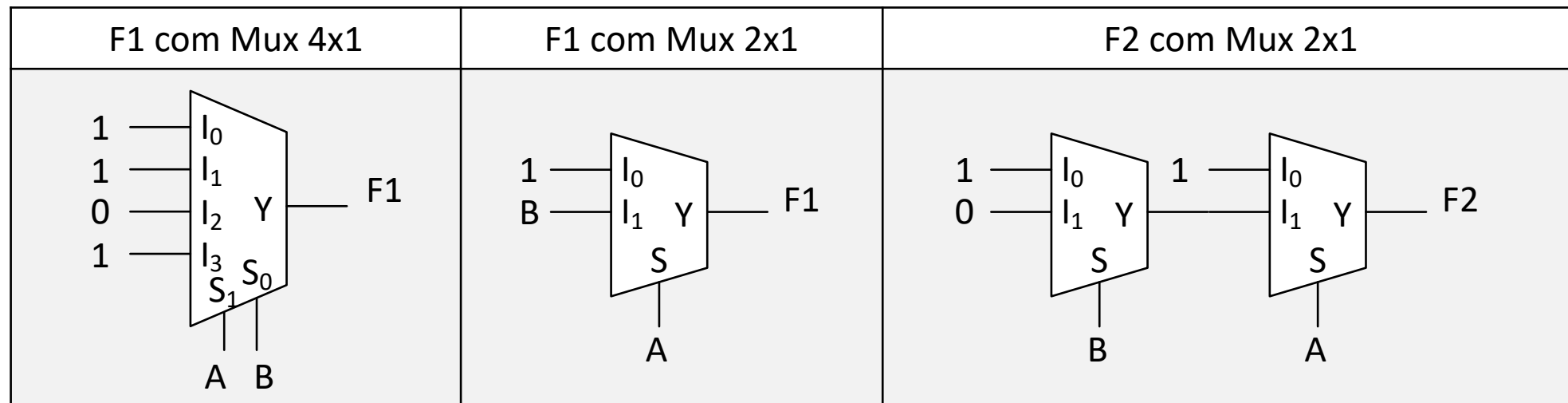
Símbolo	Estrutura interna	Símbolo	Estrutura interna
			



# Implementação de funções booleanas com *multiplexers*

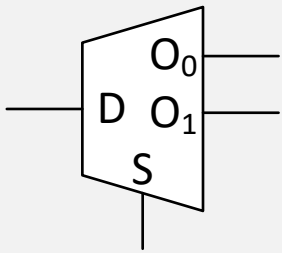
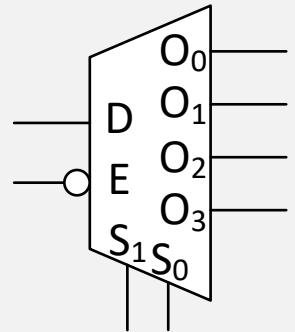
- Os multiplexers também podem ser usados para implementar funções lógicas

A	B	$F_1 = \bar{A} + B$	$F_2 = \bar{A} + \bar{B}$
0	0	1	1
0	1	1	1
1	0	0	1
1	1	1	0



# Demultiplexer

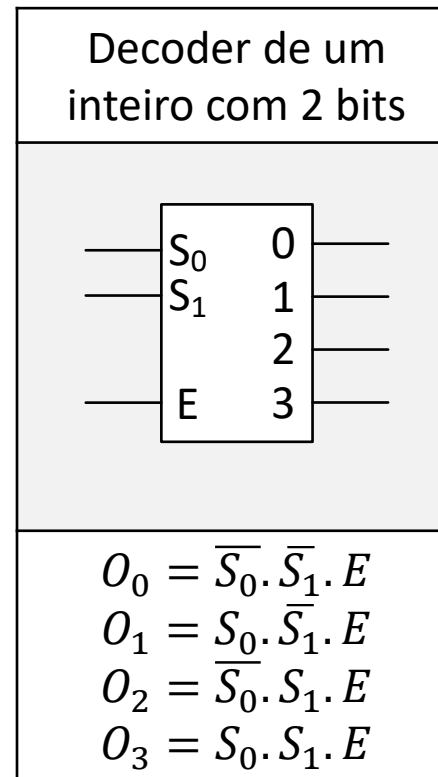
- O circuito *demultiplexer* desempenha a funcionalidade inversa do *multiplexer*, ou seja, coloca a informação presente no canal de dados (D) num de múltiplos canais de dados ( $O_n$ ) baseado numa palavra de controlo ( $S_x$ )
- Exemplos de *demultiplexers*
- Tal como o *multiplexer*, pode ter controlo de *Enable*
- O mesmo hardware pode ser usado na perspetiva de um circuito descodificador

DeMux 1x2	DeMux 1x4
	
$O_0 = \bar{S} \cdot D$ $O_1 = S \cdot D$	$O_0 = \bar{E} \cdot \bar{S}_0 \cdot \bar{S}_1 \cdot D$ $O_1 = \bar{E} \cdot S_0 \cdot \bar{S}_1 \cdot D$ $O_2 = \bar{E} \cdot \bar{S}_0 \cdot S_1 \cdot D$ $O_3 = \bar{E} \cdot S_0 \cdot S_1 \cdot D$

DeMux 1x4 com controlo de <i>Enable (active low)</i>						
E	$S_1$	$S_0$	$O_0$	$O_1$	$O_2$	$O_3$
1	-	-	0	0	0	0
0	0	0	D	0	0	0
0	0	1	0	D	0	0
0	1	0	0	0	D	0
0	1	1	0	0	0	D

# Descodificador

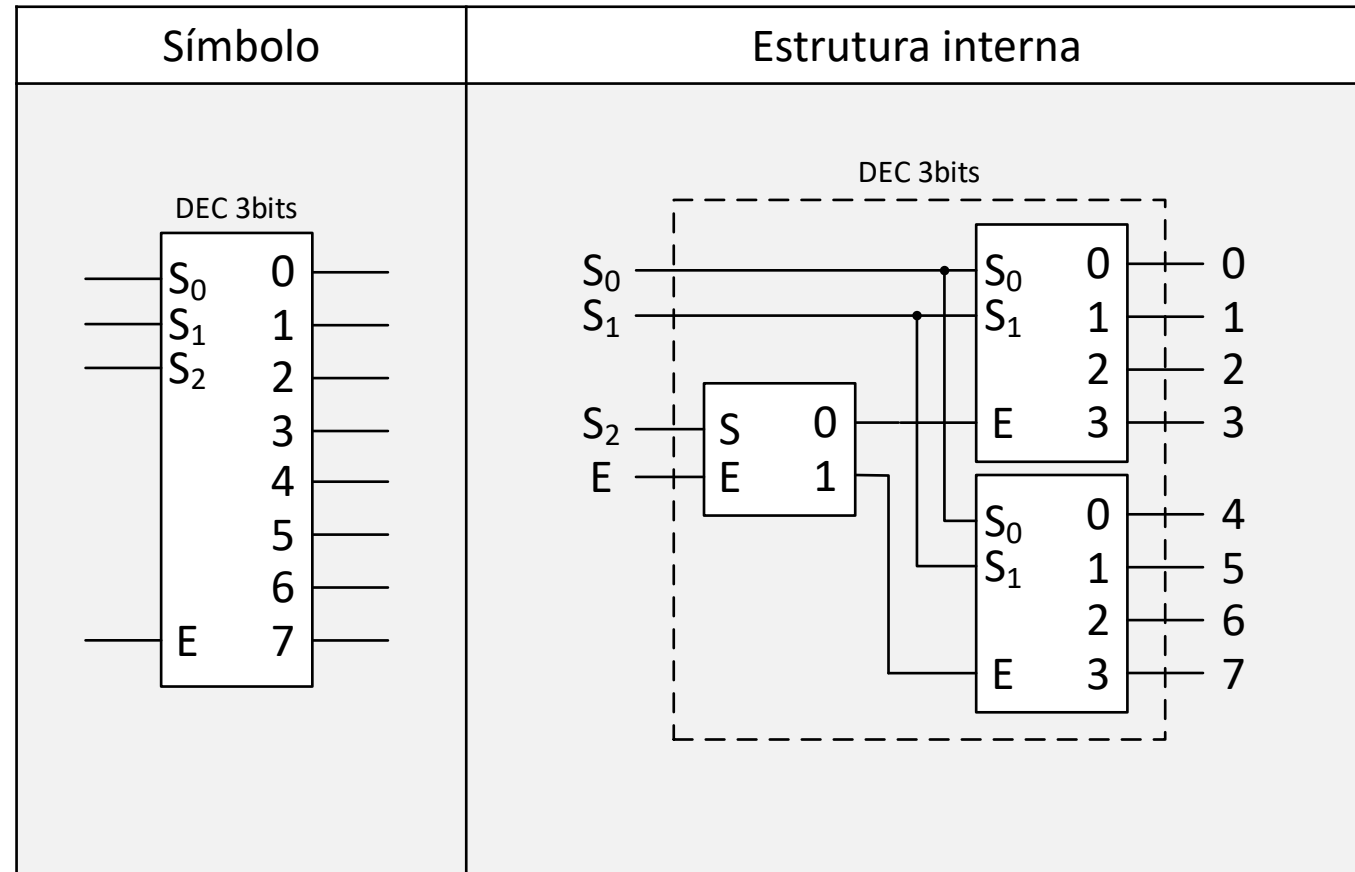
- O circuito *demultiplexer* pode ser usado na perspetiva de um descodificador. A entrada de dados passa a desempenhar o papel de controlo de *Enable*
- Enquanto *Enable* (E) ativo, apenas uma e uma só saída (0..3) permanece ativa função do código do número presente na entrada ( $S_n$ )
- Todas as saídas permanecem desativas enquanto a entrada *Enable* permanecer desativa



Decoder 2 bits						
E	$S_1$	$S_0$	$O_0$	$O_1$	$O_2$	$O_3$
0	-	-	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

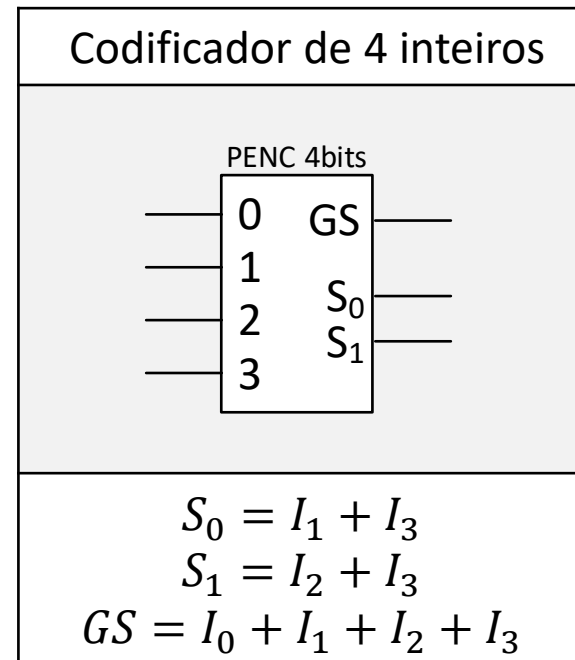
# Expansão de decodificadores

- Exemplo de um decodificador de números a 3 bits



# Codificador sem prioridade

- Circuito inverso do decodificador, ou seja, apresenta na saída ( $S_n$ ) o código do número correspondente à entrada ativa (0..3)
- Apenas uma entrada ativa de cada vez
- Necessário saída indicadora de alguma entrada ativa ou nenhuma entrada ativa (GS – *Group Select*)

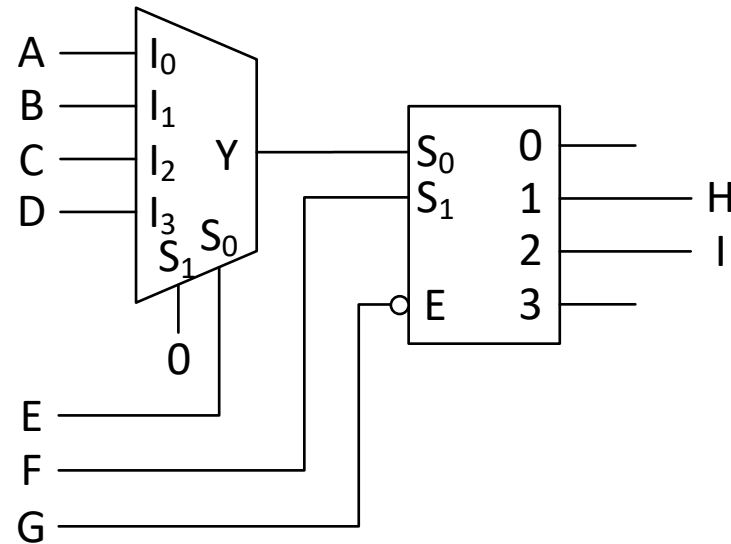


$I_3$	$I_2$	$I_1$	$I_0$	$S_1$	$S_0$	GS
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	1
0	1	0	0	1	0	1
1	0	0	0	1	1	1

# Exercício

# Análise de circuito combinatório

- Determine as expressões lógicas de H e I



# Resolução do exercício

---

- Impraticável usar tabela de verdade ou mapa de Karnaugh, dado o número de variáveis de entrada ( $7 \text{ variáveis} \Rightarrow 2^7 = 128 \text{ combinações}$ )
- Impraticável escrever as expressões genéricas do MUX e DEC e substituir pelas variáveis de entrada do circuito
- Deve analisar-se o circuito das saídas para as entradas e identificar situações que condicionam a ativação das saídas
- Análise:
  - Ambas as saídas dependem do decodificador ter a entrada de *Enable* ativa, logo G tem de ser 0 que se traduz na expressão H e I por  $\bar{G}.xxx$
  - No caso de H,  $S1_{DEC}$  tem de ser 0; no caso de I,  $S1_{DEC}$  tem de ser 1, logo
$$H = \bar{G}.\bar{F}.xxx \quad I = \bar{G}.F.xxx$$
  - No caso de H,  $S0_{DEC}$  tem de ser 1; no caso de I,  $S0_{DEC}$  tem de ser 0. Como  $S0_{DEC}$  depende da saída do MUX, analisa-se o MUX
  - As entradas 2 e 3 do Mux nunca serão selecionadas estando  $S1_{MUX}$  ligado a 0, logo C e D são irrelevantes para H e I
  - Para Y ser 1, ou  $E=0$  e  $A=1$  ou  $E=1$  e  $B=1$ , logo  $H = \bar{G}.\bar{F}.(\bar{E}.A + E.B)$
  - Para Y ser 0, ou  $E=0$  e  $A=0$  ou  $E=1$  e  $B=0$ , logo  $I = \bar{G}.F.(\bar{E}.\bar{A} + E.\bar{B})$
  - Em alternativa, no contexto da saída I, pode considerar-se a expressão de Y para ser 1 e negá-la, ou seja,
$$I = \bar{G}.F.\overline{\bar{E}.A + E.B} = \bar{G}.F.(E + \bar{A}).(\bar{E} + \bar{B})$$