

Um mouse é um dispositivo que contém um, dois, ou três botões, dependendo da estimativa que os projetistas dão para a capacidade intelectual de seus usuários (Tanenbaum, Bos; 2016).

# Mux e Demux

Paulo Ricardo Lisboa de Almeida

# Decodificador

Crie um circuito que

Recebe duas entradas  $A_0$  e  $A_1$ , que representam um número de 2 bits

Possui 4 saídas  $O_0$ ,  $O_1$ ,  $O_2$  e  $O_3$

Se o número de entrada é  $00_2$ ,  $O_0$  deve ser 1, e as demais saídas são zero

Se o número de entrada é  $01_2$ ,  $O_1$  deve ser 1, e as demais saídas são zero

Se o número de entrada é  $10_2$ ,  $O_2$  deve ser 1, e as demais saídas são zero

...

# Decodificador



# Decodificador

$A_1$	$A_0$	$O_0$	$O_1$	$O_2$	$O_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

# Decodificador

$A_1$	$A_0$	$O_0$	$O_1$	$O_2$	$O_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$$O_0 = \overline{A_0} \cdot \overline{A_1}$$

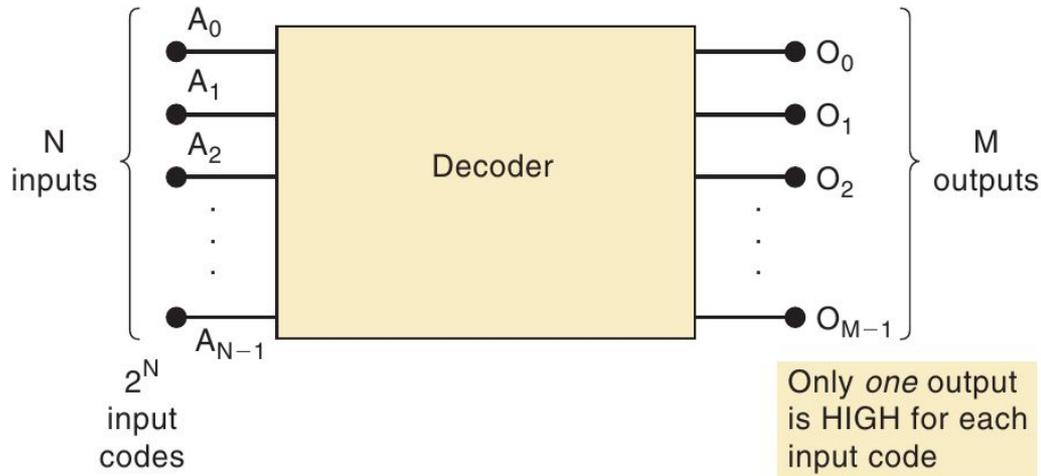
$$O_1 = \overline{A_0} \cdot A_1$$

$$O_2 = A_0 \cdot \overline{A_1}$$

$$O_3 = A_0 \cdot A_1$$

# Decodificador

**Decodificador (decoder):** Circuito lógico que aceita  $n$  entradas representando um número binário, e ativa uma de suas  $2^n$  saídas correspondente a esse número.



# Faça você mesmo

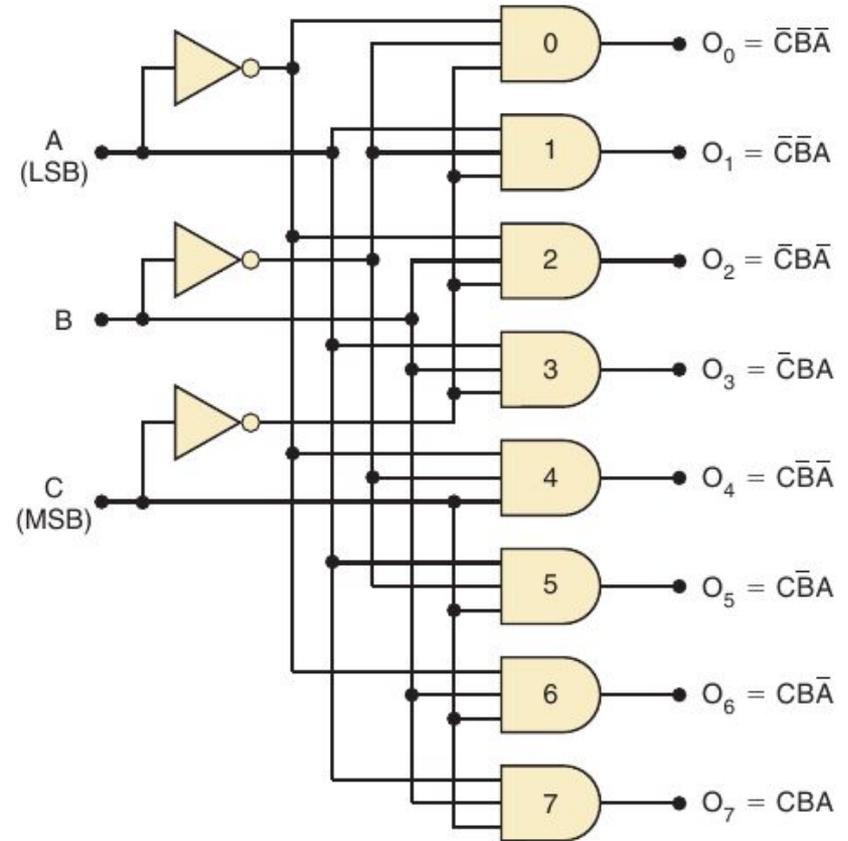
Considere o decodificador a seguir, que possui duas entradas (mais uma enable - E) e quatro saídas. Se  $E = 1$ , o circuito funciona normalmente, se  $E = 0$ , as saídas ficam em zero, independentemente da entrada.

Utilize esse circuito (pode usar mais de um) para montar um decodificador de 3 entradas  $A_2A_1A_0$ .



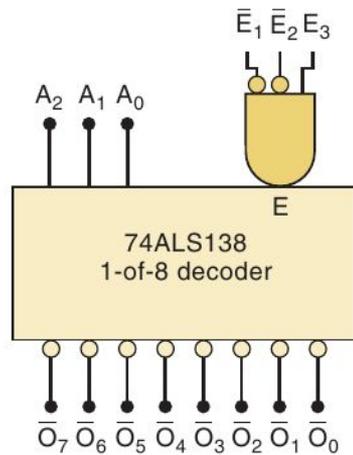
# Outro Exemplo

C	B	A	O <sub>7</sub>	O <sub>6</sub>	O <sub>5</sub>	O <sub>4</sub>	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

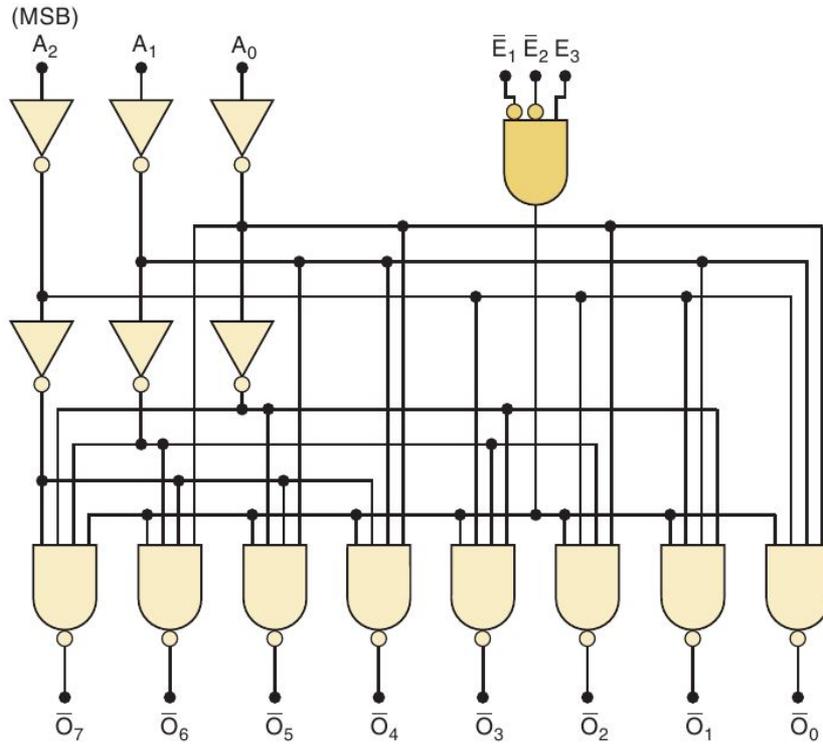


# 73138

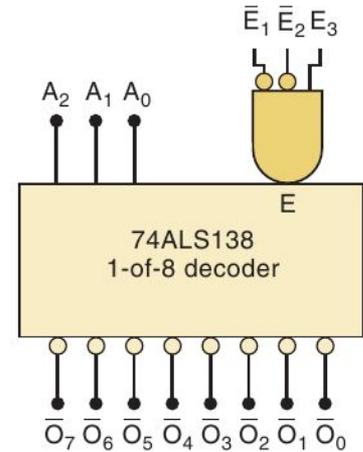
Circuito *ativo em baixa*. A porta seleccionada tem uma saída zero.



# 73138



$\bar{E}_1$	$\bar{E}_2$	$E_3$	Outputs
0	0	1	Respond to input code $A_2A_1A_0$
1	X	X	Disabled – all HIGH
X	1	X	Disabled – all HIGH
X	X	0	Disabled – all HIGH

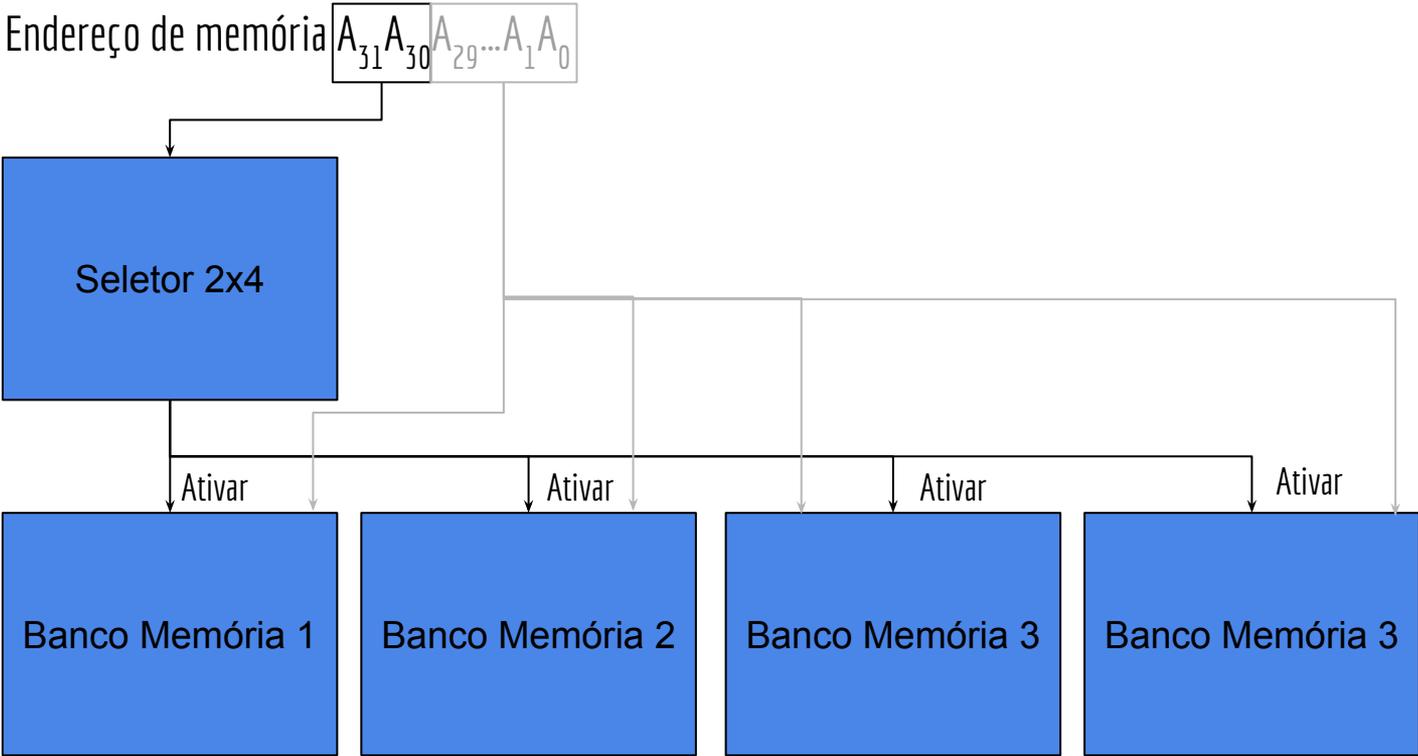


# Exemplo de uso

Decodificadores são usados em diversos circuitos

Exemplo: em memórias, onde o decodificador ativa uma memória específica de acordo com um endereço recebido.

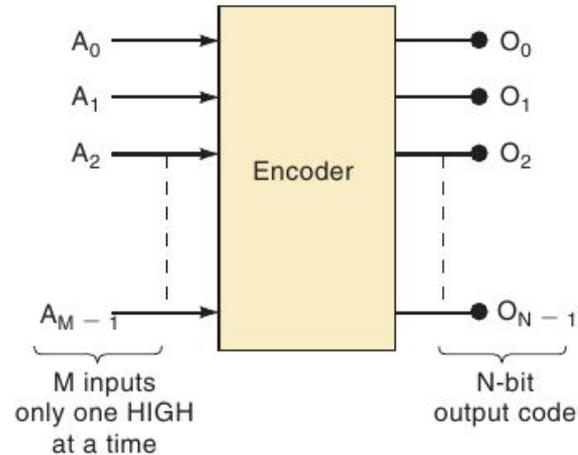
# Exemplo de uso



# Codificador

Um codificador faz o processo contrário de um decoder

**Codificador (encoder):** Circuito lógico que aceita  $2^n$  entradas, cada uma representando um número binário, e ativa suas  $n$  saídas representando o número.



# Faça você mesmo

Considere o Codificador a seguir, que possui quatro entradas e duas saídas.

Prioridade: se mais de um  $A_x$  for 1, codifique o maior  $x$

Mostre o circuito para esse codificador.



# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
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		



# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1



# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

$$O_0 = \dots$$

	$\bar{A}_1\bar{A}_0$	$\bar{A}_1A_0$	$A_1A_0$	$A_1\bar{A}_0$
$\bar{A}_3\bar{A}_2$	0	0	1	1
$\bar{A}_3A_2$	0	0	0	0
$A_3A_2$	1	1	1	1
$A_3\bar{A}_2$	1	1	1	1

# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

$$O_0 = A_3 + \bar{A}_2 A_1$$

	$\bar{A}_1 \bar{A}_0$	$\bar{A}_1 A_0$	$A_1 A_0$	$A_1 \bar{A}_0$
$\bar{A}_3 \bar{A}_2$	0	0	1	1
$\bar{A}_3 A_2$	0	0	0	0
$A_3 A_2$	1	1	1	1
$A_3 \bar{A}_2$	1	1	1	1

# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

$$O_1 = \dots$$

	$\bar{A}_1\bar{A}_0$	$\bar{A}_1A_0$	$A_1A_0$	$A_1\bar{A}_0$
$\bar{A}_3\bar{A}_2$	0	0	0	0
$\bar{A}_3A_2$	1	1	1	1
$A_3A_2$	1	1	1	1
$A_3\bar{A}_2$	1	1	1	1

# Faça você mesmo

$A_3$	$A_2$	$A_1$	$A_0$	$O_1$	$O_0$
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

$$O_1 = A_3 + A_2$$

	$\bar{A}_1\bar{A}_0$	$\bar{A}_1A_0$	$A_1A_0$	$A_1\bar{A}_0$
$\bar{A}_3\bar{A}_2$	0	0	0	0
$\bar{A}_3A_2$	1	1	1	1
$A_3A_2$	1	1	1	1
$A_3\bar{A}_2$	1	1	1	1

# Multiplexadores

É comum sistemas com múltiplas entradas, das quais desejamos escolher uma como a saída

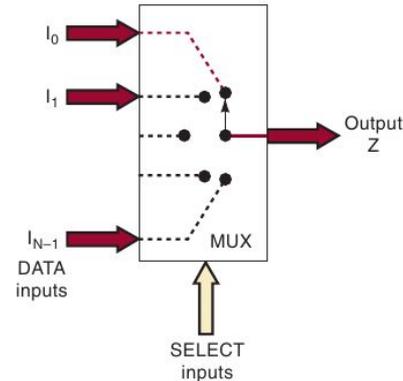
Exemplo: Um rádio de carro

Está constantemente recebendo dados de AM, FM, Flash-Drive, ...

Um “seletor” seleciona entre essas múltiplas fontes de entrada, e redireciona pra saída (que é única)

É exatamente isso o que um **multiplexador (mux)** faz

Seleciona uma dentre múltiplas entradas, e a envia para a saída



# Faça você mesmo

Crie um mux de duas entradas  $I_0$  e  $I_1$ , um seletor  $S$ , e uma saída  $Z$ .

Quando  $S=0$ ,  $I_0$  é enviado para a saída  $Z$

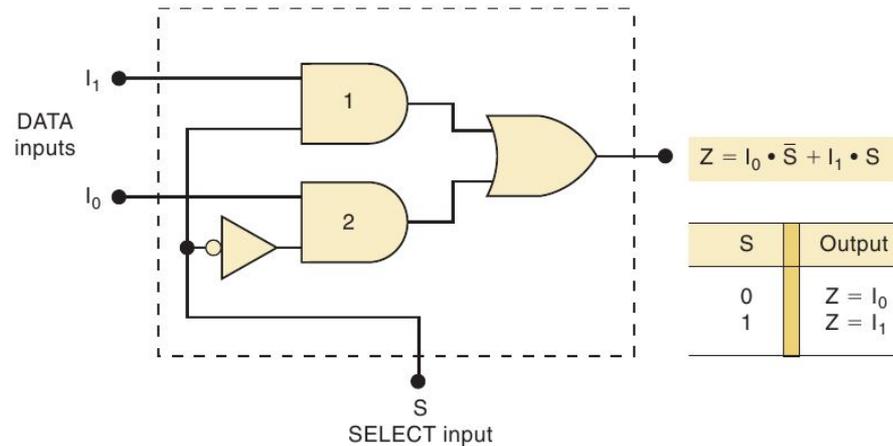
Quando  $S=1$ ,  $I_1$  é enviado para a saída  $Z$

# Faça você mesmo

Crie um mux de duas entradas  $I_0$  e  $I_1$ , um seletor  $S$ , e uma saída  $Z$ .

Quando  $S=0$ ,  $I_0$  é enviado para a saída  $Z$

Quando  $S=1$ ,  $I_1$  é enviado para a saída  $Z$



# Mux

Se o multiplexador aceitar quatro entradas,  $I_0$ ,  $I_1$ ,  $I_2$  e  $I_3$ , um seletor único funciona?

# Mux

Se o multiplexador aceitar quatro entradas,  $I_0$ ,  $I_1$ ,  $I_2$  e  $I_3$ , um seletor único funciona?

Para quatro entradas, temos quatro “endereços” possíveis

Endereço de  $I_0$ , endereço de  $I_1$ , ...

Em binário, temos os endereços  $00_2$ ,  $01_2$ ,  $10_2$  e  $11_2$

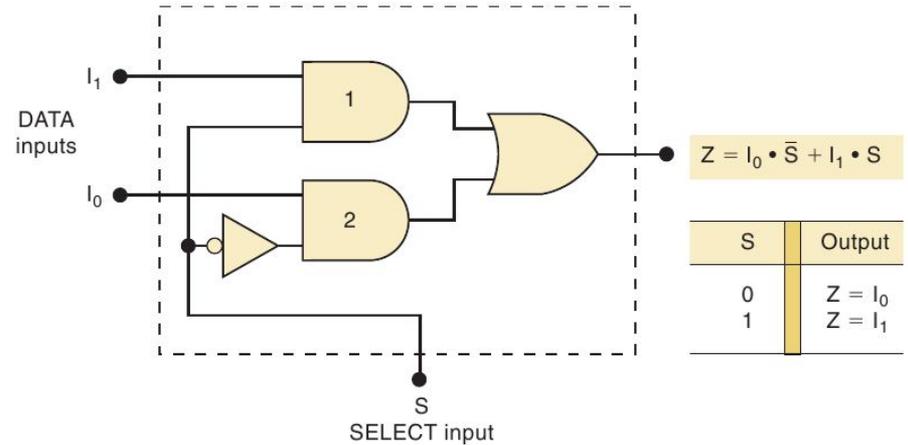
Necessários 2 bits para controlar o seletor.

$S_0$  e  $S_1$

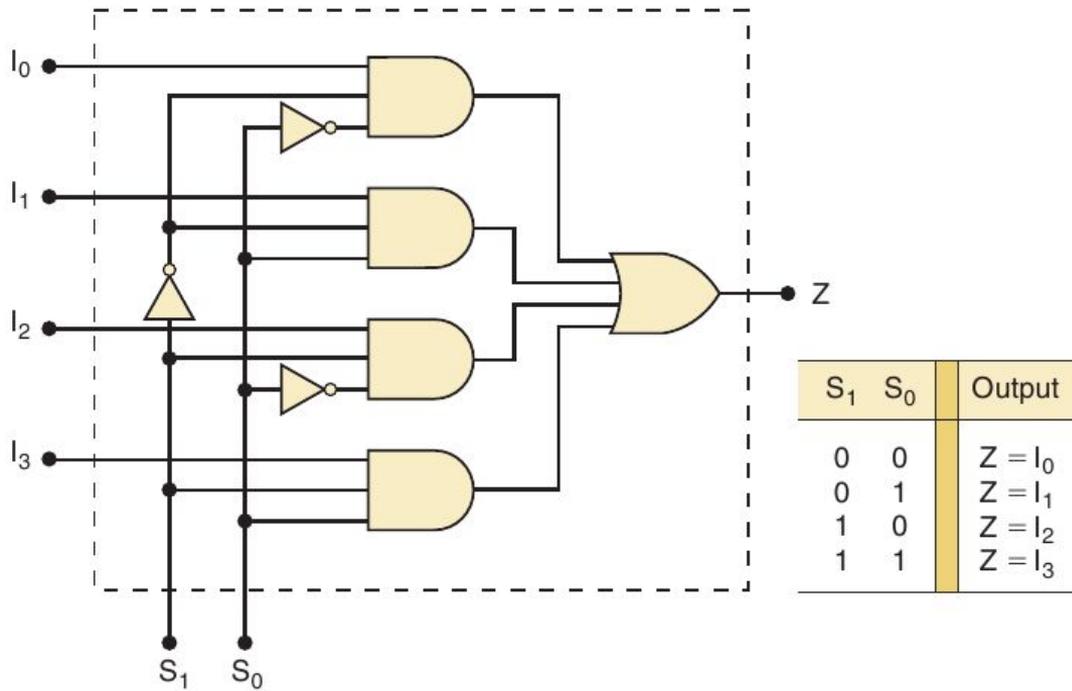
# Faça você mesmo

Crie sua versão de um multiplexador de 4 entradas

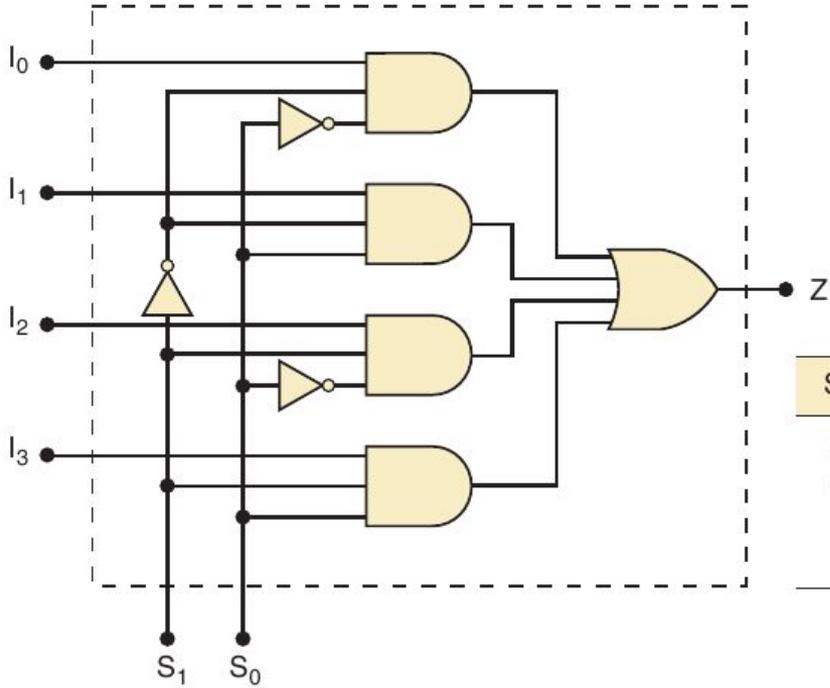
O raciocínio é análogo ao de 2 entradas, com algumas portas e entradas a mais



# Mux 4:1

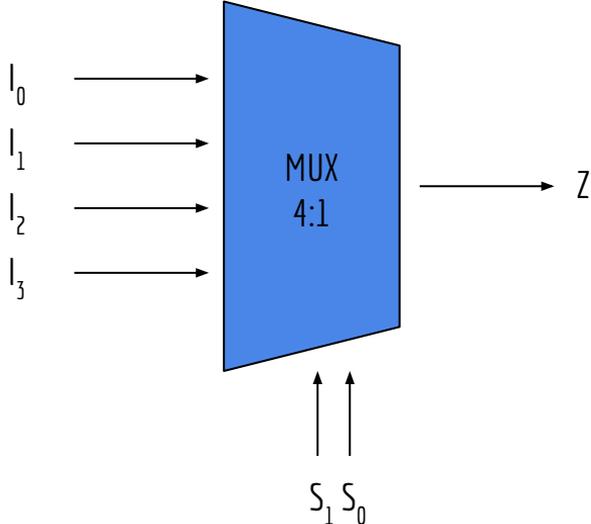


# Mux 4:1



$S_1$	$S_0$	Output
0	0	$Z = I_0$
0	1	$Z = I_1$
1	0	$Z = I_2$
1	1	$Z = I_3$

Representação de um mux 4 para 1



# Mux $n:1$

E de maneira geral, um multiplexador de  $n$  entradas possui quantos bits no seletor?

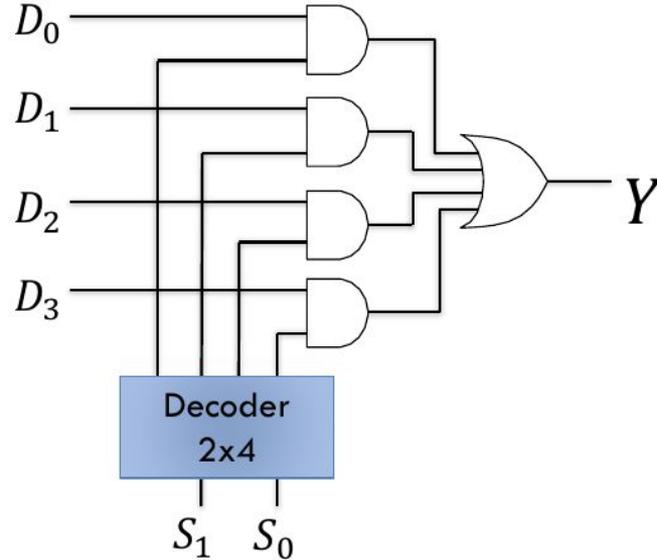
# Mux $n:1$

E de maneira geral, um multiplexador de  $n$  entradas possui quantos bits no seletor?

Considerando  $n$  uma potência de 2, são necessários  $\log_2(n)$  bits no seletor

# Outra forma

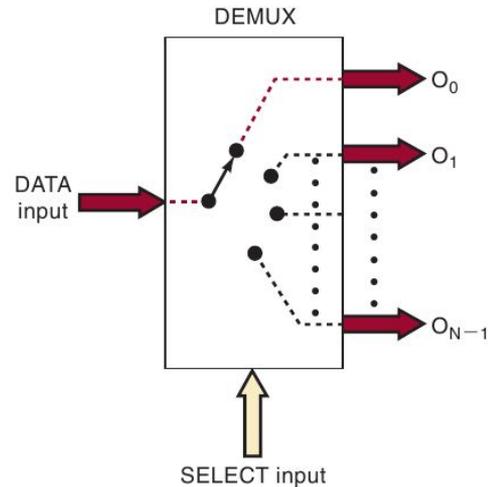
Outra forma de se criar um mux 4:1



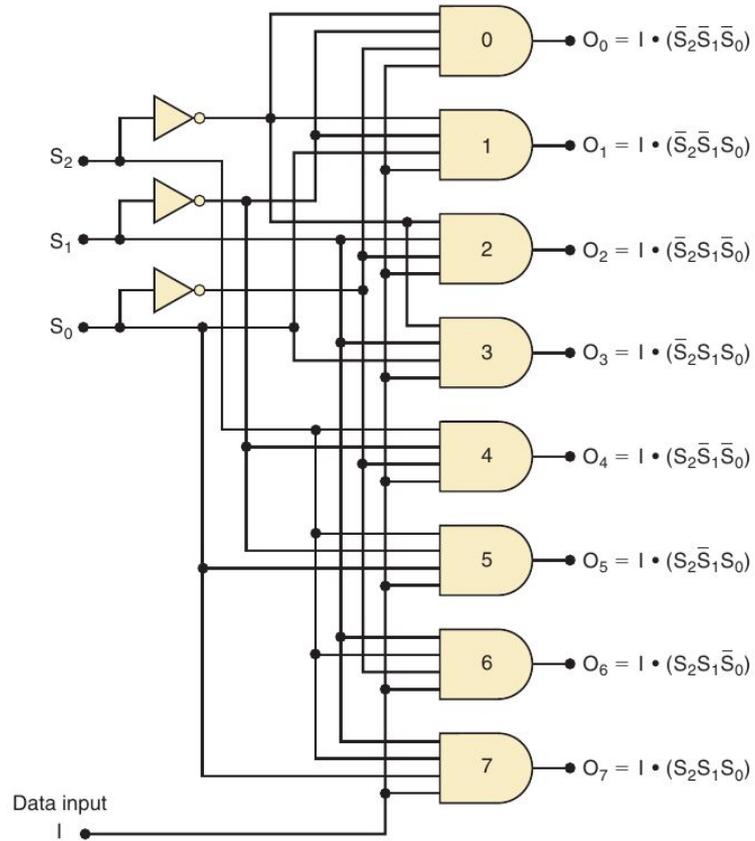
# Demultiplexador

Um **demultiplexador (demux)** faz o caminho inverso de um multiplexador

Recebe uma entrada  $I$ , e de acordo com o seletor, a envia para uma de suas  $n$  saídas  $O_0, O_1, \dots, O_{n-1}$

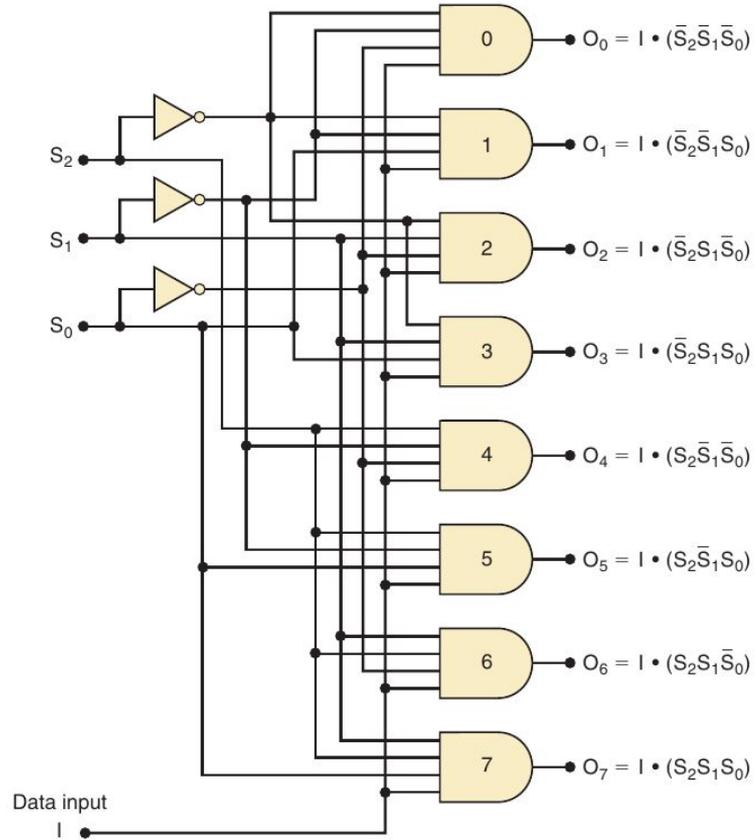
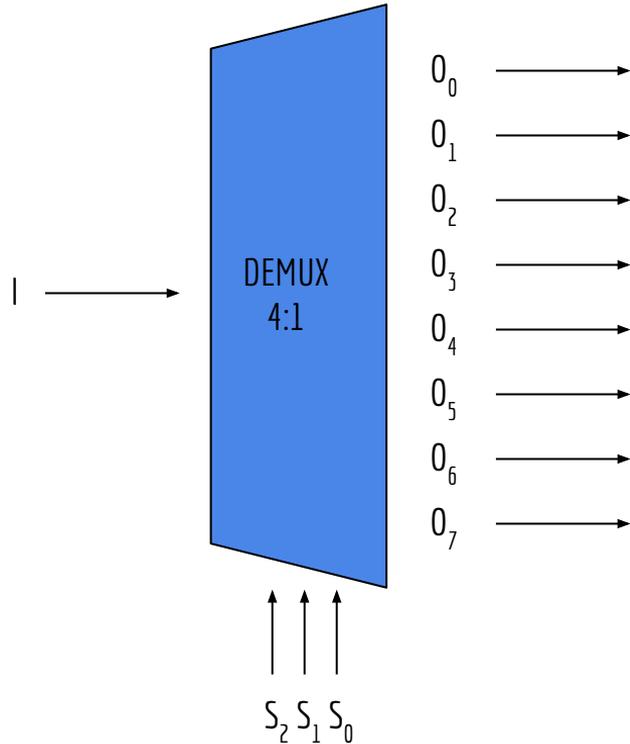


# Demultiplexador 1:8



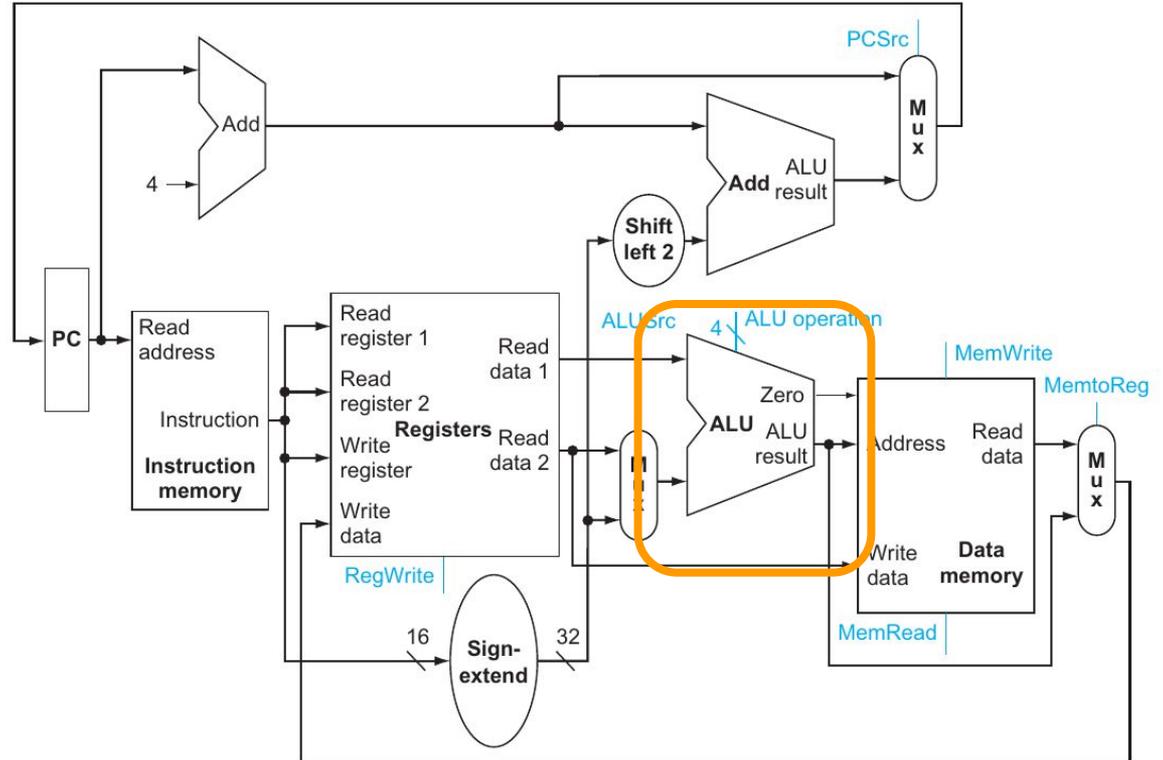
# Demultiplexador 1:8

Representação de um demux 1 para 8



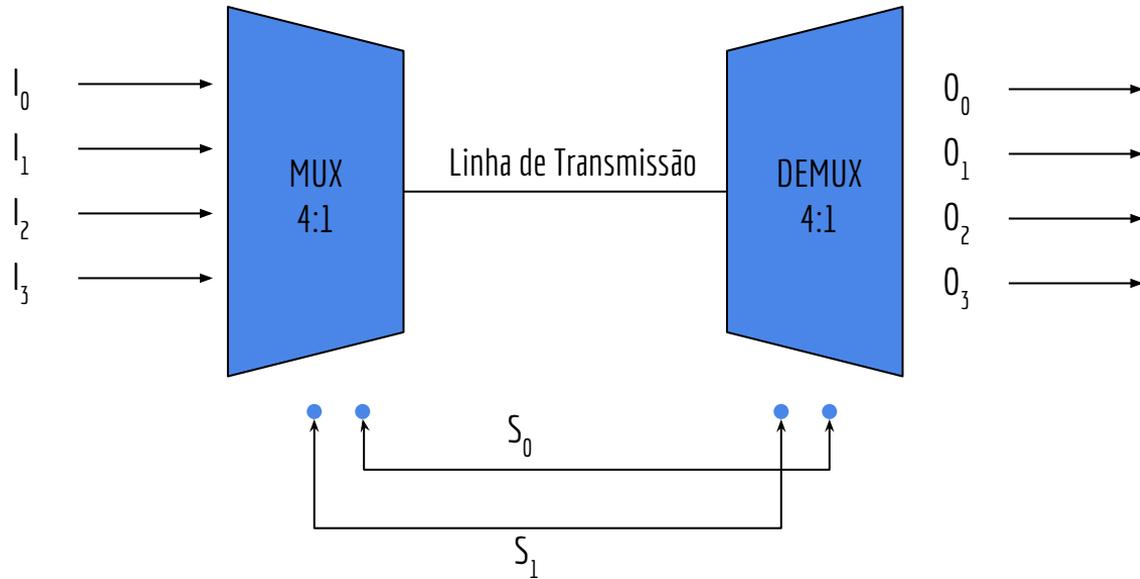
# Exemplos de uso

O seletor de operação da ALU



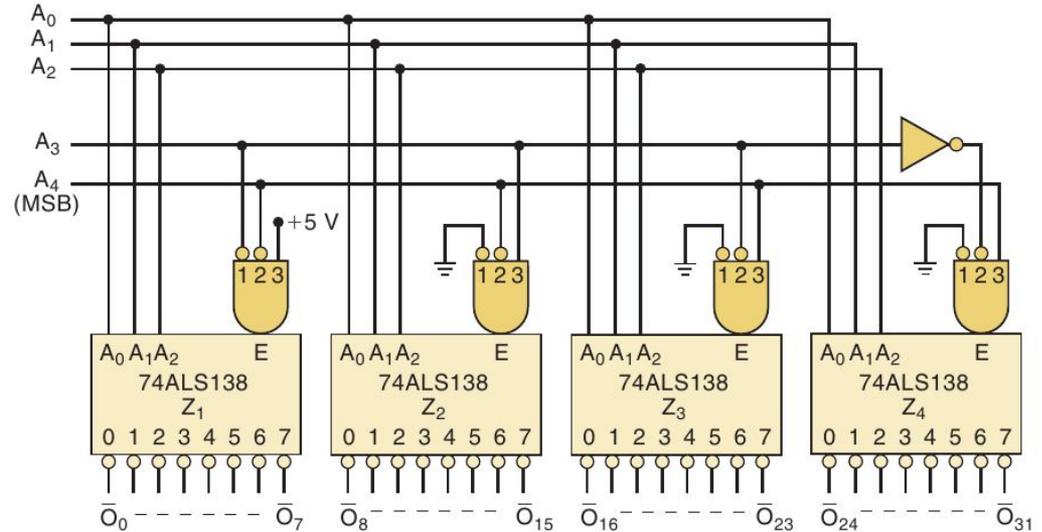
# Exemplos de uso

Transmissão e reconstrução de um sinal paralelo que precisa ser enviado em série



# Exercícios

1. Considere as conexões com circuitos 73138 a seguir e responda
- Qual será a saída quando  $A_4A_3A_2A_1A_0 = 01101$ ?
  - Qual o intervalo de entradas ativa alguma das saídas de Z4?



# Exercícios

2. Crie os seguintes codificadores

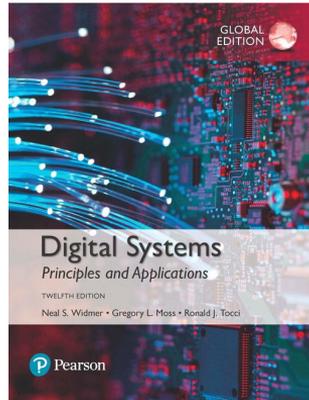
a. 2 para 1

b. 4 para 2 -> Com entradas ativas em nível baixo. Ou seja, ao enviar 0 em um  $A_x$ , estamos ativando essa entrada

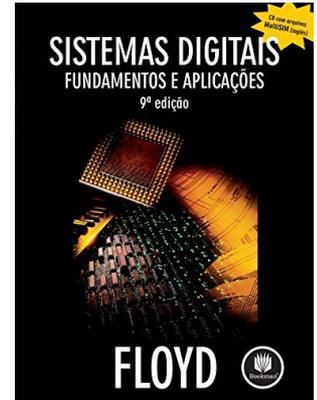
3. Utilize múltiplos multiplexadores para criar um multiplexador de 16x1. O maior multiplexador que você pode usar é 8x1.

# Referências

Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Thomas Floyd. Widmer. Sistemas Digitais: Fundamentos e Aplicações. 2009.



# Licença

Este obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

