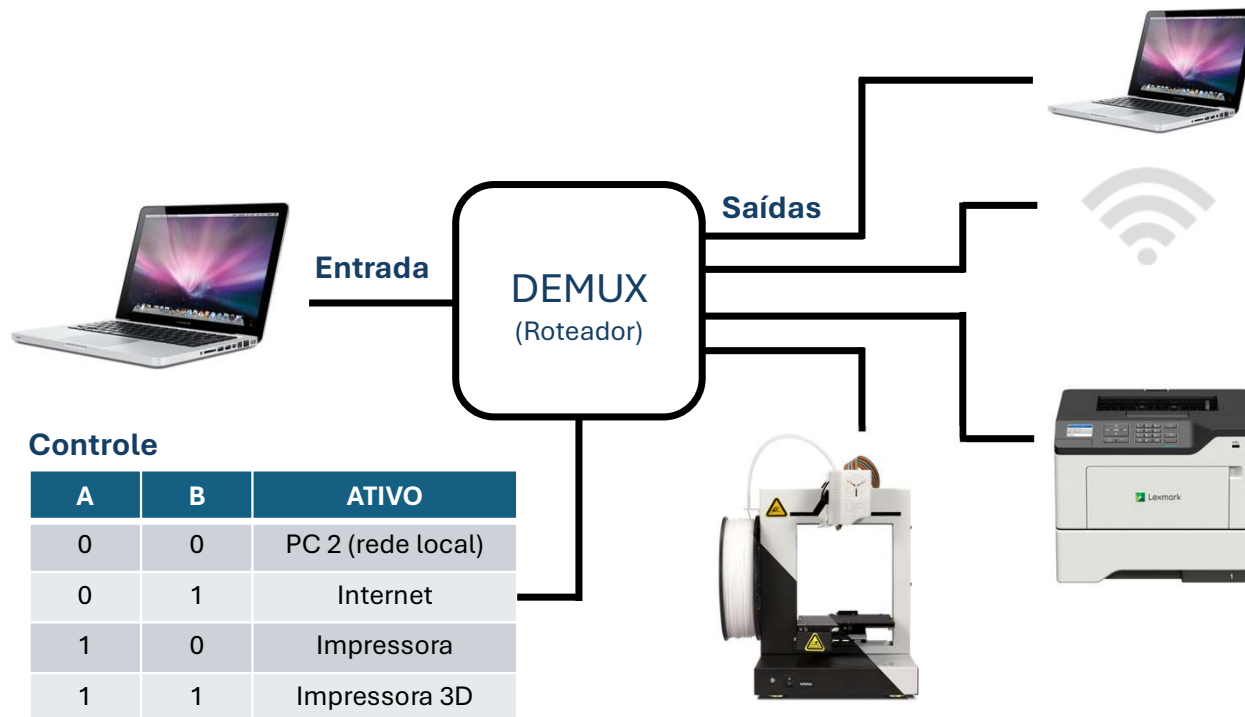


## IC – Introdução a Computação

# Circuitos Lógicos

## Demultiplexador (DMUX)

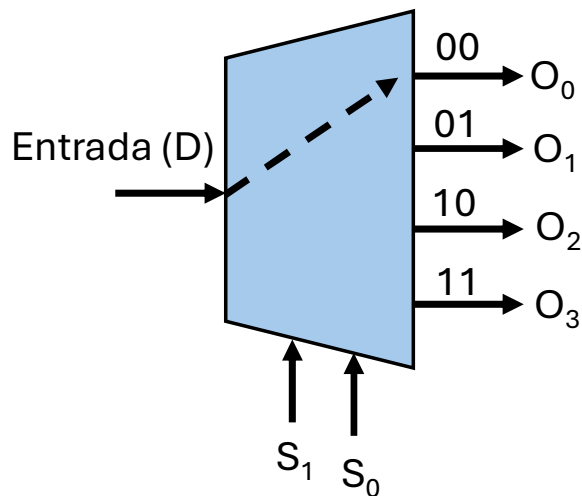
De-Multiplexador (DEMUX) efetua a função inversa a do multiplexador (MUX), ou seja, a de enviar informações contidas em um canal a vários canais de saída.



# Circuitos Lógicos

## Demultiplexadores (DMUX)

Circuito lógico que recebe uma única entrada e seleciona uma saída para transferir.



- O controle é feito por uma entrada de seleção.
- Por exemplo:
  - ❖ Seleção ( $S_1S_0$ ) = 00, seleciona Saída =  $O_0 = D$ ,  $O_1 = O_2 = O_3 = 0$
  - ❖ Seleção ( $S_1S_0$ ) = 01, seleciona Saída =  $O_1 = D$ ,  $O_0 = O_2 = O_3 = 0$
  - ❖ Seleção ( $S_1S_0$ ) = 10, seleciona Saída =  $O_2 = D$ ,  $O_0 = O_1 = O_3 = 0$
  - ❖ Seleção ( $S_1S_0$ ) = 11, seleciona Saída =  $O_3 = D$ ,  $O_0 = O_1 = O_2 = 0$

Em geral para  $2^n$  saídas, existem n variáveis de seleção.

### Aplicações:

- ❖ Em roteamento de dados
- ❖ Geração de Funções lógicas combinacionais.

# Circuitos Lógicos

## Demultiplexador 1x2 (DMUX)

O mais simples de todos

Quais são suas características?

- ❖ Quantidade de entradas = 1
- ❖ Quantidade de saídas = 2
- ❖ Bits de seleção = 1

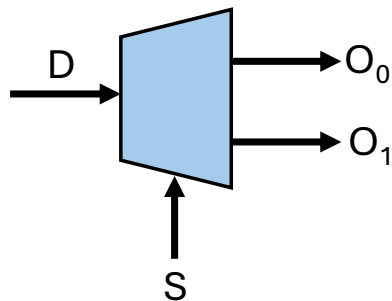
Tabela Verdade

D	S	$S_0$	$S_1$
0	0	$D = 0$	0
0	1	0	$D = 0$
1	0	$D = 1$	0
1	1	0	$D = 1$

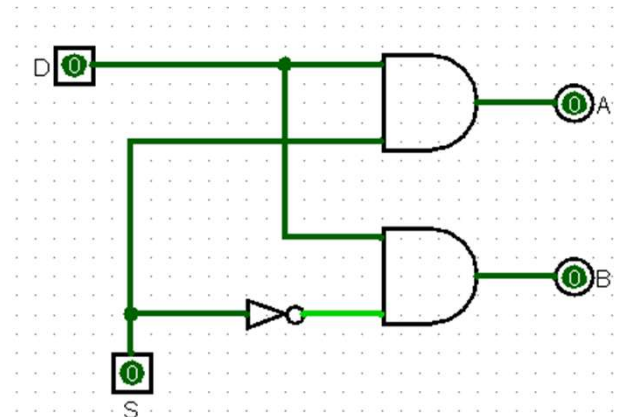
Tabela de Funcionamento

S	$O_0$	$O_1$
0	D	0
1	0	D

### Representação Gráfica:



### Nosso Circuito



$$O_0 = S'D$$

$$O_1 = SD$$

# Circuitos Lógicos

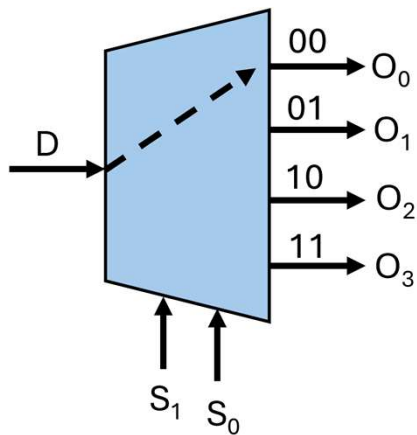
## Demultiplexador 1x4 (DMUX)

Um pouco mais complexo

Quais são suas características?

- ❖ Quantidade de entradas = 1
- ❖ Quantidade de saídas = 4
- ❖ Bits de seleção = 2

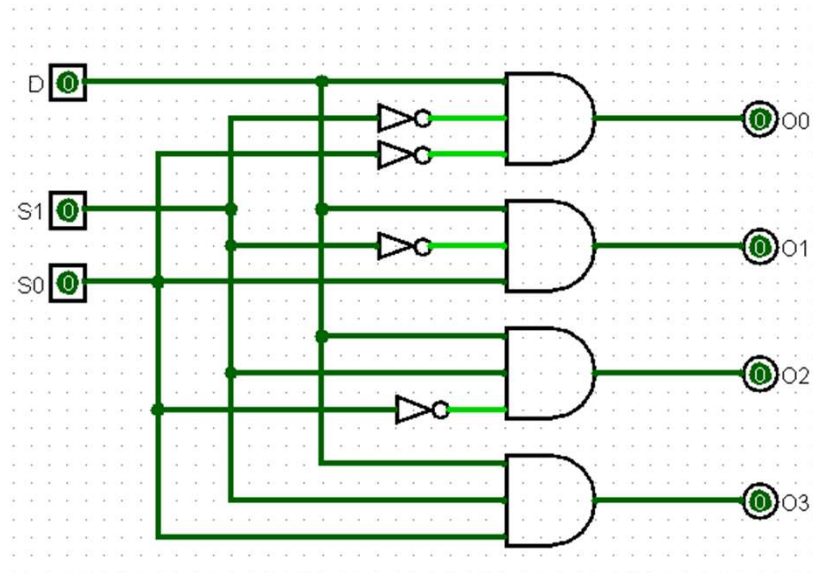
### Representação Gráfica:



### Tabela de Funcionamento

Seleção		Saída			
$S_1$	$S_0$	$O_0$	$O_1$	$O_2$	$O_3$
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

### Nosso Circuito



$$O_0 = D(S_1'S_0')$$

$$O_1 = D(S_1'S_0)$$

$$O_2 = D(S_1S_0')$$

$$O_3 = D(S_1S_0)$$

# Circuitos Lógicos

## Demultiplexador 1x8 (DMUX)

Um pouco mais complexo

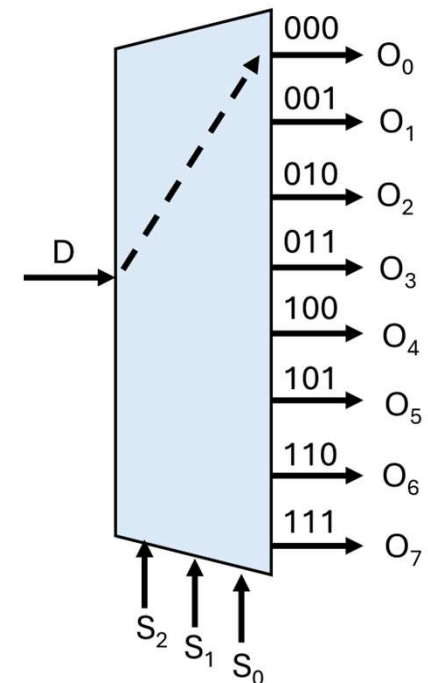
Quais são suas características?

- ❖ Quantidade de entradas = 1
- ❖ Quantidade de saídas = 8
- ❖ Bits de seleção = 3

### Tabela de Funcionamento

Seleção			Saída							
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>	O <sub>4</sub>	O <sub>5</sub>	O <sub>6</sub>	O <sub>7</sub>
0	0	0	D	0	0	0	0	0	0	0
0	0	1	0	D	0	0	0	0	0	0
0	1	0	0	0	D	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

### Representação Gráfica:



# Circuitos Lógicos

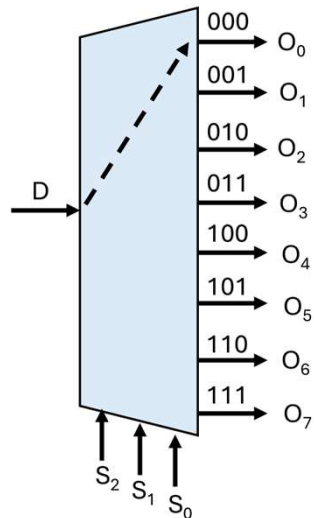
## Demultiplexador 1x8 (DMUX)

Um pouco mais complexo

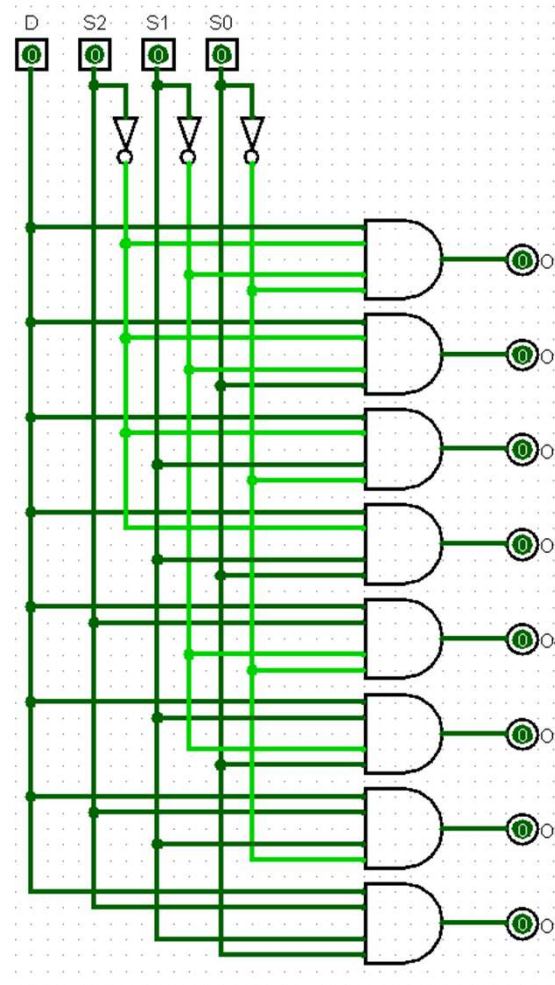
Quais são suas características?

- ❖ Quantidade de entradas = 1
- ❖ Quantidade de saídas = 8
- ❖ Bits de seleção = 3

### Representação Gráfica:



### Nosso Circuito



$$O_0 = D(S_2'S_1'S_0')$$

$$O_1 = D(S_2'S_1'S_0)$$

$$O_2 = D(S_2'S_1S_0')$$

$$O_3 = D(S_2'S_1S_0)$$

$$O_4 = D(S_2S_1'S_0')$$

$$O_5 = D(S_2S_1'S_0)$$

$$O_6 = D(S_2S_1S_0')$$

$$O_7 = D(S_2S_1S_0)$$

# Circuitos Lógicos

## Associação em Série de Multiplexadores

Os mux e demux são modulares, isto é, permitem associações que aumentam o número de canais.

Quando se necessita de um MUX com uma quantidade de canais de entrada maior do que os encontrados comercialmente em um circuito integrado, ou quando é necessário multiplexar mais de um canal de saída simultaneamente, basta fazer a associação conveniente de vários multiplexadores de forma a ampliar o número de canais de entrada ou o número de canais de saída.

Esta associação é uma forma de se ampliar a capacidade dos canais de entrada, e para tal, basta multiplexar as saídas de mais de um MUX de entrada através de um MUX de saída.

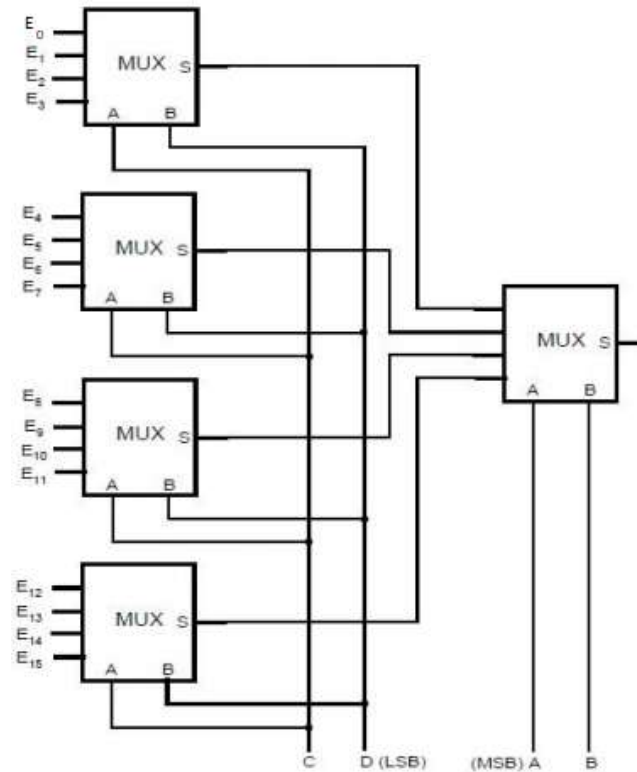


# Circuitos Lógicos

## Associação em Série de Multiplexadores

Vamos a um exemplo:

Obter um MUX de 16 entradas utilizando apenas circuitos MUX de 4 entradas.



# Circuitos Lógicos

## Associação em Paralelo de Multiplexadores

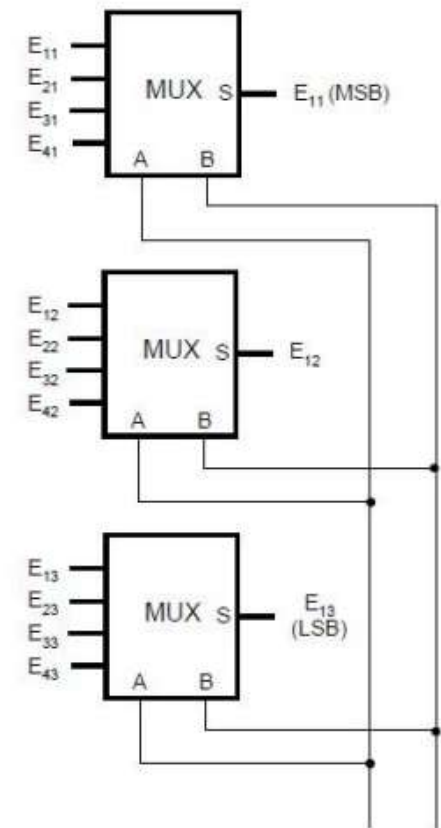
Esta associação é importante quando se necessita seleccionar informações digitais de vários bits simultaneamente.

Para isto, basta utilizar um MUX com um número de canais de entrada igual ao número de informações a serem multiplexadas sendo o número de MUX's igual ao número de bits destas informações;

Vamos olhar o exemplo a seguir:

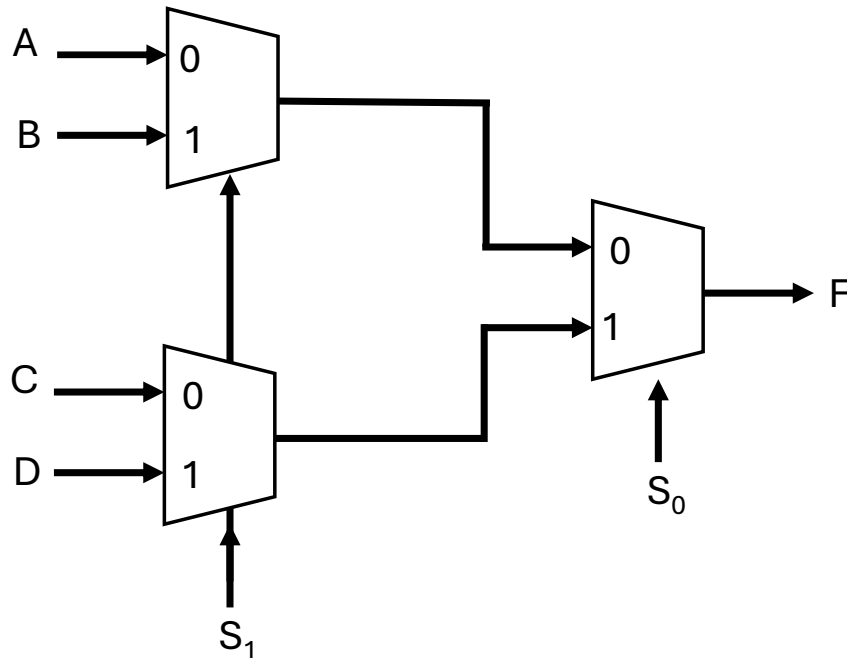
A saída será composta por uma palavra de 3 bits.

Para isso, as 4 entradas de dados diferentes (E1 , E2, E3 e E4), de 3 bits cada, será multiplexada para a saída dependendo das duas variáveis de seleção (A e B).



# Circuitos Lógicos

Vamos construir um Multiplexador 4x1 utilizando apenas o multiplexador 2x1



$$S_1 = 0 \text{ e } S_0 = 0 \quad \mathbf{F = A}$$

$$S_1 = 0 \text{ e } S_0 = 1 \quad \mathbf{F = C}$$

$$S_1 = 1 \text{ e } S_0 = 0 \quad \mathbf{F = B}$$

$$S_1 = 1 \text{ e } S_0 = 1 \quad \mathbf{F = D}$$

**Tabela de Funcionamento**

Entradas		Saída
$S_1$	$S_0$	F
0	0	A
0	1	C
1	0	B
1	1	D

# Circuitos Lógicos

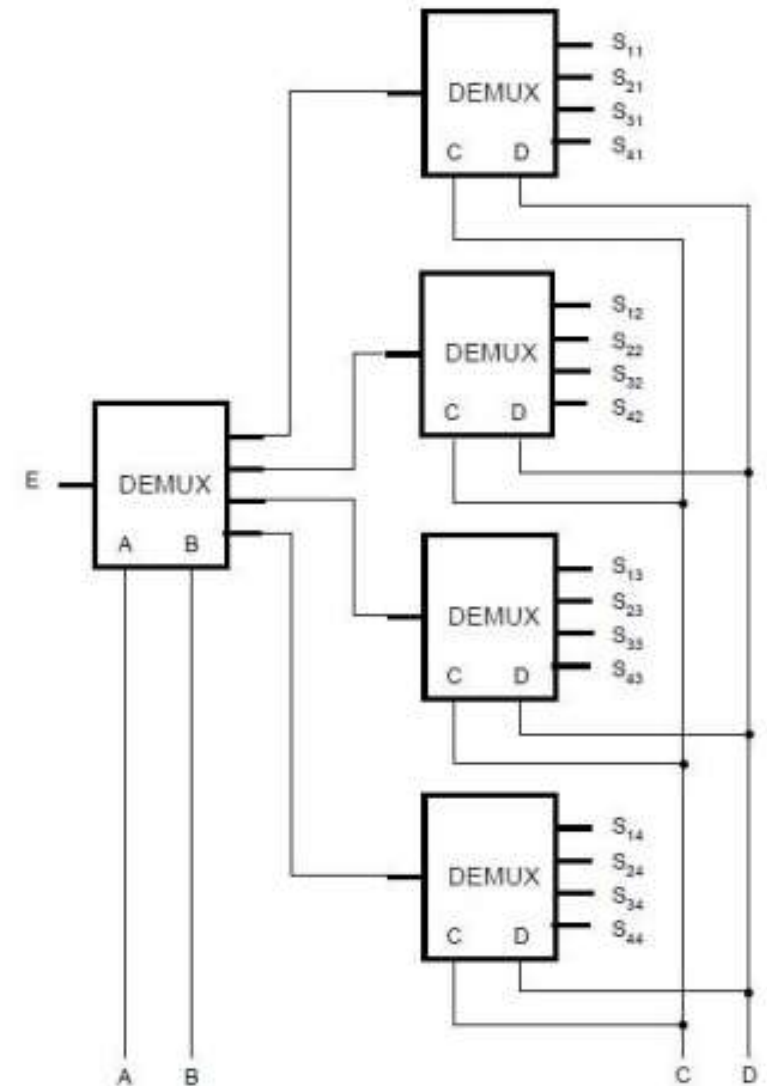
## Associação em Série de Demultiplexadores

Esta associação é utilizada para a ampliação da capacidade de canais de saída, bastando ligar os DEMUX's de saída em um DEMUX de entrada.

Vamos ver um Exemplo:

Construiu-se um DEMUX de 16 saídas utilizando apenas circuitos DEMUX de 4 saídas.

Para isto, basta utilizar 4 DEMUX's de saída demultiplexando 1 DEMUX de entrada.



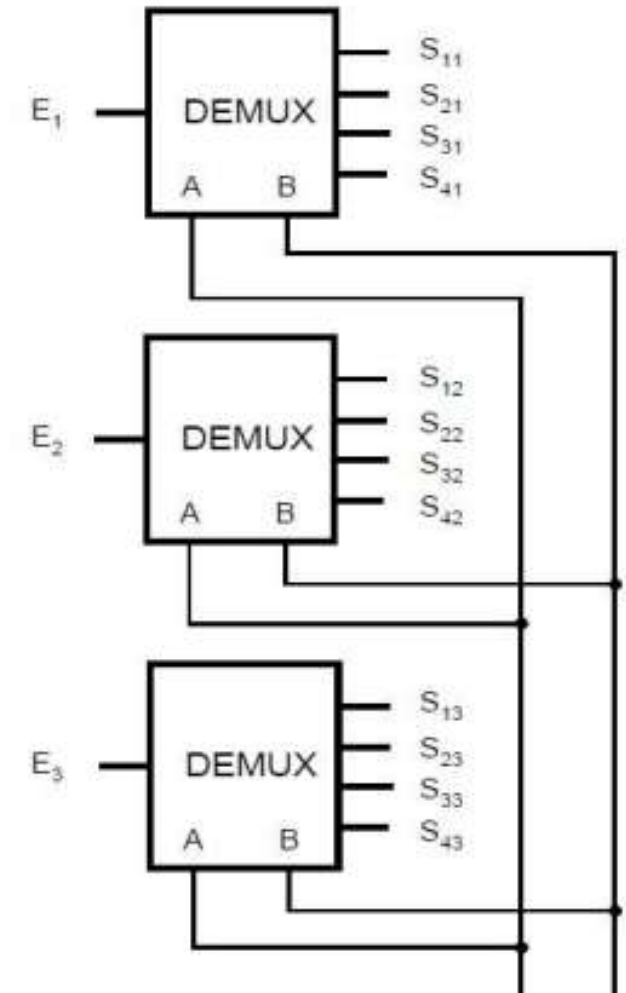
# Circuitos Lógicos

## Associação em Paralelo de Demultiplexadores

Esta associação é utilizada para a ampliação do número de canais de saída, quando se necessita demultiplexar informações digitais de vários bits simultaneamente.

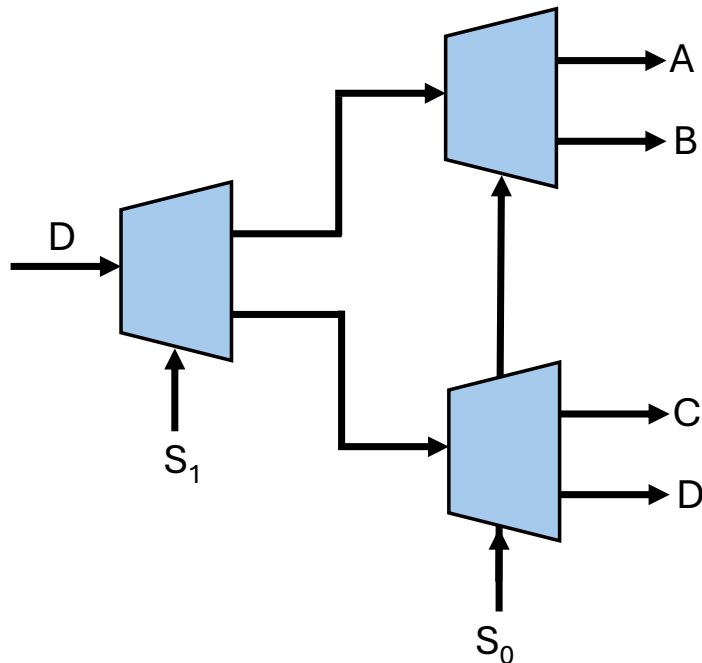
Vamos a um exemplo:

Temos uma informação composta por 3 bits ( $E_1$ ,  $E_2$ ,  $E_3$ ) para ser demultiplexada a uma das 4 saídas dos 3 DEMUX. Temos duas variáveis de seleção (A e B) e as entradas de cada DEMUX representam o dado solicitado.



# Circuitos Lógicos

Vamos construir um Demultiplexador 1x4 utilizando apenas o Demux 1x2



$S_1 = 0$  e  $S_0 = 0$    **A = D, B=C=D=0**

$S_1 = 0$  e  $S_0 = 1$    **B = D, A=C=D=0**

$S_1 = 1$  e  $S_0 = 0$    **C = D, A=B=D=0**

$S_1 = 1$  e  $S_0 = 1$    **D = D, A=B=C=0**

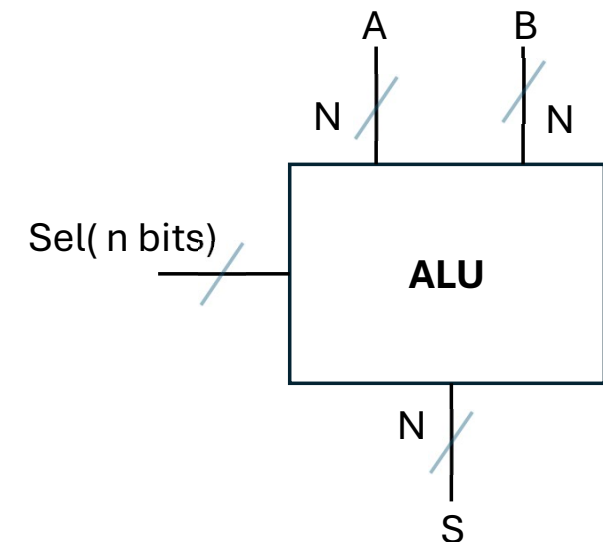
**Tabela de Funcionamento**

Seleção		Saída			
$S_1$	$S_0$	A	B	C	D
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

# Unidade Lógica-Aritmética (ALU)

## Definições

- Componente que pode executar qualquer uma das várias operações aritméticas (Somar, subtrair, incremento, etc) e lógicas (AND, OR, etc);
- As operações são realizadas com base nas entradas de controle do componente.
- A operação a ser realizada é selecionada por bits externos.



Quantidade de operações é definida por  $2^n$

Se minha ALU tem **3 bits** =  $2^3 = 8$  operações

Se minha ALU tem **4 bits** =  $2^4 = 16$  operações

# Unidade Lógica-Aritmética (ALU)

## Projeto de uma ALU



- Usar componentes em separado para cada operação, e multiplexador para selecionar o resultado da operação para saída.
- Exemplo de um projeto de ALU com a sua tabela de funcionamento.

### Seleção

x	y	z	Operação
0	0	0	$S = A + B$
0	0	1	$S = A - B$
0	1	0	$S = A + 1$
0	1	1	$S = A$
1	0	0	$S = A \text{ and } B$
1	0	1	$S = A \text{ or } B$
1	1	0	$S = A \text{ xor } B$
1	1	1	$S = A'$

Bit-Wise (bit-a-bit)

### Exemplo de operações Bit-Wise (bit-a-bit)

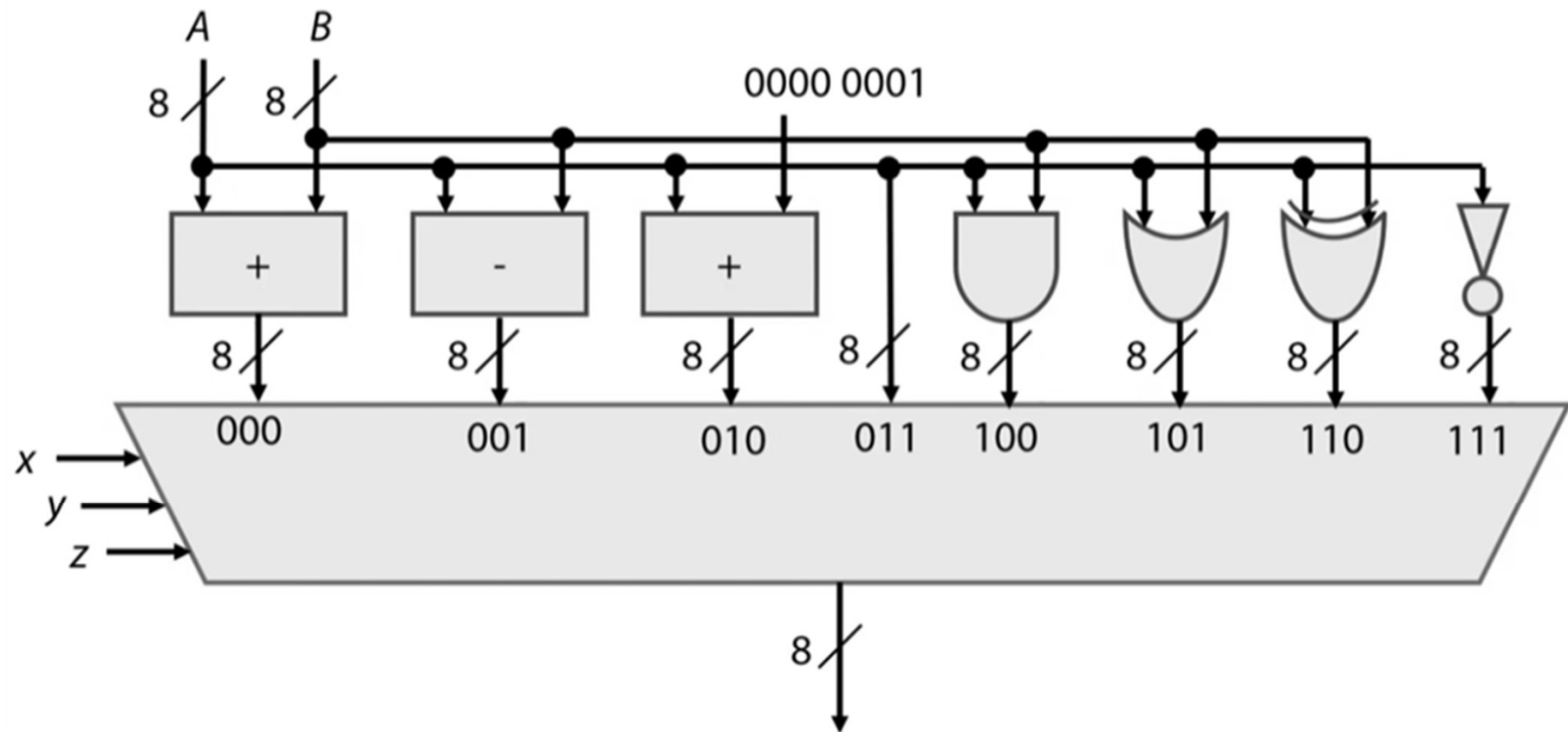
A	1	0	1	0
B	1	1	1	0

AND	1	0	1	0
OR	1	1	1	0
XOR	0	1	0	0



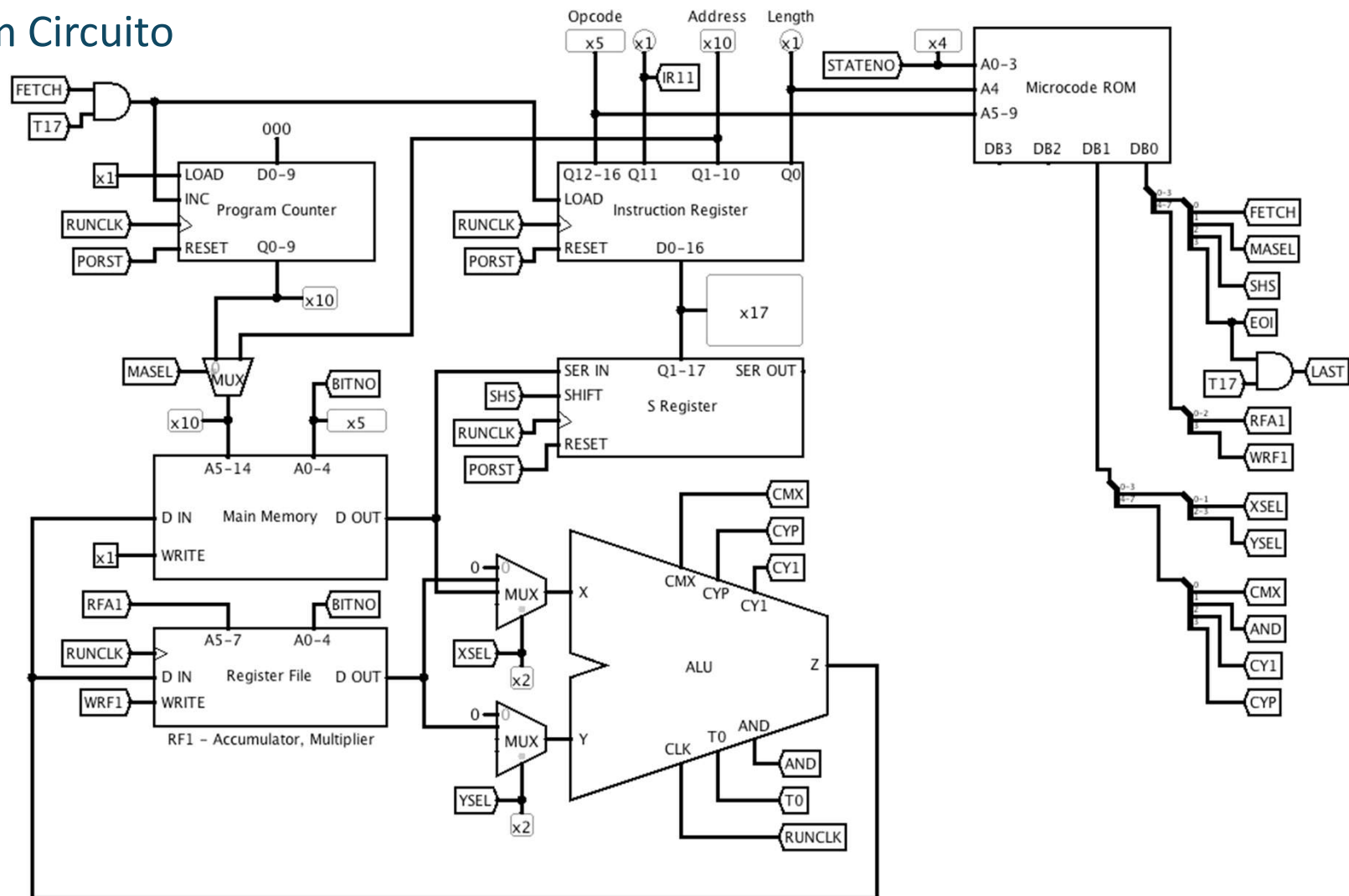
# Unidade Lógica-Aritmética (ALU)

## Projeto de uma ALU



# Unidade Lógica-Aritmética (ALU)

## Uma ALU em um Circuito



# Circuitos Lógicos

## Lista de Exercícios 6

### **Exercícios Multiplexadores:**

- 1) Projete o multiplexador 32x1 utilizando apenas MUX 8x1
- 2) Projete o multiplexador 8x1 utilizando apenas MUX 4X1

### **Exercícios Demultiplexadores:**

- 1) Projete o demultiplexador 1x32 utilizando apenas DEMUX 1x8
- 2) Projete o demultiplexador 1x8 utilizando apenas DEMUX 1x2

**ATÉ A  
PRÓXIMA  
AULA!**



## Bibliografia



TOCCI, R.; WIDMER, N.; MOSS, G. Sistemas Digitais – Princípios e Aplicações. [S.I.]: Pearson Education Limited, 2011.

FEDELI, Ricardo Daniel. Introdução à ciência da computação / Ricardo Daniel Fedeli, Erico Giulio Franco Polloni, Fernando Eduardo Peres. – 2. ed. – São Paulo: Cengage Learning, 2011.

TANENBAUM, Andrew S.. Organização Estruturada de Computadores. 6º Edição. São Paulo, Pearson Prentice Hall, 2013.



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).