#### Universidade Federal da Fronteira Sul Curso de Ciência da Computação **UFFS** Campus Chapecó

# Circuitos Combinacionais

Prof. Luciano L. Caimi lcaimi@uffs.edu.br

# **Circuitos Digitais**

- Circuitos lógicos digitais são divididos em duas grandes áreas:
- Circuitos Combinacionais: em que as saídas do circuito dependem exclusivamente do valor presente na entrada. Como nos multiplexadores, somadores, codificadores, etc...
- Circuitos Sequenciais: onde as saídas do circuito dependem dos valores presentes nas entradas e do estado anterior em que o circuito se encontra. Como exemplo temos os registradores, os contadores, as máquinas de estado, dentre outros.

- Tipos de Circuitos Combinacionais

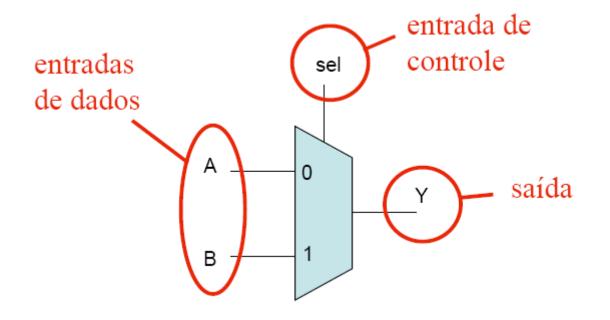
  Segundo a sua aplicação podem ser classificados em:
- Circuitos de Interconexão: seletores (conhecidos como multiplexadores), codificadores e decodificadores
- Circuitos Aritméticos: somadores, subtratores, somadores/subtratores, multiplicadores, deslocadores, comparadores e ULAS (circuitos que combinam mais de duas operações aritméticas e/ou lógicas)



## Multiplexadores (ou seletores)

#### Multiplexador 2x1:

"Sua função é selecionar uma dentre as duas entradas de dados, fazendo a entrada selecionada aparecer na saída"



Sel	A	В	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

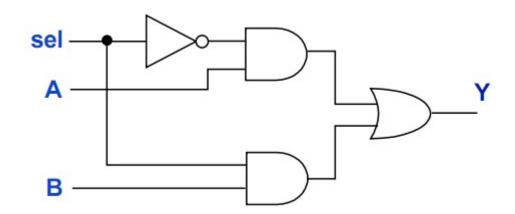


# Multiplexador 2x1 (MUX 2x1)

Sel	A	В	Y
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

	7	<u>_</u>	<b>^</b>	•
Sel	0	0	1	1
Sel	0	1	1	0
	В	ВЕ		В

$$Y = \overline{Sel} \cdot A + Sel \cdot B$$

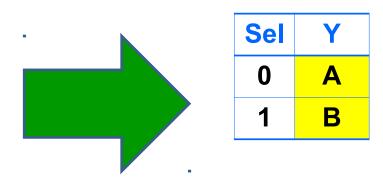




## Multiplexador 2x1 (MUX 2x1)

#### Outra forma de ver a tabela

Sel	Α	В	Υ
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

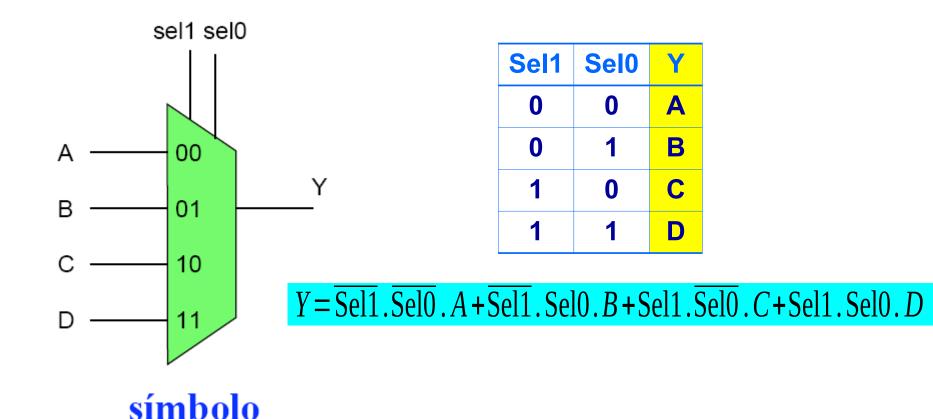


$$Y = \overline{Sel} \cdot A + Sel \cdot B$$



#### Multiplexador 4x1 (MUX 4x1)

"Sua função é selecionar uma dentre as quatro entradas de dados, fazendo a entrada selecionada aparecer na saída"





#### Implementação de Funções

Um multiplexador pode ser utilizado na implementação de uma função combinacional diretamente a partir da tabelaverdade.

Os mintermos de uma função são gerados por um multiplexador através das entradas de seleção, restando apenas ligar as entradas de dados a 0 ou a 1 em conformidade com a respectiva tabela-verdade.

Uma função de N variáveis pode ser implementada com um MUX 2N:1 (N entradas de seleção e 2N entradas de dados).



## Implementação de Funções

_			1/		
Α	В	С	Y		ABC
0	0	0	0		
0	0	1	0		
0	1	0	1		000
0	1	1	1		001
1	0	0	0		1010
1	0	1	1	, -	1 011
1	1	0	0		0—————————————————————————————————————
1	1	1	1		0 110



#### Exercícios:

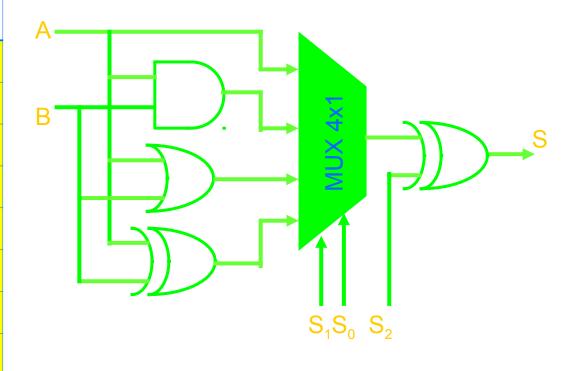
- a) Implemente um MUX 8x1 (tabela-verdade, equação, circuito)
- b) Implemente um MUX 4x1 utilizando somente MUX 2x1
- c) Implemente um MUX 8x1 utilizando somente MUX 4x1

# UFFS

#### Aplicações

MUX são utilizados para seleção de caminhos de dados em Unidades Lógicas e Aritméticas (ULAs). Observe sua utilização em uma ULA que possui a seguinte tabela verdade

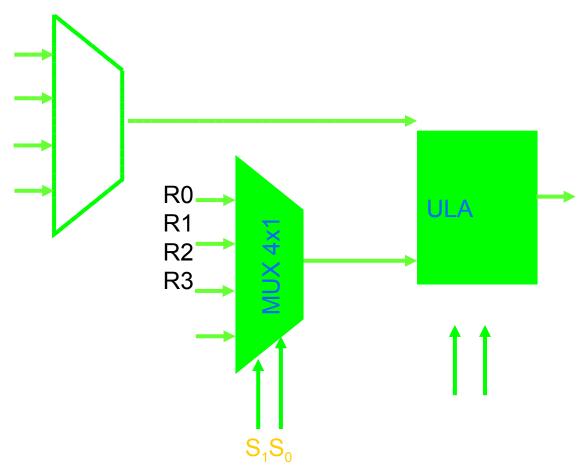
<b>S2</b>	<b>S1</b>	S0	Saída
0	0	0	Α
0	0	1	A AND B
0	1	0	A OR B
0	1	1	A XOR B
1	0	0	NOT A
1	0	1	A NAND B
1	1	0	A NOR B
1	1	1	A XNOR B







Ainda no que diz respeito a seleção de caminhos de dados



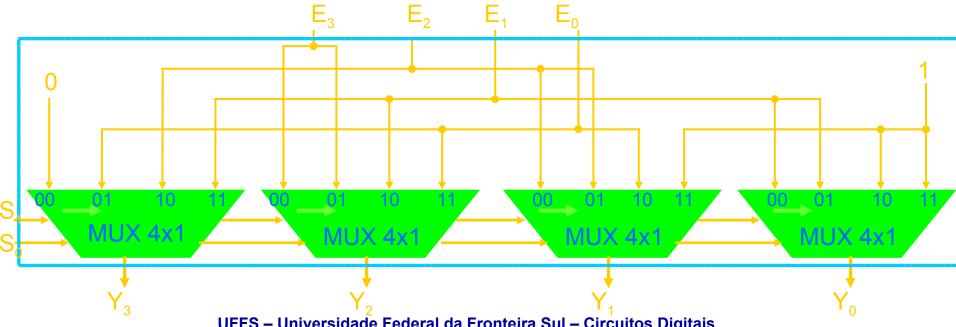
**UFFS – Universidade Federal da Fronteira Sul – Circuitos Digitais** 



#### Aplicações

O MUX pode ser usado para construção de deslocadores (shifters), que recebem uma palavra binária e fornecem o valor deslocado a direita ou a esquerda, com a inclusão de "0" ou "1", ou até a rotação da palavra

<b>S1</b>	S0	Saída
0	0	$(0,E_3,E_2,E_1)$
0	1	$(E_0, E_3, E_2, E_1)$
1	0	$(E_2, E_1, E_0, 1)$
1	1	(E <sub>1</sub> , E <sub>0</sub> ,1,1)

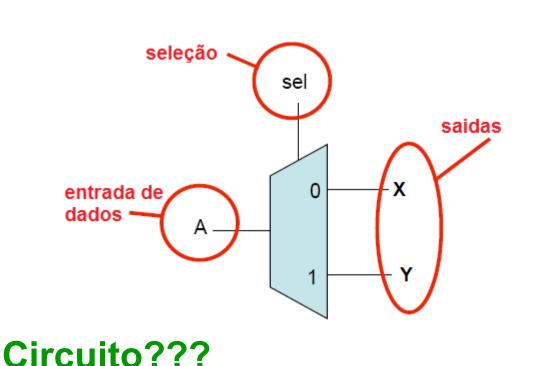




## Demultiplexadores

#### Demultiplexador 1x2:

"Realiza a função inversa dos multiplexadores, ou seja direciona a entrada para uma de 2 possíveis saídas"



Sel	A	X	Y
0	0	0	0
0	1	1	0
1	0	0	0
1	1	0	1

Sel	X	Υ
0	Α	0
1	0	Α



## Demultiplexadores

#### Demultiplexador 1x4:

"Direciona a entrada para uma de 4 possíveis saídas"

Sel0	Sel1	A	<b>S</b> 0	<b>S1</b>	<b>S2</b>	<b>S</b> 3
0	0	0	0	0	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	0
0	1	1	0	1	0	0
1	0	0	0	0	0	0
1	0	1	0	0	1	0
1	1	0	0	0	0	0
1	1	1	0	0	0	1

Sel0	Sel1	S0	<b>S</b> 1	<b>S2</b>	<b>S</b> 3
0	0	Α	0	0	0
0	1	0	Α	0	0
1	0	0	0	Α	0
1	1	0	0	0	Α

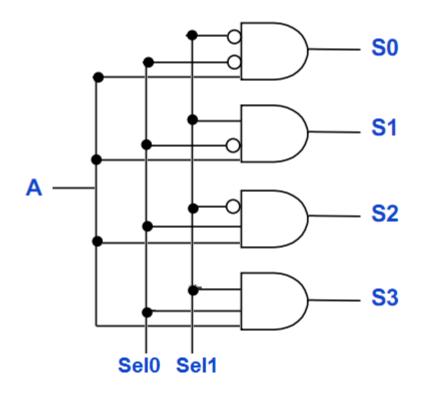
Circuito???



## Demultiplexadores

#### Demultiplexador 1x4:

Sel0	Sel1	S0	<b>S1</b>	<b>S2</b>	<b>S</b> 3
0	0	Α	0	0	0
0	1	0	Α	0	0
1	0	0	0	Α	0
1	1	0	0	0	Α





#### Exercícios:

- a) Implemente um DEMUX 1x8 (tabela- verdade, equação, circuito)
- b) Implemente um DEMUX 1x4 utilizando somente DEMUX 1x2
- c) Implemente um DEMUX 1x8 utilizando somente DEMUX 1x4



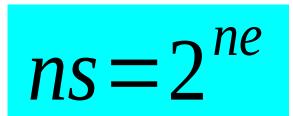
#### Decodificadores

#### Decodificador 2:4

Sua função é ativar uma e somente uma dentre as 4 saídas, de acordo com a combinação de valores das entradas

Ativar, neste caso, quer dizer <u>diferenciar</u>, <u>destacar</u>

Existe uma relação entre o número de saídas (ns) e o número de entradas (ne):





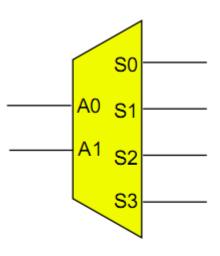
Decodificadores

Decodificador 2:4

Tabela-verdade e Símbolo

tabela-verdade

entr	adas	saídas						
A0	<b>A1</b>	S0 S1 S2 S						
0	0	1	0	0	0			
0	1	0	1	0	0			
1	0	0	0	1	0			
1	1	0	0	0	1			



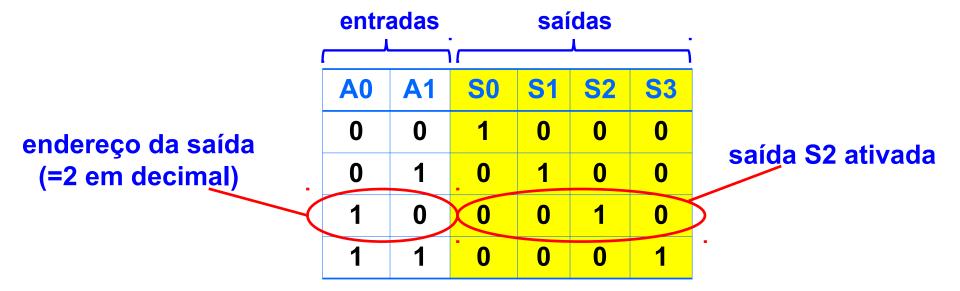
símbolo



#### Decodificadores

Decodificador 2:4

Cada combinação de entrada pode ser vista como o endereço de uma saída específica

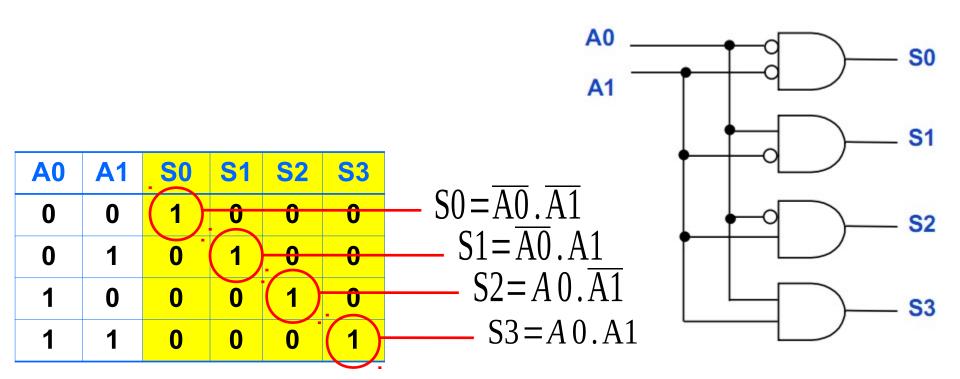




#### Decodificadores

#### Decodificador 2:4

Cada uma das 4 saídas corresponde a um mintermo diferente



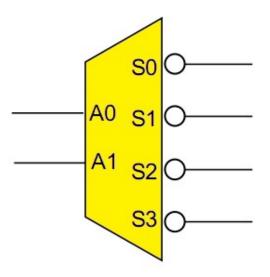


#### Decodificadores

Decodificador 2:4 - com saídas em lógica invertida (ou complementar)

ta	hal	2-1	orc	hel	ı

Como fica o circeintradas			saídas				
	<b>A0</b>	<b>A1</b>	<b>S</b> 0	<b>S1</b>	<b>S2</b>	<b>S3</b>	
	0	0	0	1	1	1	
	0	1	1	0	1	1	
	1	0	1	1	0	1	
	1	1	1	1	1	0	

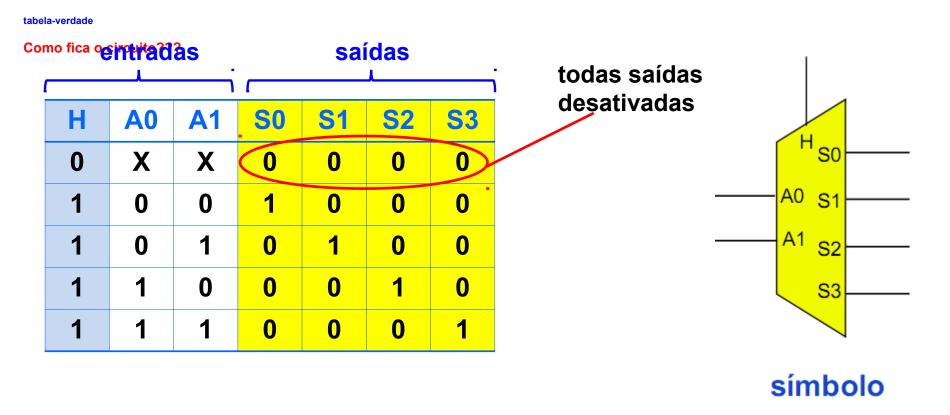


símbolo



#### Decodificadores

Decodificador 2:4 – acrescentando uma entrada de habilitação (enable)





#### Decodificadores

Decodificador 3:8, 4:16, 5:32, etc

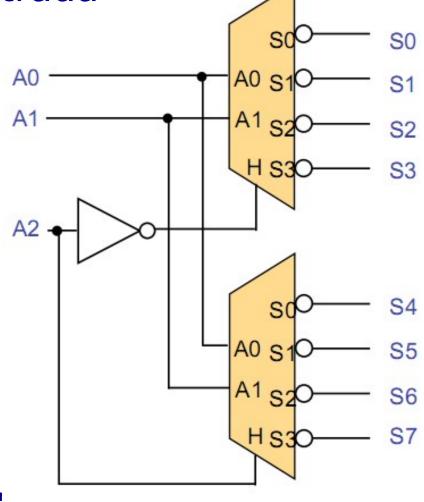
- ✓ Seguem o mesmo princípio dos decodifica-dores vistos, sempre observando a relação n:2<sup>n</sup> (número de entradas: número de saídas)
- ✓ Também se pode "montar" um decodificador a partir de decodificadores menores, que possuam entrada de habilitação



#### Decodificadores

Decodificador 3:8 - formado de 2:4, complementar, sem entrada

de habilitação





## Exemplo

#### Decodificador BCD

	В	CD		O <sub>0</sub>	<b>O</b> <sub>1</sub>	02	<b>O</b> <sub>3</sub>	<b>O</b> <sub>4</sub>	<b>O</b> <sub>5</sub>	<b>O</b> <sub>6</sub>	<b>O</b> <sub>7</sub>	<b>O</b> <sub>8</sub>	O <sub>9</sub>
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

Circuito???



#### Alguns Códigos

#### Numéricos

- Código BCD (0000; 0001; 0010 ...)
- Código Excesso de 3 (0011; 0100; 0101 ...)
- Código Gray (000; 001; 011; 010; 110 ...)
- 7 segmentos

#### **Alfanuméricos**

- Código ASCII (7 bits; estendida: 8bits)
- Código Unicode (16 bits)
- Código UTF
- Código ISSO 8859



#### Codificadores

Conceito: grosso modo, codificadores realizam a função oposta dos decodificadores

Codificadores servem para reduzir o número de bits necessários para a representação de alguma informação (facilitando sua manipulação e seu armazenamento)



#### Codificadores

Conceito: grosso modo, codificadores realizam a função oposta dos decodificadores

Codificadores servem para reduzir o número de bits necessários para a representação de alguma informação (facilitando sua manipulação e seu armazenamento)

Os principais tipos de codificadores são: binários, de prioridade.



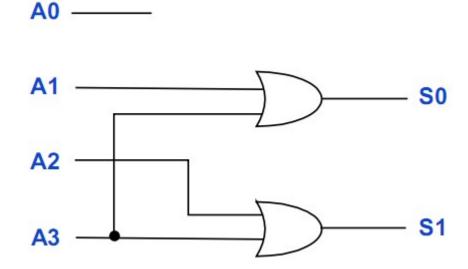
#### Codificadores

#### Codificador binário 4:2

Apenas as situações de entrada contendo somente uma posição valendo 1 são consideradas

As demais situações são tratadas como don't cares (usar Karnaugh)

	entra	saídas			
<b>A3</b>	<b>A2</b>	<b>S1</b>	<b>S0</b>		
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1





#### Exercícios

- Implementar um codificador BCD;
- Implementar um codificador Gray;
- Implementar um decodificador Gray;
- Implementar um decodificador Gray/7 segmentos;