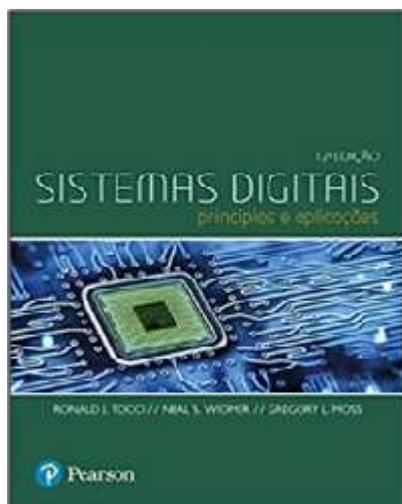


IMPLEMENTAÇÃO DE CIRCUITOS LÓGICOS



Nossos **objetivos** nesta aula são:

- Conhecer as aplicações de circuitos combinacionais Multiplexadores e Demultiplexadores.
- Conhecer um gerador e verificador de paridade.
- Praticar a implementação de circuitos combinacionais Multiplexadores, Demultiplexadores e Geradores e Verificadores de paridade.



Para esta aula, utiliza-se como referência ao item 9.6 **Multiplexadores (Seletores de dados)**, 9.7 **Aplicações Multiplexadores**, 9.8 **Demultiplexadores (Distribuidores de Dados)** do Capítulo 9 (Circuitos Lógicos MSI) e **4.7 Gerador e Verificador de Paridade** do Capítulo 4 Circuitos Lógicos Combinacionais do livro-texto:

TOCCI, Ronald J.; WIDMER, Neal S.; MOSS, Gregory L. **Sistemas Digitais: princípios e aplicações**. 12ª Ed. Editora Pearson, 2019.

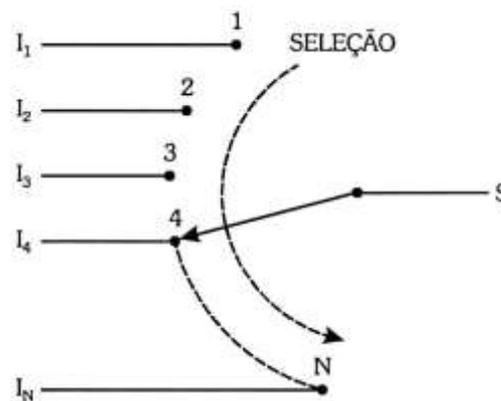
Não deixem de ler a indicação depois desta aula!

CIRCUITO COMBINACIONAL

- A saída depende única e exclusivamente das combinações entre as variáveis de entrada.
- Determina-se a tabela-verdade a partir da especificação do problema.
- A construção de circuitos combinacionais deriva de suas expressões características.
- As expressões são obtidas das tabelas-verdade que representam situações reais.
- Favorece a compreensão do funcionamento de circuitos codificadores, aritméticos, multiplexadores, demultiplexadores, geradores e verificadores de paridade etc.

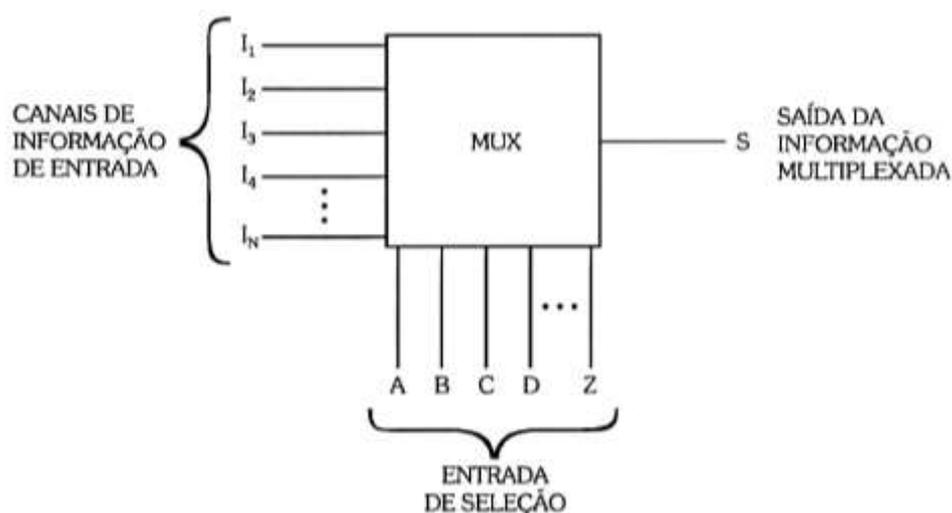
MULTIPLEXADOR

- Multiplexadores (Mux) são circuitos com uma **única saída** e **várias entradas**.
- Realiza a **seleção de uma entrada para a saída** com base em uma **palavra binária de controle**.
- Assim, permite o **envio de informações de uma de suas várias entradas para uma única saída**.
- A capacidade de multiplexação pode ser ampliada pela adição de mais entradas.
- Um circuito elementar que efetua uma multiplexação é uma chave seletora de 1 pólo e N posições:



- Exemplo: se quisermos ligar a informação **I1 na saída**, basta selecionarmos a **posição 1 da chave seletora**. Se quisermos conectar à saída a **informação I2**, é necessário **selecionar a posição 2** e assim, sucessivamente.
- O circuito Multiplex é utilizado para **enviarmos as informações contidas em vários canais (fios), a um só canal (fio)**.

BLOCO LÓGICO MULTIPLEXADOR



- A entrada de seleção tem como finalidade escolher qual das informações de entrada, ou qual dos canais de informações deve ser ligado à saída.

CIRCUITO LÓGICO - MULTIPLEXADOR 2x1

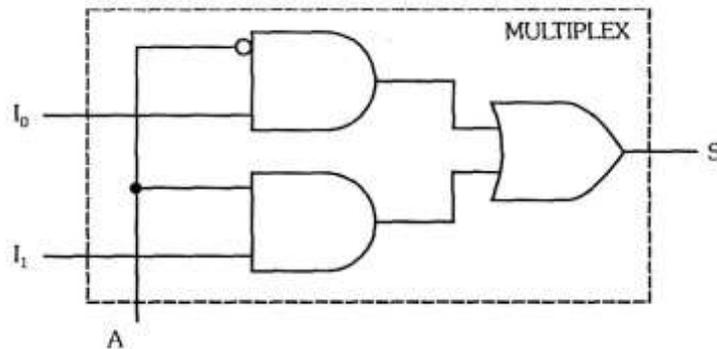


TABELA VERDADE – MULTIPLEXADOR 2x1

SELETOR	ENTRADAS		SAÍDA
A	I_0	I_1	S
0	I_0	1	$\bar{A}I_0$
1	1	I_1	AI_1

TABELA VERDADE SIMPLIFICADA

A	S
0	I_0
1	I_1

EXPRESSÃO BOOLEANA

$$S = \bar{A}I_0 + AI_1$$

- No caso do multiplex básico para **2 informações de entradas** I_0 e I_1 , temos **uma variável de seleção (A)**.
- Quando A for igual a 0, teremos na saída a mesma informação que a entrada I_0 ; se I_0 for igual a 0, S será igual a 0 e se I_0 for igual a 1, S será igual a 1.
- Neste caso, a informação I_1 será bloqueada pela porta E referente a I_1 , pois o outro terminal desta estará ligado em A que valerá 0.

- Quando A for igual a 1, I_0 será bloqueado e, analogamente, a informação I_1 aparecerá na saída.

TABELA VERDADE - MULTIPLEXADOR 4x1

SELETORES		ENTRADAS				SAÍDA
A	B	I_0	I_1	I_2	I_3	S
0	0	I_0	1	1	1	$\bar{A}\bar{B}I_0$
0	1	1	I_1	1	1	$\bar{A}BI_1$
1	0	1	1	I_2	1	$A\bar{B}I_2$
1	1	1	1	1	I_3	ABI_3

$$S = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$$

CIRCUITO LÓGICO - MULTIPLEXADOR 4x1

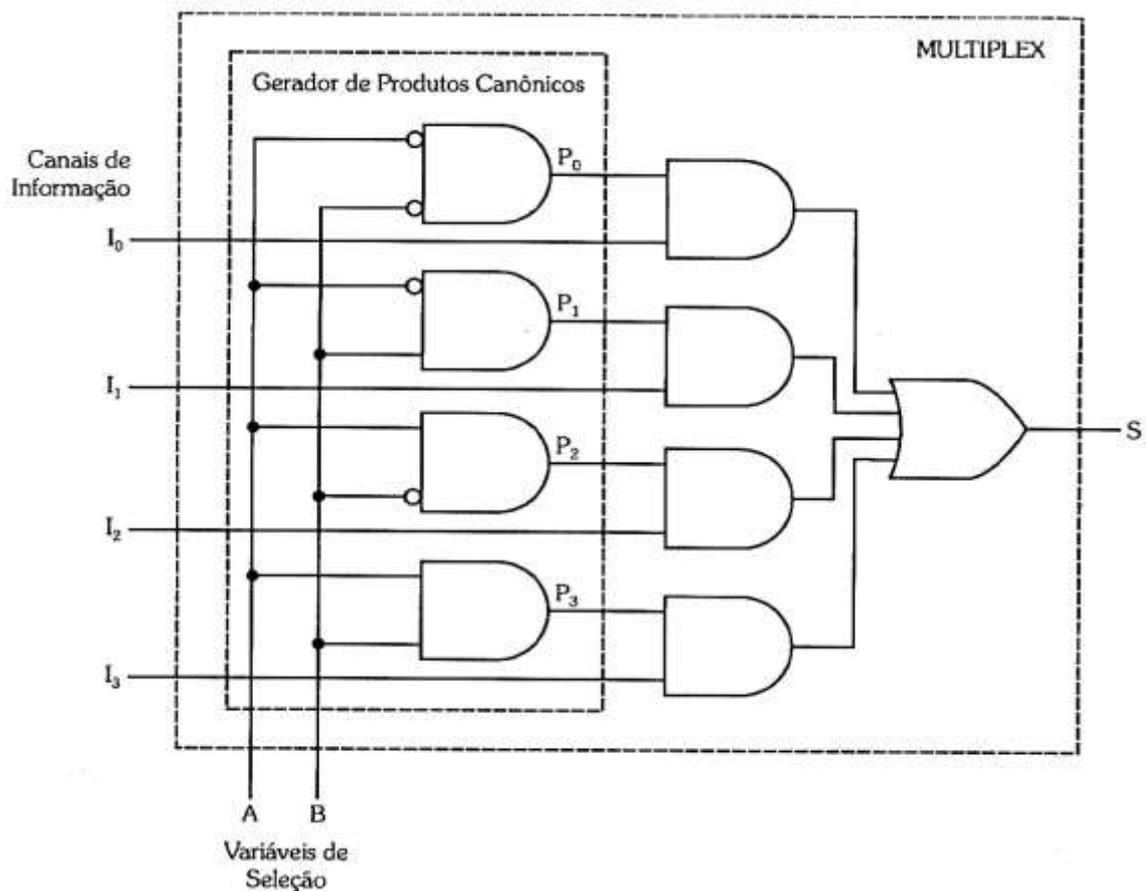
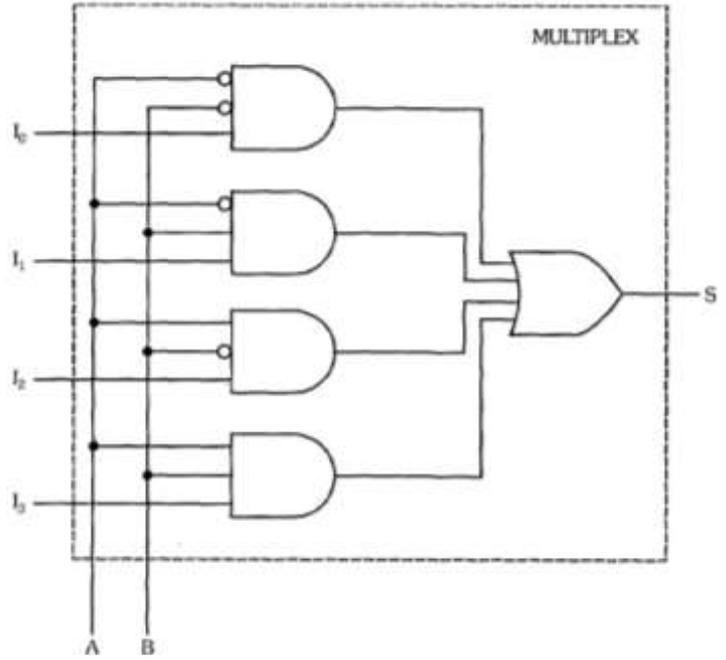
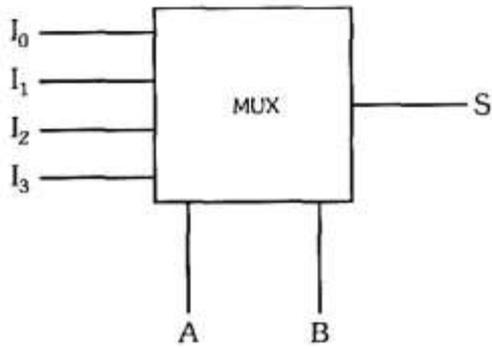


TABELA VERDADE SIMPLIFICADA

Variáveis de Seleção		Saída
A	B	S
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

CIRCUITO LÓGICO – MULTIPLEX 4x1

BLOCO LÓGICO – MULTIPLEX 4x1

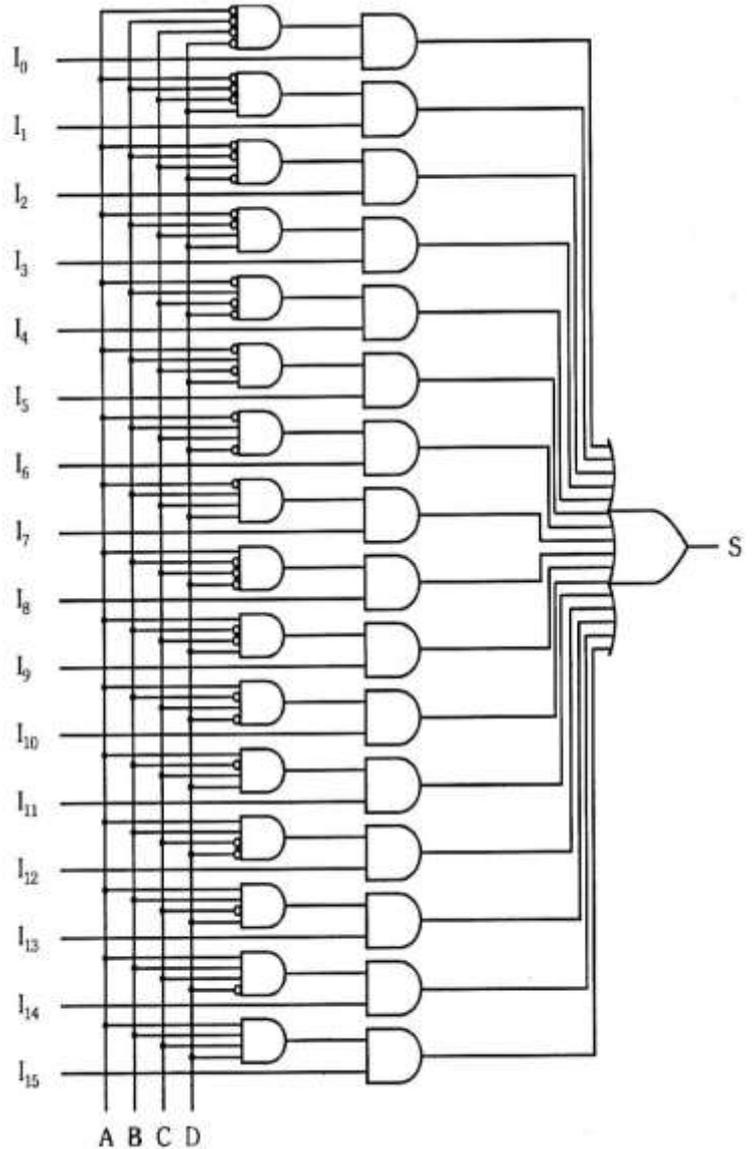


MULTIPLEX 16x1 (16 CANAIS)

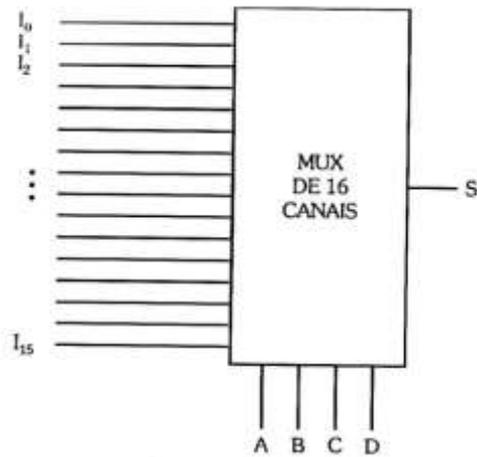
TABELA VERDADE

Variáveis de Seleção				Informações
A	B	C	D	S
0	0	0	0	I_0
0	0	0	1	I_1
0	0	1	0	I_2
0	0	1	1	I_3
0	1	0	0	I_4
0	1	0	1	I_5
0	1	1	0	I_6
0	1	1	1	I_7
1	0	0	0	I_8
1	0	0	1	I_9
1	0	1	0	I_{10}
1	0	1	1	I_{11}
1	1	0	0	I_{12}
1	1	0	1	I_{13}
1	1	1	0	I_{14}
1	1	1	1	I_{15}

CIRCUITO LÓGICO – MULTIPLEX 16x1



BLOCO LÓGICO – MULTIPLEX 16x1



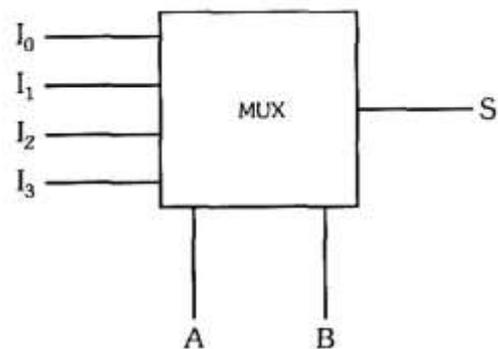
MULTIPLEXADOR COMO GERADOR DE FUNÇÕES LÓGICAS

- Além de selecionar sinais, o multiplexador pode ser usado para a implementação de funções lógicas.
- Exemplo:

EXPRESSÃO LÓGICA

$$S = \bar{A}\bar{B}I_0 + \bar{A}BI_1 + A\bar{B}I_2 + ABI_3$$

BLOCO LÓGICO – MUX 4x1

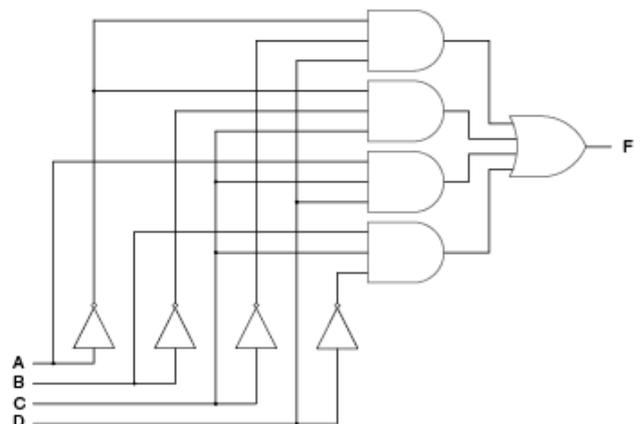


Supondo:

EXPRESSÃO LÓGICA

$$F = \bar{A}\bar{C}D + \bar{A}BC + ACD + BC\bar{D}$$

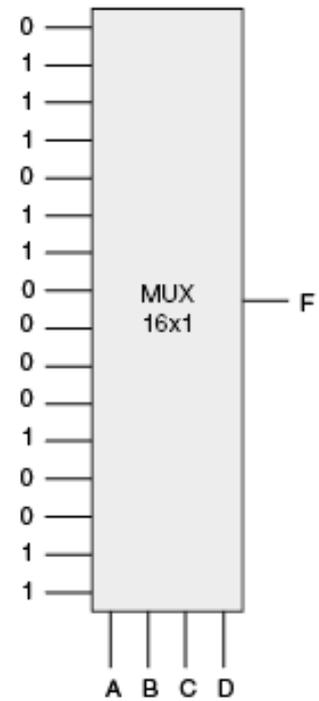
CIRCUITO LÓGICO



REDUÇÃO - SOLUÇÃO ATRAVÉS DE MUX 16x1

$$F = \bar{A} \bar{C} D + \bar{A} \bar{B} C + A C D + B C \bar{D}$$

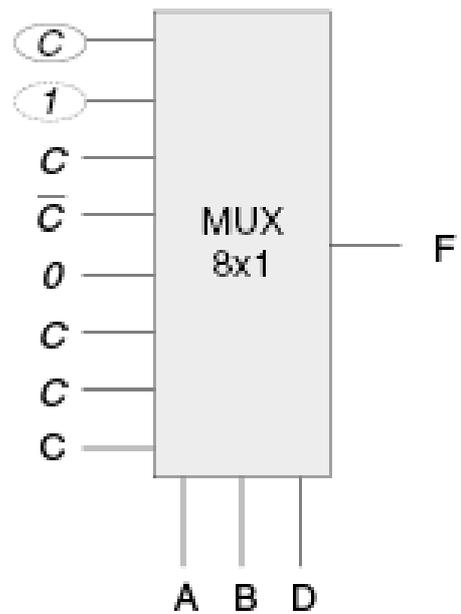
		AB			
	CD	00	01	11	10
00	0	0	0	0	0
01	1	1	0	0	0
11	1	0	1	1	1
10	1	1	1	0	0



REDUÇÃO - SOLUÇÃO ATRAVÉS DE MUX 8x1

$$F = \bar{A} \bar{C} D + \bar{A} \bar{B} C + A C D + B C \bar{D}$$

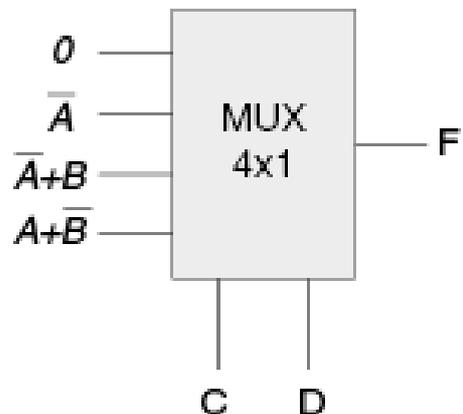
		AB			
	CD	00	01	11	10
00	0	0	0	0	0
01	1	1	0	0	0
11	1	0	1	1	1
10	1	1	1	0	0



REDUÇÃO - SOLUÇÃO ATRAVÉS DE MUX 4x1

$$F = \bar{A} \bar{C} D + \bar{A} \bar{B} C + A C D + B C \bar{D}$$

		AB			
	CD	00	01	11	10
00	0	0	0	0	0
01	1	1	0	0	0
11	1	0	1	1	1
10	1	1	1	0	0



EXERCÍCIO COM DISCUSSÃO EM DUPLAS

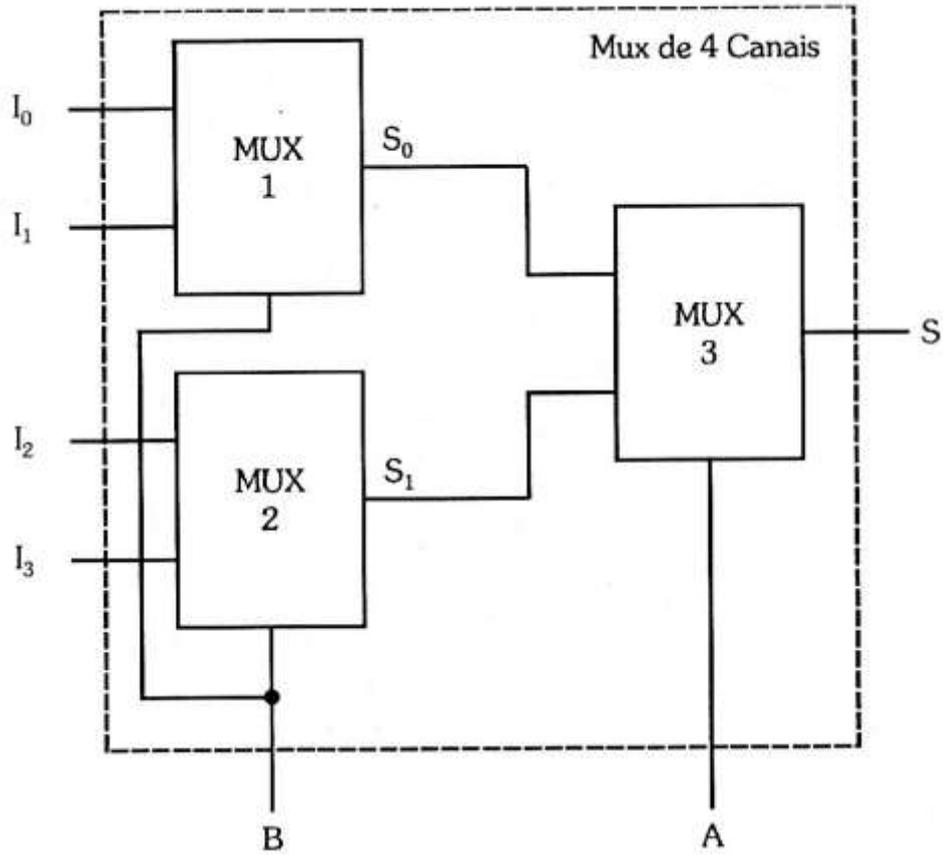
1. Utilize MUX 4x1 para implementar a seguinte expressão lógica:

$$F = \bar{A} \bar{B} \bar{C} \bar{D} + \bar{A} B \bar{C} D + \bar{A} B C \bar{D} + A \bar{B} \bar{C} \bar{D} + ABCD$$

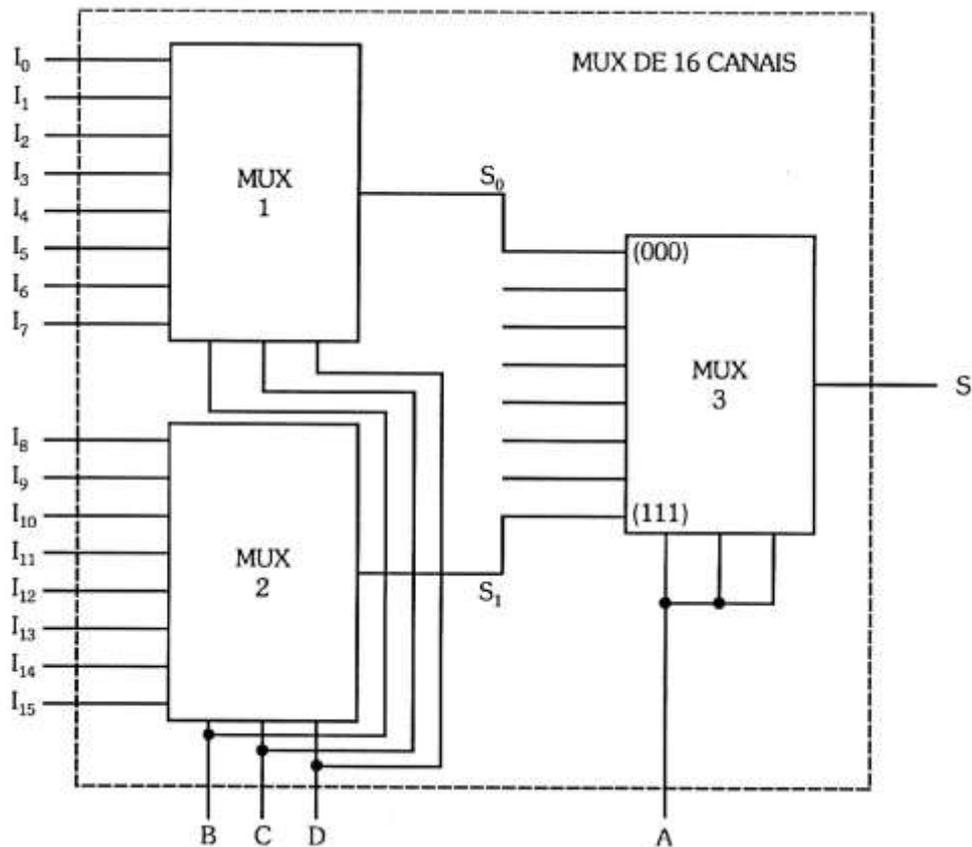
AMPLIAÇÃO DE CAPACIDADE DE UM SISTEMA MULTIPLEX

- A partir de circuitos multiplex de baixa capacidade é possível formar outros para um maior número de informações de entrada.
- Exemplo: montar um multiplex de 4 canais de informação a partir de outros de apenas 2 canais de informação.

MUX 4x1 A PARTIR DE MUX 2X1

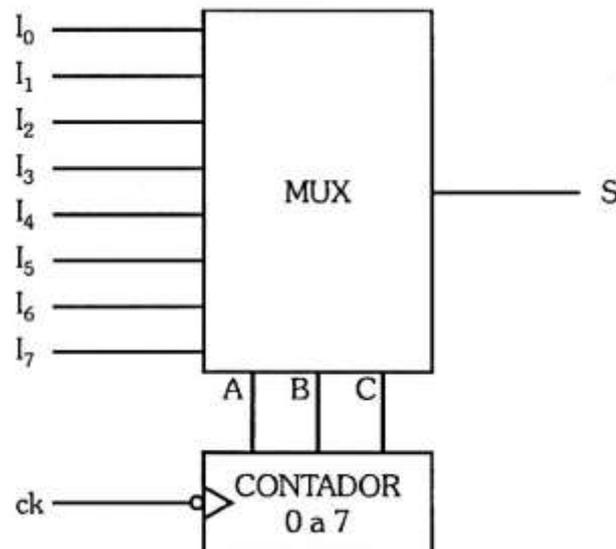


MULTIPLEX DE 16x1 (16 CANAIS) A PARTIR DE BLOCOS DE 8x1 (8 CANAIS)



ENDEREÇAMENTO SEQUENCIAL EM UM SISTEMA MULTIPLEX

- Podemos utilizar um multiplex que apresente sequencialmente na saída os dados correspondentes aos canais de informação. Para isso, é necessário conectar às entradas de seleção um circuito contador que gere a sequência de contagem desejada.



- Uma das utilidades deste sistema é a **conversão** de uma **informação paralela** em uma **informação série**, pois se o contador gerar a sequência binária, teremos sequencialmente na saída, as informações I_0 , I_1 , I_2 até I_{N-1} .

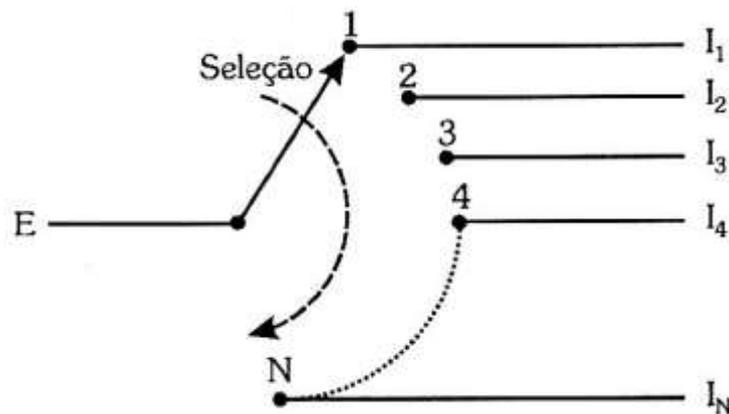
EXERCÍCIO COM DISCUSSÃO EM DUPLAS

- Crie a partir de MUX 8x1 uma configuração que reproduza a tabela verdade a seguir.

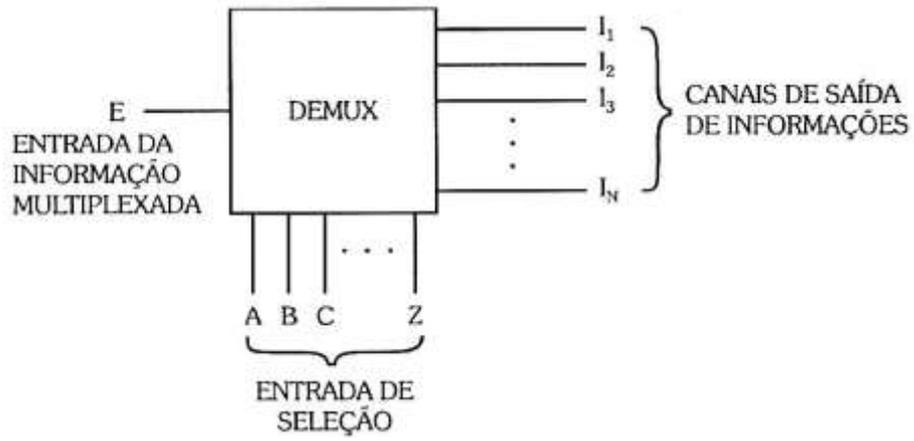
A	B	C	S_1	S_2
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

DEMÚLTIPLEXADOR

- Demultiplexadores (Demux) são circuitos que possuem **uma entrada e várias saídas**.
- Realiza a **seleção de com base da entrada em uma das saídas em função de uma palavra binária de controle**.
- Permite o **recebimento de informações por um único canal (meio) e o direciona para uma de suas saídas**.
- A capacidade de demultiplexação pode ser ampliada em mais saídas.
- Realiza o processo inverso dos multiplexadores.
- Entende-se por **Demultiplex** como sendo o **bloco que efetua a função inversa do Multiplex**, ou seja, a de enviar informações contidas em um canal a vários canais de saída.



BLOCO DEMÚLTIPLEXADOR



DEMÚLTIPLEXADOR 1x2

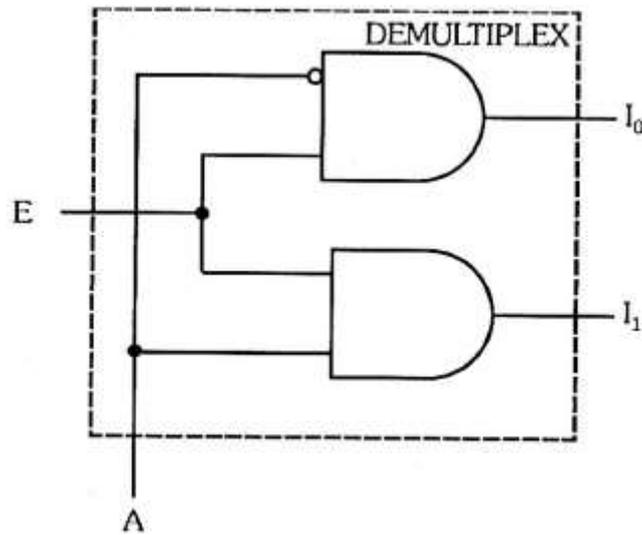


TABELA VERDADE - DEMÚLTIPLEXADOR 1x2

SELETOR	ENTRADAS	SAÍDAS	
A	E	I_0	I_1
0	E	$\bar{A}E$	AE
1	E	$\bar{A}E$	AE

TABELA VERDADE SIMPLIFICADA

Variável de Seleção	Canais de Informação	
	I_0	I_1
0	E	0
1	0	E

- Podemos notar que quando $A = 0$, a informação de entrada sairá em I_0 e quando $A = 1$, a informação de entrada sairá por I_1 .
- Assim sendo, as variáveis de seleção fornecem o endereço do local (canal de saída) por onde a informação que chega na entrada deverá sair.

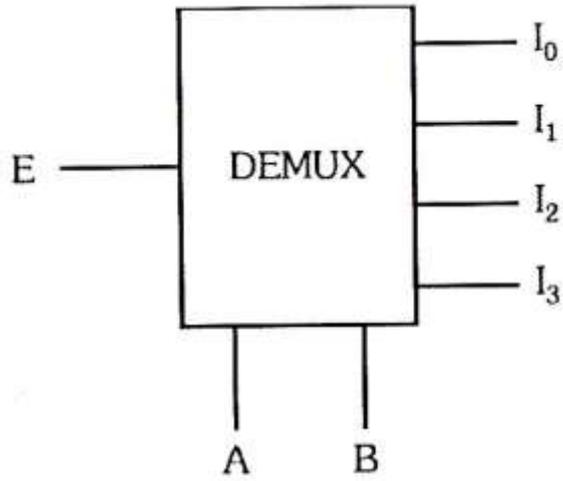
TABELA VERDADE - DEMULTIPLEXADOR 1x4

SELETORES		ENTRADA	SAIDAS			
A	B	E	I_0	I_1	I_2	I_3
0	0	E	$\bar{A}\bar{B}E$	$\bar{A}BE$	$A\bar{B}E$	ABE
0	1	E	$\bar{A}\bar{B}E$	$\bar{A}BE$	$A\bar{B}E$	ABE
1	0	E	$\bar{A}\bar{B}E$	$\bar{A}BE$	$A\bar{B}E$	ABE
1	1	E	$\bar{A}\bar{B}E$	$\bar{A}BE$	$A\bar{B}E$	ABE

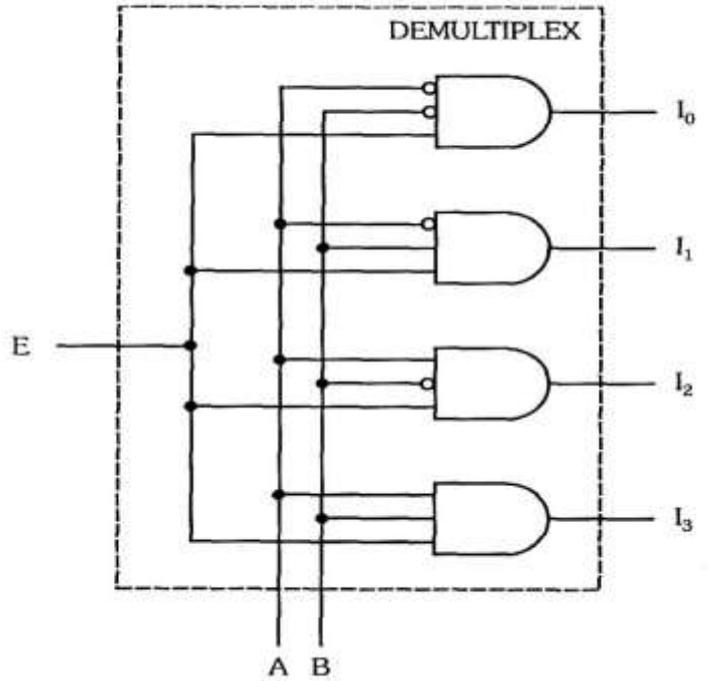
TABELA VERDADE SIMPLIFICADA

Variáveis		Canais de Saída			
A	B	I_0	I_1	I_2	I_3
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

BLOCO LÓGICO – DEMULTIPLEX 1x4



CIRCUITO LÓGICO – DEMULTIPLEX 1x4



CIRCUITO LÓGICO - DEMULTIPLEXADOR DE 1x8 (8 CANAIS)

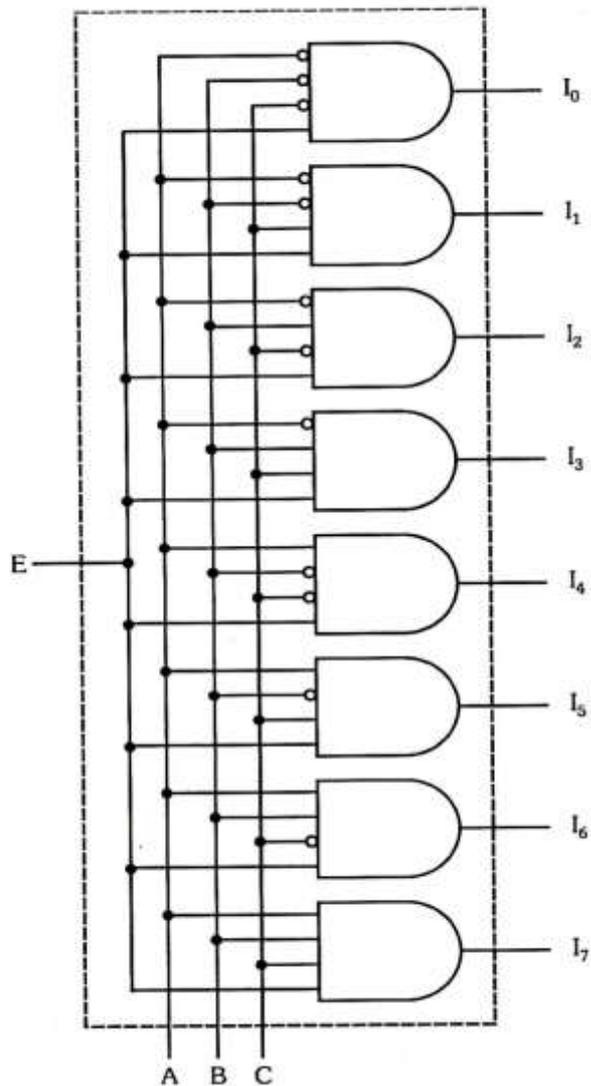


TABELA VERDADE

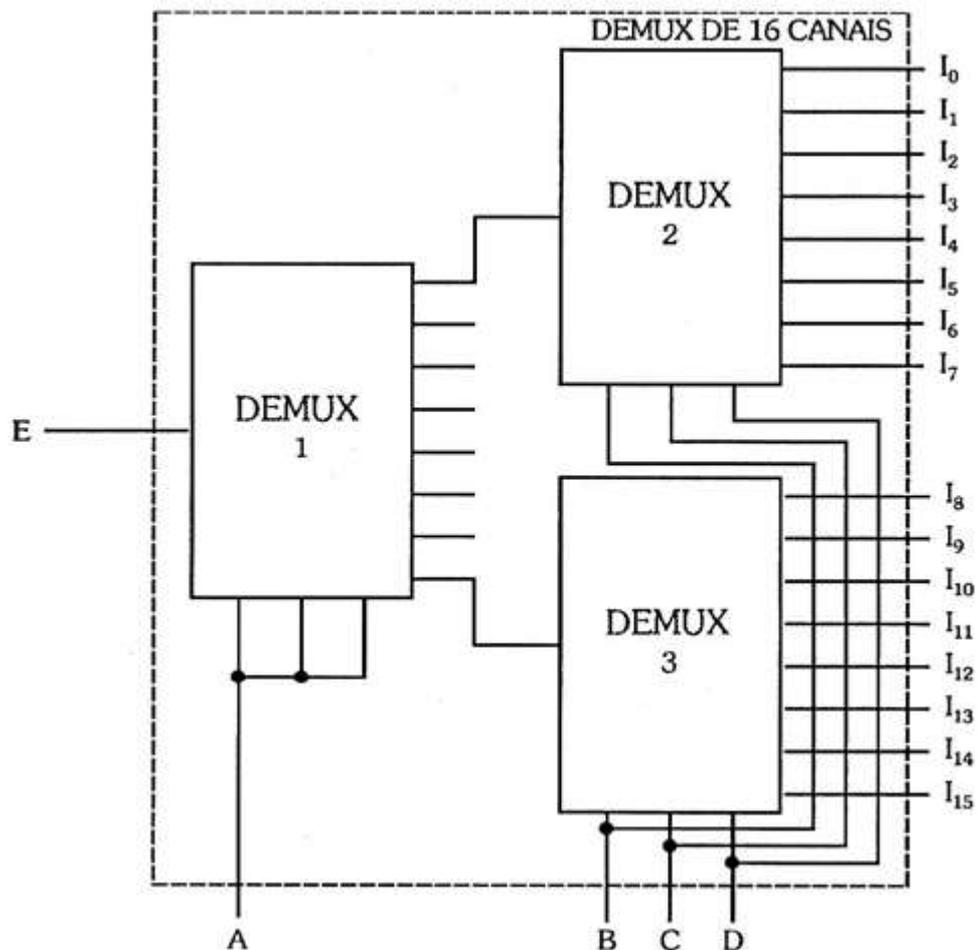
Variáveis de Seleção			Canais de Saída							
A	B	C	I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇
0	0	0	E	0	0	0	0	0	0	0
0	0	1	0	E	0	0	0	0	0	0
0	1	0	0	0	E	0	0	0	0	0
0	1	1	0	0	0	E	0	0	0	0
1	0	0	0	0	0	0	E	0	0	0
1	0	1	0	0	0	0	0	E	0	0
1	1	0	0	0	0	0	0	0	E	0
1	1	1	0	0	0	0	0	0	0	E

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

3. Crie um DEMUX 1x4 a partir de um DEMUX 1x2 que reproduza a tabela verdade a seguir.

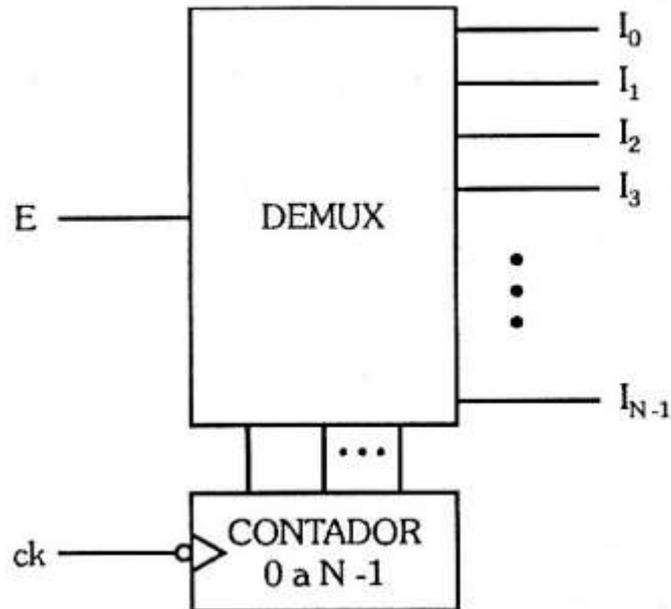
A	B	S ₀	S ₁	S ₂	S ₃
0	0	E	0	0	0
0	1	0	E	0	0
1	0	0	0	E	0
1	1	0	0	0	E

DEMÚLTIPLEXADOR DE 1x16 (16 CANAIS) A PARTIR DE DEMÚLTIPLEX 1x8 (8 CANAIS)



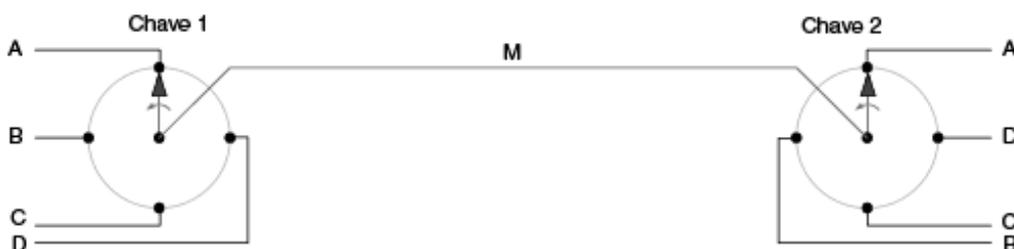
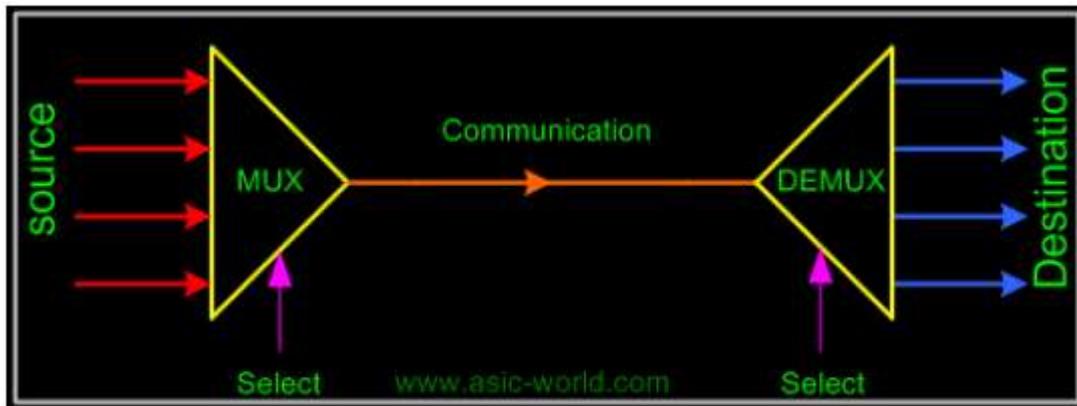
DEMÚLTIPLEX COM ENDEREÇO SEQUENCIAL

- Podemos utilizar um Demultiplex que apresente a informação de entrada saindo pelos canais de acordo com um endereçamento sequencial. Para isso, é necessário um **circuito contador ligado às entradas de seleção**.



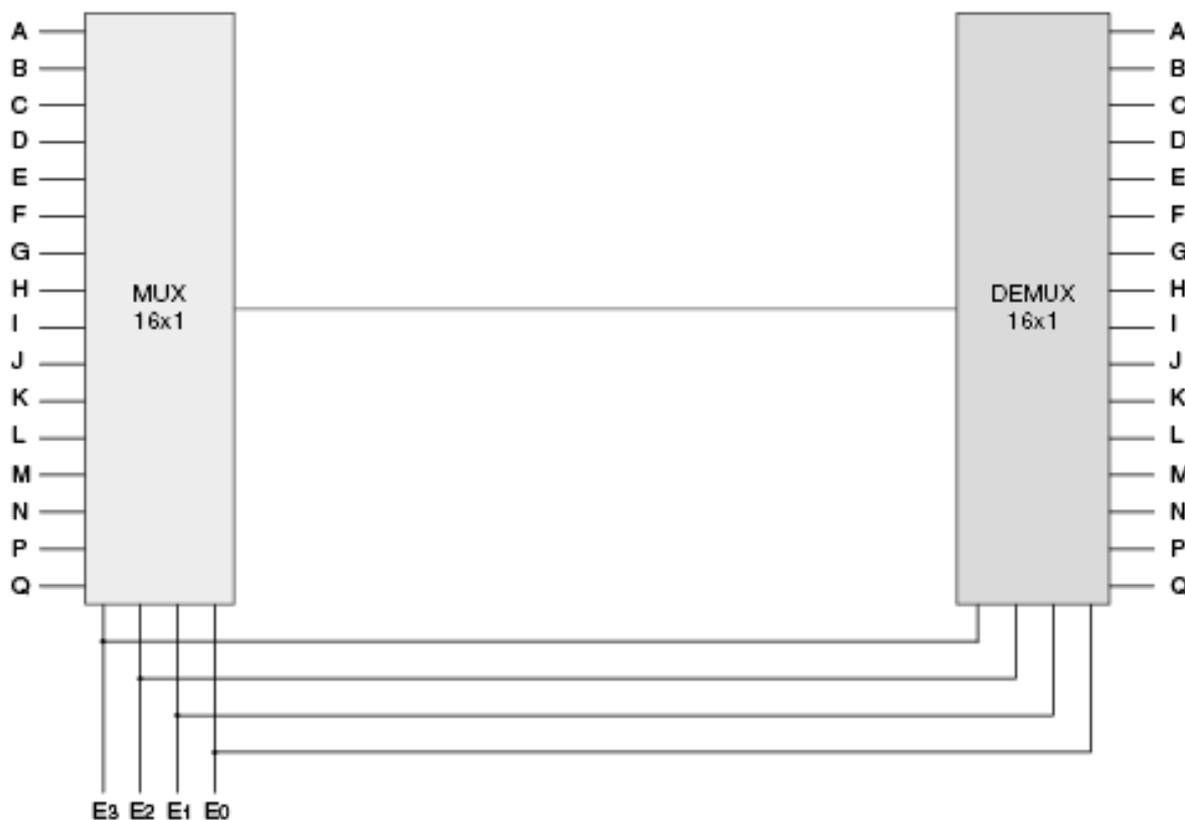
- Importante: esta configuração **não permite a conversão de informação série em paralela**, pois **não permite a saída simultânea de informações pelos canais de saída**. É necessário haver o processo de armazenamento para envio simultâneo após um tempo convenientemente dimensionado.

MUX E DEMULTIPLEX EM TRANSMISSÃO E RECEPÇÃO DE DADOS - SERIAL



- MUX e o DEMUX são muito utilizados na **transmissão e recepção de dados (transmissão digital)**.
- Dispõem-se um **único canal de comunicação para a transmissão e recepção de dados**.
- **MUX realiza a transmissão e o DEMUX a recepção de informações em intervalos de tempo diferentes por um único canal de comunicação.**
- **Sincronismo entre chaves é essencial.**

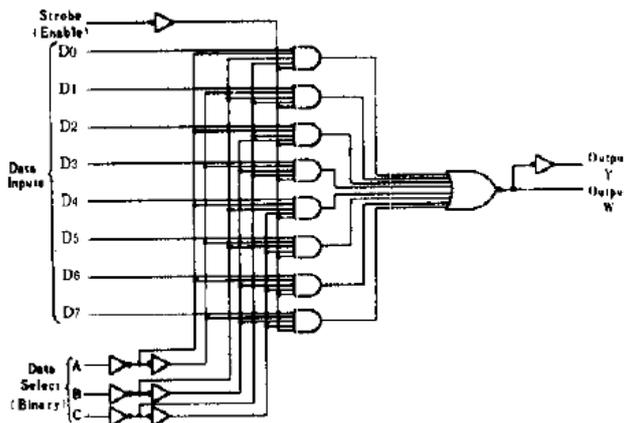
TANSMISSÃO 16 CANAIS PARALELOS POR UM CANAL SERIAL



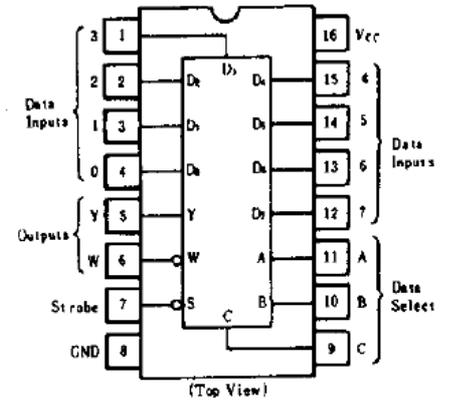
- Existem diversos CIs com o bloco funcional MUX e DEMUX, alguns exemplos são:

DATA SELECTORS MULTIPLEXERS (74LS151)

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



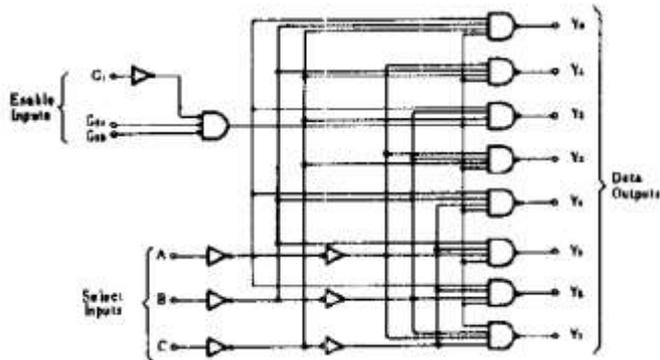
■ FUNCTION TABLE

Inputs			Outputs		
SELECT			STROBE	Y	W
C	B	A			
X	X	X	H	L	H
L	L	L	L	D ₀	\bar{D}_0
L	L	H	L	D ₁	\bar{D}_1
L	H	L	L	D ₂	\bar{D}_2
L	H	H	L	D ₃	\bar{D}_3
H	L	L	L	D ₄	\bar{D}_4
H	L	H	L	D ₅	\bar{D}_5
H	H	L	L	D ₆	\bar{D}_6
H	H	H	L	D ₇	\bar{D}_7

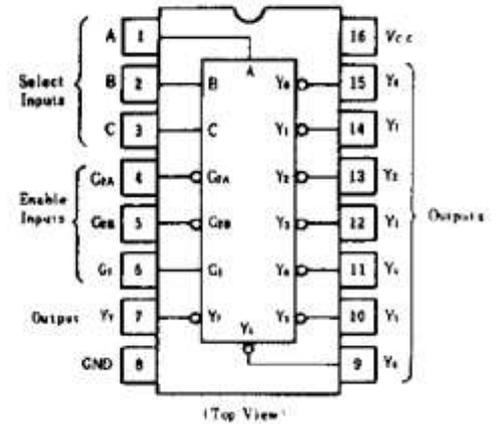
H; high level, L; low level, X; irrelevant

DATA SELECTORS DEMULTIPLEXERS (74LS138)

■ BLOCK DIAGRAM



■ PIN ARRANGEMENT



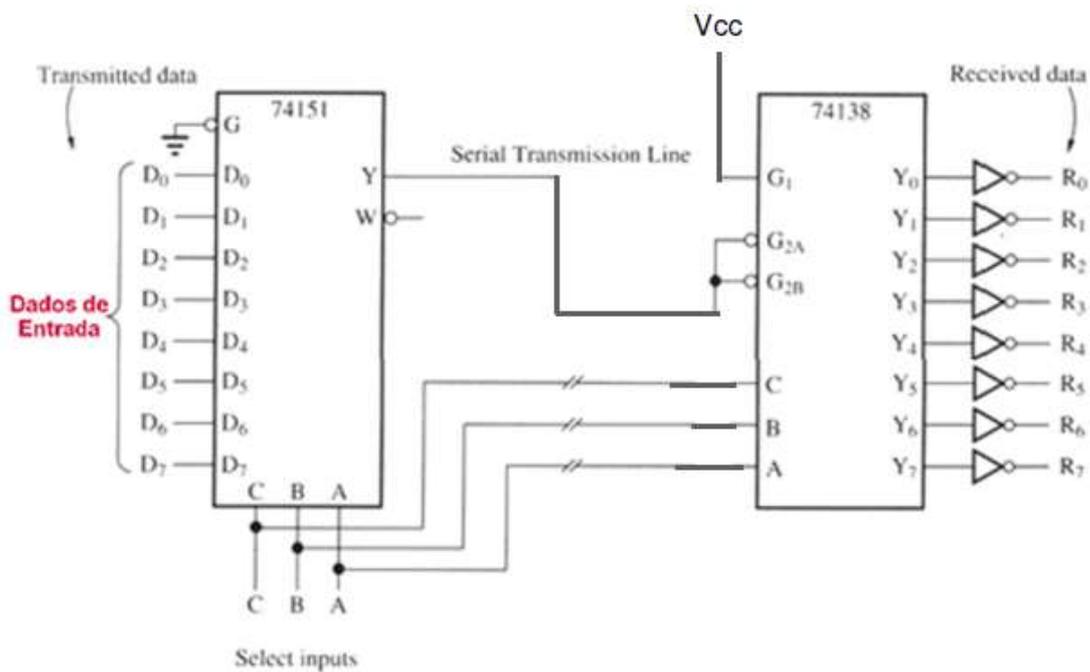
■ FUNCTION TABLE

Inputs					Outputs							
Enable		Select										
G1	G2*	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

H; high level, L; low level, X; irrelevant

*, $G_2 = G_{2A} + G_{2B}$

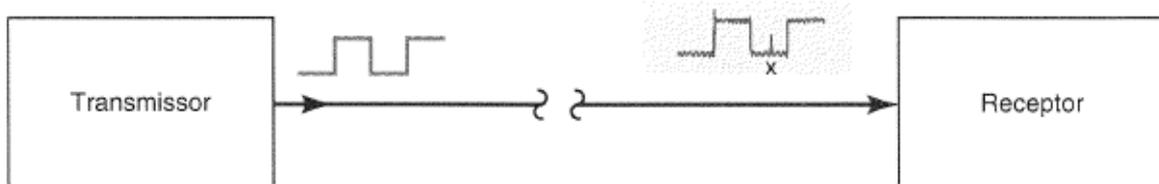
TRANSMISSÃO EM UMA LINHA DE DADOS SERIAL



BIT DE PARIDADE

- Um **bit de paridade** é um **bit extra** que é **anexado ao grupo de bits do código** que está sendo transferido de um lugar a outro.
- O **bit de paridade** pode ser **0** ou **1**, dependendo do número de 1s contido no grupo.

LINHA DE TRANSMISSÃO



- Dois métodos:
 - **Paridade par**: o valor do bit de paridade é escolhido de tal modo que o **número total de 1s no grupo de bits do código (incluindo o bit de paridade) seja um número par**.

Exemplo:

Grupo 100011 corresponde ao carácter "C" em ACSII, possui três 1s, portanto adiciona-se um bit de paridade 1 para fazermos com que o número total de 1s seja par.

O novo grupo será: 11000011.

↑
bit de paridade

Se o grupo de bits do código já conter inicialmente um número par de 1s, o bit de paridade assume valor 0.

Exemplo:

Grupo 100001 corresponde ao carácter "A" em ASCII, possui dois 1s, portanto adiciona-se um bit de paridade 0 e o novo código será:

O novo grupo será 01000001.

↑
bit de paridade

- **Paridade ímpar:** usado exatamente do mesmo modo, com exceção de que o bit de paridade é escolhido de tal maneira que o **número total de 1s (incluindo o bit de paridade)** seja um **número ímpar**.

Exemplo:

Grupo 000011, o bit de paridade deve ser 1.

Para o grupo 100011 o bit de paridade deve ser 0.

EXERCÍCIO COM DISCUSSÃO EM DUPLAS

4. Crie um gerador de paridade PAR e um verificador de paridade PAR para a transmissão e recepção em uma linha de transmissão de 4 bits de dados.

Utilize somente portas lógicas.

EXERCÍCIOS EXTRA-CLASSE

1. Construa um MUX 16x1 baseado em MUXs 4x1.
2. Construa um MUX 8x1 que envie somente números pares.
3. Construa um MUX 8x1 que envie somente números ímpares.
4. Construa um DEMUX 16x1 baseado em DEMUXs 2x1.
5. Construa um DEMUX 16x1 que recebe a sequência 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.
6. Projete um sistema de transmissão serial de dados. Na transmissão ele deve receber 16 canais, multiplexar em um canal serial para transmissão, na recepção deverá receber em um canal e demultiplexar novamente em 16 canais.
7. Crie um circuito gerador e verificador de paridade ímpar para uma linha de transmissão de 4 bits de dados.
8. Implemente as seguintes funções lógicas utilizando um MUX 16x1
 - a. $F = \bar{A} \oplus B + CB + BD + A \oplus C$
 - b. $G = \bar{A} \bar{B} D + \bar{A} \bar{B} \bar{D} + ACBD + AB \bar{D}$
 - c. $H = A \odot (B \oplus C) + BCD$
 - d. $I = (A \odot B) \oplus (A \odot C) + ACD$
9. Implemente as funções lógicas anteriores utilizando um MUX 8x1 com as seguintes linhas de controle:
 - a. ABC
 - b. BCD
 - c. ABD