

Sistemas Digitais



Operações Lógicas Mutiplexadores, demultiplexadores

Prof. Manoel Eusebio de Lima
Centro de Informática
Universidade Federal de Pernambuco

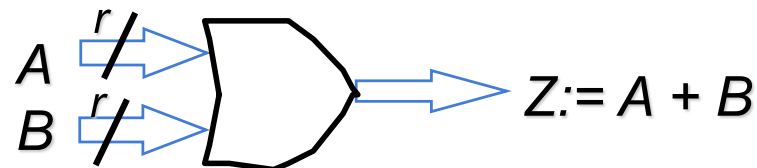
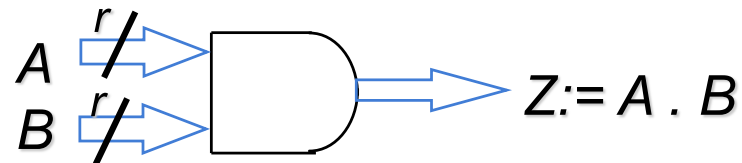
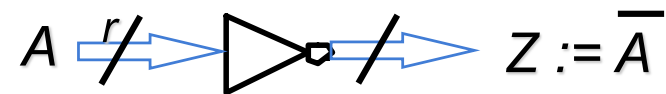
Operações com vetores

■ Possíveis formas de Operação binárias

| <u>Oper. 1</u> | <u>Oper. 2</u> | <u>Resultado</u> | <u>Comentário</u> |
|----------------|----------------|------------------|---------------------|
| – Escalar | Escalar | Escalar | Operação padrão |
| – Escalar | vetor | Escalar | Não existe |
| – Vetor | Escalar | Escalar | Não existe |
| – Vetor | Vetor | Vetor | Operação vetorial |
| – Vetor | Vetor | Escalar | Operação relacional |
| – Escalar | Vetor | Vetor | Modo Misto |
| – Vetor | Escalar | Vetor | |

Aplicações lógicas com vetores

- Operações lógicas aplicadas a vetores são simplesmente uma extensão das operações lógicas como aquelas aplicadas a escalares.
- Exemplo:



Operações Lógicas com vetores

- Operações escalares podem ser estendidas a vetores. Estas operações são importantes no nível de sistema.
- 1. Operação lógica com vetores
- 2. Operações relacionais
- 3. Operações no modo misto

Operações lógicas com vetores

| Operação | Representação | significado |
|--------------|-----------------------------|----------------------------------|
| NO | $Z := \overline{X}$ | $z_i = \overline{x_i}$ |
| AND | $Z := X \cdot Y$ | $z_i = x_i \cdot y_i$ |
| OR | $Z := X + Y$ | $z_i = x_i + y_i$ |
| OR-Exclusivo | $Z := X \oplus Y$ | $z_i = x_i \oplus y_i$ |
| Coincidência | $Z := X \odot Y$ | $z_i = x_i \odot y_i$ |
| NAND | $Z := \overline{X \cdot Y}$ | $z_i = \overline{x_i \cdot y_i}$ |
| NOR | $Z := \overline{X + Y}$ | $z_i = \overline{x_i + y_i}$ |

Operações relacionais com vetores

- Em várias tarefas de processamento é necessário verificar ou melhor comparar dois tipos de informação e tomar alguma decisão lógica. Se a decisão é verdadeira (true) ou falsa(false), em função da comparação solicitada.
- Matematicamente isto é feito através de um operador relacional. Estes operadores operam sobre vetores e produzem um resultado escalar.

$$z := X \text{ <relação> } Y$$

- Operações relacionais básicas:

< menor que

> maior que

≤ menor ou igual a

≥ maior ou igual a

= igual

≠ não igual

Aplicações - operações relacionais

- Na realização de operações relacionais deve-se tomar cuidado com o domínio dos operandos. Os operandos devem pertencer ao mesmo domínio (inteiros, ASCII, EBCDIC, ...)

- Exemplo

$x := A < B$

Para realizarmos tal operação devemos decidir qual código (domínio) será usado para representar a informação correspondente aos vetores A e B. Por exemplo A e B são números binários positivos.

Considere: $A := [a_{r-1}, a_{r-2}, a_0]$ e $B := [b_{r-1}, b_{r-2}, b_0]$

A idéia é começar a avaliar os dois vetores da direita para esquerda até encontrarmos dois bits diferentes. Se $a_i = 0$ e $b_i = 1$, então sabemos que $A < B$ e fazemos um certo $d_i = 1$. Este d_i funciona como um bit auxiliar para identificar a relação $A < B$.

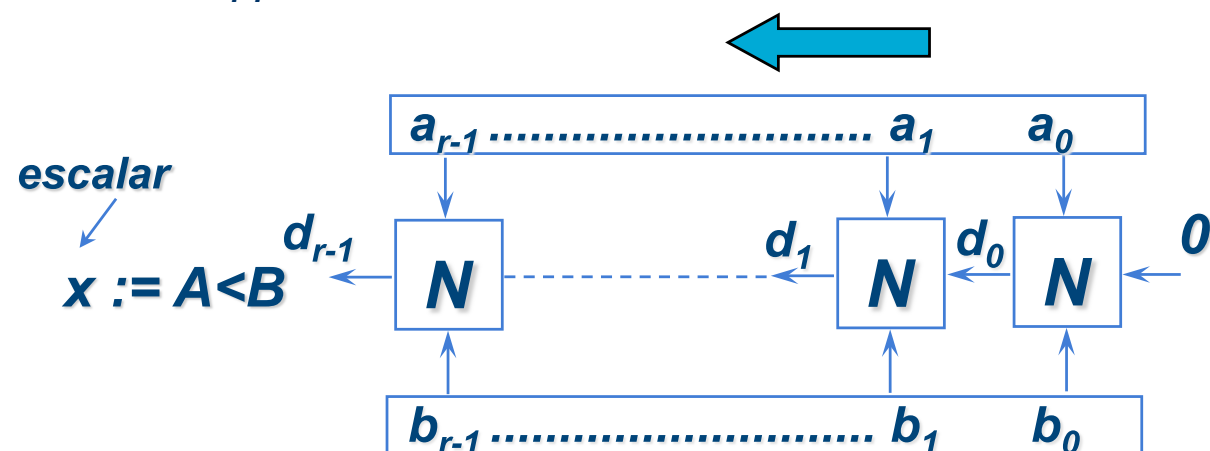
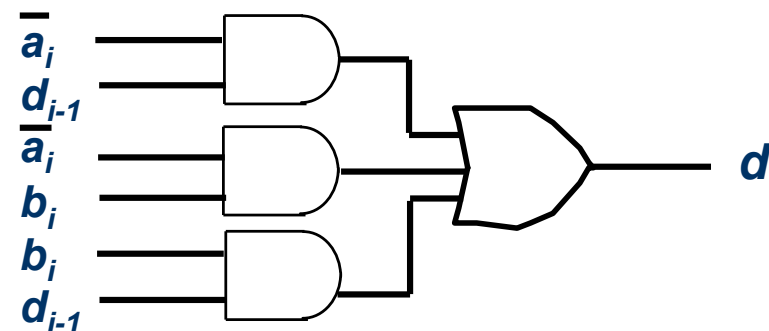
Aplicações - operações relacionais

- A tabela verdade para o sub-circuito que gera esta função pode ser dada por:

$x := A < B$

| a_i | b_i | d_{i-1} | d_i |
|-------|-------|-----------|-------|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

Onde $d_i = \bar{a}_i \cdot d_{i-1} + \bar{a}_i \cdot b_i + b_i \cdot d_{i-1}$



Operações no modo misto

- Existem operações onde nós desejamos combinar um escalar x com um vetor para formar um outro vetor.
- Para tal convencionamos que sendo x um escalar e Y e Z vetores, façamos:

<operação> no modo misto se

1. <operação> é uma das operações básicas
2. $Z := x$ <operação> Y implica que $z_i := x$ <operação> y_i , para $i = 1, 2, \dots, r$

- Algumas operações no modo misto

AND

$Z := x.Y$

$z_i := x \cdot y_i$
 $i = 1, 2, \dots, r$

$Z := \begin{cases} [0] & \text{se } x=0 \\ Y & \text{se } x=1 \end{cases}$

OR

$Z := x+Y$

$z_i := x + y_i$
 $i = 1, 2, \dots, r$

$Z := \begin{cases} Y & \text{se } x=0 \\ [1, 1, \dots] & \text{se } x=1 \end{cases}$

Aplicações - modo misto

■ Modo Misto

Modo misto é importante quando queremos selecionar um vetor de um grupo de valores.

Suponha um circuito com três entradas W , X e Y e desejamos controlar qual vetor nós devemos usar na computação. Este é o princípio da multiplexação.

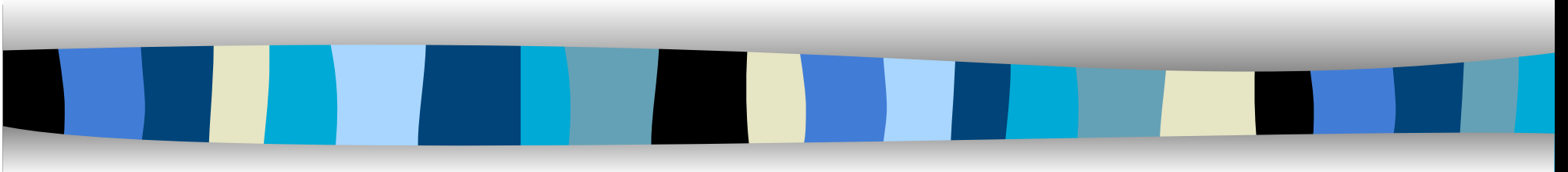
$$Z := (a_1 \cdot W) + (a_2 \cdot X) + (a_3 \cdot Y)$$

Onde:

1. se $a_1 = 1; a_2 = 0; a_3 = 0; Z := (1 \cdot W) + (0 \cdot X) + (0 \cdot Y) := W$
2. se $a_1 = 0; a_2 = 1; a_3 = 0; Z := (0 \cdot W) + (1 \cdot X) + (0 \cdot Y) := X$
3. se $a_1 = 0; a_2 = 0; a_3 = 1; Z := (0 \cdot W) + (0 \cdot X) + (1 \cdot Y) := Y$

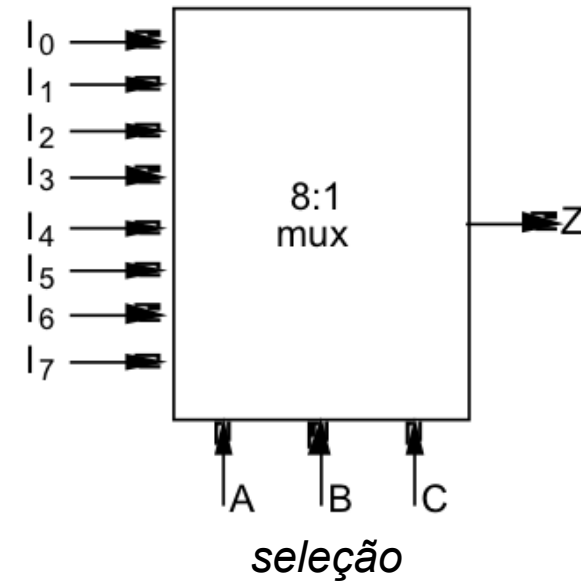
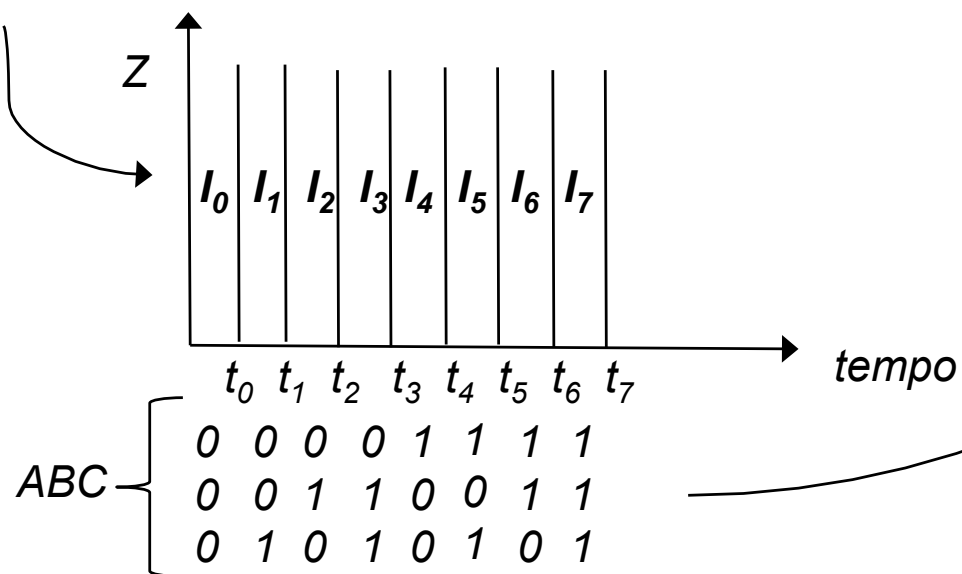
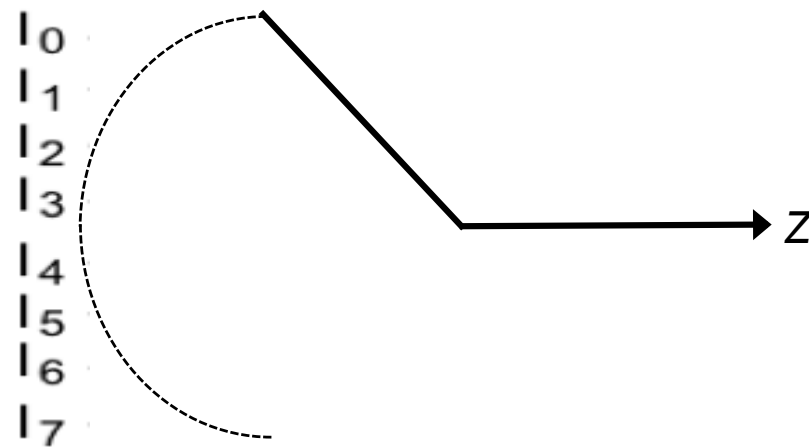
Estas idéias podem ser expandidas em expressões relacionais ou expressões lógicas para computar valores de a nas expressões acima. Estes escalares (a 's) são chamados variáveis de controle.

Aplicações



Multiplexadores/Decodificadores

Multiplexação



Multiplexação - aplicação

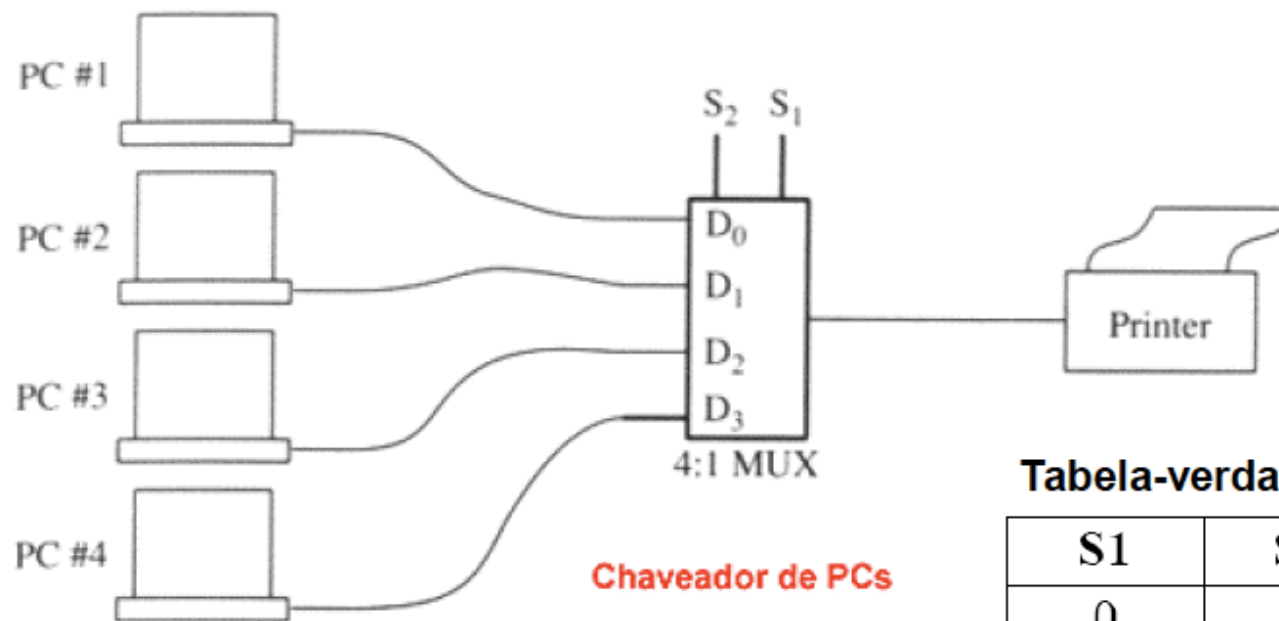


Tabela-verdade do roteador

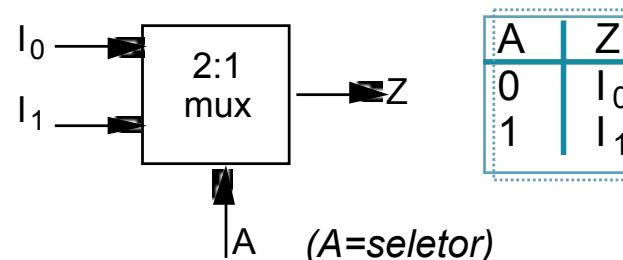
| S1 | S2 | Printer |
|----|----|---------|
| 0 | 0 | PC#1 |
| 0 | 1 | PC#2 |
| 1 | 0 | PC#3 |
| 1 | 1 | PC#4 |

Multiplexadores/Seletores

■ Conceito geral

- A seleção de 2^n entradas é feita através de n linhas de controle que endereçam cada uma destas entradas para a saída.
- Cada entrada possui um endereço determinado, o qual é, em geral, associado a um minitermo.
- A saída recebe o valor da entrada correspondente ao endereço escolhido.

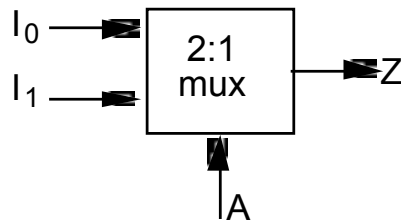
■ Exemplo de um multiplexador/seletor de duas entradas e uma saída (mux 2- \rightarrow 1)



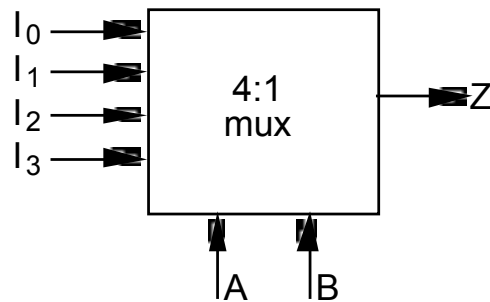
| A | Z |
|---|-------|
| 0 | I_0 |
| 1 | I_1 |

$$Z = A' I_0 + A I_1$$

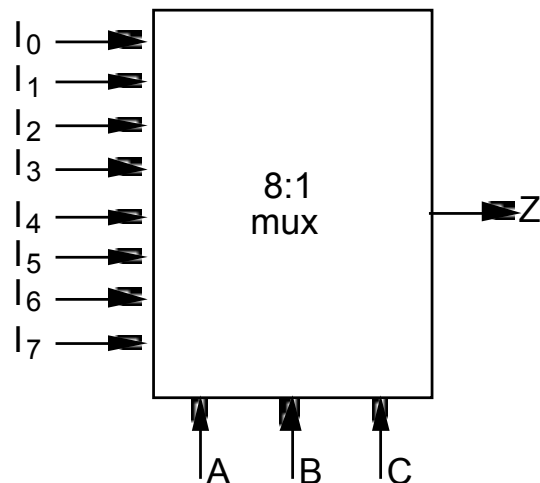
Multiplexadores/Seletores



$$Z = A' I_0 + A I_1$$



$$Z = A' B' I_0 + A' B I_1 + A B' I_2 + A B I_3$$



$$Z = A' B' C' I_0 + A' B' C I_1 + A' B C' I_2 + A' B C I_3 + A B' C' I_4 + A B' C I_5 + A B C' I_6 + A B C I_7$$

A seleção em um multiplexador pode ser dado em geral por:

n = número de variáveis

de seleção

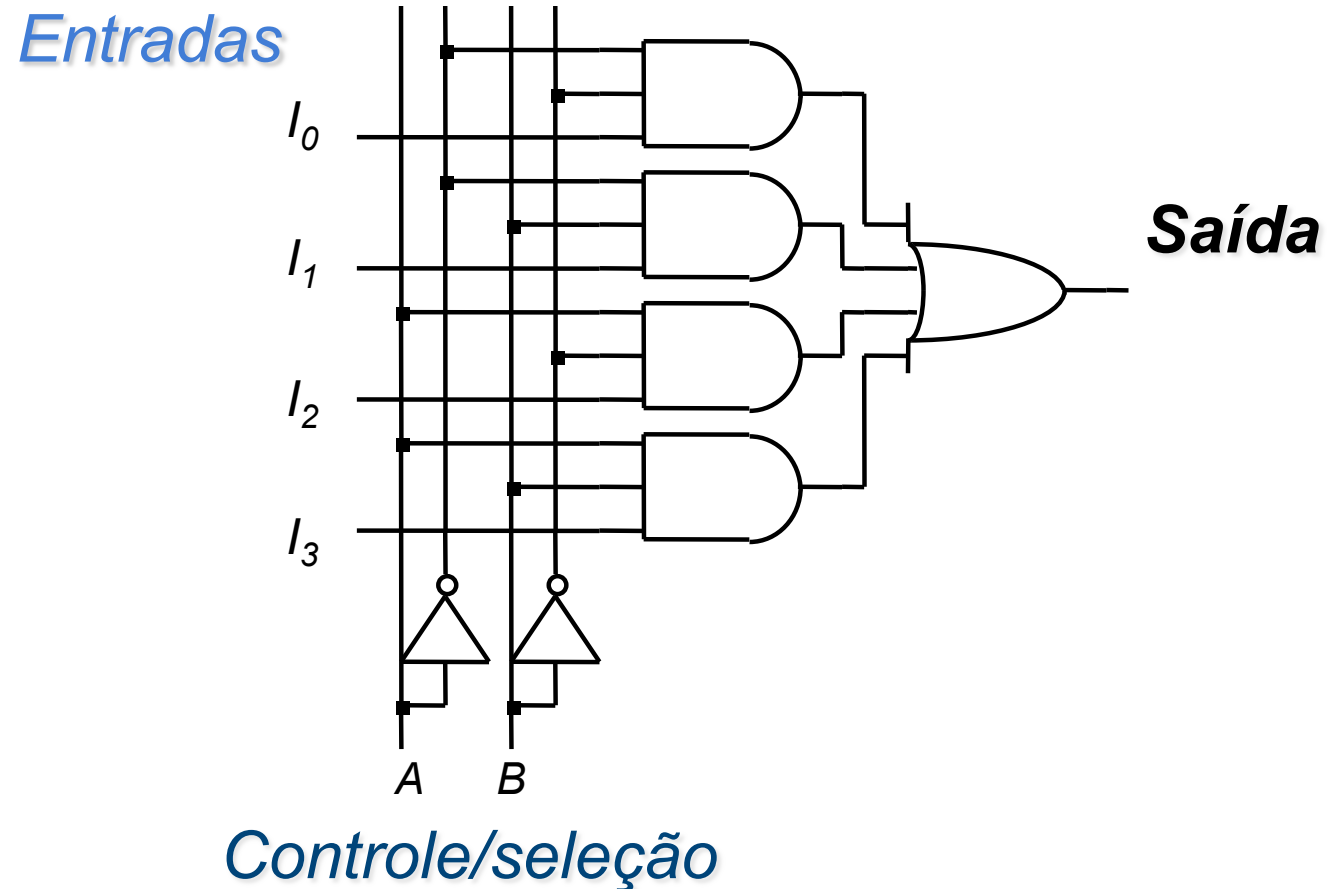
$$Z = \sum_{k=0}^{2^n-1} m_k I_k$$

m = minitermo

I = entrada

Multiplexadores/Seletores

- Implementação de um multiplexador 4:1



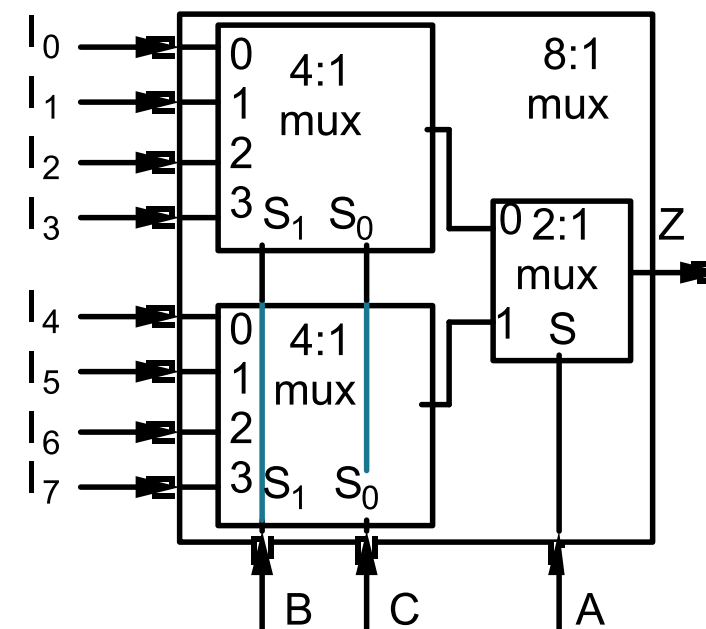
Multiplexadores/Seletores

- Implementação de grandes multiplexadores a partir de pequenos multiplexadores
- Implementação de um multiplexador 8:1 a partir de um multiplexador 2:1 e multiplexadores 4:1

Os controles B e C escolhem uma das entradas de I_0 a I_3 e ao mesmo tempo de I_4 a I_7 entre os muxs 4:1.

O controle A estabelece a saída Z através da seleção no mux 2:1, cujas entradas são saídas dos muxs 4:1

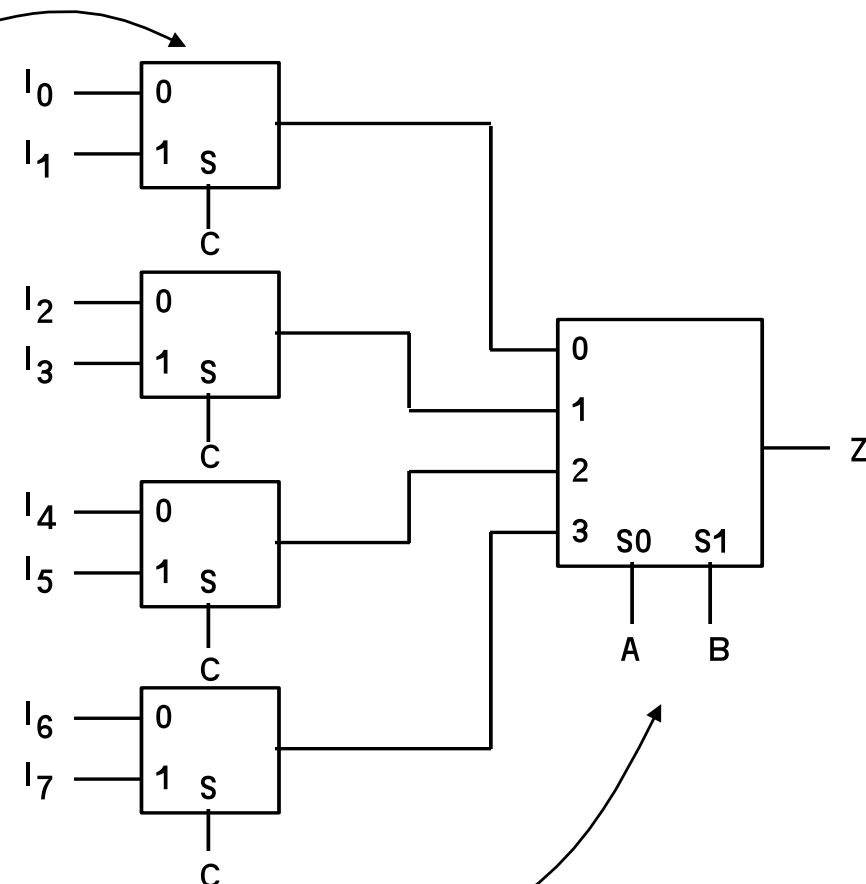
| A | B | C | Z |
|---|---|---|-------|
| 0 | 0 | 0 | I_0 |
| 0 | 0 | 1 | I_1 |
| 0 | 1 | 0 | I_2 |
| 0 | 1 | 1 | I_3 |
| 1 | 0 | 0 | I_4 |
| 1 | 0 | 1 | I_5 |
| 1 | 1 | 0 | I_6 |
| 1 | 1 | 1 | I_7 |



Multiplexadores/Seletores

- Exemplo de multiplexador 8:1 a partir de muxs 2:1 e um mux 4:1

| A | B | C | Z |
|---|---|---|-------|
| 0 | 0 | 0 | I_0 |
| 0 | 0 | 1 | I_1 |
| 0 | 1 | 0 | I_2 |
| 0 | 1 | 1 | I_3 |
| 1 | 0 | 0 | I_4 |
| 1 | 0 | 1 | I_5 |
| 1 | 1 | 0 | I_6 |
| 1 | 1 | 1 | I_7 |

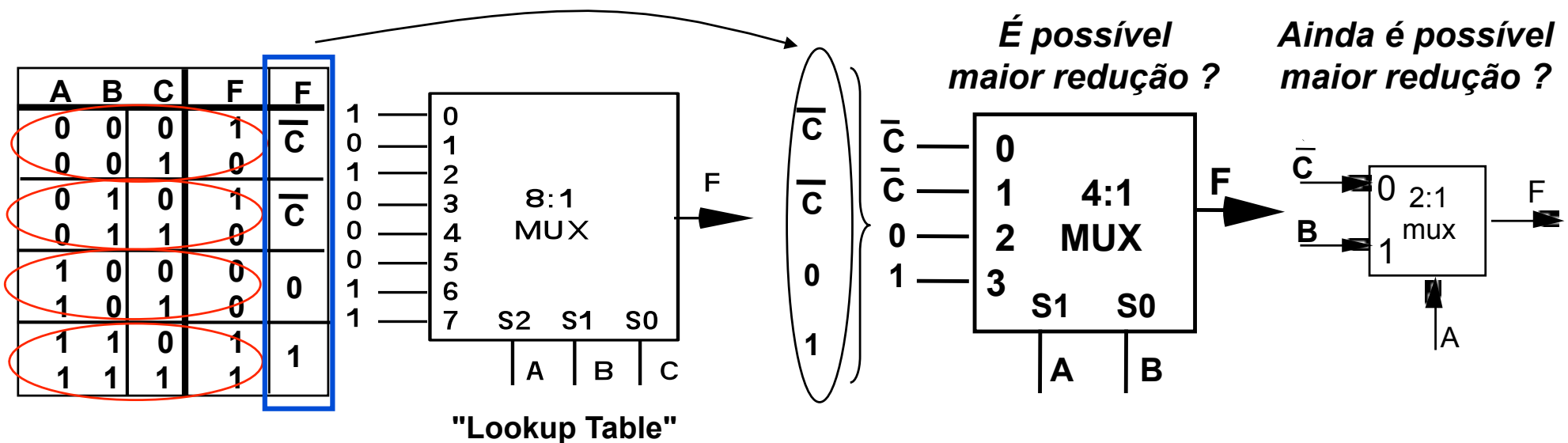


Multiplexadores/Seletores

- Multiplexadores $2^n:1$ podem implementar qualquer função de n variáveis, com $n-1$ variáveis de controle .
As demais variáveis serão entrada para o mux.

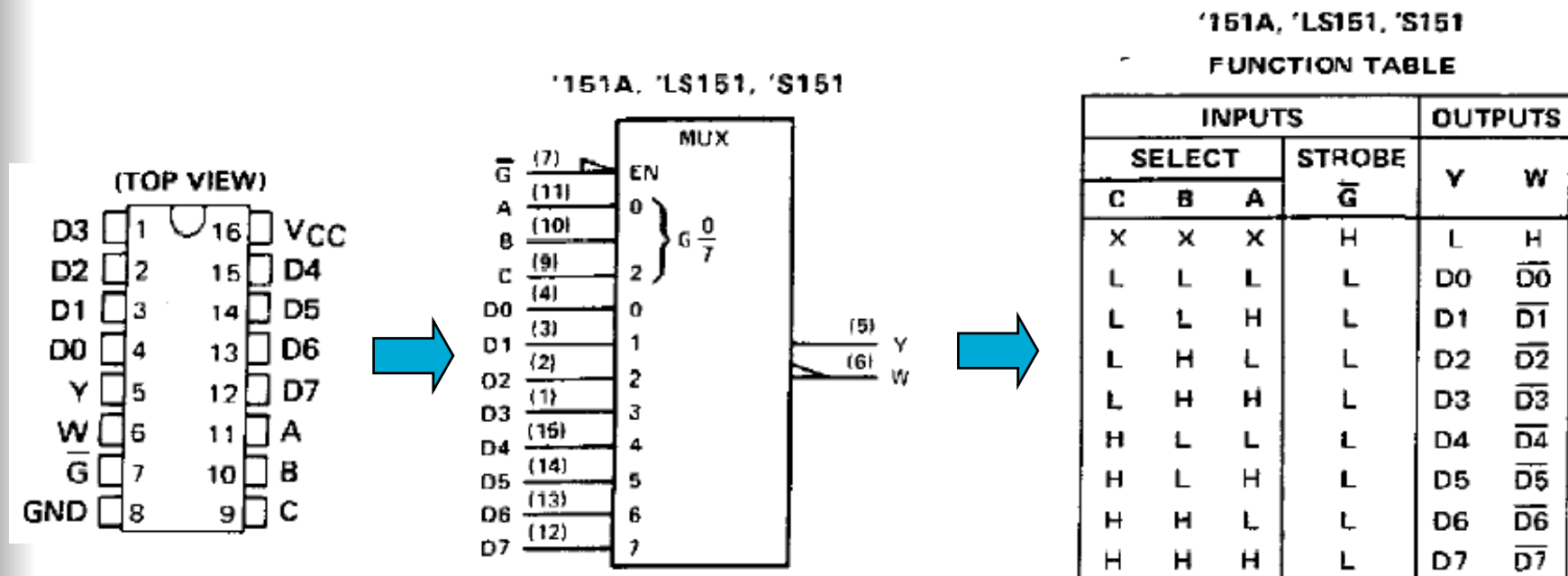
$$\begin{aligned}
 F(A,B,C) &= m_0 + m_2 + m_6 + m_7 \\
 &= A' B' C' + A' B C' + A B C' + A B C \\
 &= A' B' (C') + A' B (C') + A B' (0) + A B (1)
 \end{aligned}$$

Possíveis implementações



Multiplexador comercial – 74LS151

Mux 8:1



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

$\overline{E0}, \overline{E1} \dots \overline{E15}$ = the complement of the level of the respective E input

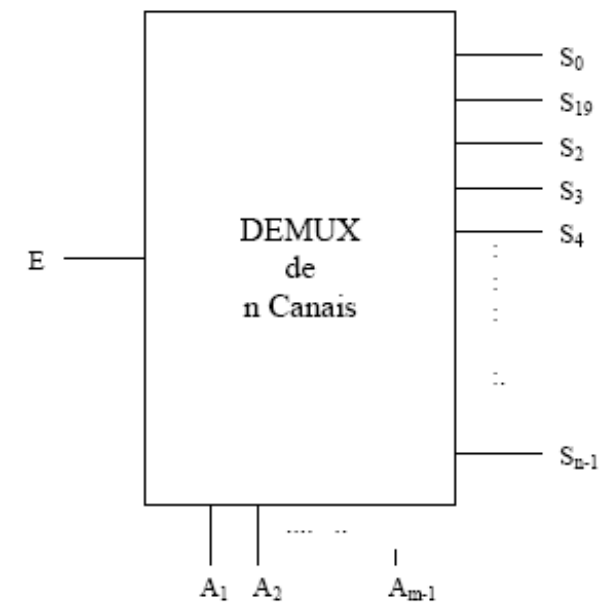
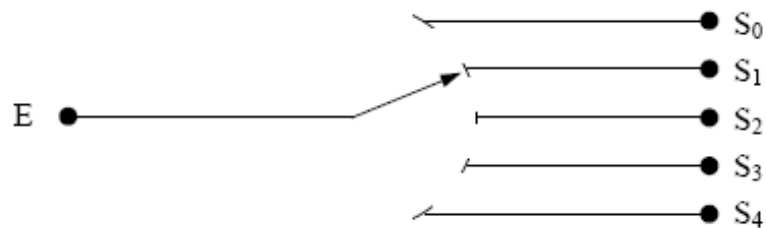
D0, D1 ... D7 = the level of the D respective input

Decodificadores/Demultiplexadores

- Decodificadores/demultiplexadores são utilizados tanto para demultiplexar sinais como para decodificar endereços em um sistema computacional.
- Os decodificadores, assim como os multiplexadores também podem ser usados para implementar funções lógicas. No entanto, em geral, estes dispositivos precisam de lógica externa em sua saída.

Demultiplexador

- O demultiplexador ou Demux é um circuito combinacional dedicado com a finalidade de selecionar, através das variáveis de seleção, qual de suas saídas deve receber a informação presente em sua única entrada, executando a operação inversa realizada pelo Mux.



Demultiplexador

- Da mesma forma que o Mux, no Demux o número de entradas está relacionado com o número de variáveis de seleção, ou seja:
 - $n = 2^m$
 - n - número de canais de saída;
 - m - número de variáveis de seleção.

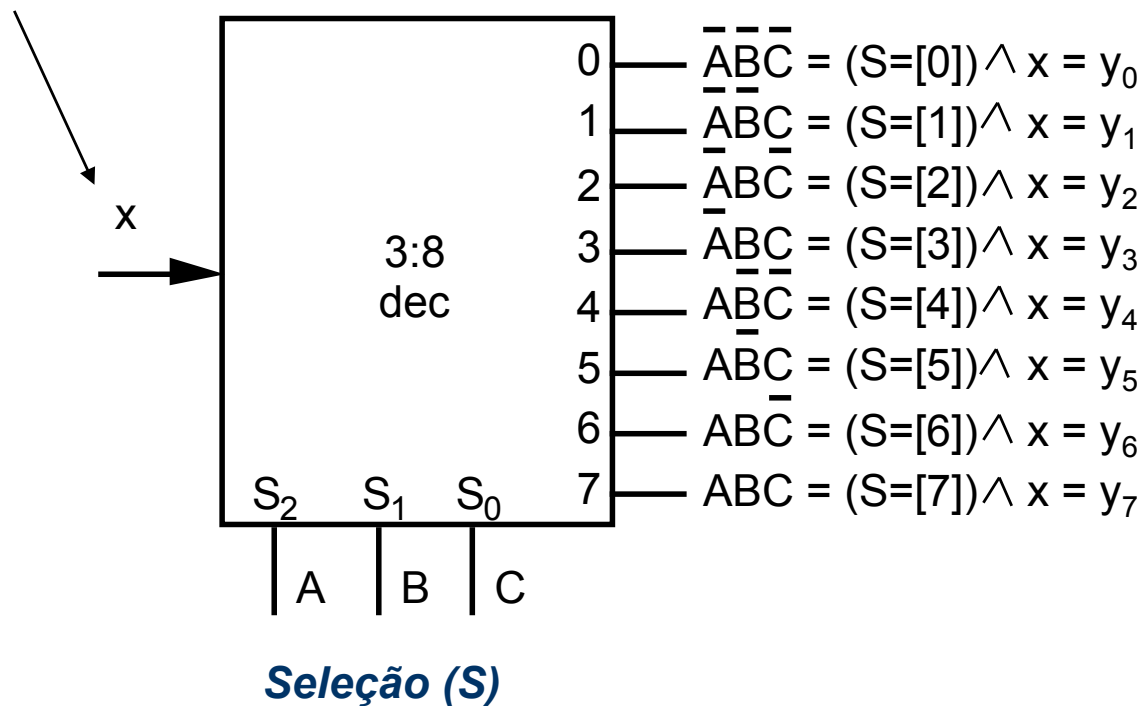
- Para:
 - $m=2$ o circuito possui quatro canais de saída,
 - $m=3$ o circuito possui oito canais de saída

- Algumas aplicações do Demux:
 - seleção de circuitos que devem receber uma determinada informação digital;
 - conversão de informação serial em paralela;

Decodificadores/Demultiplexadores

■ Demultiplexador/Decodificador 3:8 como bloco lógico

Sinal de entrada

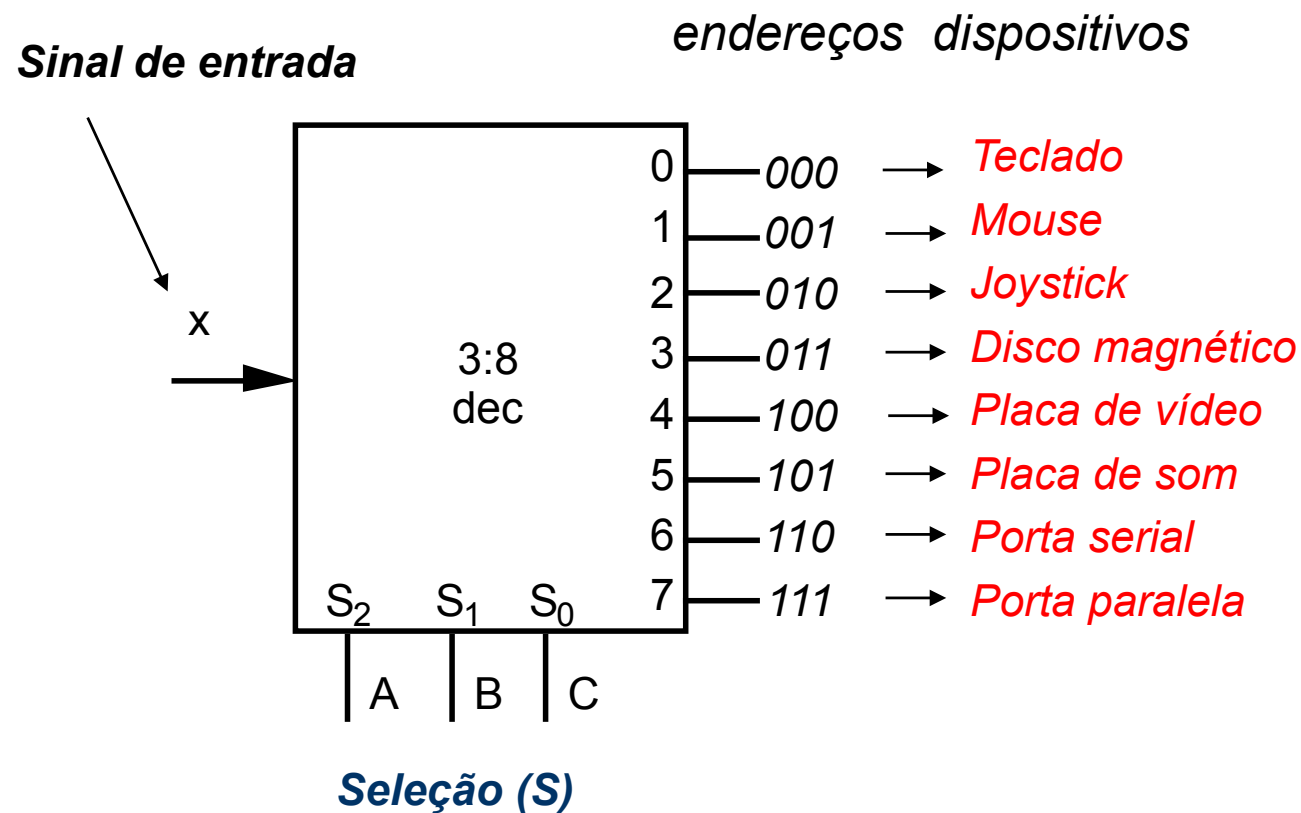


O demultiplexador envia X para saídas equivalentes a codificação dos minitermos

$$y_i = m_i \wedge x$$

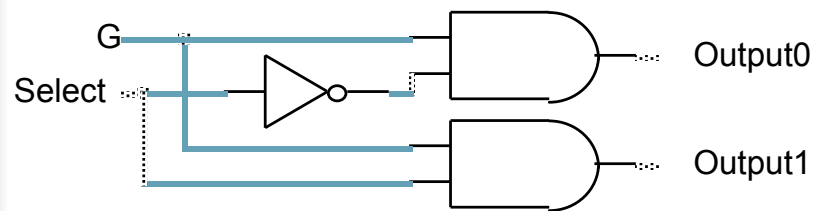
Decodificadores

Decodificadores de dispositivos

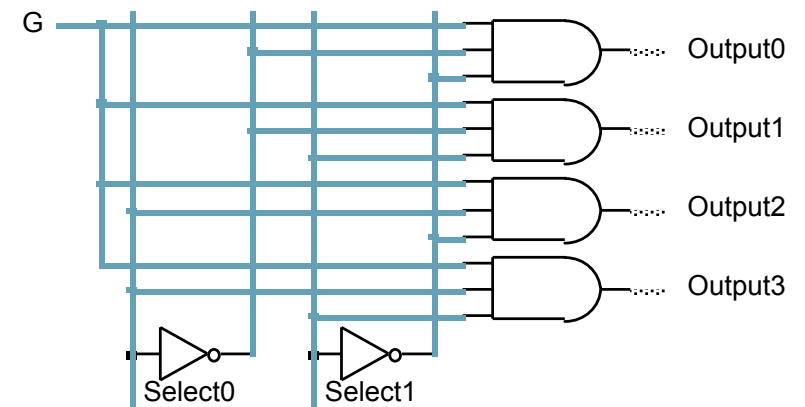


Decodificadores/Demultiplexadores

■ Alternativas de Implementação



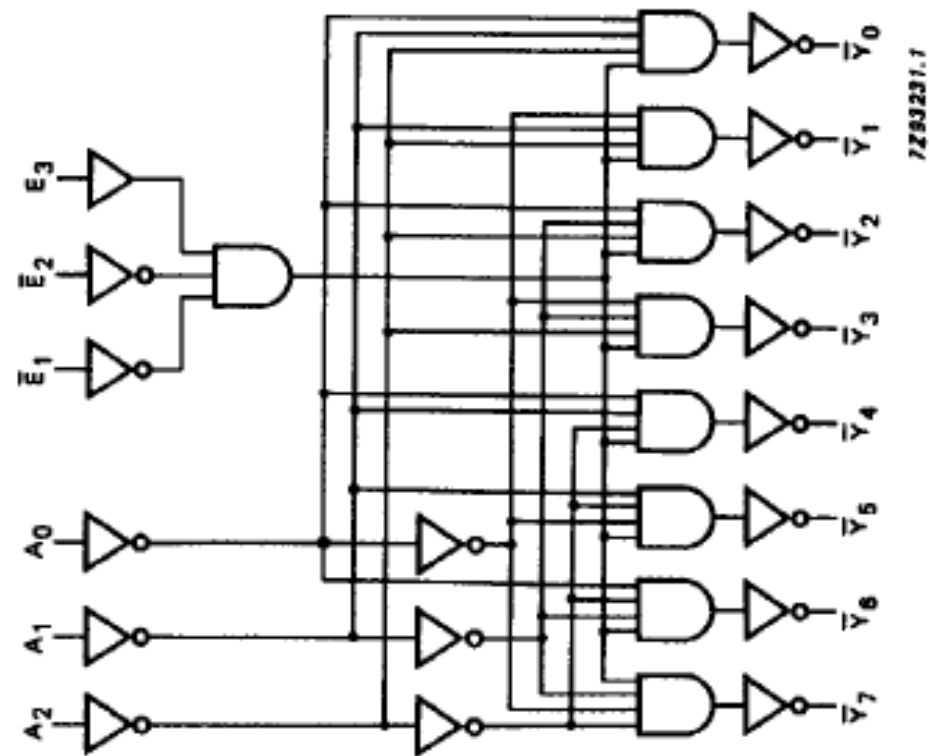
- Demultiplexador 1:2 ou
- Decodificador 1:2 com habilitação ativa em alta ("1")



- Demultiplexador 2:4 ou
- Decodificador 2:4 com habilitação em alta ("1")

Decodificador/demultiplexador 74138

Diagrama Lógico



Decodificador/demultiplexador 74138

PIN DESCRIPTION

| PIN NO. | SYMBOL | NAME AND FUNCTION |
|------------------------------|----------------------------|----------------------------|
| 1, 2, 3 | A_0 to A_2 | address inputs |
| 4, 5 | \bar{E}_1, \bar{E}_2 | enable inputs (active LOW) |
| 6 | E_3 | enable input (active HIGH) |
| 8 | GND | ground (0 V) |
| 15, 14, 13, 12, 11, 10, 9, 7 | \bar{Y}_0 to \bar{Y}_7 | outputs (active LOW) |
| 16 | V_{CC} | positive supply voltage |

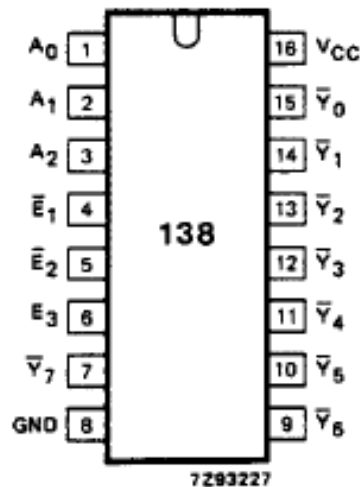


Fig.1 Pin configuration.

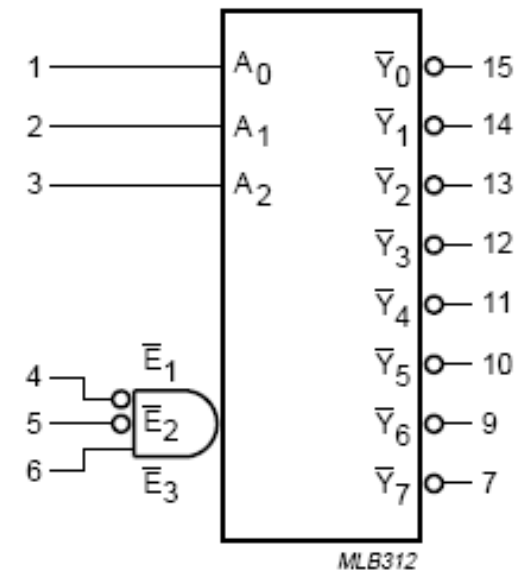


Fig.2 Logic symbol.

Decodificador 74 138 – tabela verdade

3-to-8 line decoder/demultiplexer; inverting

FUNCTION TABLE

| INPUTS | | | | | | OUTPUTS | | | | | | | |
|-------------|-------------|-------|-------|-------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| \bar{E}_1 | \bar{E}_2 | E_3 | A_0 | A_1 | A_2 | \bar{Y}_0 | \bar{Y}_1 | \bar{Y}_2 | \bar{Y}_3 | \bar{Y}_4 | \bar{Y}_5 | \bar{Y}_6 | \bar{Y}_7 |
| H | X | X | X | X | X | H | H | H | H | H | H | H | H |
| X | H | X | X | X | X | H | H | H | H | H | H | H | H |
| X | X | L | X | X | X | H | H | H | H | H | H | H | H |
| L | L | H | L | L | L | L | H | H | H | H | H | H | H |
| L | L | H | H | L | L | H | L | H | H | H | H | H | H |
| L | L | H | L | H | L | H | H | L | H | H | H | H | H |
| L | L | H | H | H | L | H | H | H | L | H | H | H | H |
| L | L | H | L | L | H | H | H | H | H | L | H | H | H |
| L | L | H | H | L | H | H | H | H | H | H | L | H | H |
| L | L | H | L | H | H | H | H | H | H | H | H | L | H |
| L | L | H | H | H | H | H | H | H | H | H | H | H | L |

Notes

1. H = HIGH voltage level
L = LOW voltage level
X = don't care

Decodificadores

■ Exemplo

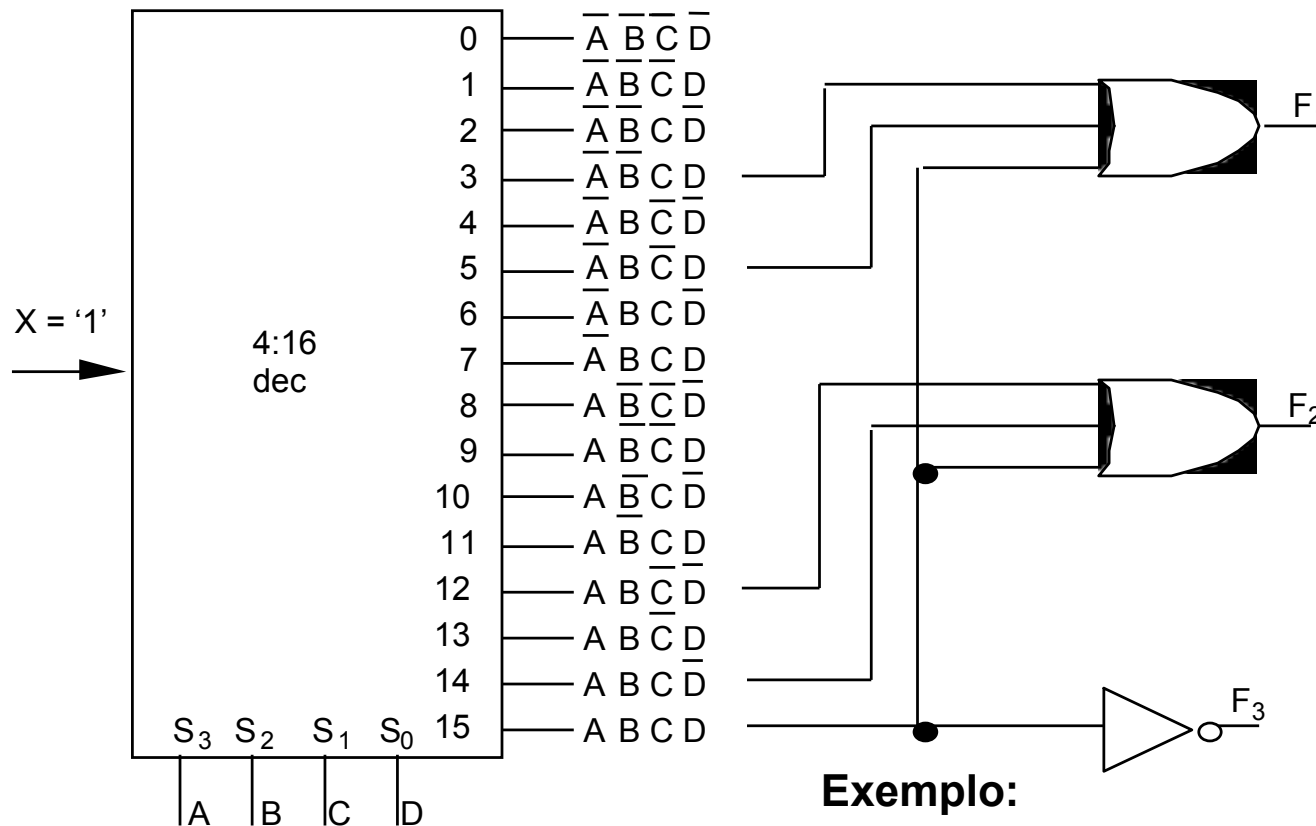
Implemente um decodificador 4- \rightarrow 16 a partir de decodificadores 2- \rightarrow 4 do tipo descrito abaixo:

| G2 | G1 | A | B | Y0 | Y1 | Y2 | Y3 |
|----|----|---|---|----|----|----|----|
| 1 | X | X | X | 1 | 1 | 1 | 1 |
| X | 0 | X | X | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 |

G1 e G2 são controles que habilitam a saída. A saída Y0-Y3 selecionada fica em '0'. As demais ficam em '1'.

Decodificadores/demultiplexadores

■ Decodificadores como gerador de funções



$$F_1 = \overline{A} \overline{B} C \overline{D} + \overline{A} \overline{B} C D + \overline{A} B \overline{C} \overline{D}$$

$$F_2 = A \overline{B} C \overline{D} + A \overline{B} C D$$

$$F_3 = (\overline{A} + \overline{B} + \overline{C} + \overline{D})$$