

INF01 118

Técnicas Digitais para Computação

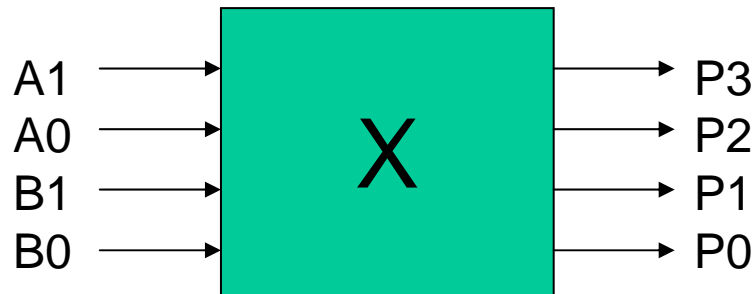
Multiplicador

Decodificador e Multiplexador

Aula 14

Multiplicador Combinacional

$$A_1A_0 \times B_1B_0 = P_3P_2P_1P_0$$



1)

A_1	A_0	B_1	B_0	P_3	P_2	P_1	P_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

2) Equações em SDP, simplificado por mapa de Karnaugh

$$P_3 = m_{15}$$

$$P_2 = \sum m(10, 11, 14)$$

$$P_1 = \sum m(6, 7, 9, 11, 13, 14)$$

$$P_0 = \sum m(5, 7, 13, 15)$$



$$P_0 = A_0B_0$$

$$P_1 = \bar{A}_1A_0B_1 + \bar{A}_1B_0A_1 + A_1\bar{A}_0B_0 + B_1\bar{B}_0A_0$$

$$P_2 = A_1\bar{A}_0B_1 + A_1B_1\bar{B}_0$$

$$P_3 = A_1A_0B_1B_0$$

Multiplicador de 2 bits. Síntese de circuito combinacional

X_1	X_0	Y_1	Y_0	P_3	P_2	P_1	P_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

$$P_3 = m_{15}$$

$$P_2 = \sum m(10, 11, 14)$$

$$P_1 = \sum m(6, 7, 9, 11, 13, 14)$$

$$P_0 = \sum m(5, 7, 13, 15)$$

P_1

$Y_1 Y_0$	00	01	11	10
$X_1 X_0$				
00	0	0	0	0
01	0	0	1	1
11	0	1	0	1
10	0	1	1	0

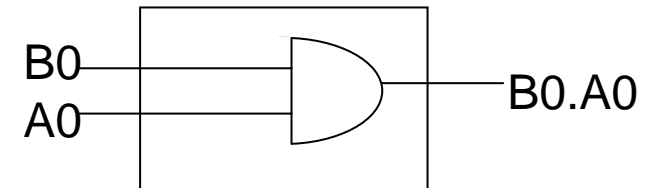
$$P_1 = \overline{X_1}X_0Y_1 + \overline{Y_1}Y_0X_1 + X_1\overline{X_0}Y_0 + Y_1\overline{Y_0}X_0$$

Multiplicador Combinacional

$$A_1A_0 \times B_1B_0 = P_3P_2P_1P_0$$

$$\begin{array}{r}
 \begin{array}{c} x \\ + \end{array}
 \begin{array}{r}
 A_1 A_0 \\
 B_1 B_0 \\
 \hline
 B_0 A_1 \\
 B_1 A_1 B_1 A_0 \\
 \hline
 P_3 P_2 P_1 P_0
 \end{array}
 \end{array}$$

1) Multiplicação é equivalente a uma operação E



2) Soma binaria dos termos:

$$\begin{aligned}\mathbf{Soma} &= \mathbf{X} \oplus \mathbf{Y} \oplus \mathbf{C}_{in} \\ \mathbf{C}_{out} &= \mathbf{XY} + \mathbf{XC}_{in} + \mathbf{YC}_{in}\end{aligned}$$

3) Substituindo os X e Y pelos termos da multiplicação:

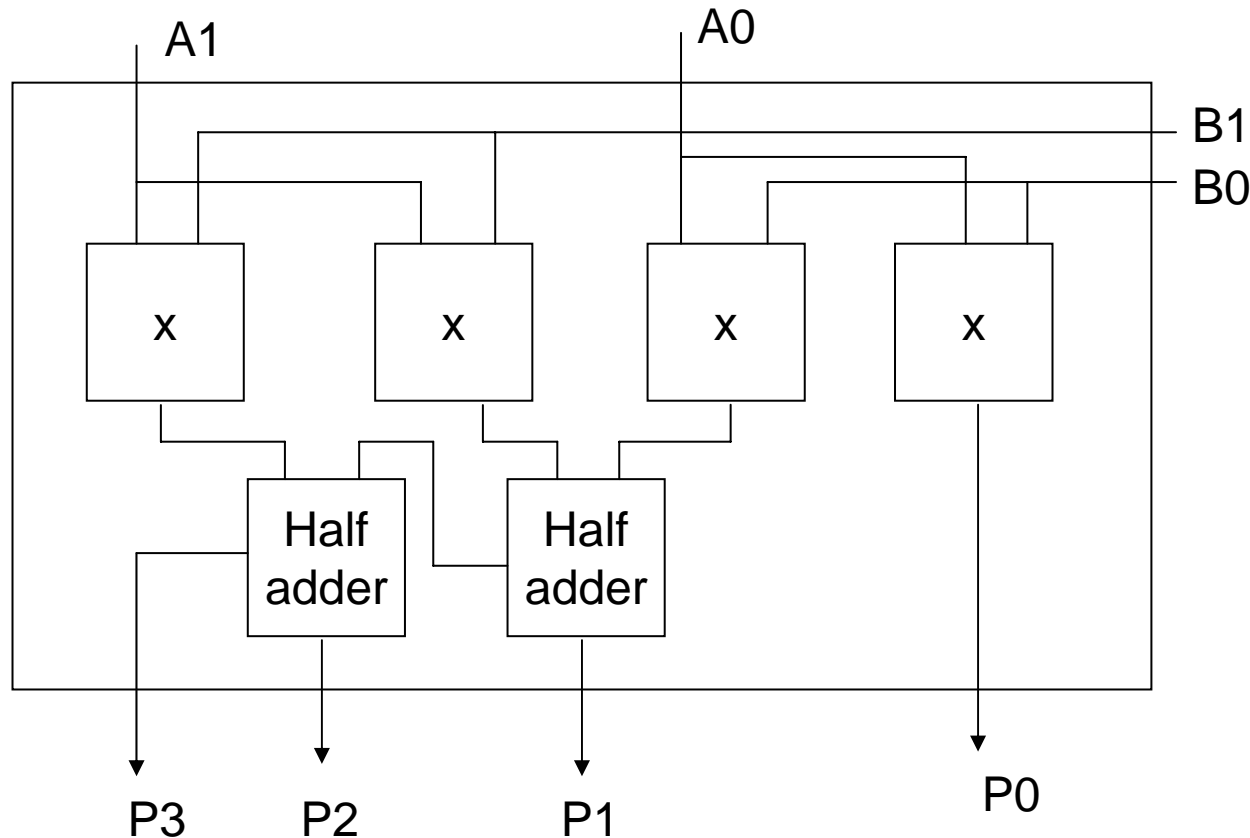
$$\mathbf{P}_0 = \mathbf{A}_0 \mathbf{B}_0$$

$$P_1 = \bar{A}_1 A_0 B_1 + \bar{A}_1 B_0 A_1 + A_1 \bar{A}_0 B_0 + B_1 \bar{B}_0 A_0$$

$$\mathbf{P}_2 = \mathbf{A}_1 \bar{\mathbf{A}}_0 \mathbf{B}_1 + \mathbf{A}_1 \mathbf{B}_1 \bar{\mathbf{B}}_0$$

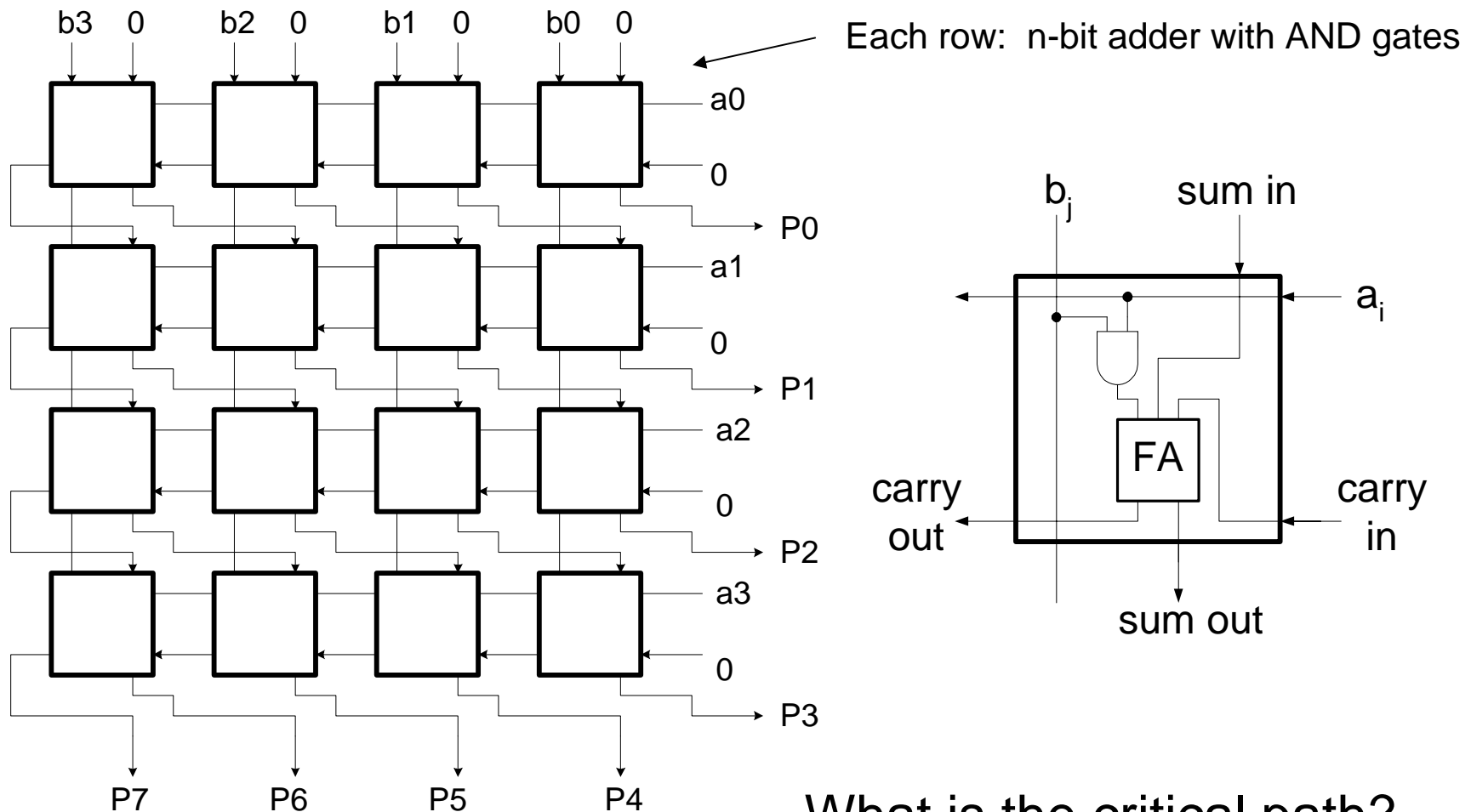
$$\mathbf{P3} = \mathbf{A}_1 \mathbf{A}_0 \mathbf{B}_1 \mathbf{B}_0$$

$$\begin{array}{r}
 \begin{array}{cccc}
 & & & A_1 A_0 \\
 & & & B_1 B_0 \\
 x & & & \\
 \hline
 & & B_0 A_1 & \\
 + & B_1 A_1 & B_1 A_0 & \\
 \hline
 P_3 & P_2 & P_1 & P_0
 \end{array}
 \end{array}$$



Array Multiplier

Generates all n partial products simultaneously.



What is the critical path?

Multiplicador em VHDL

```

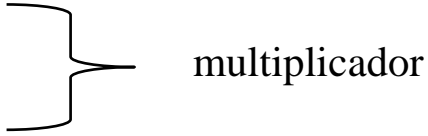
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity ULA is
  Port ( A : in  STD_LOGIC_VECTOR (3 downto 0);
        B : in  STD_LOGIC_VECTOR (3 downto 0);
        F : out STD_LOGIC_VECTOR (7 downto 0));
end ULA;

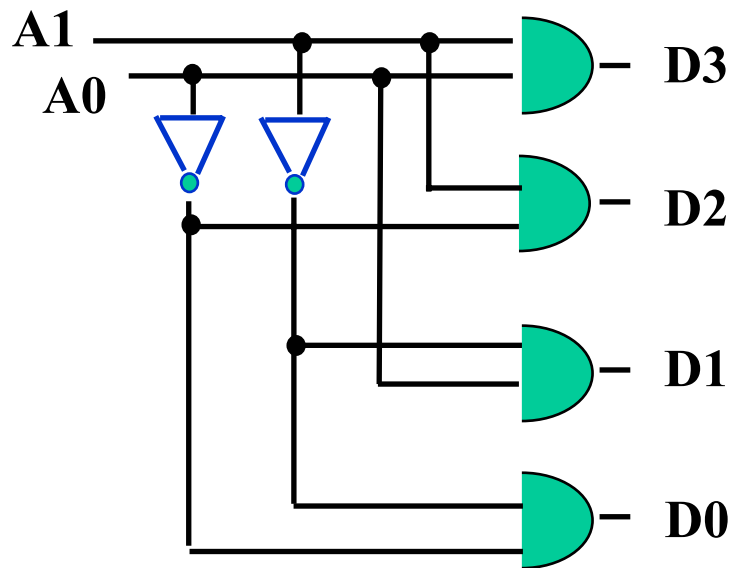
architecture Behavioral of ULA is
  Begin

  process (A, B)
  begin
    F<= A*B;
  end process;

end Behavioral;
  
```



Decodificadores



Apenas uma saída é igual a 1
Ex: se $A_1A_0 = 10$ então $D_2 = 1$

↓
“2”

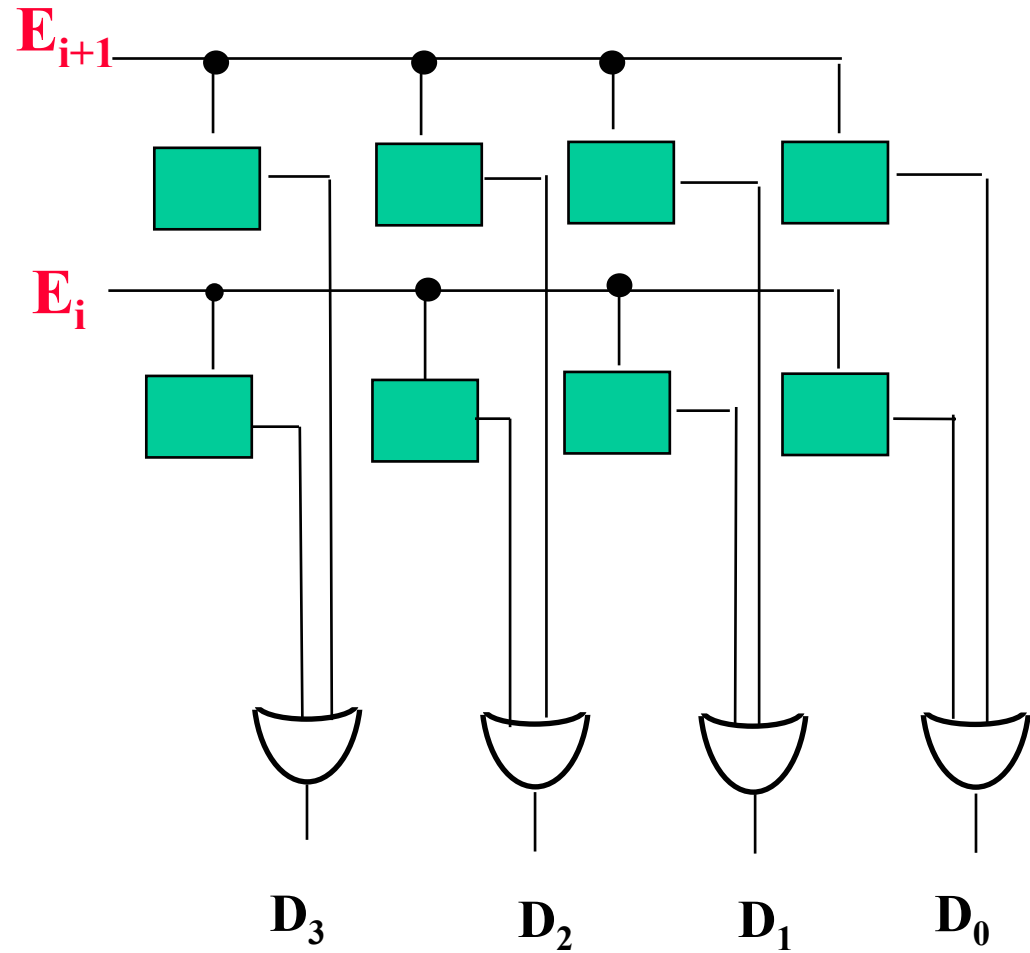
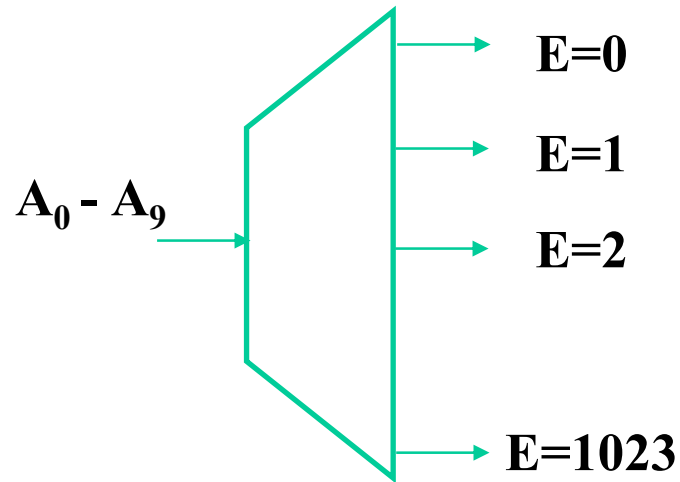
A_1	A_0	D_0	D_1	D_2	D_3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Imaginando-se que A_1A_0 seja um código, as saídas o decodificam.

Se cada saída tivesse uma lâmpada (LED) que acendesse quando esta saída fosse igual a 1, e se esta lâmpada iluminasse um número com o algarismo decodificado, teríamos explicitamente na saída a informação sobre o código detectado.

Aplicações

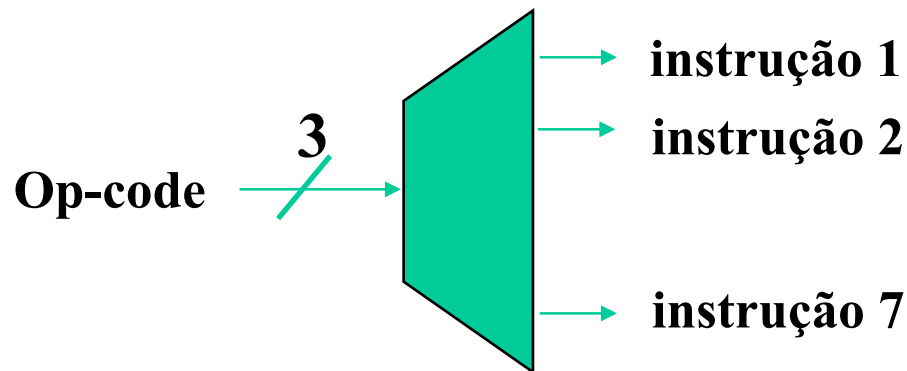
- Decodificação de endereço em uma memória



• Decodificador de instruções



Formato de uma instrução



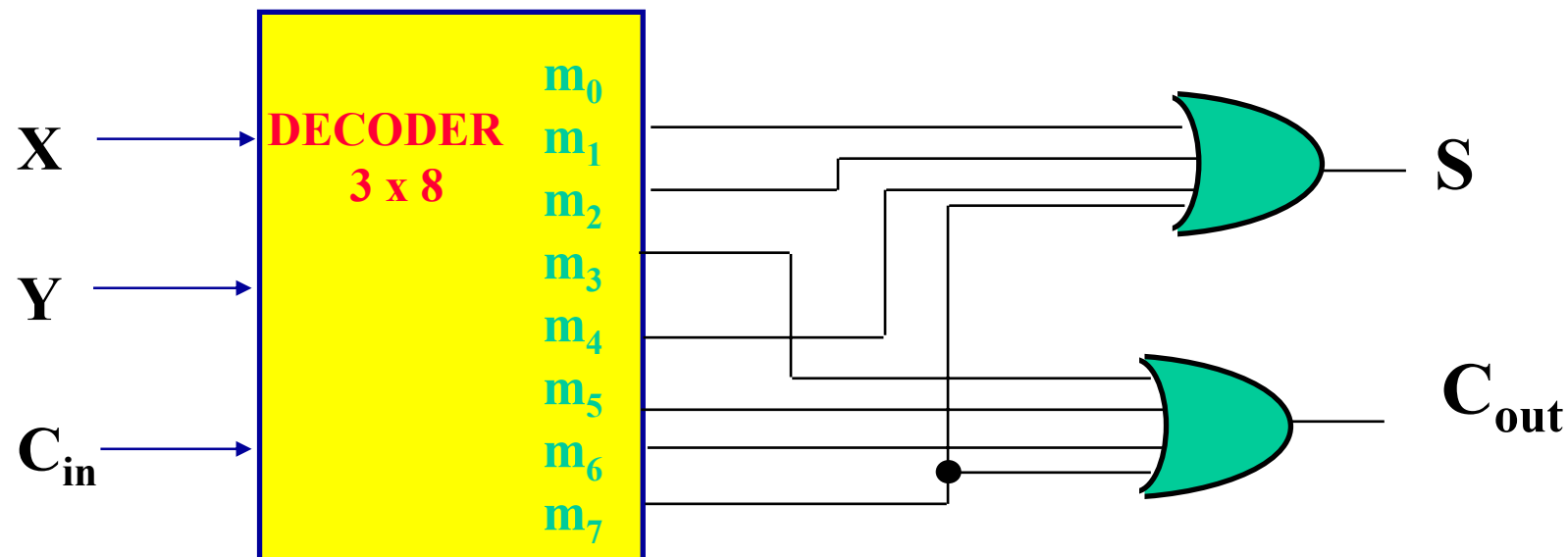
• Implementação de funções combinacionais

- Imaginando que as entradas são as variáveis X,Y,Z de uma função F
- As 8 saídas correspondem então aos mintermos
- Tomando a soma dos produtos usando mintermos, basta fazer um OR entre os mintermos para os quais a função é = 1.

Exemplo: Full Adder

$$S(X, Y, C_{in}) = \Sigma m(1, 2, 4, 7)$$

$$C_{out}(X, Y, C_{in}) = \Sigma m(3, 5, 6, 7)$$



Codificadores

Codificador simples

- 2^n entradas \rightarrow n saídas
- Apenas uma entrada pode ter valor = 1
- Saída fornece código binário correspondente à entrada ligada.

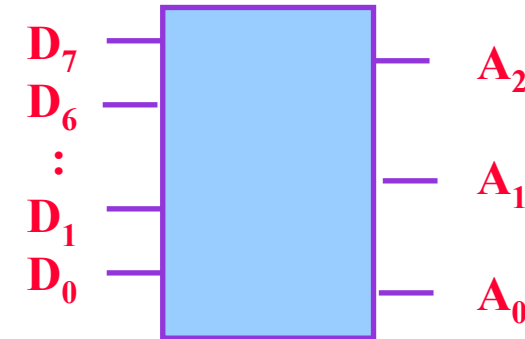


Tabela-verdade

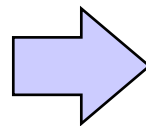
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1	0	0	0	0	0	0	0	1	1	1

Implementação

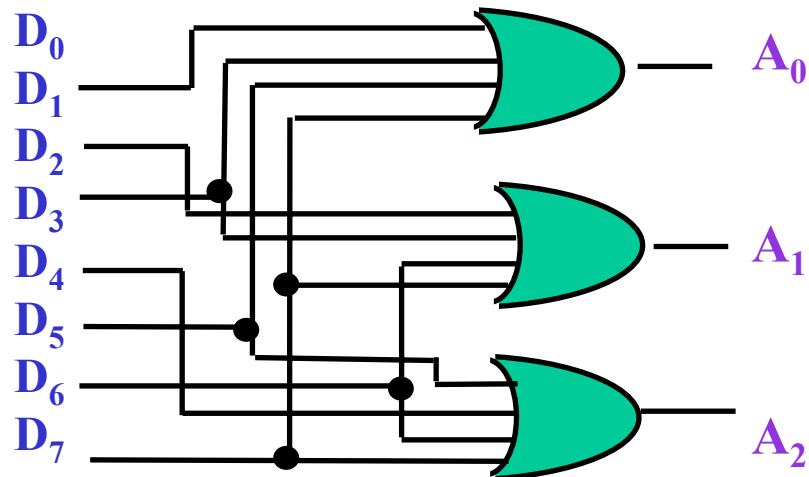
$$A_0 = D_1 + D_3 + D_5 + D_7$$

$$A_1 = D_2 + D_3 + D_6 + D_7$$

$$A_2 = D_4 + D_5 + D_6 + D_7$$



3 Portas OR
de 4 entradas



Problemas

Se mais de uma entrada = 1

Ex: $D_3 = 1$ e $D_6 = 1$



$A_2 A_1 A_0 = 1 1 1 \Rightarrow$ como se $D_7 = 1$

Se nenhuma entrada = 1

$A_2 A_1 A_0 = 000$, como se $D_0 = 1$

Codificador de prioridade

Se duas entradas são iguais a 1 simultaneamente, a entrada de maior prioridade tem precedência.

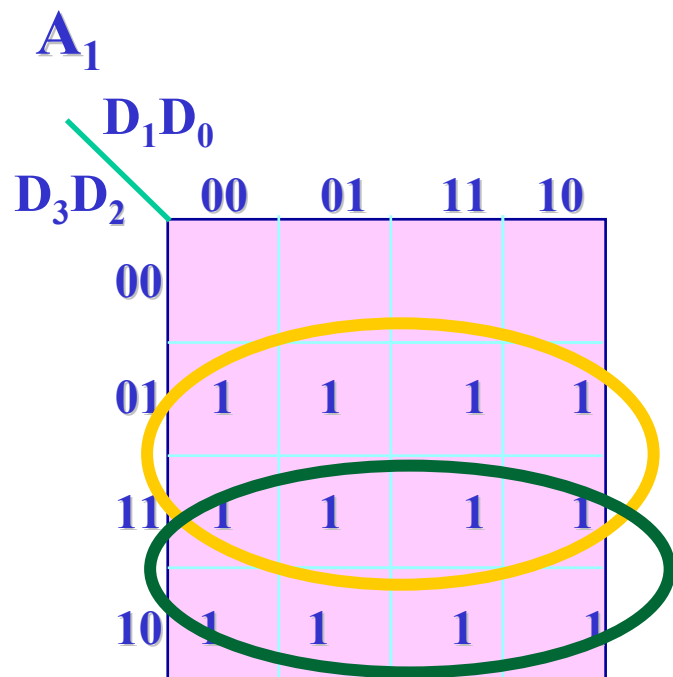
Tabela - verdade

D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

V indica saída válida
(pelo menos uma entrada = 1)

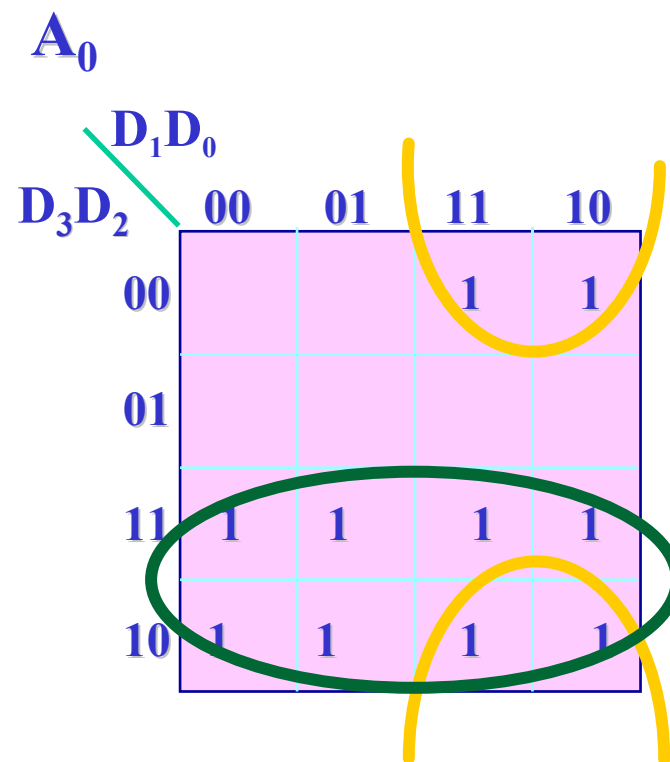
X = don't care

Mapas de Karnaugh



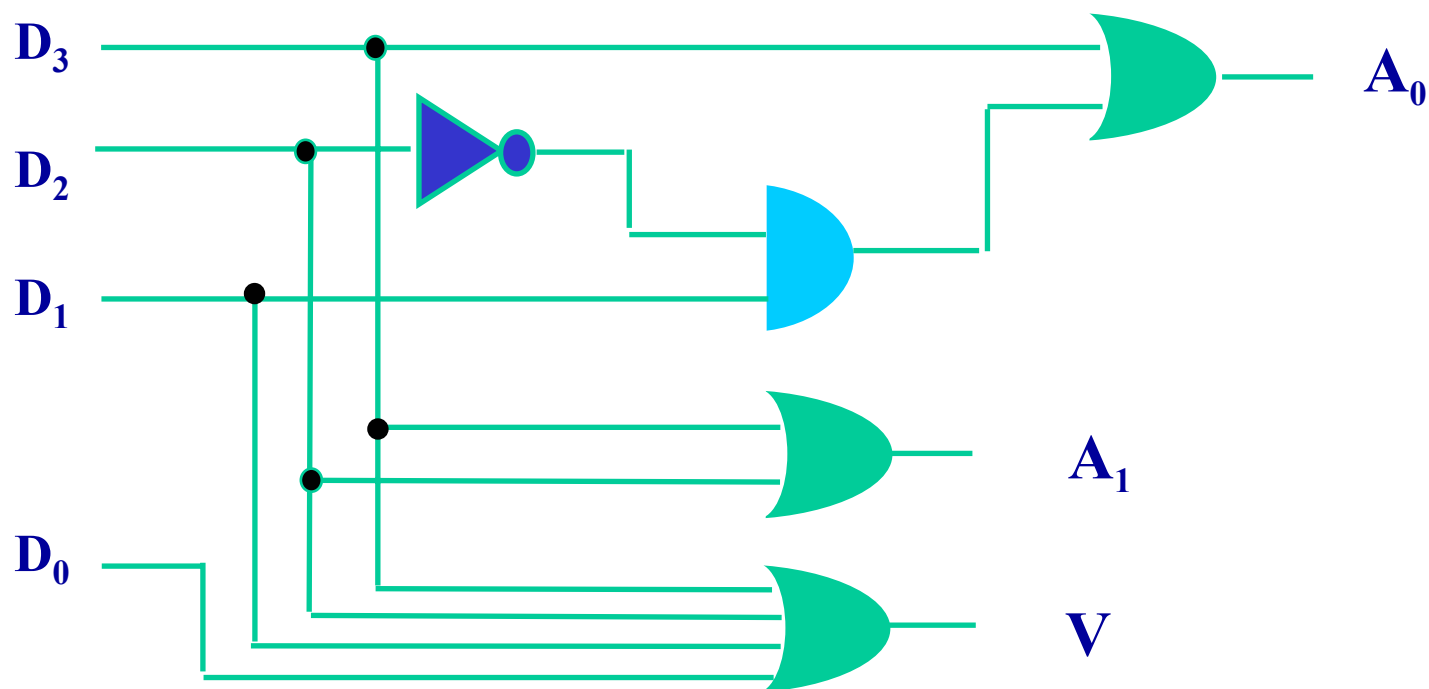
$$A_1 = D_2 + D_3$$

$$V = D_3 + D_2 + D_1 + D_0$$



$$A_0 = D_3 + D_1 \cdot \overline{D_2}$$

Implementação



Multiplexadores

- Seleciona 1 de 2^n entradas e a conecta à saída
- Seleção controlada por n sinais de controle

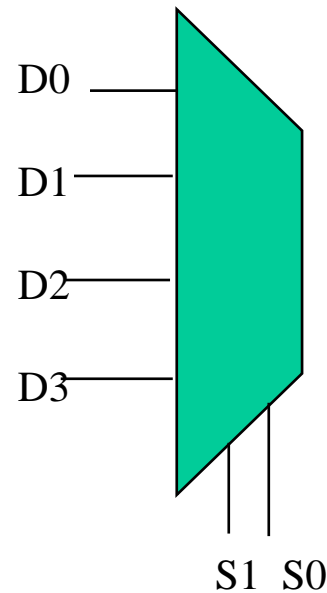
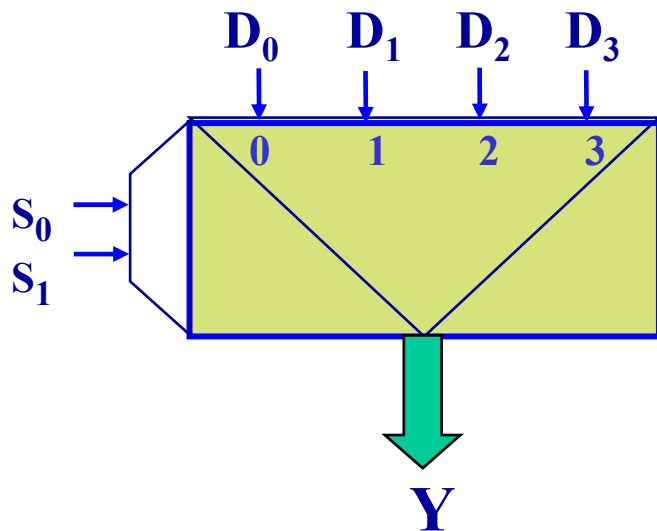
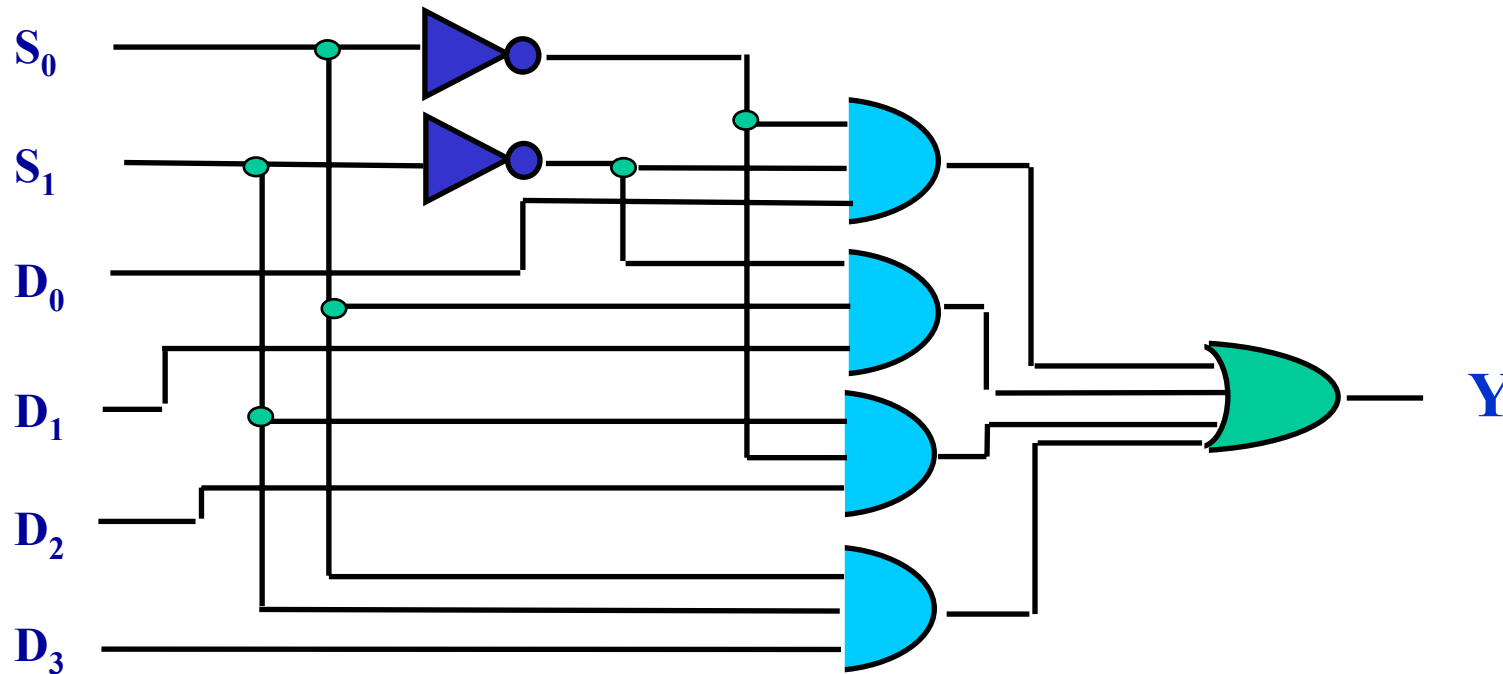


Tabela de Função

S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

$$Y = \bar{S}_0 \bar{S}_1 D_0 + \bar{S}_0 S_1 D_1 + S_0 \bar{S}_1 D_2 + S_0 S_1 D_3$$

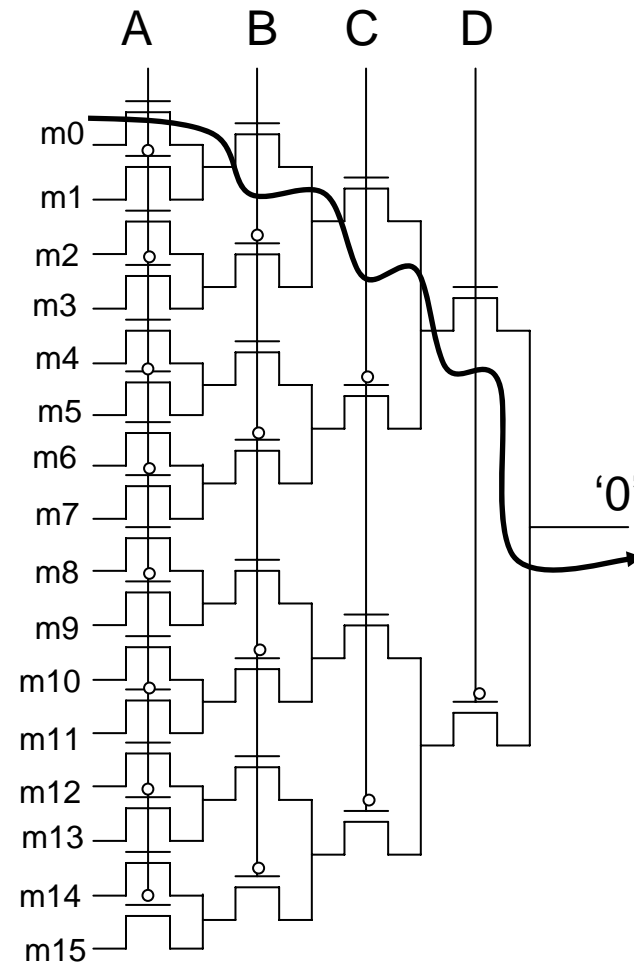
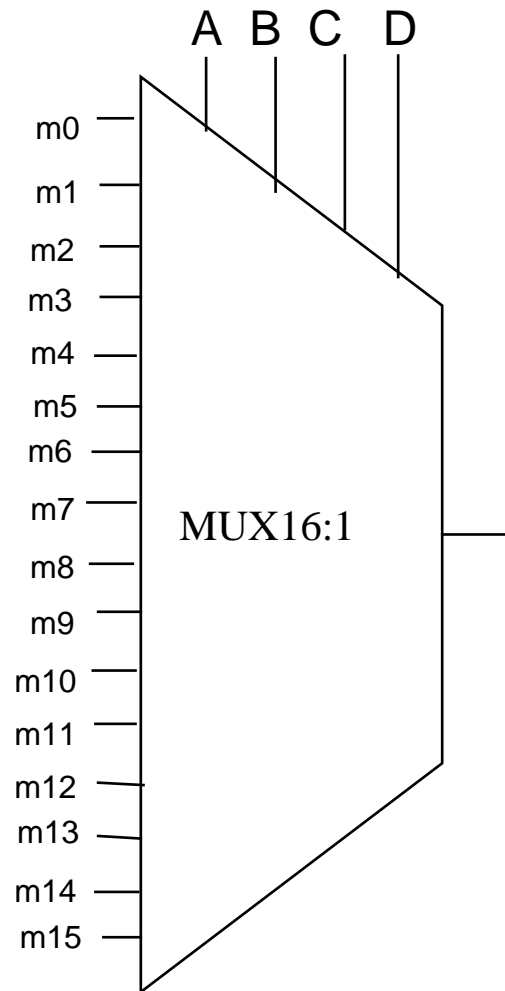
Implementação



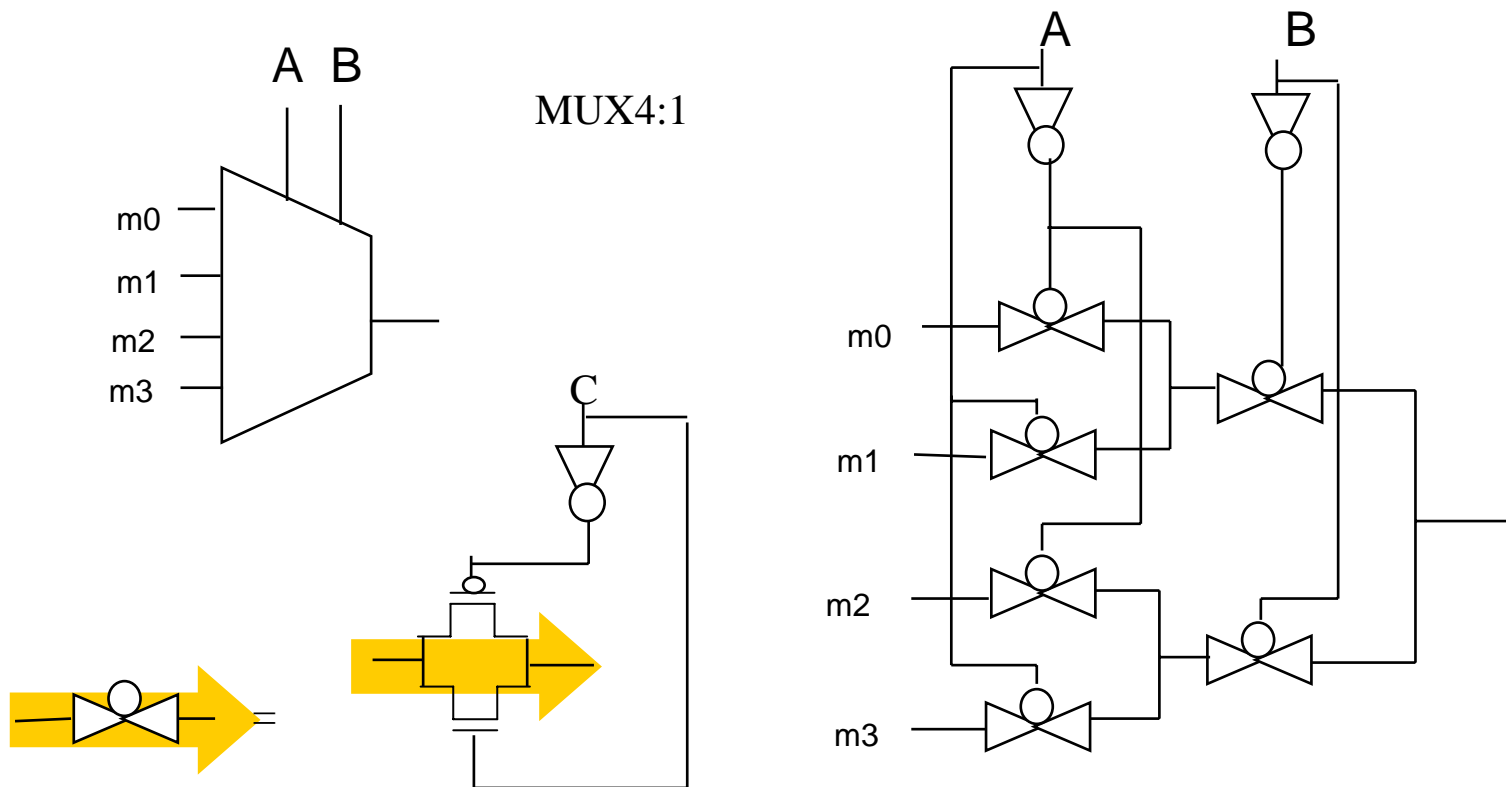
Um mux é um decodificador ao qual foram acrescentados

- Uma entrada de dados D_0 - D_3 em cada AND
(portanto decodifica S_0 - S_1 e deixa passar o D_i correspondente)
- Uma porta **OR** na saída

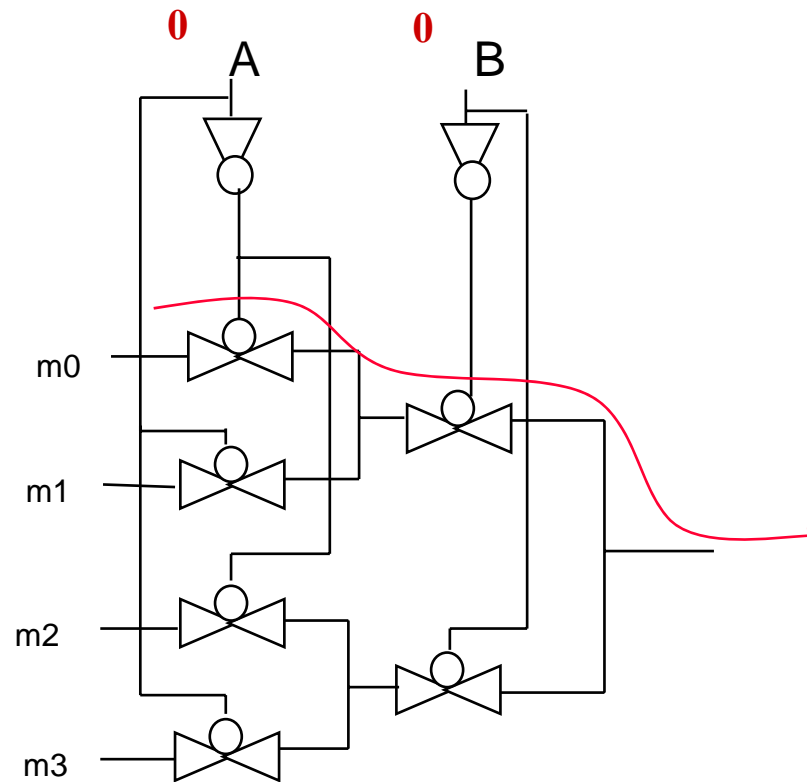
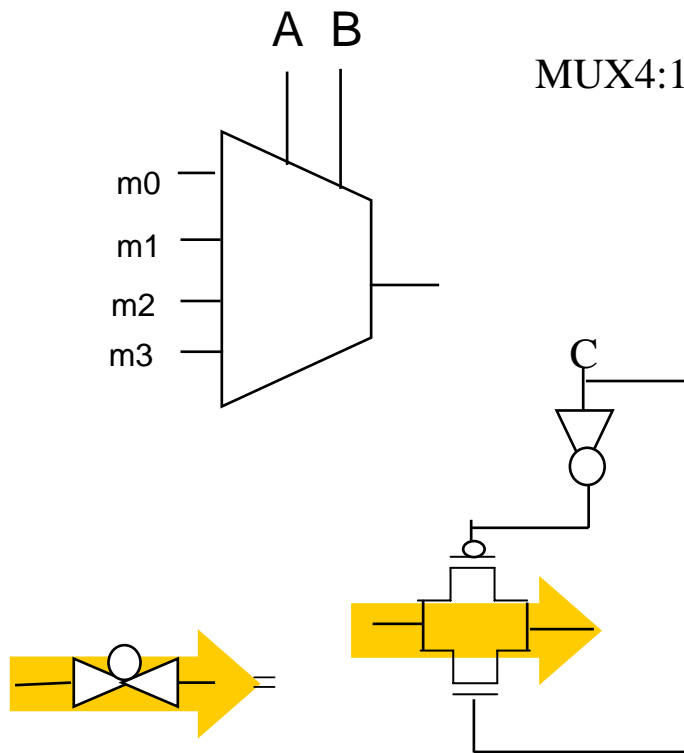
Implementação de Multiplexador por transistor de passagem



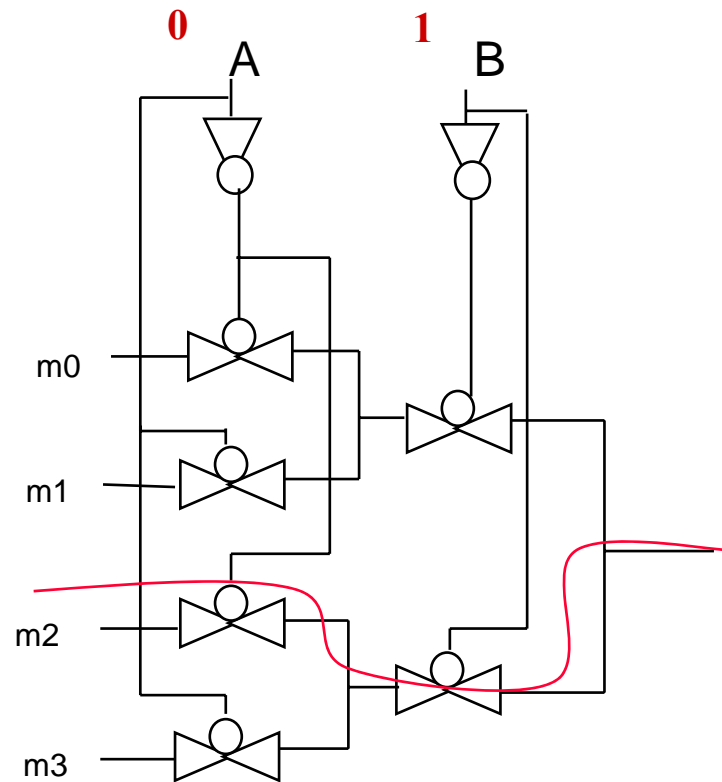
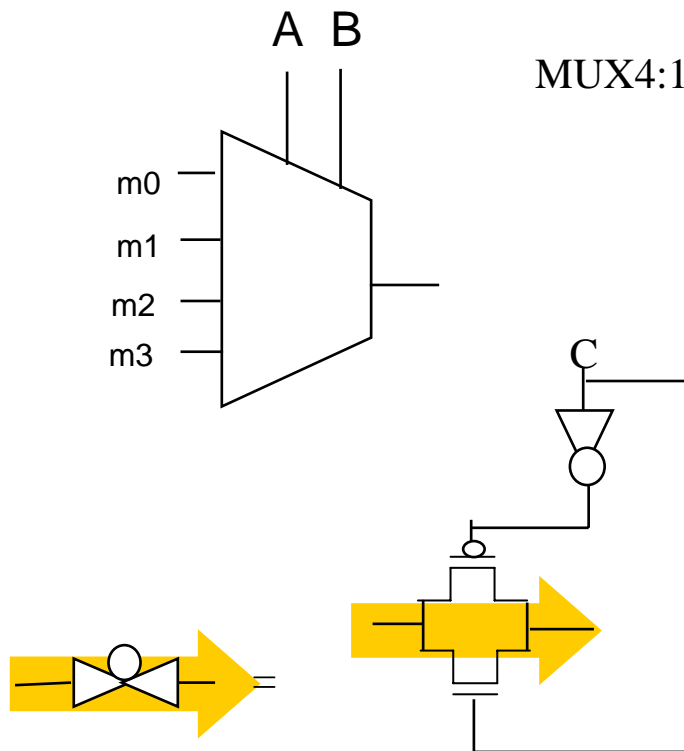
Implementação de Multiplexador por porta de transmissão (TG)



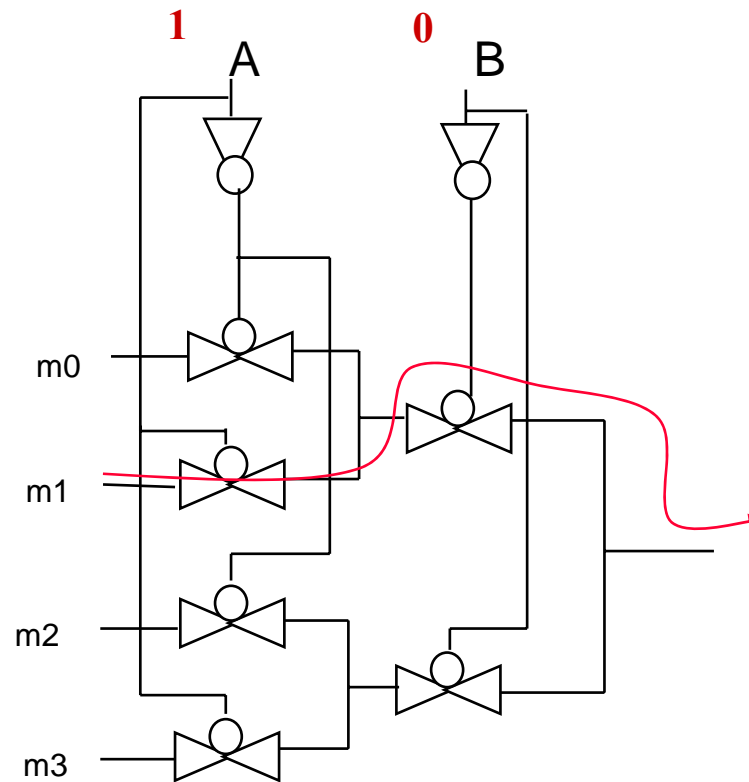
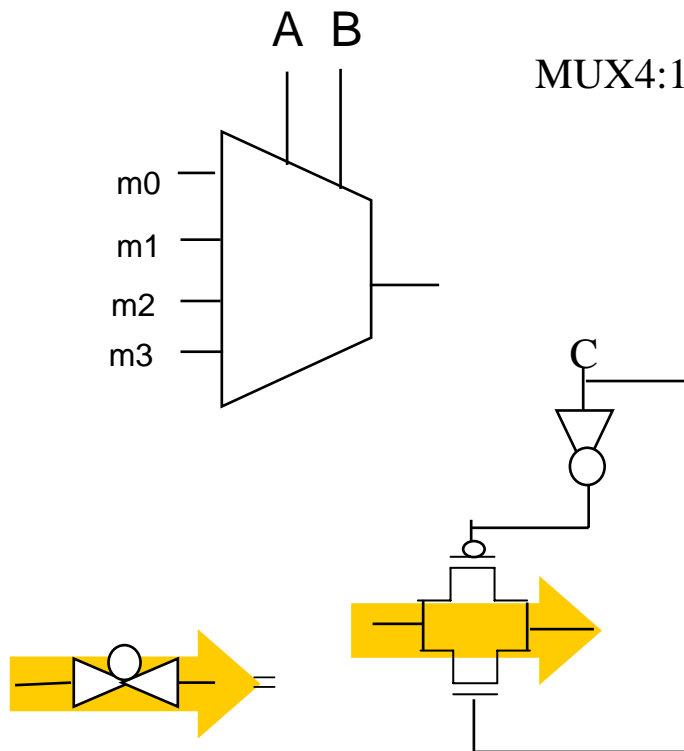
Implementação de Multiplexador por porta de transmissão (TG)



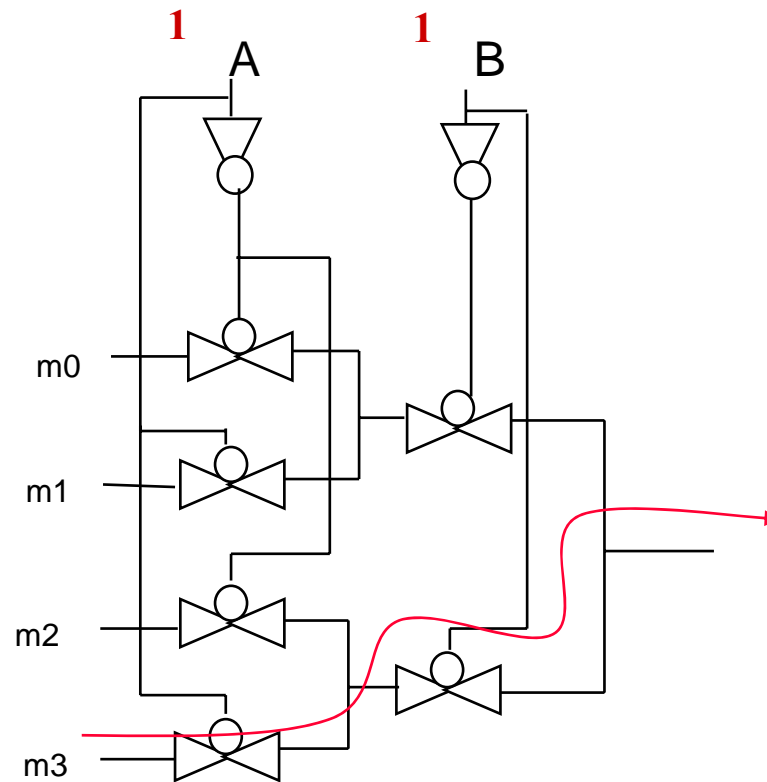
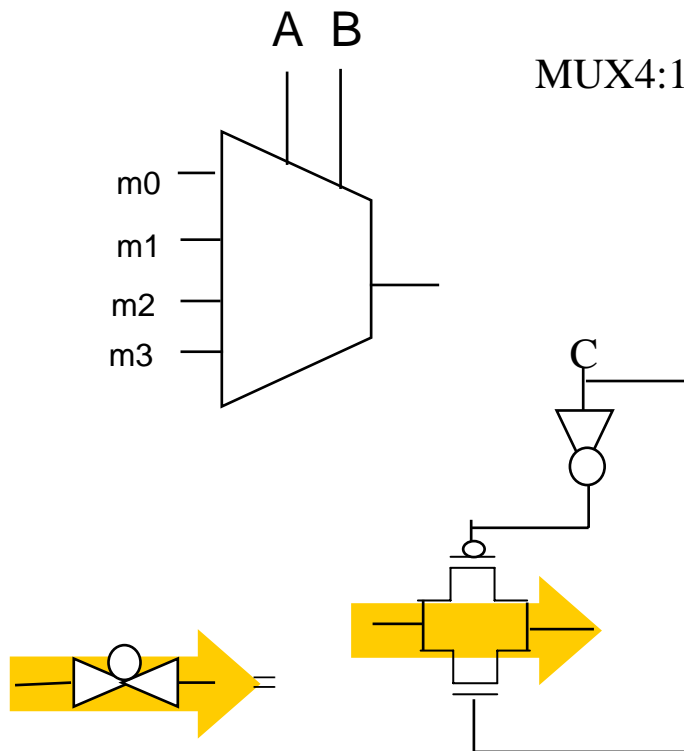
Implementação de Multiplexador por porta de transmissão (TG)



Implementação de Multiplexador por porta de transmissão (TG)



Implementação de Multiplexador por porta de transmissão (TG)



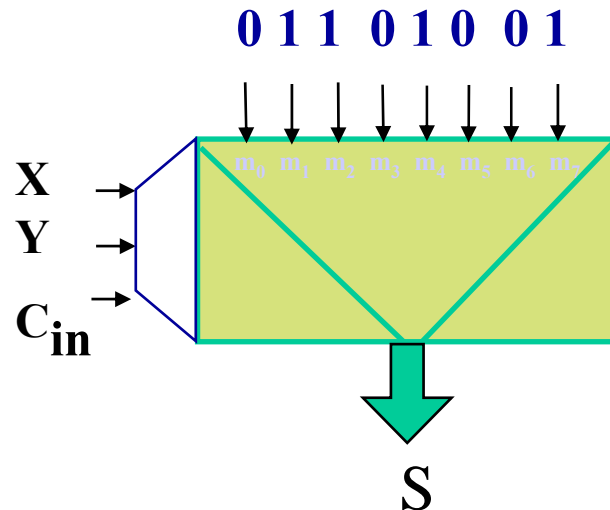
Implementação de funções booleanas

• Alternativa 1

- Usando decoder: coloca-se na saída um OR dos mintermos desejados
- Considerar que o MUX é um decoder que já tem o OR

Portanto: usar MUX, selecionando mintermos através das entradas de dados

• Exemplo de somador: $S = \Sigma m(1,2,4,7)$



- Esta solução exige, para n variáveis, um MUX de n entradas de seleção.

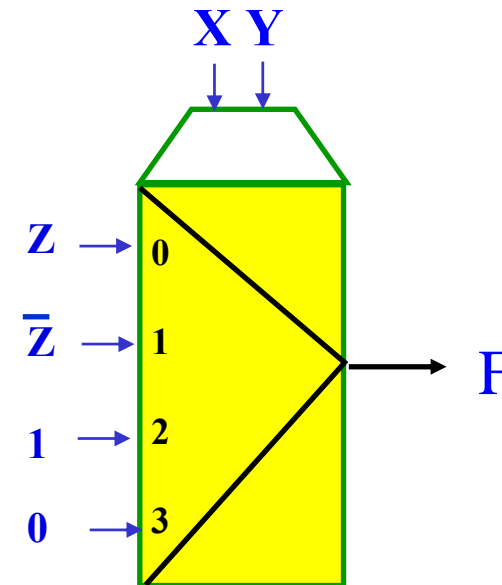
• Alternativa 2

- Solução exige, para **n** variáveis , um MUX de **n-1** entradas de seleção (metade do tamanho de um MUX com **n** entradas de seleção)
- Método
 - Aplicar **n-1** primeiras variáveis como entradas de seleção
 - Usar variável restante (**Z**) como entrada de dados
 - Expressar saída como função de \bar{Z} , **Z**, 0, 1 (as únicas 4 alternativas que existem)

• Exemplo

$$F = \sum m(1,2,4,5)$$

X	Y	Z	F	
0	0	0	0	$F=Z$
0	0	1	1	
0	1	0	1	$F=\bar{Z}$
0	1	1	0	
1	0	0	1	$F=1$
1	0	1	1	
1	1	0	0	$F=0$
1	1	1	0	



- Exemplo do somador: $S = \Sigma m(1,2,4,7)$

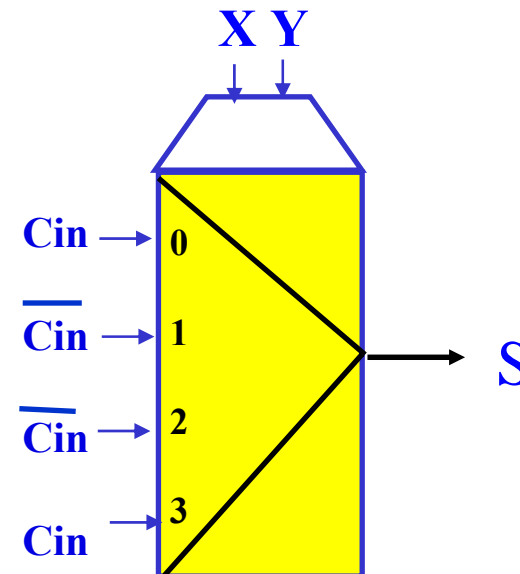
X	Y	Cin	S
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$$S = C_{in}$$

$$S = \overline{C_{in}}$$

$$S = \overline{C_{in}}$$

$$S = C_{in}$$



- Outra aplicação de multiplexadores:
Seleção de caminhos de dados

Exercícios

- **Questão 1:** Implemente as seguintes funções Booleanas utilizando multiplexadores. Mostre o circuito final composto apenas por multiplexadores. Pode-se usar MUX 2:1, MUX 4:1 e MUX 8:1, conforme preferência. Justifique a escolha.
- $F = A'.B + A'.B'.C' + B.C'.D + A.B.D' + B'.C.D$
- $G = A.B.C + A.B.C' + A'.B'.C$

- **Questão 2:** Determine a equação Booleana da função $Z1$ e $Z2$, minimizada, do circuito a seguir composto por multiplexadores 2:1.

