

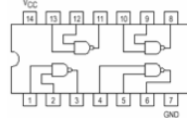
## Circuitos Integrados

Classificação quanto ao nível de integração

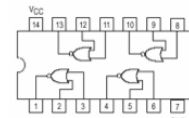
Tipo de circuito integrado	Nº de portas lógicas
SSI ("Small-scale integrated")	<10
MSI ("Medium-scale integrated")	De 10 a 100
LSI ("Large-scale integrated")	De 100 a algumas centenas
VLSI ("Very large-scale integrated")	De algumas centenas até ...

## Circuitos Integrados SSI

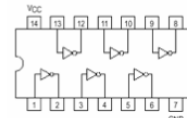
00 - Quad 2-input NAND gate



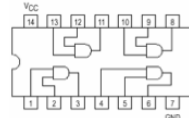
02 - Quad 2 input NOR



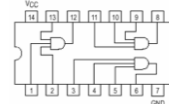
04 - Hex inverters



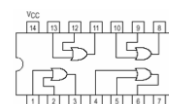
08 - Quad 2-input AND gates.



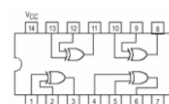
11 - Triple 3-input AND gates



32 - Quad 2-input OR gates



86 - Quad 2 Input Exclusive OR

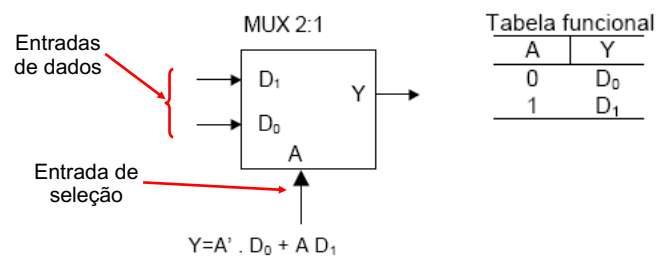


## Circuitos MSI/LSI

## Multiplexer/Seletor de dados

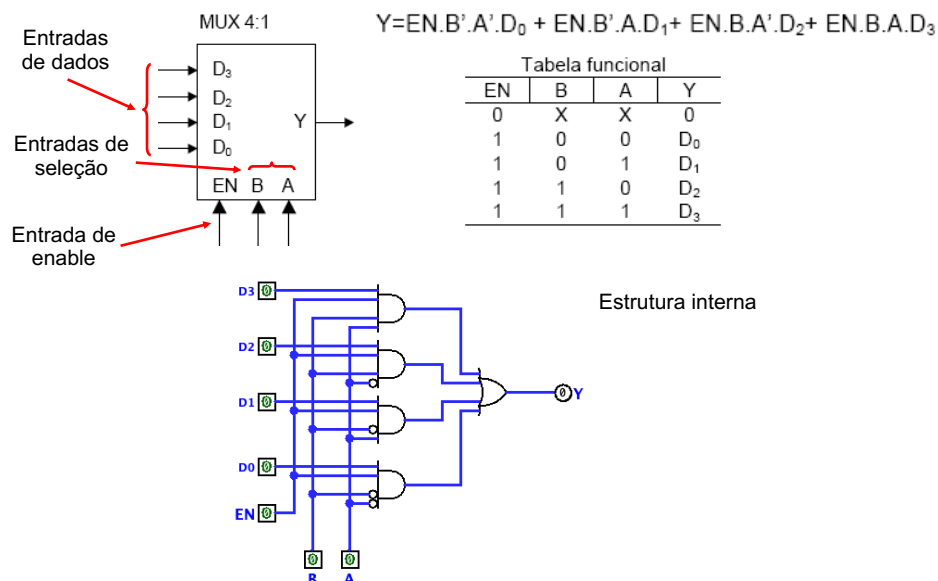
- ♦ Um multiplexer ou seletor de dados é um circuito que permite selecionar uma de entre várias entradas (entradas de dados) como fonte de informação para uma única saída.
- ♦ A seleção de uma das entradas de dados é controlada por um conjunto de entradas adicionais (entradas de seleção). Uma das  $2^N$  entradas de dados é selecionada à custa de N entradas de seleção (MUX  $2^N:1$ ).
- ♦ Os multiplexers podem ter uma entrada adicional de “enable” (EN) ou “strobe” (G) que permite controlar (ativar/desativar) a função de multiplexagem.

## Multiplexer 2:1 (MUX 2:1)



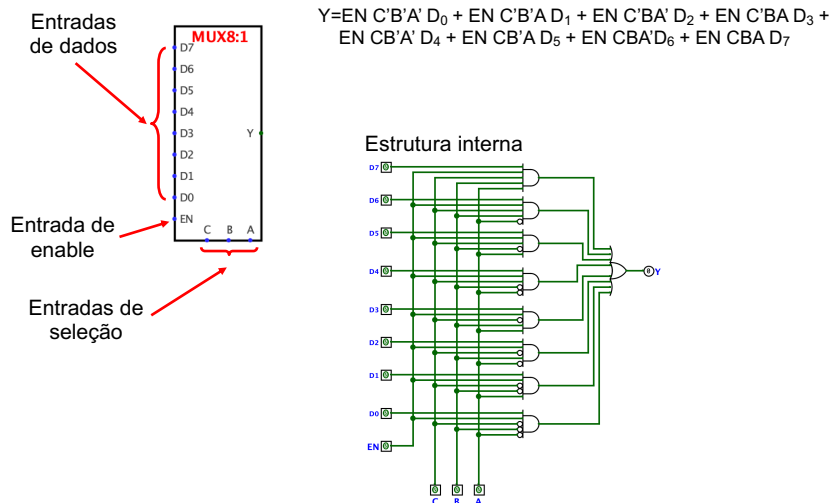
## Circuitos MSI/LSI

## Multiplexer 4:1 (MUX 4:1), com entrada adicional de “enable” (EN)



## Circuitos MSI/LSI

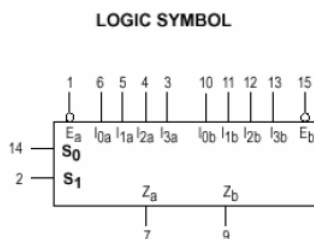
Multiplexer 8:1 (MUX 8:1), com entrada adicional de “enable” (EN)



## Circuitos MSI/LSI

54/74/XXX153 Dual 4-Line to 1-Line Data Selectors/Multiplexers

- dois multiplexers de 4 para 1 (MUX 4:1);
- entradas de dados ( $I_{3a} \dots I_{0a}, I_{3b} \dots I_{0b}$ );
- entradas de seleção comuns ( $S_1, S_0$ );
- entradas de enable independentes ( $E'_a, E'_b$  ativas a zero);
- saídas não complementadas ( $Z_a, Z_b$ )



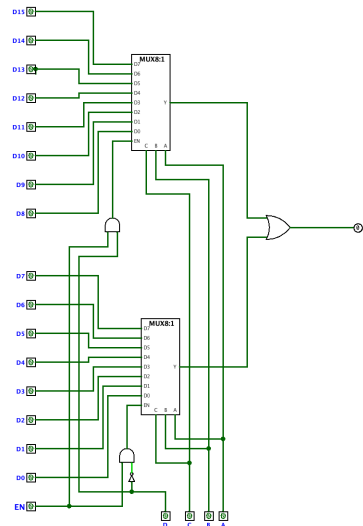
TRUTH TABLE

SELECT INPUTS		INPUTS (a or b)					OUTPUT
S <sub>0</sub>	S <sub>1</sub>	E	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	Z
X	X	H	X	X	X	X	L
L	L	L	L	X	X	X	L
L	L	L	H	X	X	X	H
H	L	L	X	L	X	X	L
H	L	L	X	H	X	X	H
L	H	L	X	X	L	X	L
L	H	L	X	X	H	X	H
H	H	L	X	X	X	L	L
H	H	L	X	X	X	H	H

## Circuitos MSI/LSI

## Expansão de Multiplexers

Exemplo: Mux 16:1 a partir de dois MUX 8:1 com entrada de enable (EN)



## Circuitos MSI/LSI

## Implementação de funções lógicas com Multiplexers

Considerando  $F(X,Y,Z) = \sum m(2,4,6,7) = \prod M(0,1,3,5)$ , fazendo corresponder as variáveis independentes X,Y,Z às entradas de seleção, as entradas de dados D2, D4, D6 e D7 devem ser ligadas a 1 e as entradas de dados D0, D1, D3 e D5 devem ser ligadas a 0.

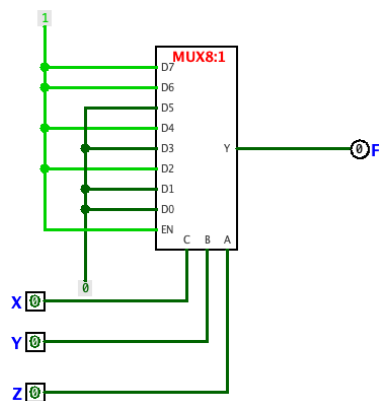


Tabela de verdade			
X	Y	Z	F(X,Y,Z)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

## Circuitos MSI/LSI

## Implementação de funções lógicas com Multiplexers

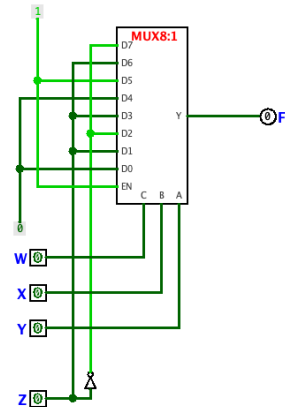
Considerando  $F(W,X,Y,Z) = \Sigma m(3,4,7,10,11,13,14)$  pode se obter uma tabela de verdade reduzida, exprimindo a saída em função de uma das variáveis de entrada.

Tabela de verdade

Nº	W	X	Y	Z	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	0

Tabela de verdade reduzida

Nº	W	X	Y	F
0	0	0	0	0
1	0	0	1	Z
2	0	1	0	Z'
3	0	1	1	Z
4	1	0	0	0
5	1	0	1	1
6	1	1	0	Z
7	1	1	1	Z'

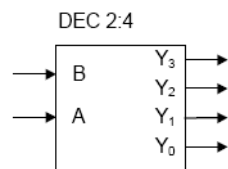


## Circuitos MSI/LSI

## Descodificadores/Demultiplexers

- ◆ Um decodificador é um circuito cujas saídas são ativadas seletivamente por combinações de valores lógicos aplicados nas entradas.
- ◆ Um decodificador com N variáveis de entrada tem  $2^N$  saídas.
- ◆ Os decodificadores podem ter uma entrada adicional de "enable" (EN) ou "strobe" (G) que permite controlar (ativar/desativar) a função de decodificação.

## Descodificador 2:4 (DEC 2:4)



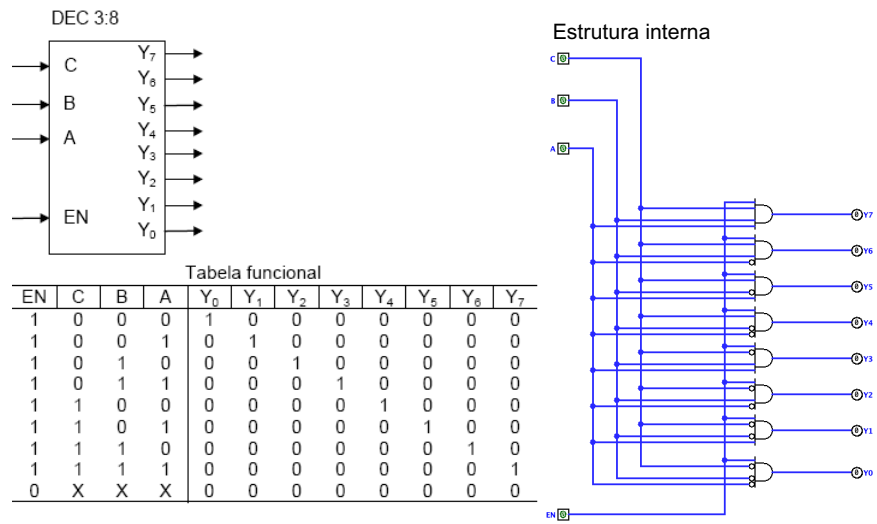
$$Y_0 = B'A'; Y_1 = B'A; Y_2 = BA'; Y_3 = BA$$

Tabela funcional

B	A	$Y_0$	$Y_1$	$Y_2$	$Y_3$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

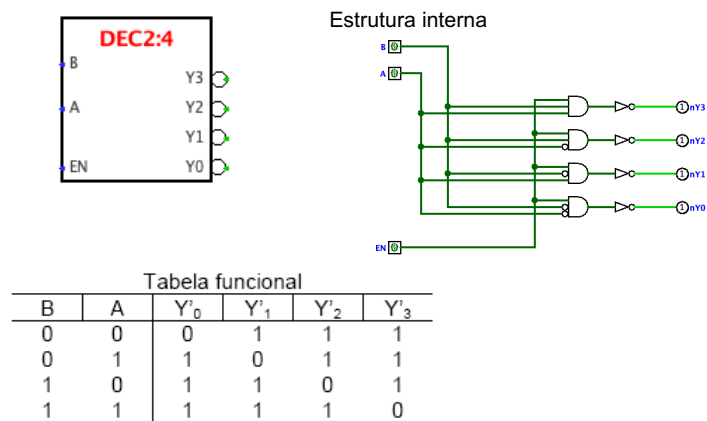
## Circuitos MSI/LSI

## Descodificador 3:8 (DEC 3:8)



## Circuitos MSI/LSI

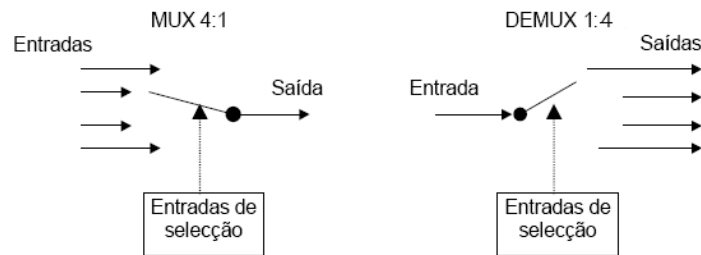
## Descodificador 2:4 (DEC 2:4) com saídas ativas a zero



## Circuitos MSI/LSI

## Demultiplexers

- ◆ Um demultiplexer é um circuito que permite encaminhar seletivamente a informação aplicada numa única entrada para uma de várias saídas.
- ◆ Um demultiplexer de 1 para  $2^N$  tem uma entrada,  $2^N$  saídas e N entradas de seleção.
- ◆ Realiza a função inversa de um multiplexer.



## Circuitos MSI/LSI

## Demultiplexers de 1:8 (DEMUX 1:8)

Um demultiplexer de 1 para 8, designado por DEMUX 1:8, permite o encaminhamento seletivo da informação aplicada na entrada para uma de entre as 8 saídas. A seleção da saída é feita por intermédio de 3 entradas de seleção.

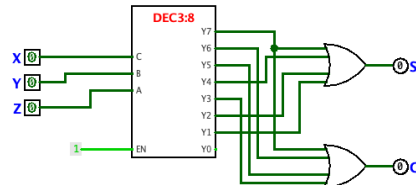
DEMUX 1:8			Tabela funcional							
C	B	A	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>
0	0	0	IN	0	0	0	0	0	0	0
0	0	1	0	IN	0	0	0	0	0	0
0	1	0	0	0	IN	0	0	0	0	0
0	1	1	0	0	0	IN	0	0	0	0
1	0	0	0	0	0	0	IN	0	0	0
1	0	1	0	0	0	0	0	IN	0	0
1	1	0	0	0	0	0	0	0	IN	0
1	1	1	0	0	0	0	0	0	0	IN

Se compararmos a tabela funcional de um DEMUX 1:8 com a tabela funcional de um DEC 3:8 com entrada de enable, verifica-se que estas são idênticas se se fizer coincidir a entrada de enable (EN) do DEC 3:8 com a entrada (IN) do DEMUX 1:8. Esta é a solução adotada nos dispositivos disponíveis no mercado, justamente designados por descodificadores/demultiplexers.

## Circuitos MSI/LSI

## Implementação de funções combinacionais com decodificadores

Um decodificador disponibiliza nas suas saídas os  $2^N$  mintermos de uma função de N variáveis.



Exemplo:

$$S(X,Y,Z) = \sum m(1,2,4,7)$$

$$C(X,Y,Z) = \sum m(3,5,6,7)$$

Se o número de mintermos da função a implementar é superior a  $2^{N/2}$ , então a função complementada é expressa por um menor número de mintermos, sendo vantajosa a utilização de uma porta NOR.



Exemplo:

$$F(X,Y,Z) = \sum m(0,1,2,3,5,7) = \prod M(4,6)$$

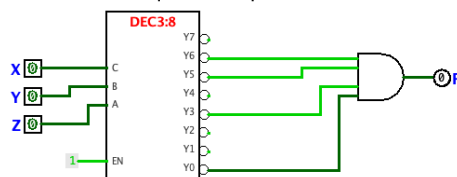
$$F'(X,Y,Z) = \sum m(4,6)$$

$$F''(X,Y,Z) = (\sum m(4,6))'$$

## Circuitos MSI/LSI

## Implementação de funções combinacionais com decodificadores

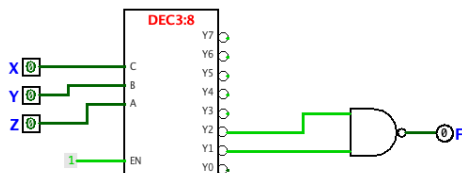
Para decodificadores construídos com portas NAND, ou seja, que apresentam as saídas activas a zero, deve ser utilizada uma porta AND com entradas ligadas às saídas do decodificador que correspondem aos maxtermos da função.



Exemplo:

$$F(X,Y,Z) = \sum m(1,2,4,7) = \prod M(0,3,5,6)$$

Se o número de maxtermos da função for superior a  $2^{N/2}$ , então a função complementada é expressa por um menor número de maxtermos, sendo vantajosa a utilização de uma porta NAND.



Exemplo:

$$F(X,Y,Z) = \prod M(0,3,4,5,6,7) = \sum m(1,2)$$

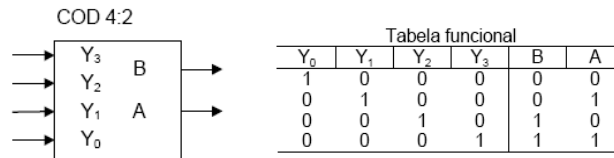
$$F'(X,Y,Z) = \prod M(1,2)$$

$$F''(X,Y,Z) = (\prod M(1,2))'$$

## Circuitos MSI/LSI

## Codificadores

- ◆ Um codificador é um circuito que efetua a operação inversa de um decodificador, ou seja, representa nas suas saídas o código binário do número da entrada que está ativa.
- ◆ Um codificador com  $2^N$  entradas tem N variáveis de saída.

Codificador **hipotético** de 4 para 2

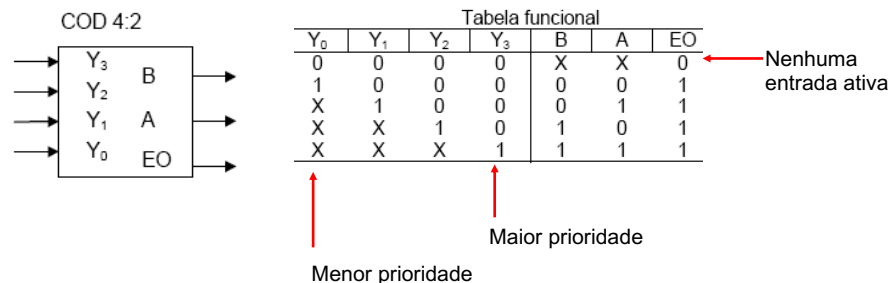
No entanto, se se observar a tabela funcional, verifica-se que não estão representadas todas as combinações possíveis para as entradas  $Y_3$ ,  $Y_2$ ,  $Y_1$ ,  $Y_0$ . Qual o valor das saídas B e A para, por exemplo,  $Y_3=1$ ,  $Y_2=0$ ,  $Y_1=1$ ,  $Y_0=0$ ? Nestas situações, deve ser considerado um sistema de prioridades. Quando mais do que uma entrada está ativa, as saídas representam o código binário do número da entrada com maior prioridade. A este tipo de circuito dá-se o nome de codificador de prioridade.

## Circuitos MSI/LSI

## Codificadores de prioridades

- ◆ Um codificador de prioridade é um circuito que representa nas saídas o código binário do número da entrada ativa com maior prioridade.
- ◆ Um codificador de prioridade com  $2^N$  entradas tem N variáveis de saída.
- ◆ Os codificadores de prioridade podem ter uma entrada adicional de “enable” (EN) ou “strobe” (G) que permite controlar (ativar/desativar) a função de codificação.
- ◆ Os codificadores de prioridade têm, normalmente, uma saída adicional que permite detetar a situação em que nenhuma das entradas está ativa.

## Codificador de prioridade de 4 para 2 (COD 4:2)



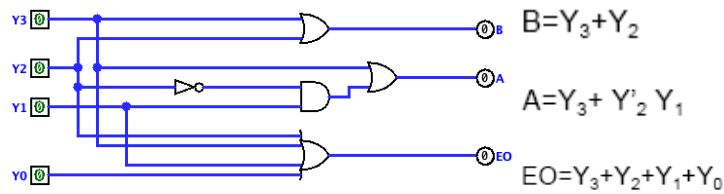
## Circuitos MSI/LSI

## Codificador de prioridade de 4 para 2 (COD 4:2)

Tabela funcional

$Y_0$	$Y_1$	$Y_2$	$Y_3$	B	A	EO
0	0	0	0	X	X	0
1	0	0	0	0	0	1
X	1	0	0	0	1	1
X	X	1	0	1	0	1
X	X	X	1	1	1	1

## Estrutura interna



## Circuitos MSI/LSI

## Comparadores

Permite determinar se dois números binários são iguais, e não o sendo, qual deles é o maior.

## Comparador de 2 bits

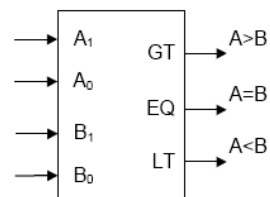


Tabela funcional

$A_1$	$B_1$	$A_0$	$B_0$	GT ( $A > B$ )	EQ ( $A = B$ )	LT ( $A < B$ )
$A_1 > B_1$	X	X	X	1	0	0
$A_1 < B_1$	X	X	X	0	0	1
$A_1 = B_1$	$A_0 > B_0$			1	0	0
$A_1 = B_1$	$A_0 < B_0$			0	0	1
$A_1 = B_1$	$A_0 = B_0$			0	1	0

GT( $A > B$ )=1 qd

$$\begin{aligned}
 &\Leftrightarrow (A_1 > B_1) \text{ ou } (A_1 = B_1 \text{ e } A_0 > B_0) \\
 &\Leftrightarrow (A_1 B_1') + ((A_1 \oplus B_1)' \cdot (A_0 B_0')) \\
 &\Leftrightarrow (A_1 B_1') + ((A_1 \oplus B_1)' \cdot (A_0 B_0'))
 \end{aligned}$$

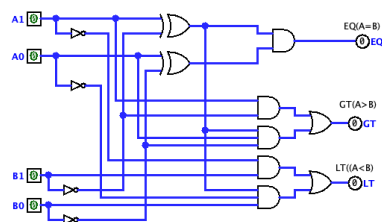
EQ( $A = B$ )=1 qd

$$\begin{aligned}
 &\Leftrightarrow (A_1 = B_1) \text{ e } (A_0 = B_0) \\
 &\Leftrightarrow (A_1 \oplus B_1)' \cdot (A_0 \oplus B_0)' \\
 &\Leftrightarrow (A_1 \oplus B_1)' \cdot (A_0 \oplus B_0)'
 \end{aligned}$$

LT( $A < B$ )=1 qd

$$\begin{aligned}
 &\Leftrightarrow (A_1 < B_1) \text{ ou } (A_1 = B_1 \text{ e } A_0 < B_0) \\
 &\Leftrightarrow (A_1' B_1) + ((A_1 \oplus B_1)' \cdot (A_0' B_0)) \\
 &\Leftrightarrow (A_1' B_1) + ((A_1 \oplus B_1)' \cdot (A_0' B_0))
 \end{aligned}$$

## Estrutura interna



## Circuitos MSI/LSI

## Adicionadores

Exemplo:

$$A = 11_{(10)} = 1011_{(2)}$$

$$B = 14_{(10)} = 1110_{(2)}$$

$$\begin{array}{r}
 \leftarrow 1 \leftarrow \quad \leftarrow 1 \leftarrow \quad \leftarrow 1 \leftarrow \quad \leftarrow 0 \leftarrow \quad \leftarrow \text{Transporte ("Carry")} \\
 \quad \quad \quad 1 \quad \quad 0 \quad \quad 1 \quad \quad 1 \quad \quad (11) \\
 + \quad \quad \quad 1 \quad \quad 1 \quad \quad 1 \quad \quad 0 \quad \quad (14) \\
 \hline
 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad (25)
 \end{array}$$

## Meio adicionador ("half-adder")

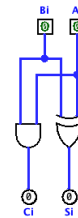
Um circuito meio adicionador realiza a soma de dois números de um bit ( $A_i$ ,  $B_i$ ), produzindo, tal como acontece na soma com números decimais, um eventual transporte a ser considerado na posição seguinte ( $C_{i+1}$ ).

Tabela funcional

$A_i$	$B_i$	$S_i$	$C_{i+1}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S_i = A_i \oplus B_i$$

$$C_{i+1} = A_i \cdot B_i$$



## Circuitos MSI/LSI

## Adicionador completo ("full-adder")

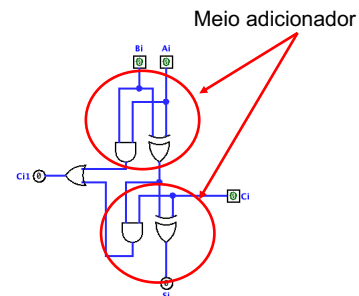
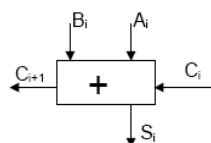
Um circuito adicionador completo realiza a soma de dois números de um bit ( $A_i$ ,  $B_i$ ), considerando um transporte ( $C_i$ ) de posições anteriores e produzindo um transporte a ser considerado na posição seguinte ( $C_{i+1}$ ).

Tabela funcional

$C_i$	$A_i$	$B_i$	$S_i$	$C_{i+1}$
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

$$S_i = A_i \oplus B_i \oplus C_i$$

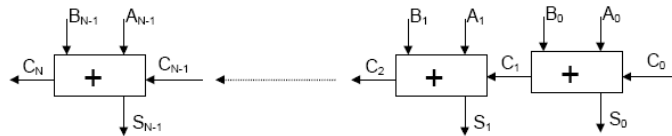
$$C_{i+1} = A_i \cdot B_i + C_i (A_i \oplus B_i)$$



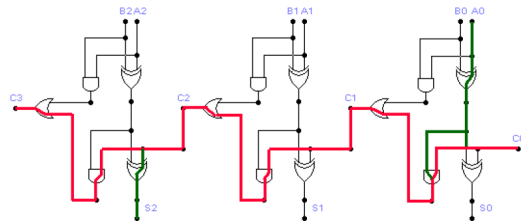
## Circuitos MSI/LSI

## Adicionador de N bits, em cascata ("ripple adder")

Com base nos adicionadores completos, é possível construir um circuito adicionador de N bits, colocando-os em cascata ("ripple adder")



Qualquer alteração nas entradas  $B_0$ ,  $A_0$  ou  $C_0$  provoca a propagação dos transportes  $C_1 \dots C_{N-1}$  ao longo da cascata. Esta solução caracteriza-se por apresentar um tempo de propagação elevado e variável em função do número de bits.

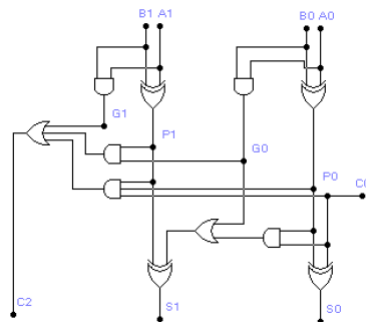


## Circuitos MSI/LSI

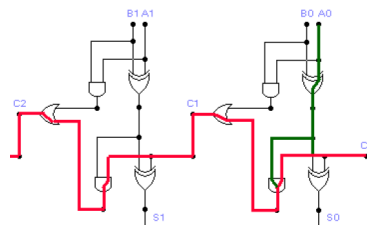
## Adicionador de N Bits, com transporte antecipado ("Carry look-ahead"/"Fast carry")

Os sinais de saída,  $S_i$  e  $C_{i+1}$ , dependem directamente dos sinais  $A_i$ ,  $B_i$ ,  $P_i$ ,  $G_i$  e  $C_0$ , onde  $P_i$  e  $G_i$  dependem directamente de  $A_i$  e  $B_i$ . Assim sendo, o tempo de propagação do adicionador é constante e independente do número de bits do adicionador. Claro que isto tem um custo, o significativo aumento da complexidade do circuito (número de portas lógicas necessárias para determinar os transportes antecipados) à medida que o número de bits do adicionador aumenta.

## Estrutura interna de um adicionador de 2 bits com transporte antecipado



## Cascata (sem transporte antecipado)



## Circuitos MSI/LSI

## Multiplicadores

Existem várias soluções para a construção de circuitos multiplicadores, nomeadamente no que se refere ao tipo de lógica utilizada, combinacional ou sequencial. Apenas se analisa um circuito multiplicador que tem por base circuitos adicionadores e alguma lógica combinacional.

## Multiplicador de números de 2 bits

