

# Sistemas Digitais

## Circuitos Aritméticos e Unidades Aritméticas e Lógicas (ALUs)

João Paulo Baptista de Carvalho

[joao.carvalho@inesc-id.pt](mailto:joao.carvalho@inesc-id.pt)



TÉCNICO  
LISBOA



# Circuitos Aritméticos

- ★ Circuitos aritméticos são aqueles que realizam operações aritméticas sobre números binários
- ★ O circuito aritmético mais simples é o que soma números de apenas 1 bit. Basta partir da conhecida tabela da soma para o obter:

X+Y	X=0	X=1
Y=0	0	1
Y=1	1	10



XY	Soma	Transporte
00	0	0
01	1	0
10	1	0
11	0	1

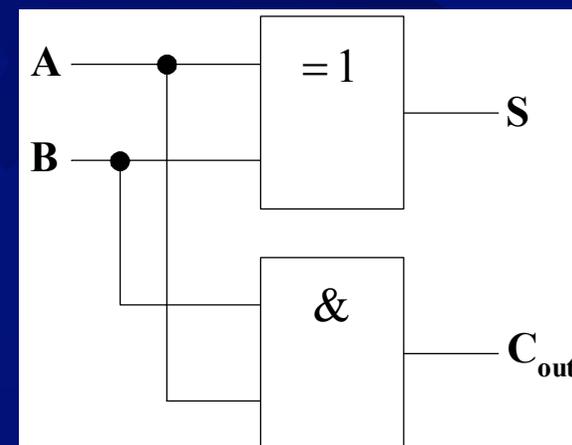


# Semi-somador (*half-adder*)

- Da tabela retira-se o seguinte circuito, que é conhecido como semi-somador:

AB	Soma (S)	Transporte ( $C_{out}$ )
00	0	0
01	1	0
10	1	0
11	0	1

$$C_{out} = A \cdot B$$
$$S = A \oplus B$$



- O nome semi-somador vem do facto de este circuito não permitir somar o transporte que venha de bits de menor peso

# Somador Completo (*full-adder*)

- ★ A um somador de 1 bit que tenha em conta o transporte de somas anteriores chama-se somador completo:

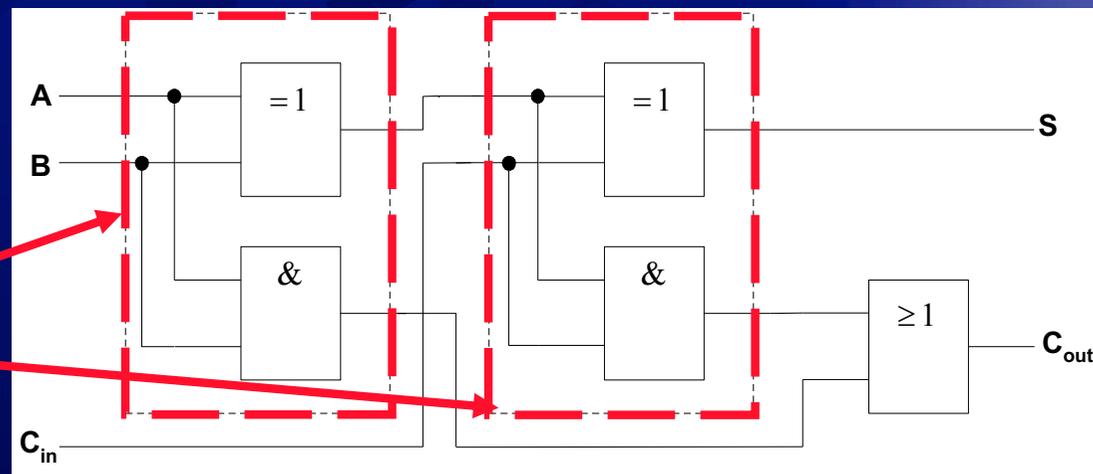
A	B	$C_{in}$	S	$C_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

		AB			
		00	01	11	10
$C_{in}$	0	0	1	0	1
	1	1	0	1	0

		AB			
		00	01	11	10
$C_{in}$	0	0	0	1	0
	1	0	1	1	1

$$\begin{aligned}
 S &= \bar{C}_{in} \bar{A} B + \bar{C}_{in} A \bar{B} \\
 &\quad + C_{in} \bar{A} \bar{B} + C_{in} A B \\
 &= A \oplus B \oplus C_{in}
 \end{aligned}$$

$$\begin{aligned}
 C_{out} &= AB + C_{in} A \bar{B} + C_{in} \bar{A} B \\
 &= AB + C_{in} (A \oplus B)
 \end{aligned}$$

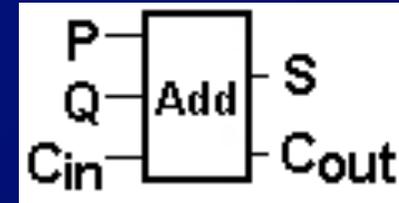


Semi-somadores

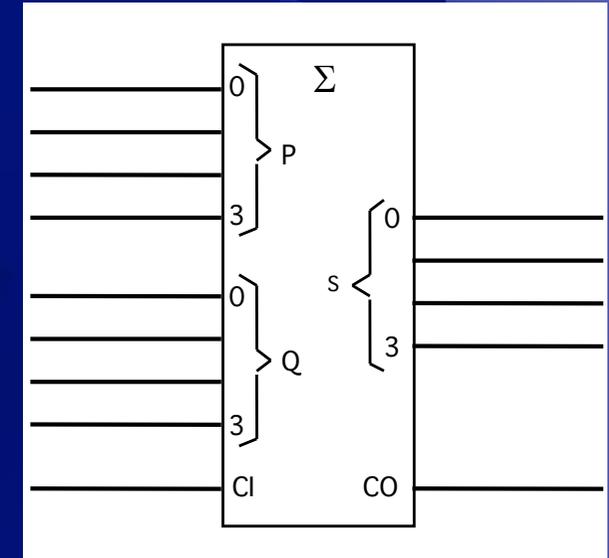
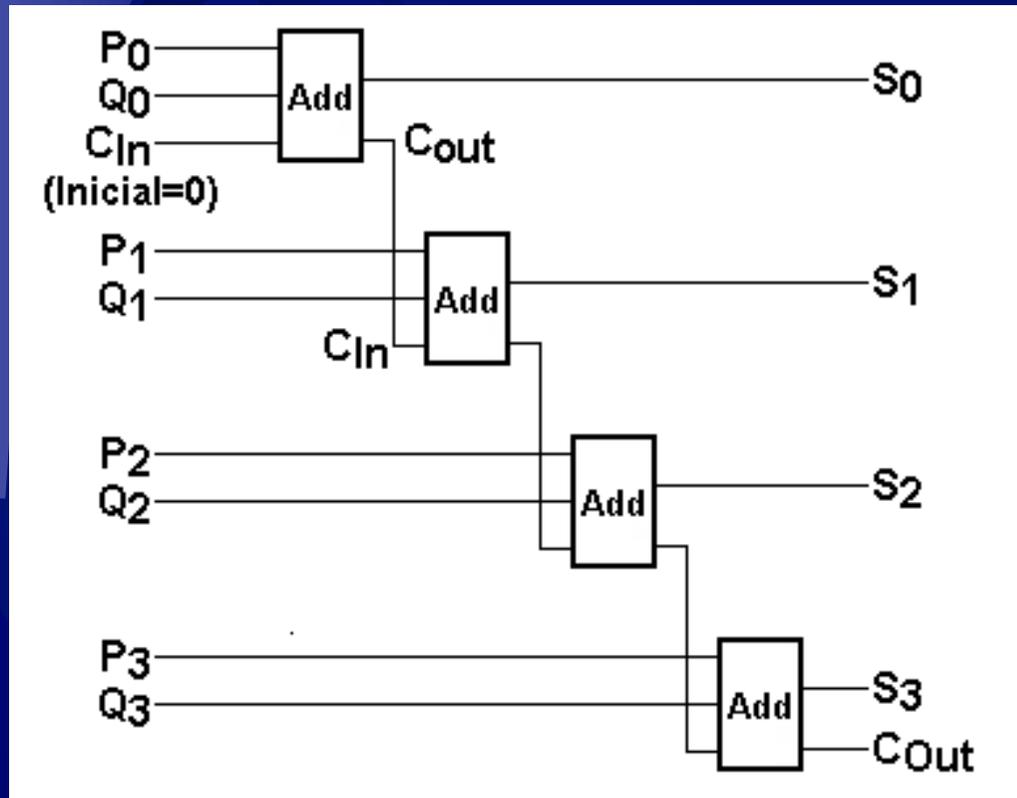


# Somadores

- Assumindo blocos de somadores completos, é possível construir somadores de n bits
- Exemplo: Um somador de 4 bits



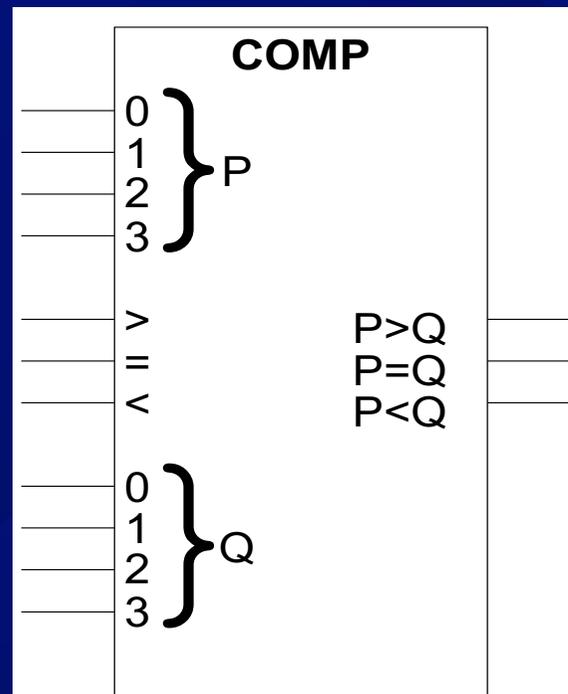
Símbolo de um Somador Completo de 4 bits:





# Comparadores

- Tal como o nome indica, um comparador permite comparar 2 números binários  $P$  e  $Q$  de  $n$  bits, indicando se  $P > Q$ ,  $P < Q$  ou  $P = Q$ .
- Exemplo, um comparador de 4 bits:

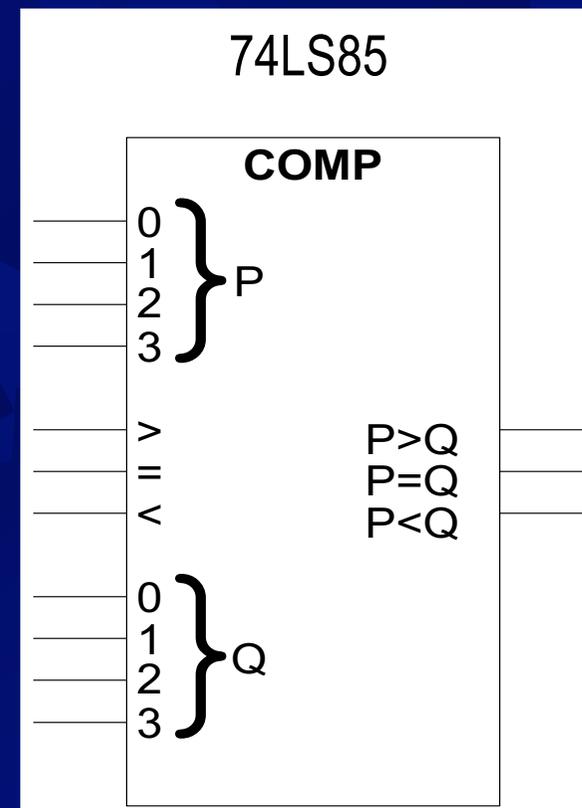
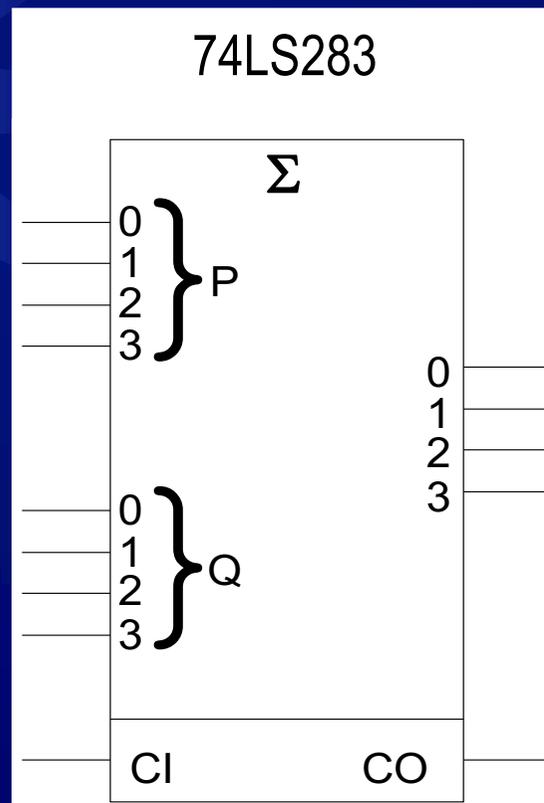


- O circuito pode ser ligado em cascata para realizar comparações entre números de  $n > 4$  bits (é essa a utilidade das entradas  $>$ ,  $=$ ,  $<$  presentes no lado esquerdo do circuito).



# Alguns CI's

- 74LS283: Somador de 4 bits rápido. Usa internamente um “carry look-ahead” – trata-se de um somador de maior rapidez, conseguida à custa de um maior número de portas lógicas.
- 74LS85: Comparador de 4 bits.



# Somas em BCD

- As operações aritméticas em BCD podem ser úteis quando se têm números em BCD sobre os quais se pretende fazer pequenas operações que não justificam a conversão para binário

- A soma em BCD é feita dígito a dígito. Por exemplo, a soma dos dígitos 3 e 5 dá o dígito 8:

$$\begin{array}{r} 0011 \\ 0101 \\ \hline 1000 \end{array}$$

As duas parcelas são BCD, assim como o resultado

- Mas se a soma for superior a 9, então o resultado obtido já não é BCD. Por exemplo,  $7 + 5 = 12$ :

$$\begin{array}{r} 0111 \\ 0101 \\ \hline 1100 \end{array}$$

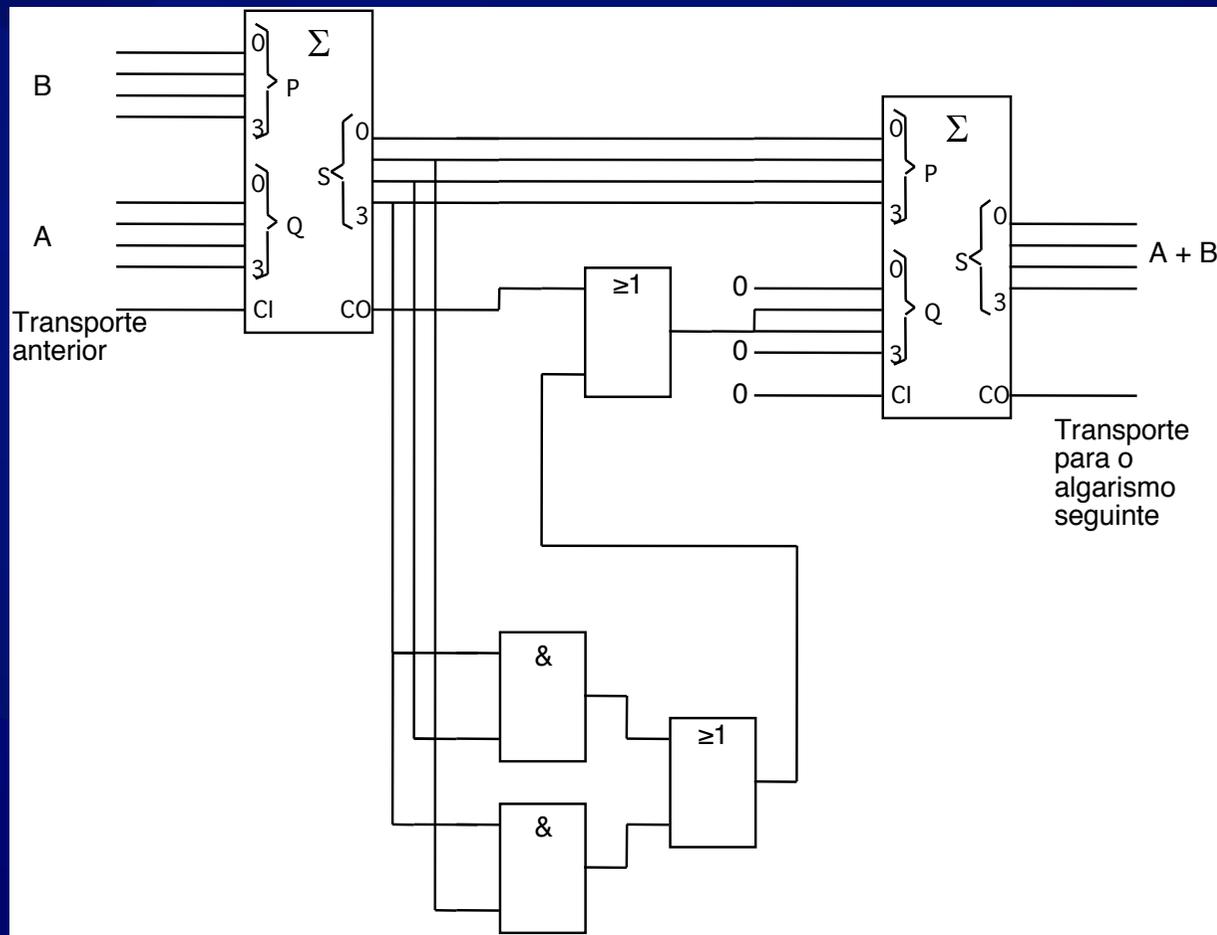
- E em certos casos o resultado é inválido e há transporte,  $8 + 9 = 17$ :

$$\begin{array}{r} 1000 \\ 1001 \\ \hline 10001 \end{array}$$



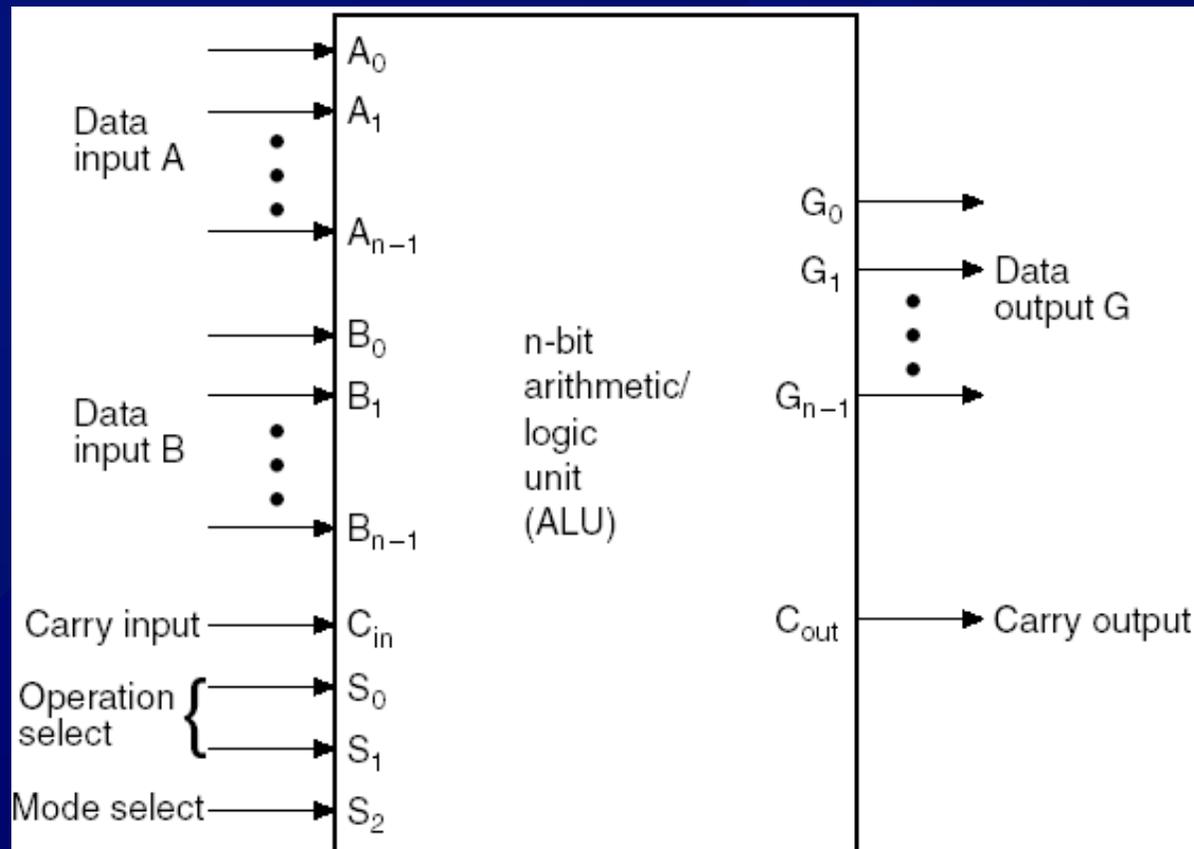
# Somas em BCD (II)

- Consegue-se corrigir o resultado obtido somando 6 (0110) ao resultado nas seguintes situações:
  - O resultado é maior que 9
  - Existe um transporte na soma



# ALU – Unidade Lógica e Aritmética

- ★ Uma ALU (Arithmetic Logic Unit) é um circuito digital que efectua operações aritméticas e lógicas com números inteiros
- ★ É um dos blocos fundamentais da CPU de um computador



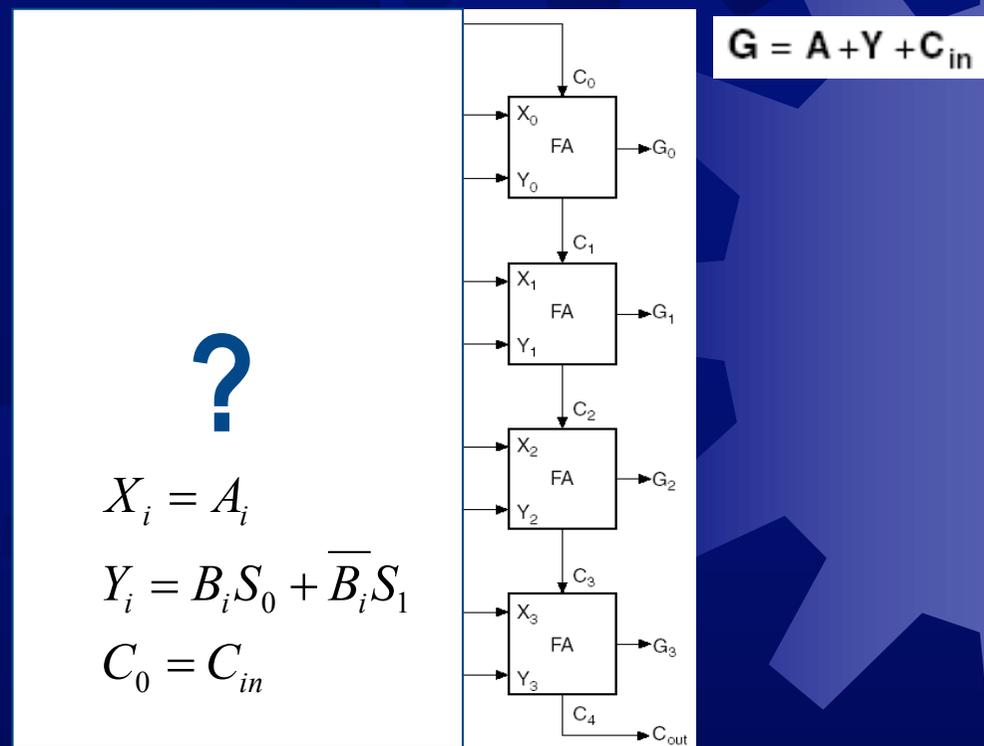
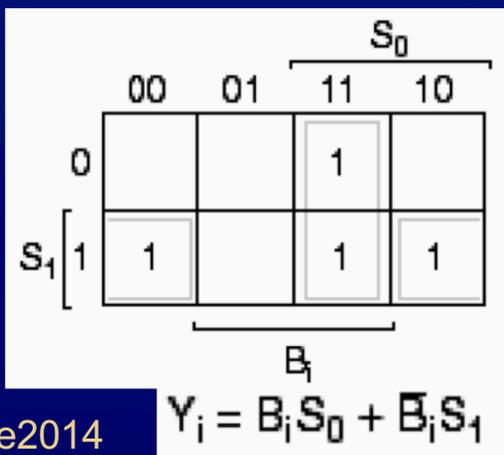


# ALU – Unidade Lógica e Aritmética

## Unidade Aritmética (Implementação)

Inputs			Output	
S <sub>1</sub>	S <sub>0</sub>	B <sub>i</sub>	Y <sub>i</sub>	
0	0	0	0	Y <sub>i</sub> = 0
0	0	1	0	
0	1	0	0	Y <sub>i</sub> = B <sub>i</sub>
0	1	1	1	
1	0	0	1	Y <sub>i</sub> = $\overline{B}_i$
1	0	1	0	
1	1	0	1	Y <sub>i</sub> = 1
1	1	1	1	

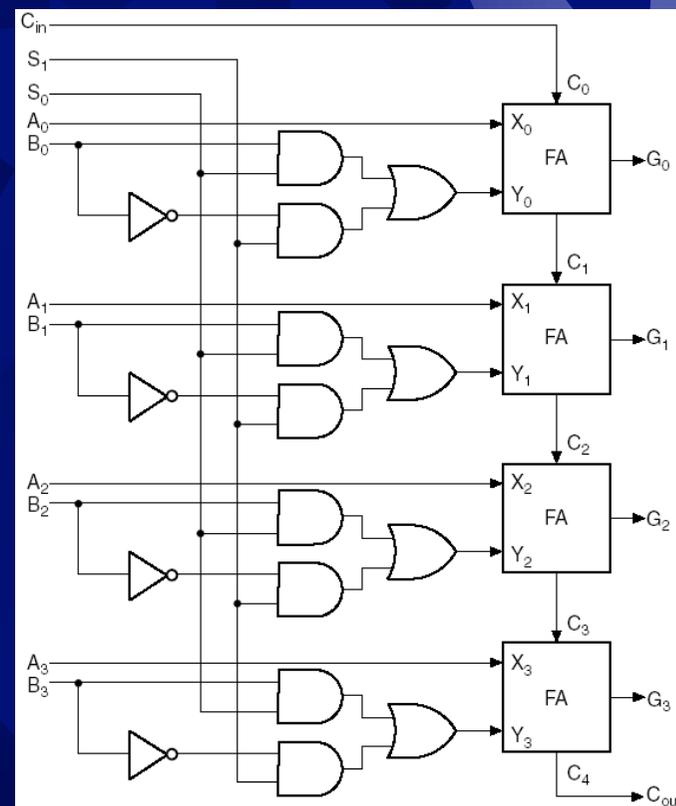
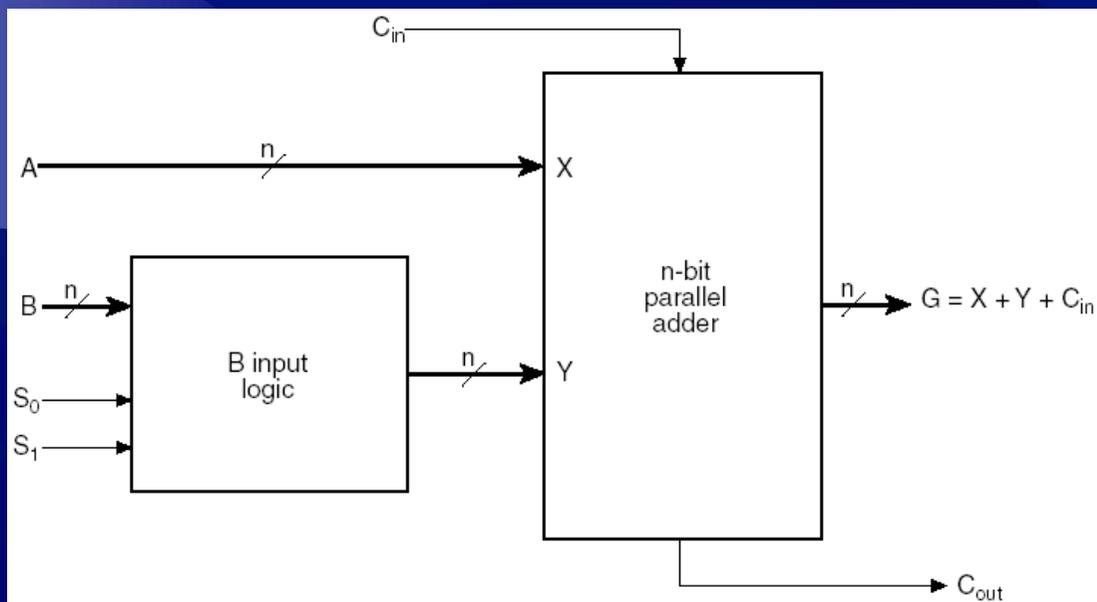
Select		Input	G = A + Y + C <sub>in</sub>	
S <sub>1</sub>	S <sub>0</sub>	Y	C <sub>in</sub> = 0	C <sub>in</sub> = 1
0	0	all 0's	G = A (transfer)	G = A + 1 (increment)
0	1	B	G = A + B (add)	G = A + B + 1
1	0	$\overline{B}$	G = A + $\overline{B}$	G = A + $\overline{B}$ + 1 (subtract)
1	1	all 1's	G = A - 1 (decrement)	G = A (transfer)



# ALU – Unidade Lógica e Aritmética

## Unidade Aritmética (logograma)

Select		Input	$G = A + Y + C_{in}$	
$S_1$	$S_0$	$Y$	$C_{in} = 0$	$C_{in} = 1$
0	0	all 0's	$G = A$ (transfer)	$G = A + 1$ (increment)
0	1	$B$	$G = A + B$ (add)	$G = A + B + 1$
1	0	$\overline{B}$	$G = A + \overline{B}$	$G = A + \overline{B} + 1$ (subtract)
1	1	all 1's	$G = A - 1$ (decrement)	$G = A$ (transfer)



# ALU – Unidade Lógica e Aritmética

## Unidade Lógica (Exemplo)

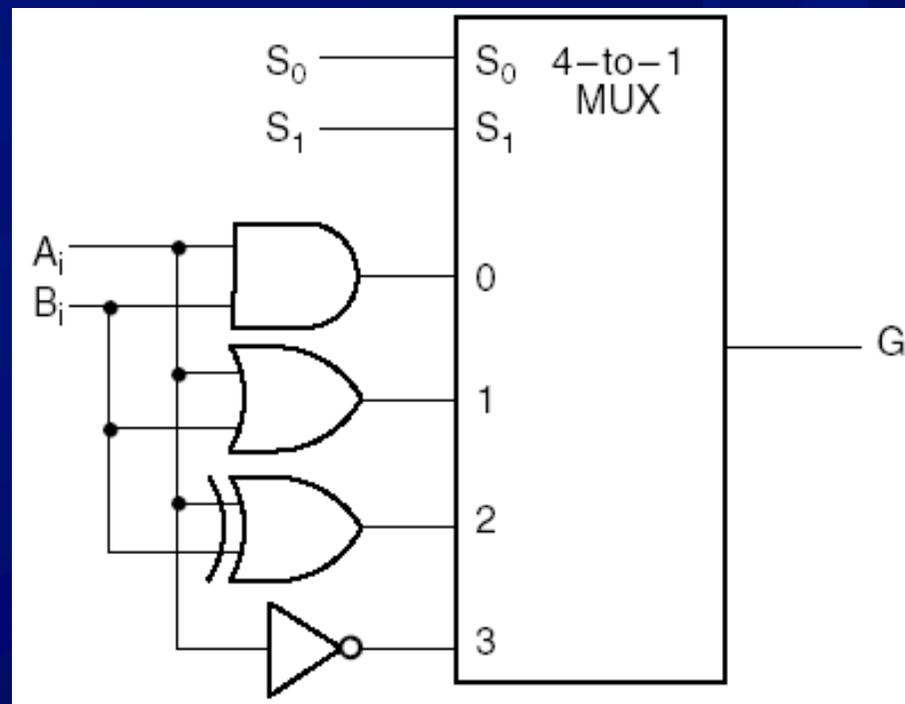
AND

OR

XOR

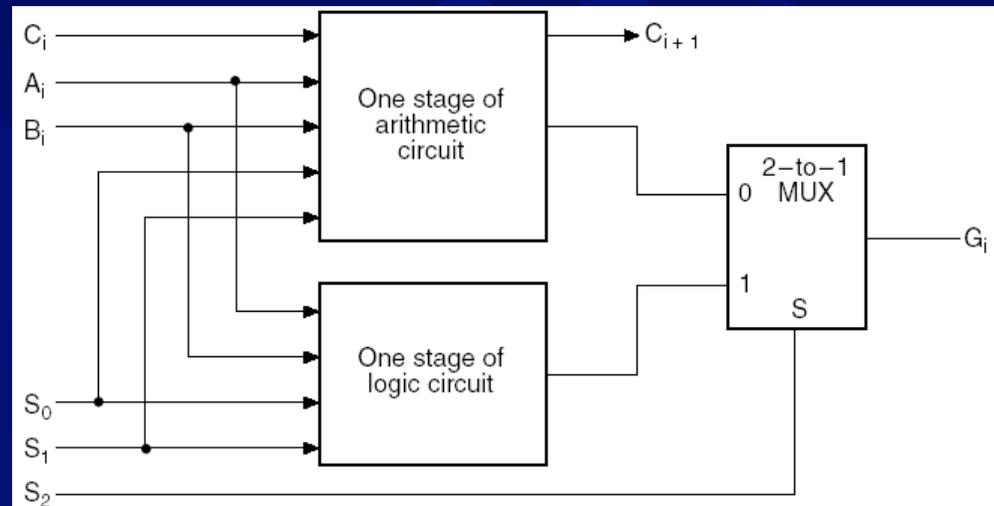
NOT

$S_1$	$S_0$	Output	Operation
0	0	$G = A \wedge B$	AND
0	1	$G = A \vee B$	OR
1	0	$G = A \oplus B$	XOR
1	1	$G = \bar{A}$	NOT



# ALU – Unidade Lógica e Aritmética

☀ Exemplo (completo):



Operation Select				Operation	Function
S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	C <sub>in</sub>		
0	0	0	0	$G = A$	Transfer $A$
0	0	0	1	$G = A + 1$	Increment $A$
0	0	1	0	$G = A + B$	Addition
0	0	1	1	$G = A + \underline{B} + 1$	Add with carry input of 1
0	1	0	0	$G = A + \underline{B}$	$A$ plus 1's complement of $B$
0	1	0	1	$G = A + \underline{B} + 1$	Subtraction
0	1	1	0	$G = A - 1$	Decrement $A$
0	1	1	1	$G = A$	Transfer $A$
1	0	0	X	$G = A \wedge B$	AND
1	0	1	X	$G = A \vee B$	OR
1	1	0	X	$G = \underline{A} \oplus B$	XOR
1	1	1	X	$G = A$	NOT (1's complement)



# Bibliografia

- ★ Arroz,G., Monteiro,J.C., Oliveira,A.,  
“Arquitectura de Computadores, dos  
Sistemas Digitais aos Microprocessadores”,  
Secções 5.1, 5.2, 5.3, 2ª Edição, 2009
- ★ Mano,M., Kime,C. – “Logic and Computer  
Design Fundamentals”, Prentice Hall, secções  
3.8 a 3.10 e 3.13



Considerere um guindaste usado para carregar cascalho. O controlo do guindaste é feito por um conjunto de 15 válvulas hidráulicas:

- ✱ 4 para elevar ou baixar a “lança” (2 velocidades possíveis p/ cada);
- ✱ 4 para rodar a “lança”, para a esquerda ou para a direita (2 velocidades possíveis p/ cada);
- ✱ 4 para enrolar ou desenrolar o cabo (2 velocidades possíveis p/ cada);
- ✱ 2 para abrir o balde (2 velocidades possíveis);
- ✱ 1 para fechar o balde.

Pretende-se um dispositivo de controlo remoto, ligado ao guindaste por um cabo multi-condutor, que deverá ter 15 botões associados às 15 válvulas supracitadas e um botão “stop” (significando nada a fazer); Para ser leve e fácil de transportar, o cabo deverá ter o menor número de condutores.

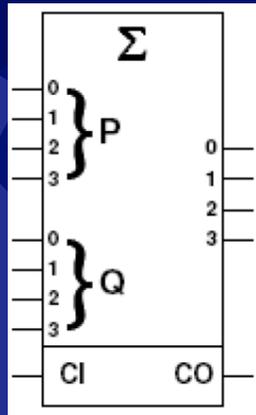
Nota importante: pressupõe-se que a cada momento só pode ser accionada uma válvula hidráulica – e, em consequência, a cada instante estará premido um e um só botão, nem que seja o botão stop...



- ✦ Pretende-se projectar um circuito que receba dois números de 4 bits,  $A_3A_2A_1A_0$  e  $B_3B_2B_1B_0$ , e apresente na saída o maior desses 2 números também em 4 bits ( $S_3, S_2, S_1, S_0$ ).
- ✦ Projecte um circuito que transforme um número binário de 4 bits num número representado em BCD.



- Considerere o circuito somador da figura abaixo. Desenhe o logograma de um circuito que realiza em hardware a operação  $-37 - 5$  utilizando os somadores que achar necessário. Indique quais os valores lógicos na saída do circuito.



- Qual a expressão lógica da função implementada pelo seguinte circuito:

