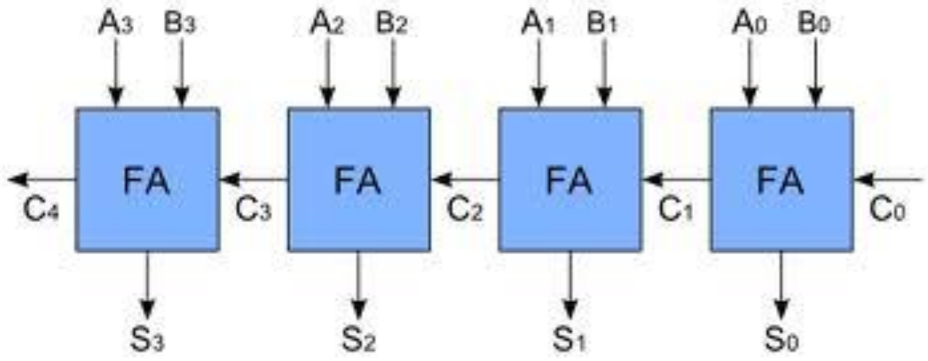
Sistemas Digitais (SD)

Circuitos combinatórios: somadores, subtractores e comparadores



Aula Anterior

- **Na aula anterior:**
  - Circuitos combinatórios típicos:
    - Descodificadores
    - Codificadores
    - Multiplexers
    - Demultiplexers

Planeamento

SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
17/Fev a 21/Fev	Introdução	Sistemas de Numeração	
24/Fev a 28/Fev	CARNAVAL	Álgebra de Boole	P0
02/Mar a 06/Mar	Elementos de Tecnologia	Funções Lógicas	VHDL
9/Mar a 13/Mar	Minimização de Funções	Minimização de Funções	L0
16/Mar a 20/Mar	Def. Circuito Combinatório; Análise Temporal	Circuitos Combinatórios	P1
23/Mar a 27/Mar	Circuitos Combinatórios	Circuitos Combinatórios	**L1**
30/Mar a 03/Abr	Circuitos Sequenciais: Latches	Circuitos Sequenciais: Flip-Flops	P2
06/Abr a 10/Abr	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA	FÉRIAS DA PÁSCOA
13/Abr a 17/Abr	Caracterização Temporal	Registos	L2
20/Abr a 24/Abr	Contadores	Circuitos Sequenciais Síncronos	P3
27/Abr a 01/Mai	Síntese de Circuitos Sequenciais Síncronos	Síntese de Circuitos Sequenciais Síncronos	L3
04/Mai a 08/Mai	Exercícios	Teste 1 Memórias	P4
11/Mai a 15/Mai	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Microprograma	L4
18/Mai a 22/Mai	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	P5
25/Mai a 29/Mai	P6	P6	L5

TÉCNICO LISBOA

Sumário

- **Tema da aula de hoje:**
  - Circuitos combinatórios típicos:
    - Somadores / Subtractores
    - Comparadores

- **Bibliografia:**
  - **M. Mano, C. Kime**: Secções 4.2 a 4.4
  - **G. Arroz, J. Monteiro, A. Oliveira**: Secções 5.1 a 5.3

Somadores

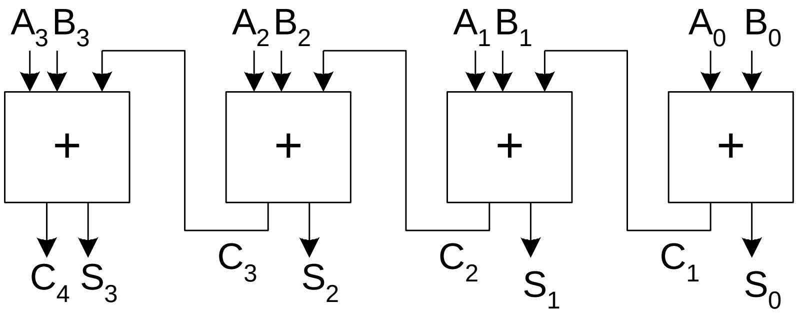
Circuito para soma aritmética

- Exemplo: Somador de 2 números de 4 bits cada.

		0	1	1	0	← Carry
7			0	1	1	1
+ 2		+	0	0	1	0
9			1	0	0	1

- A estrutura mais simples resolve 1 bit de cada vez:

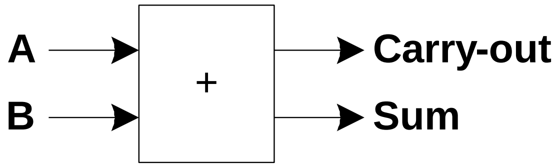
C4	C3	C2	C1	
	A3	A2	A1	A0
+	B3	B2	B1	B0
C4	S3	S2	S1	S0



Somadores

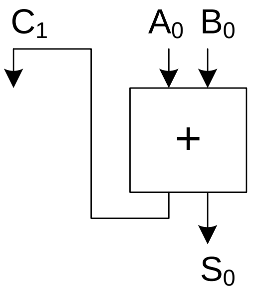
Circuito semi-somador

- O circuito semi-somador (em inglês, half-adder) soma 2 bits de entrada (sem transporte anterior) e produz 1 bit da soma e 1 bit de transporte.



- Corresponde p.ex. ao 1º passo do algoritmo de soma: soma os 2 bits de menor peso e obtém 1 bit S0 da soma e o transporte C1 para o passo seguinte.

C4	C3	C2	C1	
	A3	A2	A1	A0
+	B3	B2	B1	B0
C4	S3	S2	S1	S0



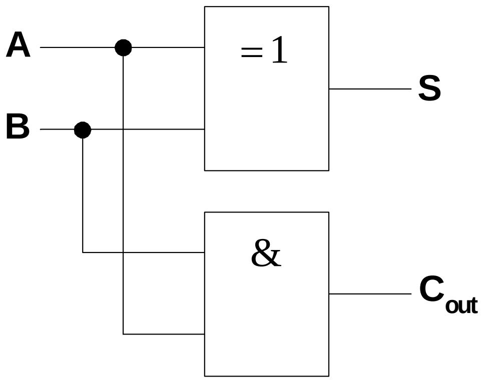
Somadores

- **Circuito semi-somador**

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

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

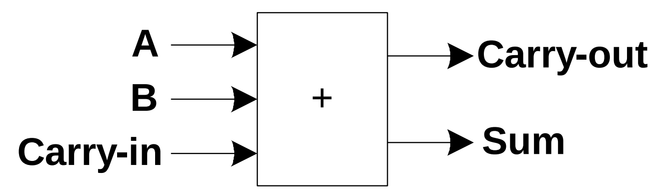
$$S = A \oplus B$$



Somadores

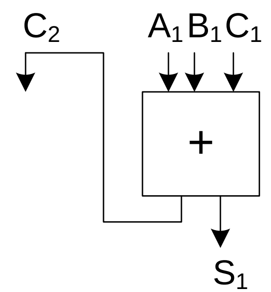
Circuito somador completo

- O circuito somador completo (em inglês, full-adder) soma 3 bits de entrada (incluindo o transporte anterior) e produz 1 bit da soma e 1 bit de transporte.



- P.ex. no 2º passo: soma 3 bits A1 e B1 e o transporte C1 do passo anterior, e obtém 1 bit S1 da soma e o transporte C2 para o passo seguinte.

C4	C3	C2	C1	
	A3	A2	A1	A0
+	B3	B2	B1	B0
C4	S3	S2	S1	S0



Somadores

Somador completo

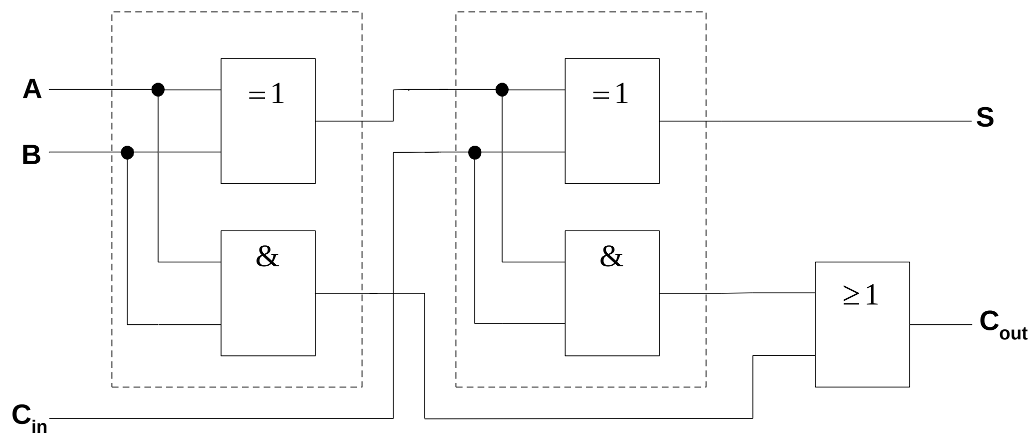
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

C_{in} \ A B	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = \overline{C}_{in} \cdot \overline{A} \cdot B + \overline{C}_{in} \cdot A \cdot \overline{B} + C_{in} \cdot \overline{A} \cdot \overline{B} + C_{in} \cdot A \cdot B = A \oplus B \oplus C_{in}$$

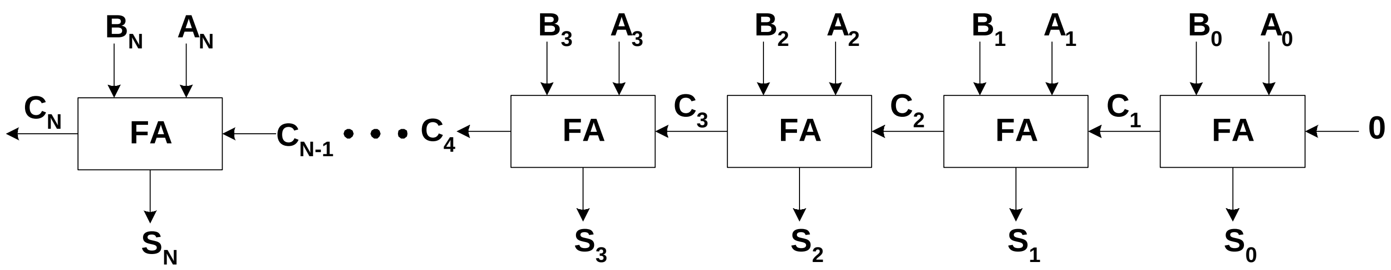
C_{in} \ A B	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$C_{out} = A \cdot B + C_{in} \cdot A + C_{in} \cdot B = A \cdot B + C_{in} \cdot (A \oplus B)$$



Somadores

Somador em cascata (ripple carry adder)



- A velocidade máxima de execução é limitada pela necessidade de propagar o “Carry” desde a soma do primeiro bit até à soma do bit mais significativo.
- No pior caso, o tempo de propagação do “Carry” será N x t_{PFA}.

Exemplo:

$$\begin{array}{cccccc} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \\ & & 0 & 0 & 0 & 0 \\ + & & 1 & 1 & 1 & 1 \\ \hline & & 1 & 1 & 1 & 1 \end{array}$$

A_0 comuta de 0 para 1

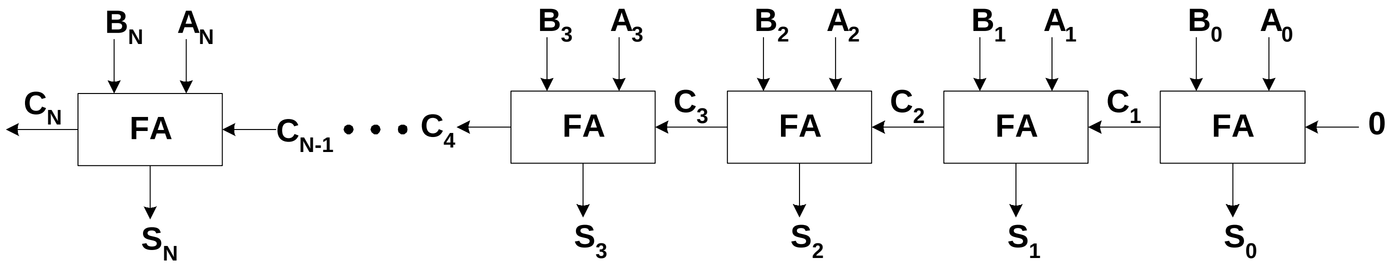
$A_i = 0, \forall_{i \neq 0}$

$B_i = 1, \forall_i$

$$\begin{array}{cccccc} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \\ & & 0 & 0 & 0 & 1 \\ + & & 1 & 1 & 1 & 1 \\ \hline & & 0 & 0 & 0 & 0 \end{array}$$

Somadores

Somador em cascata (ripple carry adder)



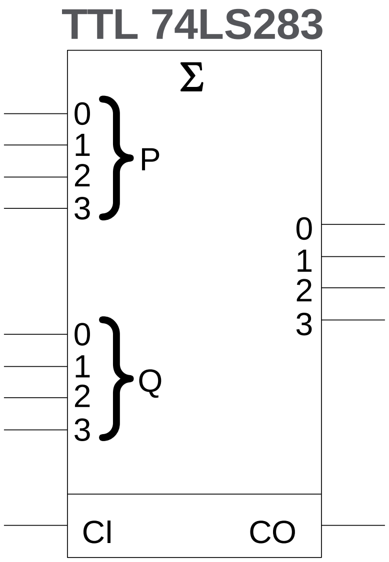
- O “Ripple Carry Adder” é o somador mais simples possível (que requer menos portas lógicas).

- Existem inúmeros circuitos alternativos para diversos compromissos velocidade/área.

Somadores

Somador de 4 bits

- Somador de 4 bits completo:
  - Soma:
    - 2 números de 4 bits cada
    - 1 bit de carry-in.
  - Gera:
    - Resultado da soma, com 4 bits
    - 1 bit de carry-out.



Representação de números com sinal

Representação de números negativos

Módulo + Sinal

- O bit mais significativo representa o sinal, e os restantes bits representam o seu valor absoluto.

  Ex.: -9 = 10001001

- **O valor zero tem duas representações…**

Decimal	Módulo + Sinal
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000
-0	1000
-1	1001
-2	1010
-3	1011
-4	1100
-5	1101
-6	1110
-7	1111
-8	-

?

Representação de números com sinal

Representação de números negativos

- **Complemento para 1**
  - O complemento para 1 de N, em n bits, é definido como $(2^n - 1) - N$.
  - $2^n - 1$ é um número constituído por n 1's.
  - Subtrair de 1 equivale a inverter o bit:

    $1 - 0 = 1$ e $1 - 1 = 0$.
  - Portanto, complementar para 1 corresponde a inverter todos os bits ($0 \rightarrow 1$ e $1 \rightarrow 0$).

    Ex.: -9 = 11110110

    $(= 11111111 - 00001001 = 255_{(10)} - 9_{(10)})$.

  - **O valor zero tem duas representações…**

Decimal	Complemento para 1
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000
-0	1111
-1	1110
-2	1101
-3	1100
-4	1011
-5	1010
-6	1001
-7	1000
-8	-

?

Representação de números com sinal

Representação de números negativos

Complemento para 2

- O complemento para 2 de N, em n bits, é definido como $2^n - N$ para $N \neq 0$, e 0 para $N = 0$.
- Portanto, complementar para 2 corresponde a complementar para 1 e somar 1.

  Ex.: -9 = 11110111

  (= 100000000 – 00001001 = $256_{(10)} - 9_{(10)}$).

- Na prática, o complemento para 2 pode ser formado do seguinte modo: mantêm-se todos os 0's menos significativos e o primeiro 1, e invertem-se todos os outros bits mais significativos.
- **Uma única representação para o valor zero.**

Decimal	Complemento para 2
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
+0	0000
-0	-
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

?

Representação de números com sinal

Números binários com sinal

- As operações usando o sistema de sinal e valor são mais complicadas, devido à necessidade de gerir separadamente o sinal e o valor.
- Por isso, são normalmente utilizadas representações em complemento. A representação em **complemento para 2** é habitualmente preferida em sistemas digitais por ter uma única representação para o valor zero, e por as operações envolvidas serem mais simples.

Decimal	Complemento para 2	Complemento para 1	Módulo + Sinal
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
+0	0000	0000	0000
-0	-	1111	1000
-1	1111	1110	1001
-2	1110	1101	1010
-3	1101	1100	1011
-4	1100	1011	1100
-5	1011	1010	1101
-6	1010	1001	1110
-7	1001	1000	1111
-8	1000	-	-

?

Representação de números com sinal

- **Extensão do sinal (complemento para 2)**
  - Representação de um número utilizando um determinado número de bits, através da adição/remoção de bits à esquerda iguais ao bit de sinal

Exemplos:

0100 =	+4	(4 bits)	→	00000100 =	+4	(8 bits)
1011 =	- 5	(4 bits)	→	11111011 =	- 5	(8 bits)
0010 =	+2	(4 bits)	→	010 =	+2	(3 bits)
1010 =	- 6	(4 bits)	→	??? =	- 6	(3 bits)

Representação de números com sinal

Soma aritmética de números com sinal usando complemento para 2

- A soma aritmética de dois números binários com sinal, representados em complemento para 2, é obtida pela simples adição dos dois números incluindo os bits de sinal. O último “carry out” não é considerado.

Exemplos:

$$\begin{array}{rr} & \boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{0} \\ 4 & 0\;1\;0\;0 \\ +\;3 & +\;0\;0\;1\;1 \\ \hline 7 & 0\;1\;1\;1 \end{array}
\qquad
\begin{array}{rr} & \boxed{1}\,\boxed{1}\,\boxed{0}\,\boxed{0} \\ 4 & 0\;1\;0\;0 \\ +\,(-3) & +\;1\;1\;0\;1 \\ \hline 1 & 0\;0\;0\;1 \end{array}$$

$$\begin{array}{rr} & \boxed{0}\,\boxed{0}\,\boxed{0}\,\boxed{0} \\ -4 & 1\;1\;0\;0 \\ +\;3 & +\;0\;0\;1\;1 \\ \hline -1 & 1\;1\;1\;1 \end{array}
\qquad
\begin{array}{rr} & \boxed{1}\,\boxed{1}\,\boxed{0}\,\boxed{0} \\ -4 & 1\;1\;0\;0 \\ +\,(-3) & +\;1\;1\;0\;1 \\ \hline -7 & 1\;0\;0\;1 \end{array}$$

Subtractores

Subtracção de números com sinal

- A subtracção de 2 números binários com sinal, representados em complemento para 2, é realizada de forma idêntica ao que acontece na representação decimal:

Exemplo:

$$
\begin{array}{rr@{\quad}ccccc}
 & 5 & & 0 & 1 & 0 & 1 \\
- & 2 & - & 0 & 0 & 1 & 0 \\
\hline
 & 3 & & 0 & 0 & 1 & 1 \\
 & & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{1} & \boxed{0} \quad \leftarrow \textit{Borrow}
\end{array}
$$

- O bit de empréstimo (*borrow*) é um valor que vai ser retirado ao bit de peso seguinte.

Subtractores

- **Subtracção de números com sinal usando complemento para 2**
  - A subtracção de dois números binários com sinal, representados em complemento para 2, é obtida do seguinte modo:
    - forma-se o complemento para 2 do subtractor
    - soma-se ao subtraendo.

Exemplo:

$$\begin{array}{r} 4 \\ -\;3 \\ \hline 1 \end{array} \qquad \begin{array}{r} 0\;1\;0\;0 \\ -\;0\;0\;1\;1 \\ \hline \end{array}$$

$$\begin{array}{r} 4 \\ +\,(-\,3) \\ \hline 1 \end{array} \qquad \begin{array}{r} 0\;1\;0\;0 \\ +\;1\;1\;0\;1 \\ \hline 0\;0\;0\;1 \end{array}$$

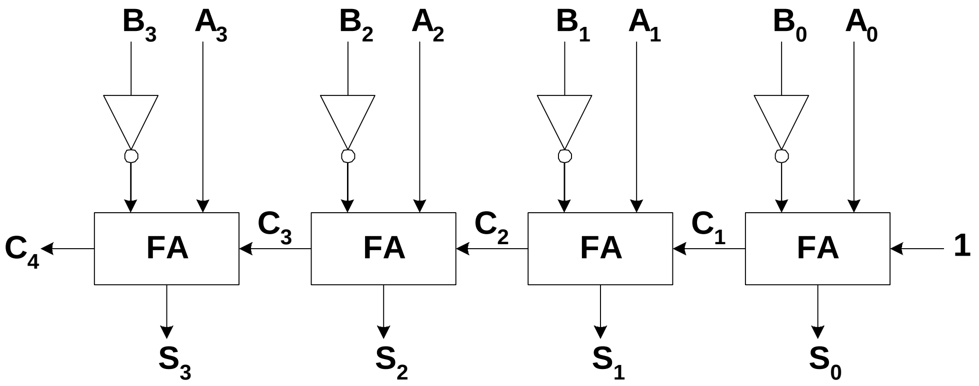
(através de complemento para 2)

$$\begin{array}{r} 1 \\ 0\;1\;0\;0 \\ +\;1\;1\;0\;0 \\ \hline 0\;0\;0\;1 \end{array}$$

(através de complemento para 1)

Subtractores

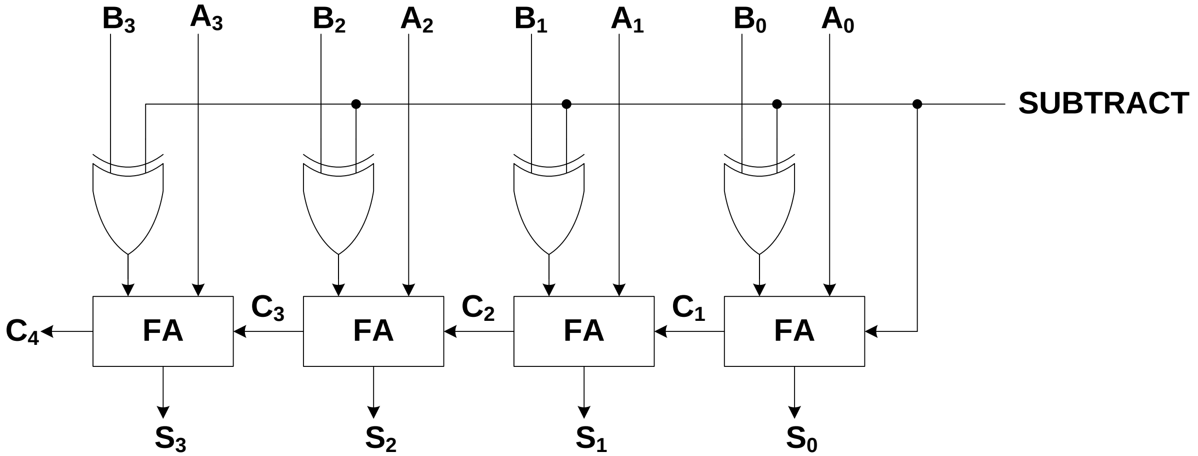
- **Subtracção de números com sinal usando complemento para 2**
  - Complemento para 2 = (Complemento para 1) + 1
    - A complementação para 1 é realizada invertendo todos os bits do subtractor.
    - A adição de 1 é efectuada pondo o Carry inicial a 1.



Circuito somador/subtractor

Circuito somador/subtractor

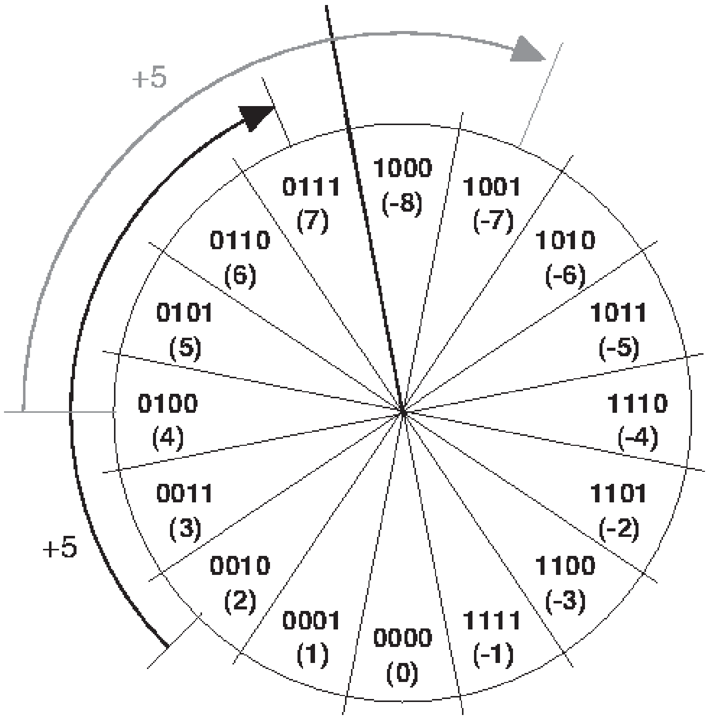
- As operações de adição e subtracção são habitualmente combinadas num único somador genérico, através da inclusão de 1 porta ou-exclusivo em cada Full-Adder.
- Quando o sinal de controlo SUBTRACT = 0, é realizada a adição A + B (os operandos B_i não são invertidos e $C_0 = 0$).
- Quando o sinal de controlo SUBTRACT = 1, é realizada a subtracção A – B (os operandos B_i são invertidos e $C_0 = 1$).



Excesso

- **Excesso (overflow)**

		$\boxed{0}$	$\boxed{1}$	$\boxed{0}$	$\boxed{0}$
4		0	1	0	0
+ 5	+	0	1	0	1
???		1	0	0	1



Excesso

- **Excesso (overflow)**
  - Para se obter um resultado correcto, na adição e na subtracção, é necessário assegurar que o resultado tem um número de bits suficiente. Se somarmos dois números de N bits e o resultado ocupar N+1 bits diz-se que ocorreu um **overflow**.
  - As unidades aritméticas digitais usam um número fixo de bits para armazenar os operandos e os resultados, sendo necessário detectar e sinalizar a ocorrência de um **overflow**.
    - Exemplo: um **overflow** pode ocorrer na adição se os dois operandos são ambos positivos ou se são ambos negativos.

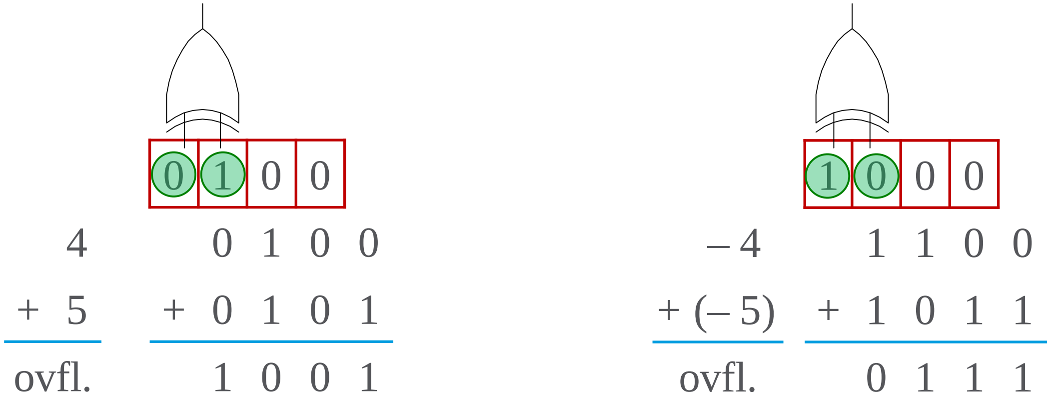
Excesso

Excesso (overflow)

- A condição de overflow pode ser detectada por inspecção dos dois bits de carry mais significativos.

Exemplo:

$$Overflow = CarryOut_{N-1} \oplus CarryOut_{N-2}$$



Excesso

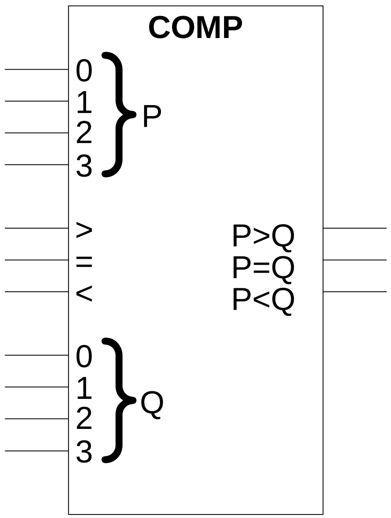
- **Qual a diferença entre os sinais de *carry* e *overflow*?**

Representação	$CO = CO_{N-1} = 1$	$O = 1$
SEM sinal	Excedeu a capacidade de representação	Sem significado
COM sinal	Sem significado	Excedeu a capacidade de representação

Circuito Comparador

- **Comparador de números de 4 bits**
  - Este circuito faz a comparação de 2 números binários de 4 bits.
  - A comparação é realizada através de uma operação de subtracção e análise do resultado.
  - O circuito pode ser ligado em cascata, para realizar comparações entre números de N > 4 bits, utilizando os 3 bits de entrada suplementares.

TTL 74LS85



Próxima Aula

- **Tema da Próxima Aula:**
  - Unidade Lógica e Aritmética (ULA)

Agradecimentos

Algumas páginas desta apresentação resultam da compilação de várias contribuições produzidas por:

- Nuno Roma
- Guilherme Arroz
- Horácio Neto
- Nuno Horta
- Pedro Tomás