



TÉCNICO LISBOA

SISTEMAS DIGITAIS (SD)

MEEC

Acetatos das Aulas Teóricas

Versão 4.0 - Português

Aula Nº 10:

Título: Circuitos combinatórios: somadores, subtratores e comparadores

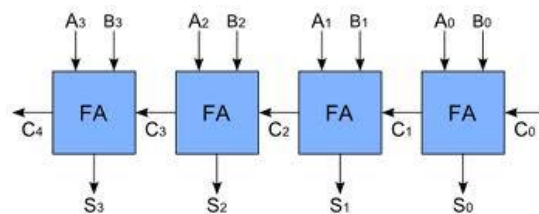
Sumário: Somadores, subtratores e comparadores.

2015/2016

Nuno.Roma@tecnico.ulisboa.pt

Sistemas Digitais (SD)

Circuitos combinatórios: somadores, subtratores e comparadores



Aula Anterior

■ Na aula anterior:

► Circuitos combinatórios típicos:

- Descodificadores
- Codificadores
- Multiplexers
- Demultiplexers



SEMANA	TEÓRICA 1	TEÓRICA 2	PROBLEMAS/LABORATÓRIO
14/Set a 19/Set	Introdução	Sistemas de Numeração e Códigos	
21/Set a 26/Set	Álgebra de Boole	Elementos de Tecnologia	P0
28/Set a 3/Out	Funções Lógicas	Minimização de Funções Booleanas (I)	L0
5/Out a 10/Out	Minimização de Funções Booleanas (II)	Def. Circuito Combinatório; Análise Temporal	P1
12/Out a 17/Out	Circuitos Combinatórios (I) – Codif., MUXs, etc.	Circuitos Combinatórios (II) – Som., Comp., etc.	L1
19/Out a 24/Out	Circuitos Combinatórios (III) - ALUs	Circuitos Sequenciais: Latches	P2
26/Out a 31/Out	Circuitos Sequenciais: Flip-Flops	Ling. de Descrição e Simulação de HW (ferramentas disponíveis no laboratório)	L2
2/Nov a 7/Nov	Caracterização Temporal	Registos	P3
9/Nov a 14/Nov	Revisões Teste 1	Contadores	L3
16/Nov a 21/Nov	Síntese de Circuitos Sequenciais: Definições	Síntese de Circuitos Sequenciais: Minimização do número de estados	P4
23/Nov a 28/Nov	Síntese de Circuitos Sequenciais: Síntese com Contadores	Memórias	L4
30/Nov a 5/Dez	Máq. Estado Microprogramadas: Circuito de Dados e Circuito de Controlo	Máq. Estado Microprogramadas: Endereçamento Explícito/Implícito	P5
7/Dez a 12/Dez	Circuitos de Controlo, Transferência e Processamento de Dados de um Processador	Lógica Programável	L5
14/Dez a 18/Dez	P6	P6	L6



■ Tema da aula de hoje:

- Circuitos combinatórios típicos:
 - Somadores / Subtractores
 - Comparadores

□ Bibliografia:

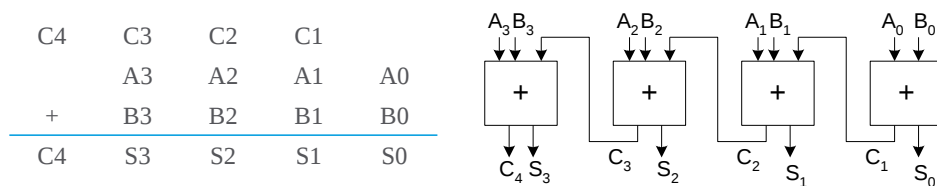
- M. Mano, C. Kime: Secções 4.2 a 4.4
- G. Arroz, J. Monteiro, A. Oliveira: Secções 5.1 a 5.3

■ Circuito para soma aritmética

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

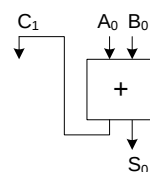
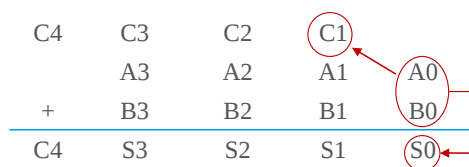
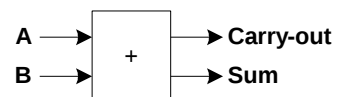
$$\begin{array}{r}
 7 \\
 + 2 \\
 \hline
 9
 \end{array}
 \quad
 \begin{array}{r}
 00 \leftarrow \text{Carry} \\
 01 \\
 + 010 \\
 \hline
 1001
 \end{array}$$

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



■ 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.

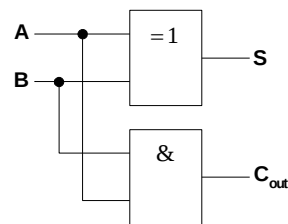


■ Circuito semi-somador

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

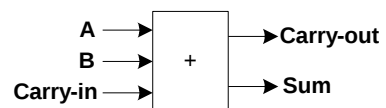
$$C_{out} = A \cdot B$$

$$S = A \oplus B$$

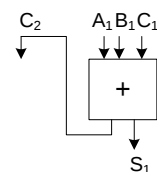


■ Circuito somador completo

- 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	A0
	A3	A2	A1	B0
+	B3	B2	B1	S0
C4	S3	S2	S1	



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

S:

AB	00	01	11	10
C _{in} 0	0	1	0	1
C _{in} 1	1	0	1	0

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

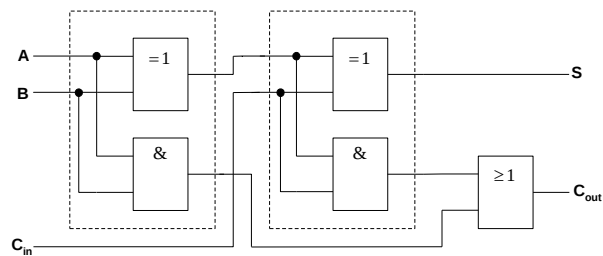
$$= A \oplus B \oplus C_{in}$$

C_{out}:

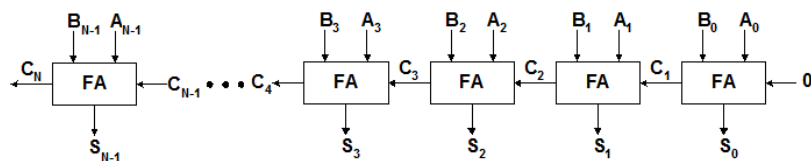
AB	00	01	11	10
C _{in} 0	0	0	1	0
C _{in} 1	0	1	1	1

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

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



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 $N \times t_{PFA}$.

Exemplo:

0 0 0 0

0 0 0 0
+ 1 1 1 1

1 1 1 1

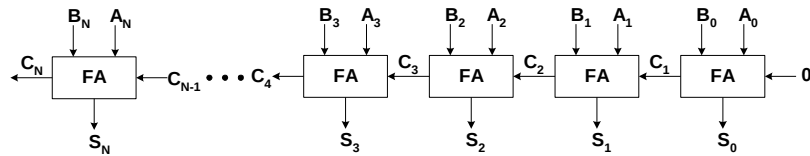
A_0 comuta de 0 para 1
 $A_i = 0, \forall i \neq 0$
 $B_i = 1, \forall i$

1 1 1 1

0 0 0 1
+ 1 1 1 1

0 0 0 0

■ Somador em cascata (ripple carry adder)

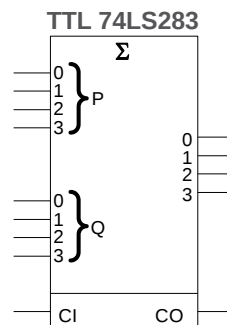


- ▶ O “Ripple Carry Adder” é o somador mais simples possível (que requer menos portas lógicas).
- ▶ Existem inúmeros circuitos alternativos para diversos compromissos velocidade/área.

■ 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 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 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 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.: $-9 = 11110111$

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

- Na prática, o complemento para 2 pode ser formado do seguinte modo: mantém-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

?

■ 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	-	-

?



■ 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:



■ 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:

$$0100 = +4 \text{ (4 bits)} \rightarrow 00000100 = +4 \text{ (8 bits)}$$

$$1011 = -5 \text{ (4 bits)} \rightarrow 11111011 = -5 \text{ (8 bits)}$$

$$0010 = +2 \text{ (4 bits)} \rightarrow 010 = +2 \text{ (3 bits)}$$

$$1010 = -6 \text{ (4 bits)} \rightarrow ??? = -6 \text{ (3 bits)}$$

■ 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}{r} 4 \quad 0 \ 1 \ 0 \ 0 \\ + 3 \quad 0 \ 0 \ 1 \ 1 \\ \hline 7 \quad 0 \ 1 \ 1 \ 1 \end{array} $	$ \begin{array}{r} 4 \quad 0 \ 1 \ 0 \ 0 \\ + (-3) \quad 1 \ 1 \ 0 \ 1 \\ \hline 1 \quad 0 \ 0 \ 0 \ 1 \end{array} $
$ \begin{array}{r} -4 \quad 1 \ 1 \ 0 \ 0 \\ + 3 \quad 0 \ 0 \ 1 \ 1 \\ \hline -1 \quad 1 \ 1 \ 1 \ 1 \end{array} $	$ \begin{array}{r} -4 \quad 1 \ 1 \ 0 \ 0 \\ + (-3) \quad 1 \ 1 \ 0 \ 1 \\ \hline -7 \quad 1 \ 0 \ 0 \ 1 \end{array} $

■ 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:

$$\begin{array}{r}
 5 \quad 0 \ 1 \ 0 \ 1 \\
 - 2 \quad - \ 0 \ 0 \ 1 \ 0 \\
 \hline
 3 \quad 0 \ 0 \ 1 \ 1
 \end{array}$$

0 0 0 1 0 ← Borrow

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

■ 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 subtrator
 - soma-se ao subtraendo.

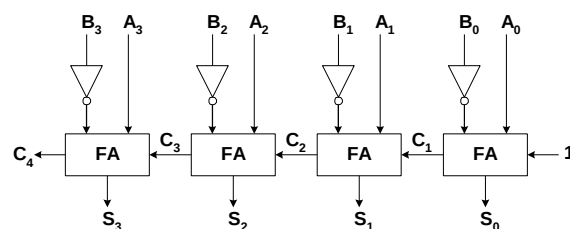
Exemplo:

$$\begin{array}{r}
 \begin{array}{r}
 4 \quad 0 \ 1 \ 0 \ 0 \\
 - 3 \quad - \ 0 \ 0 \ 1 \ 1 \\
 \hline
 1
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 4 \quad 0 \ 1 \ 0 \ 0 \\
 + (-3) \quad + \ 1 \ 1 \ 0 \ 1 \\
 \hline
 1 \quad 0 \ 0 \ 0 \ 1
 \end{array}
 \quad \rightarrow \quad
 \begin{array}{r}
 \\
 \\
 \\
 + \ 1 \ 1 \ 0 \ 0 \\
 \hline
 0 \ 0 \ 0 \ 1
 \end{array}
 \end{array}$$

(através de complemento para 2) (através de complemento para 1)

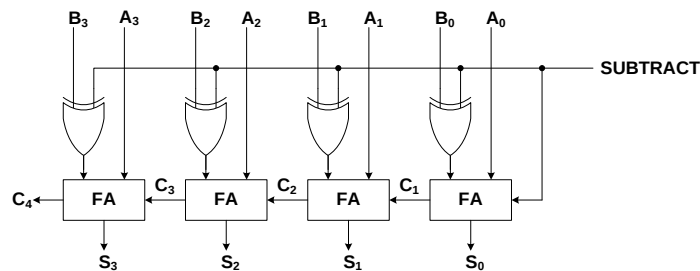
■ Subtracção de números com sinal usando complemento para 2

- ▶ Complemento para 2 = (Complemento para 1) + 1
 - A complementação para 1 é realizada invertendo todos os bits do subtrator.
 - A adição de 1 é efectuada pondo o Carry inicial a 1.



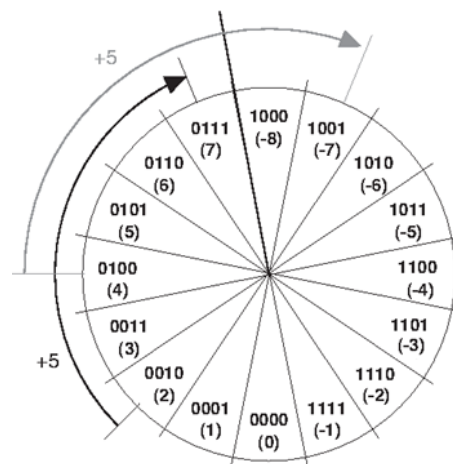
■ Circuito somador/subtractor

- ▶ As operações de adição e subtração 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 subtração $A - B$ (os operandos B_i são invertidos e $C_0 = 1$).



■ Excesso (overflow)

	0	1	0	0	
4	0	1	0	0	
+ 5	+	0	1	0	1
???	1	0	0	1	



■ Excesso (overflow)

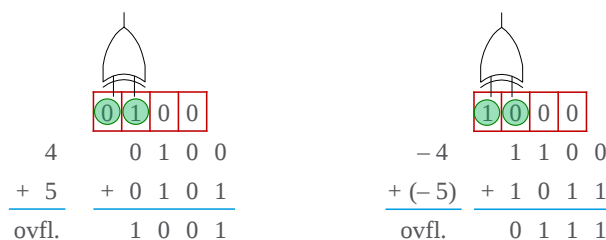
- ▶ Para se obter um resultado correcto, na adição e na subtracção, é necessário assegurar que o resultado tem um número de bits suficiente. Se somarmos dois números de N bits e o resultado ocupar N+1 bits diz-se que ocorreu um **overflow**.
- ▶ As unidades aritméticas digitais usam um número fixo de 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 (overflow)

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

Exemplo:

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

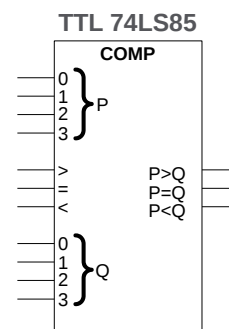


■ Qual a diferença entre os sinais de *carry* e *overflow*?

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

■ 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 subtração 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.





■ Tema da Próxima Aula:

- ▶ Unidade Lógica e Aritmética (ULA)



Agradecimentos

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

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