

# Lógica e Sistemas Digitais

Operações aritméticas

Código dos complementos

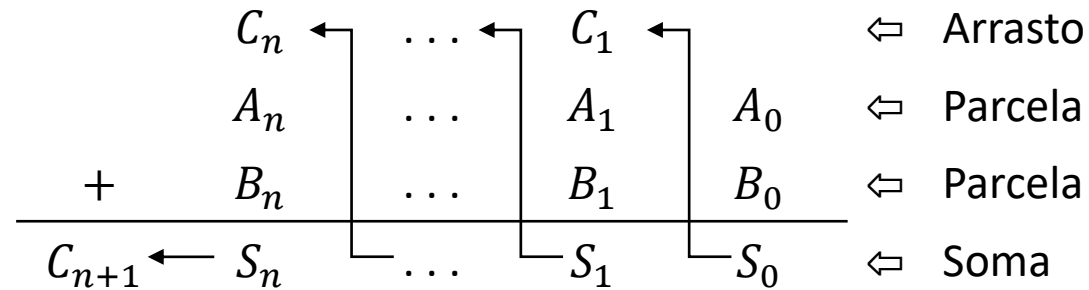
João Pedro Patriarca ([jpatri@cc.isel.ipl.pt](mailto:jpatri@cc.isel.ipl.pt))

Slides inspirados nos slides do prof. Mário Véstias



# Soma de dois números

---



- A soma de dois números binários é realizada tal como na soma de dois números na base 10
- Realizam-se somas parciais, do dígito de menor peso para o dígito de maior peso
- Sempre que a soma de dois dígitos exceder ou igualar o valor da base, o arrasto produzido é somado aos dígitos de peso seguinte

# Exemplos de somas de dois números

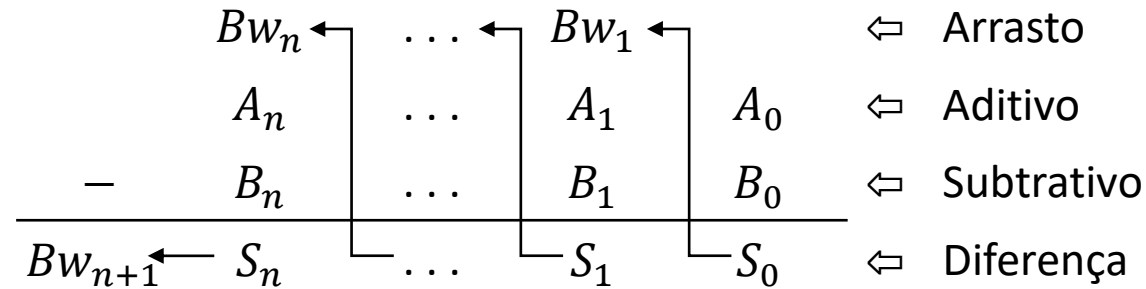
- Exemplo na base 2:

		1←	1←	1←	0←	0←	0←		Base 10	
	1	0	1	1	0	1	0	⇒	90	
+		1	1	1	1	0	0	⇒	+ 60	
	1←	0	0	1	0	1	1	0	⇒	150

- Exemplo na base 16:

		1←	1←	0←		Base 10
	3	A	D	9	⇒	15065
+	B	C	3	4	⇒	+ 48180
	F	7	0	D	⇒	63245

# Subtração de dois números



- A subtração de dois números binários é realizada tal como na subtração de dois números na base 10
- Realizam-se subtrações parciais, do dígito de menor peso para o dígito de maior peso
- Sempre que a subtração de dois dígitos for inferior a 0, é somado o valor da base ao dígito do aditivo, calculada a diferença e o arrasto é somado ao dígito do subtrativo de peso seguinte

# Exemplos de subtrações de dois números

- Exemplo na base 2:

	1	0	1	1	0	1	0	⇒	Base 10	
—		1	1	1	1	0	0	⇒	—	60
<hr/>										
	0	0	1	1	1	1	0	⇒		30

	1								1	0
—	0	+1		1	+1		1	+1	1	0
<hr/>										
	0		0		1		1		1	0

- Exemplo na base 16:

	B	C	3	4	⇒	Base 10	
—	3	A	D	9	⇒	—	15065
<hr/>							
	8	1	5	B	⇒		33115

	B		C			
—	3		A+1		D+1	9
<hr/>						
	8		1		5	B

# Multiplicação de dois número binários

- Multiplicação de dois bits A e B

A	B	F = A × B
0	0	0
0	1	0
1	0	0
1	1	1

- Multiplicação de palavras de  $n$  bits

						Base 10											
						1	0	1	1	1	⇒	23	⇒	multiplicando			
						×		1	1	0	1	⇒	×	13	⇒	multiplicador	
						<hr/>											
						1	0	1	1	1							
						0	0	0	0	0							
						<hr/>											
						1	0	1	1	1							
						<hr/>											
+	1	0	1	1	1												
						<hr/>											
1	0	0	1	0	1	0	1	1									

		Base 10			
×		23	⇒	multiplicando	
×		13	⇒	multiplicador	
		69			
		+	230		
		<hr/>		299	⇒ produto

# Multiplicação por potências de dois

- Multiplicação por 8

$$\begin{array}{r} \phantom{00000}10111 \\ \phantom{00000}\times \phantom{00000}1000 \\ \hline \phantom{00000}00000 \\ \phantom{00000}00000 \\ \phantom{00000}00000 \\ \phantom{00000}00000 \\ + \phantom{00000}10111 \\ \hline \phantom{00000}10111000 \end{array}$$

- O resultado do produto por uma potência de 2 equivale a adicionar à direita do multiplicando o número de zeros do multiplicador
- Outros exemplos:

$$135_{10} \times 4_{10} = 540_{10} \Leftrightarrow 10000111_2 \times 100_2 = 1000011100_2$$

$$87_{10} \times 32_{10} = 2784_{10} \Leftrightarrow 1010111_2 \times 100000_2 = 101011100000_2$$

# Números com sinal

- Na aritmética em geral é relevante a representação de números inteiros positivos e negativos (domínio  $\mathbb{Z}$ )
- Na ausência de limitação de representação de números, é usada a notação sinal-módulo para codificação de números
  - Na base 10 é usado o sinal da adição (+) e subtração (−) para representar o sinal. Ex: +10, -7, -125, +1276
  - Numa codificação em binário pode usar-se o bit de maior peso para representar o sinal: 0 representa um valor positivo; 1 representa um valor negativo
- Desvantagens nesta codificação:
  - Dois códigos para o valor 0;
  - Complexidade na implementação de operações aritméticas

Sinal-módulo			Número
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-0
1	0	1	-1
1	1	0	-2
1	1	1	-3



# Código dos complementos

---

- Num sistema computacional, a representação de números é finita.  
Ex: 4 bits; 8 bits; 16 bits; 32 bits; ... (por norma, num sistema computacional, o número de bits envolvidos na codificação de um número corresponde a potências de 2)
- O complemento ou o simétrico de um número é o valor que somado ao próprio número produz o valor 0 (não considerando o arrasto final)
  - Exemplo base 10 para 3 algarismos:
    - Complemento de  $123_{10}$  é  $877_{10}$  porque  $123_{10} + 877_{10} = \underline{1000}_{10}$
  - Exemplo base 16 para 4 algarismos:
    - Complemento de  $CA49_{16}$  é  $35B7_{16}$  porque  $CA49_{16} + 35B7_{16} = \underline{10000}_{16}$
  - Exemplo base 2 para 5 algarismos:
    - Complemento de  $01011_2$  é  $10101_2$  porque  $01011_2 + 10101_2 = \underline{100000}_2$
- Na base binária o complemento ou o simétrico de um valor é igualmente denominado por complemento para 2 ou complemento verdadeiro

# Determinação do complemento para 2 de um número

## Método 1

---

- O complemento para 2 é obtido a partir do complemento restringido ou complemento para 1 e, ao resultado do complemento, soma-se 1
- O complemento restringido é obtido através da negação bit a bit de todos os bits do valor
- Exemplos

Valor expresso com 6 bits	Complemento restringido	Complemento verdadeiro
101101	010010	010011
010011	101100	101101

$$\begin{array}{rcl} 101101 & \Rightarrow & 010010 \text{ Complemento para 1} \\ + & & 1 \\ \hline & & 010011 \text{ Complemento para 2} \end{array}$$

$$\begin{array}{rcl} 010011 & \Rightarrow & 101100 \text{ Complemento para 1} \\ + & & 1 \\ \hline & & 101101 \text{ Complemento para 2} \end{array}$$

# Determinação do complemento para 2 de um número

## Método 2

---

- Do bit de menor peso para o bit de maior peso, replicam-se os bits até encontrar o primeiro 1 (inclusive); a partir daí invertem-se os restantes bits
- Exemplos
  - Complemento do valor 111011000 a 9 bits  
 $111011000 \Rightarrow 000101000$
  - Complemento de 110111 a 8 bits  
 $00110111 \Rightarrow 11001001$
- Casos particulares: considerando representação a 4 bits, o complemento de 0 e 1000 (independente do método)

$$\begin{array}{l} 0000 \Rightarrow 0000 \\ 1000 \Rightarrow 1000 \end{array}$$

$$\begin{array}{r} 0000 \Rightarrow 1111 \\ + \quad 1 \\ \hline 1 \boxed{0000} \end{array}$$

$$\begin{array}{r} 1000 \Rightarrow 0111 \\ + \quad 1 \\ \hline \boxed{1000} \end{array}$$

# Correspondência entre código dos complementos e base 10

- Números naturais (domínio  $\mathbb{N}$ )

$$V_{(10)} = \sum_{i=0}^{k-1} d_i b^i = d_{k-1} b^{k-1} + d_{k-2} b^{k-2} + \dots + d_1 b^1 + d_0 b^0$$

$$1011_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 11_{10}$$

- Números relativos (domínio  $\mathbb{Z}$ )

$$V_{(10)} = \sum_{i=0}^{k-1} d_i b^i = -d_{k-1} b^{k-1} + d_{k-2} b^{k-2} + \dots + d_1 b^1 + d_0 b^0$$

$$1011_2 = -1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = -5_{10}$$

- No código dos complementos o bit de maior peso determina o sinal do número: 0 (positivo); 1 (negativo)
- Metade das codificações representa números positivos (de 0 a +7, para 4 bits); a outra metade representa número negativos (de -1 a -8, para 4 bits)
- O complemento de 0 é 0; a 4 bits não é representável o complemento de -8 (para codificar +8 são necessários, no mínimo, 5 bits)

A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	N	Z
0	0	0	0	0	+0
0	0	0	1	1	+1
0	0	1	0	2	+2
0	0	1	1	3	+3
0	1	0	0	4	+4
0	1	0	1	5	+5
0	1	1	0	6	+6
0	1	1	1	7	+7
1	0	0	0	8	-8
1	0	0	1	9	-7
1	0	1	0	10	-6
1	0	1	1	11	-5
1	1	0	0	12	-4
1	1	0	1	13	-3
1	1	1	0	14	-2
1	1	1	1	15	-1

# Exemplos

Número de bits na codificação	Número base 2	Número base 10 ( $\mathbb{N}$ )	Número base 10 ( $\mathbb{Z}$ )	Simétrico base 2	Simétrico base 10 ( $\mathbb{N}$ )	Simétrico base 10 ( $\mathbb{Z}$ )
3	110	6	-2	010	2	+2
4	0110	6	+6	1010	10	-6
4	0000	0	+0	0000	0	+0
3	100	4	-4	100	4	-4
5	00100	4	+4	11100	28	-4
4	1111	15	-1	0001	1	+1
6	111111	63	-1	000001	1	+1
4	0111	7	+7	1001	9	-7

# Subtração com código dos complementos

---

$$A - B \equiv A + (-B)$$

- Logo, subtrair  $A$  por  $B$  é equivalente a somar a  $A$  o complemento de  $B$
- Exemplos, considerando números codificados com 4 bits (operandos e resultados)

Nos  $\mathbb{N}$  e nos  $\mathbb{Z}$  operandos e resultado têm os mesmos valores

$5 - 2 = 5 + (-2) = 3$  Em binário

$$0101 - 0010 = 0101 + 1110 = 1\boxed{0011}$$

Nos  $\mathbb{N}$ :  $14 - 3 = 14 + (-3) = 11$

Nos  $\mathbb{Z}$ :  $(-2) - 3 = (-2) + (-3) = -5$

Em binário

$$1110 - 0011 = 1110 + 1101 = 1\boxed{1011}$$

# Exercícios

# Exercícios – códigos numéricos

- Preencha as células vazias da tabela.

Número de bits na codificação (min.)	Número base 2	Número base 10 (N)	Número base 10 (Z)	Simétrico base 2	Simétrico base 10 (N)	Simétrico base 10 (Z)
3		0				
3						-1
4	1001					
4			+3			
4				1011		
4	1100					
4		8				
			+12			



# Exercícios – códigos numéricos (resolvido)

- Preencha as células vazias da tabela.

Número de bits na codificação (min.)	Número base 2	Número base 10 (N)	Número base 10 (Z)	Simétrico base 2	Simétrico base 10 (N)	Simétrico base 10 (Z)
3	000	0	0	000	0	0
3	001	1	+1	111	7	-1
4	1001	9	-7	0111	7	+7
4	0011	3	+3	1101	13	-3
4	0101	5	+5	1011	11	-5
4	1100	12	-4	0100	4	+4
4	1000	8	-8	1000	8	-8 (N.R)
5	01100	12	+12	10100	20	-12

N.R. = Não representável.

O valor +8 não é representável em código dos complemento apenas com 4 bits

# Exercícios – operações adição e subtração em binário

---

- Considere números codificados com 5 bits. Realize as operações em binário (indique os valores dos operandos e resultados em binário):

$$9 + 5 =$$

$$7 - 8 =$$

$$-4 - 5 =$$

$$-14 - 6 =$$

$$21 + 3 =$$

## Exercícios – operações adição e subtração em binário (resolvido)

---

- Considere números codificados com 5 bits. Realize as operações em binário (indique os valores dos operandos e resultados em binário):

$$9 + 5 = 01001 + 00101 = 01110$$

$$7 - 8 = 00111 - 01000 = 11111$$

$$-4 - 5 = 11100 - 00101 = 10111$$

$$-14 - 6 = 10010 - 00110 = 01100$$

$$21 + 3 = 10101 + 00011 = 11000$$