



Licenciatura Ciências de Computação Licenciatura Eng^a. Física

2021/22

A.J.Proença

Tema

Introdução aos Sistemas de Computação



Estrutura do tema ISC

1. Representação de informação num computador
2. Organização e estrutura interna dum computador
3. Execução de programas num computador
4. Análise das instruções de um processador
5. Evolução da tecnologia e da eficiência

Noção de computador (1)



Um computador é um sistema físico que:

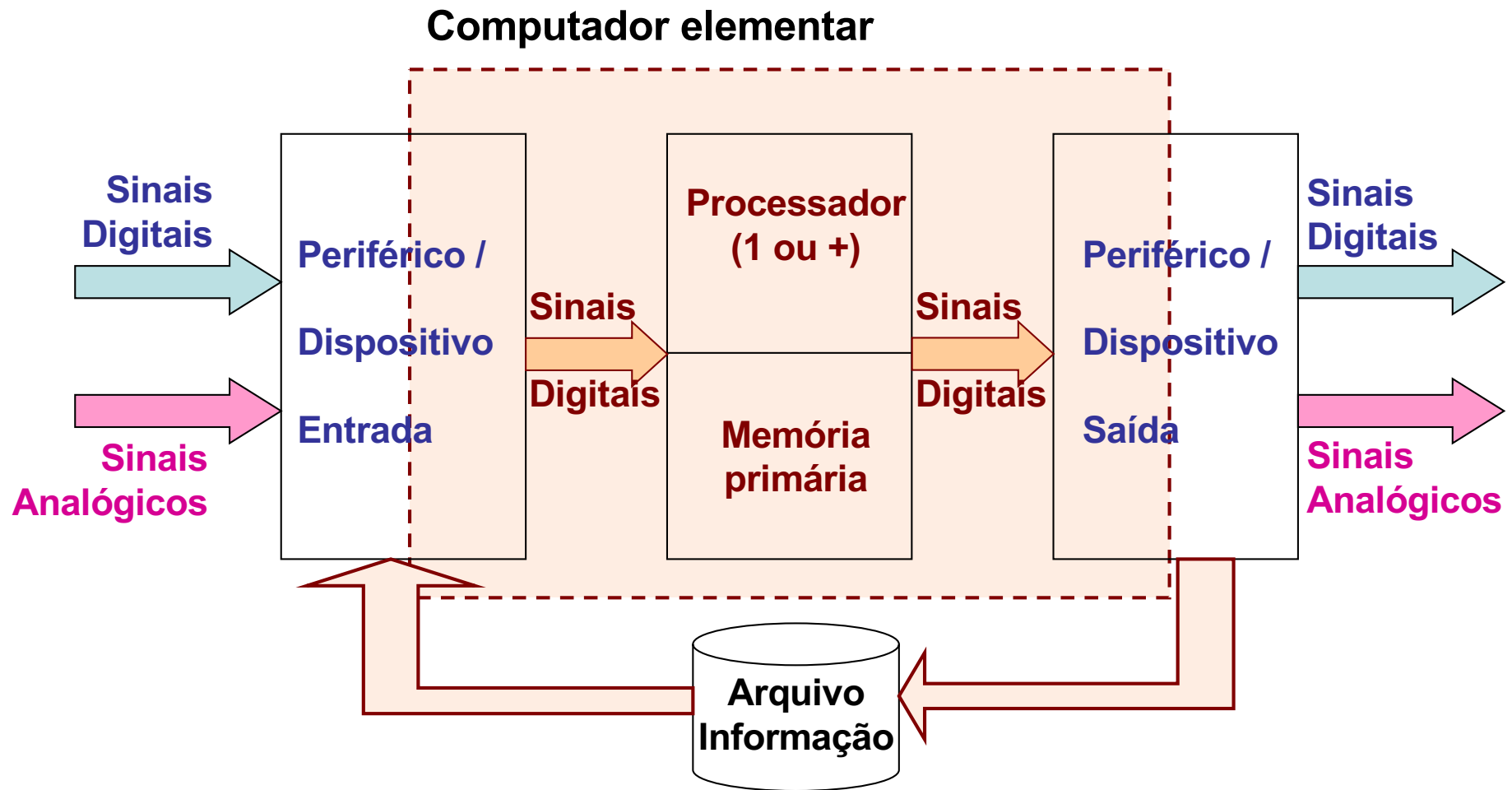
- recebe informação,
processa / arquiva informação,
transmite informação, e ...
- é programável
i.e., a funcionalidade do sistema pode ser modificada,
sem alterar fisicamente o sistema

Quando a funcionalidade é fixada no fabrico do sistema onde o computador se integra, diz-se que o computador existente nesse sistema está “embebido”: ex. *smart phone*, máq. fotográfica, automóvel, ...

Como se representa a informação num computador ?

Como se processa a informação num computador ?

Noção de computador (2)





- **Como se representa a informação num computador ?**
 - representação da informação num computador ->
- **Como se processa a informação num computador ?**
 - organização e funcionamento de um computador ->

Representação da informação num computador (1)



Como se representa a informação? – com binary digits!



Artigo Discussão

Algarismo

Um **algarismo** ou **dígito**, é um tipo de representação (um símbolo numérico, como "2" ou "5") usado em combinações (como "25") para representar **números** (como o número 25) em **sistemas de numeração posicionais**. O nome "dígito" vem do facto de os 9 dígitos (do **latim** *digitem*, "dedo") das mãos corresponderem aos 10 símbolos do sistema de numeração comum de **base 10**, isto é, o decimal (digestivo do latim antigo *decoração* . que significa nove) dígitos.

A palavra "algarismo" tem sua origem no nome do famoso matemático **Al-Khwarizmi**.

Mais:

- Cada um dos elementos de um numeral é um algarismo ou dígito:
 - Numeral com 3 dígitos: 426.
 - Numeral com 10 algarismos: 1.234.567.890
- • Dígitos **Binários**: podem ser apenas dois, o 0 (zero) e o 1 (um)

Representação da informação num computador (2)



Como se representa a informação?

- com binary digits!

Tipos de informação a representar:

- números (para cálculo)
 - » bases de numeração, inteiros (positivos e negativos)
 - » reais (*fp*), norma IEEE 754
- textos (caracteres alfanuméricos)
- conteúdos multimédia
- código para execução no computador

Sistemas de numeração: quanto vale na base 10 um n° representado numa base qualquer



1532.54₁₀ (base 10) ; quanto vale cada algarismo?

$$1 \cdot 10^3 + 5 \cdot 10^2 + 3 \cdot 10^1 + 2 \cdot 10^0 + 5 \cdot 10^{-1} + 4 \cdot 10^{-2} = 1532.54_{10}$$

Nota: a potência de 10 dá-nos a ordem do algarismo no número...

1532₆ (base 6) ; quanto vale cada algarismo na base 10?

$$1 \cdot 6^3 + 5 \cdot 6^2 + 3 \cdot 6^1 + 2 \cdot 6^0 = 416_{10}$$

1532₁₃ (base 13) ; quanto vale cada algarismo na base 10?

$$1 \cdot 13^3 + 5 \cdot 13^2 + 3 \cdot 13^1 + 2 \cdot 13^0 = 3083_{10}$$

110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?

$$1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} = 54.375_{10}$$

Sistemas de numeração: como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...

416₁₀ ; quanto vale cada algarismo na base 6?

- parte inteira ... parte decimal ...

3083₁₀ ; quanto vale cada algarismo na base 13?

- parte inteira ... parte decimal ...

54.375₁₀; quanto vale cada algarismo na base 2?

- parte inteira ... parte decimal ...

Sistemas de numeração: como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...

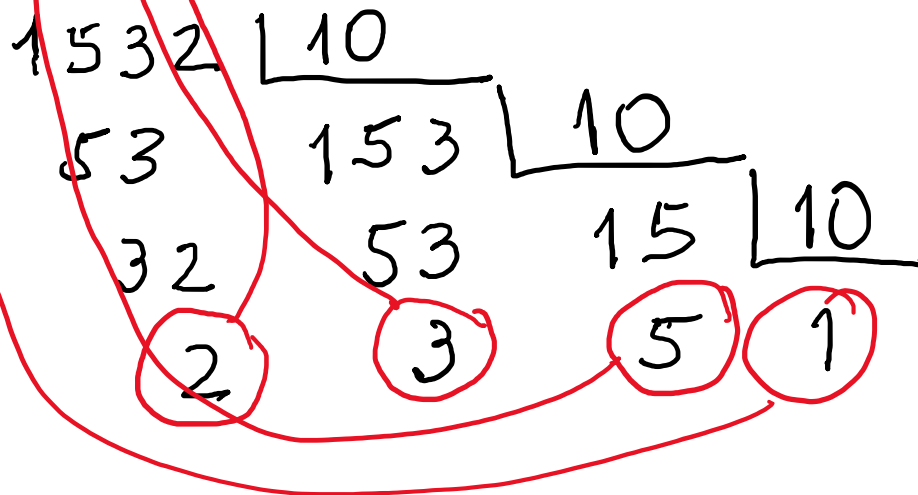
$$\begin{array}{r} 1532 \quad | \quad 10 \\ \hline 53 \quad 153 \quad | \quad 10 \\ \hline 32 \quad 5 \quad 1 \\ \hline 2 \end{array}$$

Sistemas de numeração: como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...



Sistemas de numeração: como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...

$$\begin{array}{r} 1532 \overline{) 10} \\ 53 \\ 32 \\ \underline{2} \end{array} \quad \begin{array}{r} 153 \overline{) 10} \\ 53 \\ \underline{3} \end{array} \quad \begin{array}{r} 15 \overline{) 10} \\ \underline{5} \\ 1 \end{array}$$

$$\begin{array}{r} 0.54 \\ \times 10 \\ \hline 5.40 \end{array}$$

Sistemas de numeração : como se passa um n° na base 10 para uma outra base



1532.54₁₀ (base 10) ; algoritmo para extrair os algarismos?

- parte inteira: divisão sucessiva pela base e...
- parte decimal: multiplicação sucessiva pela base e...

$$\begin{array}{r} 1532 \overline{) 10} \\ 53 \\ 32 \\ \underline{2} \end{array} \quad \begin{array}{r} 153 \overline{) 10} \\ 53 \\ \underline{15} \end{array} \quad \begin{array}{r} 15 \overline{) 10} \\ \underline{1} \end{array}$$

$$\begin{array}{r} 0.54 \\ \times 10 \\ \hline 5.40 \\ \times 10 \\ \hline 54.00 \end{array}$$

Sistemas de numeração: como se passa um n° na base 10 para uma outra base



1532₆ (base 6) ; quanto vale cada algarismo na base 10?
 $1 \cdot 6^3 + 5 \cdot 6^2 + 3 \cdot 6^1 + 2 \cdot 6^0 = 416_{10}$

416₁₀ ; quanto vale cada algarismo na base 6?
• parte inteira ... parte decimal ...

4 1 6 1 6

Sistemas de numeração: como se passa um n° na base 10 para uma outra base

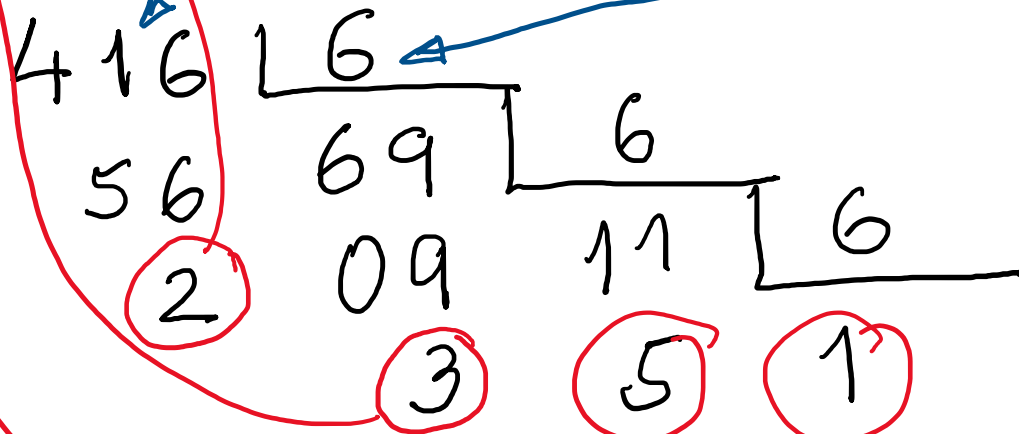


1532₆ (base 6) ; quanto vale cada algarismo na base 10?

$$1 \cdot 6^3 + 5 \cdot 6^2 + 3 \cdot 6^1 + 2 \cdot 6^0 = 416_{10}$$

416₁₀ ; quanto vale cada algarismo na base 6?

- parte inteira ... parte decimal ...



Sistemas de numeração: como se passa um n° na base 10 para uma outra base



110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?

$$1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = 54.375_{10}$$

54.375₁₀; quanto vale cada algarismo na base 2?

- parte inteira ... parte decimal ...

$$\begin{array}{r} 54 \quad 1 \quad 2 \\ 14 \quad 27 \quad 2 \\ \hline 0 \quad 07 \quad 13 \quad 2 \\ \hline 1 \quad 1 \quad 6 \quad 2 \\ \hline 0 \quad 3 \quad 2 \\ \hline 1 \quad 1 \end{array}$$

Sistemas de numeração: como se passa um n° na base 10 para uma outra base

110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?
 $1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} =$
54.375₁₀

54.375₁₀; quanto vale cada algarismo na base 2?
• parte inteira ... parte decimal ...

Handwritten conversion process for 54.375 from base 10 to base 2:

Integer part (54):

$$\begin{array}{r} 54 \div 2 = 27 \text{ remainder } 0 \\ 27 \div 2 = 13 \text{ remainder } 1 \\ 13 \div 2 = 6 \text{ remainder } 1 \\ 6 \div 2 = 3 \text{ remainder } 0 \\ 3 \div 2 = 1 \text{ remainder } 1 \\ 1 \div 2 = 0 \text{ remainder } 1 \end{array}$$

Fractional part (0.375):

$$\begin{array}{r} 0.375 \times 2 = 0.750 \text{ integer part } 0 \\ 0.750 \times 2 = 1.500 \text{ integer part } 1 \\ 0.500 \times 2 = 1.000 \text{ integer part } 1 \end{array}$$

Final result: 110110.011₂

Sistemas de numeração: caso particular da base 2



110110.011₂ (base 2) ; quanto vale cada algarismo na base 10?

$$1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = \dots$$

Para simplificar:

- eliminar os produtos, ignorar parcelas com produtos por 0

$$\bullet \text{ } 1*2^5 + 1*2^4 + 0*2^3 + 1*2^2 + 1*2^1 + 0*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = \dots$$

$$\Rightarrow 2^5 + 2^4 + 2^2 + 2^1 + 1/2^2 + 1/2^3 = \dots$$

Recomendações:

- decorar a tabuada das potências de 2 (**$2^0 + 2^{10}$**) $\longrightarrow$
- compreender as potências de 2 múltiplas de 10 $\longrightarrow$

Numeração de base 2: dicas para uma rápida conversão de potências de 2 para a base 10



$$2^0 = 1$$

$$2^1 = 2$$

$$2^2 = 4$$

$$2^3 = 8$$

$$2^4 = 16$$

$$2^5 = 32$$

$$2^6 = 64$$

$$2^7 = 128$$

$$2^8 = 256$$

$$2^9 = 512$$

$$2^{10} = 1024$$

$$2^{10} = 1024 = 1 \text{ Ki}(\text{bi}) \approx 1000 = 10^3 = 1 \text{ K}(\text{ilo})$$

...

$$2^{12} = 2^2 * 2^{10} = 4 \text{ Ki}(\text{bi}) \approx 4000 = 4 * 10^3 = 4 \text{ K}$$

...

$$2^{16} = 2^6 * 2^{10} = 64 \text{ Ki}(\text{bi}) \approx 64 * 10^3 = 64 \text{ K}$$

$$2^{20} = 1 \text{ Me}(\text{bi}) \approx 1000000 = 10^6 = 1 \text{ M}(\text{ega})$$

$$2^{30} = 1 \text{ Gi}(\text{bi}) \approx 1000000000 = 10^9 = 1 \text{ G}(\text{iga})$$

$$2^{40} = 1 \text{ Te}(\text{bi}) \approx 10^{12} = 1 \text{ T}(\text{era})$$

$$2^{50} = 1 \text{ Pe}(\text{bi}) \approx 10^{15} = 1 \text{ P}(\text{eta})$$

Vantagens destas dicas



- Na conversão de binário para base 10:

110110.011₂ (base 2)

$$\Rightarrow 2^5 + 2^4 + 2^2 + 2^1 + 1/2^2 + 1/2^3 =$$

$$= 32 + 16 + 4 + 2 + 1/4 + 1/8 = 54 + 0.375$$

- Na conversão de base 10 para binário:

$$290 = 256 +$$
$$256 - \cancel{512} \quad 2^8$$

Vantagens destas dicas



- Na conversão de binário para base 10:

110110.011₂ (base 2)

$$\Rightarrow 2^5 + 2^4 + 2^2 + 2^1 + 1/2^2 + 1/2^3 =$$

$$= 32 + 16 + 4 + 2 + 1/4 + 1/8 = 54 + 0.375$$

- Na conversão de base 10 para binário:

$$\begin{array}{r} 290 = 256 + 32 + 2 \\ \quad \quad 2^8 \quad \quad 2^5 \quad \quad 2^1 \\ -256 \\ \hline 34 \\ -32 \\ \hline 2 \end{array}$$

Vantagens destas dicas



- Na conversão de binário para base 10:

110110.011₂ (base 2)

$$\Rightarrow 2^5 + 2^4 + 2^2 + 2^1 + 1/2^2 + 1/2^3 =$$

$$= 32 + 16 + 4 + 2 + 1/4 + 1/8 = 54 + 0.375$$

- Na conversão de base 10 para binário:

$$\begin{array}{r} 290 = 256 + 32 + 2 \\ \hline -256 \\ \hline 34 \\ \hline -32 \\ \hline 2 \end{array}$$

Handwritten binary representation of 290:

$$\begin{array}{ccccccc} & 2^8 & & 2^5 & & & 2^1 \\ & \swarrow & & \swarrow & & & \downarrow \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0_2 \end{array}$$

The binary number 100100010₂ is shown with a bracket underneath it.

Sistemas de numeração: caso particular da base 16 (hexadecimal)



- Dígitos na base 16:
 $0, 1, 2, \dots, 9, \overset{10}{a}, \overset{11}{b}, \overset{12}{c}, \overset{13}{d}, \overset{14}{e}, \overset{15}{f}$
- Vantagens sobre um valor de 32 bits:
 $10100110100001110110010111010100_2$ VS. $a68765d4_{16}$
- Facilidade de conversão:
$$\begin{array}{cccccccc} 1010 & 0110 & 1000 & 0111 & 0110 & 0101 & 1101 & 0100_2 \\ a & 6 & 8 & 7 & 6 & 5 & d & 4_{16} \end{array}$$

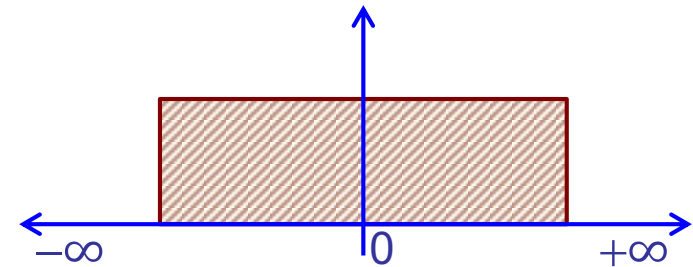
←
- Mesmo com ponto decimal:
$$\begin{array}{cccccccc} 1010011010000111011001011101.01_2 & \longleftarrow & \longrightarrow \\ 1010 & 0110 & 1000 & 0111 & 0110 & 0101 & 1101.0100_2 & \longleftarrow & \longrightarrow \\ a & 6 & 8 & 7 & 6 & 5 & d & . & 4_{16} \end{array}$$

Representação de inteiros



Gama de valores representáveis

- ideal: todos os valores e simetria em relação ao 0
- mas ...
- e quantos bits para representar um inteiro?



Representação de positivos & negativos

- estratégias
- análise dum exemplo com todos os valores possíveis
 - S+M: Sinal + Magnitude/amplitude
 - Complemento para 1
 - Complemento para 2
 - Notação por excesso

Inteiros positivos e negativos: o universo com 3 bits (1)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000				
1	001				
2	010				
3	011				
4	100				
5	101				
6	110				
7	111				

Inteiros positivos e negativos: o universo com 3 bits (2)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000	+			
1	001	+			
2	010	+			
3	011	+			
4	100	-			
5	101	-			
6	110	-			
7	111	-			

Inteiros positivos e negativos: o universo com 3 bits (3)

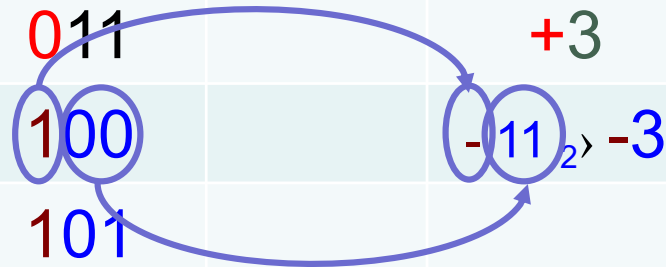


Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000	+0			
1	001	+1			
2	010	+2			
3	011	+3			
4	100	-0			
5	101	-1			
6	110	-2			
7	111	-3			

Inteiros positivos e negativos: o universo com 3 bits (4)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000		+0		
1	001		+1		
2	010		+2		
3	011		+3		
4	100		-11 ₂ > -3		
5	101				
6	110				
7	111				



Inteiros positivos e negativos: o universo com 3 bits (5)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000		+0		
1	001		+1		
2	010		+2		
3	011		+3		
4	100		$-11_2 \rightarrow -3$		
5	101		$-10_2 \rightarrow -2$		
6	110		$-01_2 \rightarrow -1$		
7	111		$-00_2 \rightarrow -0$		

Inteiros positivos e negativos: o universo com 3 bits (6)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000		+0	+0	
1	001		+1	+1	
2	010		+2	+2	
3	011		+3	+3	
4	100		$-11_2 \rightarrow -3$	$-(11+1)_2 \rightarrow -4$	
5	101		$-10_2 \rightarrow -2$		
6	110		$-01_2 \rightarrow -1$		
7	111		$-00_2 \rightarrow -0$		

Inteiros positivos e negativos: o universo com 3 bits (7)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso
0	000		+0	+0	
1	001		+1	+1	
2	010		+2	+2	
3	011		+3	+3	
4	100		$-11_2 \rangle -3$	$-(11+1)_2 \rangle -4$	
5	101		$-10_2 \rangle -2$	$-(10+1)_2 \rangle -3$	
6	110		$-01_2 \rangle -1$	$-(01+1)_2 \rangle -2$	
7	111		$-00_2 \rangle -0$	$-(00+1)_2 \rangle -1$	

Vantagem da representação em complemento para 2: mesmo hardware para soma de inteiros com ou sem sinal (1)



base 2

Base 2	Base 10
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Handwritten calculations in base 2:

$$\begin{array}{r} 001 \\ + 101 \\ \hline 110 \end{array}$$

Handwritten calculations in base 10:

$$\begin{array}{r} 1 \\ + 5 \\ \hline 6 \end{array}$$

Arrows indicate the mapping from binary to decimal and the addition of 5 to the binary value 110 to get 6 in decimal.

Comp 2	Base 2
+0	000
+1	001
+2	010
+3	011
$-(11+1)_2$ -4	100
$-(10+1)_2$ -3	101
$-(01+1)_2$ -2	110
$-(00+1)_2$ -1	111

Vantagem da representação em complemento para 2: mesmo hardware para soma de inteiros com ou sem sinal (2)

base 2

Base 2	Base 10
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Comp 2	Base 2
+0	000
+1	001
+2	010
+3	011
-(11+1) ₂ -4	100
-(10+1) ₂ -3	101
-(01+1) ₂ -2	110
-(00+1) ₂ -1	111

Handwritten calculations:

Green:
$$\begin{array}{r} 1 \\ +5 \\ \hline 6 \end{array}$$

Red:
$$\begin{array}{r} 1 \\ +(-3) \\ \hline -2 \end{array}$$

Base 2 addition:
$$\begin{array}{r} 001 \\ +101 \\ \hline 110 \end{array}$$

Vantagem da representação em complemento para 2: mesmo hardware para soma de inteiros com ou sem sinal (3)

base 2

Base 2	Base 10
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

base 2

Comp 2	Base 2
+0	000
+1	001
+2	010
+3	011
-(11+1) ₂ -4	100
-(10+1) ₂ -3	101
-(01+1) ₂ -2	110
-(00+1) ₂ -1	111

base 2

1 001

+5

6

+101

110

+(-3)

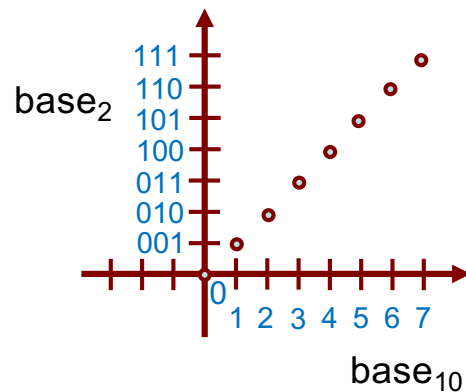
-2

mesmo somador em binário!

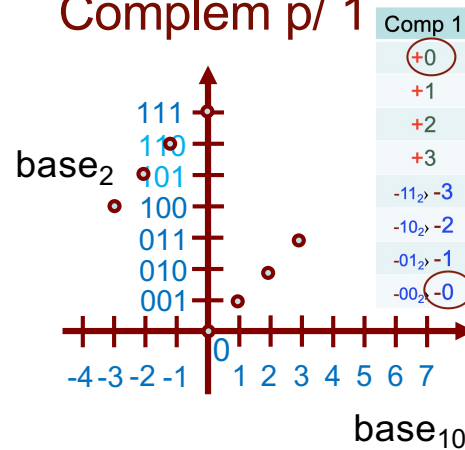
Inteiros positivos e negativos: porquê mais uma maneira de codificação?



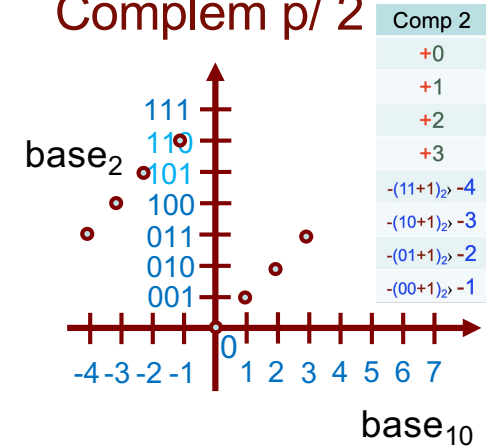
Sem sinal



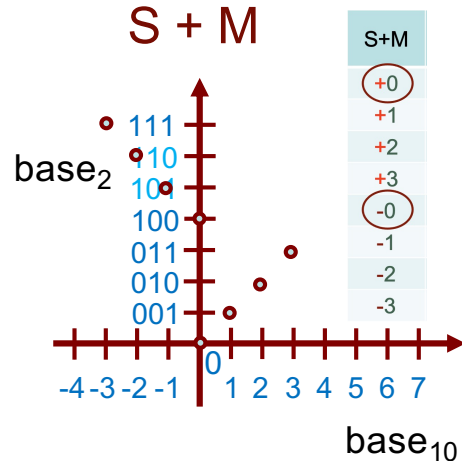
Complem p/ 1



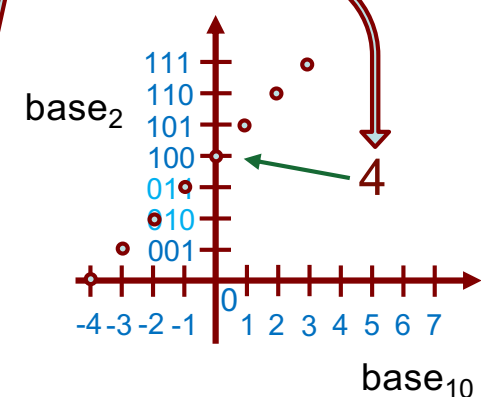
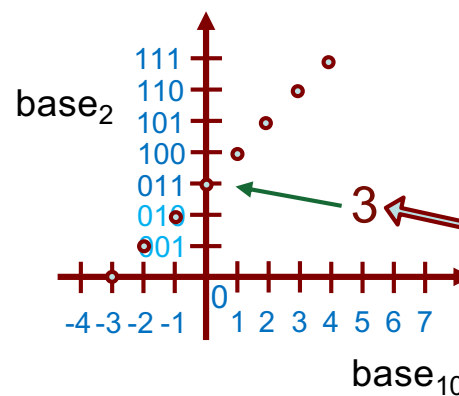
Complem p/ 2



S + M



Notação por excesso



Inteiros positivos e negativos: o universo com 3 bits (8)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso $2^{n-1}-1$
0	000				$0 - 3 \succ -3$
1	001				$1 - 3 \succ -2$
2	010				$2 - 3 \succ -1$
3	011				$3 - 3 \succ 0$
4	100				$4 - 3 \succ +1$
5	101				$5 - 3 \succ +2$
6	110				$6 - 3 \succ +3$
7	111				$7 - 3 \succ +4$

Nota: $n = \text{\#bits}$, $2^{n-1}-1 = 2^{3-1}-1 = 2^2-1 = 3$

Inteiros positivos e negativos: o universo com 3 bits (9)



Base 10	Base 2	S+M	Comp 1	Comp 2	Excesso 2^{n-1}	Excesso $2^{n-1}-1$
0	000				0 - 4 > -4	0 - 3 > -3
1	001				1 - 4 > -3	1 - 3 > -2
2	010				2 - 4 > -2	2 - 3 > -1
3	011				3 - 4 > -1	3 - 3 > 0
4	100				4 - 4 > 0	4 - 3 > +1
5	101				5 - 4 > +1	5 - 3 > +2
6	110				6 - 4 > +2	6 - 3 > +3
7	111				7 - 4 > +3	7 - 3 > +4

Nota: $n = \text{\#bits}$, $2^{n-1} = 2^{3-1} = 2^2 = 4$, $2^{n-1}-1 = 2^{3-1}-1 = 2^2-1 = 3$

Representação de reais em vírgula flutuante (1)



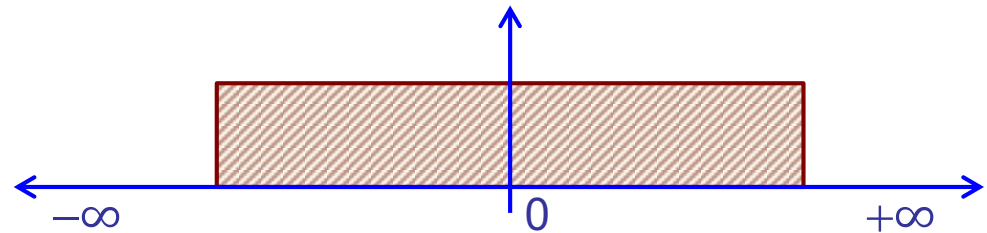
- **Gama de valores**

- esta gama é viável?
- não, porque...

entre quaisquer 2 n^os reais há um conjunto infinito de valores

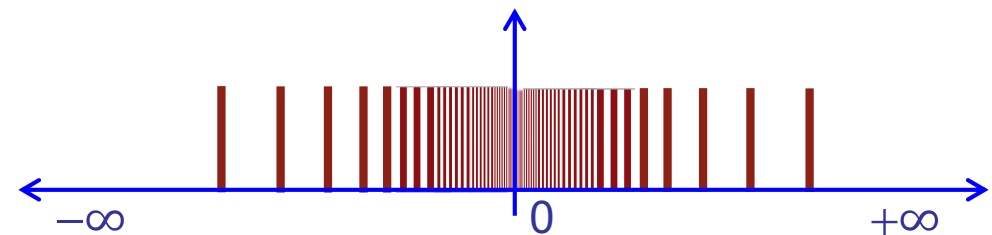
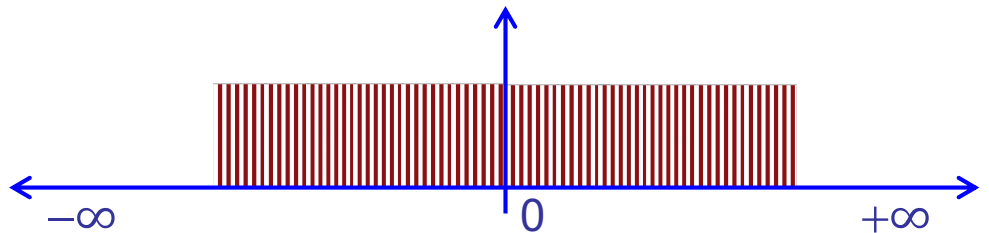
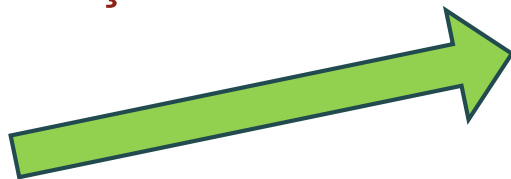
- então faz sentido a seguinte representação?

Notação vírgula fixa



- ou faria mais sentido a seguinte representação?

Notação científica



Representação de reais em vírgula flutuante (2)



- **Notação em vírgula fixa vs. notação científica**

- **Exemplos na base 10 usando apenas 5 dígitos**

- **Vírgula fixa: $\pm X X X . X X$**

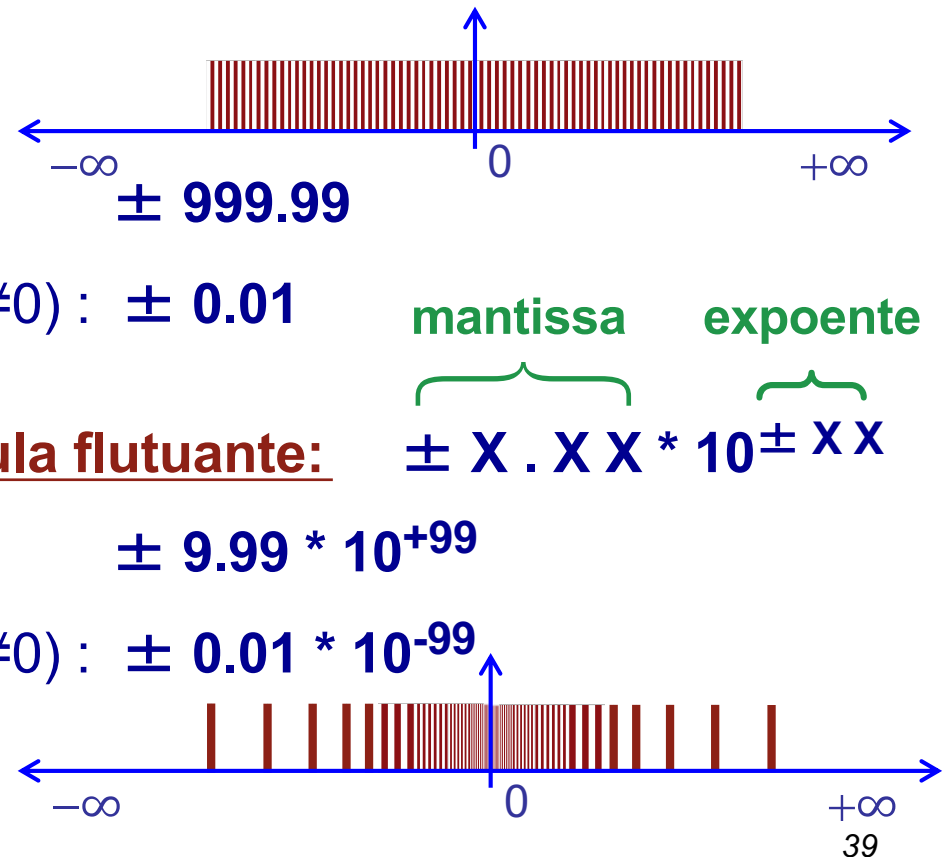
maior valor representável : ± 999.99

menor valor representável ($\neq 0$) : ± 0.01

- **Notação científica ou vírgula flutuante: $\pm X . X X * 10^{\pm X X}$**

maior valor representável : $\pm 9.99 * 10^{+99}$

menor valor representável ($\neq 0$) : $\pm 0.01 * 10^{-99}$



Representação de reais em vírgula flutuante (3)



- **Normalização na representação**

- Para garantir que cada n° a representar usa apenas uma única sequência de dígitos, há que fixar o local para o ponto decimal

- N° normalizado: à esquerda do ponto decimal apenas 1 dígito ≠ 0

$$\pm Y . XXX \dots X * \text{Radix}^{\text{Exp}} \quad 0 < Y \leq \text{maior_dígito}$$

- Menor valor normalizado representável:

$$\pm 1.000 \dots 0 * 10^{-9 \dots 9}$$

- Nota_1: a normalização não permite representar o zero

Nota_2: a normalização desperdiça muitos dígitos → subnormais

- N° subnormal: expoente fixo (menor normalizado), 0 à esq do ponto

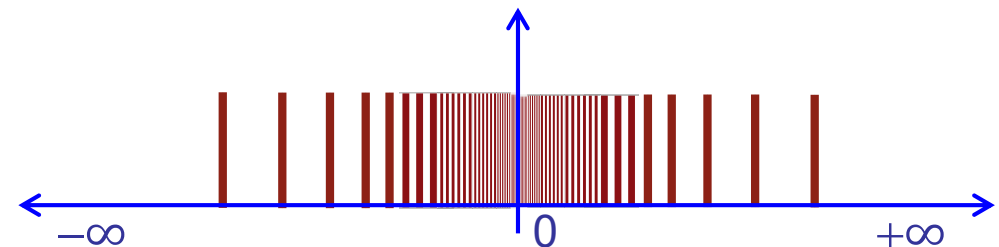
$$\text{de } \pm 0.000 \dots 0 * 10^{-9 \dots 9} \text{ a } \pm 0.999 \dots 9 * 10^{-9 \dots 9}$$

Representação binária de reais em vírgula flutuante (4)



- **Formato binário dum valor em fp**
 - formato inclui mantissa e expoente; radix é implícito (=2)
 - mantissa e expoente podem ser valores positivos ou negativos
 - formato tem nº de bits fixo, variável:
 - **reais com precisão simples: 32 bits**
 - **reais com precisão dupla: 64 bits**
 - **reais com meia precisão : 16 bits**

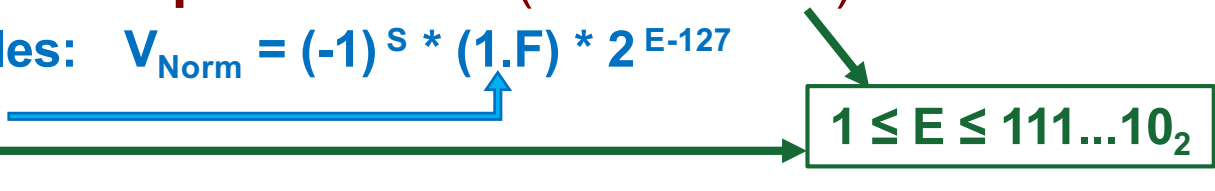
- **mantissa vs. expoente** ⇔



precisão vs. intervalo valores representáveis

A norma IEEE 754-2008 para valores em fp (1)



- **Representação do sinal na mantissa e no expoente**
 - Mantissa: $S + M$
 - Expoente: Excesso $2^{n-1} - 1$
- **Valor decimal de um fp em binário (normalizado)**
 - Precisão simples: $V_{\text{Norm}} = (-1)^S * (1.F) * 2^{E-127}$
 - Bit escondido 
- Exceções (zero, subnormais, ...): $E = 0$ ou $E = 1111...1_2$
- **Representação do zero:** $E = 0$ e $F = 0$
- **Representação e valor decimal de um fp (subnormal)**
 - Representação: $E = 0$ e $F \neq 0$
 - Precisão simples: $V_{\text{SubN}} = (-1)^S * (0.F) * 2^{-126}$
- **Representação de $\pm\infty$:** $E = 1111...1_2$ e $F = 0$
- **Representação de n.º não real:** $E = 1111...1_2$ e $F \neq 0 \rightarrow \text{NaN}$

A norma IEEE 754-2008 para valores em fp (2)



Format of Floating points IEEE754

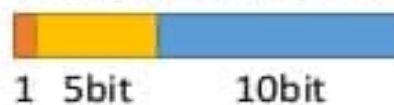
64bit = double, double precision



32bit = float, single precision



16bit = half, half precision

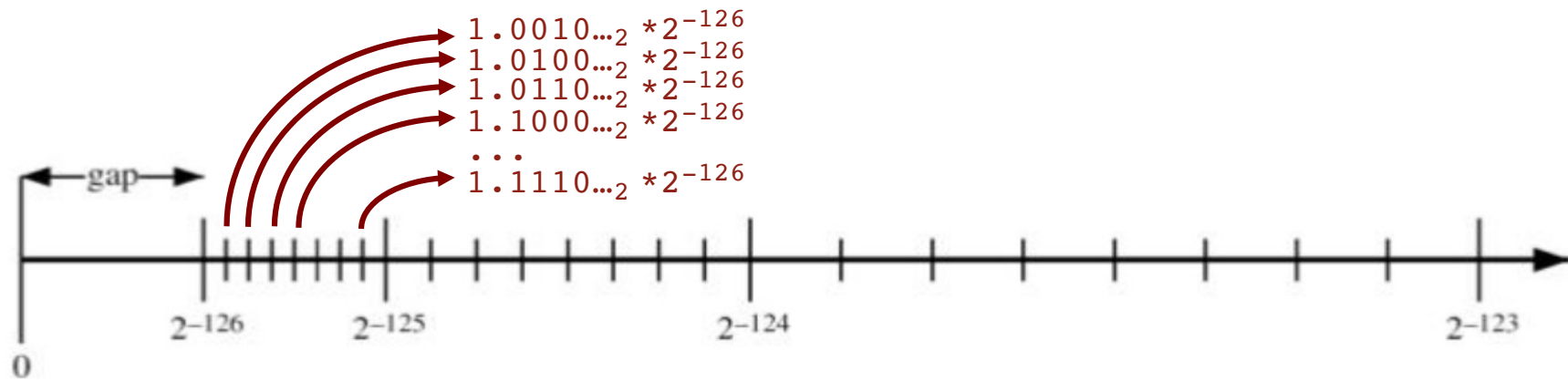


Signal

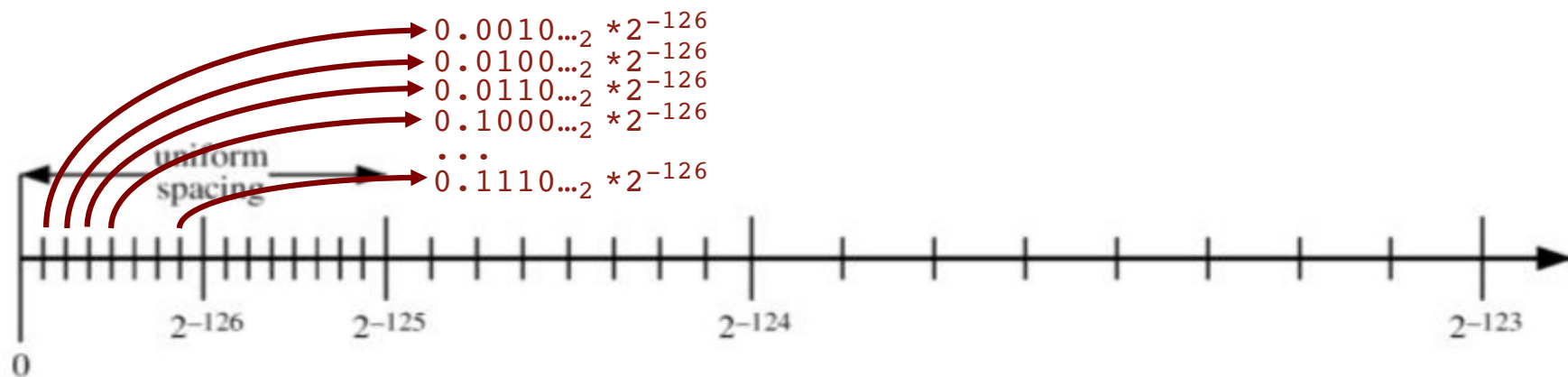
Expoent

Mantissa

O papel dos subnormais na norma IEEE 754



(a) 32-bit format without denormalized numbers



(b) 32-bit format with denormalized numbers

Representação da informação num computador (2)



Como se representa a informação?

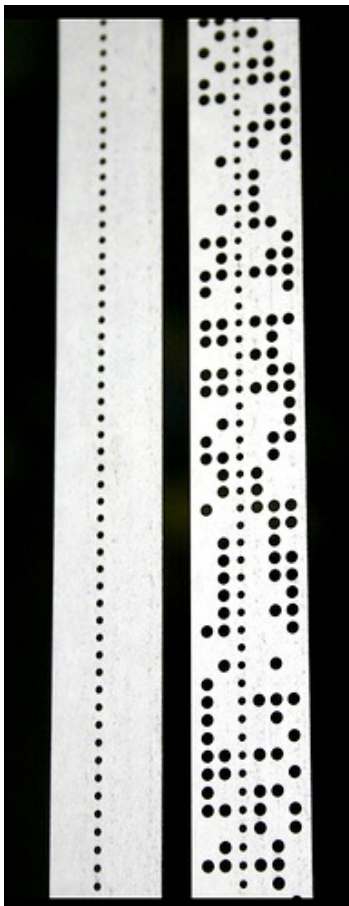
- com binary digits!

Tipos de informação a representar:

- números (para cálculo)
 - » bases de numeração, inteiros (positivos e negativos)
 - » reais (*fp*), norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
- código para execução no computador

Ex.: codificação telegráfica de texto, código de Baudot, 5-bits

- Baudot,



V	IV		I	II	III	V	IV		I	II	III
		A /	•			•	•	P %	•	•	•
	•	B 8			•	•	•	Q /	•		•
	•	C 9	•		•	•	•	R -			•
	•	D 0	•	•	•	•	•	S ;			•
		E 2		•		•	•	T !	•		•
		E' &	•	•				U 4	•		•
	•	F ̑		•	•	•	•	V '	•	•	•
	•	G 7		•		•	•	W ?		•	•
	•	H ̑	•	•		•	•	X ,		•	
		I ̑		•	•			Y 3			•
	•	J 6	•			•	•	Z :	•	•	
•	•	K (	•			•	•	̑ .	•		
•	•	L =	•	•		•	•	̑̑̑̑ Erasure			
•	•	M)		•		•	•	Figure Blank			
•	•	N N°		•	•	•	•	Letter Blank			
		O 5	•	•	•						

Letters	Figures	V	IV	I	II	III	Letters	Figures	V	IV	I	II	III
A	1			•			-	.	•		•		
E	2				•		X	9	•			•	
Y	3					•	S	7					•
/	4			•	•		Z	:	•		•	•	
1	5				•	•	W	?	•			•	•
U	6			•		•	T	2	•		•	•	•
O	7				•	•	V	1	•		•	•	•
							Letter Blank						
J	8	•	•				K	(	•	•	•		
G	9	•		•	•		M	)	•	•			
B	0												
H	1												
F	2												
C	3												
D	4												
Figure Blank													

Fig 1. The Baudot code

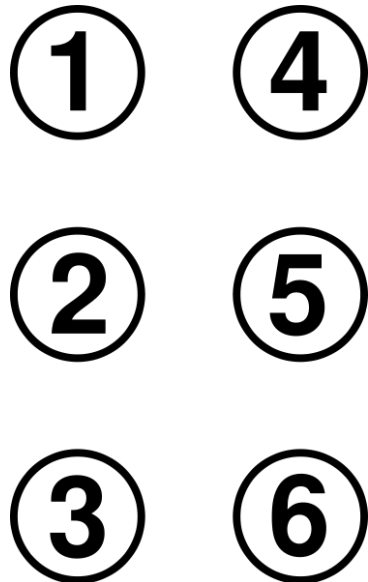


Ex.: codificação de texto em relevo, código Braille com 6-bits



- Baudot, Braille,

Alfabeto Braille



a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	x	y	z	ç	é	á	è	ú
â	ê	ì	ô	ù	à	ĩ	ü	õ	w
í	ó	ã	sinal numérico	-	'	—	...	grifo	maíuscula
,	;	:	.	\$	?	!	()	"	*
1	2	3	4	5	6	7	8	9	0
caixa alta									

Ex.: representação de texto a norma americana ASCII (7 bits)



ASCII: American Standard Code for Information Interchange

Tabela ASCII 7 bits

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2	SP	!	“	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

H	e	l	l	o		w	o	r	l	d	!
48	65	6c	6c	6f	20	77	6f	72	6c	64	21

Ex.: codificação universal de texto, UTF-8 no Unicode



- Baudot, Braille, ASCII, Unicode, (UTF-8)

Unicode Transformation Format

binary	hex	decimal	notes
00000000-01111111	00-7F	0-127	US-ASCII (single byte)
10000000-10111111	80-BF	128-191	Second, third, or fourth byte of a multi-byte sequence
11000000-11000001	C0-C1	192-193	Overlong encoding: start of a 2-byte sequence, but code point ≤ 127
11000010-11011111	C2-DF	194-223	Start of 2-byte sequence
11100000-11101111	E0-EF	224-239	Start of 3-byte sequence
11110000-11110100	F0-F4	240-244	Start of 4-byte sequence
11110101-11110111	F5-F7	245-247	Restricted by RFC 3629: start of 4-byte sequence for codepoint above 10FFFF
11111000-11111011	F8-FB	248-251	Restricted by RFC 3629: start of 5-byte sequence
11111100-11111101	FC-FD	252-253	Restricted by RFC 3629: start of 6-byte sequence
11111110-11111111	FE-FF	254-255	Invalid: not defined by original UTF-8 specification

Representação da informação num computador (3)



Como se representa a informação?

- com binary digits!

Tipos de informação a representar:

- números (para cálculo)
 - » inteiros: S+M, Compl. p/ 1, Compl. p/ 2, Excesso
 - » reais (*fp*): norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
 - » imagens fixas: BMP, JPEG, GIF, PNG, ...
 - » audio-visuais: AVI, MPEG/MP3, ...
- código para execução no computador

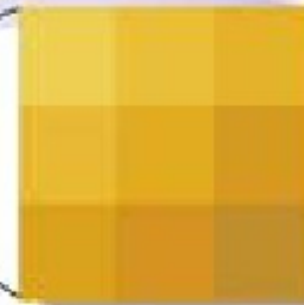
Ex.: representação de uma imagem em bitmap



You can create a 24-bit image in a graphics program such as Paint.



A graphics program saves the image line by line, from the bottom to the top.



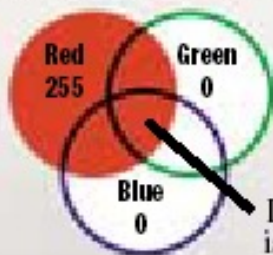
Each of the pixel's three-color values, RGB (red-green-blue), are read from left to right.

R 250 G 210 B 94	R 244 G 196 B 69	R 238 G 182 B 51
R 242 G 190 B 60	R 235 G 176 B 42	R 222 G 160 B 26
R 228 G 167 B 27	R 218 G 153 B 17	R 201 G 148 B 53

A graphics program translates the RGB values into palette values. The palette values are a software-specific decision; each program's values are different.

Forming A Pixel

A pixel is the smallest part of an image that a computer's monitor can control. Each pixel consists of three colors: red, green, and blue. Each of the three colors is assigned a value that shows its intensity; the values are from 0 to 255. You can think of each value as a percentage. For example, 127 has a 50% intensity. These are known as the RGB values.



Each palette value, a hexadecimal value in this case, is stored in the same order as displayed in the image.

FAD25E	F4C345	EEB633
F2BE3C	EBB02A	DEA01A
E4A71B	DA9911	C99435

The pixel values are stored in the bit-mapped file in the same width and depth as the original image.



Representação da informação num computador (4)



Como se representa a informação?

- com binary digits!

Tipos de informação a representar:

- números (para cálculo)
 - » inteiros: S+M, Compl. p/ 1, Compl. p/ 2, Excesso
 - » reais (*fp*): norma IEEE 754
- textos (caracteres alfanuméricos)
 - » Baudot, Braille, ASCII, Unicode, ...
- conteúdos multimédia
 - » imagens fixas: BMP, JPEG, GIF, PNG, ...
 - » audio-visuais: AVI, MPEG/MP3, ...
- código para execução no computador
 - » noção de *instruction set*

Ex.: representação de código para execução num PC



```
int x = x+y;
```

```
addl 8(%ebp), %eax
```

Idêntico à expressão
x = x + y

```
0x401046:    03 45 08
```

- Código numa linguagem de programação
 - somar 2 inteiros
- Código numa linguagem mais próxima do processador
 - somar 2 inteiros (de 4-bytes)
 - operandos:
 - x: no registo `eax`
 - y: na memória em `[(ebp)+8]`
- Código “objecto” (em hexadecimal)
 - instrução com 3-bytes
 - na memória em `0x401046`