

Sistema Computacional:

1

Níveis de um computador:

Nível 6 → user

Nível 5 → problem oriented language

) compiler

Nível 4 → assembly language

) assemble

Nível 3 → operating system machine

) operating system

Nível 2 → instruction set architecture

) direct execution or interpretation
(microprogram)

Nível 1 → microarchitecture

) hardware

Diferenças entre compiler, interpreter e machine virtual machine

Compiler → compila alto para baixo (rápido)

interpreter → interpreta e compila (mais lento)

virtual machine → emulador de um sistema de computador que é executado numa máquina física simulando um computador real.

Big-endian, little-endian computing

Big-endian → os valores menos significativos do número vêm pelo o maior endereço da memória.

Ex:	00010010 00110100 01010110	byte 2	byte 0
Memória	big	little	<u>01111000</u>
132104 →	00010010	01111000	menos significativo

132105 → 00110100 | 01010110

132106 → 01010110 | 00110100

132107 → 01111000 | 00010010

Little-endian → os valores mais de "1" vêm para o menor endereço

Representação de informações:

128x128 pixels, 3 channels, 256 different tones 1 channel → 8 bits $\rightarrow 8 \times 3$
↳ 3 bytes
 $128 \times 128 = 16384$ pixels

$3 \times 16384 \rightarrow 3$ bytes

KB Kilobyte $\rightarrow 10^3$
 MB megabyte $\rightarrow 10^6$
 GB gigabyte $\rightarrow 10^9$
 TB terabyte $\rightarrow 10^{12}$

K.B	Kilobyte	2^{10}	1024
M.B	megabyte	2^{30}	
G.B	gigabyte	2^{30}	
T.B	terabyte	2^{100}	

(2)

Sample resolution

E o número de bits usados para representar o valor para cada sample

$$S = f \times R \times T$$

$f \rightarrow$ sampling rate (Hz)

$R \rightarrow$ is the sample resolution e numero de

bits (bits)

(\Rightarrow tipo (n))

Representação de números

4 682 na base 10

$$4682 = 4 \times 10^3 + 6 \times 10^2 + 8 \times 10^1 + 2 \times 10^0$$

Passagem de valores para binário

$$4682,05 = (\dots) + 0 \times 0,1 + 5 \times 0,01$$

1 \rightarrow Dividir por 2

2 \rightarrow E o resto da divisão

3. Ler do baixo para cima

0 - 0₁₀

1 - 1₁₀

2 - 0₁₀

3 - 1₁₀

4 - 0₁₀

5 - 1₁₀

6 - 0₁₀

7 - 1₁₀

8 - 0₁₀

9 - 1₁₀

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0~~

~~0 1 0 0 0 0 0 0</del~~

3

Complementos negativos.

1 → jazer o número em binário

2 → trocar os 1 pelo 0 e vice versa

(o primeiro valor do binário diz se é negativo ou positivo)

3 → Somar 1

$$\begin{array}{r}
 3.22) \text{ a) } 00\ 111\ 00_2 \\
 \begin{array}{r}
 \begin{array}{r}
 11\ 00\ 11 \\
 + 00\ 00\ 00\ 1 \\
 \hline 11\ 00\ 100
 \end{array} & \xrightarrow{\quad \text{1º complemento} \quad} \\
 & \xrightarrow{\quad \text{2º complemento} \quad}
 \end{array}
 \end{array}
 \left. \begin{array}{l}
 \text{Somar é tipo normal mas} \\
 \text{se o valor chegar ao valor da base} \\
 \text{subtra-se a sua unidade}
 \end{array} \right\}$$

$$\left\{ (10^m - 1) - N \right\}$$

9) complement of the number
 $m \rightarrow$ number of digits
 $N \rightarrow$ the number

$$\begin{array}{r}
 10000000\ 1236 \\
 \xrightarrow{\quad \text{L} \quad} (10^4 - 1) - 1236 = 10000 - 1 - 1236 \\
 = 10000 - 1237 = 8763
 \end{array}$$

3.22)

a) $+18_{10}$ ① \rightarrow sinal + magnitude

$$18_{10} = 10010_2$$

$$\begin{array}{r}
 \boxed{0} 0 0 1 0 0 1 0_2 \\
 + : 0 \\
 \hline
 \end{array}$$

② Comp/1 \rightarrow acrescentar parâs de zeros para 8 bit

Como o mº é positivo o complemento é o próprio

$$\begin{array}{r}
 \underline{0}\ \underline{0}\ \underline{0}\ 1\ \underline{0}\ \underline{0}\ 1\ 0_2
 \end{array}$$

③ Comp/2

$$\begin{array}{r}
 \underline{0}\ \underline{0}\ \underline{0}\ 1\ \underline{0}\ \underline{0}\ 1\ 0_2
 \end{array}$$

b) $+121_{10}$

$$\begin{array}{r}
 121\ \underline{1} 2 \\
 1\ \underline{6} 0
 \end{array}
 \quad
 \begin{array}{r}
 60\ \underline{1} 2 \\
 0\ \underline{3} 0
 \end{array}
 \quad
 \begin{array}{r}
 30\ \underline{1} 2 \\
 0\ \underline{1} 5
 \end{array}
 \quad
 \begin{array}{r}
 15\ \underline{1} 2 \\
 1\ \underline{7}
 \end{array}
 \quad
 \begin{array}{r}
 7\ \underline{1} 2 \\
 1\ \underline{3}
 \end{array}
 \quad
 \begin{array}{r}
 3\ \underline{1} 2 \\
 1\ \underline{1}
 \end{array}
 \quad
 \begin{array}{r}
 1\ \underline{1} 2 \\
 1\ \underline{0}
 \end{array}$$

$$121_{10} = 1111001$$

$$\begin{array}{r}
 \underline{0}\ \underline{1}\ \underline{1}\ \underline{1}\ \underline{1}\ \underline{0}\ \underline{0}\ 1
 \end{array}$$

Cp1 signal + Cp2 signal

c) -33

4

(apenas =)

$$\begin{array}{r} 33 \\ \times 2 \\ \hline 16 \end{array}$$

$$\begin{array}{r} 16 \\ \times 2 \\ \hline 08 \end{array}$$

$$\begin{array}{r} 8 \\ \times 2 \\ \hline 04 \end{array}$$

$$\begin{array}{r} 4 \\ \times 2 \\ \hline 02 \end{array}$$

$$\begin{array}{r} 2 \\ \times 2 \\ \hline 01 \end{array}$$

$$\begin{array}{r} 1 \\ \times 2 \\ \hline 00 \end{array}$$

$$33_{10} = 100001_2$$

$$11100001_2$$

$$0001110_2$$

$$0001110_2$$

$$R: 0111111, 2^5 + 2^3 + 2^2 + 2^1 + 2^0 = 16 + 8 + 4 + 2 + 1 =$$

Complemento de 2

- Números positivos: bit de sinal 0, a magnitude representada na forma direta
- Números negativos: bit de sinal 1, a magnitude representada na forma de complemento de 2

Ex: $\underbrace{00110}_{\text{mag}} = +6$ mas $1\underbrace{0110}_{\text{mag}}$

não é 6 porque é na forma do complemento de 2

(se não quiséssemos -6)

$$10110 \rightarrow -10$$

$$\begin{array}{r} 01001 \\ -00001 \\ \hline 01010 \end{array} \rightarrow +10$$

$$\begin{array}{r} 11001 \\ +00001 \\ \hline 00111 \end{array} \rightarrow +7$$

$$0 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 16 + 8 + 4 + 2 + 1 = 31$$

de + para - é juntar o depois +1 e juntar negativo

~~+~~

é mudar o depois +1 e juntar negativo

c) -33

$$33_{10} = 100001_2$$

$$100001_2$$

0100001

$$\begin{array}{r} 0100001 \\ -0000001 \\ \hline 0100000 \end{array}$$

$$1011110$$

$$= 2^7 + 2^6 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$$

$$1011110_2$$

Bit de sinal negativo

$$+2$$

$$2^4 + 2^3 + 2^2 + 2^1 + 2^0 = 16 + 8 + 4 + 2 + 1$$

m:

$$\begin{array}{r} 7 \\ -5 \\ \hline 2 \end{array}$$

$$\begin{array}{r} 01001 \\ 00010 \\ \hline 10011 \end{array}$$

Confirmado

$$2^7 = 31 \rightarrow \text{de } 31 \text{ juntar } 0 \text{ (ultimo)} \text{ de } 2 \text{ de } 33_0 \text{ vale 1}$$

$$\begin{array}{r} 00100 \\ +0101 \\ \hline 00101 \end{array} \quad 0101 = 5 \quad \checkmark$$

(5) Overflow

$$\begin{array}{r} +9 \\ +8 \\ \hline +17 \end{array}$$

$$\begin{array}{r} 01001 \\ 01000 \\ \hline 00001 \end{array}$$

4 bits so degade to 15

lose

$$\begin{array}{r} 1111 = 2^3 + 2^2 + 2^1 + 2^0 \\ = 8 + 4 + 2 + 1 \end{array}$$

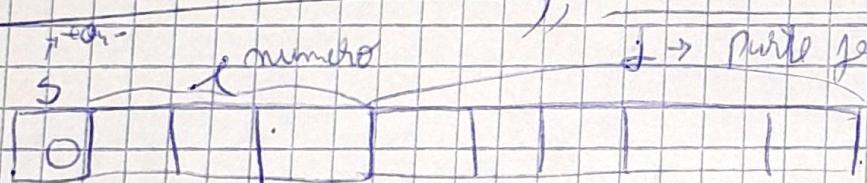
$$= 15_{10}$$

clar megalho

$$\begin{array}{r} 001001 \\ 001000 \\ \hline 010001 \end{array} \rightarrow +17 \text{ caused 1 bit overflow}$$

Cateto (±0.001) se for 4-bit decimal & logaritmo

$$\begin{array}{r} \text{num} \\ \text{ven 1} \\ \text{fim nro 0} \end{array} \left\{ \begin{array}{r} 01001 \\ +11100 \\ \hline 0101 \end{array} \right.$$



$e \rightarrow$ parte fraccional

Reservado
melhor
exemplo
parte

$$21,5 = 10101,1_2 = 10101,1_2 \times 2^4 = 1,01011_2 \times 2^4$$

$\rightarrow s \rightarrow$ sign bit (+; -)

$e \rightarrow$ exponent part (representado em base de 2)

$f \rightarrow$ mantissa fractional part of the number

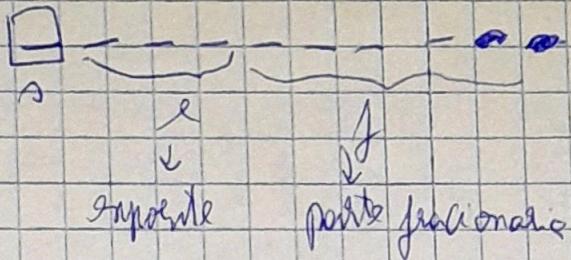
Number normalized

$$v = (-1)^s \cdot (1+f) \cdot 2^{e-bias}$$

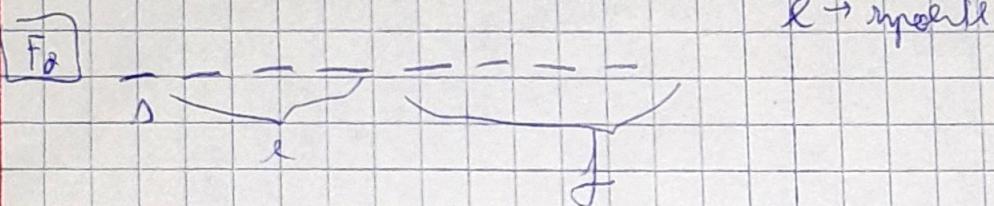
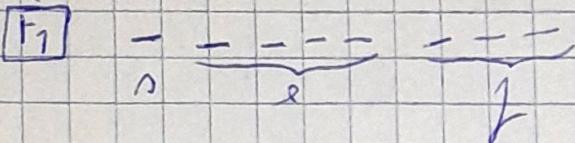
$$\text{Bins} = 2^{K-1} - 1 \quad K = \text{nº de bits de } R$$

Bits:

$$+ 3.14 \times 10^6$$



Exponente:



bit mantisa no true numero

$$11001_2 = \underbrace{1,1001}_{\text{esta normalizada}} \times 2^4$$

$$\text{bias} = 2^{n-1} - 1 = 2^{7-1} - 1 \quad n \rightarrow \text{numero de bits do exponente}$$

$V_m = (-1)^s \times 1.f_0 \times 2^{e-bias}$

$$\text{bias} = 2^{4-1} - 1 = 7$$

$$V_m = (-1)^3 \times 1.f \times 2^{7-7}$$