– tbl 2 –

Table 2.1 e parágrafo anterior

2.3 / 2.5 / 2.6

Byte Unit Symbol: B

For multiple-byte units there are systems of base 2 and 10

10^3 - kB		2^{10} - KiB
10^6 - MB		2^{20} - MiB
10^9 - GB		2^{30} - GiB
10^{12} - TB	vs	2^{40} - TiB
10^{15} - PB		2^{50} - PiB
10^{18} - EB		2^{60} - EiB
10^{21} - ZB		2^{70} - ZiB

/

2.3 - Textual Information

In a computer, each alphanumeric character is represented by a bit pattern. Most used character encoding → ASCII

↓ used in

computers, telecommunications equipment

each char. - 7 bit string

ASCII has: - 59 printable symbols

- 33 non-printing controls

In practice, one ASCII symbol is stored in a byte with the leftmost bit usually set to 0.

Some manufacturers used a 1 instead, having 128 additional chars.

Null character: 0000 0000 "\0"

Modern computers use Unicode (handles most of the world's writing)

Unicode can be implemented by different character encoding. The dominant is UTF-8

2.5 - Image

Raster image = Bitmap image (array of points)

↓

each point is a pixel

↳ Rep. by a binary pattern

Black and White: 1 and 0 (w.)

Colors: combinations of 1s and 0s

Size of the picture → part of the metadata

If 2 bits are used → 4 colours ↳ e.g: date, time ...

↳ any colour attributed to 00, 01, 11, 10

Many images use the RGB colour model. In wich all channels at 0 → Black

all channels at 255 → White

The nº of bits used to store each pixel is the Colour Depth.

The quality of the image comes from the Image Resolution (how close the pixels are to each other. Measured in dpi (nº of dots per inch)

Vector image - uses scalable shapes (eg. lines and curves). Is more efficient than bitmap imgs. at storing, since it doesn't store every pixel.

Biggest advantage: can be enlarged or reduced w/o losing quality.

2.6 - Audio

Same dos apontamentos do cap. 3.

SIGN - MAGNITUDE:

Bits that represent the number in decimal and add a bit to the left to indicate the sign.

0 → positive

1 → negative

$+120_{10}$ → 01111000_2

-120_{10} → 11111000_2

It is necessary to know how many bits are used

IF YOUR number hasn't the necessary nº of bits, ADD 0s TO THE LEFT

1100_2, negative if 4 bits are being used

positive if 6 bits are being used, 001100_2

TO ADD THEM:

SINAIS IGUAIS → ADD THE MAGNITUDE, SAME SIGN

SINAIS ≠ → DETERMINE WICH ONE HAS A LARGER MAGNITUDE, AND THEIR SINAL WILL BE THE RESULT'S SIGN.

THE MAGNITUDE WILL BE OBTAINED BY SUBTRAIR O MAIS PEQUENO AO MAIOR.

ONE'S - COMPLEMENT

165 - 43?

SUBTRAIR DECIMAIS? RETIRAR 2º NRº A 999

999 - 43 = 956

165 - 43 = 165 + 956 = (1)121

↳ isto é o CARRY

DADO UM Nº V, DE BASE r, C/ n dígitos: $r^n - 1 - V$

QUANDO CARRY: 165 - 43 = 121 + 1 = 122

ONE'S COMPLEMENT FOR BINARY Nos

" " OF 011001_2 AND 111001_2

$011001_2 = 1\times16 + 1\times8 + 1\times1 = +25_{10}$

$111001_2 = 1\times(-31) + 1\times16 + 1\times8 + 1\times1 = -6_{10}$

ONE'S COMPLEMENT FOR BINARY: SUBTRACTS FROM ALL ONES

O.C. OF 0101_2 IS $1111 - 0101 = 1010_2$ (OR) CHANGING 0 to 1 and 1 to 0

24-10? $+24_{10} = 00011000_2$

$-10_{10} = 11110101_2$

1← 1 1 1

00011000 $(+24_{10})$

11110101 $+(-10_{10})$

00001101

+ 1

00001110 $(+14_{10})$

0← 1 1 1

00001010 $(+10_{10})$

11100111 $+(-24_{10})$

11110001

+ 0

11110001 (-14_{10})

TWO'S - COMPLEMENT

AKA RADIX COMPLEMENT

$r^n - V$, logo complemento de 43 é $10^3 - 43 = 957$

$011001_2 = 1\times16 + 1\times8 + 1\times1 = 25_{10}$

$111001_2 = 1\times(-32) + 1\times16 + 1\times8 + 1\times1 = -7_{10}$

↳ pq o MSB é 1

TC can be calculated by OC + 1, so

PARA CALCULAR: FLIP BITTS, ADD 1

! ANY CARRIES INVOLVING THE MSB ARE DISCARTED

! ONLY NEGATIVE NUMBERS NEED TC!

24-10: 1 ← 1 1 1 CARRIES

0 0 0 1 1 0 0 0	$(+24_{10})$
1 1 1 1 0 1 1 0	$+(-10_{10})$
0 0 0 0 1 1 1 0	$(+14_{10})$

THIS BEING DISCARTED DIDN'T CAUSE ANY ERROR

IF 2 POSITIVES ARE ADDED AND THE RESULT IS NEGATIVE } OVERFLOW

IF 2 NEGATIVES ARE ADDE AND THE RESULT IS POSITIVE }

1 ⇐ carries		1 1 1 1 ⇐ carries	
0 1 0 1 1 0 1 0	$(+90_{10})$	0 1 0 1 1 0 1 0	$(+90_{10})$
0 0 0 1 0 1 0 0	$+(+20_{10})$	0 0 1 1 0 0 1 0	$+(+50_{10})$
0 1 1 0 1 1 1 0	$(+110_{10})$	1 0 0 0 1 1 0 0	overflow
1 1 1 1 1 ⇐ carries		1 1 1 1 ⇐ carries	
0 1 0 1 1 0 1 0	$(+90_{10})$	1 0 1 0 0 1 1 0	(-90_{10})
1 1 1 0 1 1 0 0	$+(-20_{10})$	1 1 0 0 1 1 1 0	$+(-50_{10})$
0 1 0 0 0 1 1 0	$(+70_{10})$	0 1 1 1 0 1 0 0	overflow

Excess Representations

= ao número dado − bias

$\downarrow$

2^{n-1}

EXEMPLO: SE b=3

excesso de 0000 é -3

" 0001 é -2

" 0010 é -1

...