

REPRESENTAÇÃO BINÁRIA E UNIDADES DE MEDIDA

Introdução à Ciência da Computação – ICC0001

Cronograma

- Representação de números negativos
- Representação de cores e imagens
- Operações aritméticas de números binários
- Bit X Byte
- Unidades de Medida (bits, bytes, KB, MB, GB, TB, Hz)

Representação de números negativos

- Existem formas diferentes de se representar um número negativo em binário
 - Em binário porque já estamos falando em como os computadores guardam a informação.
- Tudo depende do formalismo que é adotado
 - Magnitude com sinal
 - Complemento a um
 - Complemento a dois
 - Excesso 2^{m-1}
- **IMPORTANTE: Saiba quantos bits compõem o número representado.**

Representação de números negativos

- **Magnitude com sinal**
- O bit mais significativo é o sinal do número
 - O bit 0 (zero) indica número positivo
 - O bit 1 (um) indica número negativo
- Os demais representam a magnitude absoluta
- Exemplo:
 - **1**0001 = -1
 - **1**1001 = -9
 - **0**1001 = +9
- Duplicidade do zero: $+0_{10} = 0000\ 0000_2$ e $-0_{10} = 1000\ 0000_2$

Representação de números negativos

- **Complemento a um**
- Também tem um bit de sinal
 - 0 para +
 - 1 para –
- Para tornar um número negativo deve-se substituir todos os **zeros** por **uns** e todos os **uns** por **zero**
- Exemplo (binário de 8 bits):
 - $+9_{10}$ é igual a $0000\ 1001_2$
 - -9_{10} é igual a $1111\ 0110_2$
- **Duplicidade do zero:** $+0_{10} = 0000\ 0000_2$ e $-0_{10} = 1111\ 1111_2$
- **ESTA REPRESENTAÇÃO É OBSOLETA**

Representação de números negativos

- **Complemento a dois**

- Também tem um bit de sinal

- 0 para +
- 1 para –

Adição em binário:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ e vai } 1 \text{ (10)}$$

$$1 + 1 + 1 = 1 \text{ e vai } 1 \text{ (11)}$$

- Para tornar um número negativo deve-se substituir todos os **zeros** por **uns** e todos os **uns** por **zero** e adicionar 1 ao bit **menos** significativo

- Exemplo (binário de 8 bits):

- 9_{10} é igual a 0000 1001
- -9_{10} é igual a 1111 0111

- Caso zero: $0_{10} = 0000\ 0000_2 \Rightarrow 1111\ 1111_2 + 1_2 \Rightarrow 0000\ 0000_2$

Representação de números negativos

- **Excesso 2^{m-1}**
- Cada número é armazenado como a soma dele com 2^{m-1} , onde m é o número de bits utilizados. Assim um número de 8 bits, por exemplo, é denominado como excesso 128 ($=2^7$).
- Exemplo:
 - 9_{10} é igual a $9 + 128 = 137 = 1000\ 1001$
 - -9_{10} é igual a $-9 + 128 = 119 = 0111\ 0111$
- Interessante: é idêntico ao complemento de dois, o que muda é o sinal de magnitude

Representação de cores

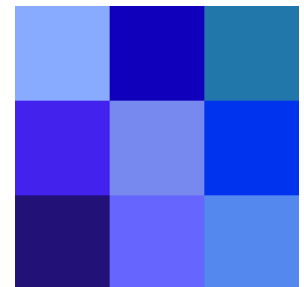
- Modelo preto e branco

- Utiliza-se apenas duas cores (com cor, sem cor)
- Representação fácil por apenas um bit



- Modelo monocromático

- Utiliza-se várias tonalidades de uma mesma cor



- Modelo colorido

- Todas as cores podem ser representadas pela misturas das três cores primárias (VERMELHO, VERDE E AZUL) em um fundo branco.

Representação de cores

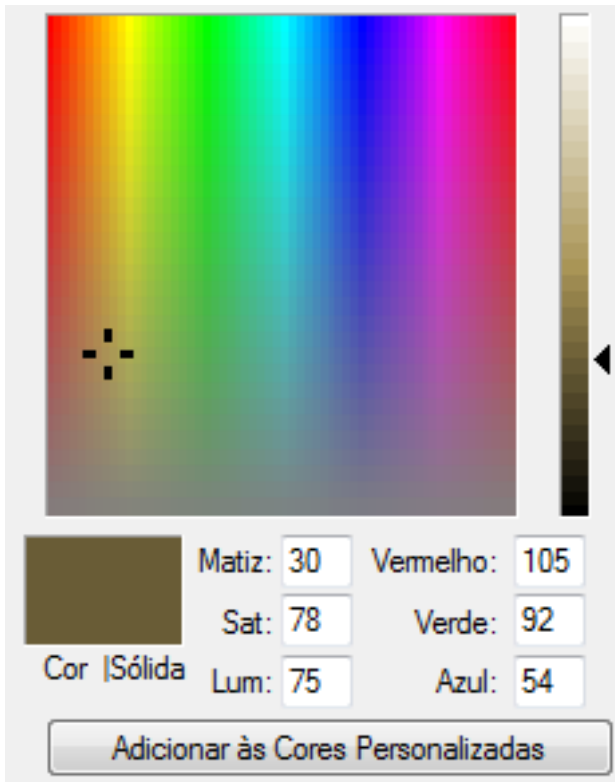
Exemplos:

Amarelo => Vermelho + Verde

Rosa => Vermelho + Azul

Marrom => Vermelho₍₊₊₎ + Verde₍₊₎ + Azul₍₊₎

Usamos um valor para indicar a quantidade de cada cor primária :



Paleta de cores do Microsoft Paint

Marrom

Vermelho: 105 / 255 (41%)

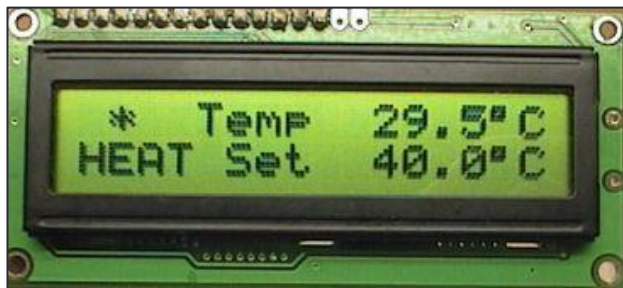
Verde: 92 – 255 (36%)

Azul: 54 – 255 (21%)

* costuma-se utilizar 256 tonalidades para cada cor primária

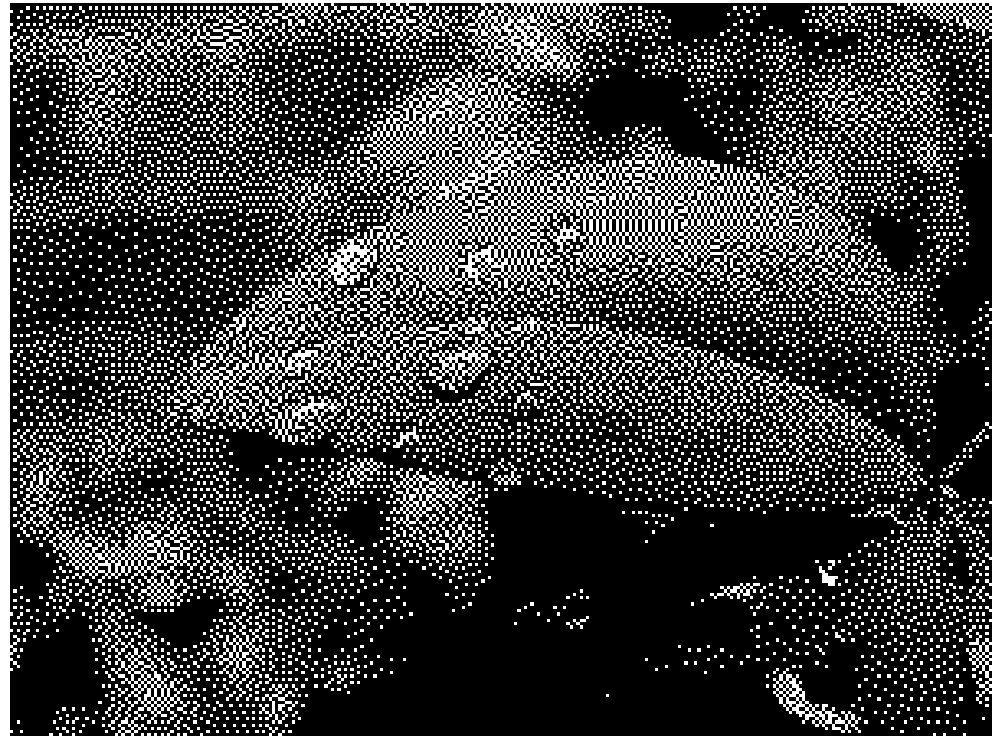
Representação de imagens

- Semelhante ao bordado em toalhas:
 - Existe um tecido com vários quadradinhos onde pode-se ou não passar um fio de alguma cor
 - Desenhos/palavras são feitas da combinação de fios de várias cores
- Monitores e outros aparelhos de display utilizam o mesmo princípio (mapa de bits ou bitmap)



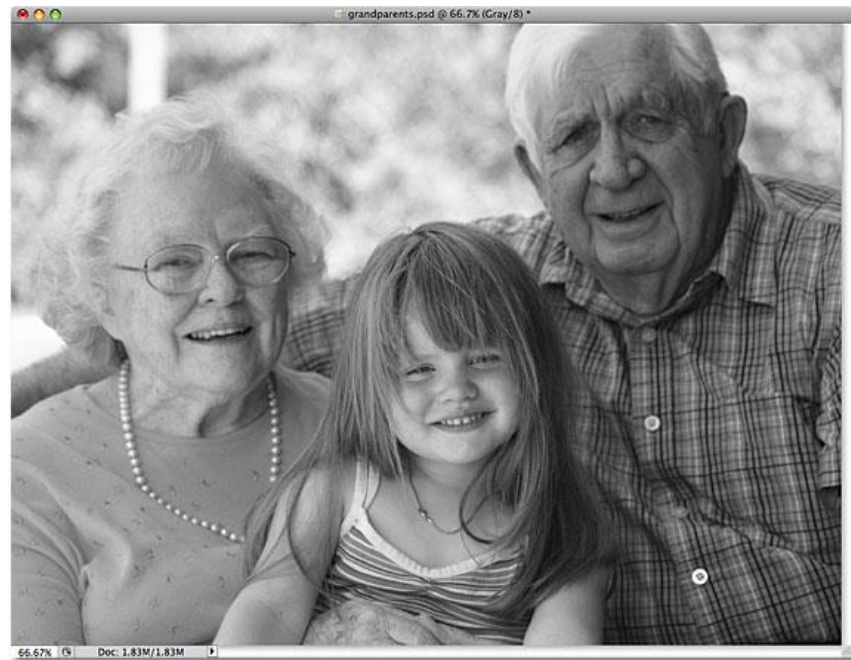
Representação de imagens

- Preto e branco



Representação de imagens

- **Modelo monocromático**



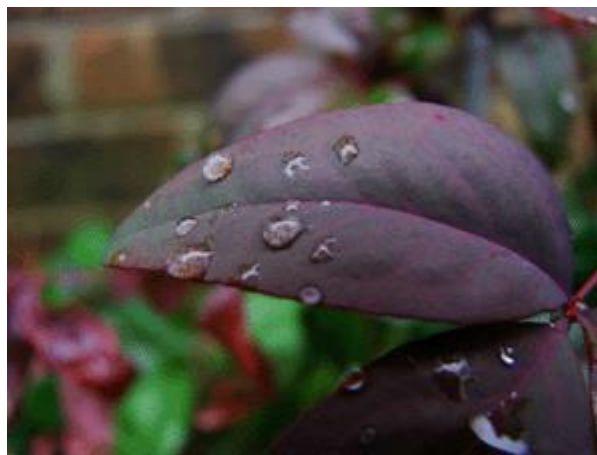
Representação de imagens

- **Modelo colorido**
(diversas profundidades de cores)

4 bits



8 bits



Representação de Letras / Símbolos

- Cada número em binário representa uma letra
- Tabela ASCII ==>
- Unicode
(<http://unicode-table.com/en/>)

0		32	?	64	@	96	`	128	Ç	160	ã	192	Ł	224	ô
1	☺	33	!	65	A	97	a	129	û	161	î	193	ł	225	ß
2	♥	34	"	66	B	98	b	130	é	162	ó	194	Ł	226	ð
3	♦	35	#	67	C	99	c	131	â	163	û	195	└	227	õ
4	♠	36	\$	68	D	100	d	132	ä	164	ñ	196	┌	228	ö
5	♣	37	%	69	E	101	e	133	à	165	ñ	197	└	229	ø
6	♠	38	&	70	F	102	f	134	á	166	ä	198	└	230	µ
7	●	39	'	71	G	103	g	135	ç	167	ö	199	└	231	þ
8	■	40	(72	H	104	h	136	ê	168	ï	200	└	232	þ
9		41)	73	I	105	i	137	ë	169	®	201	└	233	ó
10		42	*	74	J	106	j	138	è	170	¬	202	└	234	ü
11	♂	43	+	75	K	107	k	139	í	171	½	203	└	235	ù
12	♀	44	,	76	L	108	l	140	î	172	¾	204	└	236	ý
13		45	-	77	M	109	m	141	ï	173	i	205	└	237	ÿ
14	ß	46	.	78	N	110	n	142	ä	174	«	206	└	238	˘
15	☼	47	/	79	O	111	o	143	å	175	»	207	└	239	˙
16	▶	48	0	80	P	112	p	144	É	176	»	208	└	240	˚
17	◀	49	1	81	Q	113	q	145	æ	177	»	209	└	241	±
18	†	50	2	82	R	114	r	146	Æ	178	»	210	└	242	≡
19	‡	51	3	83	S	115	s	147	ö	179	»	211	└	243	¼
20	¶	52	4	84	T	116	t	148	õ	180	»	212	└	244	½
21	§	53	5	85	U	117	u	149	ö	181	»	213	└	245	¾
22	■	54	6	86	V	118	v	150	û	182	»	214	└	246	÷
23	±	55	7	87	W	119	w	160	á	183	»	215	└	247	˘
24	↑	56	8	88	X	220	■	161	í	184	»	216	└	248	˙
25	↓	57	9	89	Y	221	■	162	ó	185	»	217	└	249	˚
26	→	58	:	90	Z	222	■	163	û	186	»	218	└	250	˙
27	←	59	;	91	[223	■	164	ñ	187	»	219	└	251	˙
28	└	60	<	92	\	224	ö	165	ñ	188	»	220	└	252	˙
29	└	61	=	93]	225	þ	166	ä	189	»	221	└	253	˙
30	▲	62	>	94	^	226	ö	167	ö	190	»	222	└	254	˙
31	▼	63	?	95	_	227	ö	168	ç	191	»	223	└	255	˙

Operações aritméticas

- Soma
- Subtração
- Multiplicação
- Divisão

Soma binária

- A operação de soma entre números binários é bastante simples e análoga à operação com números decimais.
- Usa-se o transporte... (vai 1)

Adição em decimal:

$0 + 0 = 0$
 $0 + 1 = 1$
 $0 + 2 = 2$
 ...
 $0 + 9 = 9$
 $1 + 0 = 1$
 ...
 $1 + 9 = 0$ e vai 1
 $2 + 0 = 2$
 ...
 $2 + 9 = 1$ e vai 1
 ...

Adição em binário:

$0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0$ e vai 1

Exemplo:

$$\begin{array}{r} 1010 \\ 0101 + \\ \hline 1111 \end{array}$$

$$\begin{array}{r} 1111 \\ 1011 \\ 0101 + \\ \hline 10000 \end{array}$$

$$\begin{array}{r} 1111 \\ 1111 \\ 0101 + \\ \hline 10100 \end{array}$$

Subtração binária

- Também análogo ao decimal. Usamos a ideia de empréstimo.

Sub em binário:

$$0 - 0 = 0$$

$$1 - 0 = 1$$

$$1 - 1 = 0$$

$$0 - 1 = 1 \text{ e empresta } 1$$

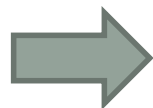
Exemplo:

$$\begin{array}{r} ^{-1} ^{-1} \\ 1010 \\ 0101 - \\ \hline 0101 \end{array}$$

Subtração binária

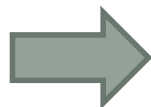
- Outra forma de realizarmos a subtração é utilizar a conversão com a representação de complemento de dois.
- Exemplo Decimal:
 - $25 - 25 = 25 - (+25) = 25 + (-25) = 00$
 - $25 - (10) = 25 - 10 = 15$
- Em binário (Complemento a dois):

$$\begin{array}{r} 011001 \\ -011001 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +100111 \\ \hline 1000000 \end{array}$$

$$\begin{array}{r} 011001 \\ -001010 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +110110 \\ \hline 1001111 \end{array}$$

Desprezar

Subtração binária

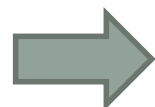
- Outra forma de realizarmos a subtração é utilizar a conversão com a representação de complemento de dois.

- Exemplo Decimal:

- $25 - 25 = 25 - (+25) = 25 + (-25) = 00$
- $25 - (10) = 25 - 10 = 15$

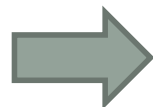
- Em binário (Complemento a dois):

$$\begin{array}{r} 011001 \\ -011001 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +100111 \\ \hline 1000000 \end{array}$$

$$\begin{array}{r} 011001 \\ -001010 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +110110 \\ \hline 1001111 \end{array}$$

Nós desprezamos o *carry*, mas ele é importante. O número **1** indica que o resultado é **positivo** e o número **0** indica que é **negativo**.



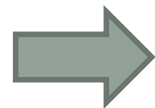
Desprezar

Subtração binária

- Outra forma de realizarmos a subtração é utilizar a conversão com a representação de complemento de dois.
- Exemplo Decimal:
 - $25 - 25 = 25 - (+25) = 25 + (-25) = 00$
 - $25 - (10) = 25 - 10 = 15$
- Em binário (Complemento a dois):

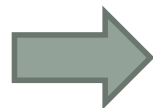
Quando o número de bits resultantes é maior do que o número de bits utilizados é dito que ocorreu **overflow**

$$\begin{array}{r} 011001 \\ -011001 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +100111 \\ \hline 1000000 \end{array}$$

$$\begin{array}{r} 011001 \\ -001010 \\ \hline \end{array}$$



$$\begin{array}{r} 011001 \\ +110110 \\ \hline 1001111 \end{array}$$

Desprezar



Multiplicação Binária

- Análogo ao sistema decimal

Mult. em binário:

$$0 \times 0 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

$$0 \times 1 = 0$$

Exemplo:

$$\begin{array}{r} 1010 \\ \times 10 \\ \hline 0000 \\ 1010+ \\ \hline 10100 \end{array}$$

Divisão Binária

- Adivinhem... Análogo ao decimal

$$\begin{array}{r}
 101100 \overline{) 100} \\
 \underline{- 100} \\
 0011 \\
 \underline{- 000} \\
 110 \\
 \underline{- 100} \\
 0100 \\
 \underline{- 100} \\
 0000
 \end{array}$$

Bit X Byte

- Bits são utilizados para representar o tamanho das *palavras* do computador, utilizadas pelas instruções do processador
 - 8 bits = 1 byte
 - 16 bits = 2 bytes = word
 - 32 bits = 4 bytes = Dword
 - 64 bits = 8 bytes = Qword
- Bits são normalmente utilizado para calcular velocidades
 - 56 bits por segundo = 56 bps
 - 10 megabits por segundo = 10 Mbps

Bit X Byte

- **Bytes** são frequentemente utilizados para representar espaços de armazenamento em mídias:
 - 1 B = 1 byte
 - 1 KB (kilobyte) = 10^3 Bytes = 1000 bytes
 - 1 MB (megabyte) = 10^6 Bytes = 1000 Kbytes
 - 1 GB (gigabyte) = 10^9 Bytes = 1000 Mbytes
 - 1 TB (terabyte) = 10^{12} Bytes = 1000 Gbytes
- Discussão entre usar potências de 2 (notação computacional) ou potências de 10 (notação de fácil assimilação para leigos)

Bit X Byte

- **Bytes** são frequentemente utilizados para representar espaços de armazenamento em mídias:
 - 1 B = 1 byte
 - 1 KiB = 2^{10} Bytes ≈ 1024 bytes
 - 1 MiB = 2^{20} Bytes ≈ 1024 KiB
 - 1 GiB = 2^{30} Bytes ≈ 1024 MiB
 - 1 TiB = 2^{40} Bytes ≈ 1024 GiB
- Kilo Binary Byte (KiB) – **kibibyte**
- Mega Binary Byte (MiB) – **mebibyte**
- Giga Binary Byte (GiB) – **gibibyte**



Bit X Byte

- Fabricantes de discos magnéticos e ópticos perceberam que era mais fácil utilizar ordens de grandeza decimais para medir seus produtos e torná-los mais **atraentes**:
 - Um disco rígido de 320 GB, na verdade tem 298 GiB
 - Um DVD de 4,7 GB possui 4,37GiB
- As pessoas gostam de pluralizar as ordens de grandeza:
 - “Pendrive de 4 **Gigas**!”
 - “HD de 2 **Teras**”



Bit X Byte

- Para determinar medida de capacidade de armazenamento utiliza-se **Bytes**!
- Na transmissão de dados entre dispositivos normalmente utiliza-se a medição em **bits**! (bps, Kbps, Mbps)
 - OBS: uma exceção comum é para transferência de arquivos que costuma ser medida em Bytes / s.

“Minha internet é de 10**Mb**ps!”

“Meu arquivo está sendo copiado a 2,5 MB/s”

Bit X Byte

- Quando falamos em transferência de dados existe ainda a questão do uso da **codificação 8b/10b**:
 - 8 bits de dados + 2 bits de paridade (controle / segurança)

Logo, uma conexão de internet de **10Mbps** seria equivalente a **1MB/s** de dados efetivos e não **1.25MB/s**

CURIOSIDADE: provedores de internet não são obrigados por lei a fornecer a velocidade máxima contratada, mas apenas uma porcentagem do que foi negociado.
(2014 => 40%)

Bit X Byte

- Exemplo de transmissão:

Tamanho do arquivo: 3 GB

Velocidade de transmissão: 2Mbps

(cod. 8b/10b) → 2 Mbps = 0,2 MB/s

Hertz (Hz)

- É uma unidade de medida para expressar frequência (s^{-1})
 - n operações por segundo
 - n 'alguma coisa' por segundo
- Em computação geralmente usado para ilustrar a frequência de operação dos processadores e barramento das placas

MHz ou GHz

