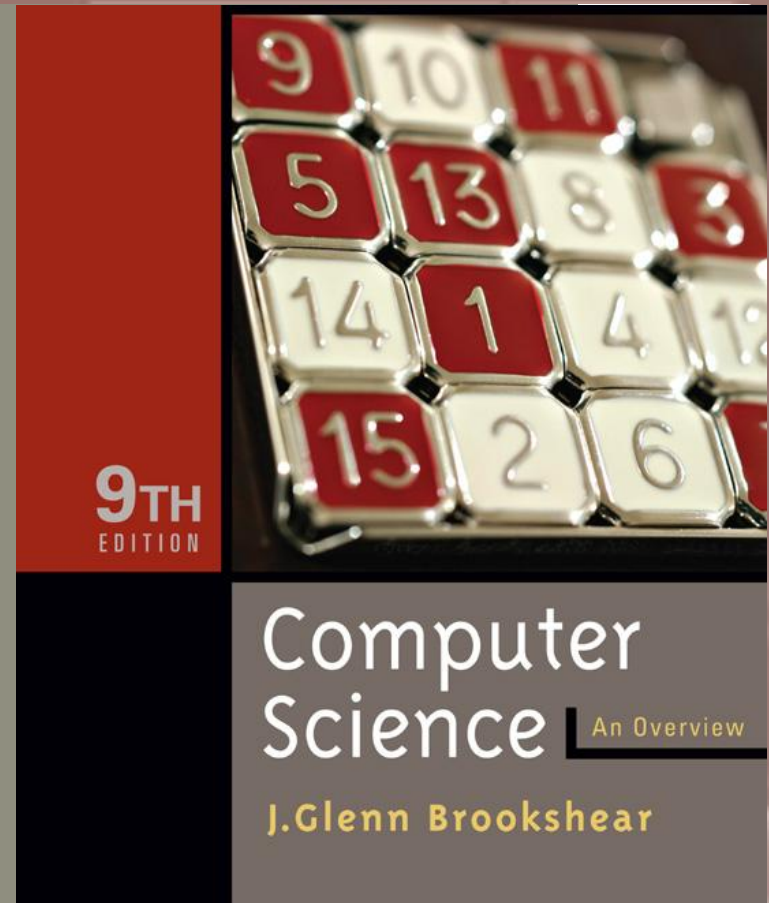




Capítulo 1

Armazenamento De Dados





Capítulo 1: Armazenamento de Dados

- 1.1 Bits e seu armazenamento
- 1.2 Memória principal
- 1.3 Armazenamento em massa
- 1.4 Representação de Informação como padrões de bits
- 1.5 O sistema binário
- 1.6 Guardando inteiros



Bits e o seu significado

- **Bit:** Dígitto Binário
(significado depende da aplicação em causa)
- Alguns significados possíveis para um bit isolado:
 - Valor numérico (1 ou 0)
 - Valor booleano (verdadeiro ou falso)
 - Voltagem (alta ou baixa)



Sequências de bits

- Todos os dados armazenados num computador são representados por sequências de bits:
 - Números
 - Caracteres de texto
 - Imagens
 - Sons
 - Tudo o resto...



Operações Booleanas

- **Operação booleana:** toda a operação que manipule um ou mais valores verdade/falso
 - Podem ser usadas para manipular bits
- Operações específicas
 - AND
 - OR
 - XOR
 - NOT



Operadores booleanos AND, OR, e XOR (ou exclusivo)

Operação AND

$$\begin{array}{r} 0 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{AND } 1 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 1 \\ \text{AND } 1 \\ \hline 1 \end{array}$$

Operação OR

$$\begin{array}{r} 0 \\ \text{OR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{OR } 1 \\ \hline 1 \end{array}$$

Operação XOR

$$\begin{array}{r} 0 \\ \text{XOR } 0 \\ \hline 0 \end{array}$$

$$\begin{array}{r} 0 \\ \text{XOR } 1 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 0 \\ \hline 1 \end{array}$$

$$\begin{array}{r} 1 \\ \text{XOR } 1 \\ \hline 0 \end{array}$$



Gates (portas lógicas)

- **Gates:** dispositivos que produzem o *output* de Operações Booleanas quando lhes são dados os valores de *input* das operações
 - Geralmente implementadas como circuitos eletrónicos
 - São as “peças” básicas na construção de computadores



Uma representação pictórica das gates AND, OR, XOR, e NOT, bem como os seus valores de *input* e *output*

AND



Inputs	Output
0 0	0
0 1	0
1 0	0
1 1	1

OR



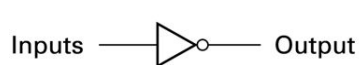
Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	1

XOR



Inputs	Output
0 0	0
0 1	1
1 0	1
1 1	0

NOT



Inputs	Output
0	1
1	0



Um exemplo de circuito lógico

Circuito

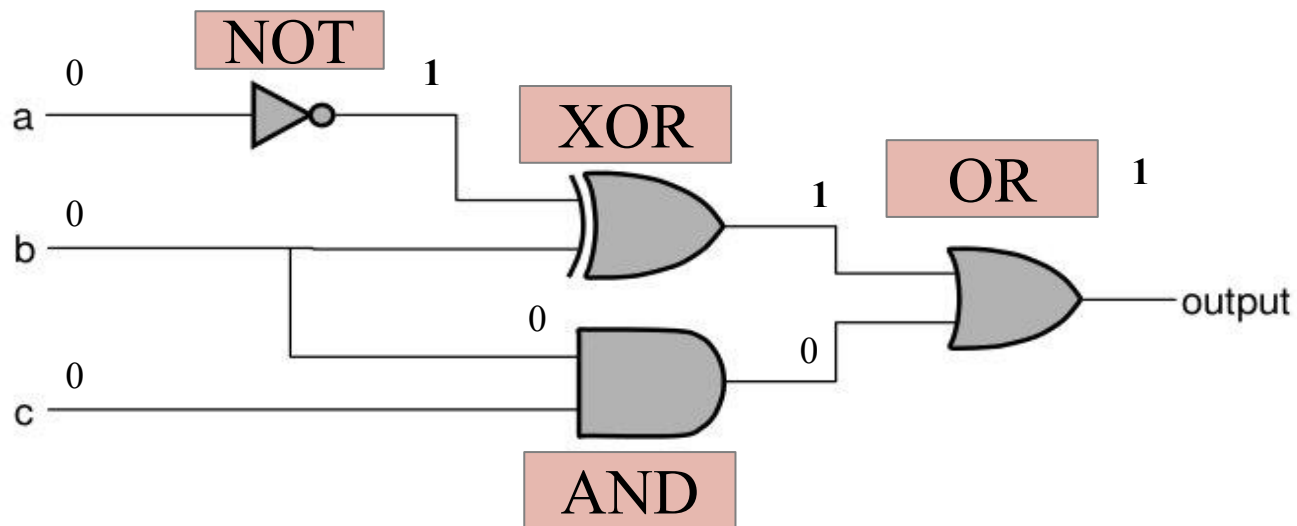


Tabela de Verdade



Input a, b, c	Output
000	1
001	1
010	0
011	1
100	0
101	0
110	1
111	1

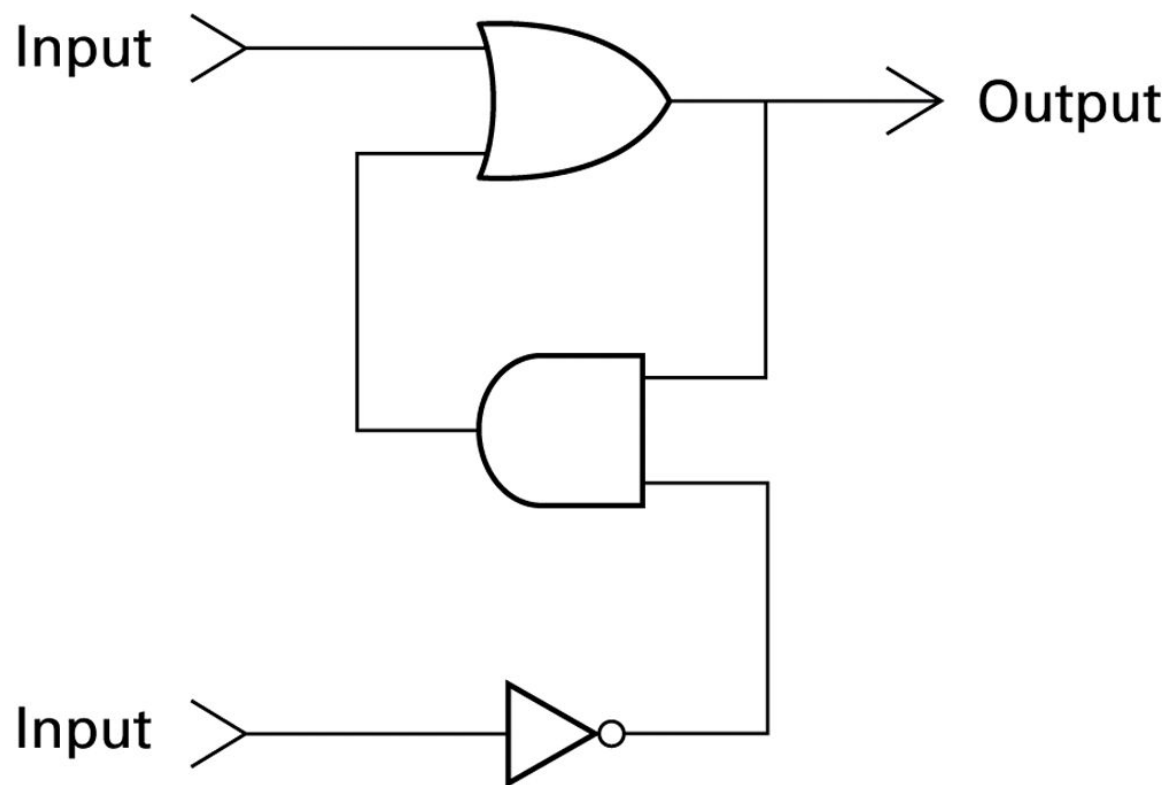


Flip-flops

- **Flip-flop** = um circuito construído a partir de gates que armazena um bit de informação
 - Tem uma linha de input que coloca o valor armazenado a 1
 - Tem uma linha de input que coloca o valor armazenado a 0
 - Se ambas as linhas de input são 0, o último valor armazenado é preservado



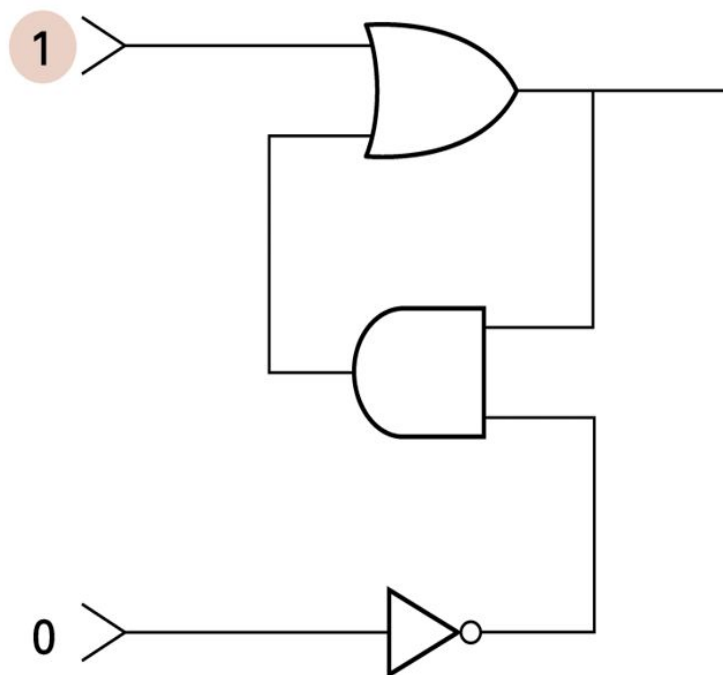
Um circuito flip-flop simples





Colocando o output de um flip-flop a 1

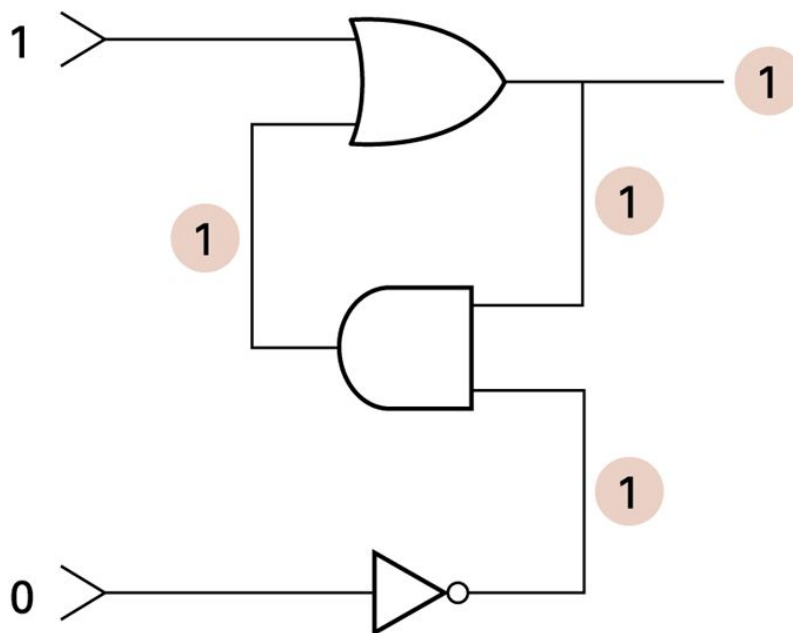
a) Um **1** é colocado no input de cima





Colocando o output de um flip-flop a 1 (cont.)

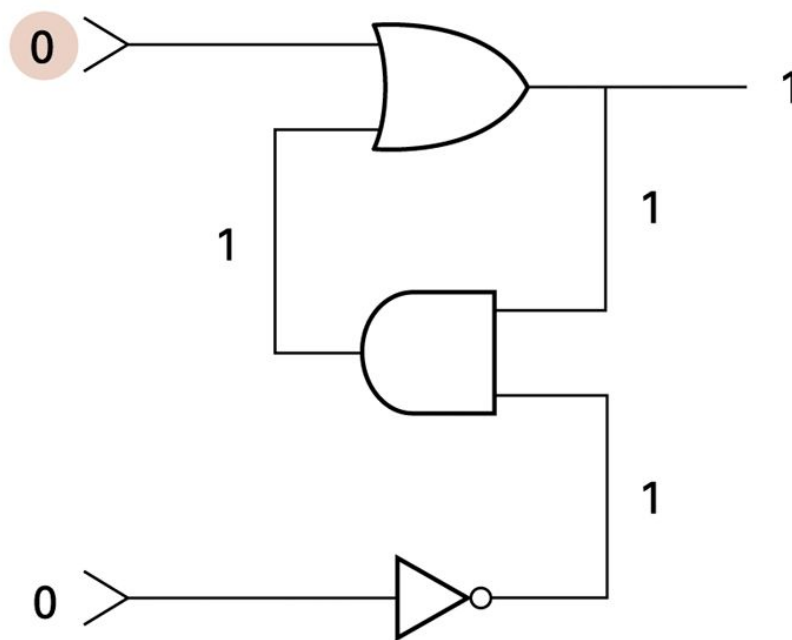
b) isto causa que o output da porta OR seja 1 e, por sua vez, o output da porta AND seja 1





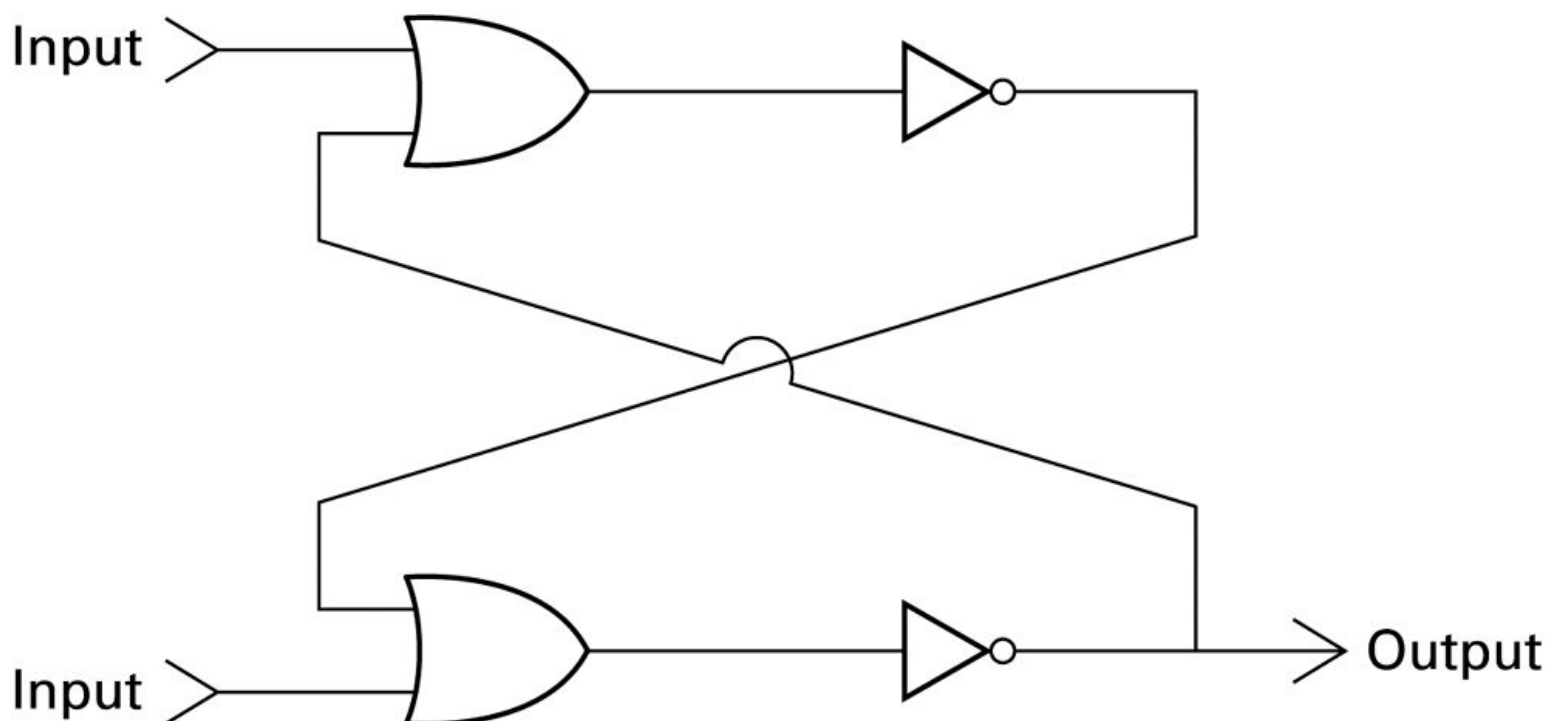
Colocando o output de um flip-flop a 1 (cont.)

c) O 1 da gate AND evita que a gate OR mude após o input superior passar a 0





Outra forma de construir um flip-flop





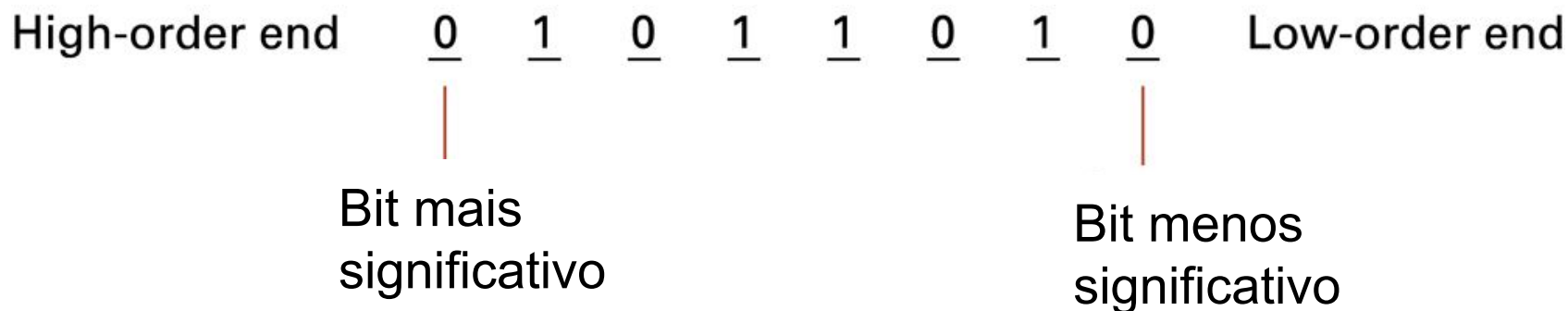
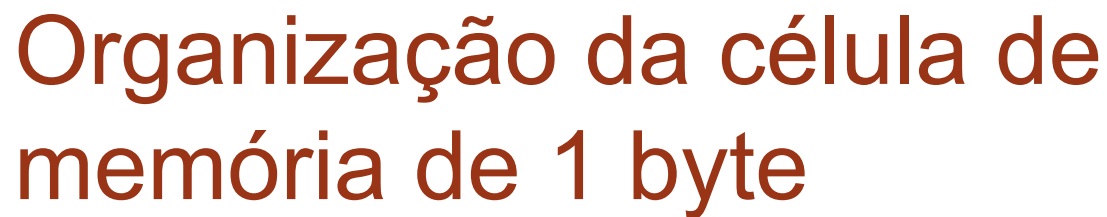
Volatilidade da memória

- **Memória volátil:** guarda o seu valor até a corrente ser desligada (ex: flip-flops)
- **Memória não volátil:** mantém o seu valor depois de a corrente ser desligada (ex: armazenamento magnético)



Memória principal: células

- **Célula (*cell*):** componente básico da memória de um computador (tipicamente tem 8 bits).
- **Byte:** sequência de 8 bits



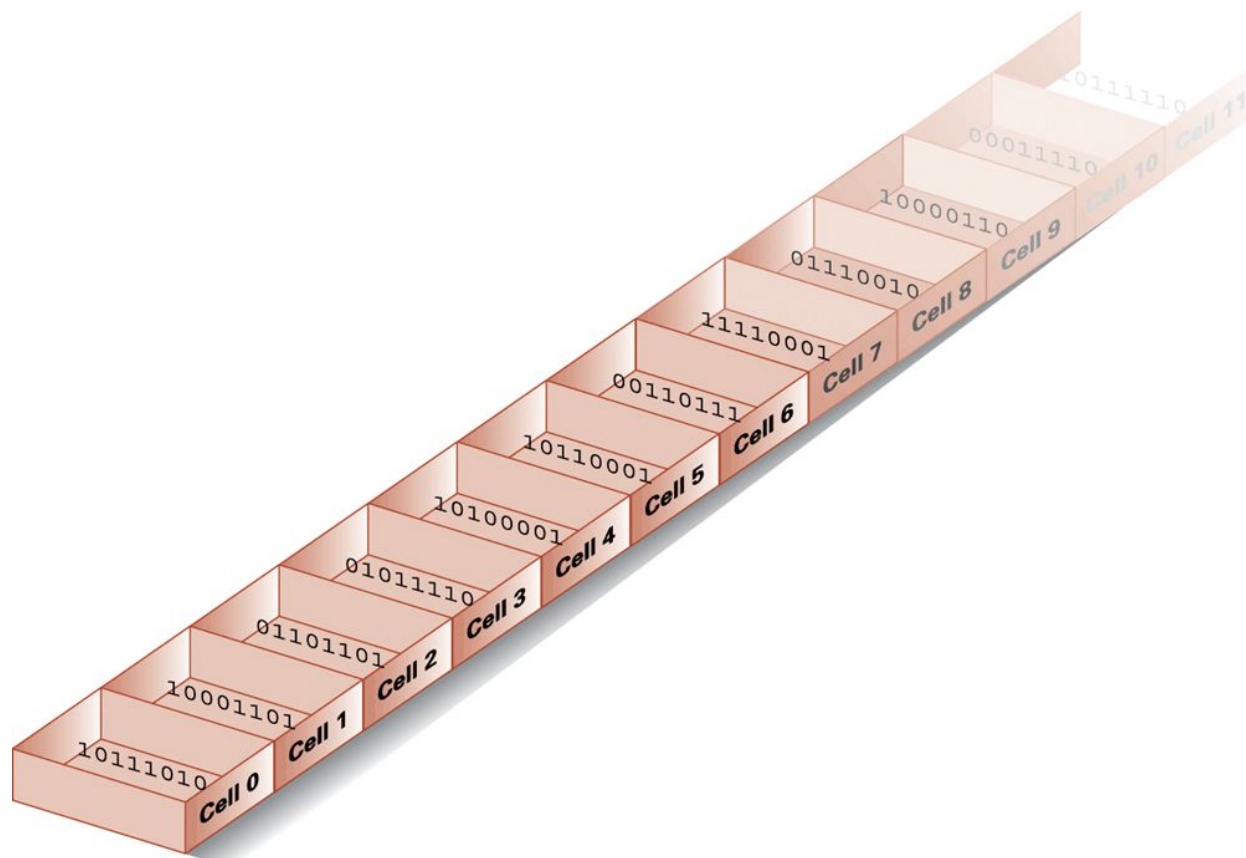


Endereços de memória

- **Endereço:** Um “nome” que identifica de forma unívoca uma célula da memória principal do computador
 - Nomes de células num computador são usualmente números consecutivos que começam do zero
 - Células têm ordem: “célula anterior” e “próxima célula” tem significado



Células de memória ordenadas por endereço





Terminologia de memória

- **Random Access Memory (RAM):** memória na qual as células individuais podem ser facilmente acedidas numa qualquer ordem





Medindo a capacidade de memória

- “Kilo-” normalmente significa 1,000;
Kilobyte (KB) = $2^{10} = 1024$
- “Mega-” normalmente significa 1,000,000;
Megabyte (MB) = $2^{20} = 1,048,576$
- “Giga-” normalmente significa 1,000,000,000;
Gigabyte (GB) = $2^{30} = 1,073,741,824$
- “tera”, “peta”, “exa”, “zetta”, “yotta”

Slides originais: © 2007 Pearson Addison-Wesley. All rights reserved

Modificados para MIM-Introdução à Informática (2425) por Pedro Ribeiro, Miguel Areias e Nuno Guimarães



Armazenamento em massa

- Não volátil (a informação mantém-se depois do computador ser desligado)
- Usualmente muito maior que a memória principal
- Usualmente muito mais lenta que a memória principal
 - Ex: discos rotativos (disco duro, CD)
 - Espera pelo posicionamento da cabeça de leitura
 - Latência de rotação

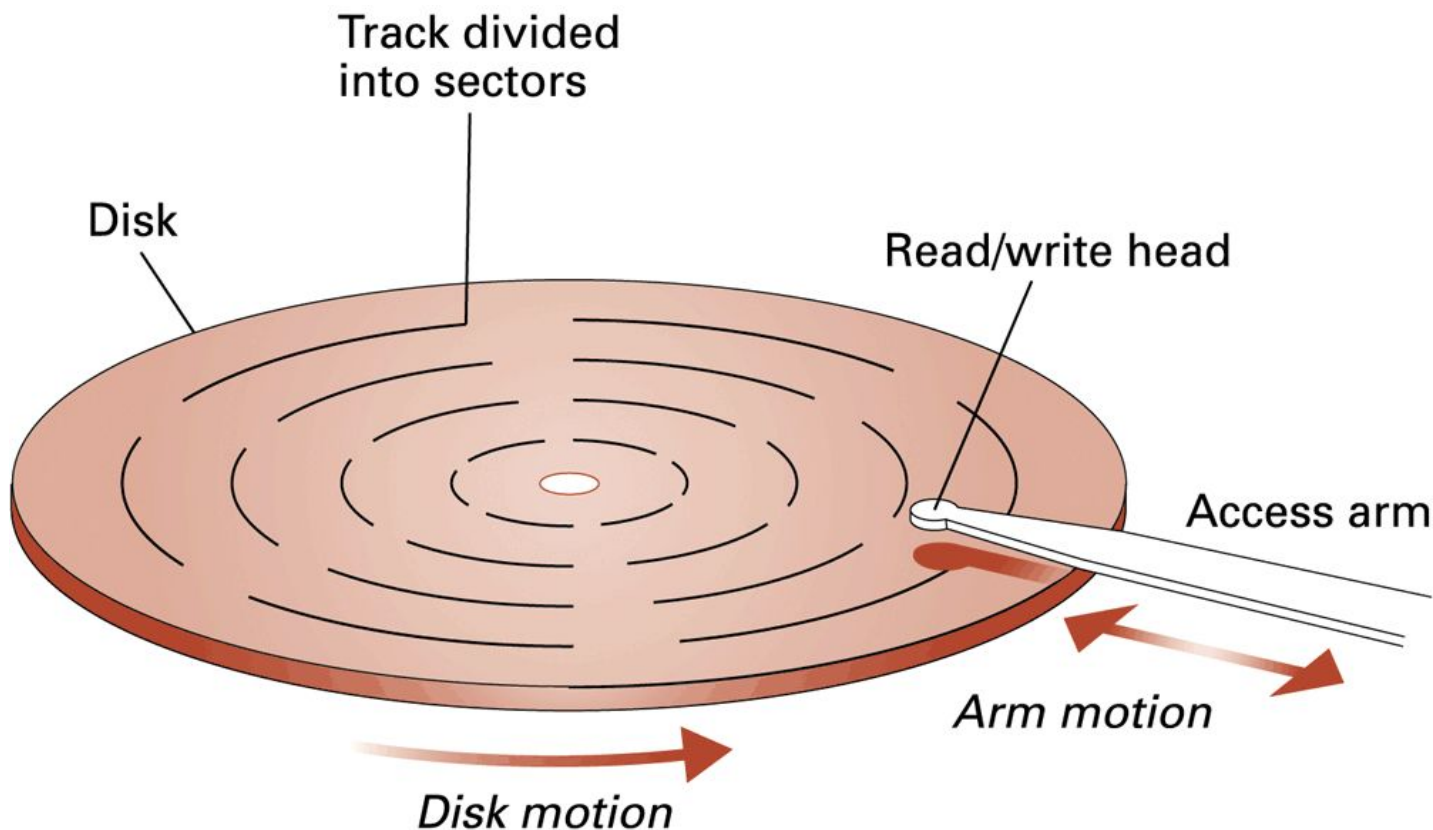


Sistemas de Armazenamento em Massa

- Sistemas magnéticos
 - Disco
 - Tape (fita)
- Sistemas óticos
 - CD
 - DVD
- Drives Flash

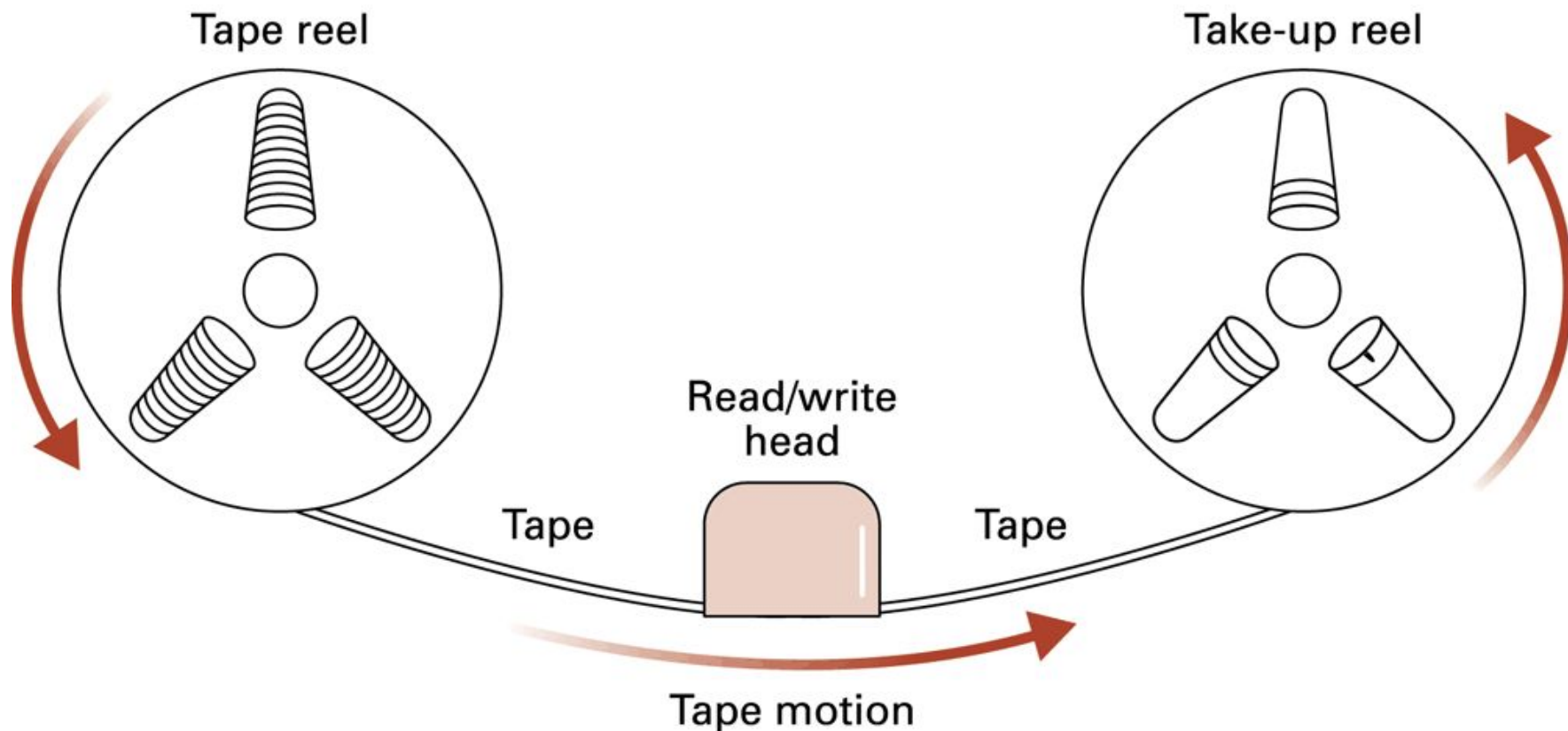


Armazenamento de um disco duro





Armazenamento em tape





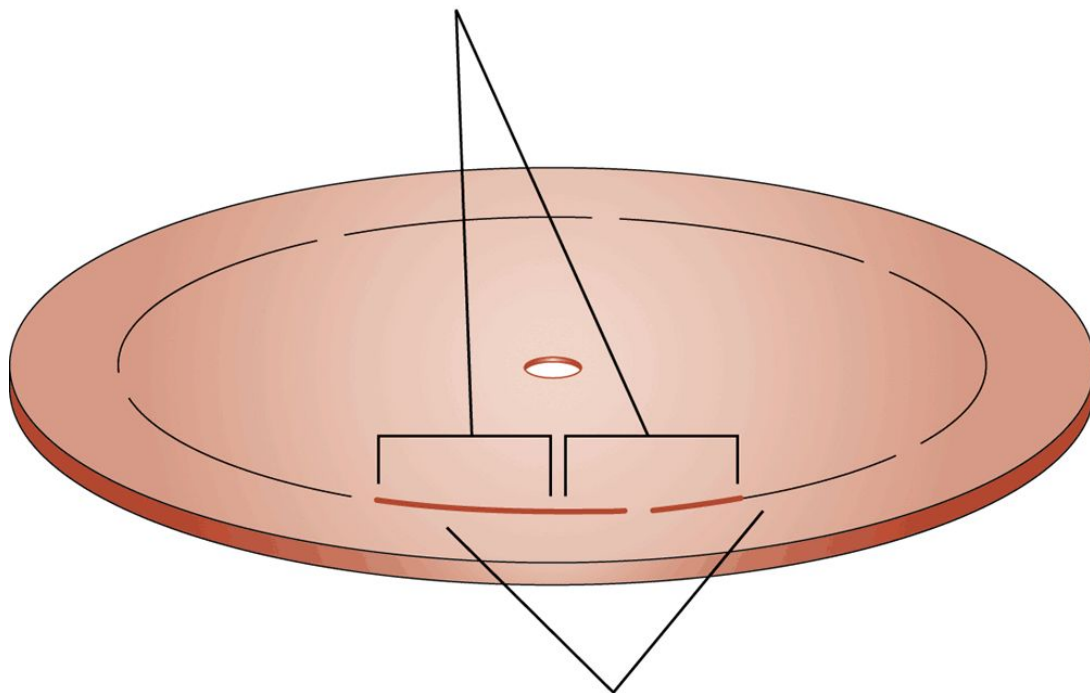
Ficheiros

- **Ficheiro:** a unidade de dados armazenados num sistema de armazenamento em massa.
 - **Registo lógico e Campo:** ordenação dos dados dentro de um ficheiro
 - **Registo físico:** um bloco de dados que respeita as características físicas do aparelho de armazenamento.



Registos lógicos vs registos físicos num disco

Registos lógicos correspondem a divisões naturais nos dados



Registos físicos correspondem ao tamanho de um setor



Representação de texto

- Cada carácter (letra, pontuação, etc.) tem associado uma sequência única de bits.
 - ASCII = 7-bit para a maioria de símbolos
 - Unicode = 16-bit
 - ISO standard proposto = 32-bit



Tabela de ASCII

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL



A mensagem “Hello.” em ASCII

01001000

H

01100101

e

01101100

l

01101100

l

01101111

o

00101110

.



Representação de valores numéricos

- Notação binária: usa bits para representar um número na base 2
- Existem algumas limitações na representação num computador de valores numéricos
 - **Overflow:** acontece quando um número é demasiado grande para ser representado
 - **Underflow:** número demasiado pequeno
 - **Truncagem:** representação sem precisão suficiente (valor entre dois valores representáveis)



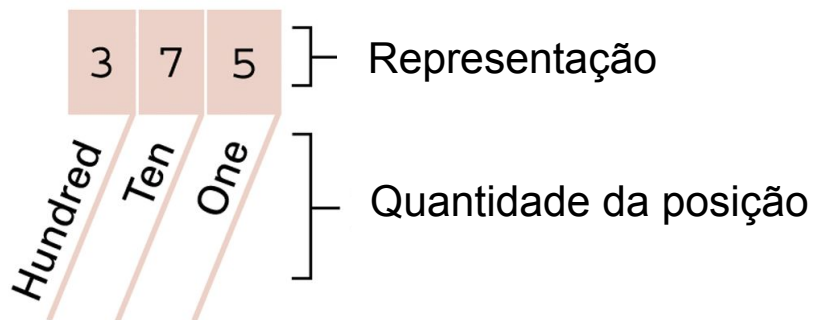
Base binária

- Sistema tradicional usa potências de 10
- O sistema binária usa... potências de 2

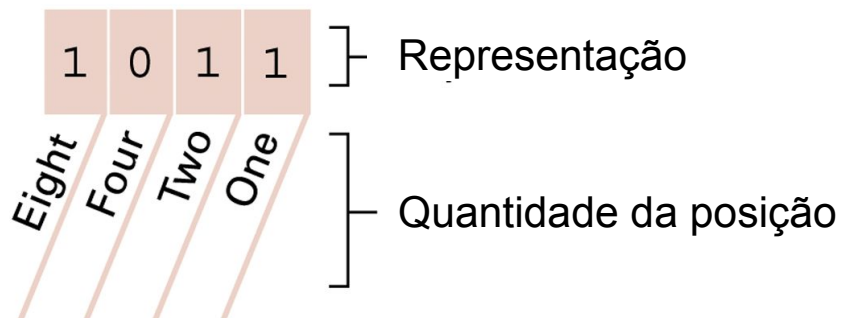


A base decimal e a base binária

Sistema de base decimal



Sistema de base binária



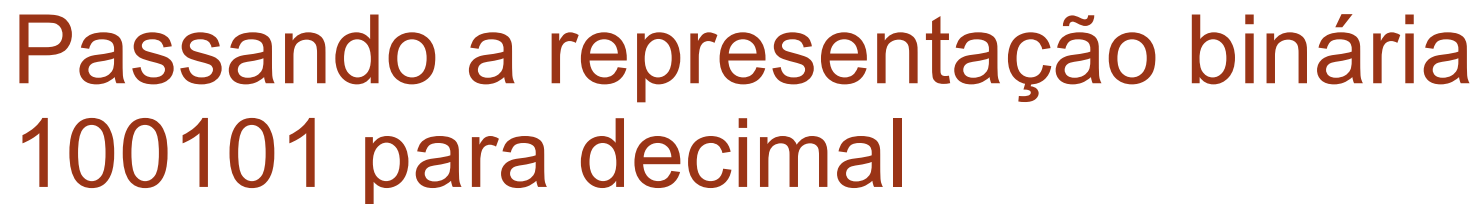


Diagram illustrating the conversion of the binary number 100101 to decimal. The binary digits are shown in a vertical column. Brackets connect each digit to its corresponding power of 2. The calculation is shown as a sum: 1 * 2⁵ + 0 * 2⁴ + 0 * 2³ + 1 * 2² + 0 * 2¹ + 1 * 2⁰ = 1 + 0 + 0 + 4 + 0 + 1 = 6. The final result is 6.



Um algoritmo para determinar a representação binária de um inteiro positivo

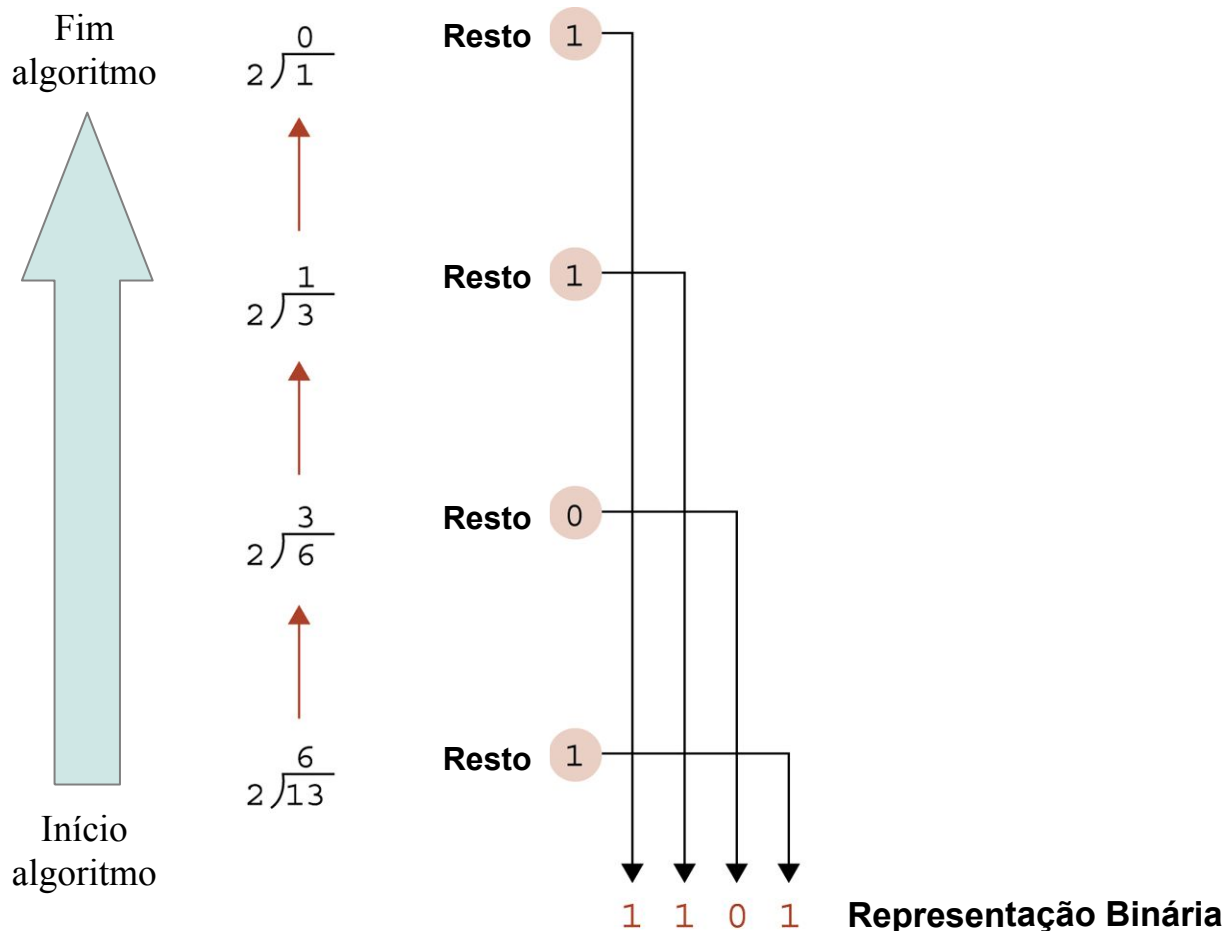
Passo 1: Dividir o valor por dois e guardar o resto da divisão

Passo 2: Enquanto o quociente obtido não for 0, continuar a dividir o novo quociente por dois e a guardar o resto da divisão

Passo 3: Após obter o quociente igual a 0, a representação binária do valor original consiste nos restos das operações listadas da direita para a esquerda pela ordem que foram registadas.



Aplicando o algoritmo para obter a representação binária de 13





Adição binária: factos

$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array}$$

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

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

$$\begin{array}{r} 1 \\ + 1 \\ \hline 10 \end{array}$$



Diagram illustrating the conversion of the binary number 101.101 to the decimal number 5 5/8. The binary number is shown with brackets indicating the value of each bit. The bits are 1, 0, 1, ., 1, 0, 1. The values are: 1 (1), 0 (0), 1 (1), . (.), 1 (1/2), 0 (0), 1 (1/8). The sum of the values is 1 + 0 + 1 + 0 + 1/2 + 0 + 1/8 = 2 + 5/8 = 2 5/8. The final result is 5 5/8.

Bit	Value	Weight	Product	Sum
1	1	1	1	1
0	0	1	0	1
1	1	1	1	2
.	.	1/2	1/2	2 1/2
1	1	1/2	1/2	2 1/2
0	0	1/4	0	2 1/2
1	1	1/8	1/8	2 5/8

Final result: 5 5/8



Representação de valores inteiros

- Como representar valores negativos?
- A representação mais usada é a de **complemento para dois**
 - $-x$ é representado por $2^n - x$



Notação de complemento para dois

usando representações de tamanho 3

Representação em Bits	Valores Representados
011	3
010	2
001	1
000	0
111	-1
110	-2
101	-3
100	-4

usando representações de tamanho 4

Representação em Bits	Valores Representados
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-1
1110	-2
1101	-3
1100	-4
1011	-5
1010	-6
1001	-7
1000	-8



Adições em complemento para dois

Problema em base 10		Problema em complemento para 2		Resposta em base 10
$\begin{array}{r} 3 \\ + 2 \\ \hline \end{array}$	→	$\begin{array}{r} 0011 \\ + 0010 \\ \hline 0101 \end{array}$	→	5
$\begin{array}{r} -3 \\ + -2 \\ \hline \end{array}$	→	$\begin{array}{r} 1101 \\ + 1110 \\ \hline 1011 \end{array}$	→	-5
$\begin{array}{r} 7 \\ + -5 \\ \hline \end{array}$	→	$\begin{array}{r} 0111 \\ + 1011 \\ \hline 0010 \end{array}$	→	2



Algoritmo para codificar o valor -6 em complemento para 2

Notação em complemento para 2
de 6 com 4 bits

0 1 1 0

Copiar os bits da
direita para a
esquerda até um “1”
seja copiado

Notação em complemento para
2 de -6 com 4 bits

1 0 1 0

Complementar os
restantes bits