Sistemas Numéricos

Sistemas Numéricos

1. Definição:

Sistema numérico é um **conjunto de caracteres** e regras matemáticas que são utilizados para **representar números**.

Sistemas Numéricos Antigos

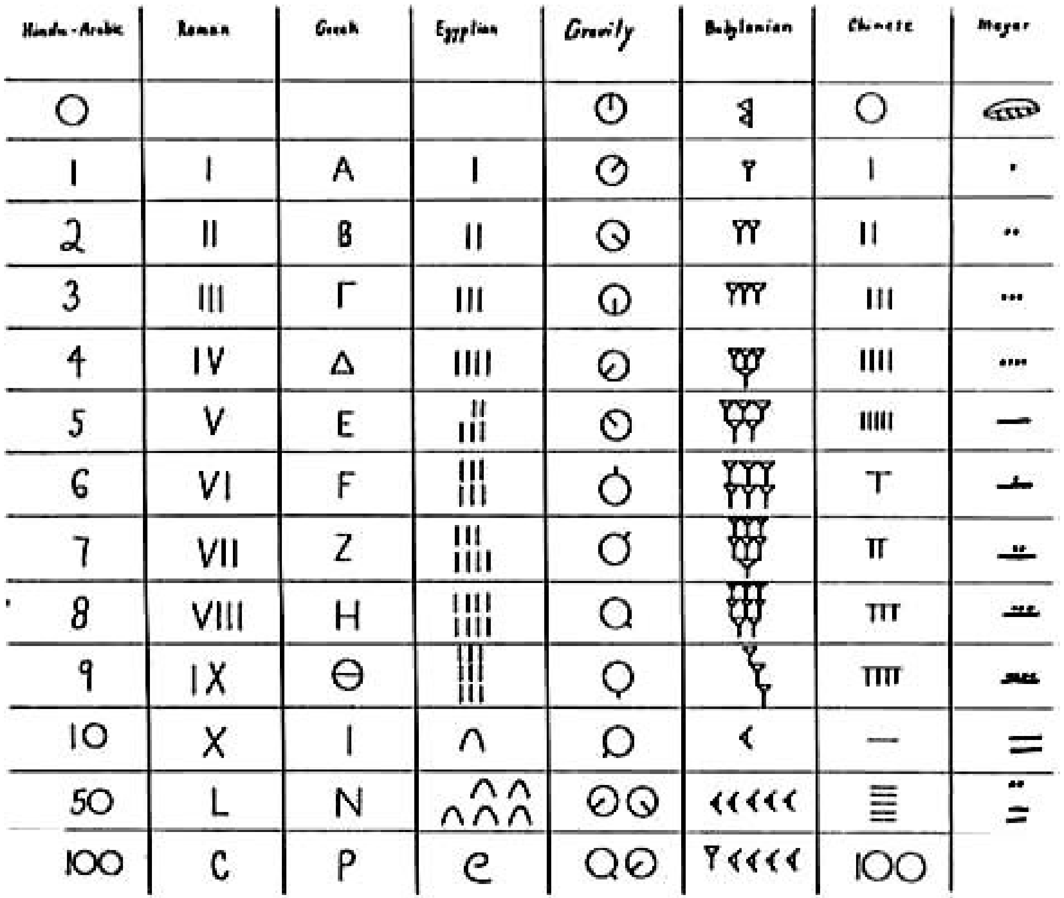
Sistema Romano;

Chinês;

Grego;

Arábico;

etc...



Sistemas Numéricos

1. Definição:

O sistema decimal (arábico) contém 10 algarismos, sendo:

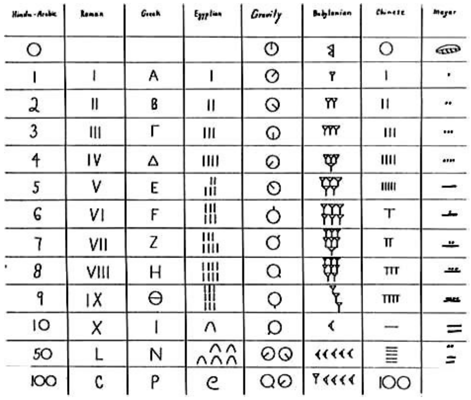
0 1 2 3 4 5 6 7 8 9

Depois do nove a contagem reinicia acrescendo-se uma dezena, e sucessivamente acrescendo o próximo elemento da sequência.

9... 10... 11... 12

19... 20... 21... 22...

99... 100... 101... 102....



Decomposição de números base decimal em potências

de 10 (b = 10).

$$\mathbf{N_{10} = a_n.b^{n-1} + a_{n-1}.b^{n-2} + \ldots + a_1.b^0 + a_m.b^{-1} + a_{m-1}.b^{-2} \;\ldots}$$

n = dígitos da parte inteira

m = dígitos da parte fracionária

b = base a_i = algarismo

Ex.: 325.453 = 300 + 20 + 5 + 0.4 + 0.05 + 0.003

$(213)_{10} = 2 . 10^2 + 1 . 10^1 + 3 . 10^0 = 200 + 10 + 3 = 213$

$(43.84)_{10} = 4 . 10^1 + 3 . 10^0 + 8 . 10^{-1} + 4 . 10^{-2} = 40+3+0.8+0.04 = 43.84$

Sistemas Numéricos

2. Sistemas Numéricos Computacionais

No computador, todas as informações são representadas e processadas na forma binária.

Sistema Binário: possui apenas 2 algarismos – **0 e 1.**

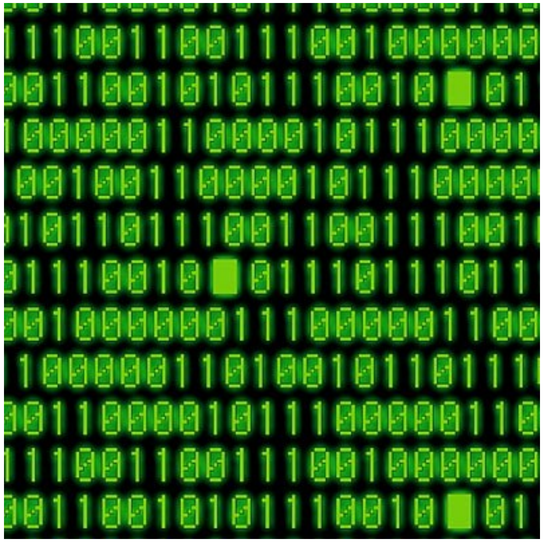
Razão: simplicidade de representação dos mesmos por:

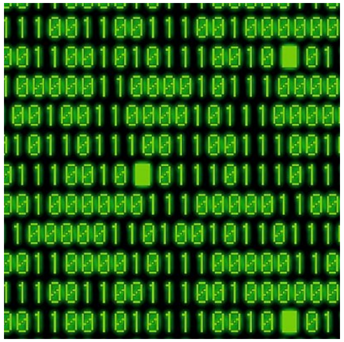
– dispositivos elétricos

– eletrônicos

– mecatrônicos

– magnéticos





Sistemas Numéricos

Sistema Binário: possui apenas 2 algarismos – **0 e 1.**

Na prática cada dígito recebe a denominação de **bit** (**binary digit**)

ex.: $(101001)_2$ 6 bits – base 2

O conjunto de **8 bits** é chamado de **byte** – termo bastante utilizado na informática.

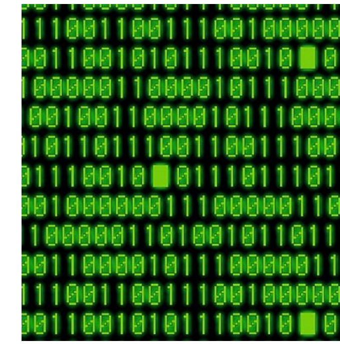
Logo, se **n = número de bits**, $\mathbf{2^n}$ é quantidade de números representados.

1. quantos e quais números podem ser representados em 4 bits ?
2. e em 1 byte ?
3. e em 4 bytes ?

Sistemas Numéricos

Sistemas Numéricos

Unidades da base binária



1 nibble	-	4 bits	
1 byte	-	8 bits	
1 KB	-	1024 bytes	(2^{10})
1 MB	-	1024 KB	(2^{20})
1 GB	-	1024 MB	(2^{30})
1 TB	-	1024 GB	(2^{40})
1 PB	-	1024 TB	(2^{50})

.....

Sistemas Numéricos

Sistema Binário: possui apenas 2 algarismos – **0 e 1.**

Contagem decimal	Contagem binária
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
...	...

Como exercício – dê sequência da contagem até 32.

Sistemas Numéricos

Sistema Octal (base)$_8$: no sistema de numeração hexadecimal existem 7 algarismos:

0 1 2 3 4 5 6 7

O objetivo é facilitar a representação de cadeias binárias muito grandes:

DEC	BIN	OCTAL	DEC	BIN	OCTAL
0	0	0	8	1000	10
1	1	1	9	1001	11
2	10	2	10	1010	12
3	11	3	11	1011	13
4	100	4	12	1100	14
5	101	5	13	1101	15
6	110	6	14	1110	16
7	111	7	15	1111	17

Sistemas Numéricos

Sistemas Numéricos

Sistema Hexadecimal (base)$_{16}$: no sistema de numeração hexadecimal existem 16 algarismos:

0 1 2 3 4 5 6 7 8 9

A B C D E F

O objetivo é facilitar a representação de

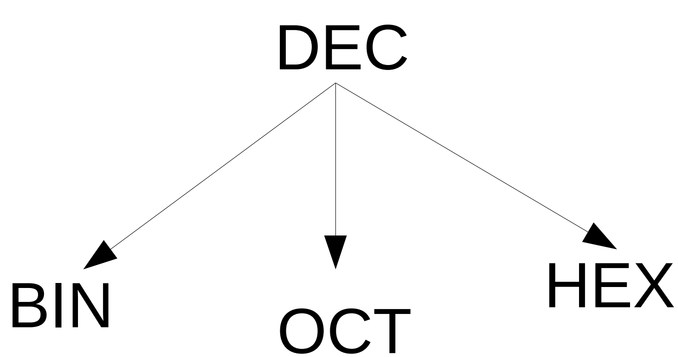
cadeias binárias muito grandes

A = 10 ; B = 11; C = 12; D = 13; E = 14; F = 15;

49	4E	44	58	28	00
00	00	00	00	00	00
E8	0F	00	00	00	00
00	00	00	00	C2	01
7C	0D	00	00	00	00
7B	0D	00	00	00	00
90	64	AF	EB	87	0A
90	64	AF	EB	87	0A

Tabela dos Decimais

Considerando a ordem de contagem de cada base é possível montar uma tabela em que se possa observar qual relação existe entre 2 números de bases diferentes.



DEC.	BIN.	OCT.	HEXAD.
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
...	...	...	...
32	100000	40	20
64	1000000	100	40
128	10000000	200	80
256	100000000	400	100
1234	10011010010	2322	4D2

3. Conversão de decimal para qualquer base

Utilizamos o **método das divisões sucessivas** pelo valor da respectiva base:

Ex.: converter o número $(213)_{10}$ para **binário $(???)_2$**.

213 / 2 = 106 e sobra 1

106 / 2 = 53 e sobra 0

53 / 2 = 26 e sobra 1

213 = 1101 0101

Exercícios de conversão de base:

a) $(75)_{10} \rightarrow (?)_2$

b) $(324)_{10} \rightarrow (?)_2$

c) $(129)_{10} \rightarrow (?)_2$

d) $(254)_{10} \rightarrow (?)_2$

e) $(170)_{10} \rightarrow (?)_2$

f) $(32.768)_{10} \rightarrow (?)_2$

3. Conversão de <u>decimal</u> para qualquer base

Utilizamos o **método das divisões sucessivas** pelo valor da respectiva base:

Ex.: converter o número $(213)_{10}$ para **octal $(???)_8$**.

213 / 8 = 26 e sobra 5

26 / 8 = 3 e sobra 2

213 = 325

Exercícios de conversão de base:

a) $(75)_{10}$ → $(?)_8$

b) $(324)_{10}$ → $(?)_8$

c) $(129)_{10}$ → $(?)_8$

d) $(254)_{10}$ → $(?)_8$

e) $(170)_{10}$ → $(?)_8$

f) $(32.768)_{10}$ → $(?)_8$

3. Conversão de decimal para qualquer base

Utilizamos o **método das divisões sucessivas** pelo valor da respectiva base:

Ex.: converter o número $(213)_{10}$ para **hexadecimal $(???)_{16}$**.

213 / 16 = 13 e sobra 5

213 = D5

Exercícios de conversão de base:

a) $(75)_{10} \rightarrow (?)_{16}$

b) $(324)_{10} \rightarrow (?)_{16}$

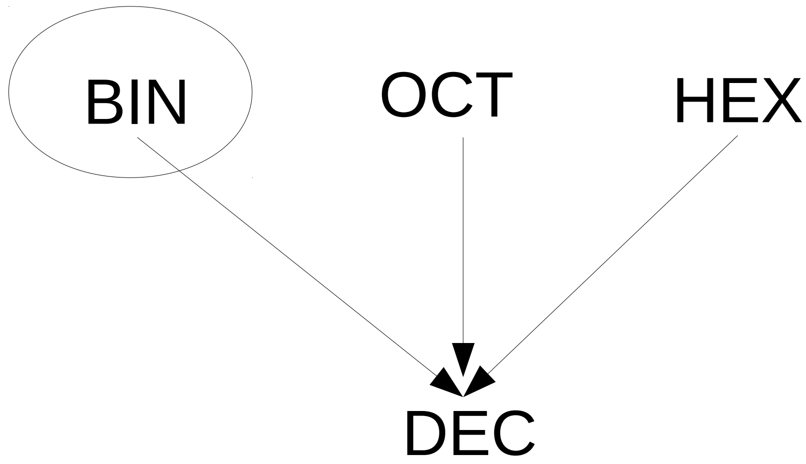
c) $(129)_{10} \rightarrow (?)_{16}$

d) $(254)_{10} \rightarrow (?)_{16}$

e) $(170)_{10} \rightarrow (?)_{16}$

f) $(32.768)_{10} \rightarrow (?)_{16}$

4. Conversão de qualquer base para Decimal



$$\mathbf{N_{10} = a_n.b^{n-1} + a_{n-1}.b^{n-2} + \ldots + a_1.b^0 + a_m.b^{-1} + a_{m-1}.b^{-2} \ \ldots}$$

n = dígitos da parte inteira

m = dígitos da parte fracionária

b = base $\qquad a_i$ = algarismo

$(1000\ 1101)_2 \rightarrow (?)_{10}$

$(0101\ 0101)_2 \rightarrow (?)_{10}$

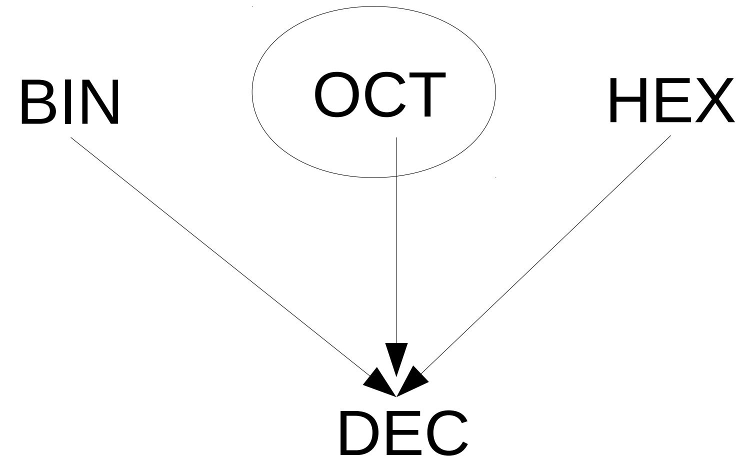
$(1001\ 1111)_2 \rightarrow (?)_{10}$

$(1101\ 0000\ 0101)_2 \rightarrow (?)_{10}$

$(1010\ 1011\ 1100)_2 \rightarrow (?)_{10}$

$(1111\ 1100\ 1101)_2 \rightarrow (?)_{10}$

4. Conversão de qualquer base para Decimal



$$N_{10} = a_n.b^{n-1} + a_{n-1}.b^{n-2} + \dots + a_1.b^0 + a_m.b^{-1} + a_{m-1}.b^{-2} \;\dots$$

n = dígitos da parte inteira

m = dígitos da parte fracionária

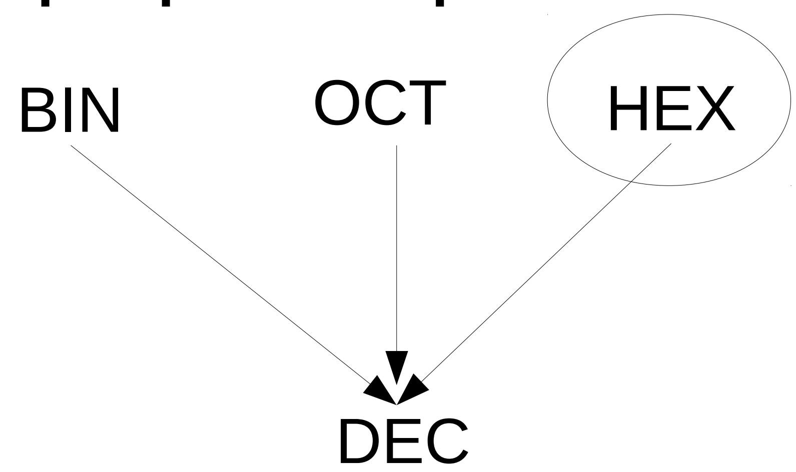
b = base a_i = algarismo

$(215)_8 \rightarrow (?)_{10}$ $(6405)_8 \rightarrow (?)_{10}$

$(125)_8 \rightarrow (?)_{10}$ $(5274)_8 \rightarrow (?)_{10}$

$(707)_8 \rightarrow (?)_{10}$ $(1425)_8 \rightarrow (?)_{10}$

4. Conversão de qualquer base para Decimal



$$\mathbf{N_{10} = a_n.b^{n-1} + a_{n-1}.b^{n-2} + \dots + a_1.b^0 + a_m.b^{-1} + a_{m-1}.b^{-2} \;\dots}$$

n = dígitos da parte inteira

m = dígitos da parte fracionária

b = base $\quad a_i$ = algarismo

$(8D)_{16} \rightarrow (?)_{10}$

$(55)_{16} \rightarrow (?)_{10}$

$(A6)_{16} \rightarrow (?)_{10}$

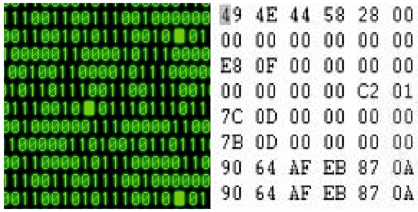
$(D05)_{16} \rightarrow (?)_{10}$

$(ABC)_{16} \rightarrow (?)_{10}$

$(99BA)_{16} \rightarrow (?)_{10}$

5. Conversão de binário ←→ octal

e binário ←→ hexadecimal



Binário-Octal, os bits são agrupados de **3 a 3**, a partir do bit da direita.

A conversão é realizada associando o algarismo numérico octal correspondente.

BIN.	OCT.
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7

$$(0001\ 0010\ 1110)_2$$

$$(100\ 101\ 110)_2$$

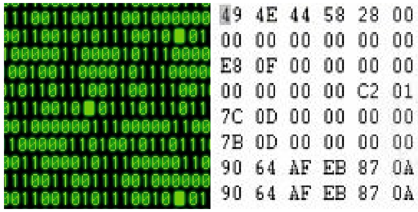
$$(\ 4\quad 5\quad 6\ )_8$$

$$(\ 6\quad 0\quad 7\quad 1\ )_8$$

$$(\ 110\ 000\ 111\ 001\ )_2$$

$$(\ 1100\ 0011\ 1001\ )_2$$

5. Conversão de binário ←→ octal
e binário ←→ hexadecimal



Binário-Hexadecimal, os bits são agrupados de **4 a 4**, a partir do bit da direita.

A conversão é realizada associando o algarismo numérico hexadecimal

BIN.	HEXAD.
0	0
1	1
10	2
11	3
100	4
101	5
110	6
111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

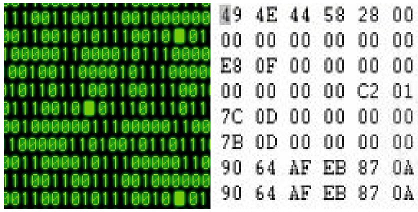
$$(0010\ 1101\ 1011\ 1101)_2$$

$$(\ 2\ \ D\ \ B\ \ D\ )_{16}$$

$$(\ 9\ \ 5\ \ C\ )_{16}$$

$$(\ 1001\ 0101\ 1100\ )_2$$

6. Conversão de octal ← → hexadecimal



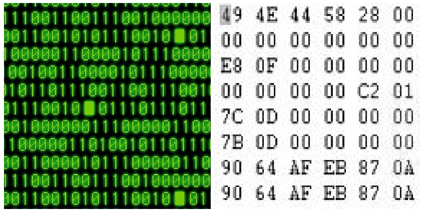
Converter para binário e logo após para o sistema numérico desejado.

OCT.	BIN.	HEXAD.
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
10	1000	8
11	1001	9
12	1010	A
13	1011	B
14	1100	C
15	1101	D
16	1110	E
17	1111	F

octal ← → binário ← → hexadecimal

$(\mathrm{AF35})_{16} \rightarrow (?)_8$

$(3173)_8 \rightarrow (?)_{16}$



7. Operações aritméticas no sistema binário

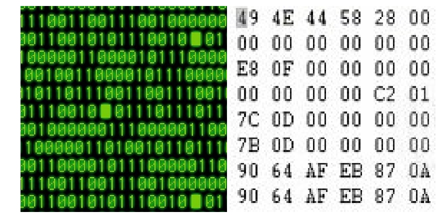
Adição Sistema Binário

0+0=0 0+1=1 1+0=1 1+1=10

	$(11)_2$	$(3)_{10}$		$(110)_2$	$(6)_{10}$
+	$(10)_2$	$(2)_{10}$	+	$(111)_2$	$(7)_{10}$
=	$(101)_2$	$(5)_{10}$	=	$(1101)_2$	$(13)_{10}$

Resolvam e tirem a prova em base decimal

1. $(11001)_2 + (1011)_2 =$?
2. $(100111)_2 + (1110)_2 + (1011)_2 =$?
3. $(11011)_2 + (1111)_2 + (1110)_2 =$?



7. Operações aritméticas no sistema binário

Subtração Sistema Binário

0–0 = 0   0–1=1   1–0=1   1–1=0

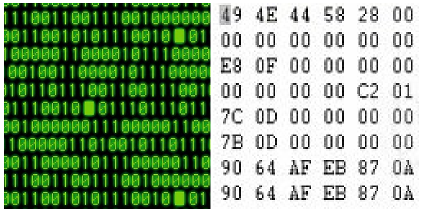
Obs.: **0–1 = 1** e passa 1 para próximo bit

$$
\begin{array}{lll}
 & (111)_2 & (7)_{10} \\
- & (100)_2 & (4)_{10} \\
= & (011)_2 & (3)_{10}
\end{array}
$$

$$
\begin{array}{lll}
 & (10110)_2 & (22)_{10} \\
- & (1101)_2 & (13)_{10} \\
= & (01001)_2 & (9)_{10}
\end{array}
$$

Resolvam e tirem a prova em base decimal

1. $(1111\ 1111)_2 - (1010\ 0100)_2 = ?$
2. $(11001)_2 - (1110)_2 = ?$
3. $(110001)_2 - (11010)_2 = ?$
4. $(1111\ 0001)_2 - (1110\ 0101)_2 = ?$



7. Operações aritméticas no sistema binário

Multiplicação Sistema Binário

0*0 = 0   0*1 = 0   1*0 = 0   1*1 = 1

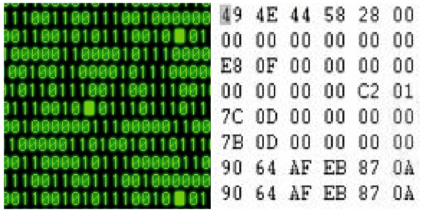
	$(110)_2$	$(6)_{10}$
x	$(11)_2$	$(3)_{10}$
	110+110 = 10010	$(18)_{10}$

Resolvam e tirem a prova em base decimal

1. $(11011)_2 \times (101)_2 = ?$
2. $(101110)_2 \times (1101)_2 = ?$
3. $(0110\ 0100)_2 \times (1100\ 1000)_2 = ?$

Sistemas Numéricos

8. Números Positivos e Negativos



Representação decimal de números negativos **+, –**

Computacionalmente, estes símbolos não podem ser utilizados.

Forma 1 – definir um bit de sinal.

– positivo bit de sinal 1

– negativo bit de sinal 0

Forma 2 – Complemento 2

– mas primeiro precisa-se converter um número para complemento 1.

Exemplo:

1100 1101 → 0011 0010 + 1 = 0011 0011