Sistemas Digitais

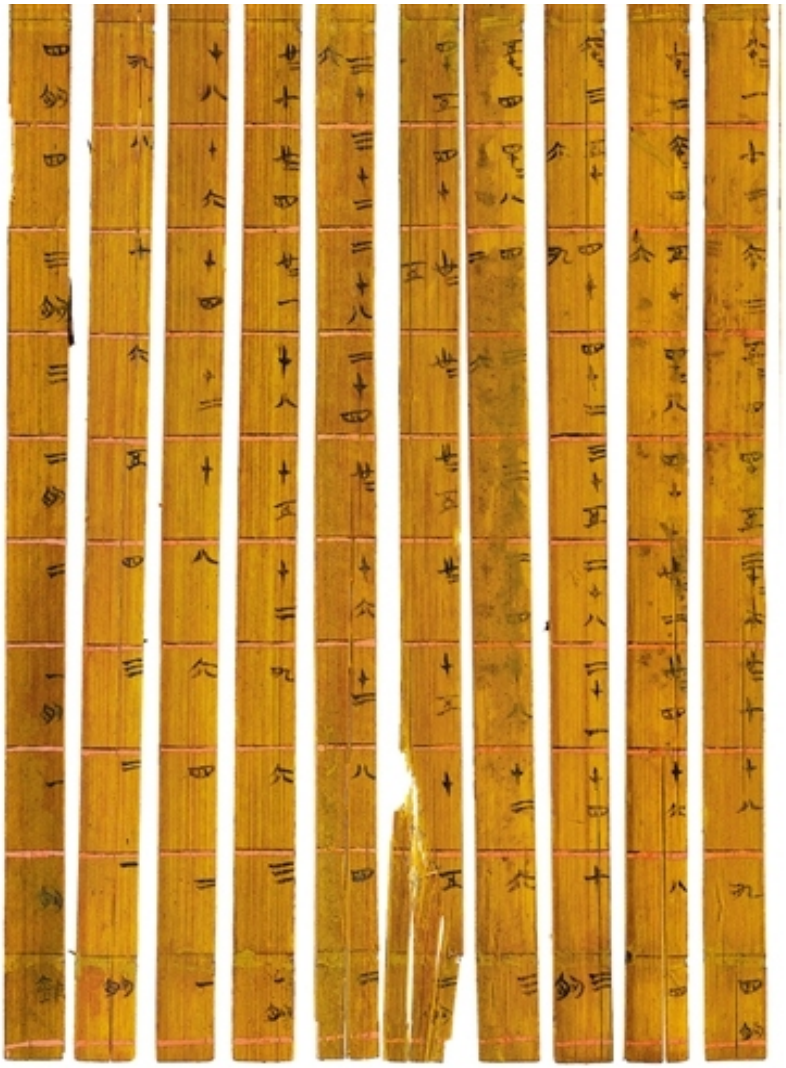
Sistemas de Numeração

Aula 02

Prof. Leandro Nogueira Couto
UFU – Monte Carmelo
05/2013

O Sistema Decimal

- Sistema **decimal**, também chamado de **denário** ou **base dez**
- Decimal pois usamos **10 símbolos** que chamamos de **algarismos**
- Usado no mundo todo, com numerais hindu-arábicos, romanos, chineses, etc.

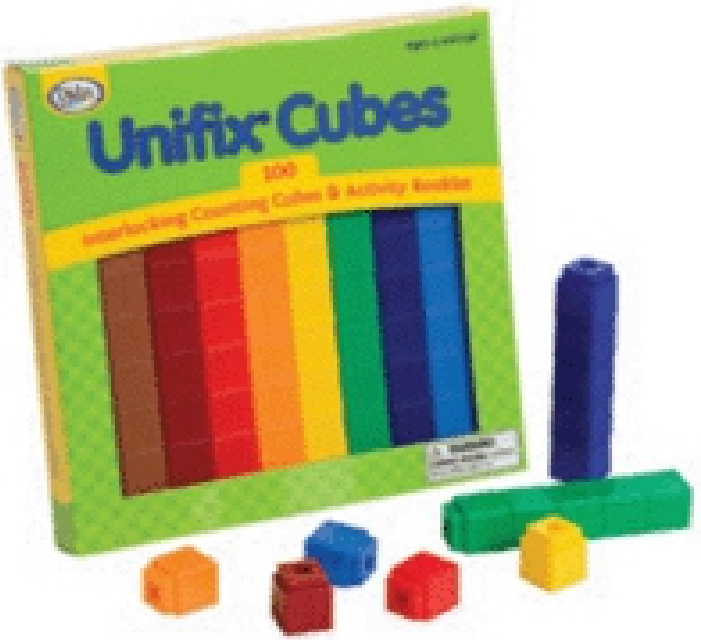


O Sistema Decimal

- Porque contamos como contamos no sistema decimal?
  - Entendimento essencial para compreendermos outros sistemas
  - Todo sistema numérico é fundamentalmente igual

O Sistema Decimal

- Porque depois do 9 vem o 10?
- Como fazemos pra interpretar o valor do número 384 sabendo que estamos em um sistema base 10?
- Quantos possíveis números conseguimos representar com 3 dígitos na base 10? E porque?
  - Elaboração na lousa

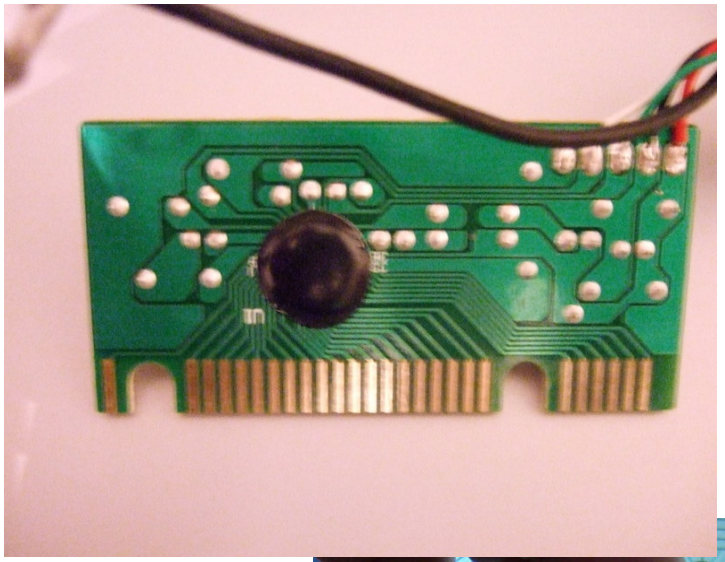


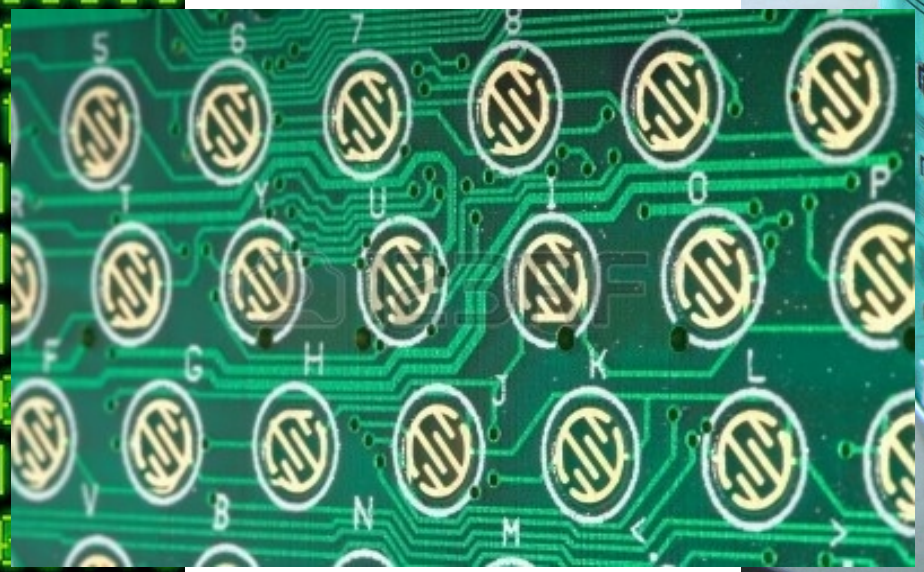
O Sistema Binário

- Porque?
  - Computação é feita de 1s e 0s!
  - Com “energia”/sem “energia”
- É possível contar até 100 apenas com 1s e 0s?
- O que muda?
- O que não muda?

O Sistema Binário

- Se eu tenho um teclado de 50 teclas, de quantos algarismos preciso pra representar todas as possíveis combinações?





O Sistema Binário

- Nomenclatura
  - 1 algarismo: **bit**
  - 4 algarismos: **nibble**
  - 8 algarismos: **byte**
- Tabela com alguns valores:

Binary	Decimal
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	10
1011	11
1100	12
1101	13
1110	14
1111	15

Conversão Bin-Dec

- Conversão binário-decimal
  - Lembramos que lemos 594 como:
    - $4x10^0 + 9x10^1 + 5x10^2$
  - Vamos tentar fazer o mesmo com o número 101 (equivale ao 5). Note que mudamos a base (da base 10 pra base binária 2):
    - $1x2^0 + 0x2^1 + 1x2^2$
  - Note que podemos usar a notação 594_{10} para dizer que o número 594 está na base 10 e a notação 101_2 para dizer que 101 está na base 2

Conversão Bin-Dec

- Exemplos:
  - Converta para decimal:

    01110

    1010

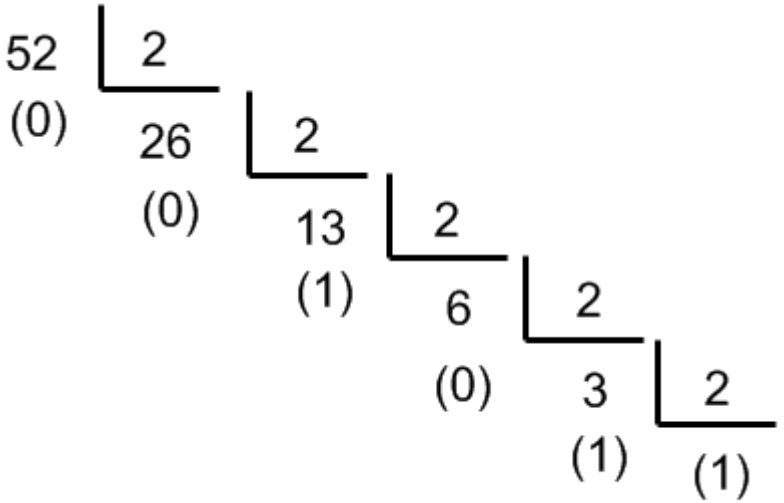
    1100110001

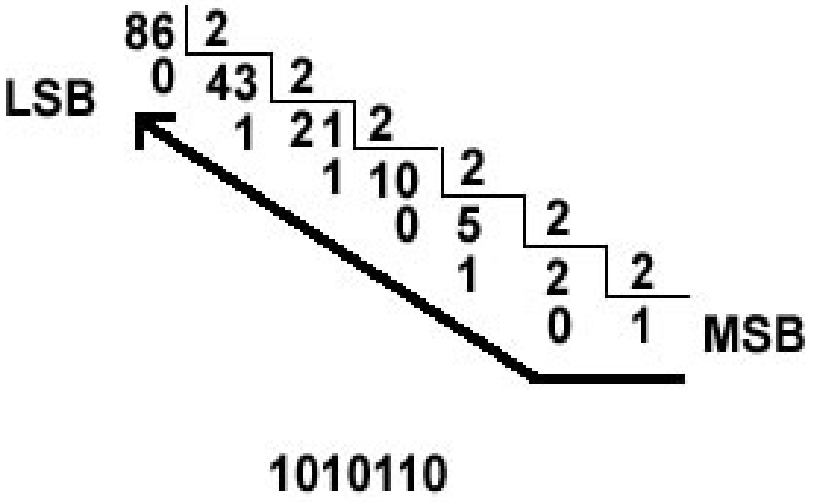
Conversão Dec-Bin

- Conversão decimal-binário
  - Método das divisões sucessivas (na lousa)
  - Exemplos:

    52 em binário é 110100

    86 em binário é 1010110





Conversão Bin-Dec

- Exemplos:
  - Converta para binário:

    15

    47

    512

Sistema Binário

- E se temos um número fracionário, é possível converter?
  - Por exemplo: 10.5

    $5x10^{-1} + 0x10^{0} + 1x10^{1}$
- Tratamos a vírgula/ponto normalmente. Por exemplo, como fica 101.101 em decimal?

Sistema Octal

- Além do sistema binário, existem outros sistemas úteis para a Computação?
- O sistema Octal, por exemplo, é um sistema base 8, composto dos algarismos 0,1,2,3,4,5,6,7
  - Na prática já não é tão usado em Sistemas Digitais. Mas serve como boa introdução para o próximo sistema...
- Como ficam as conversões no sistema Octal?
  - Converta **144 Oct-Dec**. Converta **92 Dec-Oct**.
- Porque o programadores sempre confundem o Natal com o Halloween?

  Porque Dec 25 = Oct 31

Sistema Octal

- As conversões no sistema Octal são análogas!
- E para converter Octal para Binário e vice-versa?
  - Como 8 é uma potência de 2, a conversão é muito simples
  - Basta converter cada algarismo diretamente para seu equivalente em binário, respeitando o padrao de bits do sistema (no Octal é 3, pois $8=2^3$)
  - Assim, temos que 27_8 e binário = 010 (2) e 111 (7)

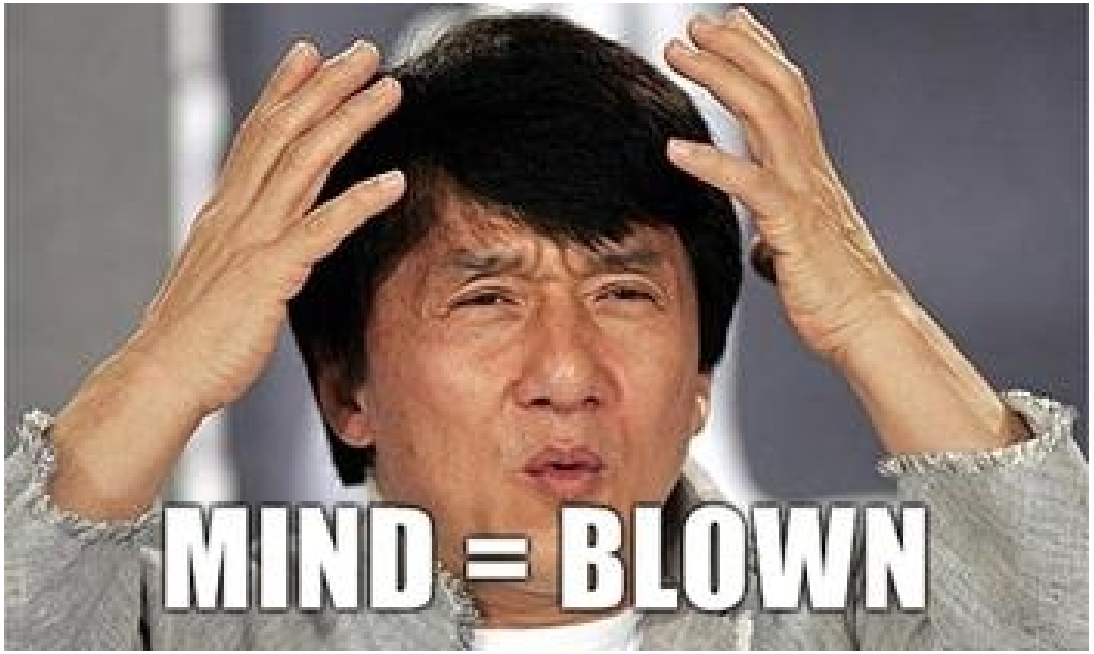
    Portanto $27_8 = 10111_2$
  - Para conversão Bin-Oct, fazemos o processo inverso!
  - Exs: Converta 110010 e 11010101 para Octal

Sistema Hexadecimal

- O Sistema Hexadecimal possui 16 algarismos
  - Até agora vimos no máximo 10...
- Quais são esses algarismos?

Sistema Hexadecimal

- O Sistema Hexadecimal possui 16 algarismos
  - Até agora vimos no máximo 10...
- Quais são esses algarismos?
  - 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F



Conversão Hex-Dec e Dec-Hex

- $3F_{16}$ para decimal:

  $Fx16^0 + 3x16^1$ (lembrando que F representa 15)
- Faça o mesmo para $1C3_{16}$ e 238_{16}
- 30_{10} e 1000_{10} para hexadecimal com o método das divisões sucessivas?

Conversão Hex-Bin e Bin-Hex

- Para converter de e para binário, é só fazer o mesmo que com o sistema Octal, mas agora com 4 bits por algarismo (já que $16 = 2^4$)
- Por exemplo, C13 e 1ED em binário?
- E quanto é 10011000 em Hexadecimal?

Conversão Hex-Bin e Bin-Hex

- Pela conversão com o binário ser simples e ele ser mais “compacto”, o hexadecimal é muito usado na computação.
- Na página a seguir, uma tabela com os sistemas que vimos.
- Próxima aula: operações aritméticas nos sistemas novos...

Dec	Hex	Oct	Bin
0	0	000	00000000
1	1	001	00000001
2	2	002	00000010
3	3	003	00000011
4	4	004	00000100
5	5	005	00000101
6	6	006	00000110
7	7	007	00000111
8	8	010	00001000
9	9	011	00001001
10	A	012	00001010
11	B	013	00001011
12	C	014	00001100
13	D	015	00001101
14	E	016	00001110
15	F	017	00001111

Dec	Hex	Oct	Bin
16	10	020	00010000
17	11	021	00010001
18	12	022	00010010
19	13	023	00010011
20	14	024	00010100
21	15	025	00010101
22	16	026	00010110
23	17	027	00010111
24	18	030	00011000
25	19	031	00011001
26	1A	032	00011010
27	1B	033	00011011
28	1C	034	00011100
29	1D	035	00011101
30	1E	036	00011110
31	1F	037	00011111

Dec	Hex	Oct	Bin
32	20	040	00100000
33	21	041	00100001
34	22	042	00100010
35	23	043	00100011
36	24	044	00100100
37	25	045	00100101
38	26	046	00100110
39	27	047	00100111
40	28	050	00101000
41	29	051	00101001
42	2A	052	00101010
43	2B	053	00101011
44	2C	054	00101100
45	2D	055	00101101
46	2E	056	00101110
47	2F	057	00101111

Dec	Hex	Oct	Bin
48	30	060	00110000
49	31	061	00110001
50	32	062	00110010
51	33	063	00110011
52	34	064	00110100
53	35	065	00110101
54	36	066	00110110
55	37	067	00110111
56	38	070	00111000
57	39	071	00111001
58	3A	072	00111010
59	3B	073	00111011
60	3C	074	00111100
61	3D	075	00111101
62	3E	076	00111110
63	3F	077	00111111

Dec	Hex	Oct	Bin
64	40	100	01000000
65	41	101	01000001
66	42	102	01000010
67	43	103	01000011
68	44	104	01000100
69	45	105	01000101
70	46	106	01000110
71	47	107	01000111
72	48	110	01001000
73	49	111	01001001
74	4A	112	01001010
75	4B	113	01001011
76	4C	114	01001100
77	4D	115	01001101
78	4E	116	01001110
79	4F	117	01001111

Dec	Hex	Oct	Bin
80	50	120	01010000
81	51	121	01010001
82	52	122	01010010
83	53	123	01010011
84	54	124	01010100
85	55	125	01010101
86	56	126	01010110
87	57	127	01010111
88	58	130	01011000
89	59	131	01011001
90	5A	132	01011010
91	5B	133	01011011
92	5C	134	01011100
93	5D	135	01011101
94	5E	136	01011110
95	5F	137	01011111

Dec	Hex	Oct	Bin
96	60	140	01100000
97	61	141	01100001
98	62	142	01100010
99	63	143	01100011
100	64	144	01100100
101	65	145	01100101
102	66	146	01100110
103	67	147	01100111
104	68	150	01101000
105	69	151	01101001
106	6A	152	01101010
107	6B	153	01101011
108	6C	154	01101100
109	6D	155	01101101
110	6E	156	01101110
111	6F	157	01101111

Dec	Hex	Oct	Bin
112	70	160	01110000
113	71	161	01110001
114	72	162	01110010
115	73	163	01110011
116	74	164	01110100
117	75	165	01110101
118	76	166	01110110
119	77	167	01110111
120	78	170	01111000
121	79	171	01111001
122	7A	172	01111010
123	7B	173	01111011
124	7C	174	01111100
125	7D	175	01111101
126	7E	176	01111110
127	7F	177	01111111

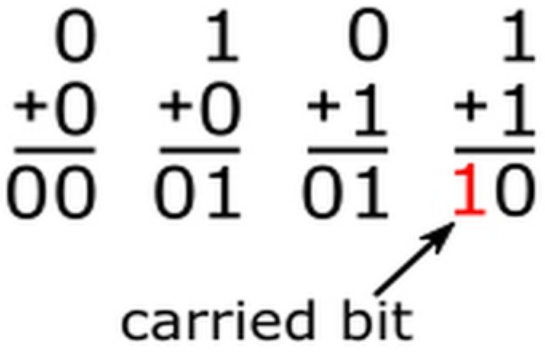
Operações com binários

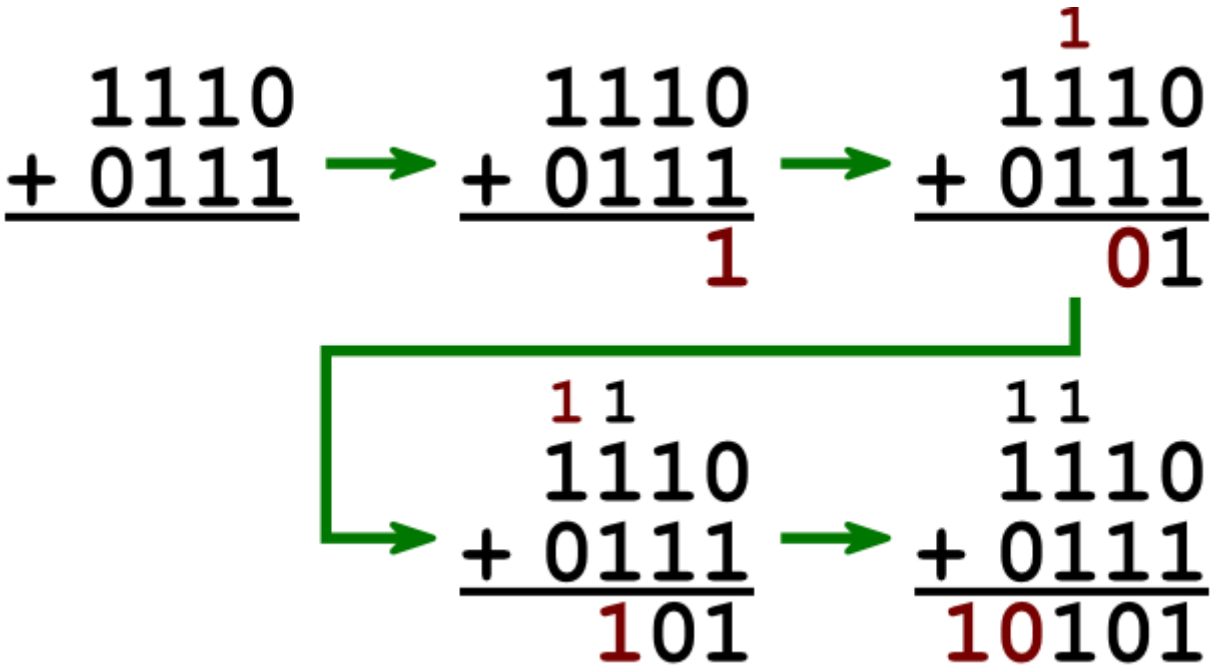
- Soma
- Subtração
- Multiplicação

  (Não usaremos a divisão na análise de circuitos lógicos)

Operações com binários

- Soma
  - Idêntica à soma decimal. Lembrando sempre de considerar o *carry* (o “vai um”) quando a soma der mais de 10_2





Operações com binários

- Soma
  - Exemplos:

    11001 + 1011

    11111 + 11111

Operações com binários

- Subtração
  - Idêntica à subtração decimal. Lembrando sempre de considerar o “empréstimo” quando subtraímos um número menor de um maior

	Minuend		Subtrahend		Difference	Borrow out
Rule 1	0	–	0	=	0	
Rule 2	0	–	1	=	1	and borrow 1
Rule 3	1	–	0	=	1	
Rule 4	1	–	1	=	0	

Operações com binários

- Subtração
  - Exemplos:

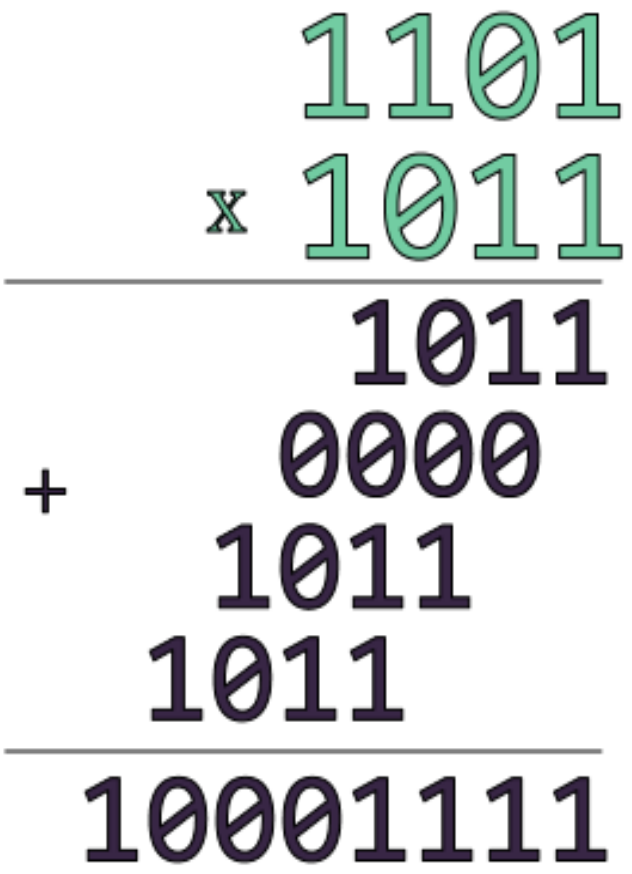
    111 – 100

    1000 – 111

    10010 – 10001

Operações com binários

- Multiplicação
  - Nesse caso não há diferença alguma em relação à multiplicação decimal
  - É ainda **mais simples,** pois só multiplicamos por 0 ou 1!



Operações com binários

- Multiplicação
  - Exemplos:

    100101 * 1001

    1111 * 1111

    10110 * 101

Números binários

- Representando positivos e negativos
  - Bit de Sinal à direita

    0 para positivo, 1 para negativo
  - Ex:

    +35 = **0**0100011

    -73 = **1**1001001
- Como o computador “sabe” qual é o bit de sinal?
  - Pode ser convencionado previamente, por exemplo, um número máximo de bits. Pode ser, por exemplo, 1 byte (8 bits). Então o oitavo bit sempre é o sinal
  - Com 8 bits, conseguimos representar 255 números (porque?)

- Curiosidade: antigamente, quando memória (RAM) e espaço (ROM) eram escassos, era comum limitar o tamanho de memória usado para armazenar dados a 1 byte ou outros valores.

Números binários

- Pergunta, se alocamos 1 byte, ou 8 bits, para um dado na memória, o que acontece se uma operação der um resultado maior?
- Lembrando que 2^8-1 é 255_{10} que é 11111111_2