Arquitectura de Computadores I

Representação de Números e Aritmética Binária

António M. Gonçalves Pinheiro

Departamento de Física
Universidade da Beira Interior
Covilhã - Portugal

pinheiro@ubi.pt

Aritmética Binária

- Base Binária
- Base Hexadecimal
- Base Octal
- Representação de Números
- Números Inteiros e Reais
- Números Inteiros sem e com sinal
- Números Reais - Representações de Ponto Fixo e Ponto Flutuante
- Códigos BCD
- Circuitos Aritméticos
- ALU - Unidades Lógicas e Aritméticos

Base Numéricas

Base b

$(a_{N-1}a_{N-2}...a_2a_1a_0)_b = (a_{N-1}\, b^{N-1} + a_{N-2}\, b^{N-2} + ... + a_2\, b^2 + a_1\, b + a_0)_{10}$, com $a_i \in \{0, 1, ..., b-1\}$

Base Binária

$(b_{N-1}b_{N-2}...b_2b_1b_0)_2 = (b_{N-1}\, 2^{N-1} + b_{N-2}\, 2^{N-2} + ... + b_2\, 2^2 + b_1\, 2 + b_0)_{10}$, com $b_i \in \{0, 1\}$

Exemplo:

$(10101001)_2 = 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2 + 1 =$

$= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^3 + 1 = 128 + 32 + 8 + 1 = (169)_{10}$

Base Numéricas

Base b

$(a_{N-1}a_{N-2}...a_2a_1a_0)_b = (a_{N-1}\ b^{N-1}+a_{N-2}\ b^{N-2}+...+a_2\ b^2+a_1\ b+a_0)_{10}$, com $a_i \in \{0, 1, ..., b-1\}$

Base Hexadecimal

$(x_{N-1}x_{N-2}...x_2b_1x_0)_{16} = (x_{N-1} \times 16^{N-1} + x_{N-2} \times 16^{N-2} + ... + x_2 \times 16^2 + x_1 \times 16 + x_0)_{10}$, com $x_i \in \{0, 1..., 8, 9, \text{A,B,C,D,E,F}\}$

Exemplo:

$$(3\text{FB})_{16} = 3 \times 16^2 + \text{F} \times 16 + \text{B} =$$
$$= 3 \times 16^2 + 15 \times 16 + 11 = 768 + 240 + 11 = (1019)_{10}$$

Nota:

$3_{16}=(0011)_2$
$F_{16}=(15)_{10}=(1111)_2$
$B_{16}=(11)_{10}=(1011)_2$

$$(3\text{FB})_{16} = (0011\,1111\,1011)_2 = 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2 + 1 = (1019)_{10}$$

Base Numéricas

Base b

$(a_{N-1}a_{N-2}...a_2a_1a_0)_b = (a_{N-1}\ b^{N-1}+a_{N-2}\ b^{N-2}+...+a_2\ b^2+a_1\ b+a_0)_{10}$, com $a_i \in \{0,1,...,b-1\}$

Base Octal

$(o_{N-1}o_{N-2}...o_2b_1o_0)_8 = (o_{N-1}\times 8^{N-1}+o_{N-2}\times 8^{N-2}+...+o_2\times 8^2+o_1\times 8+o_0)_8$, com $o_i \in \{0,1...,7\}$

Exemplo:

$$(1773)_8 = 1\times 8^3+7\times 8^2+7\times 8+3 =$$
$$= 512+448+56+3 = (1019)_{10}$$

Nota:

$1_8=(001)_2$
$7_8=(111)_2$
$7_8=(111)_2$
$3_8=(011)_2$

$$(3\mathbf{FB})_{16} = (001\,111\,111\,011)_2 = 2^9+2^8+2^7+2^6+2^5+2^4+2^3+2+1 = (1019)_{10}$$

Base Numéricas

$b = 10$	$b = 2$				$b = 16$	$b = 8$
0	0	0	0	0	**0**	0
1	0	0	0	1	**1**	1
2	0	0	1	0	**2**	2
3	0	0	1	1	**3**	3
4	0	1	0	0	**4**	4
5	0	1	0	1	**5**	5
6	0	1	1	0	**6**	6
7	0	1	1	1	**7**	7
8	1	0	0	0	**8**	10
9	1	0	0	1	**9**	11
10	1	0	1	0	**A**	12
11	1	0	1	1	**B**	13
12	1	1	0	0	**C**	14
13	1	1	0	1	**D**	15
14	1	1	1	0	**E**	16
15	1	1	1	1	**F**	17

Aritmética Binária - Números Inteiros

Números Inteiros sem sinal

Usam normalmente a representação binária.

com um byte: $b_7b_6b_5b_4b_3b_2b_1b_0$ entre 0 e 2^8-1=255
com 16 bits: $b_{15}b_{14}....b_3b_2b_1b_0$ entre 0 e 2^{16}-1
com N bits: $b_{N-1}b_{N-2}....b_3b_2b_1b_0$ entre 0 e 2^N-1

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal

$$
\begin{array}{cccccccccc}
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1
\end{array}
\qquad
\begin{array}{cccc}
 & & 9 & 3 \\
+ & 1 & 5 & 8 \\
\hline
 & 2 & 5 & 1
\end{array}
$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal

$$
\begin{array}{cccccccccccccc}
 & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & & & & & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & & & & 9 & 3 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & & + & 1 & 5 & 8 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & & 2 & 5 & 1
\end{array}
$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros sem sinal - "Overflow"

1			**1**	**1**	**1**							
	1	1	0	1	1	1	0	1		2	2	1
+	1	0	0	1	1	1	1	0	+	1	5	8
1	0	1	1	1	1	0	1	1		1	2	3
										3	**7**	**9**

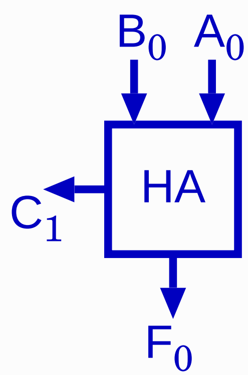
NOTA: Quando o número de bits não é suficiente para representar o resultado final, diz-se que ocorreu um "*overflow*".

Aritmética Binária - Números Inteiros

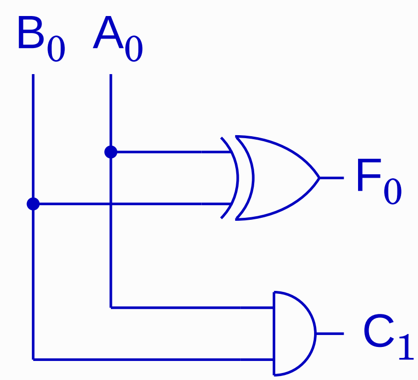
Soma de Números Inteiros sem sinal - Circuitos Somadores

"half-adder"

Somador de dois bits



B_0	A_0	C_1	F_0
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

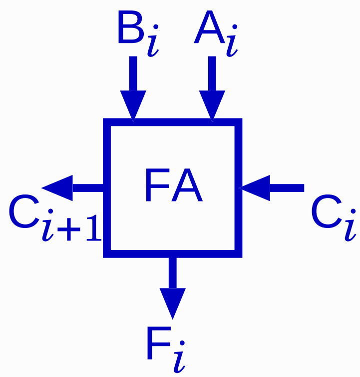


Aritmética Binária - Números Inteiros

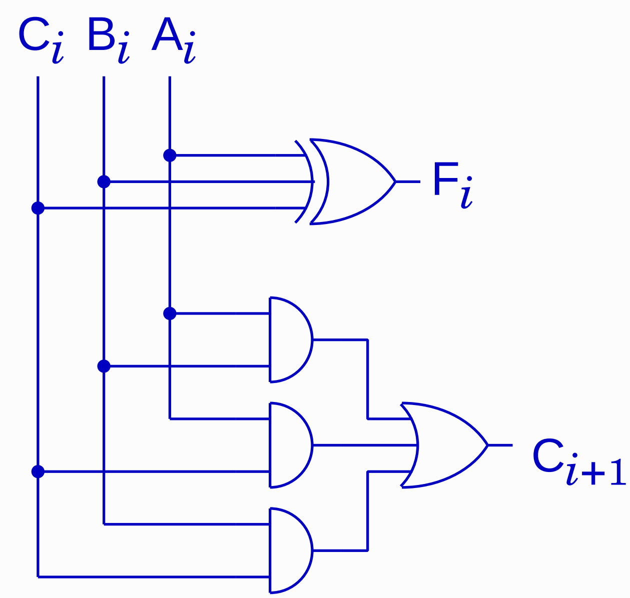
Soma de Números Inteiros sem sinal - Circuitos Somadores

"full-adder"

Somador de três bits

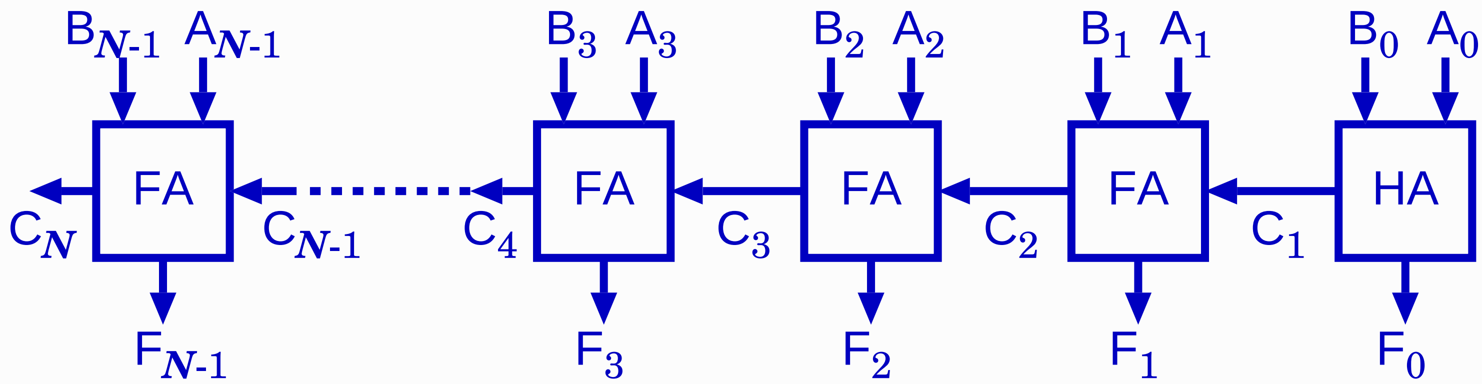


C_i	B_i	A_i	C_{i+1}	F_i
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



Aritmética Binária - Números Inteiros

Somador de Números Inteiros sem sinal com N bits



Aritmética Binária - Números Inteiros

Exemplo

Projecte um circuito baseado numa célula que iterativamente verifique qual é o maior de dois números.

Aritmética Binária - Números Inteiros

Números Inteiros com sinal

Sinal módulo

bit mais significativo representa o sinal ($- \rightarrow 1$)

Ex: 10010111= $-2^4 + 2^2 + 2 + 1 \; = \; -23$

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}-1\right), ..., 2^{N-1}-1\right\}$

Complemento para UM

bit mais significativo representa o sinal ($- \rightarrow 1$)

o módulo de um número negativo obtem-se por negação de todos os bits

Ex: 11101000= - (00010111)= $-(2^4 + 2^2 + 2 + 1) \; = \; -23$

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}-1\right), ..., 2^{N-1}-1\right\}$

Aritmética Binária - Números Inteiros

Números Inteiros com sinal

Complemento para DOIS

bit mais significativo representa o sinal (- $\rightarrow$ 1)
o módulo de um número negativo obtem-se por negação de todos os bits seguido de soma por 1
Ex: 11101001= - (00010110)+1 = - (00010111) = - $2^4 + 2^2 + 2 + 1$ = -23

Intervalo Representado com N bits: $\left\{-\left(2^{N-1}\right), ..., 2^{N-1} - 1\right\}$

NOTA: Mais utilizado pois permite operações aritméticas directas.

100000 em C_2 representa o número -2^{N-1}
(Se N=8, 10000000=-128)

Aritmética Binária - Números Inteiros

Soma de Números Inteiros com sinal

$$
\begin{array}{cccccccccccccc}
 & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & & & & & \\
 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & & & & 9 & 3 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & & + & - & 9 & 8 \\
\hline
 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & & - & & 5
\end{array}
$$

$$1\,0\,0\,1\,1\,1\,1\,0 \xrightarrow{C_2} 0\,1\,1\,0\,0\,0\,1\,0 = 98$$

$$1\,1\,1\,1\,1\,0\,1\,1 \xrightarrow{C_2} 0\,0\,0\,0\,0\,1\,0\,1 = 5$$

Aritmética Binária - Números Inteiros

Soma de Números Inteiros com sinal

$$\begin{array}{ccccccccccccc}
 & & & \mathbf{1} & \mathbf{1} & \mathbf{1} & & & & & & & \\
 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & & - & 3 & 5 \\
+ & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & + & - & 9 & 8 \\
\hline
 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & & 1 & 2 & 3 \\
 & & & & & & & & & - & \mathbf{1} & \mathbf{3} & \mathbf{3}
\end{array}$$

$$1\,1\,0\,1\,1\,1\,0\,1 \xrightarrow{C_2} 0\,0\,1\,0\,0\,0\,1\,1 = 35$$

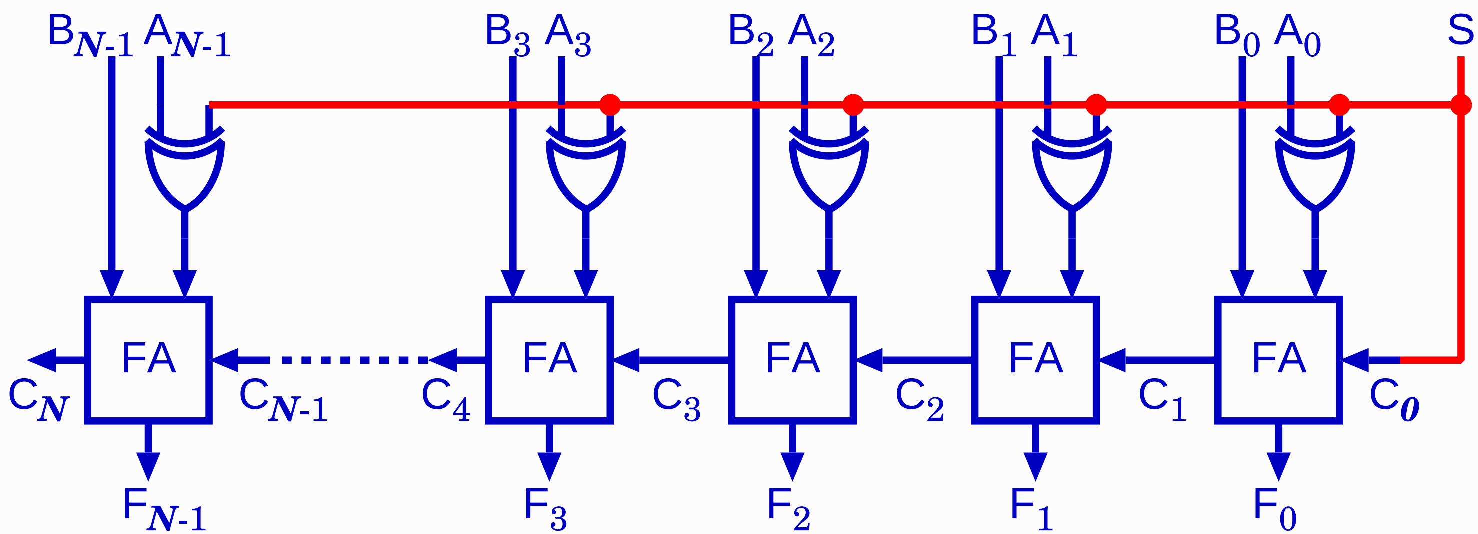
$$1\,0\,0\,1\,1\,1\,1\,0 \xrightarrow{C_2} 0\,1\,1\,0\,0\,0\,1\,0 = 98$$

NOTA: “Overflow”negativo porque $-133 < -128 = -2^7$

Aritmética Binária - Números Inteiros

Somador/Subtractor de Números Inteiros com Sinal com N bits

S=0 F=B+A Somador
S=1 F=B−A Subtractor

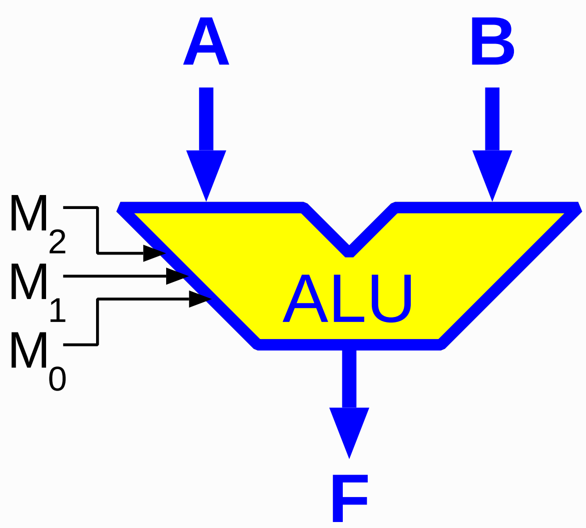


S=1 faz-se negação de todos os bits do número A e soma-se 1 → ***Complemento para 2***.
NOTA: $0 \oplus A = A$, $1 \oplus A = \overline{A}$

Aritmética Binária - Números Inteiros

Unidade Lógica e Aritmética - ALU

Sistema combinacional concebido para fazer diferentes operações Aritméticas e Lógicas entre duas palavras binárias



M_2	M_1	M_0	Operação
0	0	0	A + B
0	0	1	A - B
0	1	0	A + 1
0	1	1	A - 1
1	0	0	$A \cdot B$
1	0	1	$A + B$
1	1	0	$A \oplus B$
1	1	1	$\overline{A}$

Nota:

M_2=0 - Operações Aritméticas

M_2=0 - Operações Lógicas

Aritmética Binária - Números Inteiros

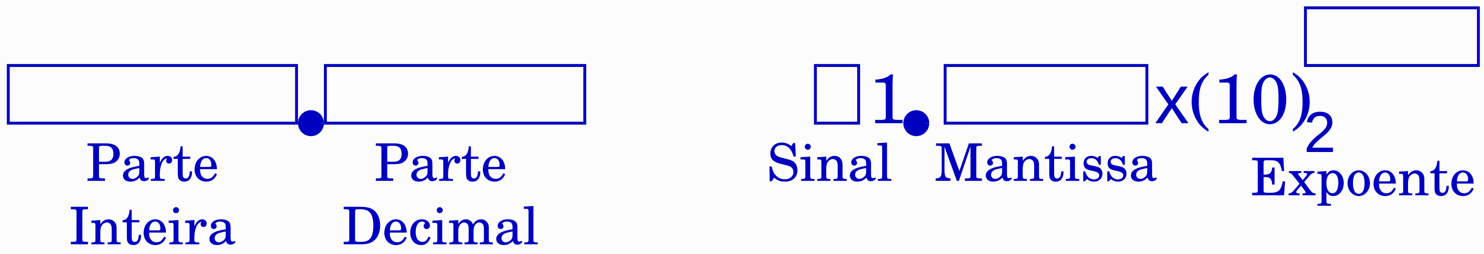
Exemplo

Projecte um circuito de que resulte uma variável O_v que para o circuito somador/subtractor estudado sinalize sempre que há um “overflow” negativo ou positivo.

Aritmética Binária - Números Reais

Ponto Fixo

Ponto Flutuante



Underflow -
Número com valor absoluto muito pequeno diferente de zero que não pode ser representado.
Overflow -
Número com valor absoluto muito grande que não pode ser representado.

FPU - “Floating Point Unit”
Circuito que se destina a fazer operações com números em vírgula flutuante.

Aritmética Binária - Números Reais

Norma do IEEE 754-1985

Nome	Número de bits	Sinal	Mantissa	Expoente	Polarização
Precisão Simples	32	1	23	8	127
Precisão Dupla	64	1	52	11	1023

Valor representado em decimal:

$$(-1)^{\text{Sinal}} \times \left(1 + \sum_{n=1}^{p-1} \text{Mantissa}(n) \times 2^{-n}\right) \times 2^{\text{Expoente-Polarização}}$$

Formato:

Sinal	Expoente	Mantissa

Casos especiais:

Tipo	**Expoente**	**Mantissa**
Zero	00...000	00...000
Infinito	11...111	00...000
NaN	11...111	$\neq 0$
Num. não normalizado	00...000	$\neq 0$

Códigos BCD - "Binary Code Decimal"

Dígito Decimal	BCD 8421	BCD 2421	BCD X-3
0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1
9	1 0 0 1	1 1 1 1	1 1 0 0