Sistema de Numeração em Computação

B → base do sistema de numeração

n → quantidade de dígitos disponíveis para representar os números

$$a = \sum_{i=0}^{n-1} (x_i \cdot B^i)$$

$$x_i \in \{0, 1, \dots, B-1\}$$

Para uma determinada base B, empregando-se n dígitos tem-se: B^n combinações distintas, ou seja, B^n números distintos.

ex: Base é decimal empregando-se 3 dígitos para a sua representação:

$$10^3 = 1000 \text{ números possíveis}$$

999 a ∅

Repare que a base binária requer uma quantidade grande de dígitos para representar números:

ex: 255 = 11111111
decimal binário

Números binários terão grandes cadeias de "0" e "1", o que poderá

poderiam causar erros visuais quando realizarmos leituras. Assim costuma-se empregar a base 8 e base 16 (Hexadecimal) na representação de uma forma mais compacta para representar números binários

A Tabela 2.1 abaixo lista os primeiros 16 números em binário, decimal, octal e hexadecimal.

Binário	Decimal	Octal	Hexadecimal
0000	0	00	0
0001	1	01	1
0010	2	02	2
0011	3	03	3
0100	4	04	4
0101	5	05	5
0110	6	06	6
0111	7	07	7
1000	8	10	8
1001	9	11	9
1010	10	12	A
1011	11	13	B
1100	12	14	C
1101	13	15	D
1110	14	16	E
1111	15	17	F

Tabela 2.1 - Números em binário, decimal, octal e hexadecimal

Cada dígito do sistema binário, (0 e 1), é denominado de bit ($\underline{b}$inary dig$\underline{i}$T)

4 bits → nibble

8 bits → byte

Múltiplos do sistema decimal

K, M, G, T, P

No sistema binário

Kb e KB ⇢ $2^{10} = 1024$

Kilo bit   Kilo byte

$$\text{Megabit} = 2^{10} \cdot 2^{10} = 2^{10} \cdot \text{Kbit}$$

Representação de números

Inteiro positivo

Sinal - magnitude

Complemento de 1

complemento de 2.

Inteiros Positivos:

n dígitos $\Rightarrow 2^n$ números

faixa de representação:

0 a $2^n - 1$

$2^5 = 32 \Rightarrow$ 00000 a 11111

(0) decimal (31 decimal)

$2^8 = 256 \Rightarrow$ 0000 0000 1111 1111

(0) decimal (255 decimal)

00 Hexadecimal FF Hexadecim

- soma-se normalmente
- pode haver estouro (overflow)

Representação sinal-magnitude

→ Representação de números positivos e negativos

→ Utiliza-se, em geral, o dígito mais significativo para representar o sinal : 1 --- "−"
0 --- "+"

→ Redução da faixa de representação, assim tem-se agora (n-1) dígitos para representar a magnitude

→ Representação da faixa: $[-(2^{n-1}-1), (2^{n-1}-1)]$

ex: $n = 4$

$+(2^3 - 1) = +7 \Rightarrow 0111$

$-(2^3 - 1) = -7 \Rightarrow 1111$

→ 2 representações para o valor "0" → 0000
→ 1000

→ É possível representar no máximo : $2^n - 1$ valores distintos

→ Pode ser vista como composta por 2 parcelas :

S(a)M(a) → magnitude a

↳ sinal do número a

→ As operações aritméticas de soma e subtração em computador requer vários passos :

Inferir sobre as magnitudes

Inferir sobre os sinais

7

S(a)	S(c)	S(d)	M(d)	Exemplo
+	+	+	M(a)+M(c)	5 + 7 = 12
–	–	–	M(a)+M(c)	-5 + -7 = -12
+	–	se M(a)≥M(c), + se M(a)<M(c), –	M(a)–M(c) M(c)–M(a)	7 + -5 = 2 5 + -7 = -2
–	+	se M(a)>M(c), – se M(a)≤M(c), +	M(a)–M(c) M(c)–M(a)	-7 + 5 = -2 -5 + 7 = 2

Representação em complemento de 1

→ Permitir que a operação de soma seja realizada de forma única.

→ números positivos na forma normal (tipo inteiro po

→ números negativos na forma de complemento

- O complemento de um número "a" é obtido subtraindo-se esto número da maior quantidade representável. (Teoria)
- Ou, a representação negativa é obtida simplesmente invertendo-se (ou complementando-se) cada bit do número.

Ex $4 = 0100 \Rightarrow -4 = 1011$

$7 = 0111 \Rightarrow -7 = 1000$

Faixa de representação:

$$\left[-\left(\frac{B^n}{2}-1\right), +\left(\frac{B^n}{2}-1\right)\right]$$

Ex: base binária com 4 dígitos

faixa: $\left(-\left(\frac{2^4}{2}-1\right), \frac{2^4}{2}-1\right)$

$-7, 7$

binário	decimal
0111	7
0110	6
0101	5
0100	4
0011	3
0010	2
0001	1
0000	0
1111	-0
1110	-1
1101	-2
1100	-3
1011	-4
1010	-5
1001	-6
1000	-7

Representação negativa pode ser obtida, simplesmente, complementando-se cada bit do número

Ex: 4 = 0100 => -4 = 1011

7 = 0111 => -7 = 1000

→ Na base 2 (binária) com representação em complemento de 1 tem-se:

Se o bit mais significativo for "1" => nº negativo

caso contrário, MSB = "0" => nº positivo.

→ Problema:

Dupla representação para o zero

→ A magnitude de um número positivo é direta
" " " " " negativo é obtida aplicando-se
o complemento.

Ex: 1000 em complemento de 1

↳ significa que o número é negativo

Qual é a sua magnitude? (seu valor sem considerarmos o sinal)

R: complemento (1000) => $0\underbrace{111}_{7_2}$

1000 = -7

complemento do 1 decimal com sinal

"

Soma de binário na representação em complemento de 1.

$$
\begin{array}{rcr}
-4 & = & 1011 \\
+3 & & 0011 \\
\hline
-1 & & 1110
\end{array}
\qquad\qquad
\begin{array}{rr}
-4 & 1011 \\
-2 & 1101 \\
\hline
-6 & 11000 \\
 & +1 \\
 & 1001
\end{array}
$$

Alguns resultados necessitam de correção:

→ Para se obter a soma correta em complemento de 1, basta somar o vai-um ao resultado.

$$
\begin{array}{rr}
 & -0 \\
+ & -0 \\
\hline
\end{array}
\qquad
\begin{array}{r}
1111 \\
1111 \\
\hline
11110 \\
+ \\
\hline
1111\ (-0)
\end{array}
\qquad\qquad
\begin{array}{rl}
1111 & (-0) \\
0100 & (+4) \\
\hline
10011 & \\
+1 & \\
\hline
0100 &
\end{array}
$$

subtração:

Regra prática complementa-se o subtraendo e efetua-se a soma em complemento de 1.

Representação em complemento de 2

Faixa de representação

$$\left(-\frac{2^n}{2}, \frac{2^n}{2}-1\right)$$

$n=4 \Rightarrow$ faixa $(-8, 7)$

$n=8 \Rightarrow$ faixa $(-128, 127)$

- Não existe mais a dupla representação do zero.
- Faixa positiva continua a mesma.
- Faixa negativa sofre um deslocamento de uma unidade
- Para a base 2 : o dígito mais significativo indicará o sinal:

  0 => positivo

  1 => negativo (igual aos sistemas anteriores)

• Regra Prática para se obter a representação negativa: complementa-se os bits (semelhante ao complemento de 1) e soma-se 1.

complemento de 2	decimal com sinal	
0111	7	
0110	6	
0101	5	
0100	4	
0011	3	
0010	2	
0001	1	
0000	0	
1111	-1	→ Todos bits com "1" ⇒ -1
1110	-2	
1101	-3	
1100	-4	
1011	-5	
1010	-6	
1001	-7	
1000	-8	→ Apenas 1 bit (msb) em "1"

- Soma feita diretamente. (usada em computação)
- Subtração: complementa-se (em comp. de 2) o subtraendo e efetua-se a soma.

- Overflow: ESTOURO da representação.

Binário	Inteiros positivos	Sinal-magnitude	Complemento de 1	Complemento de 2
0000	0	+0	0	0
0001	1	1	1	1
0010	2	2	2	2
0011	3	3	3	3
0100	4	4	4	4
0101	5	5	5	5
0110	6	6	6	6
0111	7	7	7	7
1000	8	–0	–7	–8
1001	9	–1	–6	–7
1010	10	–2	–5	–6
1011	11	–3	–4	–5
1100	12	–4	–3	–4
1101	13	–5	–2	–3
1110	14	–6	–1	–2
1111	15	–7	–0	–1

Tabela 2.16 - Interpretação de números binários

Alu abaixo permite fazer subtração em complemento de 1 e complemento de 2. Lembrando que para somarmos números com mais de 1bit deve-se cascatear vários circuitos de ALU . No caso de usarmos subtração em complemento de 2, deve-se colocar “1” na entrada **Vem-um** do somador da parte LSB.

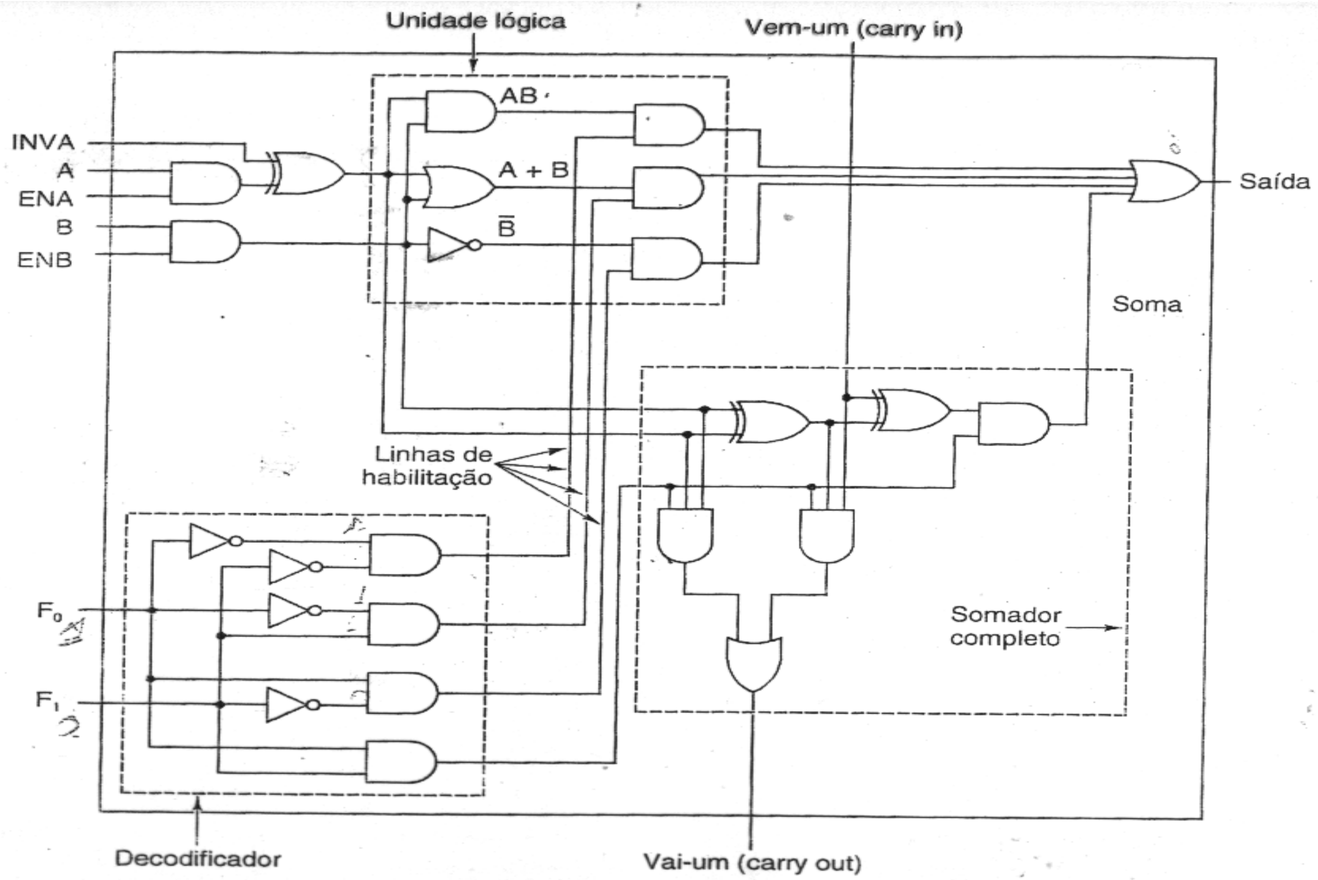

Fig. 3.19 UAL de 1 bit.