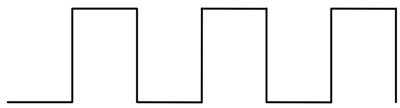
Codificação

1. Introdução

- A unidade básica de memória é o digito binário (bit).
- Para representar diferentes em memória é necessário que o bit armazene pelo menos 2 valores.
- A informação pode ser armazenada através da distinção entre valores diferentes de alguma grandeza física contínua.

C 2 R r {
! +
@ $\longleftrightarrow$ **codificação**



caracteres/números
símbolos especiais
caracteres gráficos

grandezas físicas
contínuas (ex. i ,V)
representando valores

- Problema: quanto mais valores tiverem de ser distinguidos, menor é a separação entre valores adjacentes e menor é a confiabilidade da memória.
- Sistemas de numeração: $2 \leq SN \leq \infty$
- Dentre os vários sistemas de numeração possíveis o mais eficiente é o binário (distingue somente 2 estados de grandezas físicas).
- Os computadores digitais utilizam o sistema de numeração binário como base para representar a informação. Mas a forma como isso é feito pode variar de fabricante para fabricante.

1944 / 0001 1001 0100 0100 / 0000011110011000

2. Sistemas de Numerações Comuns

Binário	Base 2
Octal	Base 8
Decimal	Base 10
Hexadecimal	Base 16

Exemplos de codificação
- Código BCD
- Código 7-segmentos

Sistema Decimal
- Dígitos {0 . . .9}
- Número decimal em potência de 10

Sistema Binário
- Dígitos {0 ,1}
- Número binário em potência de 2

3. Conversão

Binário para decimal

Multiplicar cada dígito pelas potências da base e somar os resultados

$D_{n-1} . 2^{n-1} + D_{n-2} . 2^{n-2} + ... + D_0 . 2^0$

Conversão de decimal para binário

Divisões sucessivas pela base (2)

Ex: $(24)_{10}$ -> $(?)_2$

24 / 2 =	12 / 2=	6 / 2 =	3 / 2 =	$\underline{1}$
$\lfloor$ $\underline{0}$	$\lfloor$ $\underline{0}$	$\lfloor$ $\underline{0}$	$\lfloor$ $\underline{1}$	

$(11000)_2$

Ex: $(227)_{10}$ -> $(?)_2$

4 Operações aritméticas em binário

Adição		Ex:	10011010
Similar ao decimal		+	11011100
Método prático (casar 1´s)			------------
	carry	1<-	01110110

Complemento de dois

É uma forma de representar números negativos em binário. Ex: $(-500)_{10}$ -> $(?)_2$

	0	+
O bit mais significativo = sinal		
O restante é magnitude	1	-

Complemento de dois = complemento de um + 1

Ex:

13	->	00001101	->	11110010	comp.1
			+	00000001	1

-13	------------------------>			11110011	comp.2

Somando 9 -13 = -4		00001001	
	+	11110011	1
		------------	
bit sinal(1)	->	11111100	(negativo)
comp.1		00000011	
+1		00000001	
comp.2		00000100	(magnitude)

Com 8 bits temos números sem sinal de 0 a 255

Com 8 bits temos números sinalizados de

01111111 +127 ... 0 ... 10000000 -128

Subtração binária

Utiliza o complemento de 2 e realiza uma soma.

	$(91)_{10}$	$(01011011)_2$		01011011
-	$(46)_{10}$	$(00101110)_2$	11010001+1=	11010010
-------------		---------------		------------
	$(45)_{10}$	$(?)_2$	1<-	00101101

Multiplicação binária

-Método do lápis

	$(11)_{10}$	$(1011)_2$	multiplicando
x	$(09)_{10}$	$(1001)_2$	multiplicador
	-----------	--------------	
	$(99)_{10}$	1011	
		0000	produtos
		0000	parciais
		1011	

		01100011	produto

Simplificação do método do lápis. Os produtos parciais são o próprio multiplicando ou zero dependo do dígito do multiplicador. Faz-se então a soma e deslocamento para direita para cada dígito do multiplicador.

Divisão binária

-Método do lápis

Divisões sucessivas de partes do dividendo de acordo com o tamanho do divisor. Se a parte do dividendo for maior que o divisor, o quociente é 1. Caso contrário o quociente é 0. É usada a subtração binária no processo de obtenção do resto da divisão.Ex.

dividendo	1001000	110	divisor
-	0110		
	0110	1100	quociente
-	0110		
	0000		

-Método das subtrações sucessivas: “Lento”.

-Método subtrai e desloca: método do lápis com deslocamento à esquerda inserindo-se o próximo dígito do dividendo.

5 Sistema Hexadecimal

Dígitos {0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}

Número hexadecimal em potência de 16

16^3=4.096 $16^{2=}$256 16^1=16 16^0=1

6 Conversão

Hexadecimal para decimal

Multiplicar cada dígito pelas potências da base e somar os resultados

$D_{n-1} . 16^{n-1} + D_{n-2} . 16^{n-2} + ... + D_0 . 16^0$

Conversão de decimal para hexadecimal

Divisões sucessivas pela base (16)

Ex: $(227)_{10}$ -> $(?)_{16}$

227 / 16 = 14 / 16= 0

$\lfloor \underline{3}$ $\lfloor \underline{14}$

$(E3)_{16}$

Conversão hexadecimal binário

A cada 4 dígitos binários equivale a um dígito hexadecimal, e vice-versa.

1010=A 1011=B 1100=C 1101=D 1110=E 1111=F

Ex:

$(E3)_{16}$ = $(1110 \quad 0011)_2$

16. Operações aritméticas

Convertendo para binário/decimal
Diretamente em hexa

O “vai-um” é produzido sempre que a soma for maior ou igual a 16. Ex:

10 + FF = 10F (Soma-16^n=Hexa)

onde n é o maior número em que a potência 16^n <=Soma

Subtração Hexadecimal

-Diretamente em hexa

O empréstimo na subtração hexa significa 16 unidades. Ex:

10F -	FF	=>	F-F		=	0
			0-F	(16+0-F)=>(16-15)	=	1
			1-1	(emprestou)	=	0

6 Sistema Octal

Dígitos =\{0,1,2,3,4,5,7\}

Potência de 8

Três dígitos binários equivalem a um dígito octal.

Adição e Subtração :

7 Código BCD

Decimal codificado em binário é aplicado em:

> contadores de frequência
> voltímetros digitais
> calculadoras com display em decimal
> outros

Informação de cada dígito decimal está contida numa palavra binária de 4 bits.

BCD Decimal Ex: $(529)_{10}$ = $(0101\ 0010\ 1001)_{bcd}$

BCD	Decimal
0000	0
0001	1
0010	2
....	
1001	9
1001-1111	são inválidos.

8 Aritmética BCD

O vai-um sempre que o dígito for maior que 9. Em caso de dígito BCD inválido, deve-se somar $(110)_2$ ou $(6)_{10}$. Ex:

	35	0011 0101			9		1001
+	23	0010 0011		+	8		1000
---------		------------		-------			------
	58	0101 1000	válido	00017		1	0001 incorreto
						0	0110 add 6

						1	0111 BCD 17