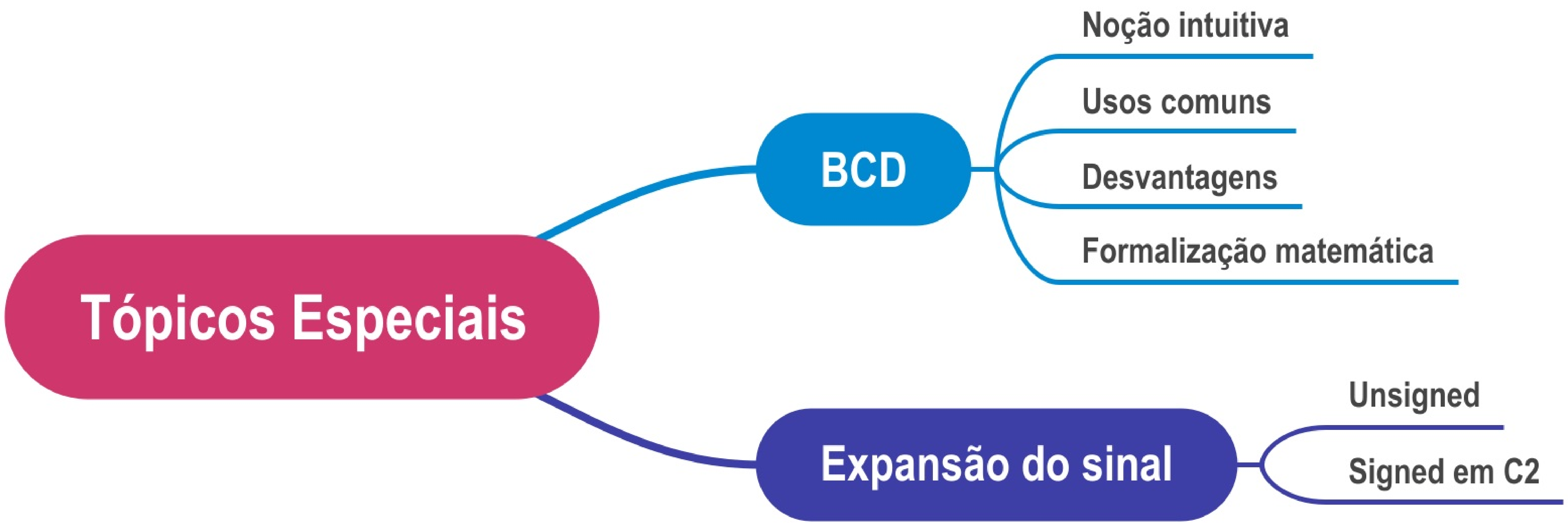
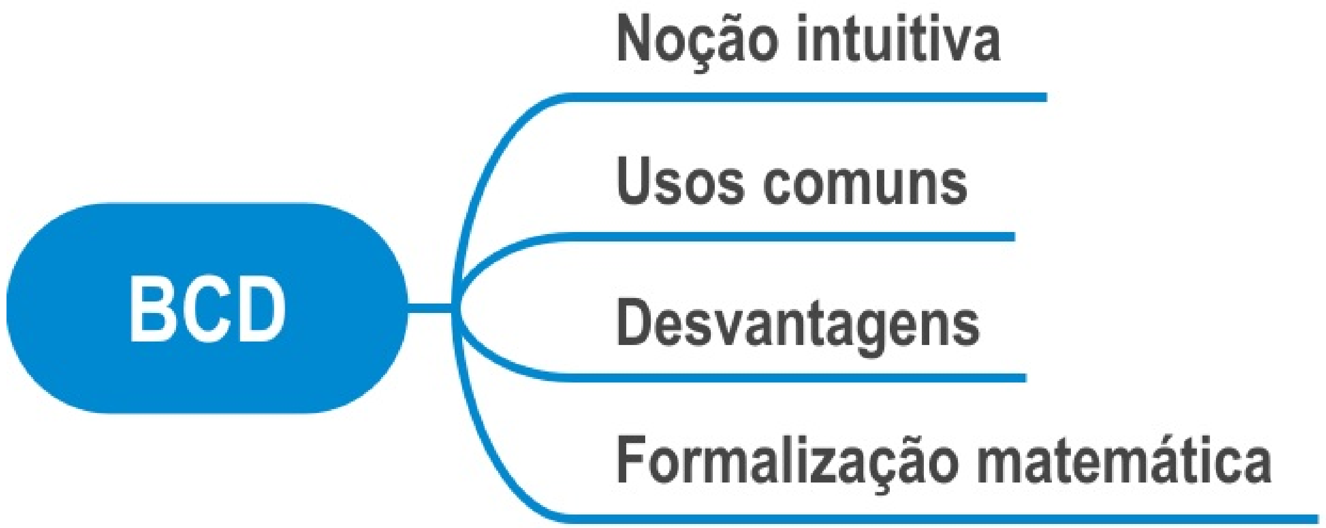
FUNDAMENTOS DA COMPUTAÇÃO

Tópicos especiais



Binary-coded decimal (BCD): decimal codificado em binário



O que é BCD?

É um conjunto de encodings para números decimais onde cada algarismo é representado por um número fixo de bits. Em geral usamos grupos de:

- **4 bits (nibble)**
- **8 bits (byte)**

Decimal	**Binário**	**BCD-4**	**BCD-8**
12	1100	00010010	0000000100000010

Decimal	**Binário**	**BCD-4**	**BCD-8**
123	1111011	000100100011	000000010000001000000011

Existem diversos encodings BCD

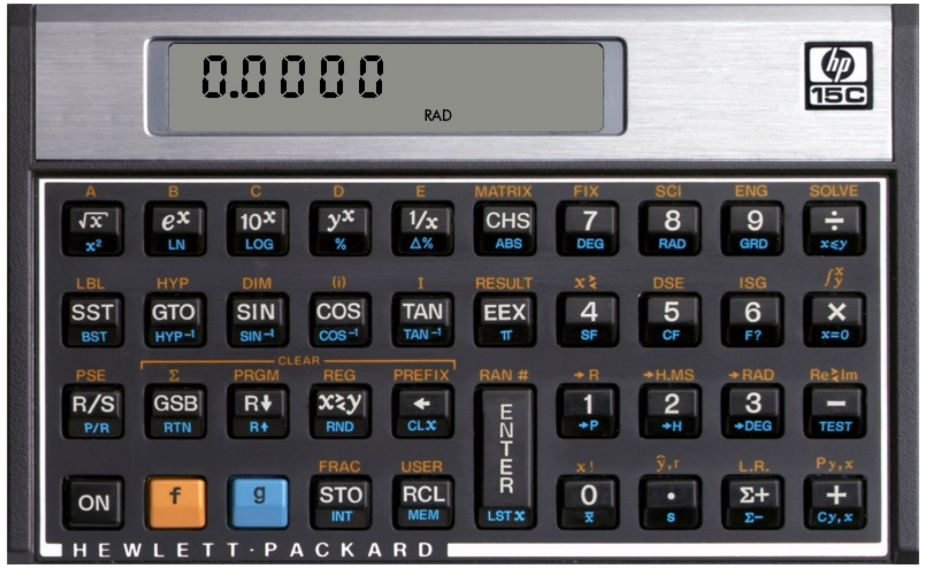
O mais comum é o Natural BCD (Simple BCD, BCD 8421), que faz o encoding dos algarismos 0 até 9:

BCD				Decimal
8	4	2	1	
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	Inválido
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

Por que usar BCD?

Atualmente é pouco usado mas encontra aplicações em situações nas quais a imprecisão dos binários em ponto fixo ou em ponto flutuante não pode ser tolerada:

- cálculos financeiros
- cálculos industriais
- dispositivos como calculadoras, relógios digitais, voltímetros digitais, contadores eletrônicos, displays
- transferência de informações decimais para dentro e para fora de sistemas digitais



Exemplo: 1,66 em Q2.8 e em BCD-4

Q2.8 (1,6563)	BCD-4 (1,66)
01.10101000	0001.01100110

Vantagens e desvantagens

Vantagens:

- precisão na representação
- fácil entendimento
- "escala" (grupos) em base 10
- conversão para caractere ou display é simples
- BCD para ponto fixo e ponto flutuante
- bits extras podem indicar sinal, infinito e valores especiais

Desvantagens:

- maior complexidade nos circuitos aritméticos
- performance menor
- só representa algarismos, ignorando letras e outros símbolos
- ocupa mais espaço do que binário

Formalização matemática

BCD inteiro, unsigned, grupos de 4 bits:

Seja x um número binário inteiro não negativo, formado por n algarismos, em notação **BCD 8421**, agrupados em g grupos de 4 bits. Dentro de cada um dos g grupos, os 4 bits serão representados por sua posição i, que varia de 0 até 3, contada da direita para a esquerda. Representaremos o vetor de algarismos desse número por $\vec{x}$. Para representarmos os algarismos individuais em seus respectivos grupos, dentro do vetor, usaremos a notação $x_{g,i}$ do seguinte modo:

$$[\underbrace{x_{g-1,3}, x_{g-1,2}, x_{g-1,1}, x_{g-1,0}}_{\text{grupo: } g-1}, \underbrace{x_{g-2,3}, x_{g-2,2}, x_{g-2,1}, x_{g-2,0}}_{\text{grupo: } g-2}, \dots, \underbrace{x_{0,3}, x_{0,2}, x_{0,1}, x_{0,0}}_{\text{grupo: } 0}]$$

A conversão $\mathrm{BCD_uD}$ (binário em BCD unsigned para decimal) de um vetor binário $\vec{x}$ com tamanho n, com g grupos de 4 bits é dada pela fórmula abaixo, onde d_g é o valor decimal de um grupo binário de 4 bits:

$$\begin{aligned}\mathrm{BCD_uD} &= \sum_{g=0}^{(n/4)-1} 10^g \times d_g \\ &= \sum_{g=0}^{(n/4)-1} \left(10^g \times \sum_{i=0}^{3} x_{g,i} 2^i\right)\end{aligned}$$

Faixa de representação de cada byte BCD:

- **não comprimido (1 decimal/byte): 0 até 9**
- **comprimido (2 decimais/byte): 0 até 99**

Formalização matemática

BCD inteiro, unsigned, grupos de b bits:

Seja x um número binário inteiro não negativo, formado por n algarismos, em notação **BCD 8421**, agrupados em g grupos de b bits. Dentro de cada um dos g grupos, os b bits serão representados por sua posição i, que varia de 0 até $b-1$, contada da direita para a esquerda. Representaremos o vetor de algarismos desse número por $\vec{x}$. Para representarmos os algarismos individuais em seus respectivos grupos, dentro do vetor, usaremos a notação $x_{g,i}$ do seguinte modo:

$$[\underbrace{x_{g-1,b-1}, x_{g-1,b-2}, x_{g-1,\ldots}, x_{g-1,1}, x_{g-1,0}}_{\text{grupo: } g-1}, \underbrace{x_{g-2,b-1}, x_{g-2,b-2}, x_{g-2,\ldots}, x_{g-2,1}, x_{g-2,0}}_{\text{grupo: } g-2}, \ldots, \underbrace{x_{0,b-1}, x_{0,b-2}, x_{0,\ldots}, x_{0,1}, x_{0,0}}_{\text{grupo: } 0}]$$

A conversão $\mathrm{BCD_uD}$ (binário em BCD unsigned para decimal) de um vetor binário $\vec{x}$ com tamanho n, com g grupos de b bits é dada pela fórmula abaixo, onde d_g é o valor decimal de um grupo binário de b bits:

$$\mathrm{BCD_uD} = \sum_{g=0}^{(n/b)-1} 10^g \times d_g$$

$$= \sum_{g=0}^{(n/b)-1} \left(10^g \times \sum_{i=0}^{b-1} x_{g,i} 2^i \right)$$

Faixa de representação de cada byte BCD:

- **não comprimido (1 decimal/byte): 0 até 9**
- **comprimido (2 decimais/byte): 0 até 99**

Expansão do sinal

Expansão do sinal: por quê?

Em algumas situações temos um binário com **n** bits e preisamos representar esse binário com **n + m** bits. Como aumentar (expandir) o binário sem alterar seu valor?

O processo depende se o número é unsigned ou se é signed (e, em nosso caso específico, em complemento de 2). Se for unsigned, basta completar com zeros à esquerda. Exemplo:

Expandir 2 e 209, em binário unsigned, de oito para dezesseis bits:

0 0 0 0 0 0 1 0

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

1 1 0 1 0 0 0 1

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

Expansão do sinal: por quê?

Se for signed, em complemento de 2, basta copiar o bit de sinal:

Expandir 20 e -20, em binário em complemento de 2, de oito para dezesseis bits:

0 0 0 1 0 1 0 0

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

1 1 1 0 1 1 0 0

_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _

O processo inverso (contração do sinal) também funciona!

Contrair 20 e -20, em binário em complemento de 2, de vinte e quaro para oito bits:

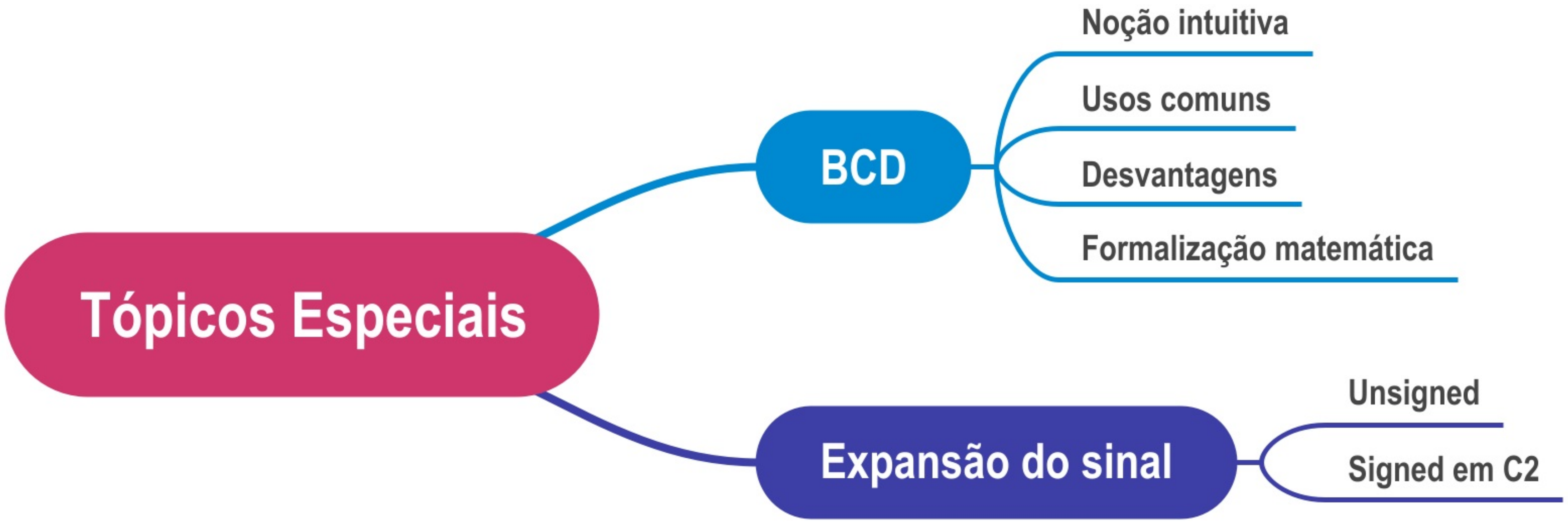
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0 0

_ _ _ _ _ _ _ _

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 0

_ _ _ _ _ _ _ _

Resumo



Tópicos Especiais
BCD
Noção intuitiva
Usos comuns
Desvantagens
Formalização matemática
Expansão do sinal
Unsigned
Signed em C2