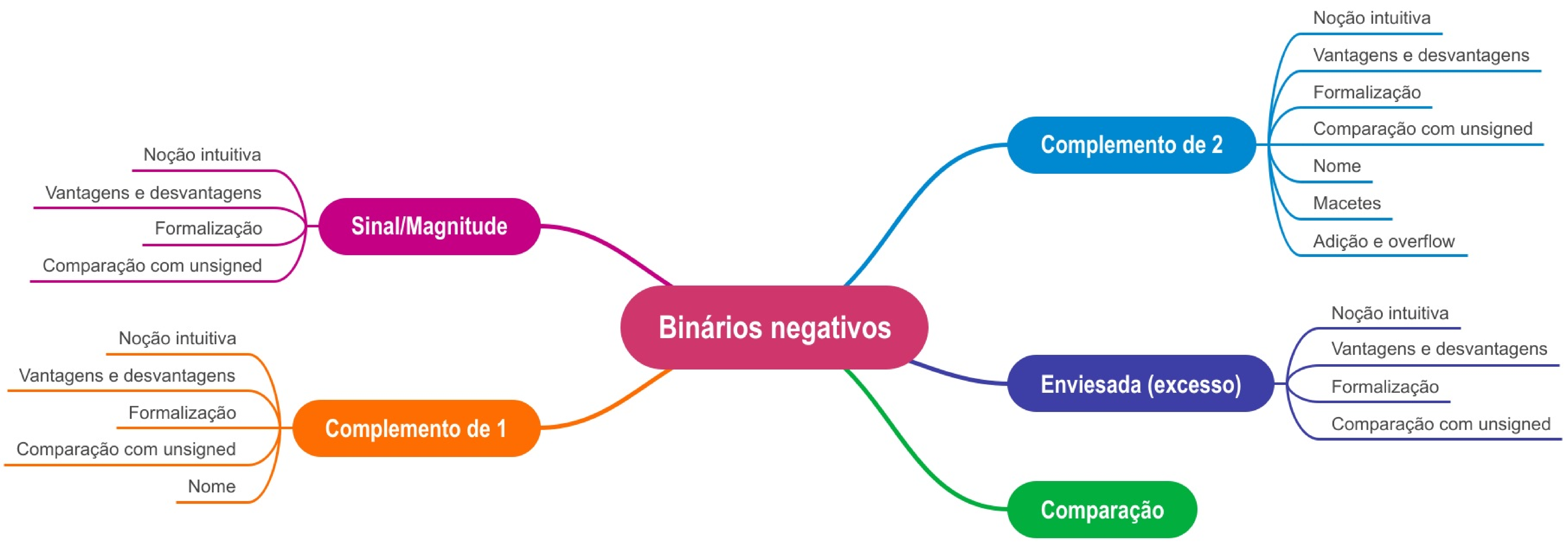
FUNDAMENTOS DA COMPUTAÇÃO

Representação de binários negativos



Sinal/Magnitude

Sinal/Magnitude

É a notação mais simples e intuitiva:

- MSB = sinal
  - 0 = positivo
  - 1 = negativo

- Outros n - 1 bits:
  interpretar como unsigned,
  representando o valor (magnitude)

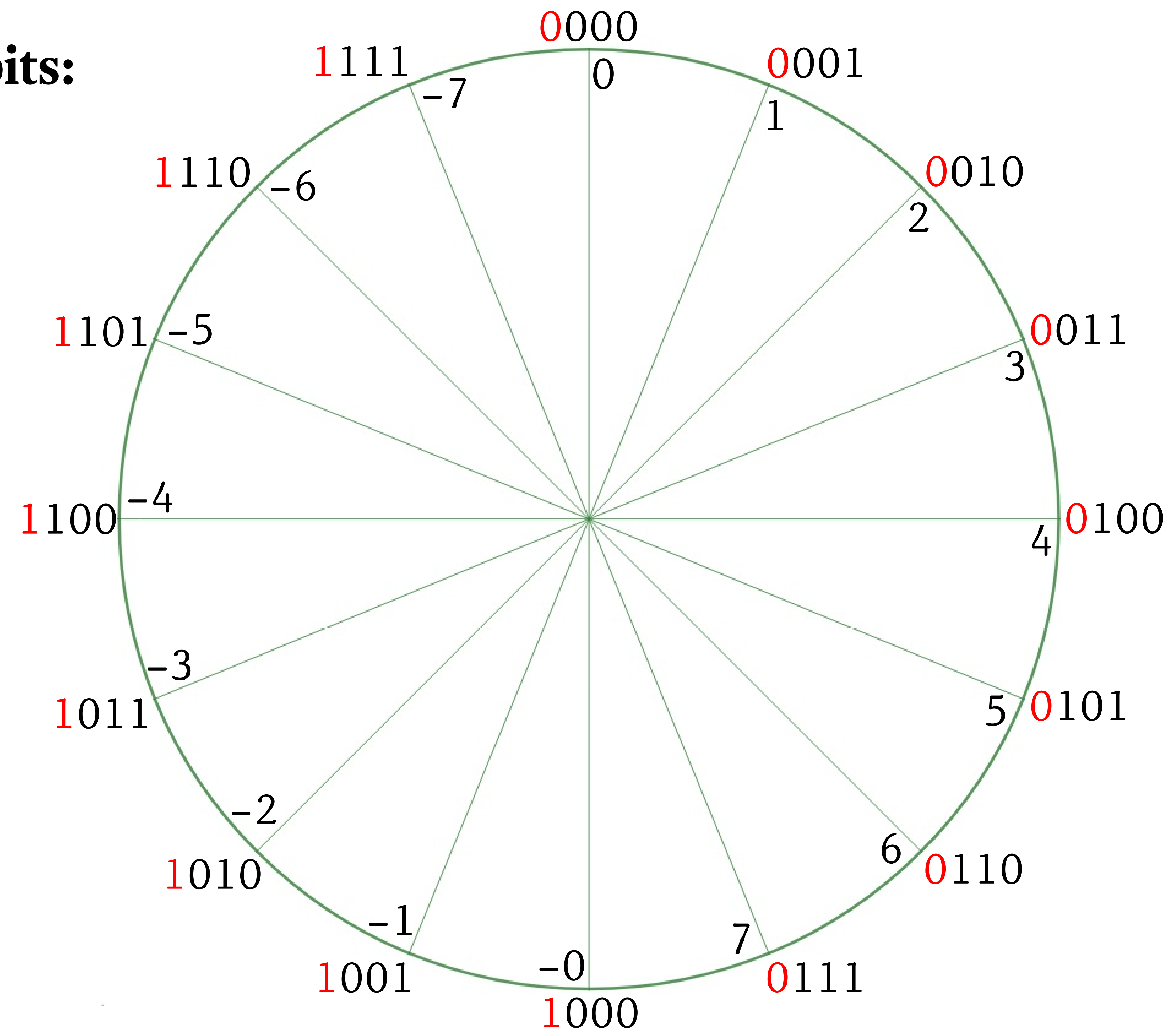
00011011

10011011

Sinal/Magnitude

Exemplo com números de 4 bits:

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



Sinal/Magnitude

Vantagens:

- simples e intuitiva
- semelhante ao modo como escrevemos
- "simétrica": mesma quantidade de números positivos e negativos

Desvantagens:

- duas representações para o zero: +0 e -0
- dificulta cálculos, por ex.: adição binária não funciona

$$\begin{array}{r} 1101 \\ +\ 0101 \\ \hline \end{array}$$

- dificulta comparação:

  +0 = -0 (?)

Sinal/Magnitude

Formalização:

Seja x um número inteiro binário formado por n algarismos, em notação **sinal/magnitude**. Representaremos o vetor de algarismos desse número por $\vec{x}$, ou então por $[x_{n-1}, x_{n-2}, \ldots, x_0]$ para denotar os algarismos individuais dentro do vetor. A posição que um determinado algarismo ocupa em $\vec{x}$ será denominada por i (contada da direita para esquerda, iniciando em 0), e a notação $\stackrel{\text{def}}{=\!=}$ significa "é definido por". Assim, a conversão $\mathrm{B_{s/m}}D$ (binário em sinal/magnitude para decimal) de um vetor binário $\vec{x}$ com tamanho n é dada por:

$$\mathrm{B_{s/m}D}(\vec{x}_n) \stackrel{\text{def}}{=\!=} (-1)^{x_{n-1}} \times \left(\sum_{i=0}^{n-2} x_i 2^i \right)$$

Total: 2^n (?)

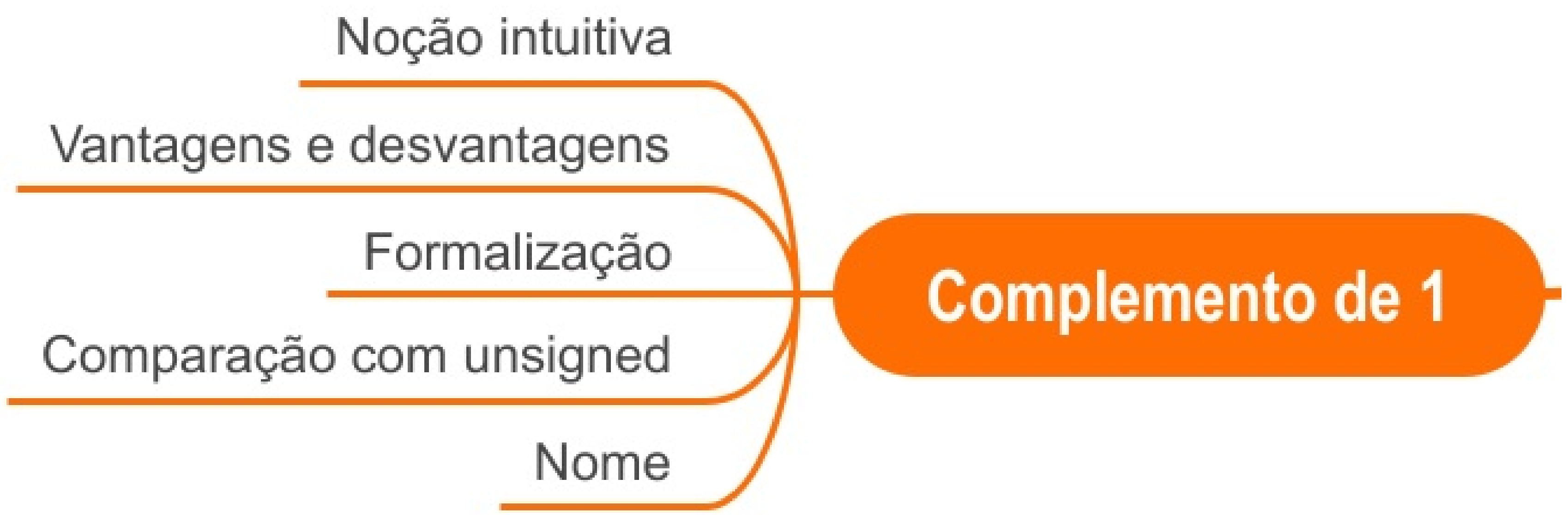
Faixa: $[-2^{n-1} + 1 \; ; \; 2^{n-1} - 1]$

Sinal/Magnitude

Comparação com unsigned, binários de 4 bits:

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

Complemento de 1



Complemento de 1

É uma notação mais complexa, que utiliza o complemento bit a bit para representar os números negativos e o sinal.

- MSB: sinal e peso ponderado
  - 0 = positivo
  - 1 = negativo

- Outros n - 1 bits:
  - interpretar como unsigned representando o valor (magnitude) a ser somado com o peso ponderado do sinal.

Complemento:

$0 \rightarrow 1$

$1 \rightarrow 0$

Complemento bit a bit (complemento de 1):

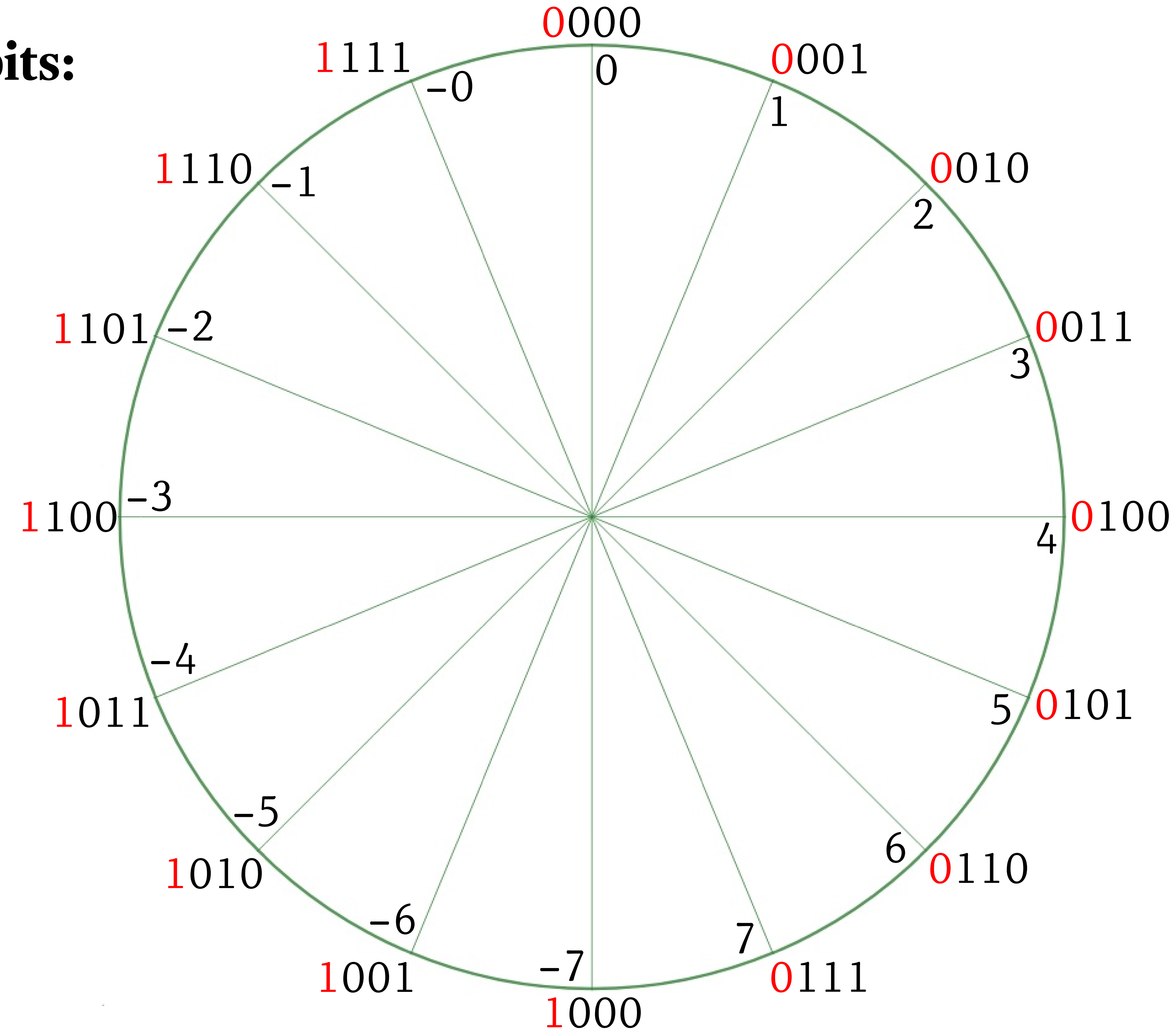
00110101 x $\vec{x}$

11001010 $\overline{x}$ $\overline{\vec{x}}$

Complemento de 1

Exemplo com números de 4 bits:

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



Complemento de 1

Vantagens:

- "simétrica": mesma quantidade de números positivos e negativos
- fácil de encontrar os valores pelo complemento

1101

Desvantagens:

- mais complexa que sinal/magnitude
- duas representações para o zero: +0 e -0
- dificulta cálculos, por ex.: adição binária não funciona

$$\begin{array}{r} 0100 \\ +\ 1101 \\ \hline \end{array}$$

- dificulta comparação:

  +0 = -0 (?)

Complemento de 1

Formalização:

Seja x um número inteiro binário formado por n algarismos, em notação **complemento de 1**. Representaremos o vetor de algarismos desse número por $\vec{x}$, ou então por $[x_{n-1}, x_{n-2}, \ldots, x_0]$ para denotar os algarismos individuais dentro do vetor. A posição que um determinado algarismo ocupa em $\vec{x}$ será denominada por i (contada da direita para esquerda, iniciando em 0), e a notação $\stackrel{\text{def}}{=}$ significa "é definido por". Assim, a conversão $\mathrm{B_{c1}}D$ (binário em complemento de 1 para decimal) de um vetor binário $\vec{x}$ com tamanho n é dada por:

$$\mathrm{B_{c1}D}(\vec{x}_n) \stackrel{\text{def}}{=} -(2^{n-1} - 1)x_{n-1} + \left(\sum_{i=0}^{n-2} x_i 2^i \right)$$

Total: 2^n **(?)**

Faixa: $[-2^{n-1} + 1 \; ; \; 2^{n-1} - 1]$

Complemento de 1

Comparação com unsigned, binários de 4 bits:

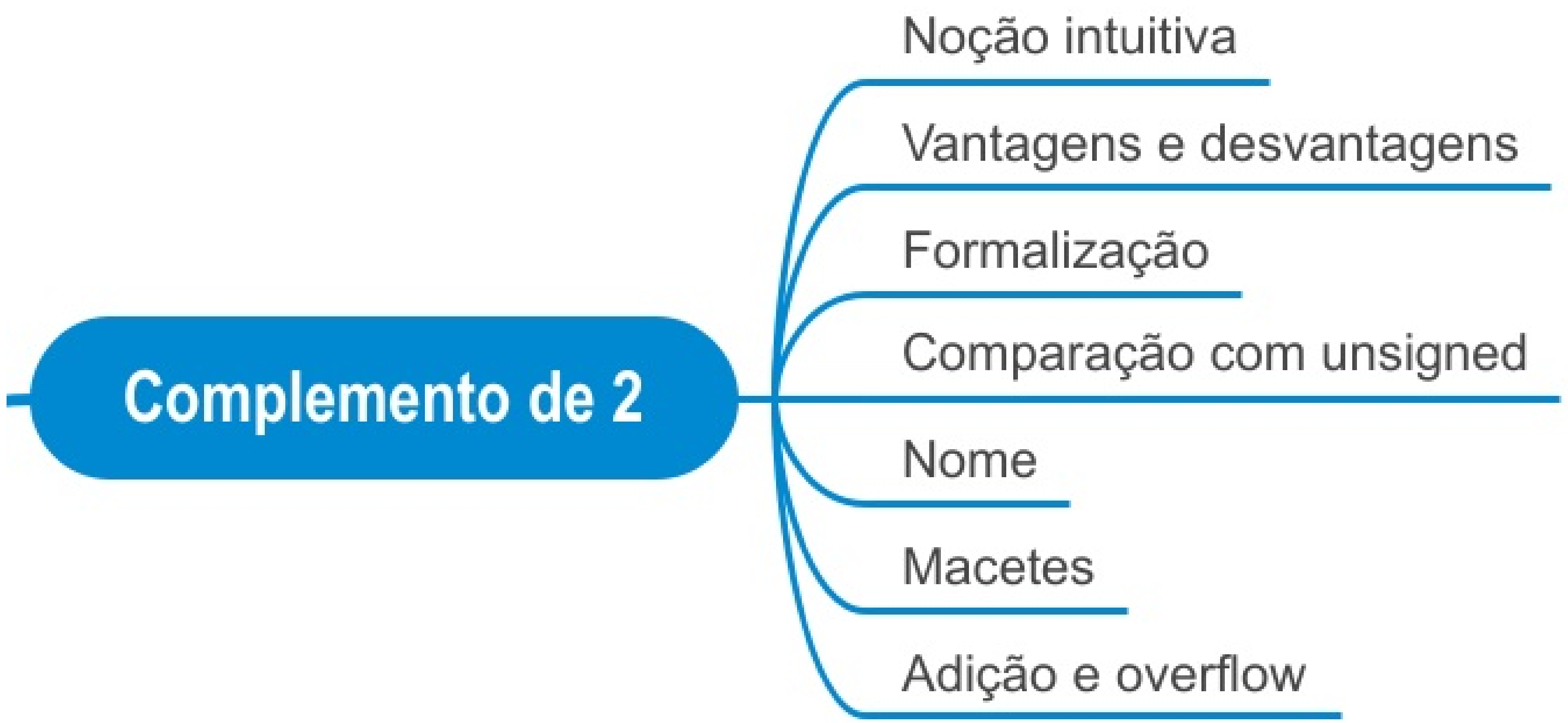
-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Decimal
								0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	Unsigned
	1000	1001	1010	1011	1100	1101	1110	0000 1111	0001	0010	0011	0100	0101	0110	0111									Complemento de 1

Complemento de 1

Por que esse nome? Seja x um número inteiro binário formado por n algarismos. O **complemento bit a bit** desse número, chamado de **complemento de 1**, será representado por $\overline{x}$. Considere que DB é uma função que converte um número decimal em um número binário com n algarismos. Assim, o complemento bit a bit de x será dado por:

$$\overline{x} = \mathrm{DB}(2^n - 1) - x$$

Complemento de 2



Complemento de 2

É uma notação mais complexa, onde o padrão de bits dos valores positivos e negativos de mesma magnitude são idênticos, quando lidos da direita para a esquerda, até (e incluindo) o primeiro algarismo 1, e, a partir daí, o padrão de bits é o complemento um do outro. O zero é um caso especial.

- MSB: sinal e peso ponderado
  - 0 = positivo
  - 1 = negativo

- Outros n - 1 bits:
  - interpretar como unsigned representando o valor (magnitude) a ser somado com o peso ponderado do sinal.

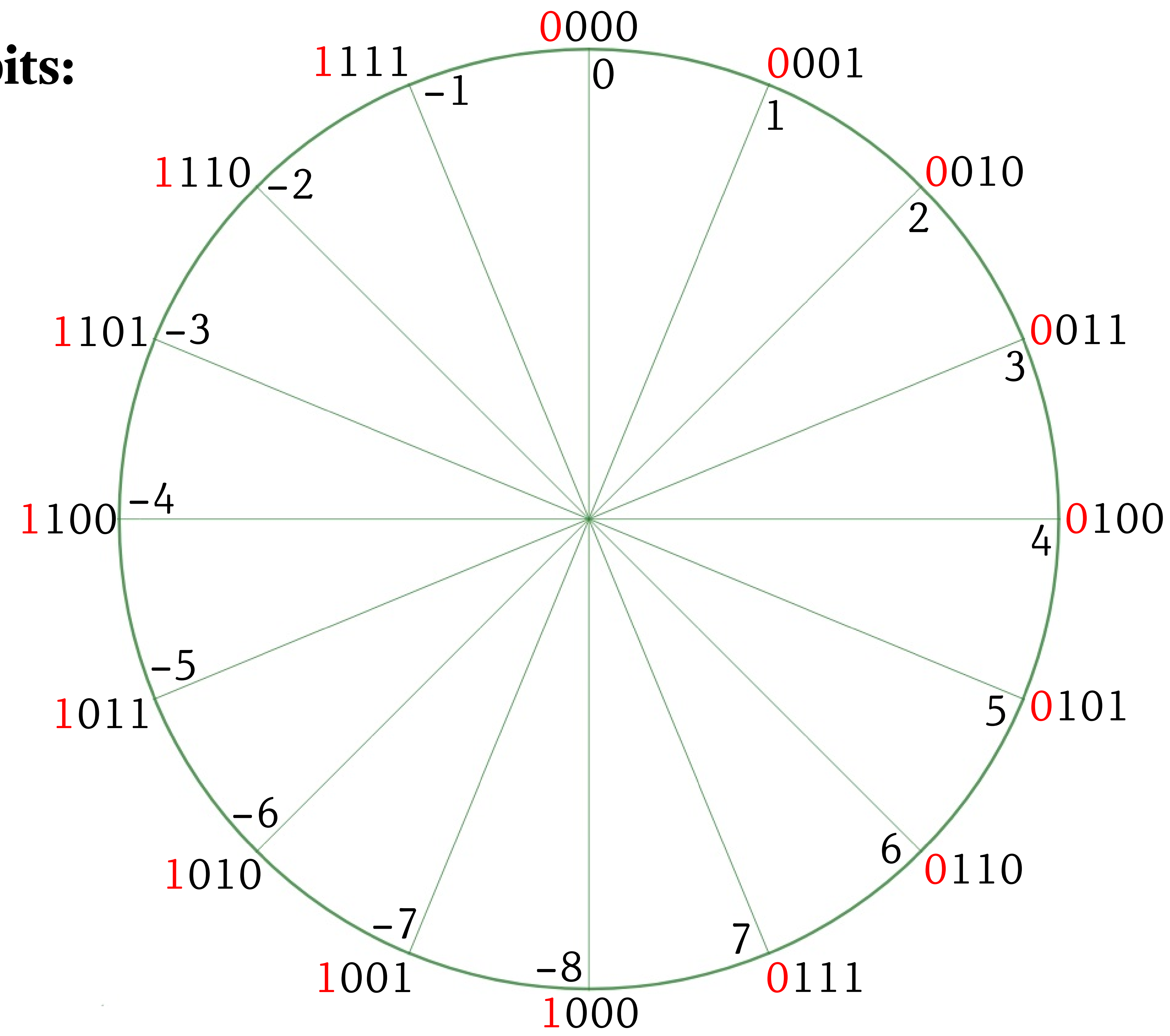
01001100
10110100

01111111
10000001

Complemento de 2

Exemplo com números de 4 bits:

Decimal	Complemento de 2
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
- 1	1111
- 2	1110
- 3	1101
- 4	1100
- 5	1011
- 6	1010
- 7	1001
- 8	1000



Complemento de 2

Vantagens:

- fácil de encontrar os valores: "copia e complementa" ou "reverter o sinal"
- apenas uma representação para o zero: 0
- facilita comparações (0 = 0)
- facilita cálculos, por ex.: adição binária funciona

$$\begin{array}{r} 0100 \\ +\ 1101 \\ \hline \end{array}$$

Desvantagens:

- mais complexa que complemento de 1
- "assimétrica": mais números negativos do que positivos (um a mais)

Complemento de 2

Formalização:

Seja x um número inteiro binário formado por n algarismos, em notação **complemento de 2**. Representaremos o vetor de algarismos desse número por $\vec{x}$, ou então por $[x_{n-1}, x_{n-2}, \ldots, x_0]$ para denotar os algarismos individuais dentro do vetor. A posição que um determinado algarismo ocupa em $\vec{x}$ será denominada por i (contada da direita para esquerda, iniciando em 0), e a notação $\stackrel{\text{def}}{=\!=}$ significa "é definido por". Assim, a conversão $\mathrm{B_{c2}}D$ (binário em complemento de 2 para decimal) de um vetor binário $\vec{x}$ com tamanho n é dada por:

$$\mathrm{B_{c2}D}(\vec{x}_n) \stackrel{\text{def}}{=\!=} -(2^{n-1})x_{n-1} + \left(\sum_{i=0}^{n-2} x_i 2^i\right)$$

Total: 2^n

Faixa: $[-2^{n-1}\ ;\ 2^{n-1} - 1]$

Complemento de 2

Comparação com unsigned, binários de 4 bits:

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

Complemento de 2

Por que esse nome?

Seja x um número inteiro binário formado por n algarismos. Considere que DB é uma função que converte um número decimal em um número binário com n algarismos. O **complemento de 2** desse número, x_{c2}, será dado por:

$$x_{c2} = \mathrm{DB}(2^n) - x$$

Complemento de 2

Existem duas formas "rápidas" para encontrar o complemento de 2 de um número:

- "copia e complementa"

01001100
10110100

- "reverter o sinal" (complementa e soma 1)

$$
\begin{aligned}
x_{c2} &= \mathrm{DB}(2^n) - x \\
&= \overline{x} + 1
\end{aligned}
$$

Complemento de 2

A adição é feita normalmente:

4 + 3

A subtração é feita através da adição com o complemento:

4 - 3 → 4 + (-3)

Overflow:

- adição de positivo com negativo nunca causa overflow.

- adição de dois positivos ou dois negativos pode causar overflow, que é indicado se o sinal do resultado estiver trocado.

4 + 3

$$\begin{array}{r} 0100 \\ +\ 0011 \\ \hline \end{array}$$

4 - 3

$$\begin{array}{r} 0100 \\ +\ 1101 \\ \hline \end{array}$$

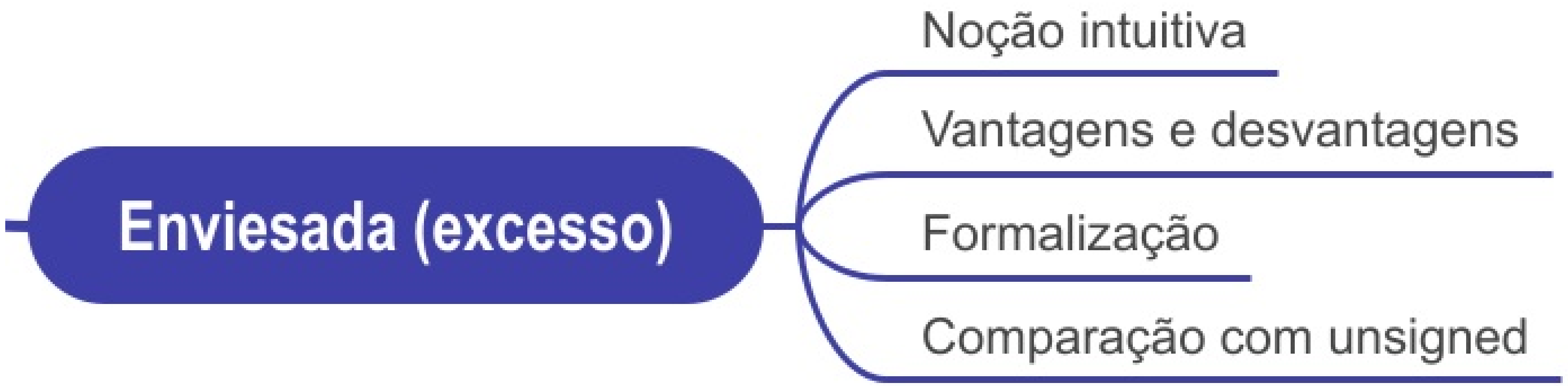
4 + 5

$$\begin{array}{r} 0100 \\ +\ 0101 \\ \hline \end{array}$$

(-4) + (-6)

$$\begin{array}{r} 1100 \\ +\ 1010 \\ \hline \end{array}$$

Enviesada (excesso)



Enviesada (excesso)

É uma notação simples com as seguintes características:

- o número mais negativo só tem 0s: 000...000
- o número mais positivo só tem 1s: 111...111
- o zero, em geral, inicia com 1 e o resto são 0s: 100...000

- MSB: sinal
  - 0 = negativo
  - 1 = positivo

- Interpretação:
  - a interpretação não é direta, depende do viés (excesso) que está sendo considerado. Calcule o valor binário como se o número fosse unsigned (todos os bits), e depois subtraia o viés.

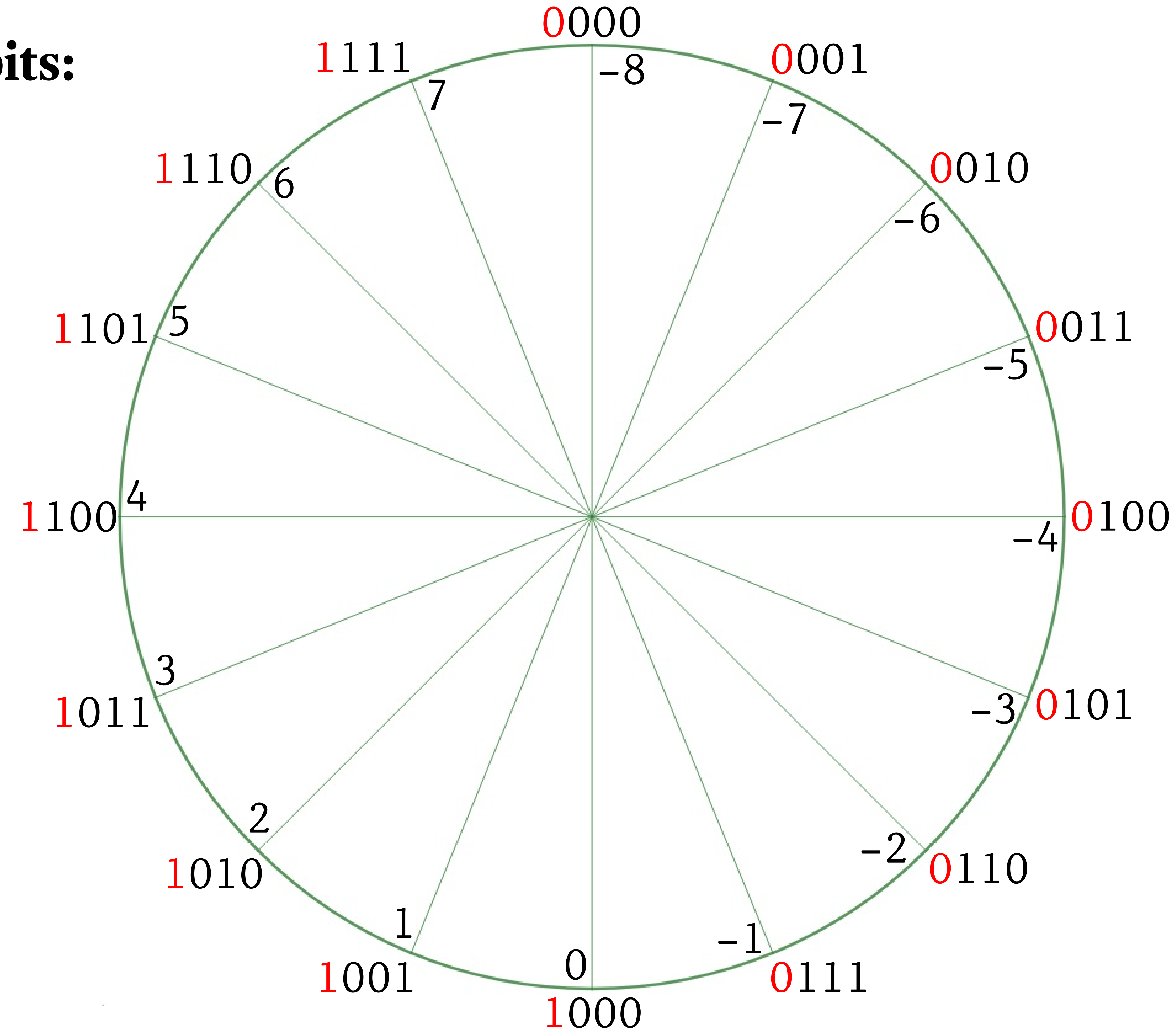
- O viés, em geral, depende da quantidade n de bits: $\text{Viés} = 2^{n-1}$

Enviesada (excesso)

Exemplo com números de 4 bits:

Decimal	Enviesada
- 8	0000
- 7	0001
- 6	0010
- 5	0011
- 4	0100
- 3	0101
- 2	0110
- 1	0111
0	1000
1	1001
2	1010
3	1011
4	1100
5	1101
6	1110
7	1111

Viés (excesso): 8



Enviesada (excesso)

Vantagens:

- mesmo com o viés, é uma notação relativamente simples
- apenas uma representação para o zero: 0
- facilita comparações (0 = 0)

Desvantagens:

- mais complexa que sinal magnitude
- "assimétrica": mais números negativos do que positivos (um a mais)
- dificulta cálculos, por ex.: adição binária não funciona

-4 + 5 Viés: 8

$$\begin{array}{r} 0100 \\ +\ 1101 \\ \hline \end{array}$$

Enviesada (excesso)

Formalização:

Seja x um número inteiro binário formado por n algarismos, em notação **enviesada** com viés 2^{n-1}. Representaremos o vetor de algarismos desse número por $\vec{x}$, ou então por $[x_{n-1}, x_{n-2}, \ldots, x_0]$ para denotar os algarismos individuais dentro do vetor. A posição que um determinado algarismo ocupa em $\vec{x}$ será denominada por i (contada da direita para esquerda, iniciando em 0), e a notação $\stackrel{\text{def}}{=\!=}$ significa "é definido por". Assim, a conversão $\mathrm{B_eD}$ (binário em notação enviesada para decimal) de um vetor binário $\vec{x}$ com tamanho n e viés dado por 2^{n-1} é dada por:

$$\mathrm{B_eD}(\vec{x}_n) \stackrel{\text{def}}{=\!=} \left(\sum_{i=0}^{n-1} x_i 2^i \right) - 2^{n-1}$$

Total: 2^n

Faixa: $[-2^{n-1} \; ; \; 2^{n-1} - 1]$

Enviesada (excesso)

Comparação com unsigned, binários de 4 bits com viés de 8:

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

Viés: 8

Comparação entre as notações

Comparação entre as notações

Comparação com binários de 4 bits:

-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Decimal
								0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	Unsigned
	1111	1110	1101	1100	1011	1010	1001	0000 1000	0001	0010	0011	0100	0101	0110	0111									Sinal/Magnitude
	1000	1001	1010	1011	1100	1101	1110	0000 1111	0001	0010	0011	0100	0101	0110	0111									Complemento de 1
1000	1001	1010	1011	1100	1101	1110	1111	0000	0001	0010	0011	0100	0101	0110	0111									Complemento de 2
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111									Enviesada

Viés: 8

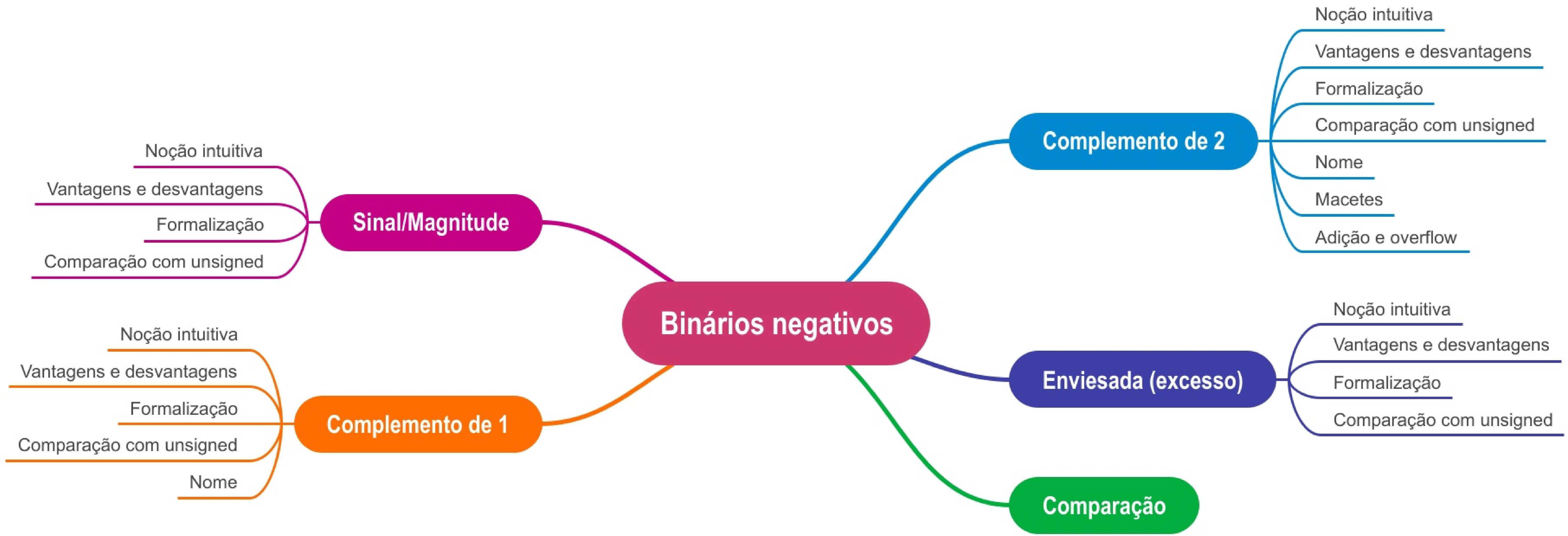
Comparação entre as notações

Comparação com unsigned, binários de 4 bits:

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

Viés: 8

Em resumo


Binários negativos
Sinal/Magnitude
Noção intuitiva
Vantagens e desvantagens
Formalização
Comparação com unsigned
Complemento de 1
Noção intuitiva
Vantagens e desvantagens
Formalização
Comparação com unsigned
Nome
Complemento de 2
Noção intuitiva
Vantagens e desvantagens
Formalização
Comparação com unsigned
Nome
Macetes
Adição e overflow
Enviesada (excesso)
Noção intuitiva
Vantagens e desvantagens
Formalização
Comparação com unsigned
Comparação