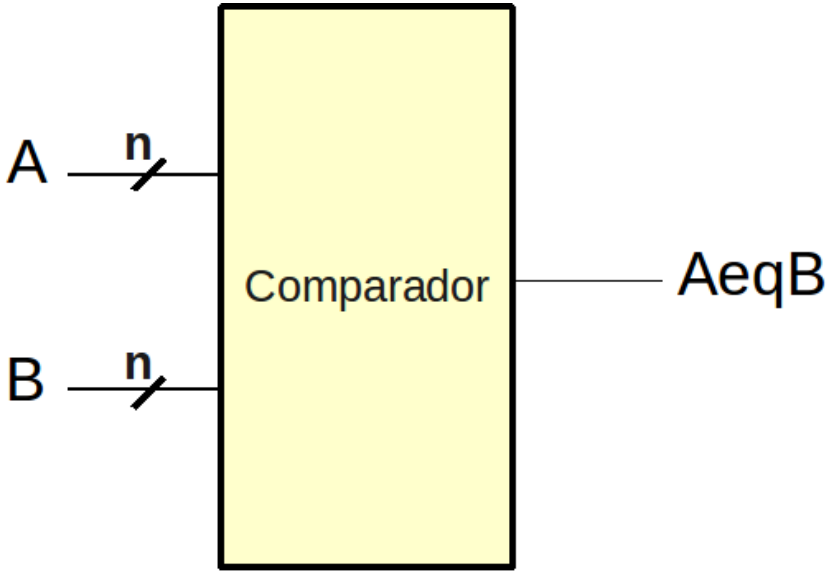
Circuitos Combinacionais

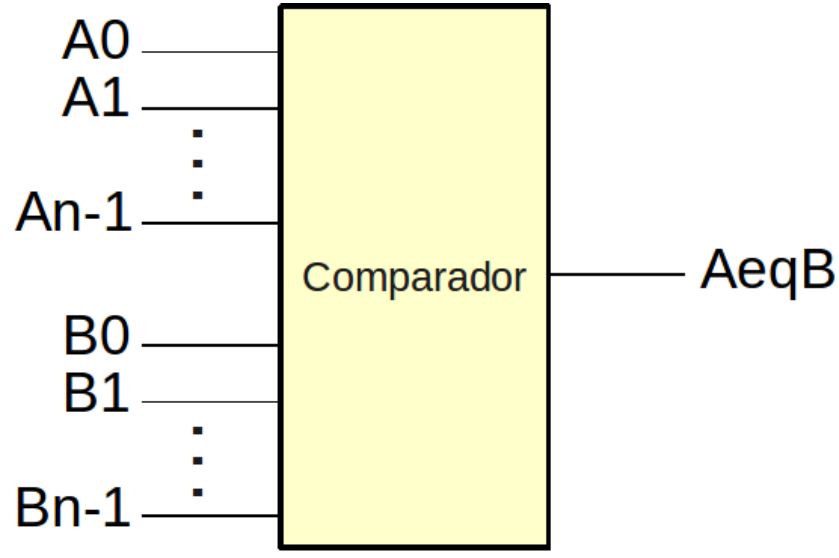
- **Circuito combinacional**:
  - Possui portas lógicas conectadas para produzir valor dos sinais de saída
  - **Não** possui armazenamento de valores no circuito
  - Valor dos sinais de saída depende **apenas** dos valores dos sinais de entrada

- **Circuitos Combinacionais Básicos**
  - Habilitação / Desabilitação
  - Multiplexador / Demultiplexador
  - Codificador / Decodificador
  - Gerador de paridade / Verificador de paridade
  - Circuitos aritméticos:
    - Shifter (deslocador)
    - Comparador
    - Somador / subtrator

Comparador de Igualdade de Dados de n bits

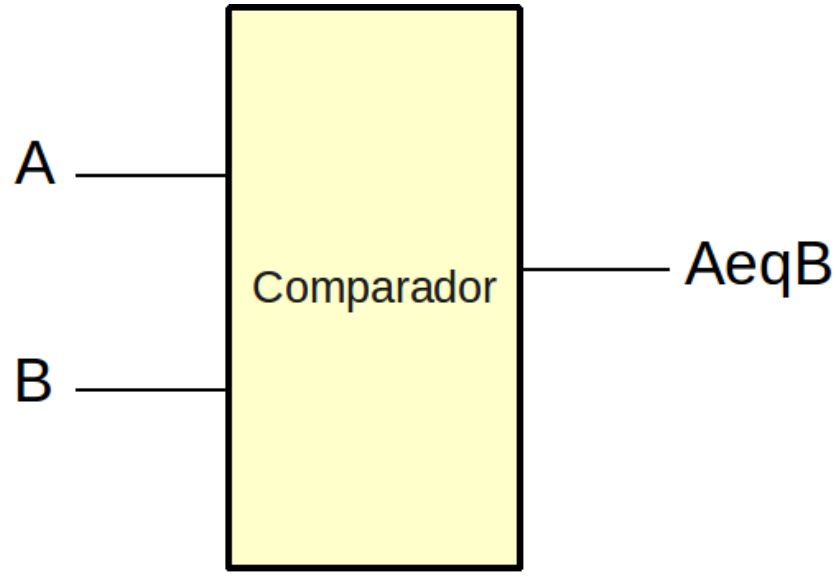
- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinal de saída**:
  - AeqB: indica se A = B (AeqB = 1) ou não (AeqB = 0)

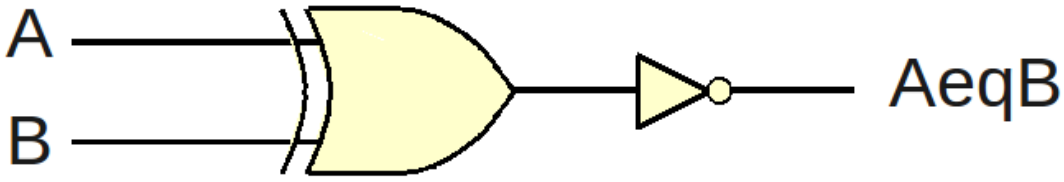




Comparador de Igualdade de Dados de 1 bit

- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
- **Sinal de saída**: AeqB



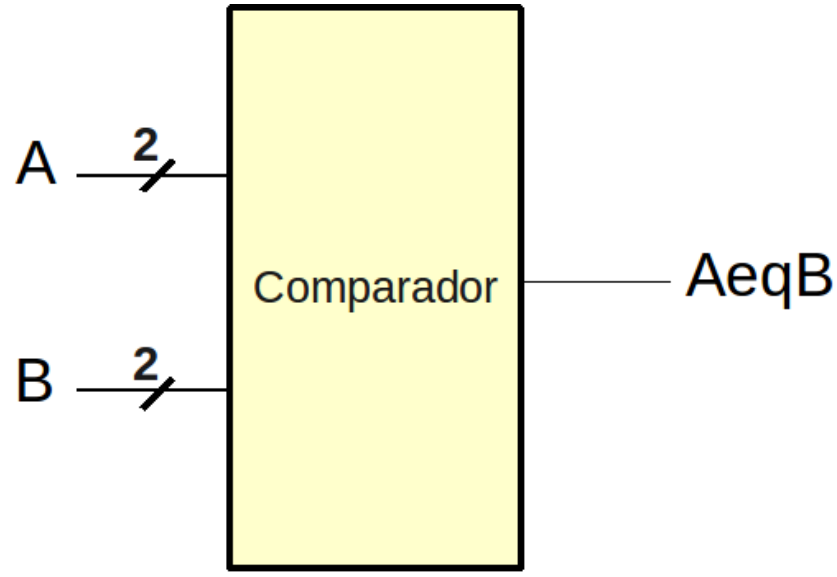


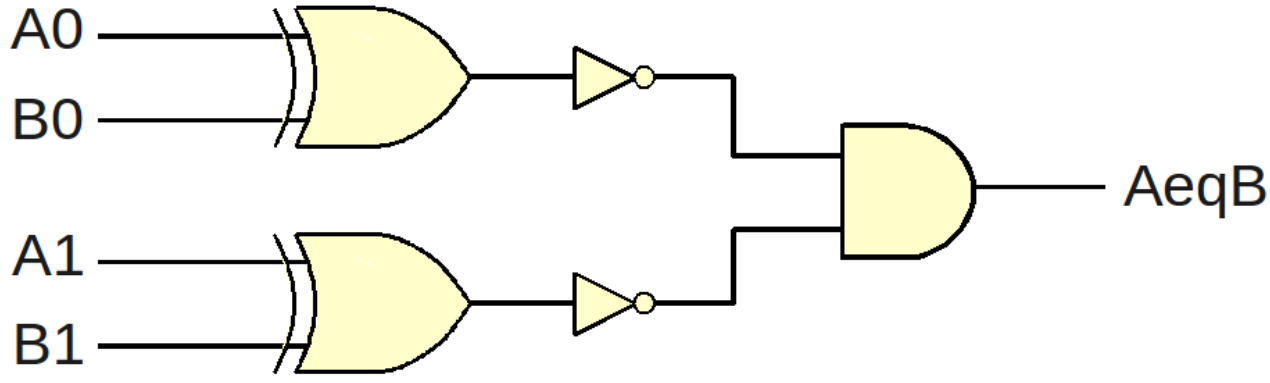
Porta XOR

Entradas		Saída
A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Comparador de Igualdade de Dados de 2 bits

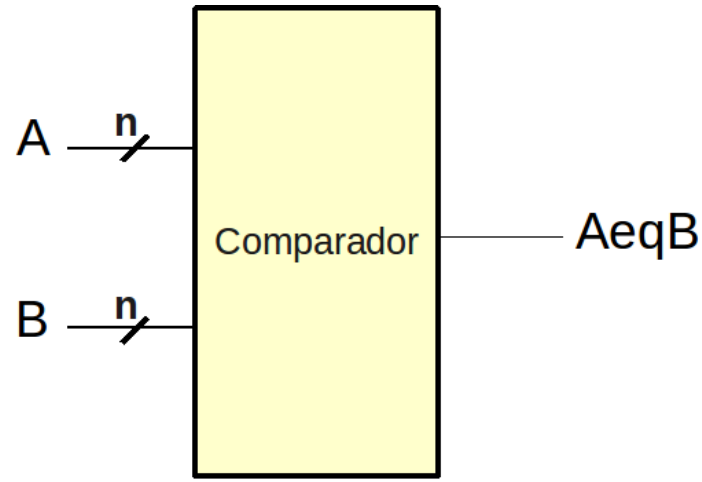
- **Sinais de entrada**:
  - Dado A de 2 bits: A_1 A_0
  - Dado B de 2 bits: B_1 B_0
- **Sinal de saída**: AeqB

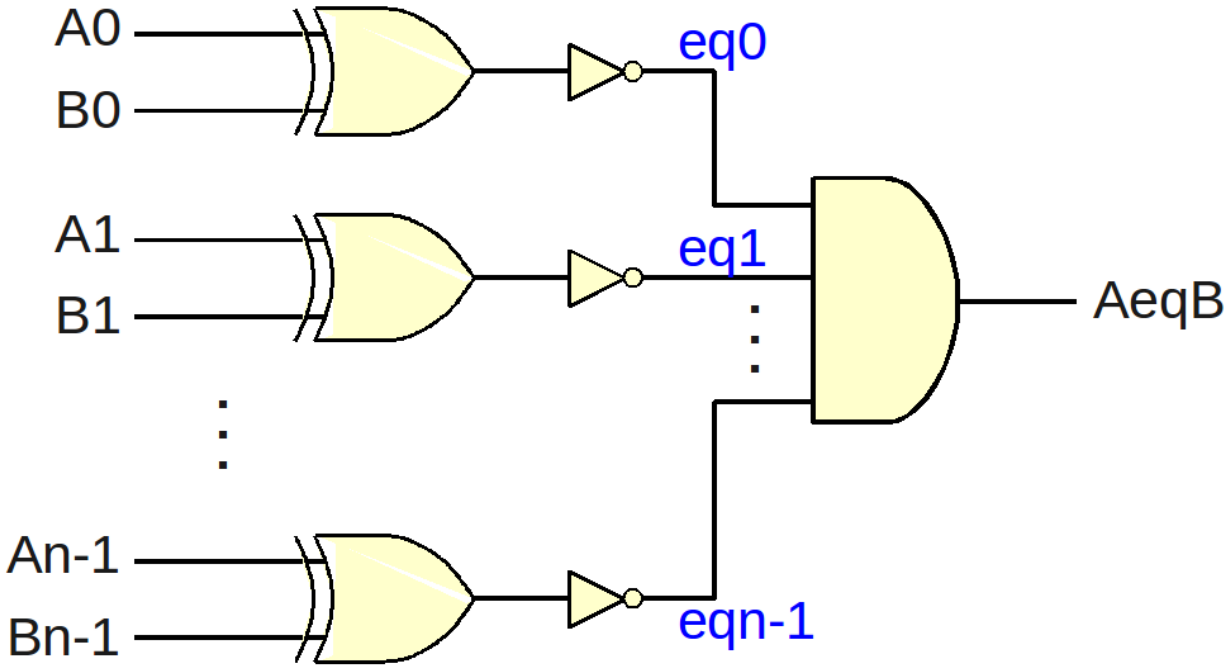




Comparador de Igualdade de Dados de n bits

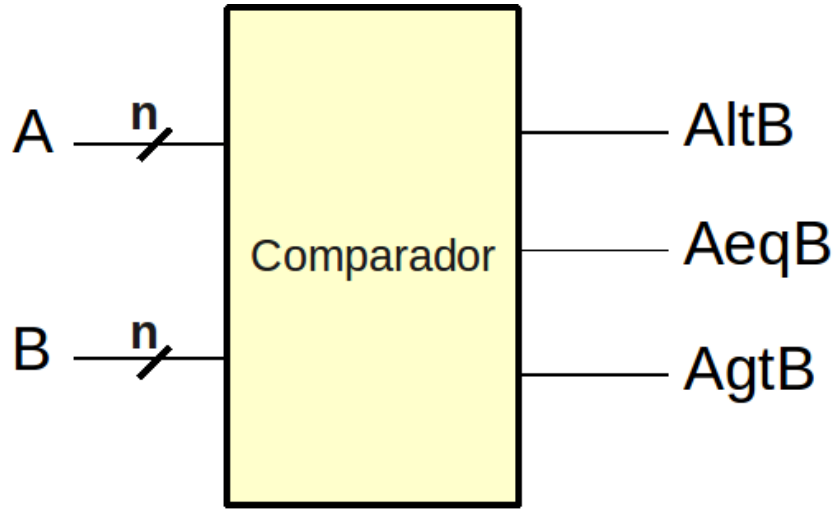
- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinal de saída**: AeqB





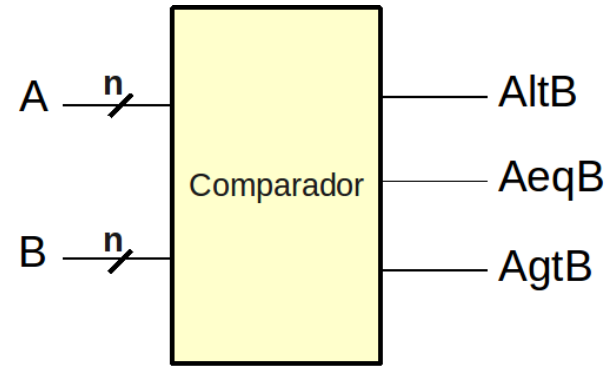
Comparador de Magnitude de Dados de n bits

- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinais de saída**:
  - AltB: indica se A < B (AltB = 1) ou não (AltB = 0)
  - AeqB: indica se A = B (AeqB = 1) ou não (AeqB = 0)
  - AgtB: indica se A > B (AgtB = 1) ou não (AgtB = 0)



Comparador de Magnitude de Dados de n bits

- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinais de saída**: AltB, AeqB, AgtB

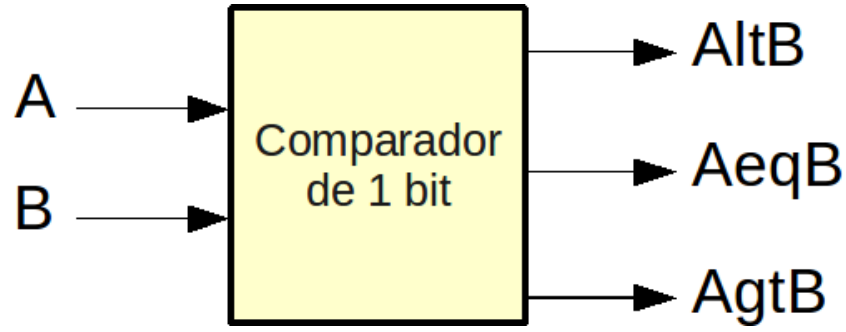


- **Ideia**:
  - Comparar a partir do bit mais significativo para menos significativo

A_{n-1}	B_{n-1}	Significado
0	0	$A_{n-1} = B_{n-1} \Rightarrow$ Compara bit seguinte $(n-2)$
0	1	A < B
1	0	A > B
1	1	$A_{n-1} = B_{n-1} \Rightarrow$ Compara bit seguinte $(n-2)$

Comparador de Magnitude de Dados de 1 bit

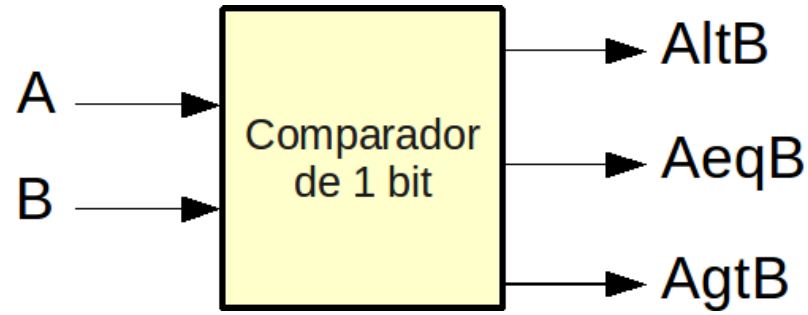
- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
- **Sinais de saída**:
  - AltB: indica se A < B ou não
  - AeqB: indica se A = B ou não
  - AgtB: indica se A > B ou não

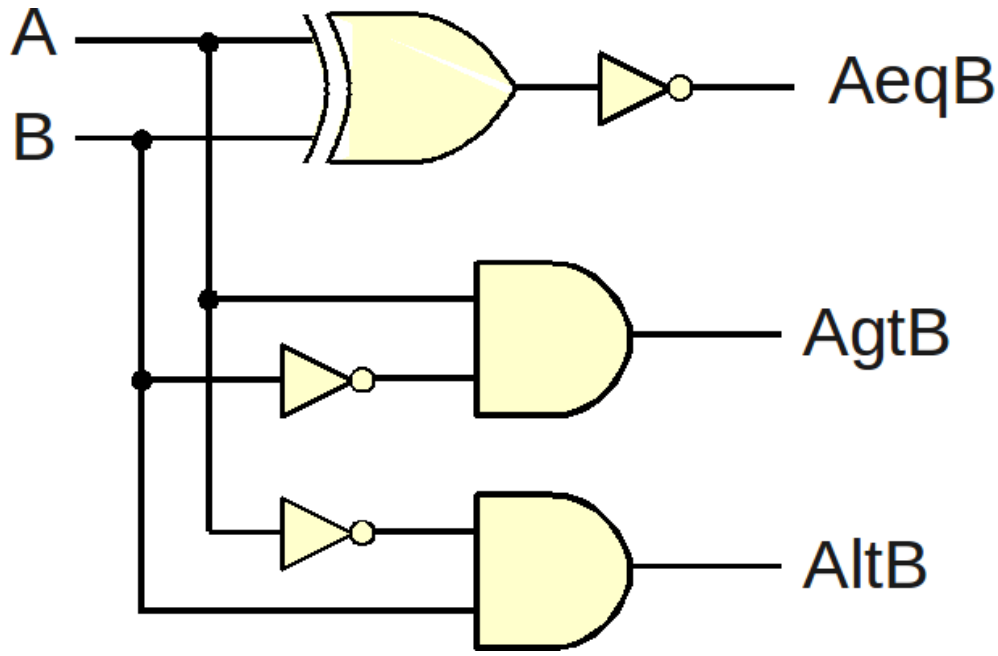


Entradas		Saídas		
A	**B**	**AgtB**	**AeqB**	**AltB**
0	0	0	1	0
0	1	0	0	1
1	0	1	0	0
1	1	0	1	0

Comparador de Magnitude de Dados de 1 bit

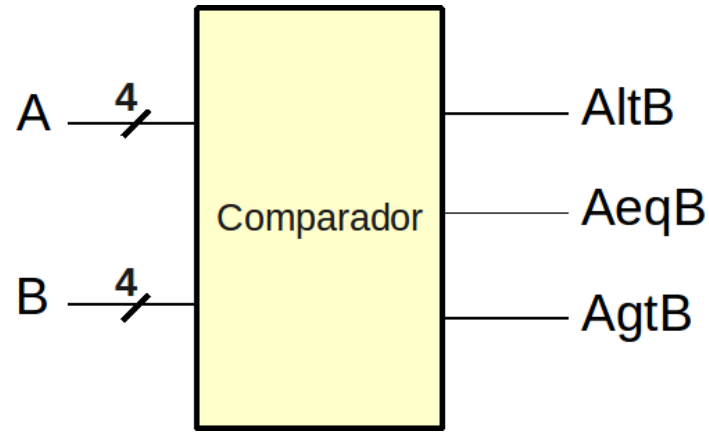
- $AeqB = \overline{A \oplus B}$
- $AgtB = A \bullet \overline{B}$
- $AltB = \overline{A} \bullet B$





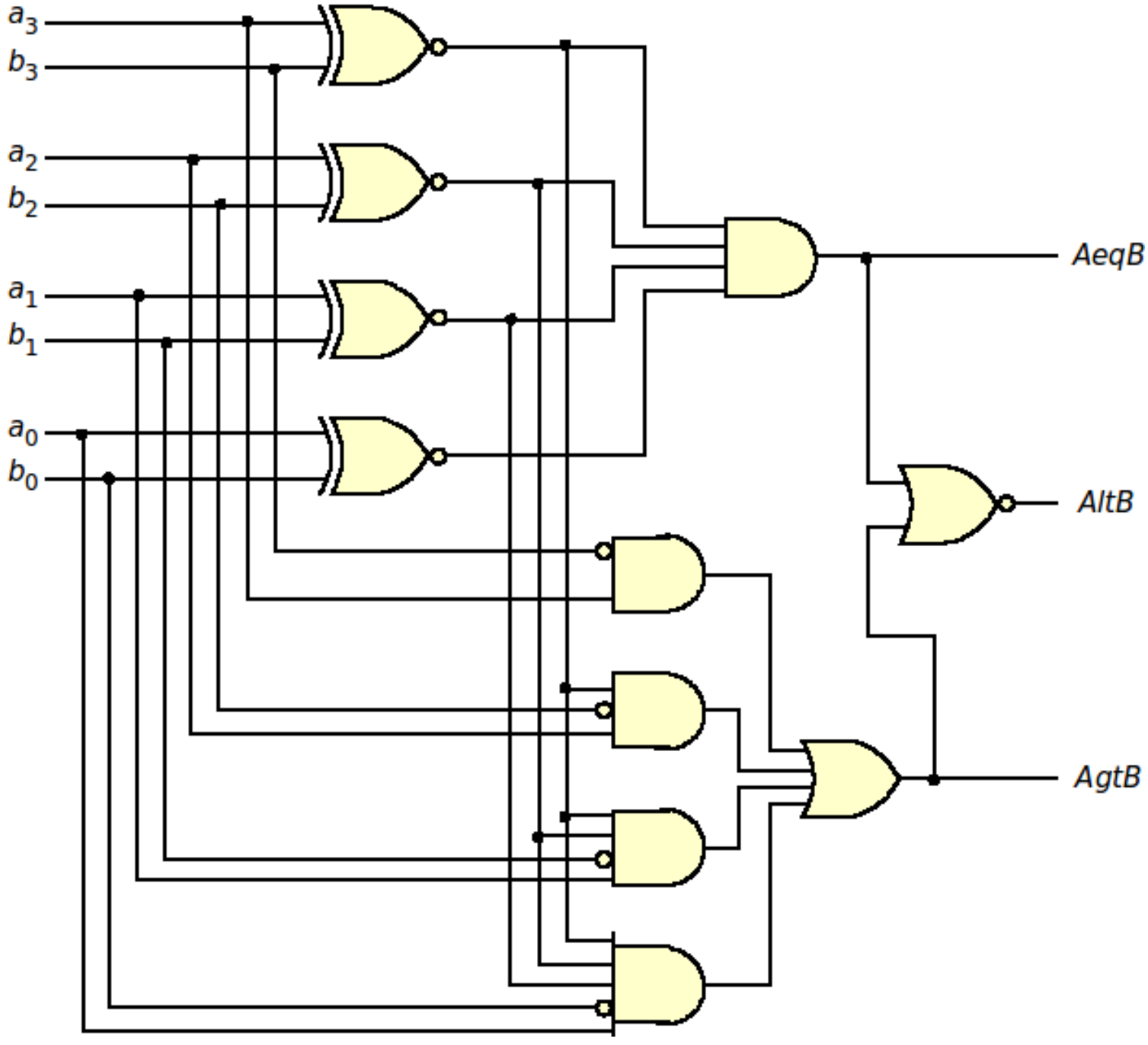
Comparador de Magnitude de Dados de 4 bits

- **Sinais de entrada**:
  - Dado A de 4 bits: $A_3 A_2 A_1 A_0$
  - Dado B de 4 bits: $B_3 B_2 B_1 B_0$
- **Sinais de saída**: $AltB$, $AeqB$, $AgtB$



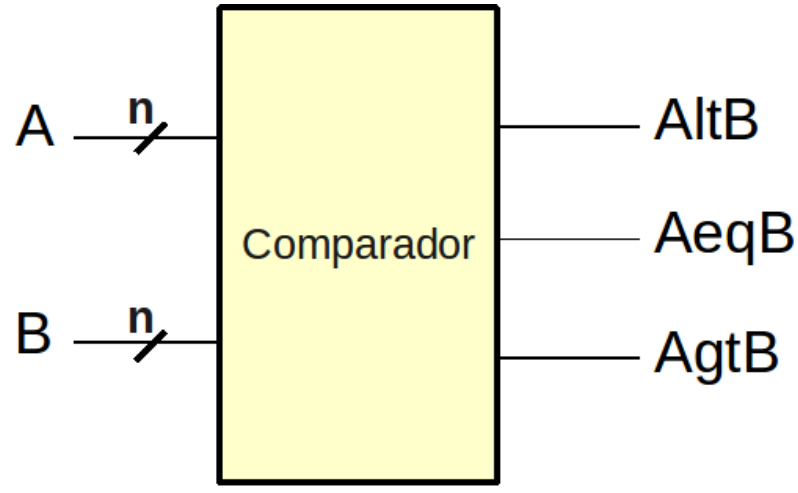
- **Ideia**:
  - $eq_i = \overline{A_i \oplus B_i}$
  - $AeqB = eq_3 \bullet eq_2 \bullet eq_1 \bullet eq_0$
  - $AgtB = A_3 \bullet \overline{B_3} \;+\; eq_3 \bullet A_2 \bullet \overline{B_2} \;+\; eq_3 \bullet eq_2 \bullet A_1 \bullet \overline{B_1} \;+\; eq_3 \bullet eq_2 \bullet eq_1 \bullet A_0 \bullet \overline{B_0}$
  - $AltB = \overline{AeqB + AgtB}$

Comparador de Magnitude de Dados de 4 bits



Comparador de Magnitude de Dados de n bits

- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinais de saída**: AltB, AeqB, AgtB
- **Ideias**:
  - Comparar a partir do bit mais significativo para menos significativo
  - Usar técnica de **replicação**

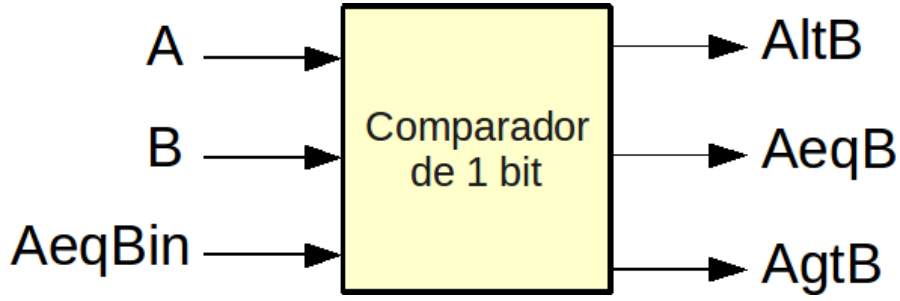


Comparador de Magnitude de Dados de 1 bit

- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
  - **AeqBin**: ⇐ **Modificação**
    - Se AeqBin = 0, já determinou que A > B ou A < B
    - Se AeqBin = 1, ainda não determinou que A > B ou A < B
- **Sinais de saída**:
  - AltB: indica se A < B ou não
  - AeqB: indica se A = B ou não
  - AgtB: indica se A > B ou não

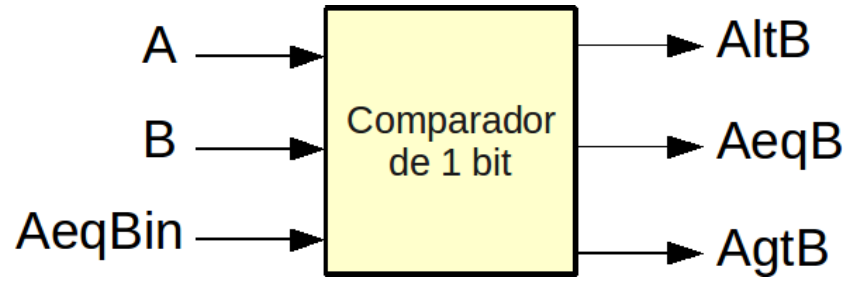
Apenas se AeqBin for 1.

Se AeqBin for 0, saídas ficam em 0.



Comparador de Magnitude de Dados de 1 bit

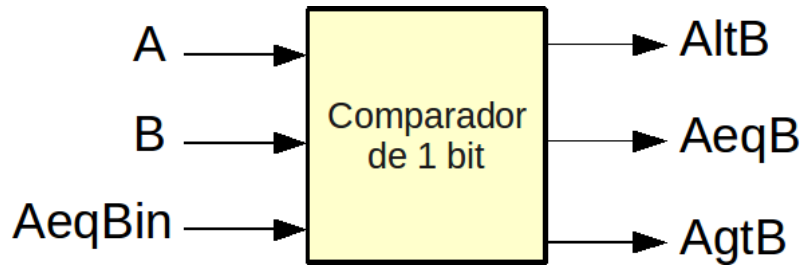
- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
  - AeqBin
- **Sinais de saída**: AltB, AeqB, AgtB

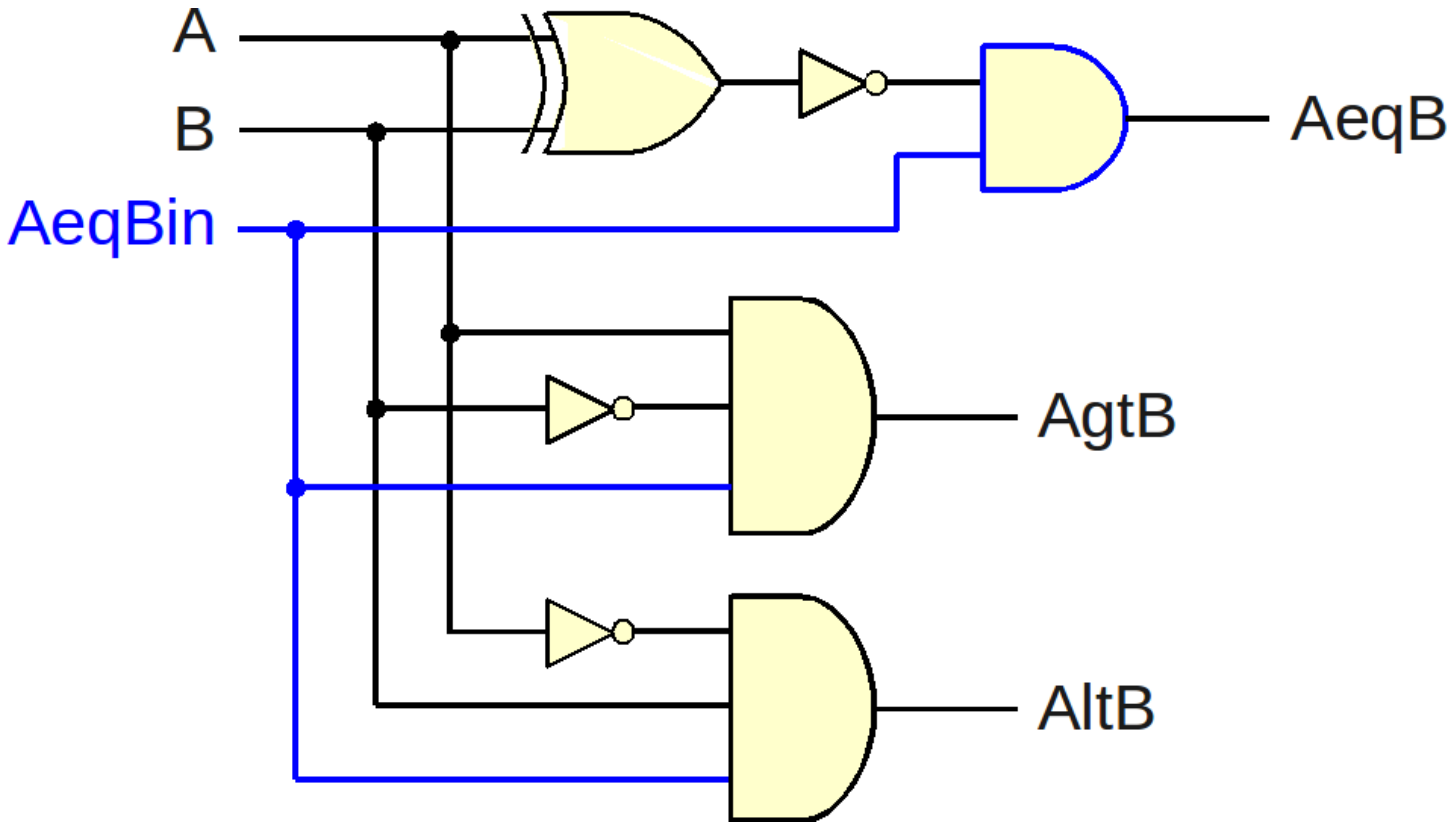


Entradas			Saídas		
AeqBin	**A**	**B**	**AgtB**	**AeqB**	**AltB**
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0

Comparador de Magnitude de Dados de 1 bit

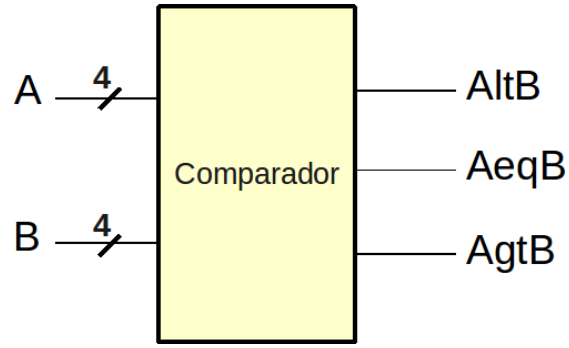
- $AeqB = (\overline{A \oplus B}) \bullet AeqBin$
- $AgtB = (A \bullet \overline{B}) \bullet AeqBin$
- $AltB = (\overline{A} \bullet B) \bullet AeqBin$

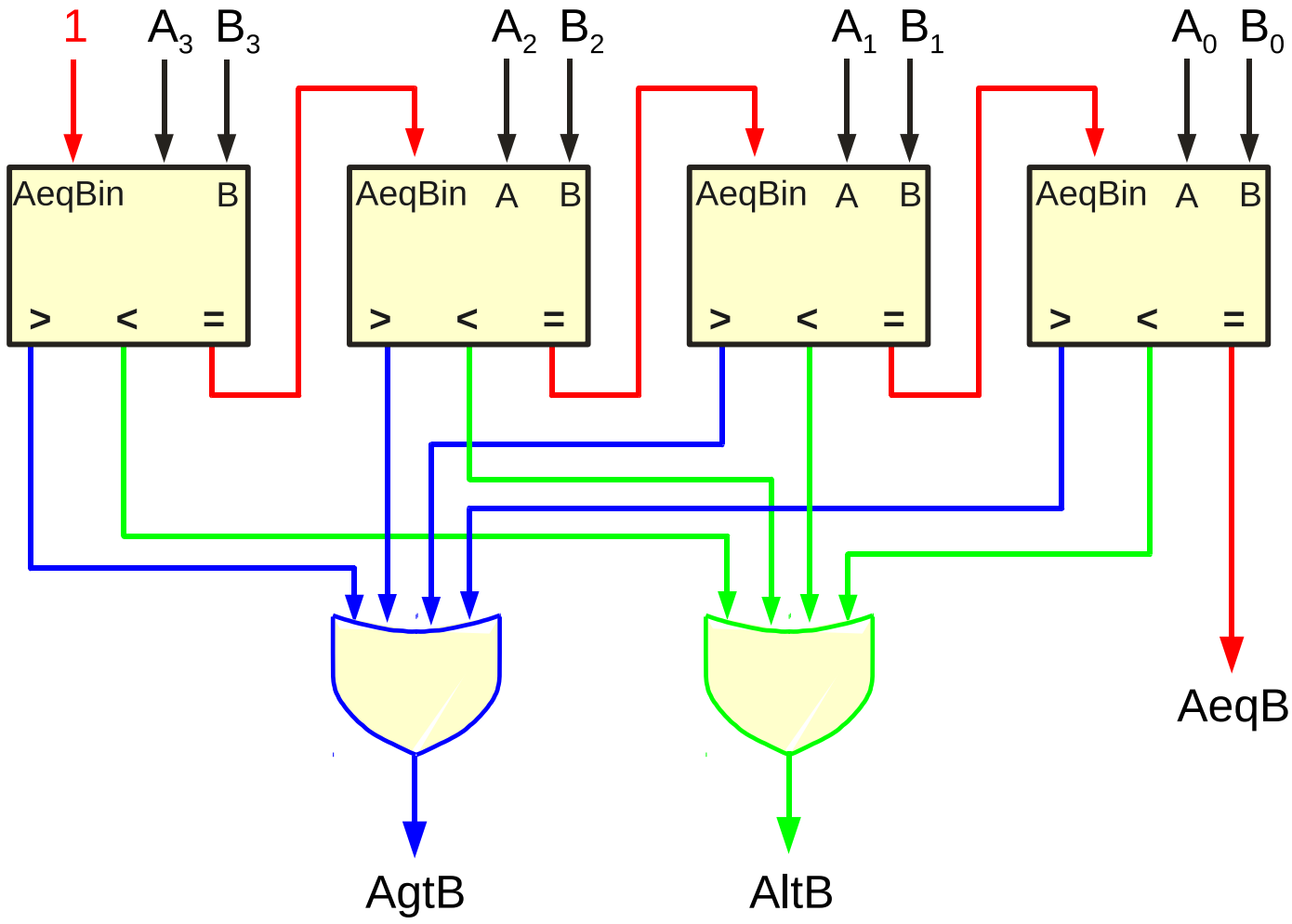




Comparador de Magnitude de Dados de 4 bits

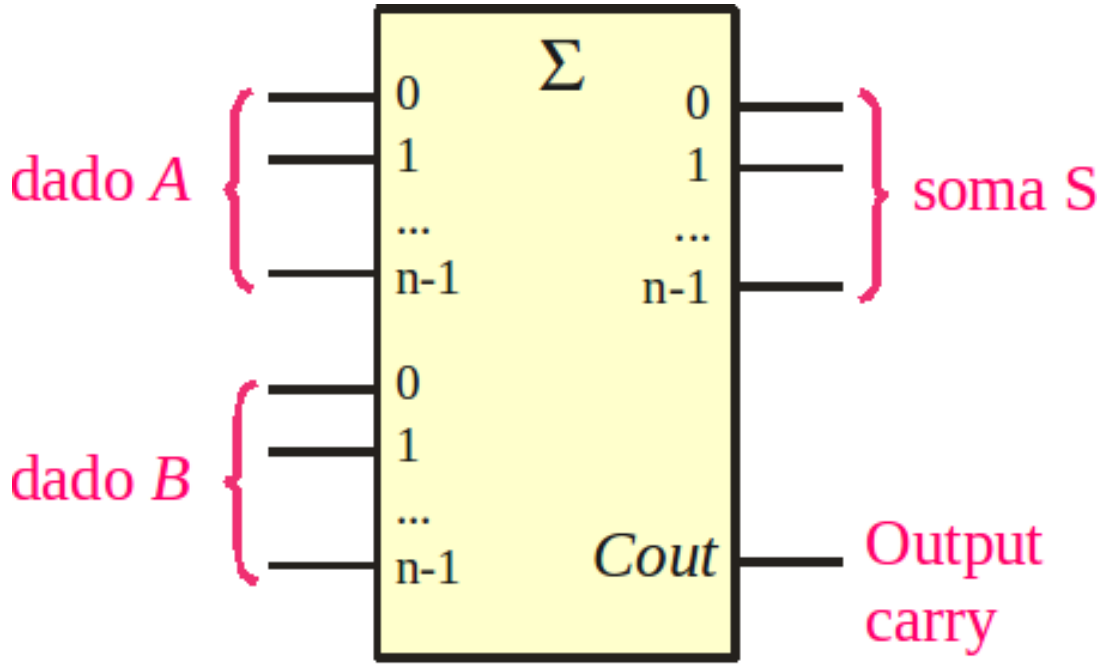
- **Sinais de entrada**: Dados A e B de 4 bits
- **Sinais de saída**: AltB, AeqB, AgtB
- **Usando 4 comparadores de 1 bit**





Somador de Dados de n bits

- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
- **Sinais de saída**:
  - Soma S de n bits: S_{n-1} ... S_2 S_1 S_0
  - CarryOut

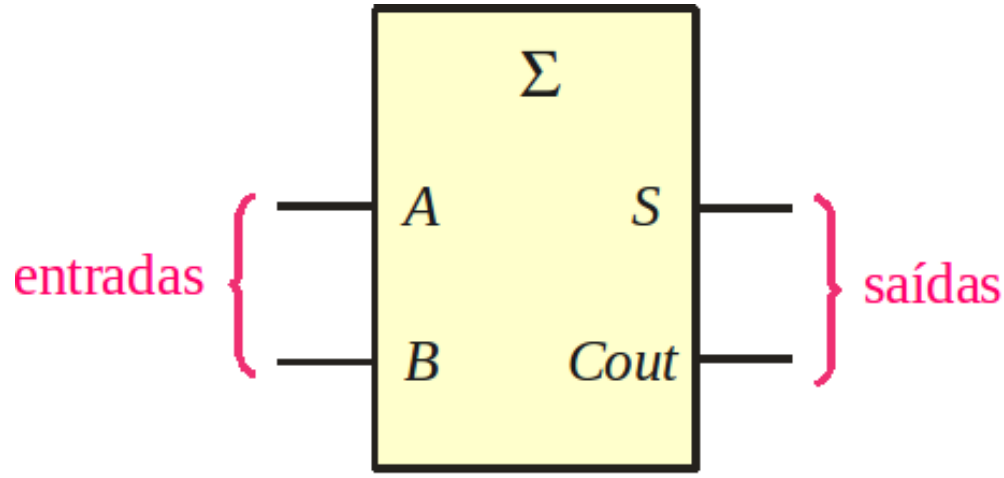


Somador de Dados de n bits

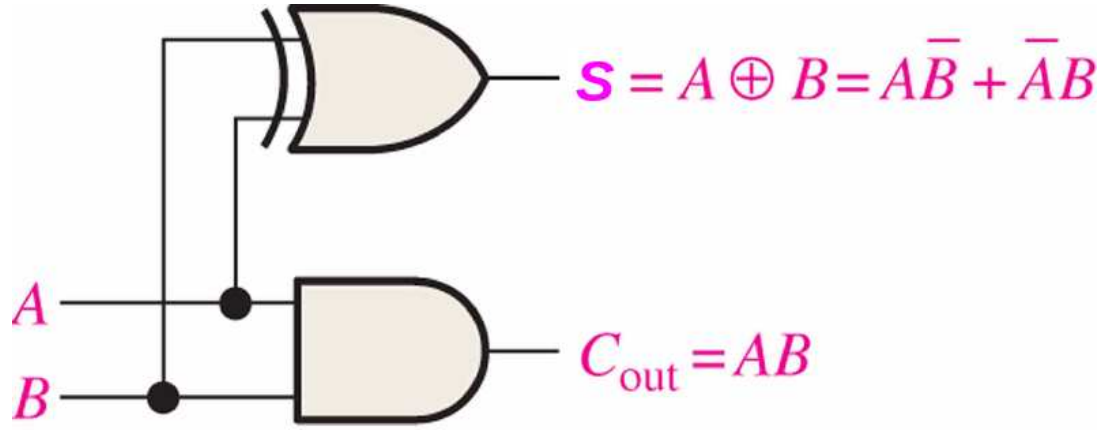
- Como construir?
- **Ideia**:
  - Construir circuito somador de dados de 1 bit
  - Construir somador de n bits usando n somadores de 1 bit
- **Técnica**: **Replicação** ou **bit-slice**
- **Somadores simples de 1 bit**:
  - Meio somador (**half-adder**)
  - Somador completo (**full-adder**)

Meio Somador (Half-Adder)

- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
- **Sinais de saída**:
  - Soma S de 1 bit
  - CarryOut



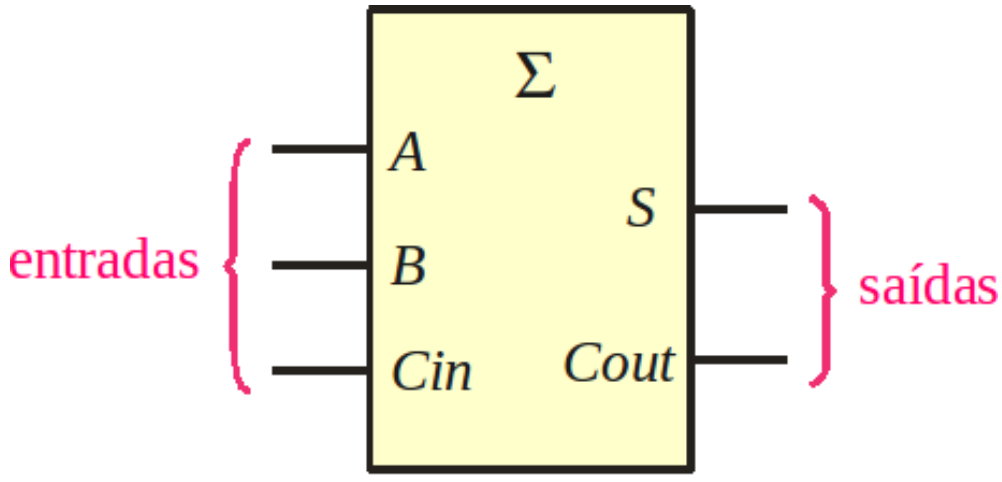
Inputs		Outputs	
A	B	C_{out}	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



Somador Completo (Full-Adder)

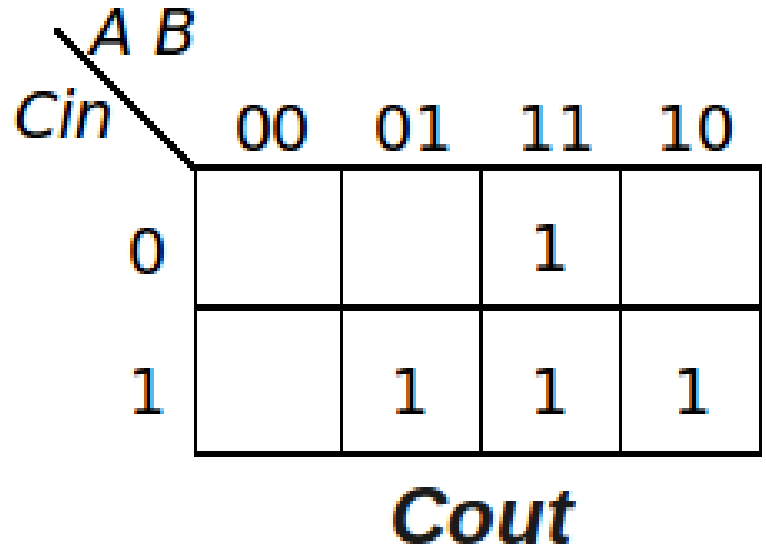
- **Sinais de entrada**:
  - Dado A de 1 bit
  - Dado B de 1 bit
  - CarryIn
- **Sinais de saída**:
  - Soma S de 1 bit
  - CarryOut

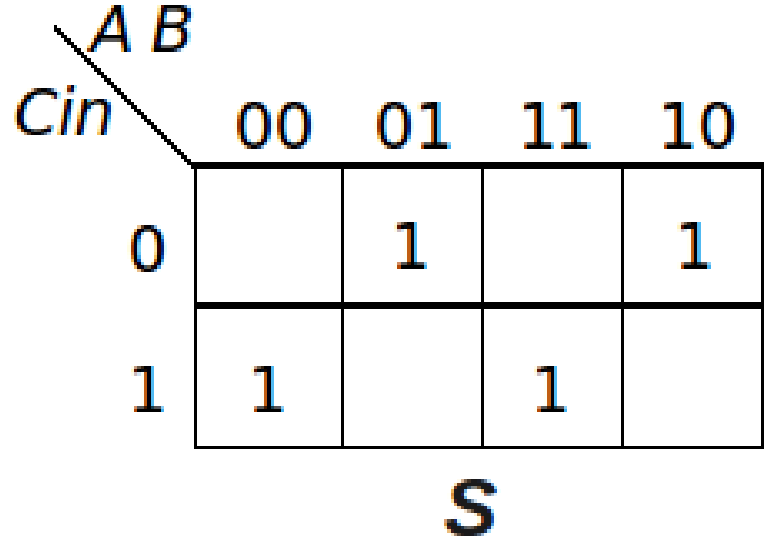
Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1



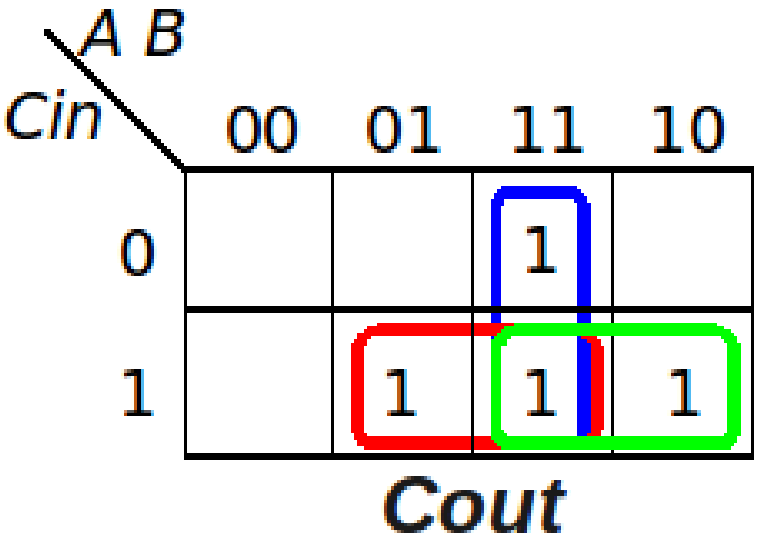
Somador Completo (Full-Adder)

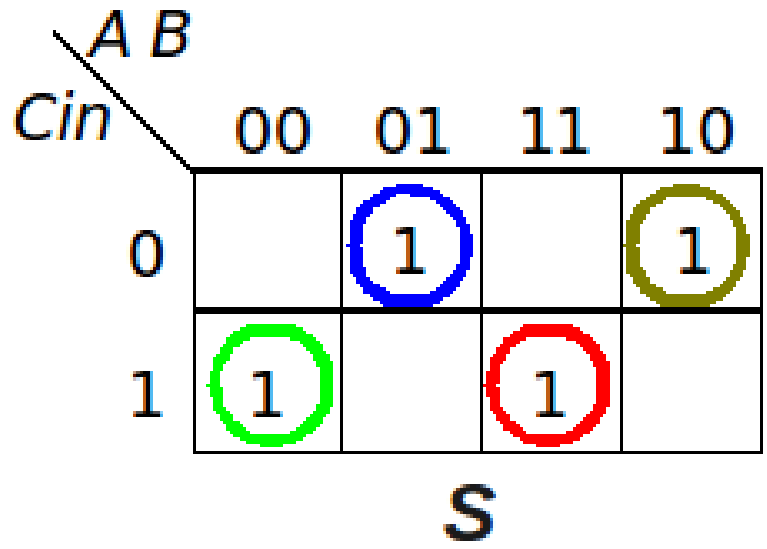
Inputs			Outputs	
A	B	C_{in}	C_{out}	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1





Somador Completo (Full-Adder)





$$Cout = A \bullet B + B \bullet Cin + A \bullet Cin$$

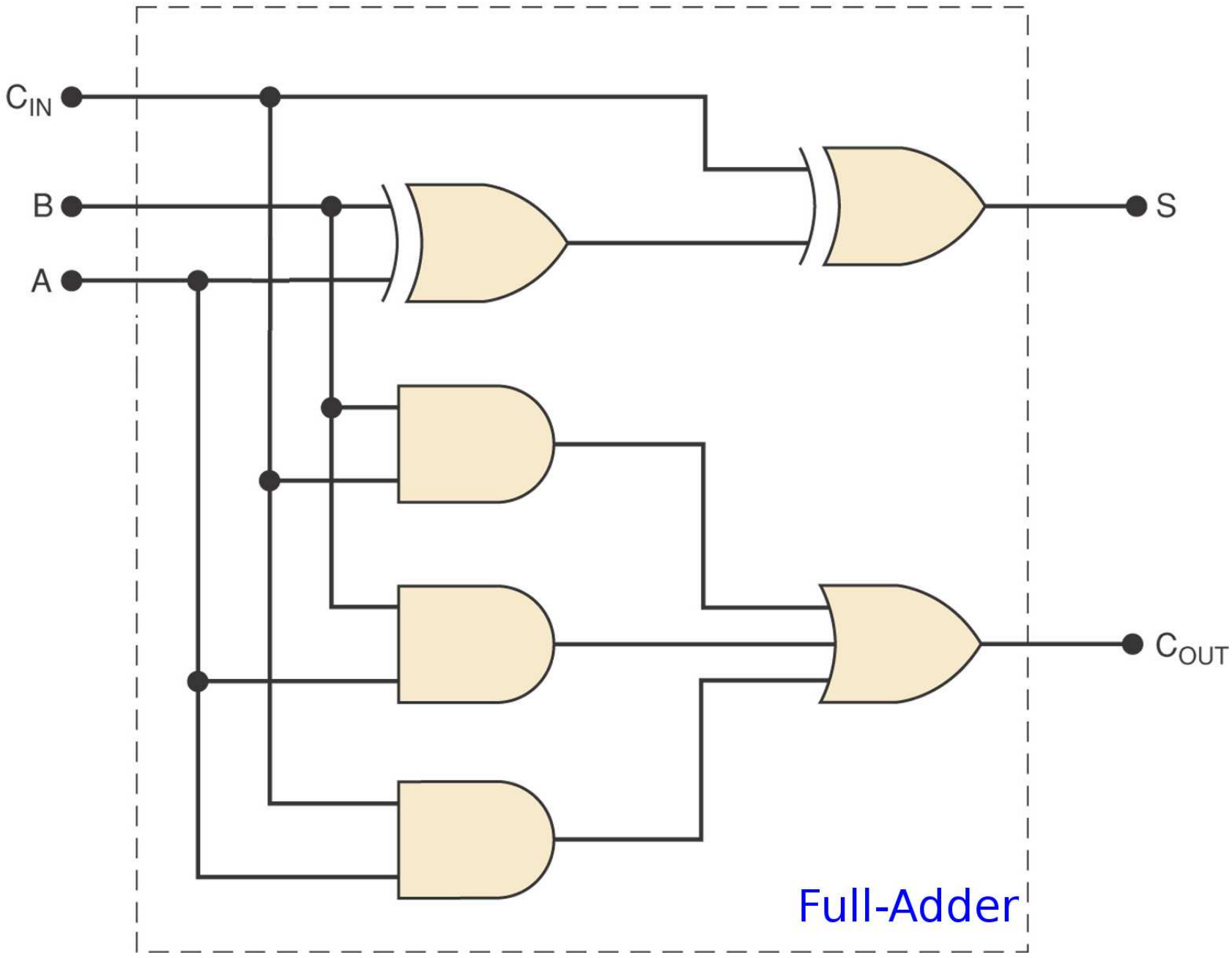
$$S = \overline{A} \bullet \overline{B} \bullet Cin + A \bullet B \bullet Cin + \overline{A} \bullet B \bullet \overline{Cin} + A \bullet \overline{B} \bullet \overline{Cin}$$

$$= (\overline{A} \bullet \overline{B} + A \bullet B) \bullet Cin + (\overline{A} \bullet B + A \bullet \overline{B}) \bullet \overline{Cin}$$

$$= (\overline{A \oplus B}) \bullet Cin + (A \oplus B) \bullet \overline{Cin}$$

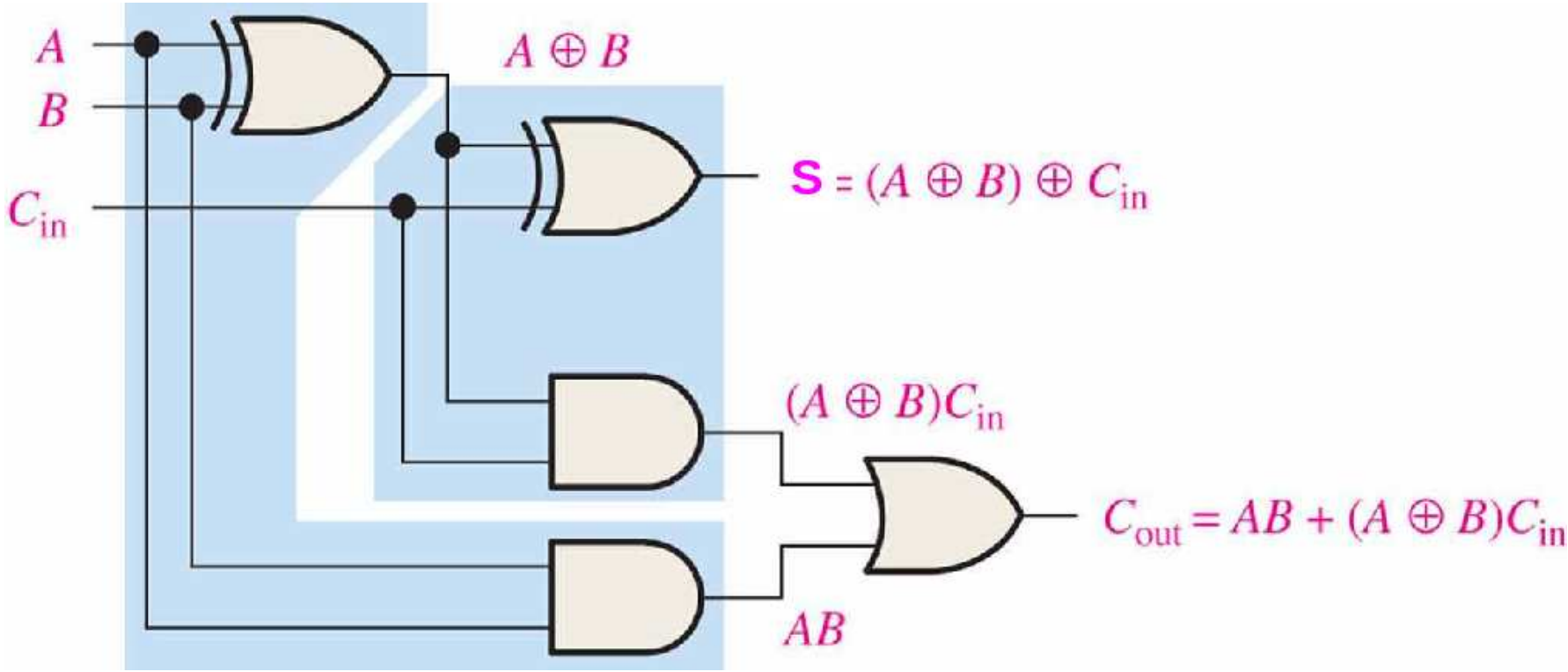
$$= (A \oplus B) \oplus Cin$$

Somador Completo (Full-Adder)



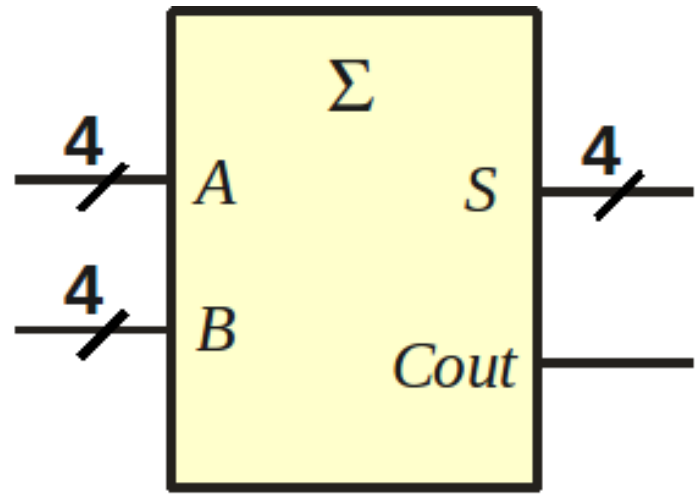
Somador Completo (Full-Adder)

- Implementado usando 2 half-adders:



Somador de Dados de 4 bits

- **Sinais de entrada**:
  - Dado A de 4 bits: $A_3\ A_2\ A_1\ A_0$
  - Dado B de 4 bits: $B_3\ B_2\ B_1\ B_0$
- **Sinais de saída**:
  - Soma S de 4 bits: $S_3\ S_2\ S_1\ S_0$
  - CarryOut

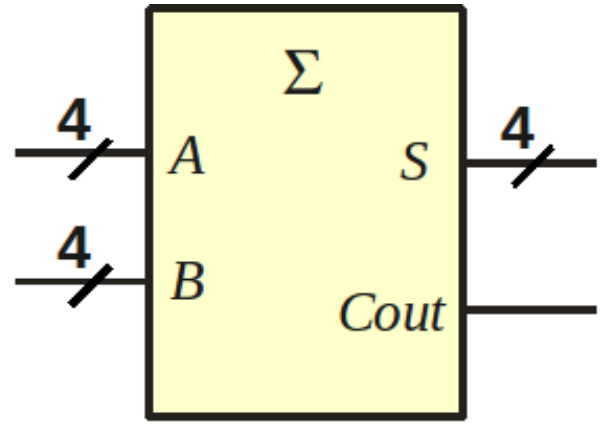


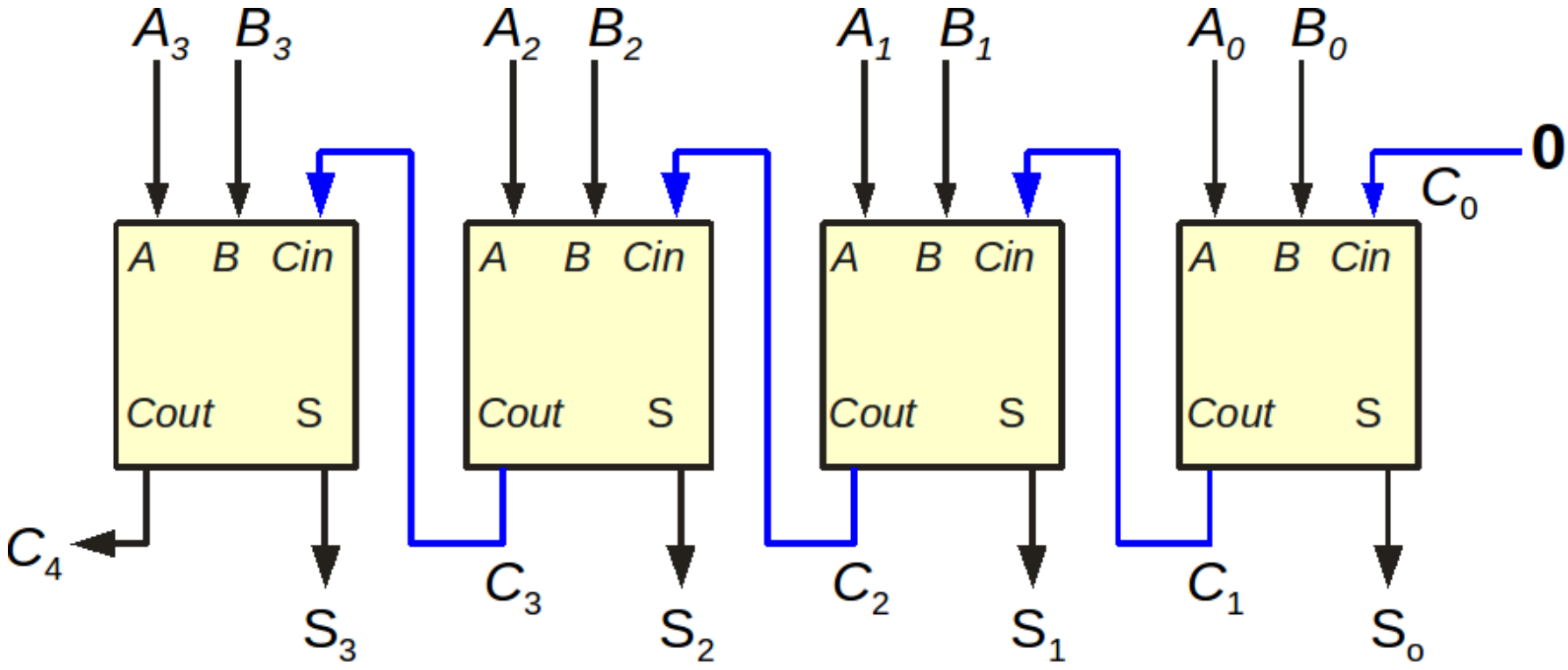
- **Replicação**:
  - Construído usando 4 **somadores completos** de 1 bit

- **Somador de 4 bits com ripple carry** (carry “em cascata”):
  - 4 somadores completos de 1 bit conectados,
    do bit **menos** significativo para bit **mais** significativo

Somador de Dados de 4 bits

- **Somador de 4 bits com ripple carry**:
  - Valor fornecido em C_0 ?





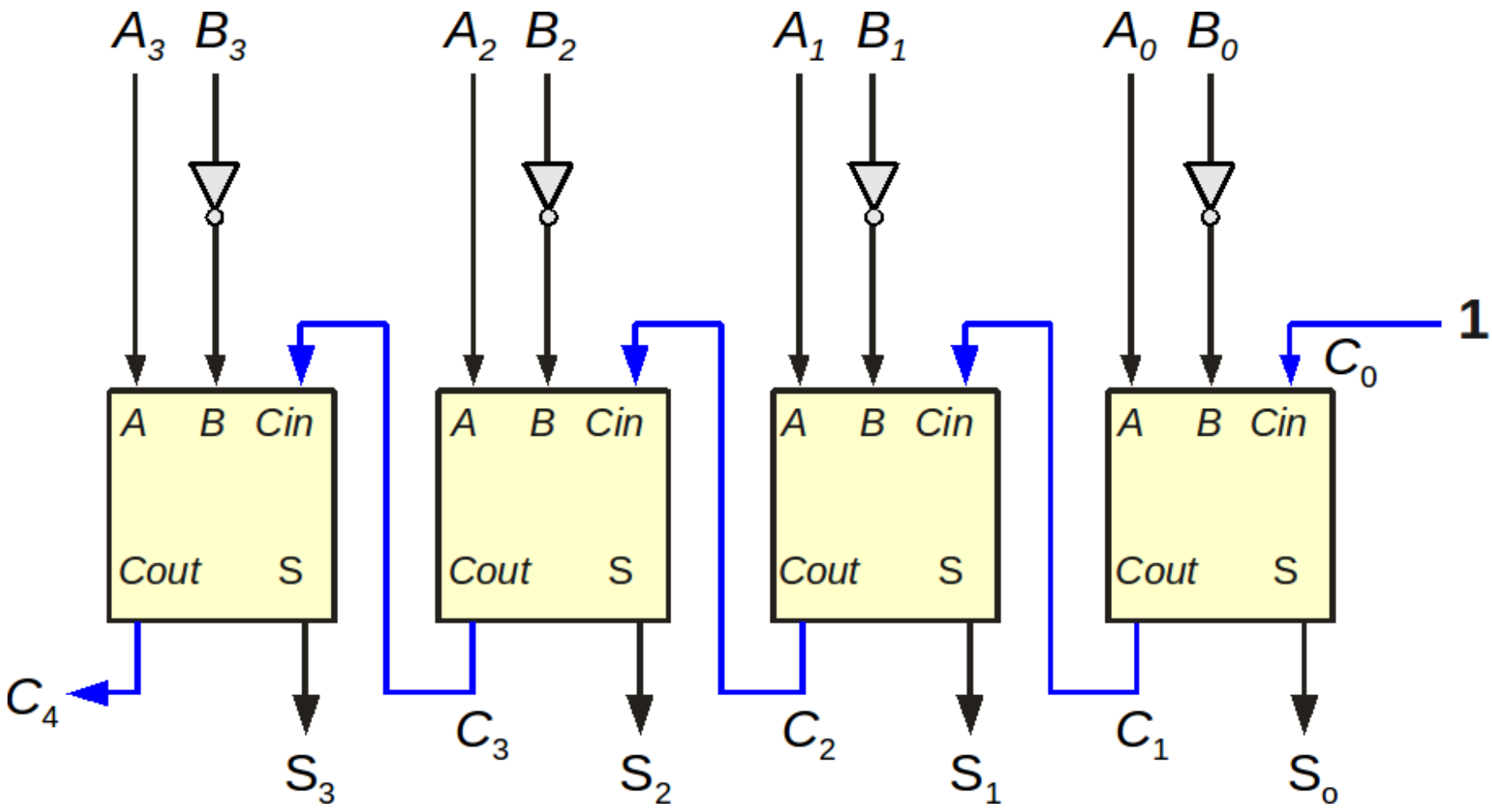
Subtração de Dados de n bits

- **Como implementar subtração** S = A − B ?
  - Dados A e B de n bits
  - Resultado S de n bits

- **Ideia**:
  - S = A − B
    = A + (−B)
    = A + complemento_a_2(B)
    = A + complemento_a_1(B) + 1
    = A + NOT(B) + 1
  - Realizar subtração através de uma soma
  - Usar inversores na entrada B
  - Como somar 1 no resultado S ?

Subtrador de Dados de 4 bits

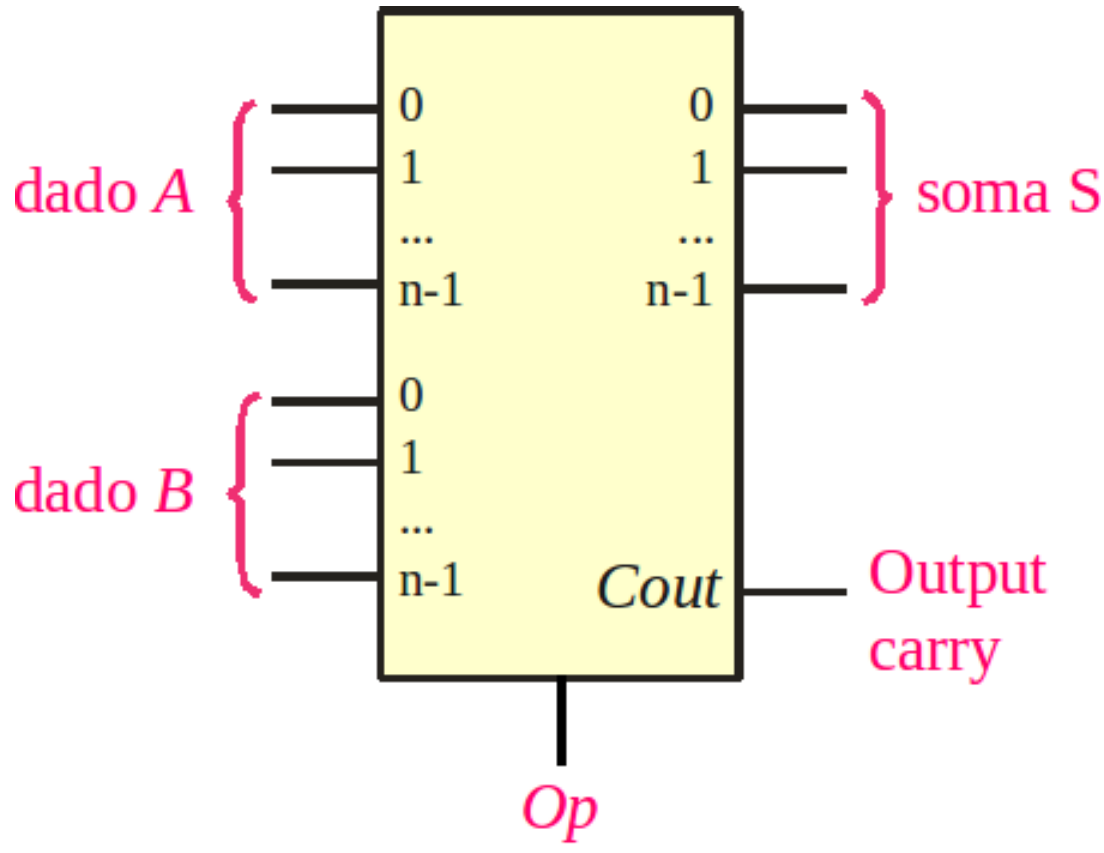
- Usando 4 somadores completos:



Somador/Subtrador de Dados de n bits

- **Sinais de entrada**:
  - Dado A de n bits: A_{n-1} ... A_2 A_1 A_0
  - Dado B de n bits: B_{n-1} ... B_2 B_1 B_0
  - Op: Op = 0 $\Rightarrow$ Soma
    Op = 1 $\Rightarrow$ Subtração

- **Sinais de saída**:
  - Resultado S de n bits:
    S_{n-1} ... S_2 S_1 S_0
    - Se Op = 0, S = A + B
    - Se Op = 1, S = A $-$ B
  - CarryOut



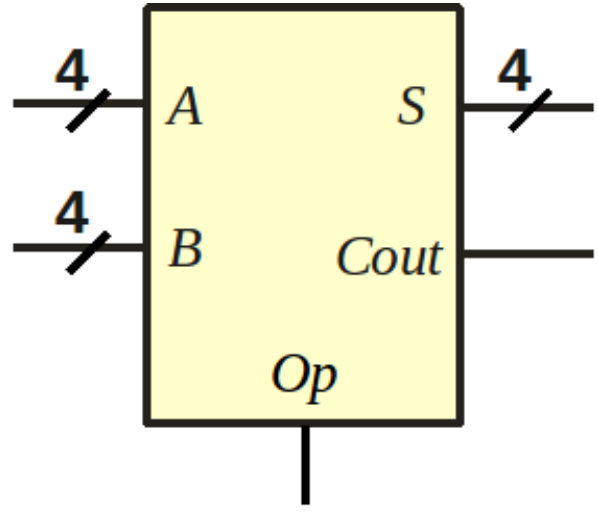
Somador/Subtrador de Dados de 4 bits

- **Sinais de entrada**:
  - Dado A de 4 bits: $A_3\ A_2\ A_1\ A_0$
  - Dado B de 4 bits: $B_3\ B_2\ B_1\ B_0$
  - Op: Op = 0 $\Rightarrow$ Soma
    Op = 1 $\Rightarrow$ Subtração

- **Sinais de saída**:
  - Resultado S de 4 bits:

    $S_3\ S_2\ S_1\ S_0$

    - Se Op = 0, S = A + B
    - Se Op = 1, S = A $-$ B
  - CarryOut



- **Como selecionar operação soma ou subtração ?**

Somador/Subtrador de Dados de 4 bits