

# Lógica e Sistemas Digitais

Síntese de somador, somador/subtrator e  
respetivos indicadores de erro e relacionais

João Pedro Patriarca ([jpatri@cc.isel.ipl.pt](mailto:jpatri@cc.isel.ipl.pt))

Slides inspirados nos slides do prof. Mário Véstias



# Somador de dois bits - semissomador (*half-adder*)

Símbolo

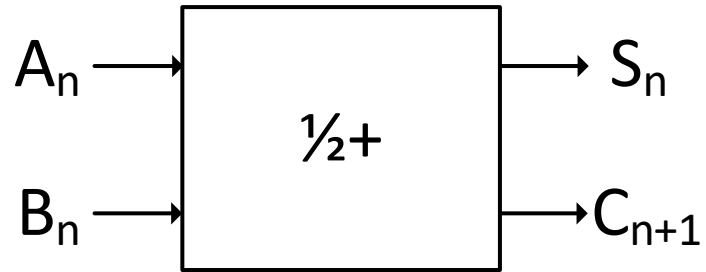


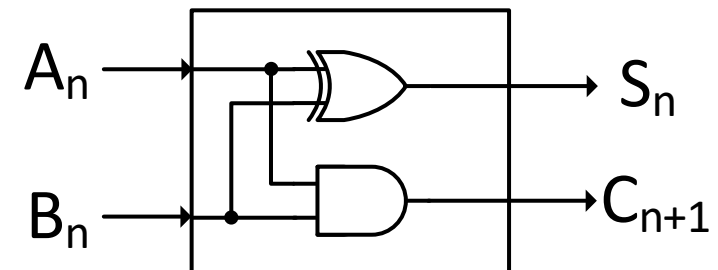
Tabela de verdade

$A_n$	$B_n$	$S_n$	$C_{n+1}$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

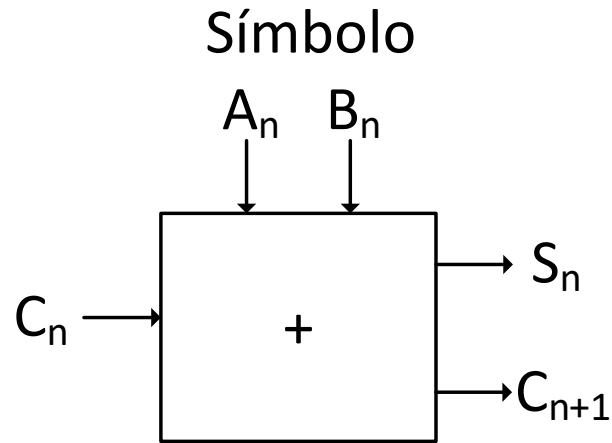
Expressões

$$S_n = \overline{A_n} \cdot B_n + A_n \cdot \overline{B_n} = A_n \oplus B_n$$
$$C_{n+1} = A_n \cdot B_n$$

Implementação interna



# Somador de três bits – somador completo (*full-adder*)



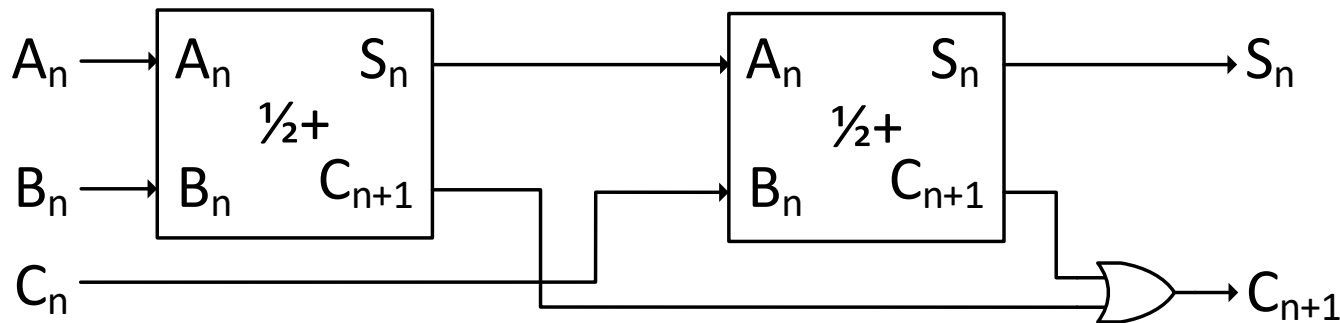
Expressões

$$\begin{aligned} S_n &= A \oplus B \oplus C_n \\ C_{n+1} &= A_n \cdot B_n + (A_n \oplus B_n) \cdot C_n \\ &\equiv A_n \cdot B_n \oplus (A_n \oplus B_n) \cdot C_n \\ &\equiv A_n \cdot B_n + A_n \cdot C_n + B_n \cdot C_n \end{aligned}$$

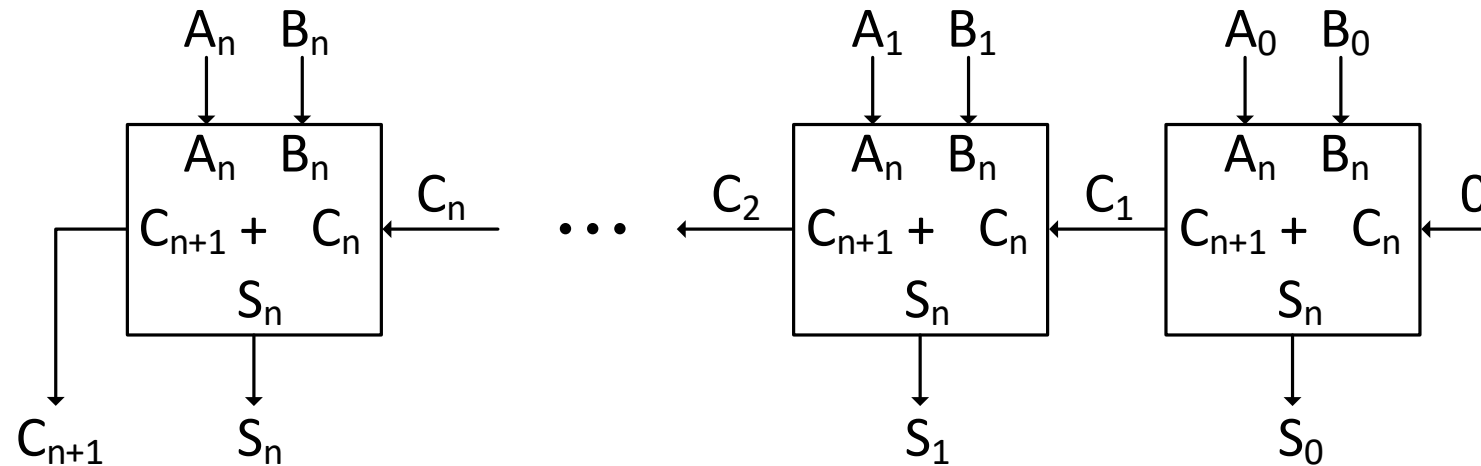
Tabela de verdade

$C_n$	$A_n$	$B_n$	$S_n$	$C_{n+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Implementação



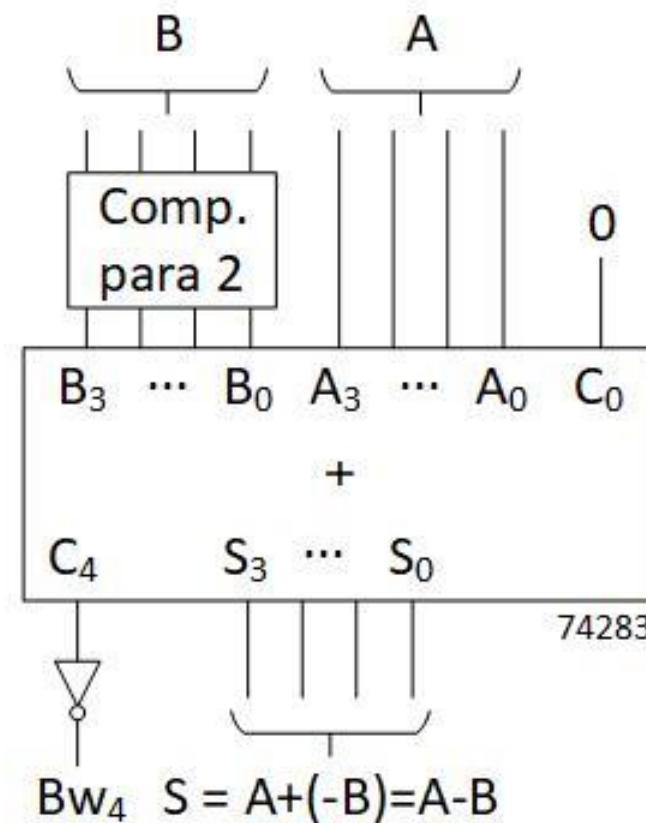
# Somador de palavras de $n$ bits



- O tempo de realização de uma adição depende do tempo de propagação de todos os *carries*
- O CI 74283 implementa um somador de palavras de 4 bits (operandos e resultado) composto por 4 somadores completos de 1 bit. Expõe ainda na interface física  $C_0$  ( $C_n$  da célula que produz  $S_0$ ) e  $C_4$  ( $C_{n+1}$  da célula que produz  $S_3$ ) para permitir a concatenação de dois ou mais somadores

# Subtrator de palavras de 4 bits – versão 1

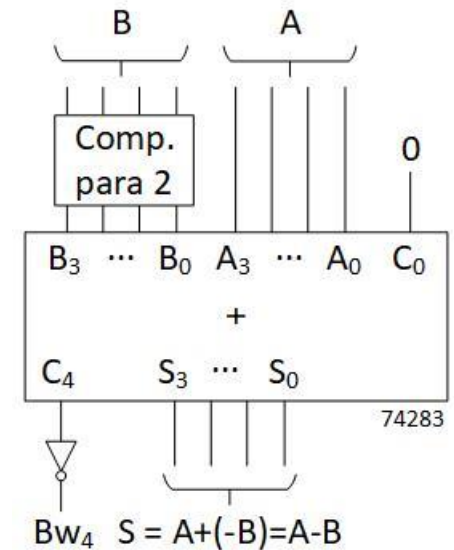
- Visto anteriormente, uma subtração pode ser realizada à custa de uma adição com o complemento:  $A - B = A + (-B)$



# Subtrator de palavras de 4 bits – versão 1

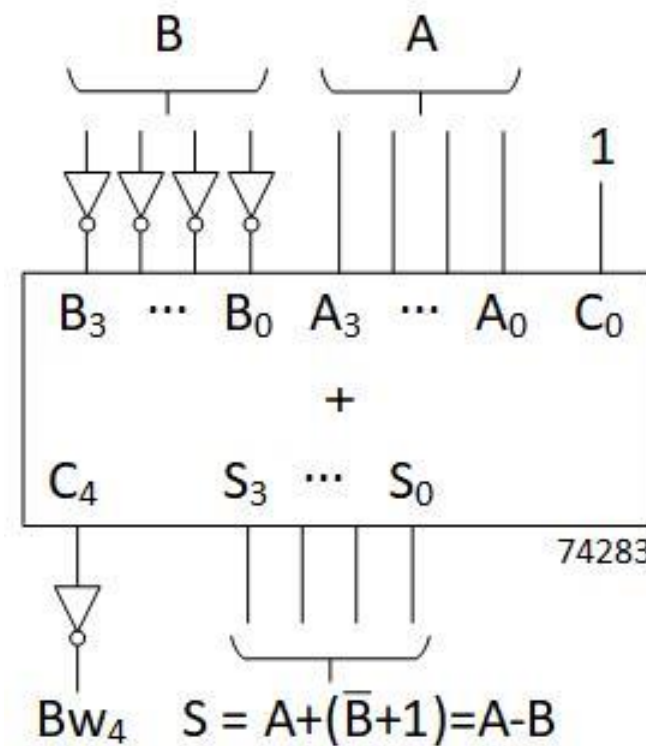
- Visto anteriormente, uma subtração pode ser realizada à custa de uma adição com o complemento:  $A - B = A + (-B)$

- A implementação do bloco “Complemento para 2” envolve o complemento bit a bit de B (complemento restringido) e adição com 1
- O complemento restringido corresponde à negação bit a bit de B e para a adição com 1 deve ser aproveitada a entrada  $C_0$  do somador



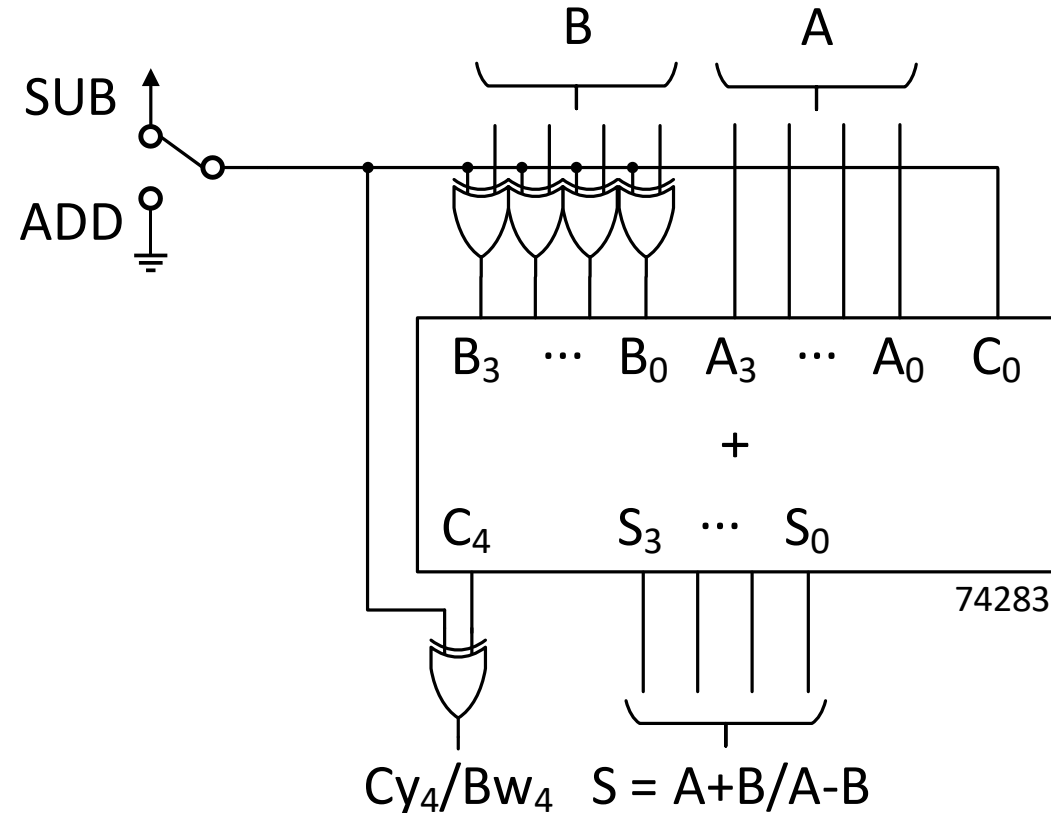
# Subtrator de palavras de 4 bits – versão 2

- Subtrator baseado no complemento restringido e aproveitamento da entrada  $C_0$  do somador para adicionar 1:  $-B = \bar{B} + 1$



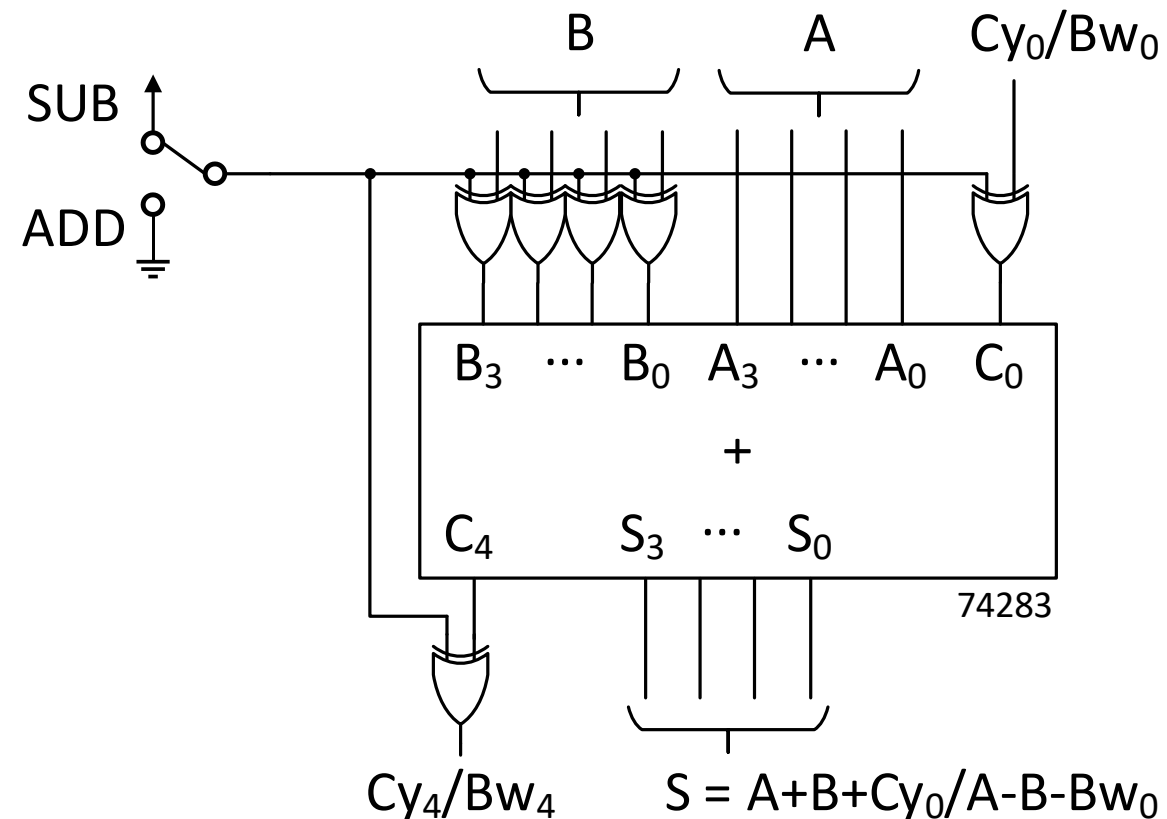
# Somador/subtrator de palavras de 4 bits

- Utilizando o mesmo somador, facilmente se constrói um somador/subtrator cuja operação é selecionada com base num bit de seleção (não permite apresentar os dois resultados em simultâneo)
- Relevante o seletor a 1 representar a subtração e a 0 representar a adição



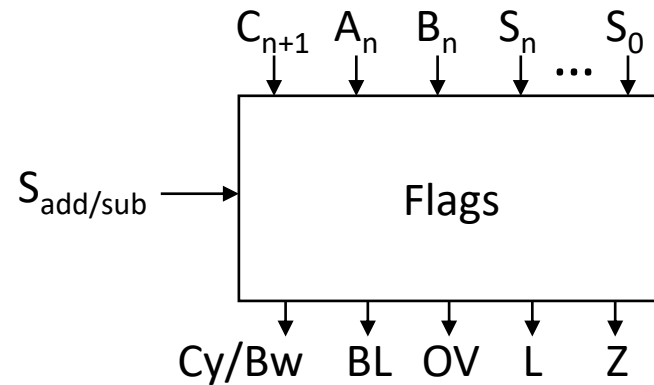


# Somador/subtrator de palavras de 4 bits concatenável



# Síntese das *flags* Cy/Bw (*Carry/Borrow*), OV (*overflow*), BL (*below*), L (*less*) e Z (*zero*)

$$\begin{array}{r} A_n \quad \dots \quad A_1 \quad A_0 \\ + \quad B_n \quad \dots \quad B_1 \quad B_0 \\ \hline C_{n+1} \quad S_n \quad \dots \quad S_1 \quad S_0 \end{array}$$



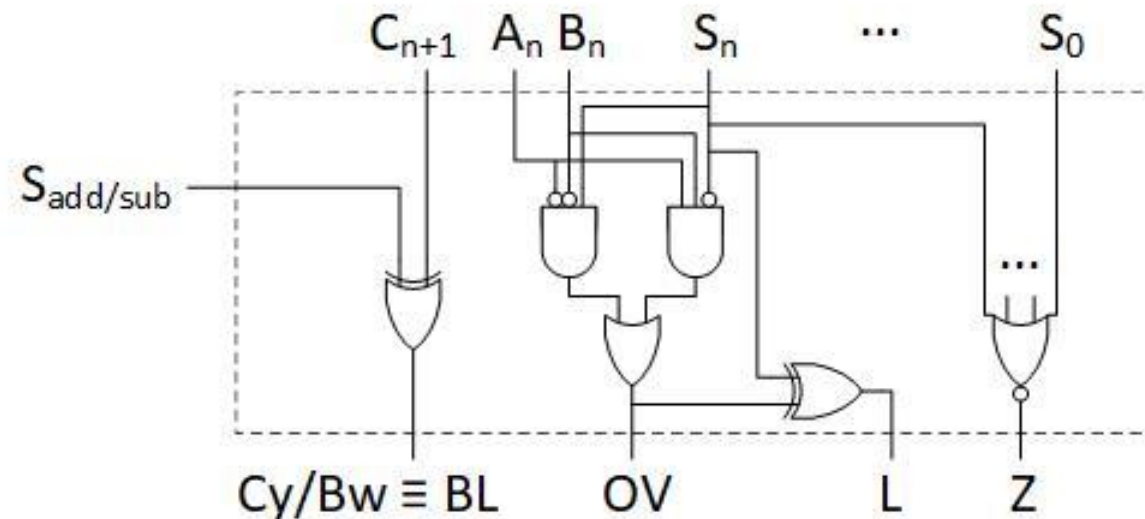
$$Cy/Bw = S_{add/sub} \oplus C_{n+1}$$

$$OV = \overline{A_n} \cdot \overline{B_n} \cdot S_n + A_n \cdot B_n \cdot \overline{S_n}$$

$$BL = Cy/Bw$$

$$L = S_n \oplus OV$$

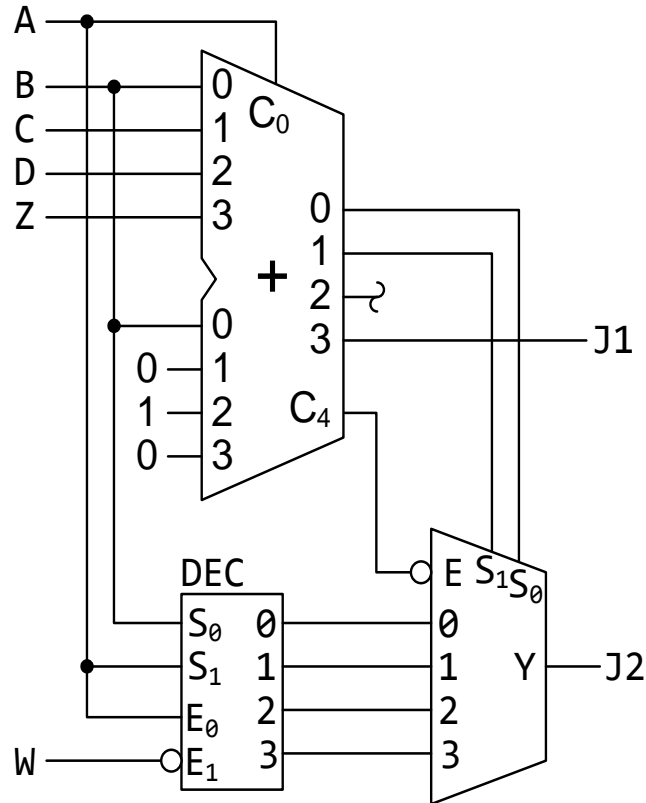
$$Z = \overline{S_n + \dots + S_1 + S_0}$$



# Exercício

# Análise de circuito combinatório #1

- Determine as expressões lógicas J1 e J2



# Resolução do exercício análise de circuito combinatório #1

- Impraticável usar tabela de verdade ou mapa de Karnaugh, dado o número de variáveis de entrada (6 variáveis  $\Rightarrow 2^6 = 64$  combinações)
- Impraticável escrever as expressões genéricas dos circuitos, substituir pelas variáveis de entrada do circuito e proceder às simplificações
- Deve analisar-se o circuito das saídas para as entradas e identificar situações que condicionam a ativação das saídas

- Análise para determinar J1:

- Tratando-se da saída com peso 8 do somador e com um conjunto de entradas com valores constantes, o melhor será realizar a conta “ao alto”

	$D + B.C$	$B.C$	$B$	$A$
	$Z$	$D$	$C$	$B$
+	0	1	0	$B$
<hr/>				
	$Z.(D + B.C)$	$Z \oplus (D + B.C)$	$-$	$B \oplus C$
				$A$

- Logo

$$J1 = Z \oplus (D + B.C)$$

# Resolução do exercício análise de circuito combinatório #1 - continuação

---

- Análise para determinar J2:
  - J2 depende dos *Enables* do Dec e do Mux
  - Logo, A tem de ser 1, W tem de ser 0 e C4 (somador) tem de ser 0
  - O facto de S1 do Dec ser A, apenas se consegue ativar as saídas 2 e 3 do Dec
  - O facto da saída 0 do somador depender de A (pela conta anterior) e A ter de ser 1, a análise do Mux fica limitada à entrada 3 do mesmo
  - Logo

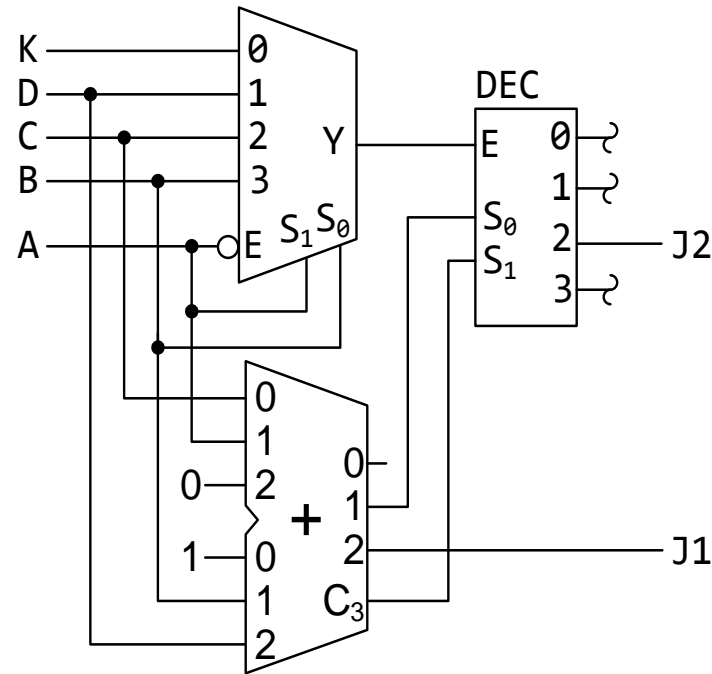
$$J2 = A \cdot \bar{W} \cdot \overline{Z \cdot (D + B \cdot C)} \cdot S_{1mux} \cdot S_0 \cdot B, \text{ substituindo } S_1 \text{ e } S_0 \text{ fica}$$

$$J2 = A \cdot \bar{W} \cdot \overline{Z \cdot (D + B \cdot C)} \cdot (B \oplus C) \cdot A \cdot B, \text{ simplificando porque a expressão apenas dá 1 se } B=1 \text{ e já depende de } A=1$$

$$J2 = A \cdot \bar{W} \cdot \overline{Z \cdot D} \cdot \bar{C} \cdot B$$

# Análise de circuito combinatório #2

- Determine as expressões lógicas J1 e J2



# Resolução do exercício análise de circuito combinatório #2

- Impraticável usar tabela de verdade ou mapa de Karnaugh, dado o número de variáveis de entrada (5 variáveis  $\Rightarrow 2^5 = 32$  combinações)
- Impraticável escrever as expressões genéricas dos circuitos, substituir pelas variáveis de entrada do circuito e proceder às simplificações
- Deve analisar-se o circuito das saídas para as entradas e identificar situações que condicionam a ativação das saídas
- Análise para determinar J1:
  - Tratando-se da saída com peso 4 do somador e com um conjunto de entradas com valores constantes, o melhor será realizar a conta “ao alto”

$$\begin{array}{r}
 A.B + A.C + B.C \qquad C \\
 0 \qquad A \qquad C \\
 + \qquad D \qquad B \qquad 1 \\
 \hline
 D.(A.B + A.C + B.C) \quad D \oplus (A.B + A.C + B.C) \quad A \oplus B \oplus C \quad \bar{C}
 \end{array}$$

- Logo

$$J1 = D \oplus (A.B + A.C + B.C)$$



# Resolução do exercício análise de circuito combinatório #2 - continuação

- Análise para determinar J2:

- J2 depende do *Enable* do Dec e dos respetivos seletores S0 e S1 terem na entrada 0 e 1, respetivamente
- O Enable do Dec depende da saída do Mux estar ativa, logo a entrada E do Mux tem de estar Enable e, portanto, A tem de ser 0
- A a 0 condiciona a seleção às entradas 0 e 1 do Mux
- A saída do Mux fica reduzida à expressão

$$Y_{mux} = \bar{A}.(\bar{B}.K + B.D)$$

- O facto de J2 depender de A a 0 permite simplificar as expressões do somador

	$B.C$	$C$	
	0	0	$C$
+	$D$	$B$	1
<hr/>			
$B.C.D$	$D \oplus (B.C)$	$B \oplus C$	$\bar{C}$

- E a expressão de J2 fica

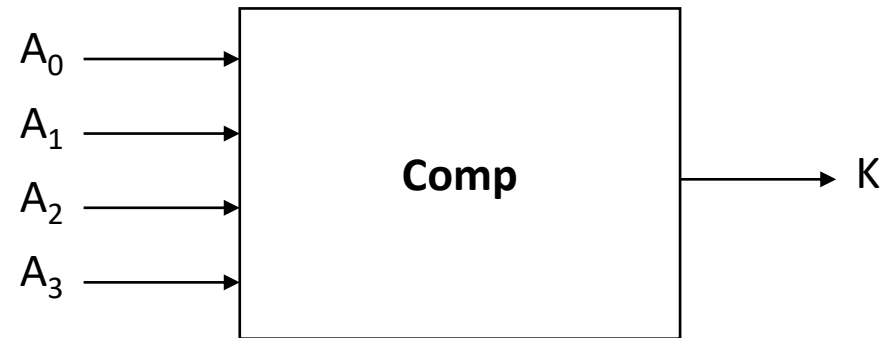
$J2 = \bar{A}.(\bar{B}.K + B.D).\overline{B \oplus C}.B.C.D$ , simplificando porque a expressão apenas dá 1 se B=C=D=1

$$J2 = \bar{A}.B.C.D$$

# Funções com somadores #1

---

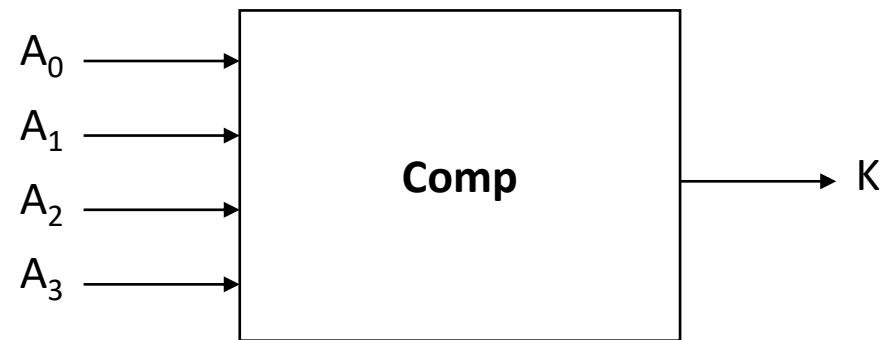
- Utilizando apenas somadores completos de operandos de  $n$  bits, implemente o circuito **Comp**, que ativa a saída  $K$  quando  $A > +3$ . O operando  $A$  é um inteiro com sinal codificado em código dos complementos a 4 bits.



## Funções com somadores #2

---

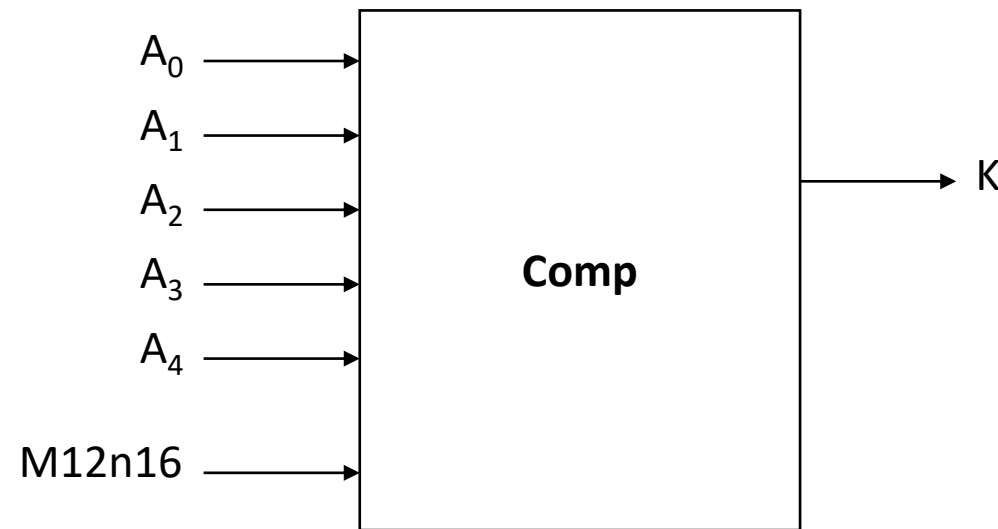
- Utilizando preferencialmente apenas somadores completos de 4 bits, na menor quantidade possível, realize o sistema **Comp** apresentado na figura. A saída K fica ativa quando  $4 < A < 12$  considerando A como um número inteiro, representado em código binário natural.



## Funções com somadores #3

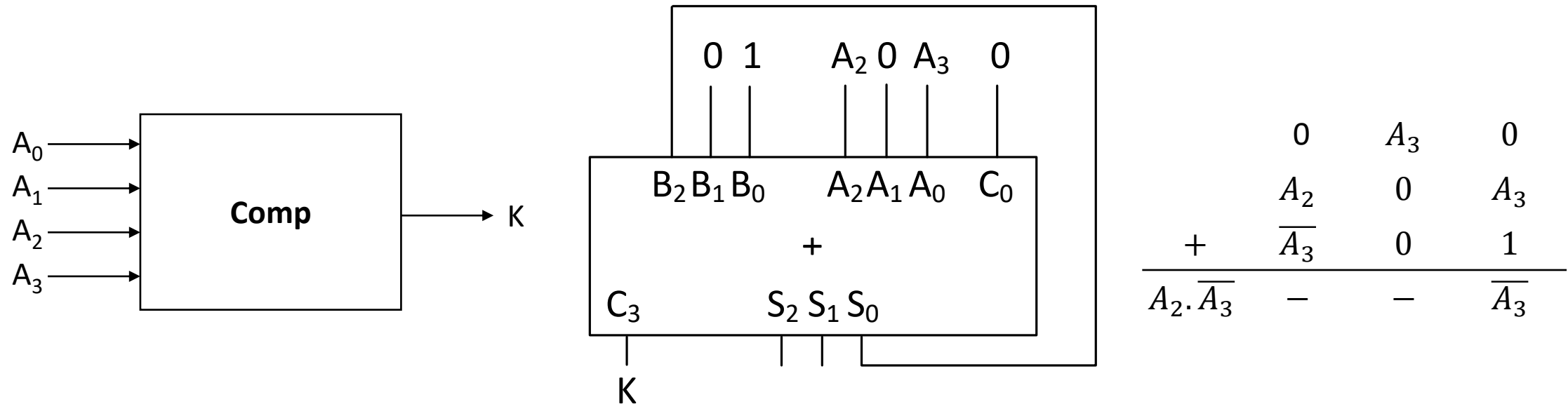
---

- Utilizando um somador completo de operandos de 4 bits, realize o sistema apresentado na figura. A saída K fica ativa quando  $A > 12$  se a entrada M12n16 está a um ou quando  $A \geq 16$  se a entrada M12n16 está a zero. A é entendido como um número natural. Se necessário poderá usar portas lógicas na menor quantidade possível.



# Solução funções com somadores #1

- Utilizando apenas somadores completos de operandos de  $n$  bits, implemente o circuito **Comp**, que ativa a saída  $K$  quando  $A > +3$ . O operando  $A$  é um inteiro com sinal codificado em código dos complementos.  $K = A_2 \cdot \overline{A_3}$

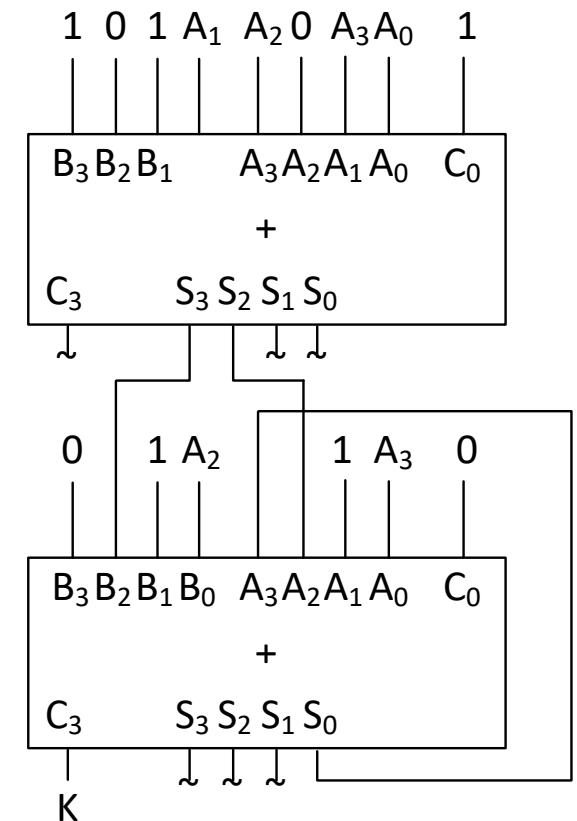


# Solução funções com somadores #2

- Utilizando preferencialmente apenas somadores completos de 4 bits, na menor quantidade possível, realize o sistema **Comp** apresentado na figura. A saída K fica ativa quando  $4 < A < 12$  considerando A como um número inteiro, representado em código binário natural.  $K = (A_3 \oplus A_2) \cdot \overline{A_3} \cdot A_2 \cdot \overline{A_1} \cdot \overline{A_0}$

$$\begin{array}{r}
 0 \quad A_3 + A_1 + A_0 \quad A_1 + A_0 \quad 1 \\
 A_2 \quad 0 \quad A_3 \quad A_0 \\
 + \quad 1 \quad 0 \quad 1 \quad A_1 \\
 \hline
 - \quad \overline{A_2} \quad A_3 + A_1 + A_0 \quad - \quad -
 \end{array}$$

$$\begin{array}{r}
 A_3 + \overline{A_2} + A_1 + A_0 \quad 1 \quad A_3 \cdot A_2 \quad 0 \\
 A_3 \oplus A_2 \quad A_3 + A_1 + A_0 \quad 1 \quad A_3 \\
 + \quad 0 \quad \overline{A_2} \quad 1 \quad A_2 \\
 \hline
 (A_3 \oplus A_2) \cdot \overline{A_3} \cdot A_2 \cdot \overline{A_1} \cdot \overline{A_0} \\
 (A_3 + \overline{A_2} + A_1 + A_0) \quad - \quad - \quad - \quad A_3 \oplus A_2
 \end{array}$$



# Solução funções com somadores #3

- Com M12n16 a 1 adiciona 3 a A gerando arrasto se  $A > 12$  e  $A < 16$ . Neste caso  $K=1$  por C4.
- Com M12n16 a 1, se  $A \geq 16$   $K=1$  por A4 gerando o somador arrasto ou não
- Com M12n16 a 0 o somador nunca gera arrasto e  $K=1$  exclusivamente por A4

