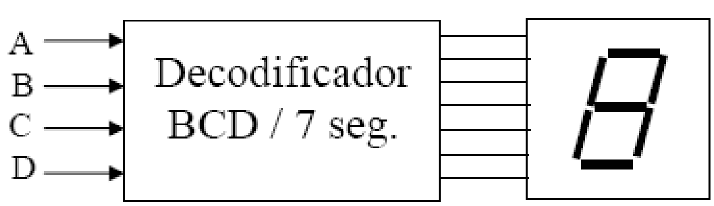
Revisão da aula anterior

- **Códigos binários:** seqüências de números binários dispostos de alguma maneira: BCD8421, excesso 3, Gray, Johnson, etc.;

- **Transformação entre códigos:** codificadores e decodificadores;

- **Projetos de decodificadores (passos):**
  - P1: Levantar a tabela verdade da transformação desejada;
  - P2: Para cada variável de saída proceder minimização;
  - P3: De posse das minimizações de todas as variáveis de saída, determinar o circuito digital que implementa a transformação desejada.

- **Display** de 7 segm.
  - Cátodo comum (Lig. = 1)
  - Ânodo comum (Lig. = 0)



Circuitos aritméticos

Existem alguns circuitos combinacionais que possuem funções específicas. Dentre eles, os circuitos aritméticos têm grande importância, pois constituem a **ULA** (*Unidade Lógico-Aritmética*) dos processadores.

Essa classe de circuitos implementa a aritmética binária básica. Assim sendo, os circuitos aritméticos básicos dividem-se em:

- Somadores;
- Subtratores;
- Somadores/subtratores;

Os circuitos aritméticos, como todo circuito combinacional, pode ser implementado utilizando-se portas booleanas básicas (AND, OR, NOT, XOR…).

Seu projeto segue os mesmos moldes dos vistos até aqui.

O meio somador (*half adder*)

A soma binária é dada por:

$$\begin{array}{cccc} 0 & 0 & 1 & {}^{1}1 \\ +0 & +1 & +0 & +1 \\ \hline 0 & 1 & 1 & 10 \end{array}$$

A tabela verdade da soma é dada por:

ENTRADA		SAÍDA	
A	B	S	Ts
0	*0*	*0*	*0*
0	*1*	*1*	*0*
1	*0*	*1*	*0*
1	*1*	*0*	*1*

Ts – Transporte de saída (*carry out*)

0 + 0 = 0 e Ts = 0

0 + 1 = 1 e Ts = 0

1 + 0 = 1 e Ts = 0

1 + 1 = 0 e Ts = 1

Minimizando, teremos:

S	$\overline{B}$	B
$\overline{A}$	0	1
A	1	0

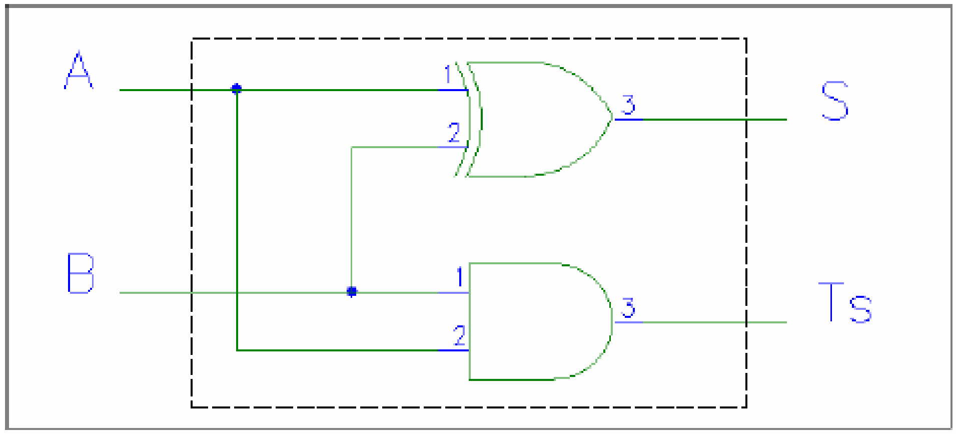
Ts	$\overline{B}$	B
$\overline{A}$	0	0
A	0	1

$$S = A \oplus B$$

$$Ts = A.B$$

O circuito obtido das expressões minimizadas é:

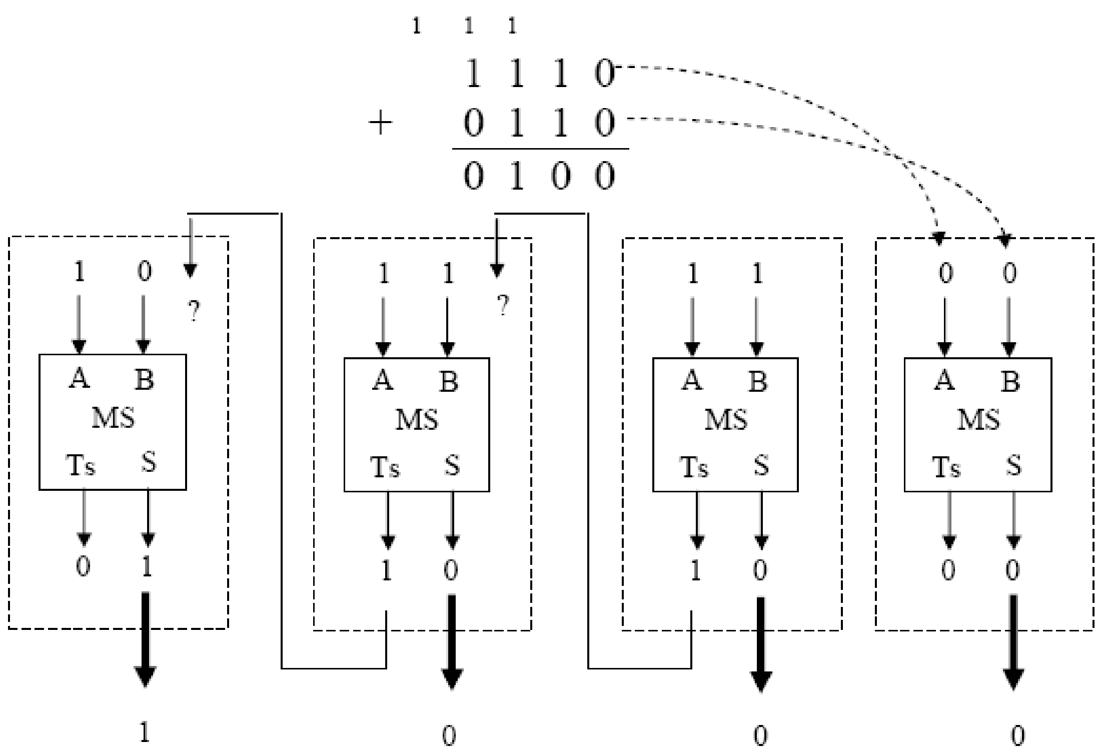
Meio somador



Esse circuito é usualmente resumido pelo bloco lógico do meio somador, que é dado por:

Consideração sobre o meio somador

O meio somador somente é capaz de realizar somas de dois números de 1 bit. Embora para somas com mais de um bit possamos utilizar mais de uma unidade de meios somadores (sendo uma para cada bit), isso não é feito porque esse bloco funcional não observa o bit “vai um” proveniente de outras unidades. Seja por exemplo a soma binária:



RESPOSTA INCORRETA DEVIDO À NÃO OBSERVAÇÃO DOS BITS “VAI UM”

O somador completo (*full adder*)

Um somador mais eficiente pode ser conseguido observando-se o bit "vai um" proveniente do resultado da soma do bit anterior. Assim, para o somador completo:

ENTRADA			SAÍDA	
A	B	Te	S	Ts
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

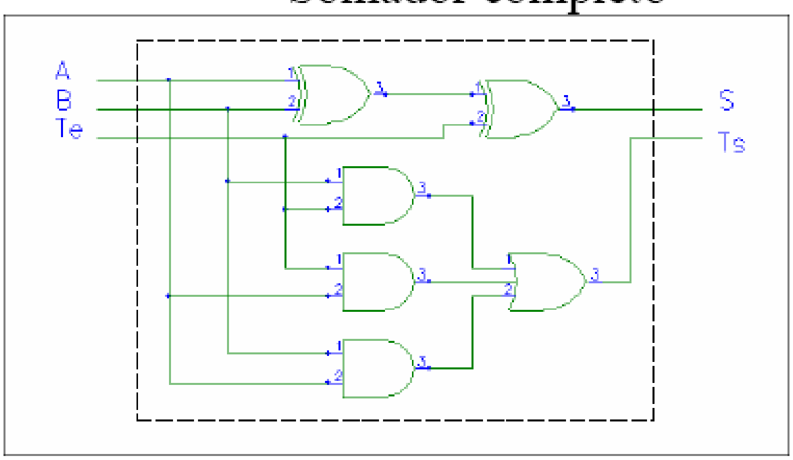
Minimizando as variáveis de saída:

$$S = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + A.\overline{B}.\overline{Te} + A.B.Te$$

$$Ts = \overline{A}.B.Te + A.\overline{B}.Te + A.B.\overline{Te} + A.B.Te$$

Através das expressões minimizadas, podemos obter o circuito para o somador completo:

Somador completo



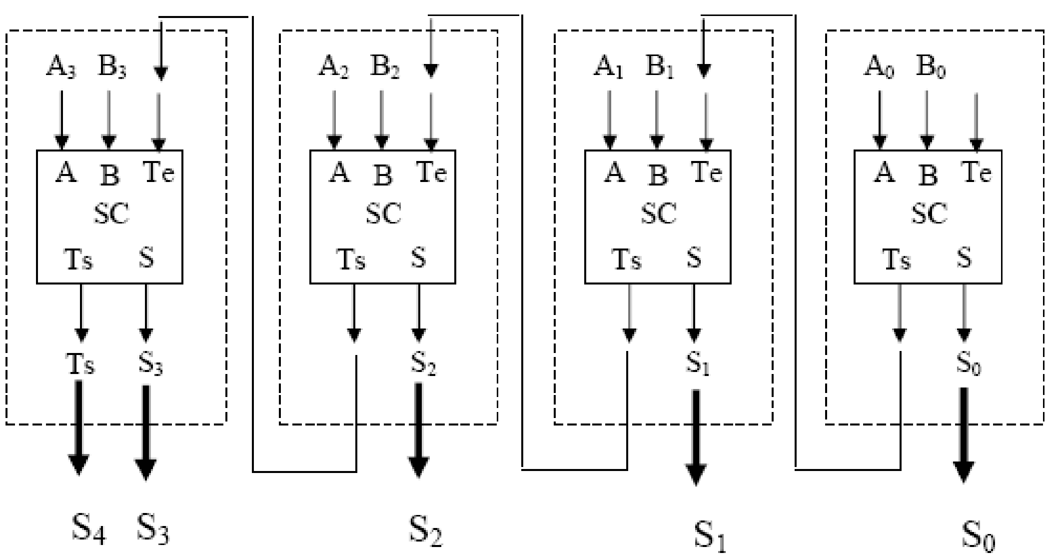
O bloco funcional para o somador completo é:

Sejam dois números binários de 4 bits:

$$\begin{cases} A \Rightarrow A_3 \;\; A_2 \;\; A_1 \;\; A_0 \\ B \Rightarrow B_3 \;\; B_2 \;\; B_1 \;\; B_0 \end{cases}$$

Uma soma utilizando o somador completo é dada por:

$$\begin{array}{rcccc} & A_3 & A_2 & A_1 & A_0 \\ + & B_3 & B_2 & B_1 & B_0 \\ \hline (S_4) & S_3 & S_2 & S_1 & S_0 \end{array}$$



Meio Subtrator (*half subtractor*)

Subtração binária:

0	0	1	1
- 0	- 1	- 0	- 1
0	1 (*)	1	0

A tabela verdade da subtração para um bit é:

ENTRADA		SAÍDA	
A	B	S	Ts
0	*0*	*0*	*0*
0	*1*	*1*	*1*
1	*0*	*1*	*0*
1	*1*	*0*	*0*

Ts – Transporte de saída

0 - 0 = 0 e Ts = 0

0 - 1 = 1 e Ts = 1

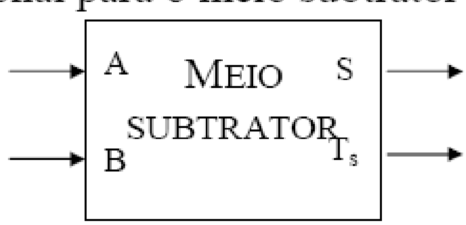
1 - 0 = 1 e Ts = 0

1 - 1 = 0 e Ts = 0

Minimizando, teremos:

$$\begin{cases} S = A \oplus B \\ Ts = \overline{A}.B \end{cases}$$

O bloco funcional para o meio subtrator é:



(*) O meio subtrator incorre nos mesmos problemas que o meio somador

O subtrator completo (*full subtractor*)

Da forma análoga ao somador completo, o subtrator completo opera com o bit "vem um". Assim, sua tabela verdade é dada por:

ENTRADA			SAÍDA	
A	B	Te	S	Ts
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Te – Transporte de entrada

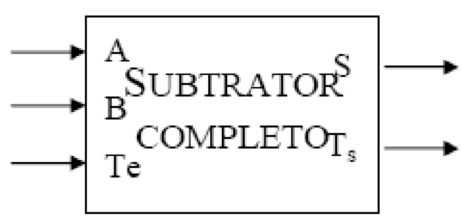
Ts – Transporte de saída

Minimizando as variáveis de saída teremos:

$$S = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + A.\overline{B}.\overline{Te} + A.B.Te$$

$$Ts = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + \overline{A}.B.Te + A.B.Te$$

O bloco funcional do subtrator será dado por:



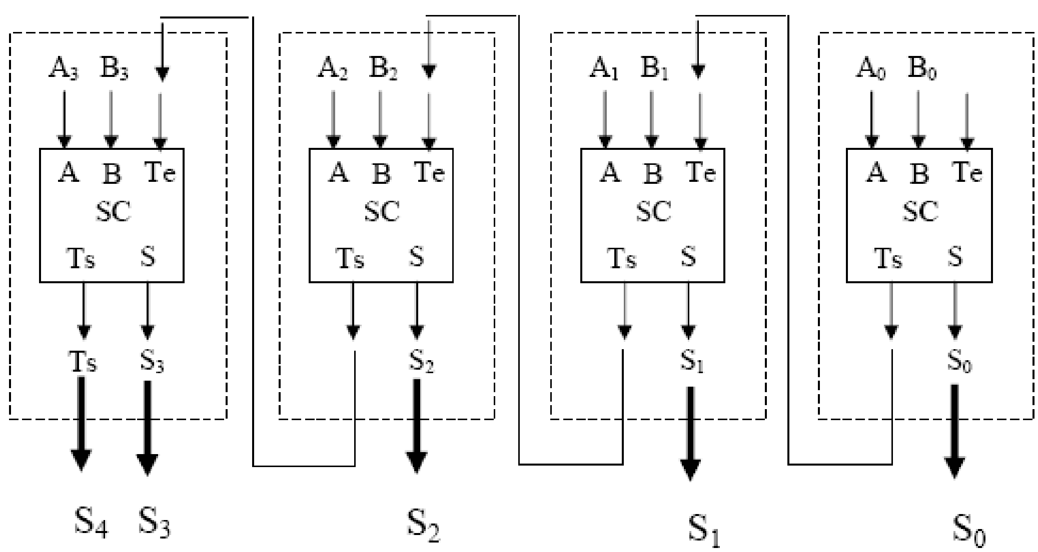
Exemplo de subtração

Sejam dois números binários de 4 bits:

$$\begin{cases} A \Rightarrow A_3 \ \ A_2 \ \ A_1 \ \ A_0 \\ B \Rightarrow B_3 \ \ B_2 \ \ B_1 \ \ B_0 \end{cases}$$

Uma subtração utilizando o subtrator completo é dada por:

$$\begin{array}{rcccc} & A_3 & A_2 & A_1 & A_0 \\ - & B_3 & B_2 & B_1 & B_0 \\ \hline & S_3 & S_2 & S_1 & S_0 \end{array}$$



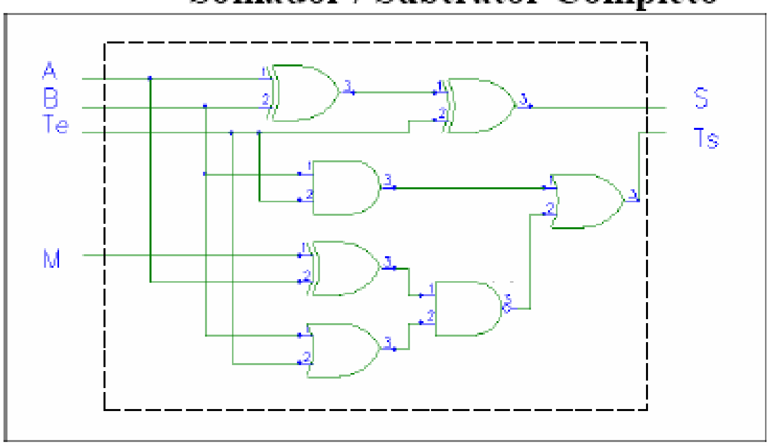
Somador Subtrator

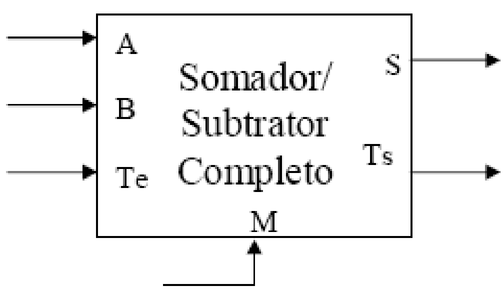
M	A	B	Te	S	Ts
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	1	1

Soma completa (M=0)

Subtração completa (M=1)

Somador / Subtrator Completo





Soma completa com meio-somadores

	S=A⊕B Ts=A.B
	$S = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + A.\overline{B}.\overline{Te} + A.B.Te$ $Ts = \overline{A}.B.Te + A.\overline{B}.Te + A.B.\overline{Te} + A.B.Te$

Das expressões para o somador completo, teremos:

Se "S":

$$S = Te(\overline{A}.\overline{B} + A.B) + \overline{Te}.(\overline{A}.B + A.\overline{B})$$

$$S = Te.\overline{(A \oplus B)} + \overline{Te}.(A \oplus B)$$

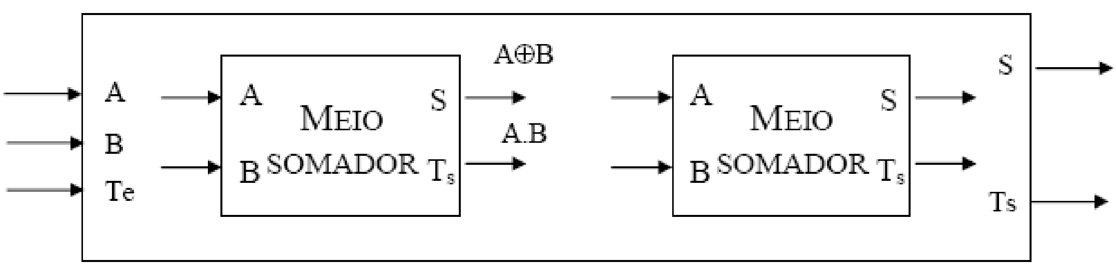
$$S = Te \oplus (A \oplus B)$$

$$\boxed{S = A \oplus B \oplus Te}$$

De "Te":

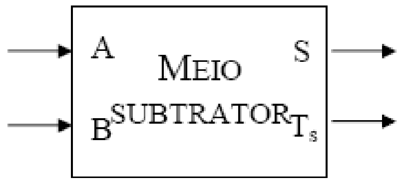
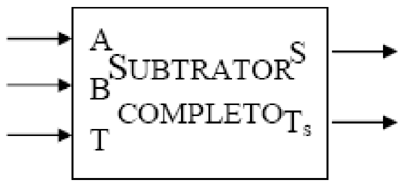
$$Ts = Te(\overline{A}.B + A.\overline{B}) + A.B(\overline{Te} + Te)$$

$$\boxed{Ts = Te.(A \oplus B) + AB}$$



(somador completo)

Subtração completa c/ meio-subtratores

	S=A⊕B $Ts = \overline{A}.B$
	$S = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + A.\overline{B}.\overline{Te} + A.B.Te$ $Ts = \overline{A}.\overline{B}.Te + \overline{A}.B.\overline{Te} + \overline{A}.B.Te + A.B.Te$

Das expressões para o somador completo, teremos:

Se "S":

$$S = Te(\overline{A}.\overline{B} + A.B) + \overline{Te}.(\overline{A}.B + A.\overline{B})$$

$$S = Te.\overline{(A \oplus B)} + \overline{Te}.(A \oplus B)$$

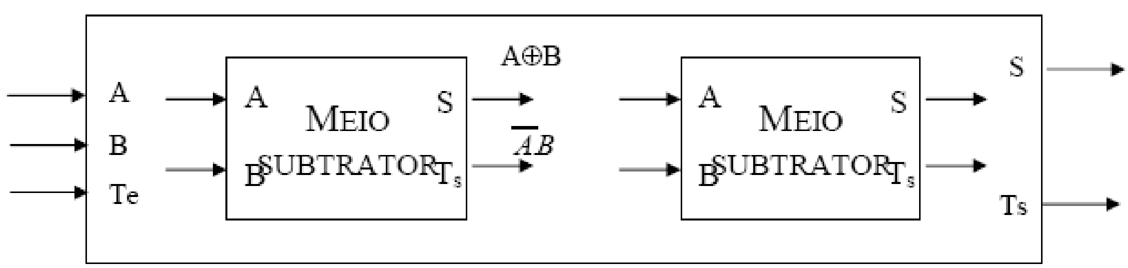
$$S = Te \oplus (A \oplus B)$$

$$\boxed{S = A \oplus B \oplus Te}$$

De "Te":

$$Ts = Te(\overline{A}.\overline{B} + A.B) + \overline{A}.B(\overline{Te} + Te)$$

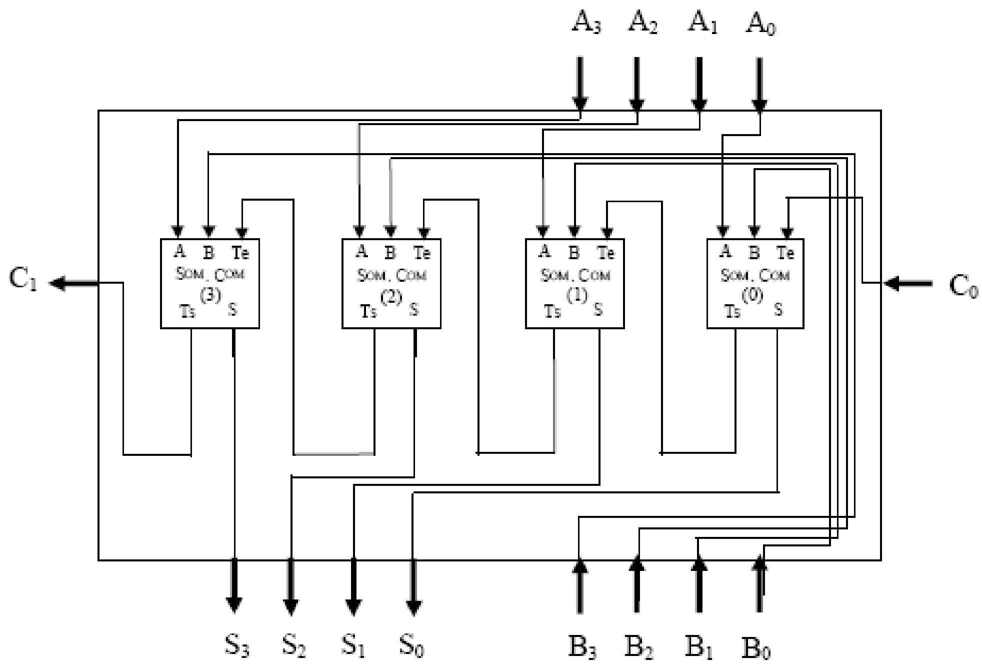
$$\boxed{Ts = Te.\overline{(A \oplus B)} + \overline{A}B}$$



(subtrator completo)

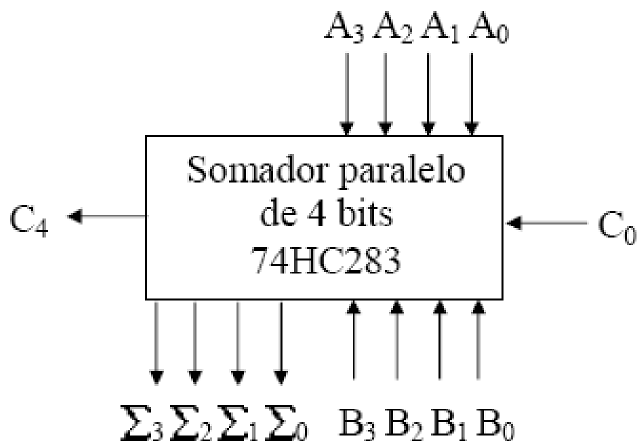
Somador paralelo integrado

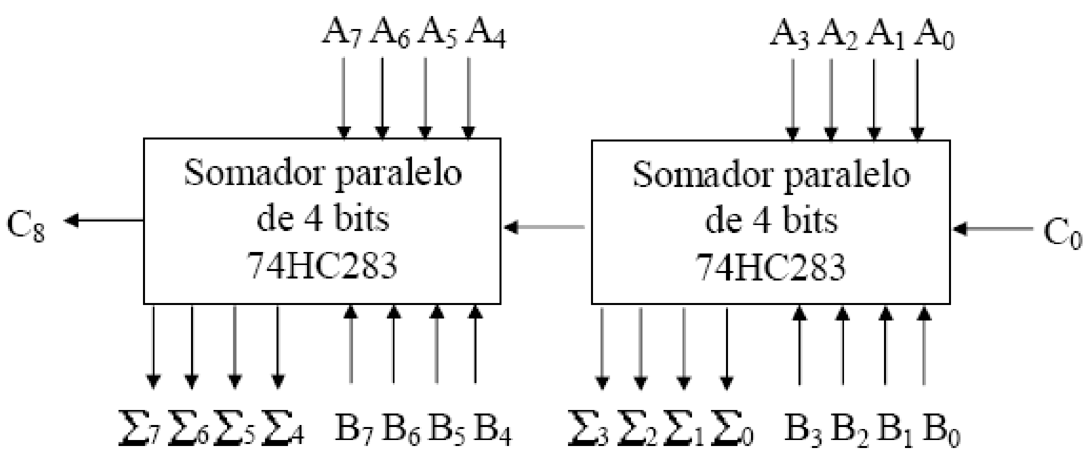
Dentre os diversos somadores paralelos disponíveis na forma de CIs (circuitos integrados), o mais comum é o de 4 bits. Um circuito de soma paralela integrada desse tipo é, em sua excência, bastante similar aos circuitos apresentados com somadores completos. Exemplificando:



Somador paralelo cascateado

Para realizar somas com números maiores que 4 bits utilizando o somador paralelo de 4 bits, podemos proceder o cascateamento de diversos CIs.



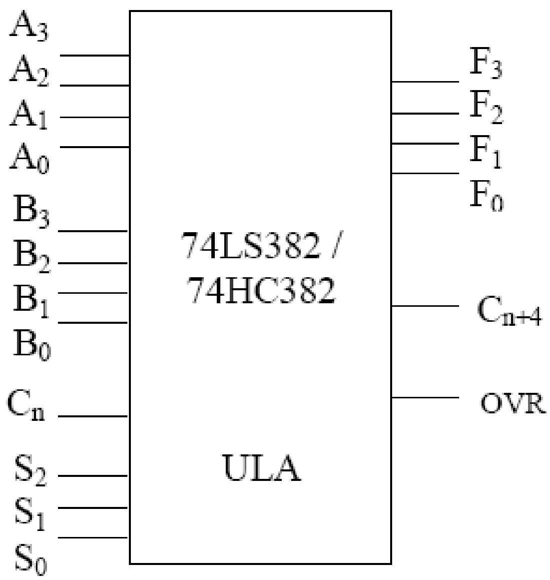


ULA (Unidade Lógico Aritmética)

Existem CIs disponíveis comercialmente que agrupam diversas das operações aritméticas em um único circuito, denominado ULA.

Embora não tenham a mesma capacidade das ULAs de processadores, tais circuitos podem ser encontrados desde formas bastante simples (com poucas operações) até circuitos bastante elaborados.

Um exemplo é apresentado abaixo:



S_2	S_1	S_0	**Operação**
0	0	0	CLEAR
0	0	1	B menos A
0	1	0	A menos B
0	1	1	A mais B
1	0	0	$A \oplus B$
1	0	1	A OR B
1	1	0	A.B
1	1	1	PRESET

A: número de entrada de 4 bits
B número de entrada de 4 bits
C_n: carry de entrada para a posição LSB
S: entradas de seleção de operação de três bits
F: número de saída de 4 bits
C_{n+4}¨carry de saída da posição MSB
OVR: indicador de ovserflow