ESCOLA DE ENGENHARIA DE SÃO CARLOS

Departamento de Engenharia Elétrica e de Computação

SEL 0412 Tecnologia Digital

Aula 8: Circuitos Lógicos Combinacionais_ Parte 1:

- Gerador de Paridade
- Comparador de Magnitude
- Representação de números binários com sinal
- Circuitos Aritméticos:
  - ✓ Meio Somador
  - ✓ Somador Completo
  - ✓ Subtrator
  - ✓ Multiplicadores
- Unidade Lógico e Aritmética (ULA)

Profa. Luiza Maria Romeiro codá

(luiza@sc.usp.br)

CIRCUITOS COMBINACIONAIS

CIRCUITOS COMBINACIONAIS

- são aqueles em que o sinal de saída depende única e exclusivamente das combinações dos sinais de entrada, não possuem nenhum tipo de memória, ou seja, as saídas não dependem de nenhum estado anterior do **circuito**;

CIRCUITOS COMBINACIONAIS COM FINALIDADE ESPECÍFICA

Podem ser elaborados tanto a partir de componentes discretos básicos (portas AND, OR, INV, etc) como obtidos totalmente na forma integrada CIs com tecnologia TTL/CMOS com função específica:

Exemplos :

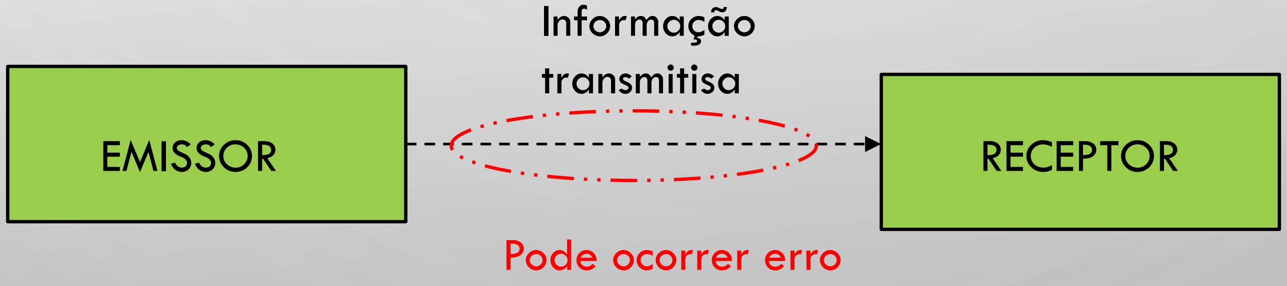
- Gerador de Paridade
- Comparadores
- Somadores
- Subtratores
- Multiplicadores
- Codificadores
- Decodificadores
- Multiplexadores
- Demultiplexadores
- ULA

GERADOR / VERIFICADOR DE PARIDADE

GERADOR / VERIFICADOR DE PARIDADE

Quando dados binários são transmitidos por um meio de transmissão qualquer (linha telefônica, RF etc.), erros de transmissão podem ocorrer. Uma das maneiras de detectar erros é a utilização do bit deparidade.

Processo de Comunicação



GERADOR / VERIFICADOR DE PARIDADE

Tipos de Paridade que um dado binário pode ter:

paridade par:

diz-se que um número binário tem paridade par quando o mesmo, independentemente do seu número de bits, possui um número PAR de ”uns” (1);

Ex.: 11000, 0011, 0000, 10100110

paridade ímpar:

diz-se que um número binário tem paridade ímpar quando o mesmo, independentemente do seu número de bits, possui um número ÍMPAR de ”uns” (1);

Ex.: 11010, 1011, 0001, 11100110

GERADOR / VERIFICADOR DE PARIDADE

- Gerador de Paridade: É um circuito que, na transmissão do sinal (Emissor), gera um bit para ser transmitido juntamente com o dado, tranformando esse dado em paridade Par ou Ímpar, de acordo com o desejado.

- Verificador de Paridade: É um circuito que, na recepção do sinal, ao receber o dado, gera um bit de paridade Par ou Ímpar, de acordo com a mesma paridade que foi enviado.

O mesmo circuito utilizado na Geração da paridade no Tranmissor é igual ao de verificação no Receptor:

GERADOR / VERIFICADOR DE PARIDADE

Dados				Paridade Par	Paridade ìmpar
I3	I2	I1	I0	P	P_I
0	0	0	0	0	1
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	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

Gerador de Paridade de 4 variáveis:

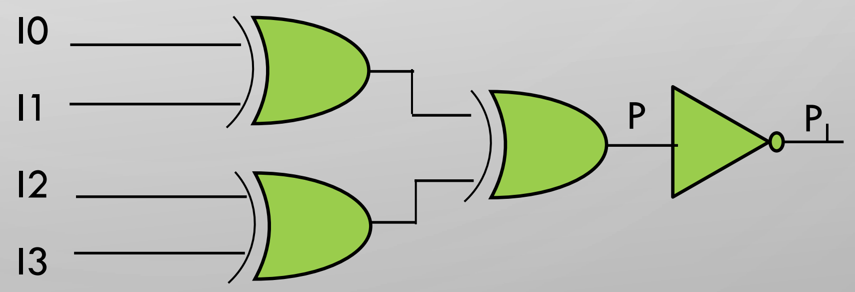
Paridade Par: O número de bits das 4 variáveis mais o sinal P deve conter número par de uns

Paridade Ímpar: O número de bits das 4 variáveis mais o sinal P deve conter número ímpar de uns

$$P = I0 \oplus I2 \oplus I2 \oplus I3$$

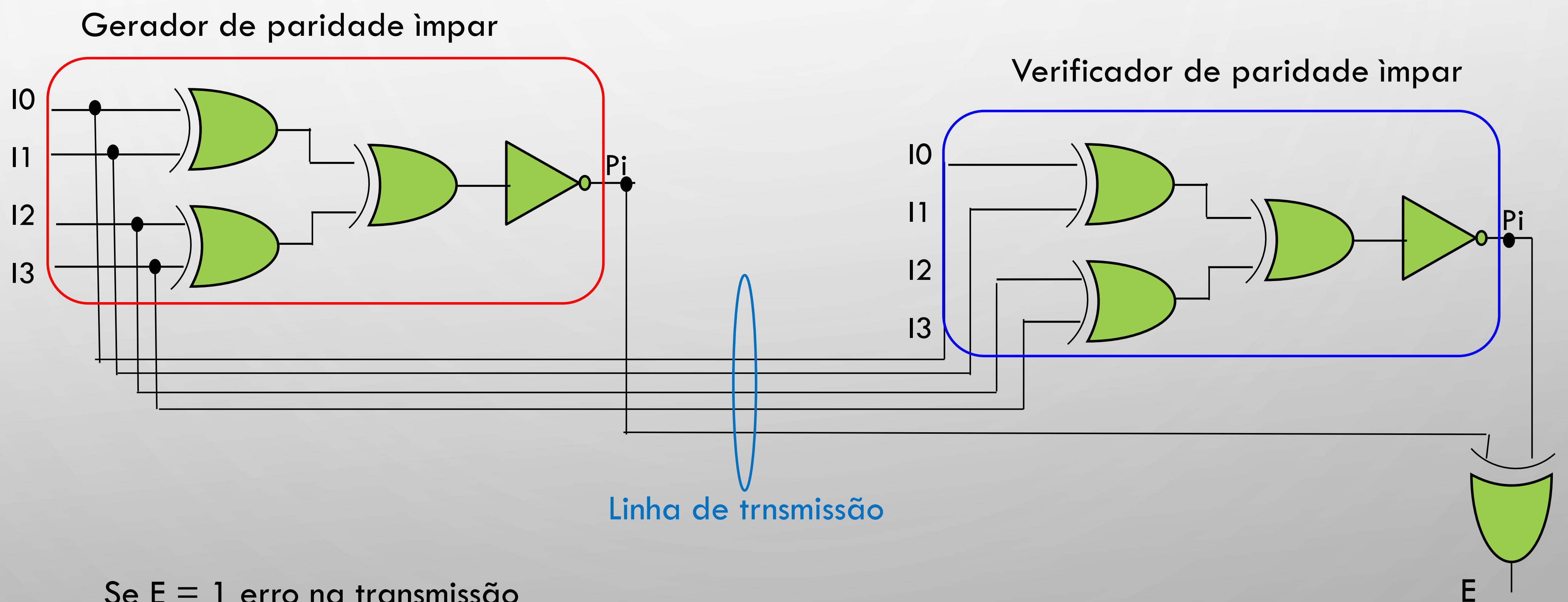
$$P_I = \overline{I0 \oplus I2 \oplus I2 \oplus I3}$$

Circuito:



GERADOR / VERIFICADOR DE PARIDADE

Transmissão de sinal com de 4 dados (variáveis) e bit de Paridade ímpar



Se E = 1 erro na transmissão

Se E = 0 não ocorreu erro na transmissão

GERADOR / VERIFICADOR DE PARIDADE

Exemplos de CIs Comerciais:

74180: Gerador e verificador de paridade Par/Ímpar de 9 bits

74280: Verificador/Gerador de paridade Par/Impar de 9 bits

COMPARADOR DE MAGNITUDE

COMPARADOR DE MAGNITUDE

O comparador é um circuito que realiza a comparação entre duas palavras de n bits (A e B), indicando na saída o relacionamento (do ponto de vista de valores) entre estas duas palavras.

Podendo apresentar saídas que sinalizam $A=B$, $A>B$, $A<B$, $A\geq B$, $A\leq B$ e $A\neq B$

COMPARADOR DE MAGNITUDE

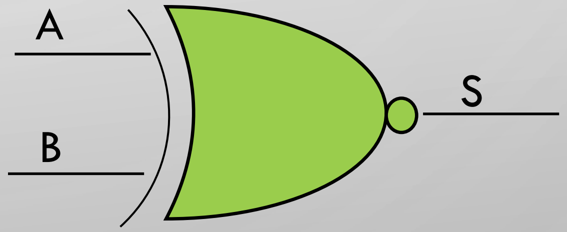
Exemplo: Comparador de dados (palavras) 1 Bit

Se A = B então S = 1
Se A≠ B então S = 0

$$S = \overline{\bar{A}.B + A.\bar{B}} = \overline{A \oplus B}$$

Tabela verdade

B	A	S
0	0	1
0	1	0
1	0	0
1	1	1



COMPARADOR DE MAGNITUDE

Exemplo1: Comparador de dados (palavras) 2 Bits

Sejam as entradas A = A1 A0 e B= B1 B0, implemente um circuito tal que:

Se A = B então S0 = 1

Se A≠ B então S0 = 0

Exemplo 1 Comparador de dados (palavras) 2 Bits

Resolução:

Tabela Verdade:

A1	A0	B1	B0	S1
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Se A = B então S = 1

Se A≠ B então S = 0

S

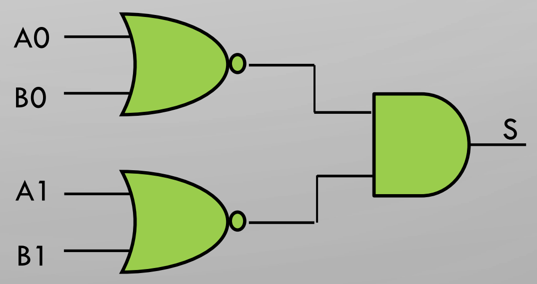
	$\overline{B1}.\overline{B0}$	$\overline{B1}.B0$	$B1.B0$	$B1.\overline{B0}$
$\overline{A1}.\overline{A0}$	1			
$\overline{A1}.A0$		1		
$A1.A0$			1	
$A1.\overline{A0}$				1

$$S1 = \overline{B1}.\overline{B0}.\overline{A1}.\overline{A0} + \overline{B1}.B0.\overline{A1}.A0 + B1.B0.A1.A0 + B1.\overline{B0}.A1.\overline{A0}$$

$$S1 = \overline{B1}.\overline{A1}\,(\overline{B0}.\overline{A0} + B0.A0) + B1.A1.(B0.A0 + \overline{B0}.\overline{A0})$$

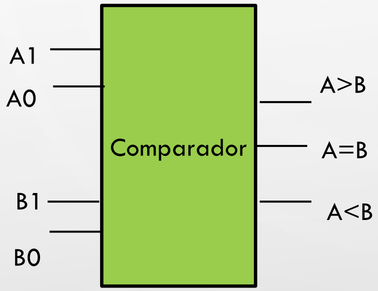
$$S1 = \overline{B1}.\overline{A1}\,(\overline{A0 \oplus B0}) + B1.A1.(\overline{A0 \oplus B0})$$

$$S1 = (\overline{A0 \oplus B0}).(\overline{A1 \oplus B1})$$



COMPARADOR

Exercício 1:Comparador de duas palavras de 2 bits com m saídas A =B, A>B e A<B



COMPARADOR

Exemplos de CIs Comerciais:

- 7485: Comparador de magnitude de 4 bits
- 74677: Comparador de endereço de 16 bits com enable
- 74678: Comparador de endereço de 16 bits com latch
- 74679: Comparador de endereço de 12 bits com latch
- 74680: Comparador de endereço de 12 bits com enable
- 74682: Comparador de magnitude de 8 bits
- 74683: Comparador de magnitude de 8 bits com coletor aberto
- 74684: Comparador de magnitude de 8 bits
- 74685: Comparador de magnitude de 8 bits com coletor aberto
- 74686: Comparador de magnitude de 8 bits com enable
- 74687: Comparador de magnitude de 8 bits com enable
- 74688: Comparador de magnitude de 8 bits
- 74689: Comparador de magnitude de 8 bits com coletor aberto

CIRCUITOS ARITMÉTICOS

CIRCUITOS ARITMÉTICOS:

Dentro do conjunto de circuitos combinacionais aplicados para finalidade específica, nos sistemas digitais destacam-se os circuitos aritméticos. São utilizados, principalmente, para construir a ULA (Unidade Lógica Aritmética) dos microprocessadores e, ainda, encontrados disponíveis em circuitos integrados comerciais. Alguns deles são:

- Meio somador
- Somador Completo
- Meio subtrator
- Subtrator
- Multiplicador

CIRCUITOS ARITMÉTICOS: MEIO SOMADOR

MEIO SOMADOR

Um meio somador de 1 bit soma o bit da palavra A com o da palavra B tendo resultados a soma(S) e um carry(Cout)

Palavra A de 1 bit ⇒ A

Palavra B de 1 bit ⇒ B +

$\boxed{C_{out}\ S}$ ⇒ Resultado (2 bits)

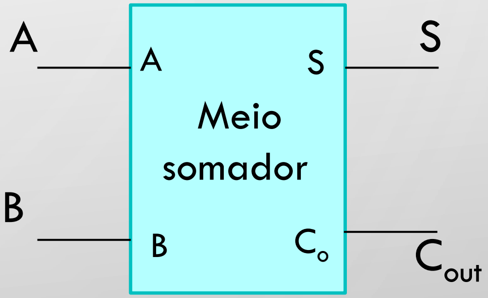
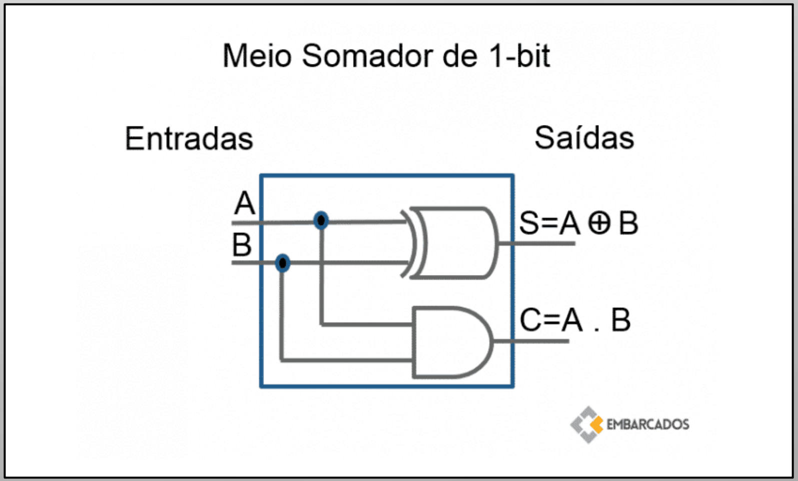


Tabela verdade

A	B	Cout	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



CIRCUITOS ARITMÉTICOS: SOMADOR COMPLETO

SOMADOR COMPLETO DE 1 BIT

Um somador completo de 1 bit soma o bit da palavra A com o bit da palavra B e com um carry inicial(Cin), e o resultado é obtido na saída S e o carry final em Cout

Carry inicial (‘0’ ou ‘1’) ⇒ C_{in}

Palavra A de 1 bit ⇒ A +

Palavra B de 1 bit ⇒ B +

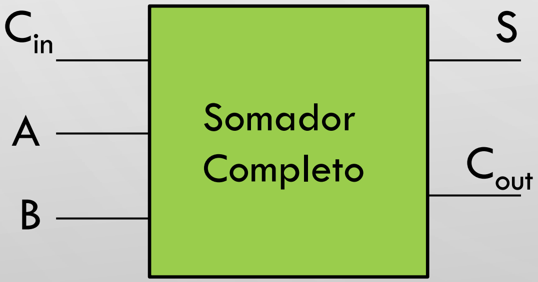
C_{out} S ⇒ Resultado (2 bits)

ENTRADAS			SAÍDAS		Resultado
Cin	B	A	Cout	S	decimal
0	0	0	0	0	0
0	0	1	0	1	1
0	1	0	0	1	1
0	1	1	1	0	2
1	0	0	0	1	1
1	0	1	1	0	2
1	1	0	1	0	2
1	1	1	1	1	3

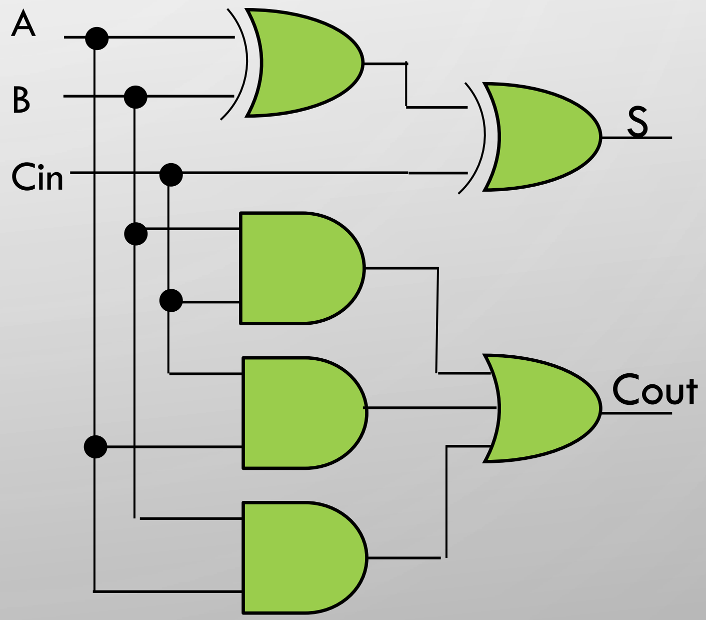
SOMADOR COMPLETO DE 1 BIT

$$S = A \oplus B \oplus C$$

$$Cout = A.B + A.Cin + B.Cin$$

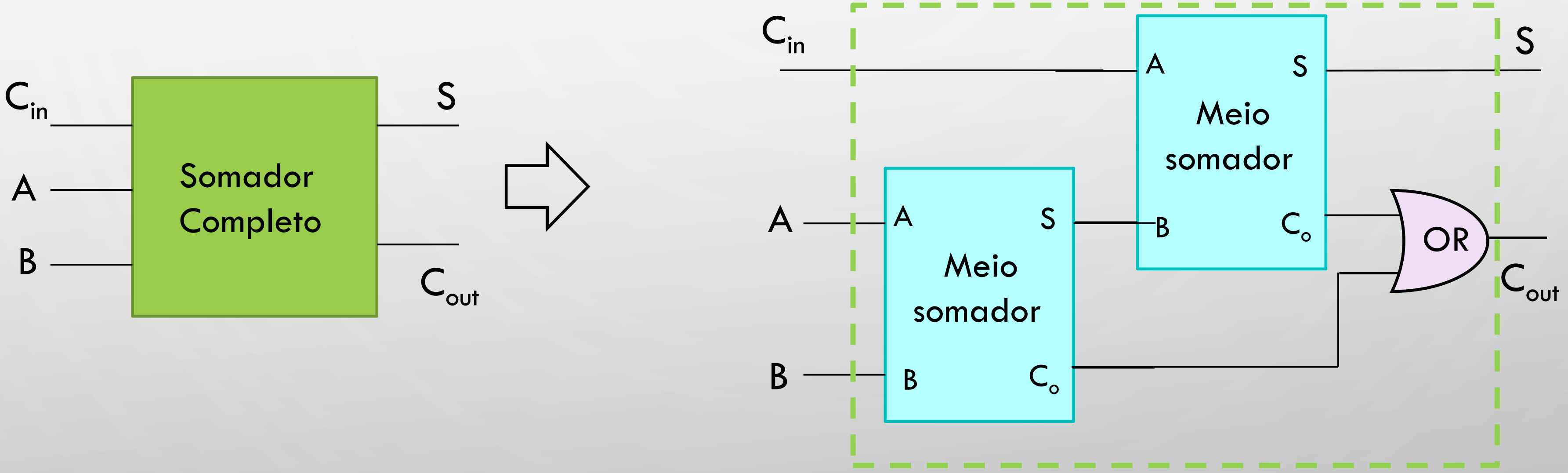


Circuito:



SOMADOR COMPLETO DE 1 BIT

Somador completo com circuitos meio somadores



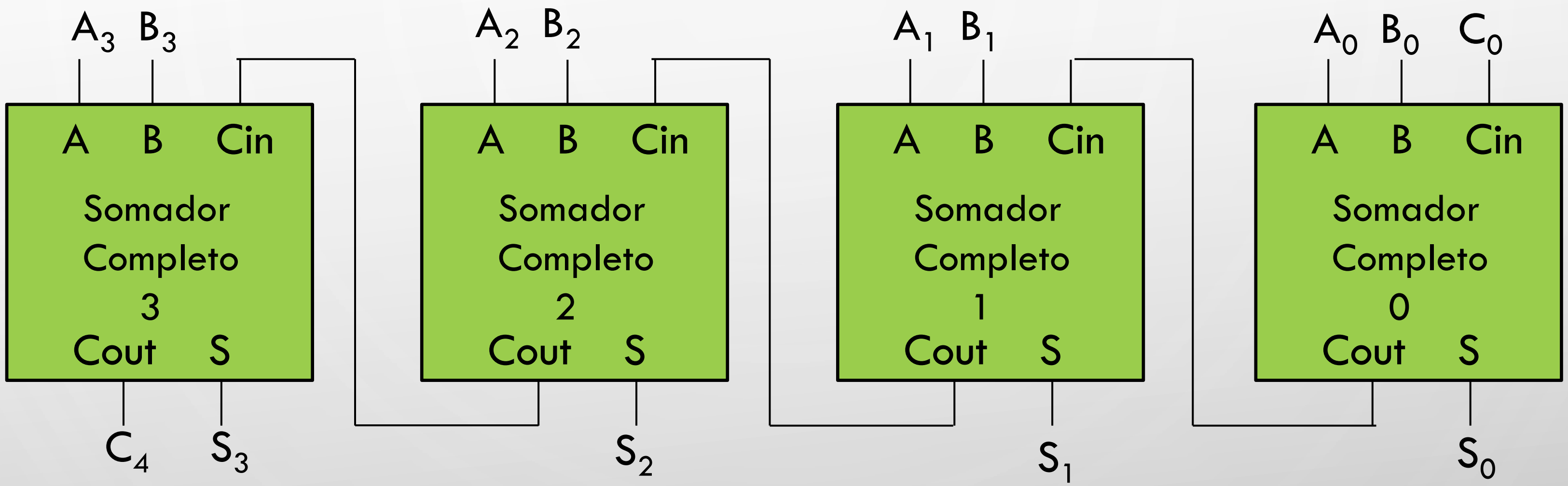
Carry inicial ('0' ou '1') ⇒ C_{in}

Palavra A de 1 bit ⇒ A +

Palavra B de 1 bit ⇒ B +

$\boxed{C_{out}\ S}$ ⇒ Resultado (2 bits)

SOMADOR COMPLETO PARALELO DE 4 BITS



Palavra A

Palavra B

$$
\begin{array}{r}
+C_0 \\
+\begin{bmatrix} A_4 & A_3 & A_2 & A_1 \\ B_4 & B_3 & B_2 & B_1 \end{bmatrix} \\
\hline
C4\ S_4\ S_3\ S_2\ S_1
\end{array}
$$

SOMADORES

Exemplos de CIs Comerciais:

- 7480: Somador completo com disparo
- 7482: Somador completo de 2 bits
- 7483: Somador completo de 4 bits
- 74283: Somador binário completo de 4 bits

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

- Em sistemas digitais as operações podem ser realizadas tanto com números negativos como positivos ;
- Existe a necessidade de representar o sinal do número (+ ou -);

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Representação de Sinal: Sinal-magnitude

Acrescenta-se um bit ao dado, denominado **bit de sinal**.

Exemplo: Um dado de 8 bits, 7 bits são usados para o valor absoluto (magnitude) e 1 bit para o sinal

b7	b6	b5	b4	b3	b2	b1	b0
0	0	1	1	0	1	0	0
MSB							LSB

$\Rightarrow +52_{10}$

Magnitude = 52_{10}

Bit de sinal

b7	b6	b5	b4	b3	b2	b1	b0
1	0	1	1	0	1	0	0
MSB							LSB

$\Rightarrow -52_{10}$

Magnitude = 52_{10}

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Representação de Sinal : Sinal-magnitude

Esse sistema de representação sinal-magnitude, embora seja bastante direto para fins de compreensão, a implementação desse sistema em circuitos lógicos é mais complexa do que a representação por complemento 2 , além do inconveniente de apresentar duas possíveis representações para o zero, + 0 e - 0Exemplo:

Bit de sinal

b7	b6	b5	b4	b3	b2	b1	b0	
0	0	0	0	0	0	0	0	⇨ $+0_{10}$
MSB							LSB	

Magnitude $= 0_{10}$

b7	b6	b5	b4	b3	b2	b1	b0	
1	0	0	0	0	0	0	0	⇨ -0_{10}
MSB							LSB	

Magnitude $= 0_{10}$

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Complemento de 1

Complemento de 1 de um número binário consiste em substituir cada 0 por 1 e cada 1 por 0, ou seja, seu complemento

Exemplo:

1	0	0	1	0	1	1

⇨ Número binário original

0	1	1	0	1	0	0

⇨ Complementa-se bit a bit

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Complemento de 2

Complemento de 2 de um número binário: Obtém-se o complemento 1 de um número binário e soma-se 1

Exemplo:

1	0	0	1	0	1	1	⇨ Número binário original = 75_{10}
0	1	1	0	1	0	0	⇨ Complementa-se bit a bit
					+	1	⇨ Soma-se 1 ao complemento de 1 de = 75_{10}
0	1	1	0	1	0	1	⇨ Complemento de 2 de = 75_{10}

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Representação de Sinal por Complemento de 2

- Se o número for positivo será representado da mesma forma que em sinal-magnitude, ou seja, na forma binária direta com um bit de sinal 0 no MSB;
- Se o número for negativo, a magnitude é representada na forma de complemento de 2 com um bit de sinal 1 no MSB

Exemplo:

0	1	0	0	1	0	1	1

$\Rightarrow + 75_{10}$

Bit de sinal

Binário $= 75_{10}$

1	0	1	1	0	1	0	1

$\Rightarrow - 75_{10}$

Bit de sinal

Complemento de 2 de 75_{10}

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Representação de Sinal por Complemento de 2

Transformar um binário positivo no seu negativo:

Obter o complemento de 2 do número positivo com sinal, o sinal negativo é gerado no processo:

Exemplo:

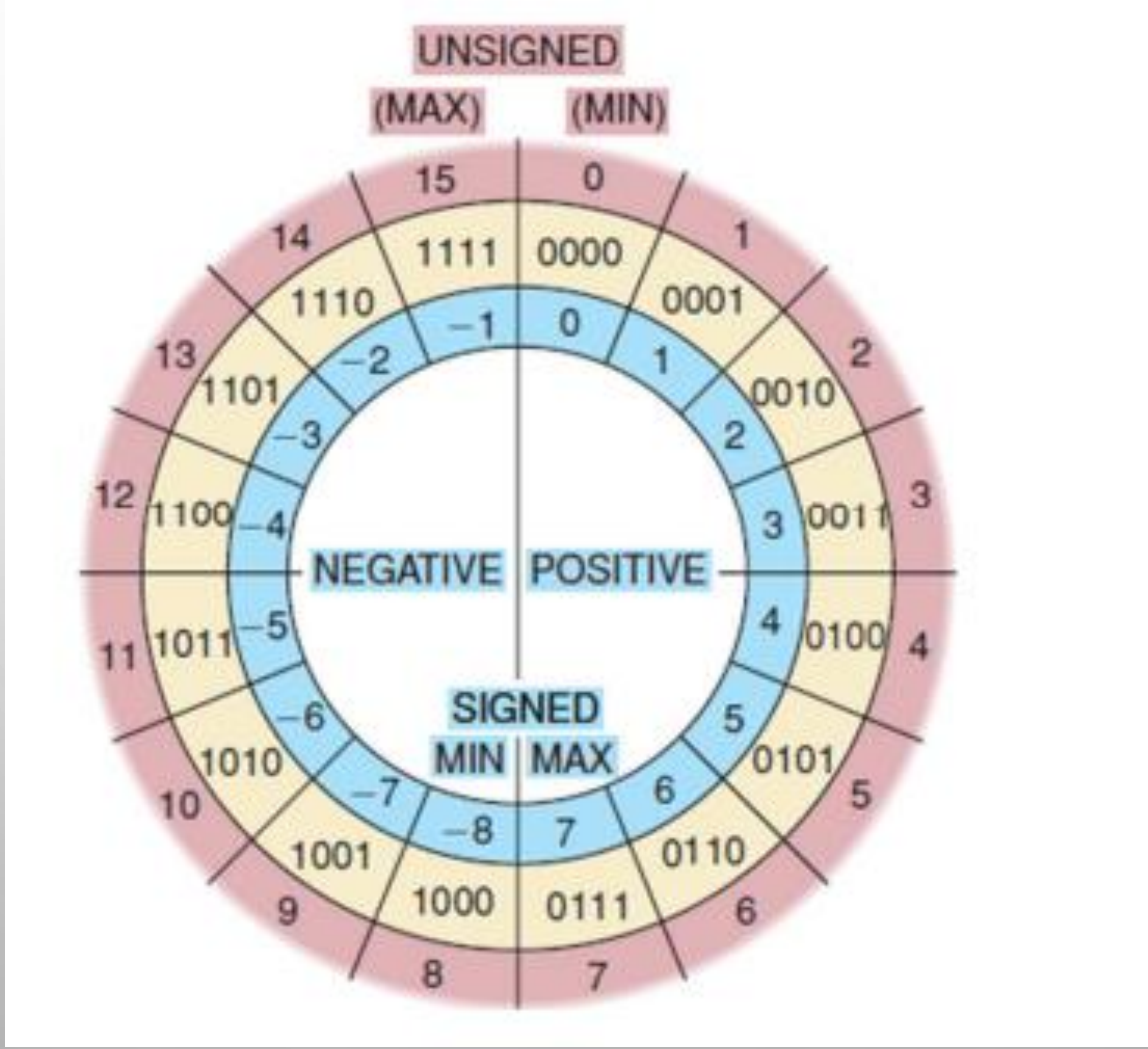
0	1	0	0	1	0	1	1	⇨	$+75_{10}$
1	0	1	1	0	1	0	0	⇨	Complemento de 1
							+ 1		Complemento de 2
1	0	1	1	0	1	0	1	⇨	-75_{10}

Bit de sinal

NÚMEROS NEGATIVOS EM COMPLEMENTO 2

DECIMAL	BINÁRIO EM COMPLEMENTO 2
+7	0111
+6	0110
+5	0101
+4	0100
+3	0011
+2	0010
+1	0001
0	0000
-1	1111
-2	1110
-3	1101
-4	1100
-5	1011
-6	1010
-7	1001
-8	1000

NÚMEROS NEGATIVOS EM COMPLEMENTO 2



Ref da figura : https://www.professores.uff.br/lbertini/wp-content/uploads/sites/108/2017/08/Capitulo-5-Aritmetica-Digital.pdf

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Adição e Subtração utilizando representação por Complemento de 2

1° caso: A > B ⇨ A positivo e B positivo

Exemplo:

$$\begin{array}{cc} & A \\ + & B \\ \hline & S \end{array}$$

	0	0	0	0	0	Carry
	0	1	0	0	0	1ª Parcela : $+8_{10}$
+	0	0	1	0	1	2ª Parcela : $+5_{10}$
	0	1	1	0	1	Soma : $+13_{10}$

Bit de sinal (positivo)

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Adição e Subtração utilizando representação por Complemento de 2

2° caso: A > B ⇨ A positivo e B negativo

Exemplo:

$$
\begin{array}{cc}
 & A \\
+ & (-B) \\
\hline
 & S
\end{array}
$$

⇨

Último Carry é desconsiderado

	$\boxed{1}$	1	0	0	0	Carry
	0	1	0	0	0	1ª Parcela : $+8_{10}$
+	1	1	0	1	1	2ª Parcela : -5_{10}
	0	0	0	1	1	Soma : $+3_{10}$

Bit de sinal (positivo)

	0	0	1	0	1	5
	1	1	0	1	0	Complemento 1 de 5
				+	1	
	1	1	0	1	1	Complemento 2 de 5 = -5

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Adição e Subtração utilizando representação Complemento de 2

3° caso: A < B ⇨ A positivo e B negativo

Exemplo:

$$\begin{array}{cc} & A \\ + & (-B) \\ \hline & S \end{array}$$

	0	0	0	0	0	Carry
	0	0	1	0	1	1ª Parcela : $+5_{10}$
+	1	1	0	0	0	2ª Parcela : -8_{10}
	1	1	1	0	1	Soma : -3_{10}

Bit de sinal (negativo)

8

0 1 0 0 0

1 0 1 1 1

+ 1

1 1 0 0 0

Complemento 2 de 8 = -8

Complemento 1 de 8

	0	0	0	1	1 = +3
	1	1	1	0	0 = complemento 1 de 3
				+	1
	1	1	1	0	1

Complemento 2 de 3 = -3

REPRESENTAÇÃO DE NÚMEROS COM SINAIS

Adição e Subtração utilizando representação Complemento de 2

4° caso: A < B ou A>B ⇨ A negativo e B negativo

Exemplo:

Último Carry é desconsiderado

	1	1	0	0	0	Carry
		1	1	0	1 1	1ª Parcela : -5_{10}
+		1	1	0	0 0	2ª Parcela : -8_{10}
		1	0	0	1 1	Soma : -3_{10}

$$+\ \frac{\begin{matrix}(-A)\\(-B)\end{matrix}}{S}$$

Bit de sinal

Que é o complemento 2 de 13

5

	0	0	1	0	1
	1	1	0	1	0
+					1
	1	1	0	1	1

Complemento 1 de 5

Complemento 2 de 5 = -5

13

	0	1	1	0	1
	1	0	0	1	0
+					1
	1	0	0	1	1

Complemento 1 de 13

Complemento 2 de 13 = -3

OVERFLOW ARITMÉTICO

- O overflow só pode ocorrer se os dois números somados tiverem o mesmo sinal
- O overflow ocorre quando os resultados excedem a faixa de representação.
- Na soma de números positivos não sinalizados, o último vai um (carry out) já significa overflow

Exemplo:

Último Carry é desconsiderado

		1	1	0	0	0	Carry
A			0	1	0	0	1	1ª Parcela : $+9_{10}$
+B	+		0	1	0	0	0	2ª Parcela : $+8_{10}$
S			1	0	0	0	1	Soma : -15_{10}

Bit de sinal

Magnitude incorreta

0	1	1	1	1	15
1	0	0	0	0	Complemento 1 de 15
			+	1	
1	0	0	0	1	Complemento 2 de 15 = -15

A + B deveria ser +17, porém como 17 não pode ser representado com 4 bits, o valor excede a faixa de representação

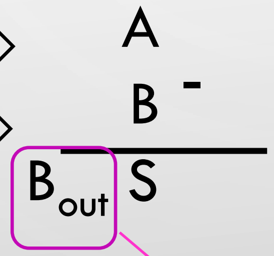
CIRCUITOS ARITMÉTICOS: MEIO SUBTRATOR

MEIO SUBTRATOR

Um meio somador de 1 bit subtrai o bit da palavra B do bit da palavra A tendo resultados a subtração (S) e um Borrow(Bout)

Palavra A de 1 bit

Palavra B de 1 bit



Empréstima (sinal)

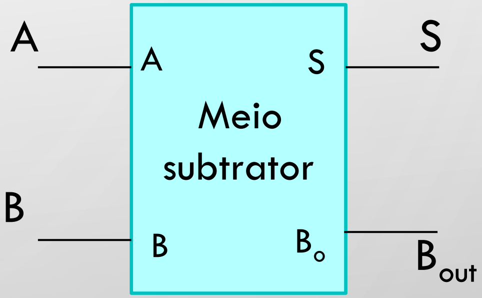
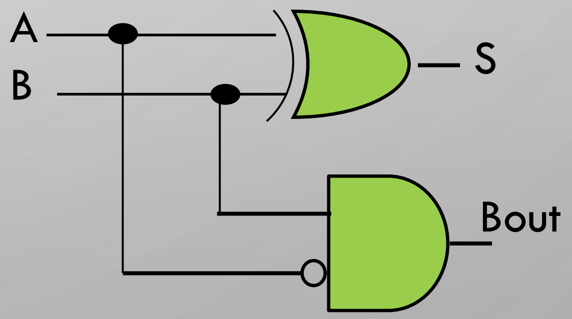


Tabela verdade

A	B	S	Bout
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

$$S = A \oplus B$$

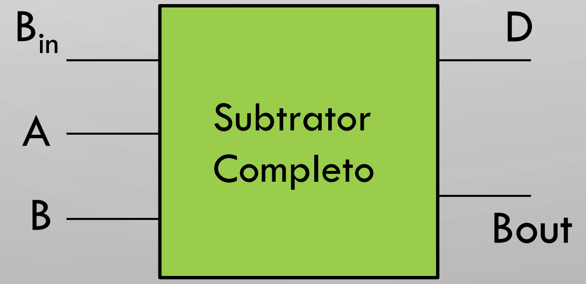
$$Bout = \bar{A}.B$$



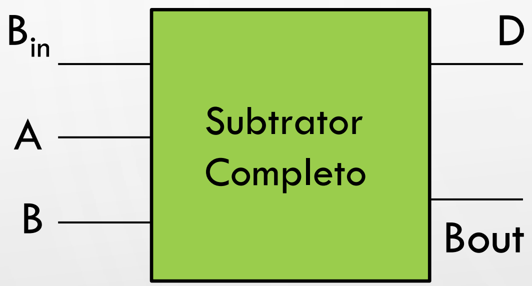
CIRCUITOS ARITMÉTICOS: SUBTRATOR COMPLETO

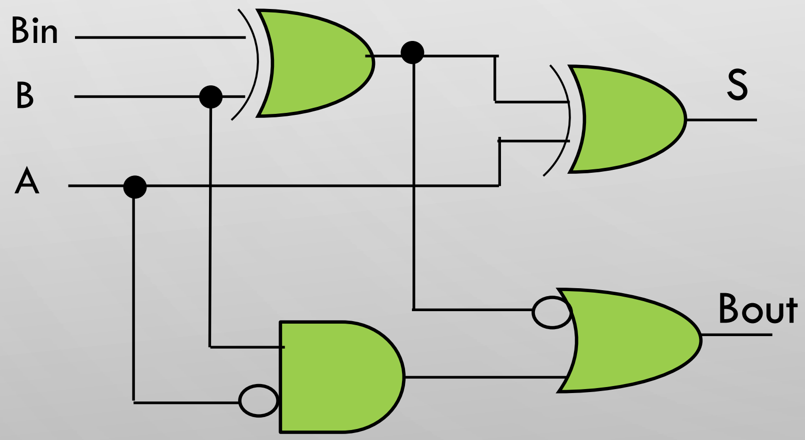
SUBTRATOR COMPLETO

- subtrator completo de 1 bit, é um circuito combinacional que realiza a subtração de dois bits, um é minuendo e outro é o subtraendo, levando em consideração o empréstimo do bit minuendo inferior adjacente anterior.
- Este circuito possui três entradas e duas saídas .
- As três entradas A, B e Bin denotam o minuendo, o subtraendo e o empréstimo anterior, respectivamente. As duas saídas, D e Bout, representam a diferença e a saída do empréstimo, respectivamente.



SUBTRATOR COMPLETO





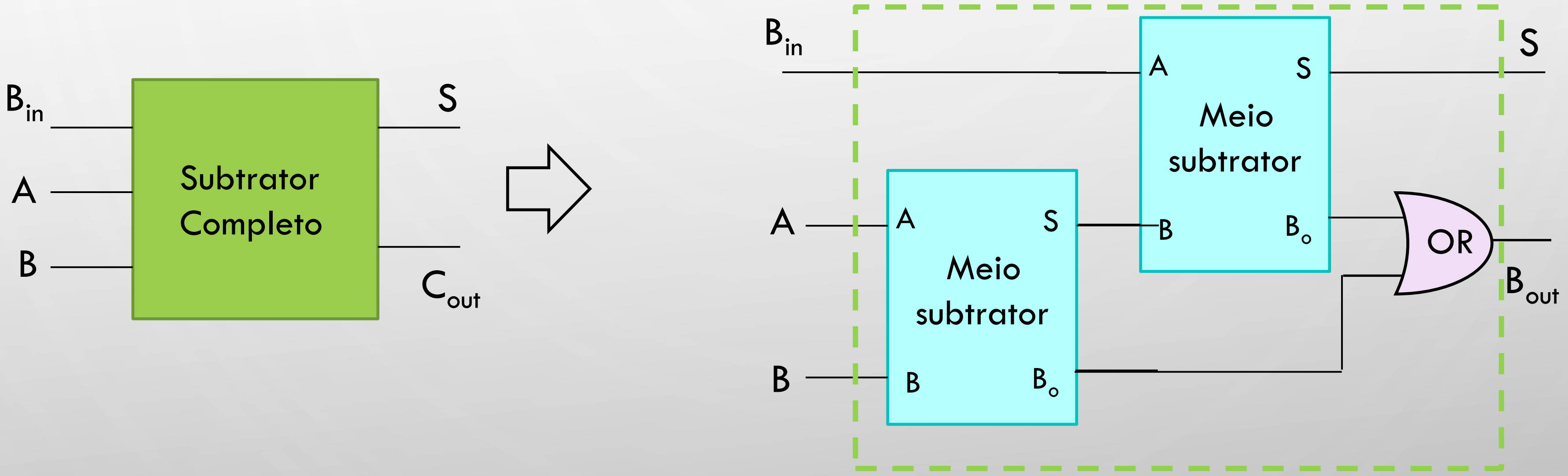
ENTRADAS			SAÍDAS		Resultado
A	B	Bin	D	Bout	
0	0	0	0	0	0
0	0	1	1	1	1
0	1	0	1	1	1
0	1	1	0	1	2
1	0	0	1	0	1
1	0	1	0	0	2
1	1	0	0	0	2
1	1	1	1	1	3

$$D = A \oplus B \oplus Bin$$

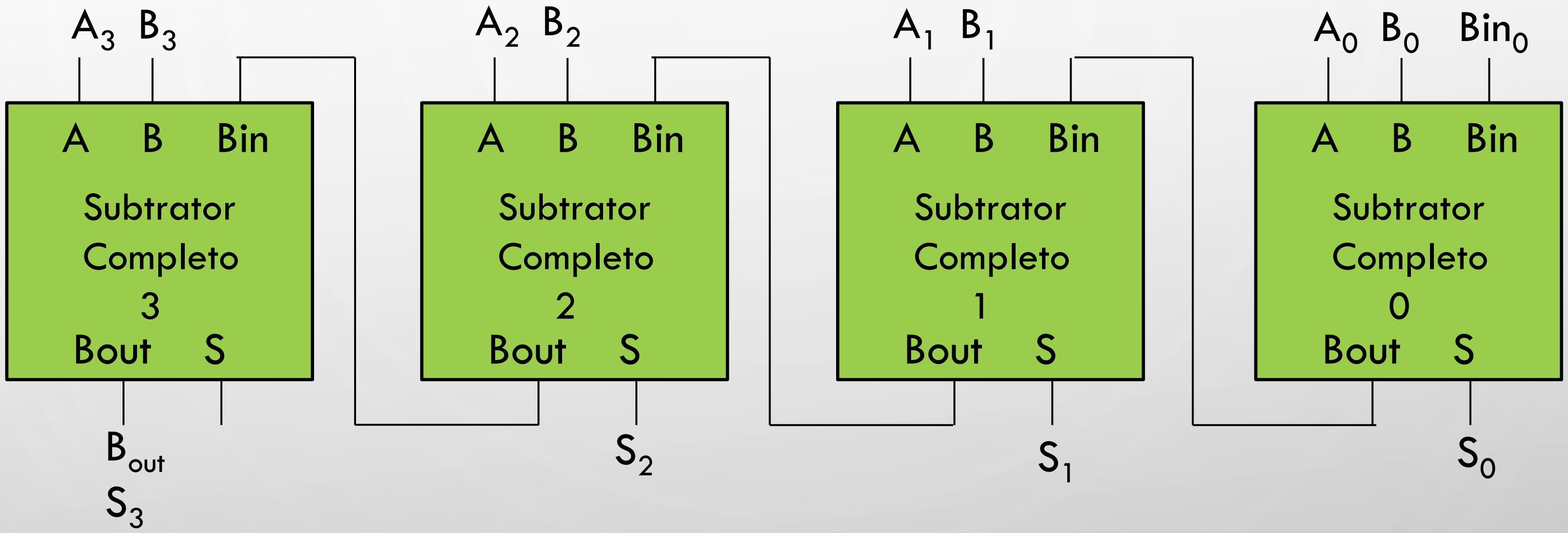
$$Bout = \bar{A}.B + \overline{Bin \oplus B}$$

SUBTRATOR COMPLETO DE 1 BIT

Subtrator completo com circuitos meio subtratores



SUBTRATOR COMPLETO PARALELO DE 4 BITS



Palavra A ⇒
Palavra B ⇒

$$
\begin{array}{r}
- \begin{bmatrix} A_4 & A_3 & A_2 & A_1 \\ B_4 & B_3 & B_2 & B_1 \end{bmatrix} \\
\hline
S_4 \; S_3 \; S_2 \; S_1 \\
Bout \qquad Bin
\end{array}
$$

SUBTRATOR COMPLETO PARALELO DE 4 BITS

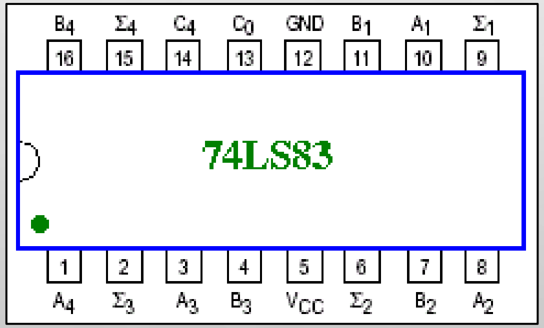
Exemplo 1:

Implementar um somador/subtrator Paralelo para 2 palavras de 4 bits (A e B) usando o CI 7483.

O circuito apresenta uma entrada de controle P que seleciona qdo P= 0 soma e quando P= 1 subtração.

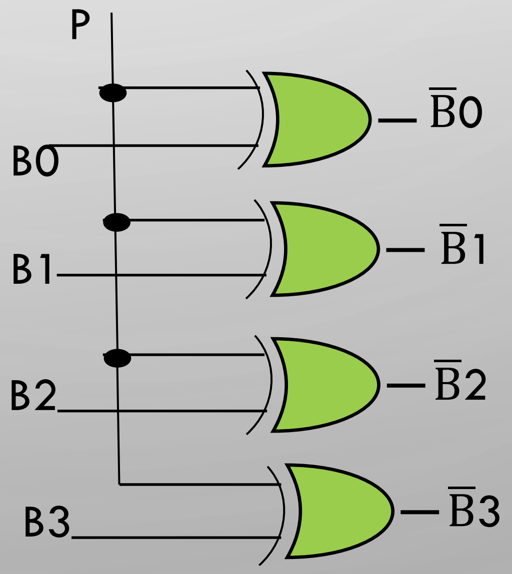
Para fazer a inversão utilize o circuito ou exclusivo como um inversor controlável.

Para simplificação do circuito a ser implementado, a subtração deve reproduzir os passos:



$$
\begin{array}{r}
+C_0 \\
\left[\begin{array}{cccc} A_4 & A_3 & A_2 & A_1 \end{array}\right. \\
+\left.\begin{array}{cccc} B_4 & B_3 & B_2 & B_1 \end{array}\right] \\
\hline
C4 \;\; \boldsymbol{\Sigma}_4\, \boldsymbol{\Sigma}_3\, \boldsymbol{\Sigma}_2\, \boldsymbol{\Sigma}_1
\end{array}
$$

Inversor Controlável



Quando P = 0, S = B

Quando P= 1, S=$\overline{B}$

Tabela verdade:

P	B	S
0	0	0
0	1	1
1	0	1
1	1	0

SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1:

A > B   A = 8 e B = 2 então 8 – 2 = +6

A (1000)b
B (0010)b Inverte B

$$
\begin{array}{r} (1\ 0\ 0\ 0)b \\ +\ (1\ 1\ 0\ 1)b \\ \hline \boxed{1}\ 0\ 1\ 0\ 1 \end{array}
$$

Resultado da Soma com carry = 1
Para obter o Resultado da Subtração Correto:
Soma-se 1 ao resultado da Soma, então:

$$
\begin{array}{r} 1\ 0\ 1\ 0\ 1 \\ +1 \\ \hline \end{array}
$$

$\boxed{(1\ 0\ 1\ 1\ 0)_b = 6_{10}}$

A < B   A = 2 e B = 8 então 2 – 8 = -6

A (0010)b
B (1000)b Inverte B

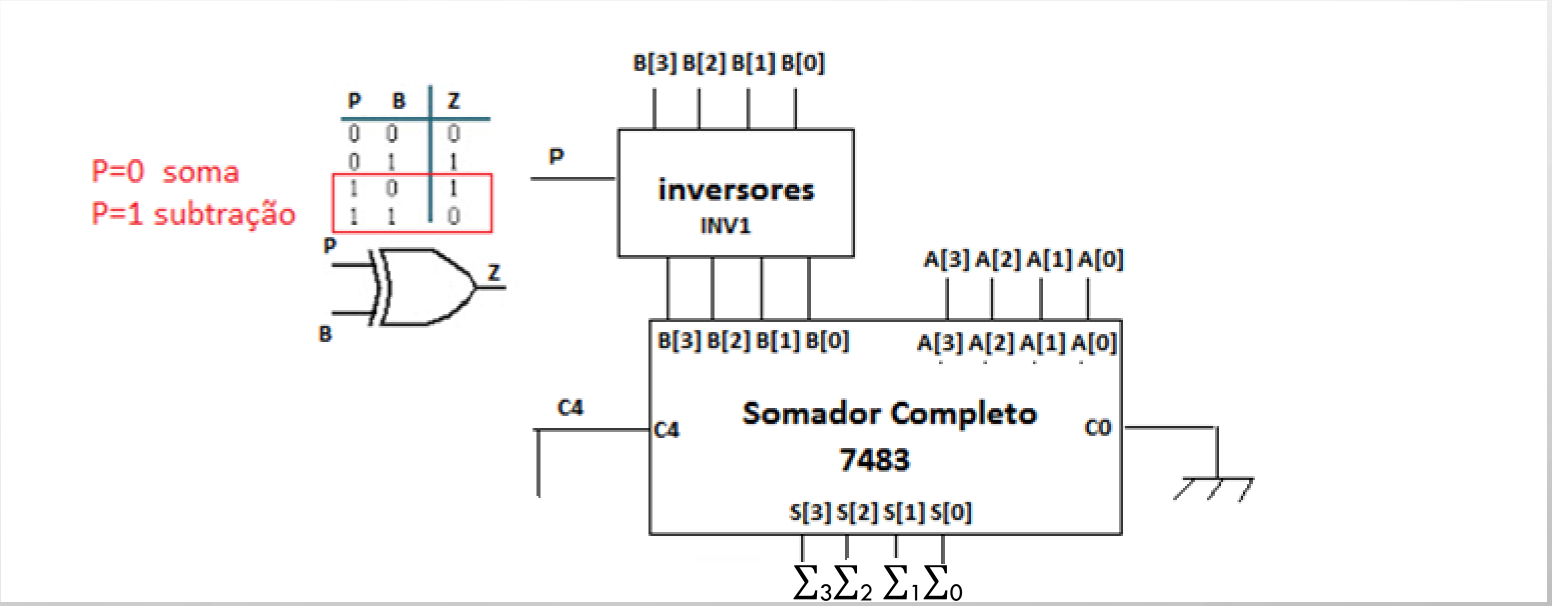
$$
\begin{array}{r} (0\ 0\ 1\ 0)b \\ +\ (0\ 1\ 1\ 1)b \\ \hline \boxed{0}\ 1\ 0\ 0\ 1 \end{array}
$$

Resultado da Soma com carry = 0
Para obter o Resultado da Subtração Correto:
Complementar o resultado da Soma (sem o carry),
então: 0 1 0 0 1 invertido $\boxed{(0\ 0\ 1\ 1\ 0)_b = 6_{10}}$

SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1:

1° Estágio da montagem:



- Se P = 0 o inversor controlado não inverte as entradas B e na saída do somador 7483 tem o resultado da soma da palavra A com a palavra B;
- Se P = 1 o inversor controlado inverte as entradas B e a saída do somador 7483 tem o resultado da soma da palavra A com a palavra B invertida;

SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1:

2° Estágio da montagem:

Verificação do carry do somador:
Se Carry = 1 então soma-se 1 à saída do 7483
Se Carry = 0, deve-se inverter a saída apenas se for subtração

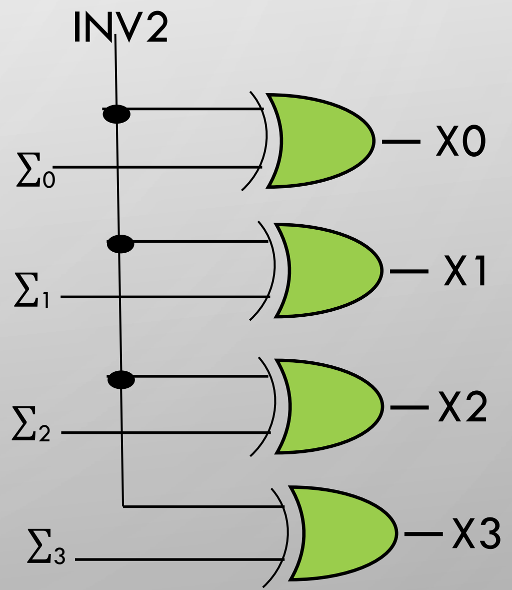
Tabela verdade para gerar sinal para o 2° conjunto de inversores:

P	C4	INV2
0	0	0
0	1	0
1	0	1
1	1	0

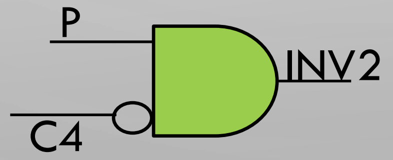
subtração

P= 0 é Soma, independente do valor de C4, INV2 = 0

P=1 e Carry = 0 deve-se inverter a saída



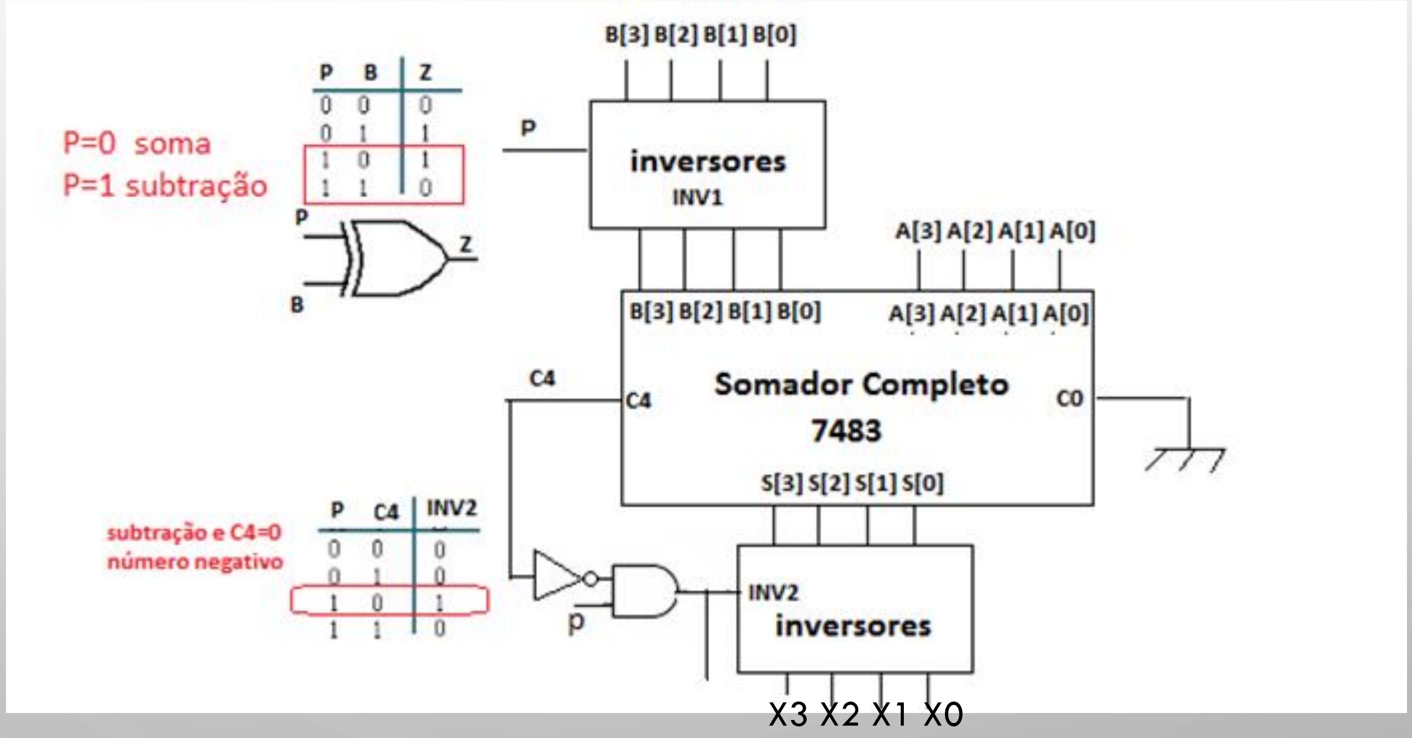
$INV2 = P.\overline{C4}$



SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1:

1° e 2° Estágios da montagem:



SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1:

2° Estágio da montagem:

Verificação do carry do somador:
Se Carry = 1 então soma-se 1 à saída do 7483

Tabela verdade para gerar sinal 1 na entrada C0 do 2° somador:

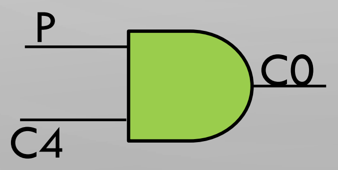
P	C4	C0
0	0	0
0	1	0
1	0	0
1	1	1

subtração

P= 0 é Soma, independente do valor de C4 , C0 =0

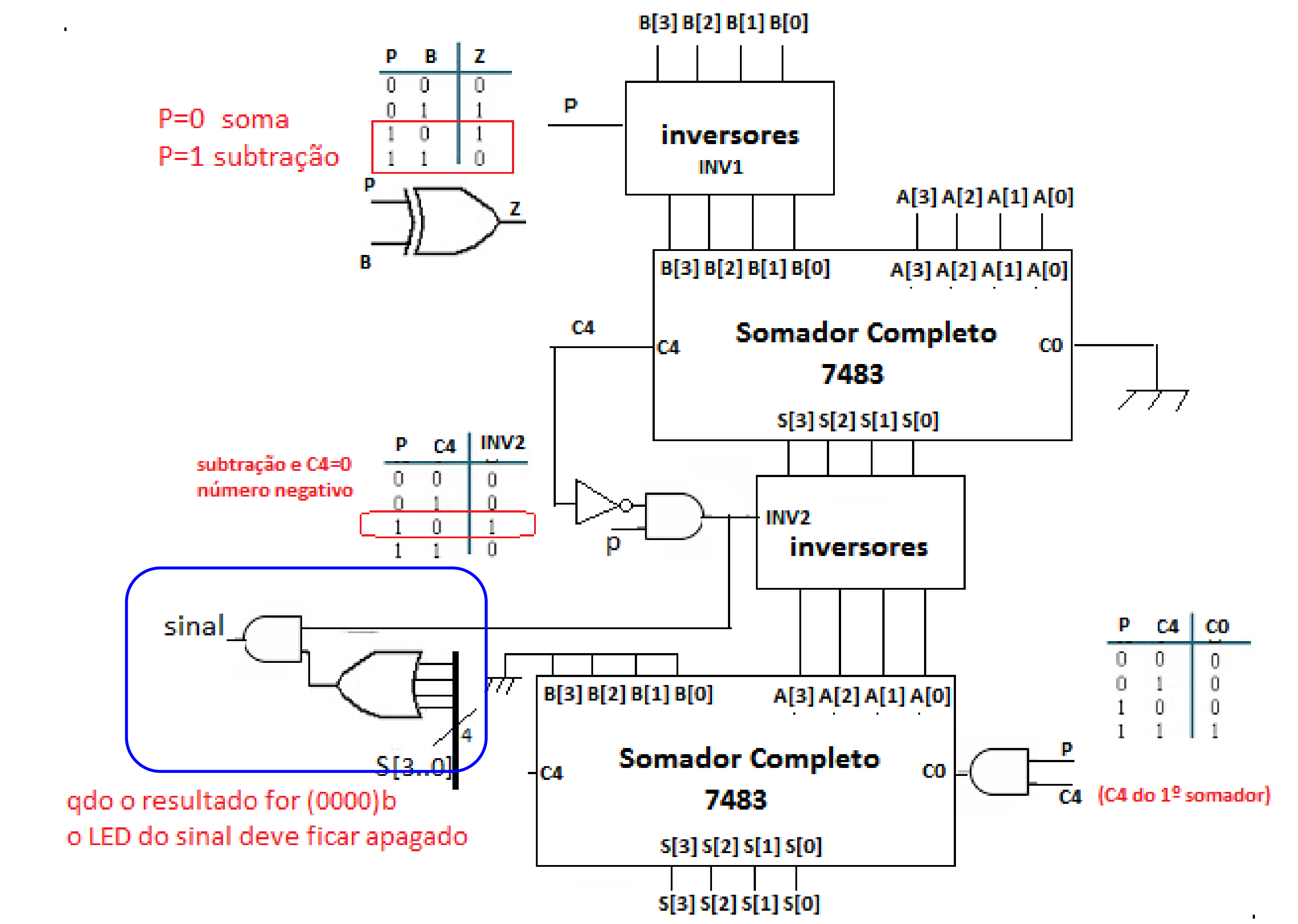
P=1 e Carry = 1 deve-se somar 1

$$C0 = P.C4$$



SUBTRATOR COMPLETO PARALELO DE 4 BITS

Exemplo 1: Circuito Final:



CIRCUITOS ARITMÉTICOS: MULTIPLICADORES

MULTIPLICADORES

multiplicação binária:

0 x 0 = 0
0 x 1 = 0
1 x 0 = 0
1 x 1 = 1

- A operação de multiplicação binária é similar à operação AND;
- A multiplicação de duas palavras com N bits resulta em uma palavra de (2 x N) bits caso tenha o carry mais significativo e ((2 x N) – 1) bits caso não tenha o carry mais significativo

MULTIPLICADORES

Multiplicador de 2 x 2

multiplicador de 2 palavras binárias de 2 bits que resulta em uma palavra de 4 bits (caso seja considerado o carry mais significativo (C3)).

		A_1	A_0
	X	B_1	B_0
C3	C2	$A_1.B_0$	$A_0.B_0$
	$A_1.B_1$	$A_0.B_1$	
P_3	P_2	P_1	P_0

AND

OR

$P_0 = A_0 \cdot B_0$

$P_1 = A_1 \cdot B_0 + A_0 \cdot B_1$

$P_2 = A_1 \cdot B_1 + C_2$

$P_3 = C_3$

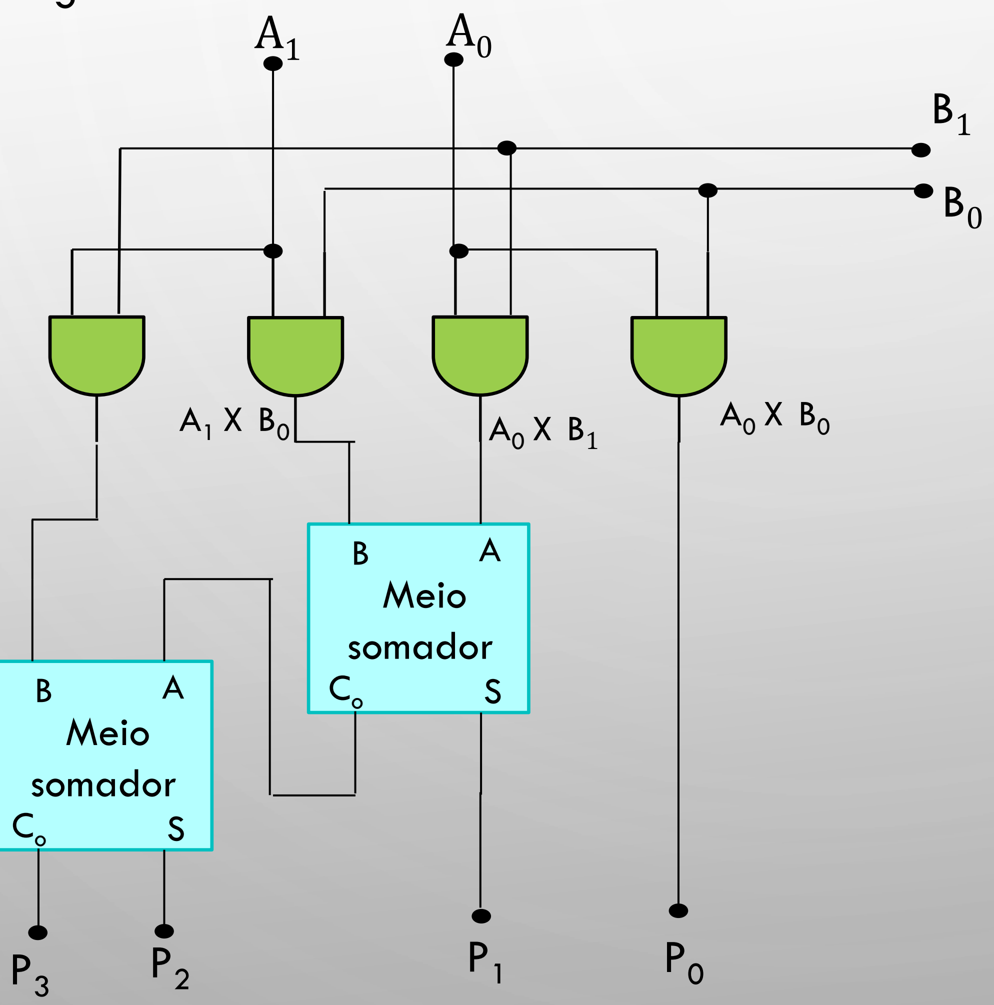
Exemplo numérico:

		1	1
	X	1	0
0	0	0	0
	1	1	
0	1	1	0

MULTIPLICADORES

Multiplicador de 2 x 2

Circuito Lógico:



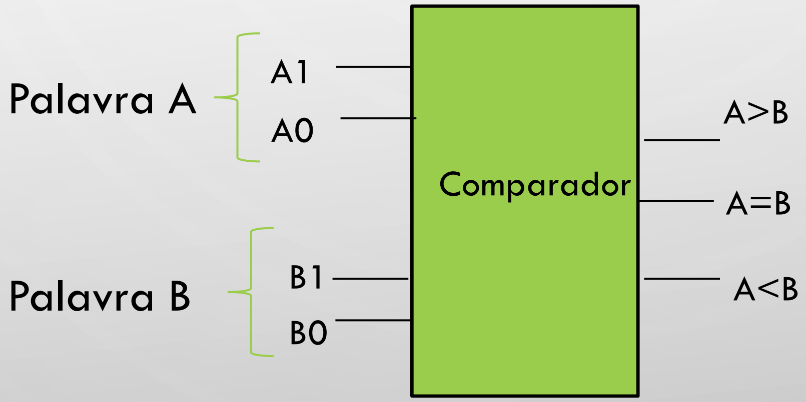
$P_0 = A_0 X B_0$

$P_1 = A_1 X B_0 + A_0 X B_1$

$P_2 = A_1 X B_1 + C_2$

$P_3 = C_3$

Exercício 1: Comparador de duas palavras de 2 bits com saídas A =B, A>B e A<B



Exercício N°2 : Implemente um circuito lógico com 4 variáveis de entrada (A B C D) e 4 saídas (S3 S2 S1 S0) tal que:

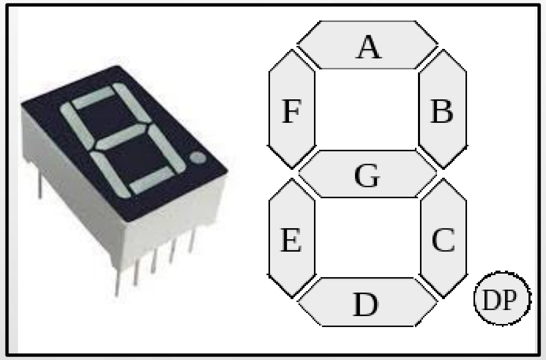
- quando a entrada for $\leq (1001)b$ a saída seja igual à entrada;
- quando a entrada for $>$ (1001)b a saída seja igual à entrada somada ao valor (0110)b .

Mostre o resultado em um display de 7 segmentos catodo comum, e o overflow ligue em um LED na configuração anodo comum. Utilize portas lógicas básicas (se necessário) e CIs comerciais com aplicação específica.

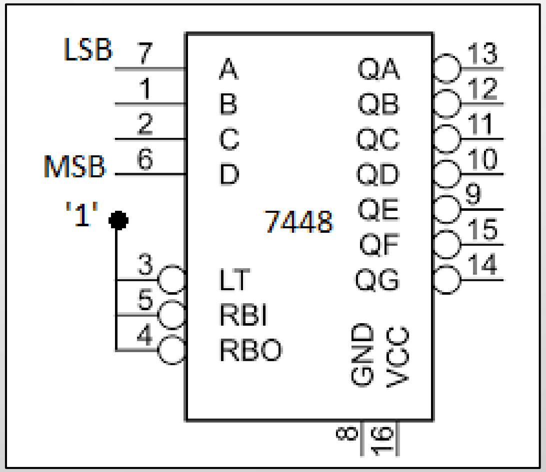
- **Exercício N°2 :**
- quando a entrada for $\leq (1001)b$ a saída seja igual à entrada;
- quando a entrada for $> (1001)b$ a saída seja igual à entrada somada ao valor (0110)b .

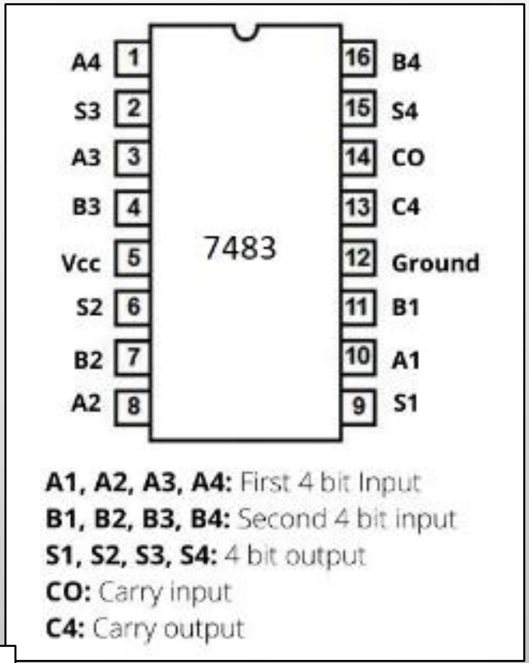
CIs comerciais a serem utilizados:

Display catodo comum



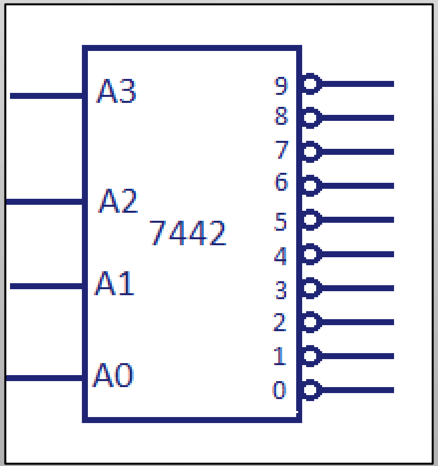
Decodificador BCD para display 7 segmentos catodo comum



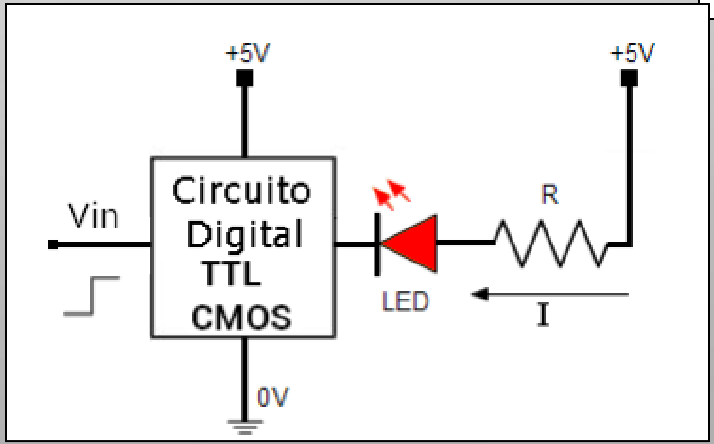


Somador completo
De 2 palavras de 4 bits

Decodificador BCD para Decimal

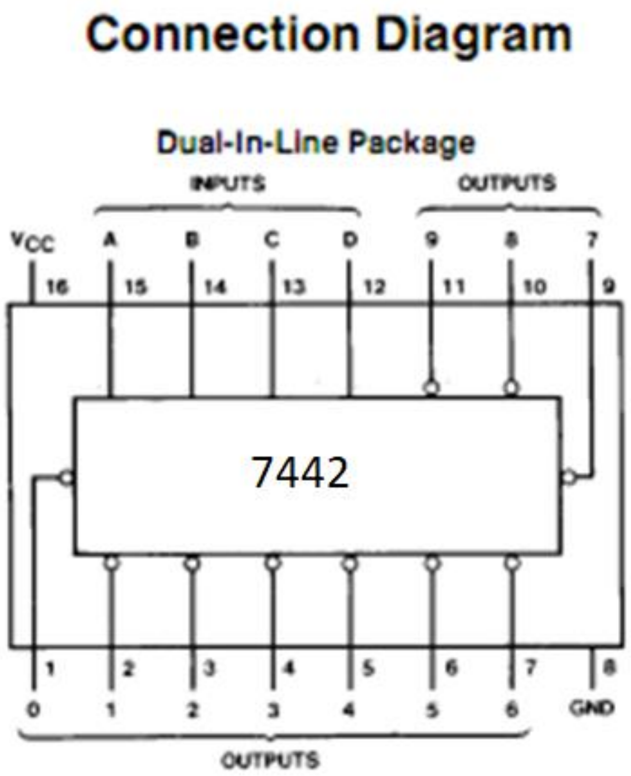


LED configuração anodo ciomum



Exercício N°2 : (continuação)

Decodificador BCD para Decimal



Function Table

No.	BCD Inputs				Decimal Outputs									
	D	C	B	A	0	1	2	3	4	5	6	7	8	9
0	L	L	L	L	L	H	H	H	H	H	H	H	H	H
1	L	L	L	H	H	L	H	H	H	H	H	H	H	H
2	L	L	H	L	H	H	L	H	H	H	H	H	H	H
3	L	L	H	H	H	H	H	L	H	H	H	H	H	H
4	L	H	L	L	H	H	H	H	L	H	H	H	H	H
5	L	H	L	H	H	H	H	H	H	L	H	H	H	H
6	L	H	H	L	H	H	H	H	H	H	L	H	H	H
7	L	H	H	H	H	H	H	H	H	H	H	L	H	H
8	H	L	L	L	H	H	H	H	H	H	H	H	L	H
9	H	L	L	H	H	H	H	H	H	H	H	H	H	L
INVALID	H	L	H	L	H	H	H	H	H	H	H	H	H	H
	H	L	H	H	H	H	H	H	H	H	H	H	H	H
	H	H	L	L	H	H	H	H	H	H	H	H	H	H
	H	H	L	H	H	H	H	H	H	H	H	H	H	H
	H	H	H	L	H	H	H	H	H	H	H	H	H	H
	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = High Level
L = Low Level

Sai zero nas saídas decimais

FIM