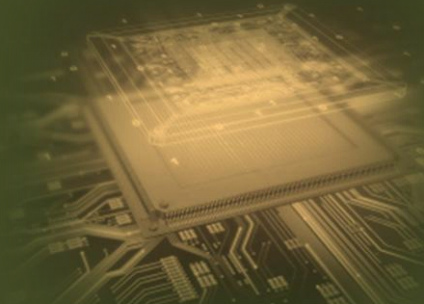


Sistemas Digitais



- Circuitos Aritméticos

Monitoria SD 2011.2

Daniel Alexandro/Reniê Delgado/Vanessa Ogg

Circuitos Aritméticos

- Circuitos Aritméticos são aqueles que realizam operações aritméticas sobre números binários;
- O Circuito Aritmético mais simples é o que soma números de apenas 1 bit;
- Os Circuitos Aritméticos são fundamentais na construção de um Computador;
- Iremos mostrar neste momento quatro tipos de Circuitos Aritméticos: **Somador**, **Subtrator**, **Multiplicador** e **Divisor**.

Representação de Números com Sinal

- Para representarmos números binários com sinal, é necessário acrescentarmos mais um bit a frente da magnitude do número (módulo do número);
- Quando o bit de sinal é 0, significa que o número é positivo. Quando o bit de sinal é 1, significa que o número é negativo.
- Veja os exemplos abaixo:

10011 -> Número 19 em decimal

Agora considerando o bit de sinal, temos:

010011 -> Número 19 (positivo)

110011 -> ATENÇÃO! Número 13 (negativo)

O Número 19 (negativo) em binário seria: **101101**

- Nos casos onde o número é negativo, é necessário realizar o **Complemento de 2** para saber qual é a magnitude do número. Esta técnica será explicada nos próximos slides!

Meio Somador

- As regras básicas para adição binária são:

1. $0 + 0 = 0$

2. $0 + 1 = 1$

3. $1 + 0 = 1$

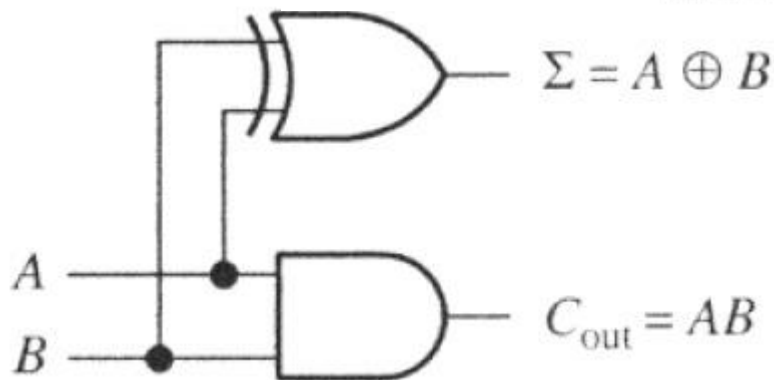
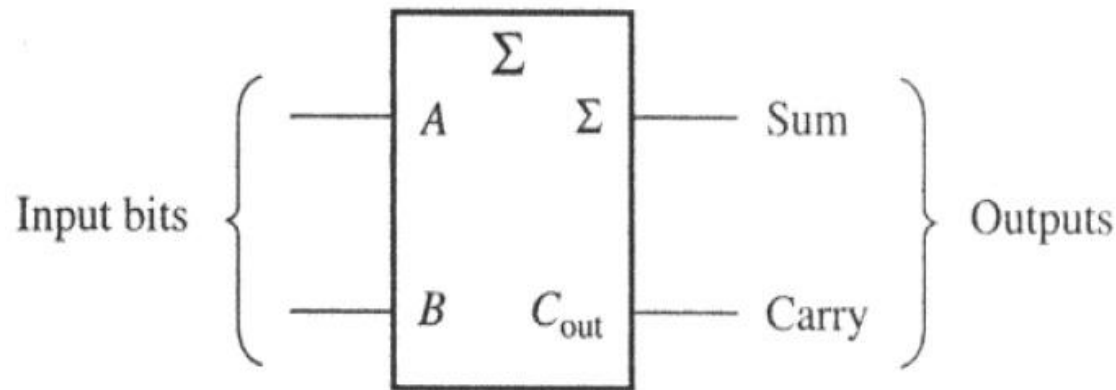
4. $1 + 1 = 10$

OBS.: O “1” no último resultado é o “vai-um” (*carry*) gerado por ter sido esgotada a capacidade de contagem. O *carry* deve ser acrescentado à soma dos bits imediatamente mais significativos à esquerda daqueles que deram origem ao *carry*.

- Estas operações são realizadas por um circuito lógico denominado **Meio Somador (Half Adder)**

Meio Somador

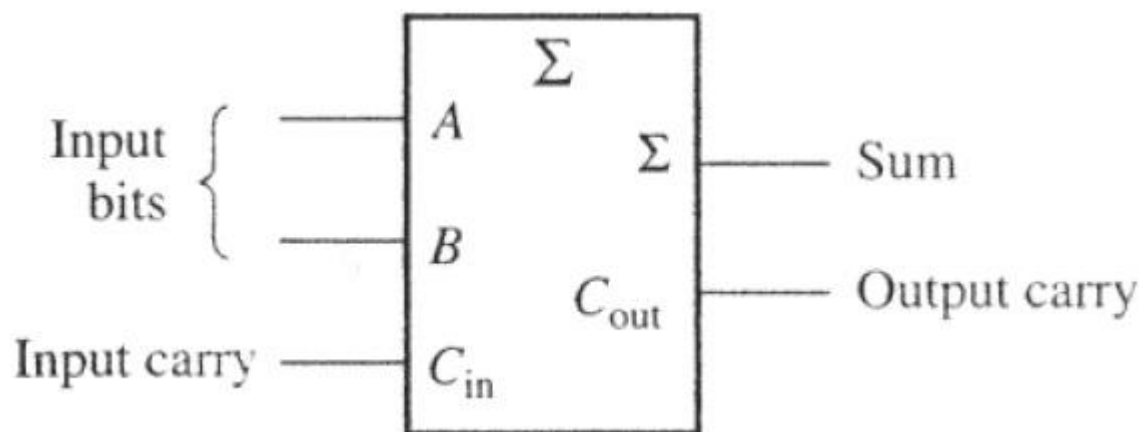
- Um Meio Somador recebe dois bits de entrada A e B e produz dois bits de saída: o **Bit de Soma** ($\Sigma = A + B$) e o **Bit de Carry** (C_{out}). Observe abaixo o Símbolo Lógico, a Tabela Verdade e o Diagrama Lógico de um Meio Somador:



A	B	C_{out}	Σ
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Somador Completo

- Um **Somador Completo (Full Adder)** recebe três bits de entrada A, B e C_{in} (o último correspondendo ao eventual *carry* gerado na operação com bits menos significativos) e produz dois bits de saída: o **Bit de Soma** ($\Sigma = A + B$) e o **Bit de Carry** (C_{out}). Observe abaixo o Símbolo Lógico do Somador Completo e nos slides posteriores a Tabela Verdade (com suas devidas simplificações) e o Diagrama Lógico:



Somador Completo (Tabela Verdade)

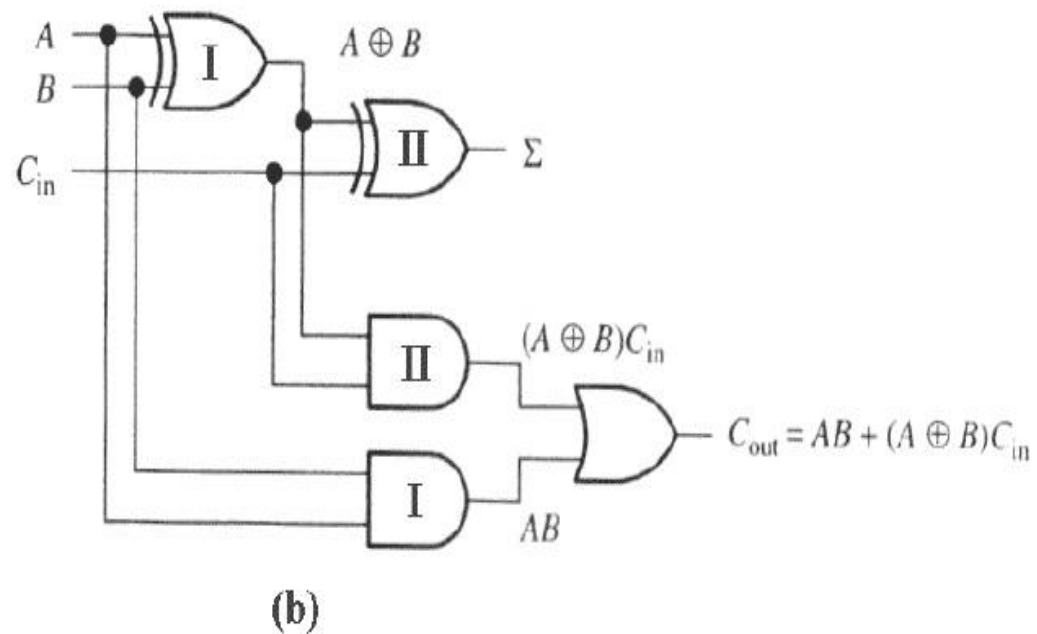
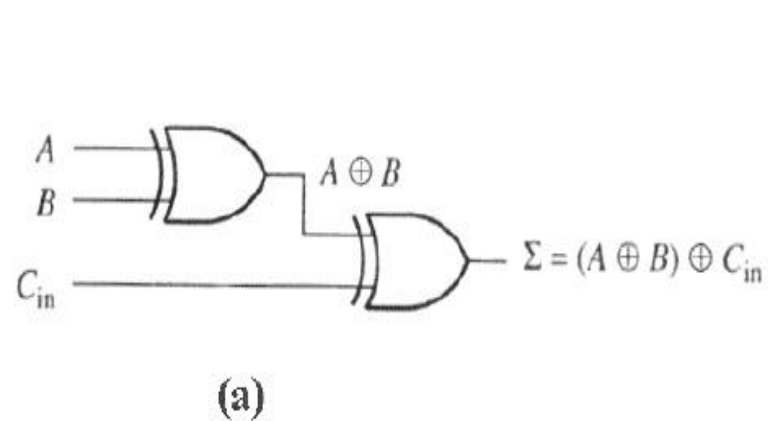
A	B	C _{in}	C _{out}	Σ
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

- $\Sigma = A'B'C_{in} + A'BC_{in}' + AB'C_{in}' + ABC_{in}$
- $\Sigma = C_{in}' \cdot (AB' + A'B) + C_{in} \cdot (A'B' + AB)$
- $\Sigma = C_{in}' \cdot (A \oplus B) + C_{in} \cdot (A \oplus B)'$
- $\Sigma = (A \oplus B) \oplus C_{in}$

- $C_{out} = A'BC_{in} + AB'C_{in} + ABC_{in}' + ABC_{in}$
- $C_{out} = AB \cdot (C_{in}' + C_{in}) + C_{in} \cdot (A'B + AB')$
- $C_{out} = AB + C_{in} \cdot (A \oplus B)$

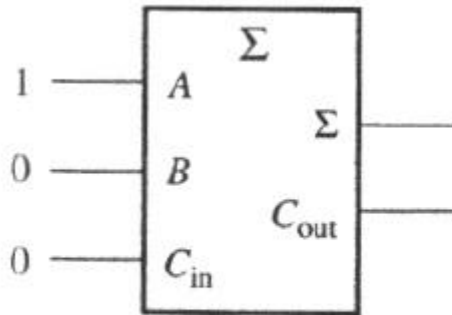
Somador Completo (Diagrama Lógico)

- a) Lógica necessária para formar a soma dos bits de entrada A e B com carry de entrada C_{in} ;
- b) Diagrama Lógico do Somador Completo, composto pelos Meio Somadores I e II

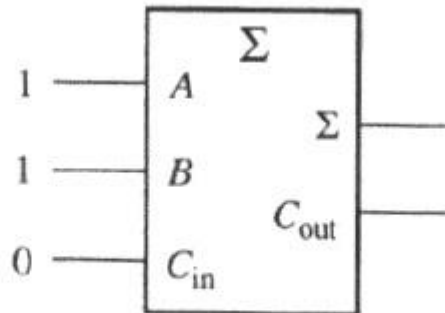


Exemplo

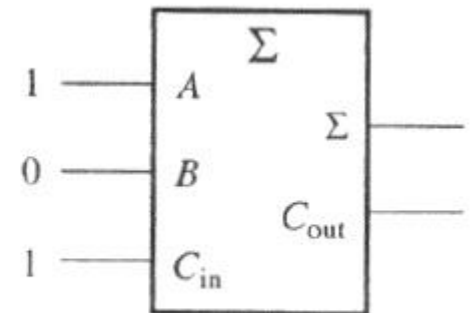
- Determine as saídas Σ e C_{out} para as situações de entrada dos somadores completos mostrados abaixo:



(a)



(b)



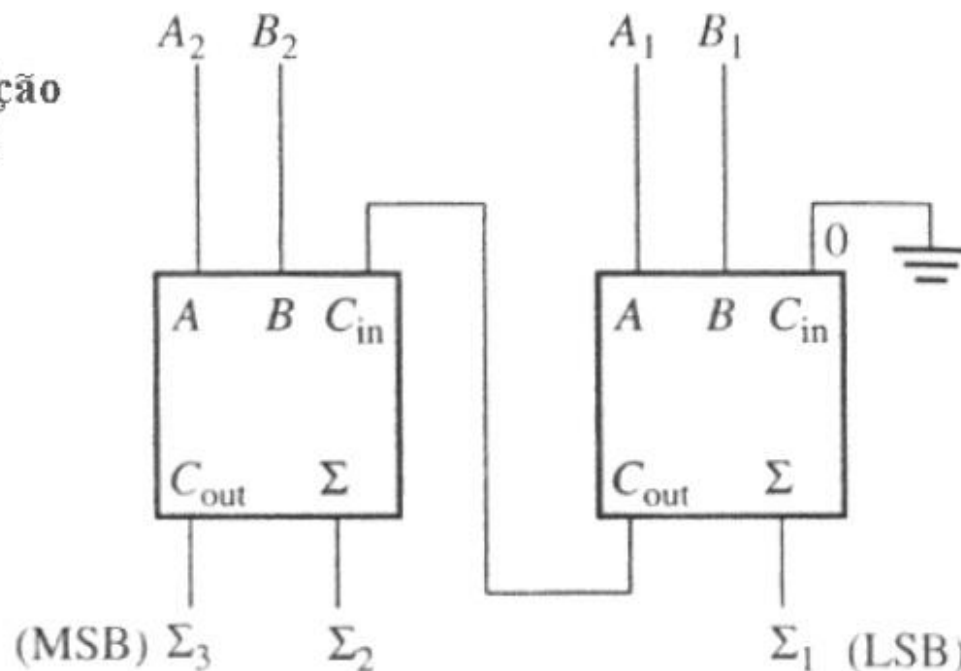
(c)

Somador em Paralelo

- Para somar palavras binárias de **N bits**, é necessário colocar em paralelo **N somadores completos**;
- Observe atentamente o exemplo abaixo de um Somador em Paralelo com 2 bits de palavra. Repare que o Somador Completo menos significativo pode ser substituído por um Meio Somador, pois não se tem *Carry* de entrada.

Regra geral para adição de palavras de 2 bits:

$$\begin{array}{r} A_2 A_1 \\ + B_2 B_1 \\ \hline \Sigma_3 \Sigma_2 \Sigma_1 \end{array}$$



Subtrator

- A subtração ($A - B$) entre duas palavras binárias A e B é executada através da soma da primeira palavra binária (que neste caso é A) com a segunda palavra (que neste caso é B) complementada a 2 ($A + (-B)$);
- A operação de **Complemento de 2** é equivalente ao acréscimo do sinal “-” ao número binário.
- Antes de a técnica do Complemento de 2, devemos estar cientes como funciona a técnica do **Complemento de 1**, que nada mais é do que a inversão de todos os bits da palavra binária, ou seja, cada bit do número binário é substituído pelo seu complemento. Veja o exemplo abaixo:

10010110 -> 01101001 (Número Complementado a 1)

Complemento de 2

- O Complemento de 2 é o sistema mais usado para representação de números inteiros com sinal nos computadores modernos;
- O dígito mais significativo (MSB) é o que informa o sinal do número;
- Os números são escritos da seguinte forma:
 - **Positivos:** Sua Magnitude é representada na sua forma binária direta, e um bit de sinal 0 é colocado na frente do MSB [(Bit 0) + o número em binário]
 - **Negativos:** Sua Magnitude é representada na forma de complemento de 2, e um bit de sinal é colocado na frente do MSB [Pegamos o número em binário e aplicamos o Complemento de 1 e depois somamos 1 ao valor Complementado em 1 na posição do bit menos significativo]

Ex.: O número **0110** fica **1001** em complemento de 1
Depois somamos **1001** com **0001** que fica **1010**

Exemplo Resolvido

- Represente o número decimal **-14** em um número binário com sinal utilizando a técnica de complemento de 2.

Como **-14** é um número negativo, não basta somente acrescentar um 0 na frente da magnitude do número em binário. Devemos encontrar o número decimal **+14** em binário, acrescentarmos o bit de sinal (que neste caso é 0, pois **+14** é positivo) e aplicarmos o complemento de 2.

1. Primeiramente, **+14** em binário é: **1110**
2. Acrescentando o bit de sinal, temos: **01110**
3. Aplicando o Complemento de 2, temos:
 - ✓ Primeiro, complementamos o número a 1: **10001**
 - ✓ Por fim, somamos **10001** com **00001** e temos: **10010**

Negação

A **Negação** é converter um número binário positivo em um número binário negativo equivalente ou vice-versa.

Exemplos

1. Utilize a técnica do complemento de 2 e represente os números decimais abaixo em números binários com sinal:
 - a) **-12**
 - b) **10**
 - c) **12**
 - d) **-23**

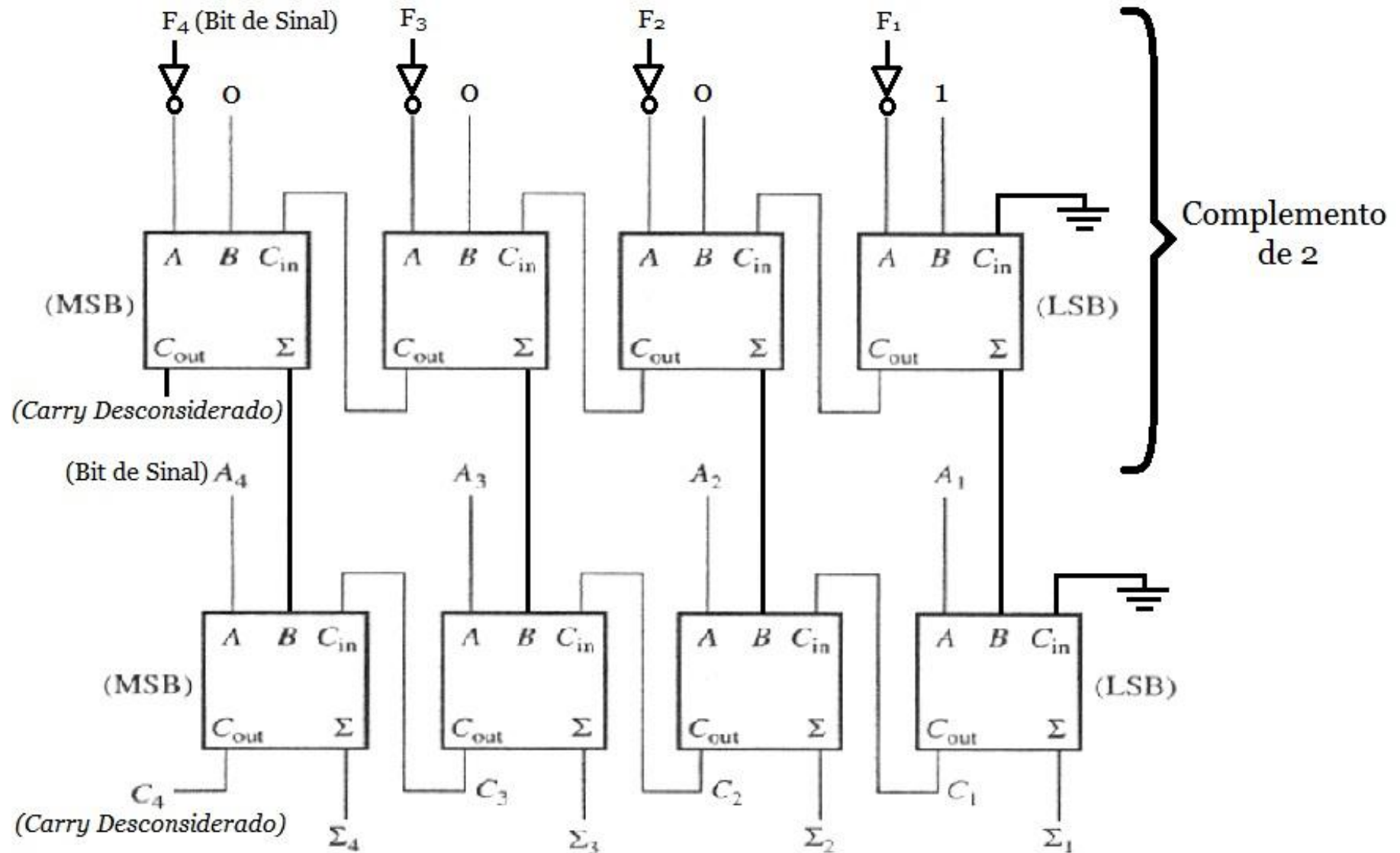
Adição no Sistema Complemento de 2

- Existem 5 casos de adição no sistema de complemento de 2 que devem ser considerados. São eles:
 1. Dois números positivos: O resultado é um número positivo;
 2. Um número positivo e outro número menor e negativo: O resultado é um número positivo;
 3. Um número positivo e outro número maior e negativo: O resultado é um número negativo;
 4. Dois números negativos: O resultado é um número negativo;
 5. Números iguais e sinais opostos: O resultado é zero.
- Em todos os casos, o bit de sinal participa do processo de soma;
- O *Carry* gerado pelos bits de sinal é sempre desconsiderado.

Subtração

- Para realizar a subtração entre dois números binários, basta somar o minuendo com a negação do subtraendo;
- **Ex.:** Considere duas palavras binárias (**A** e **F**) com três bits de magnitude e um bit de sinal (bit mais significativo). Digamos que você queira subtrair **F** de **A**. Em primeiro lugar, deve-se obter o complemento de 2 da palavra **F**: (**F₄F₃F₂F₁**) e depois somar a palavra binária **A**: (**A₄A₃A₂A₁**) com o complemento de 2 de **F**. Observe a figura do próximo slide e repare que o resultado desta soma será a **subtração de A por F** (**Σ₄Σ₃Σ₂Σ₁**). Os bits **F₄** e **A₄** são os bits de sinal. Os bits (**F₃F₂F₁**) e (**A₃A₂A₁**) são os bits de magnitude. O bit de *Carry* **C₄** é desconsiderado.

Subtração



Multiplicador

- As regras básicas para multiplicação binária são:

1. $0 \times 0 = 0$

2. $0 \times 1 = 0$

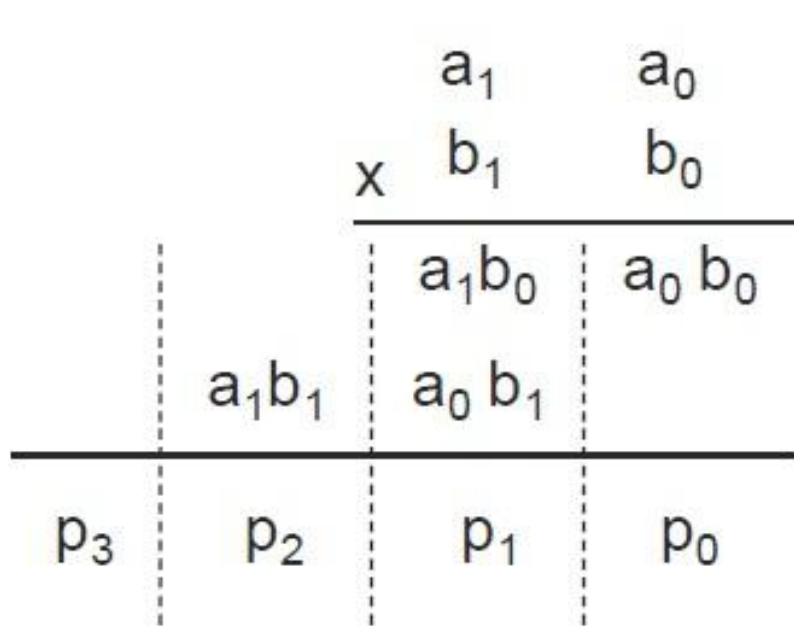
3. $1 \times 0 = 0$

4. $1 \times 1 = 1$

- Ao analisarmos os resultados, podemos perceber que a operação de multiplicação binária é similar à operação AND;
- A multiplicação de duas palavras com **N bits** resulta em uma palavra de $(2 \times N)$ bits caso tenha o *carry* mais significativo e $(2 \times N) - 1$ bits caso não tenha o *carry* mais significativo.

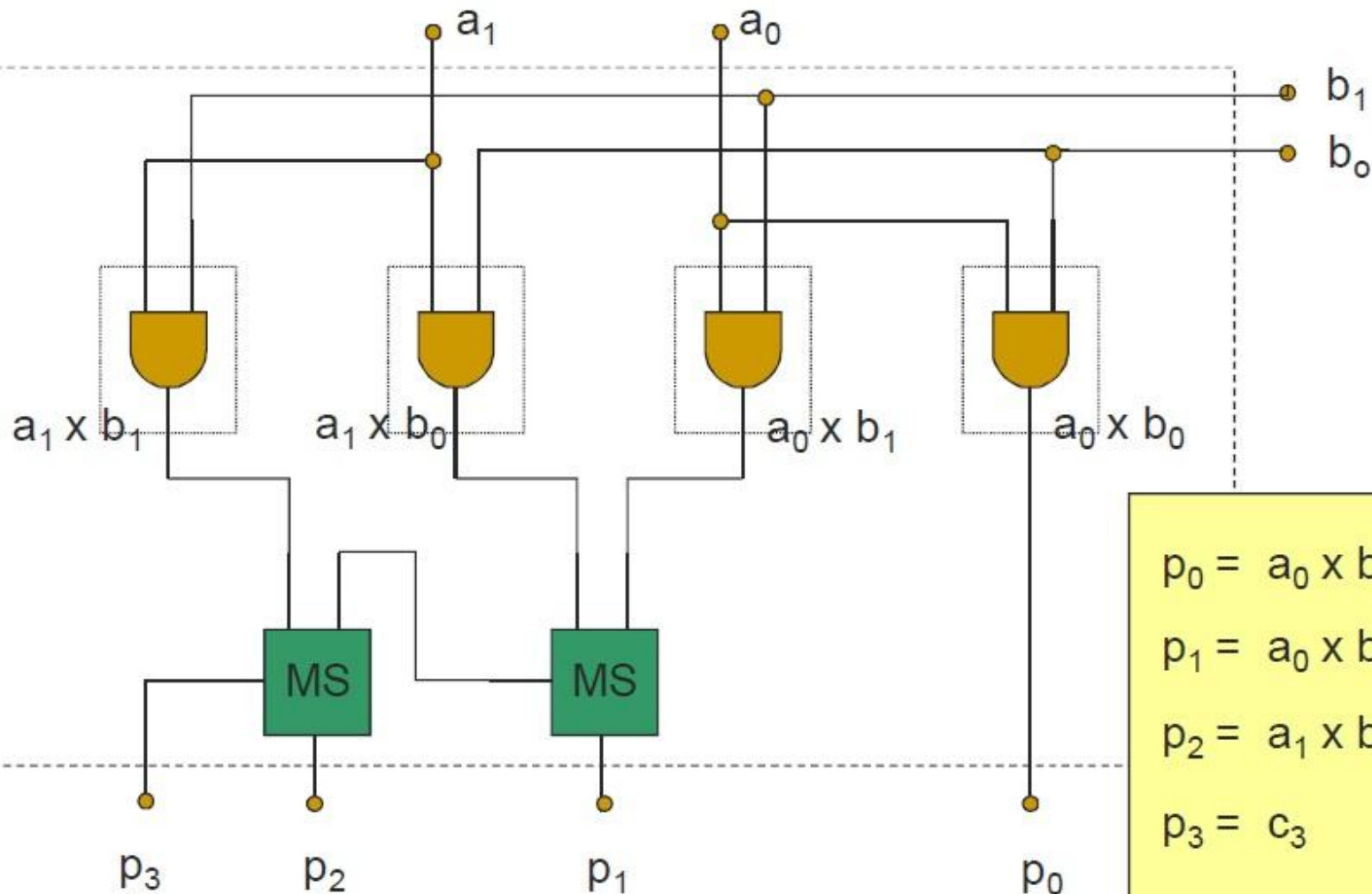
Multiplicador 2 x 2

- A figura abaixo mostra um multiplicador de 2 palavras binárias de 2 bits que resulta em uma palavra de 4 bits [caso seja considerado o *carry* mais significativo (**C₃**)]. No slide posterior, mostra o circuito lógico deste multiplicador.



$$\left\{ \begin{array}{l} p_0 = a_0 \times b_0 \\ p_1 = a_0 \times b_1 + a_1 \times b_0 \\ p_2 = a_1 \times b_1 + c_2 \\ p_3 = c_3 \end{array} \right.$$

Multiplicador 2 x 2 (Circuito Lógico)



$$p_0 = a_0 \times b_0$$

$$p_1 = a_0 \times b_0 + a_1 \times b_0$$

$$p_2 = a_1 \times b_1 + c_2$$

$$p_3 = c_3$$

Exemplo Resolvido

- Faça a Multiplicação de **1101** por **1011**, mostrando os produtos parciais e o produto final.

1101	multiplicando
<u>1011</u>	multiplicador
1101	}
1101	
0000	
<u>1101</u>	
10001111	produto

- OBS.:** Observe que neste caso foi considerado o *carry* mais significativo.

Divisor

- Inicialmente, examina-se o dividendo começando com o MSB (bit mais significativo) e determina-se o número de bits requerido para exceder o valor do divisor;
- Quando achar este valor, coloque 1 no quociente e subtraia o divisor do valor do dividendo selecionado;
- Então transporte o próximo bit mais significativo do dividendo para o atual resto.

Exemplo Resolvido

- Faça a Divisão de **100111** por **110**, mostrando os restos parciais e o resto final.

$$D = 39 = 1\ 0\ 0\ 1\ 1\ 1$$

$$d = 6 = 1\ 1\ 0$$

	1 0 0 1 1 1		
	1 1 0	Não Resta	→
Resto Parcial	1 0 0 1		
	1 1 0	Resta	→
Resto Parcial	0 0 1 1 1		
	1 1 0	Resta	→
Resto Parcial	0 0 0 0 1 1		
	1 1 0	Não Resta	→
Resto	0 0 0 0 1 1		

	1 1 0
0 1 1 0	

Quociente

$$\text{Quociente} = 6 = 1\ 1\ 0$$

$$\text{Resto} = 3 = 1\ 1$$

Exercícios

1. Construa um Somador em Paralelo que tenha duas palavras binárias de 4 bits (1 bit de sinal + 3 bits de magnitude).
2. Faça a Negação binária dos números decimais abaixo.
 - a) **+32**
 - b) **+6**
 - c) **-54**
 - d) **-13**
3. Construa um Subtrador que tenha duas palavras binárias de 3 bits (1 bit de sinal + 2 bits de magnitude).
4. Construa um Multiplicador 3x3 mostrando o circuito lógico do mesmo.