

Organização e Arquitetura de Computadores I

Aritmética Computacional

Sumário

- Unidade Lógica e Aritmética
- Representação de Números Inteiros
- Aritmética de Números Inteiros
- Representação de Números de Ponto Flutuante
- Aritmética de Números de Ponto Flutuante

Unidade Lógica e Aritmética

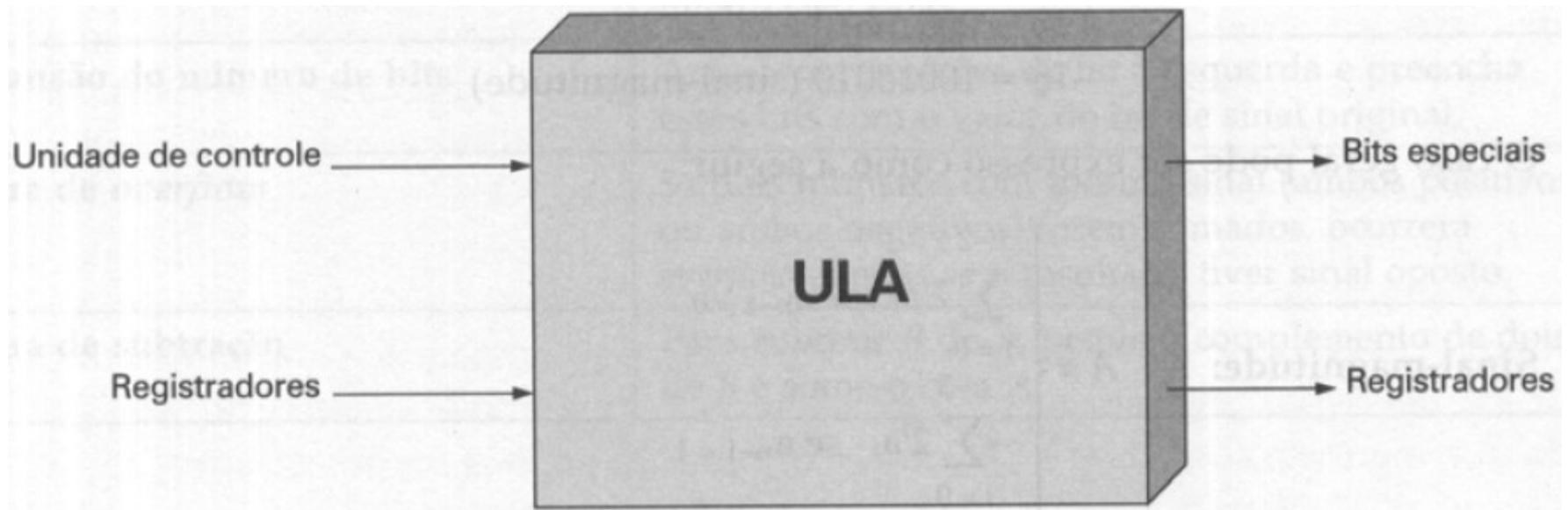
- A Unidade Lógica e Aritmética (ULA)
 - É a parte do computador que de fato executa as operações aritméticas e lógicas sobre os dados.
 - Todos os outros elementos do computador servem, principalmente, para trazer os dados a serem processados pela ULA e receber os resultados das operações efetuadas.
- Constitui o núcleo ou a essência do computador.

Unidade Lógica e Aritmética

- A ULA é baseada em dispositivos lógicos digitais simples, capazes de armazenar dígitos binários e efetuar operações simples de lógica booleana.
- Os dados são fornecidos à ULA em registradores e os resultados de uma operação são armazenados em registradores.
- A ULA pode também ativar bits especiais (*flags*) para indicar o resultado de uma operação.

Organização e Arquitetura de Computadores I

Unidade Lógica e Aritmética



Unidade Lógica e Aritmética

- A aritmética computacional geralmente opera com dois tipos de números muito diferentes: números inteiros e números de ponto flutuante. Em ambos os casos, a escolha da representação é uma questão crucial de projeto.

Representação de Números Inteiros

- No sistema de números binários, é possível representar números arbitrários usando os dígitos zero e um.

$$-1101,0101_2 = -13,3125_{10}$$

- Para armazenar esses números no computador, não é possível usar os sinais de menos e vírgula. Apenas dígitos binários (0 e 1) podem ser usados.

Representação de Números Inteiros

● Representação sinal-magnitude

- Utiliza o bit mais significativo da palavra como um bit de sinal: se o bit mais significativo for 0, o número será positivo; se for 1, o número será negativo.
- É a forma mais simples de representação que emprega um bit de sinal.

$$+18_{10} = 00010010$$

$$-18_{10} = 10010010$$

Sinal

Magnitude

Representação de Números Inteiros

● Representação sinal-magnitude

- Algumas desvantagens dessa forma de representação:
 - Numa operação de adição e subtração, é preciso considerar tanto a magnitude quanto o sinal dos dois operandos;
 - Existem duas representações de 0:

$$+0_{10} = 00000000$$

$$-0_{10} = 10000000$$

Representação de Números Inteiros

● Representação em complemento de dois

- Assim como a representação sinal-magnitude, a representação em complemento de dois usa o bit mais significativo como bit de sinal. Entretanto, os demais bits são interpretados de maneira diferente.
- Considere um inteiro A de n bits:
 - Se A for positivo, então o bit de sinal, n_{a-1} , será igual a zero.
 - Se A é um número negativo ($A < 0$), o bit de sinal, n_{a-1} , é 1.

Representação de Números Inteiros

● Representação em complemento de dois

- Exemplo de aplicação do complemento de 2 no inteiro positivo

0110_2 (5_{10}):

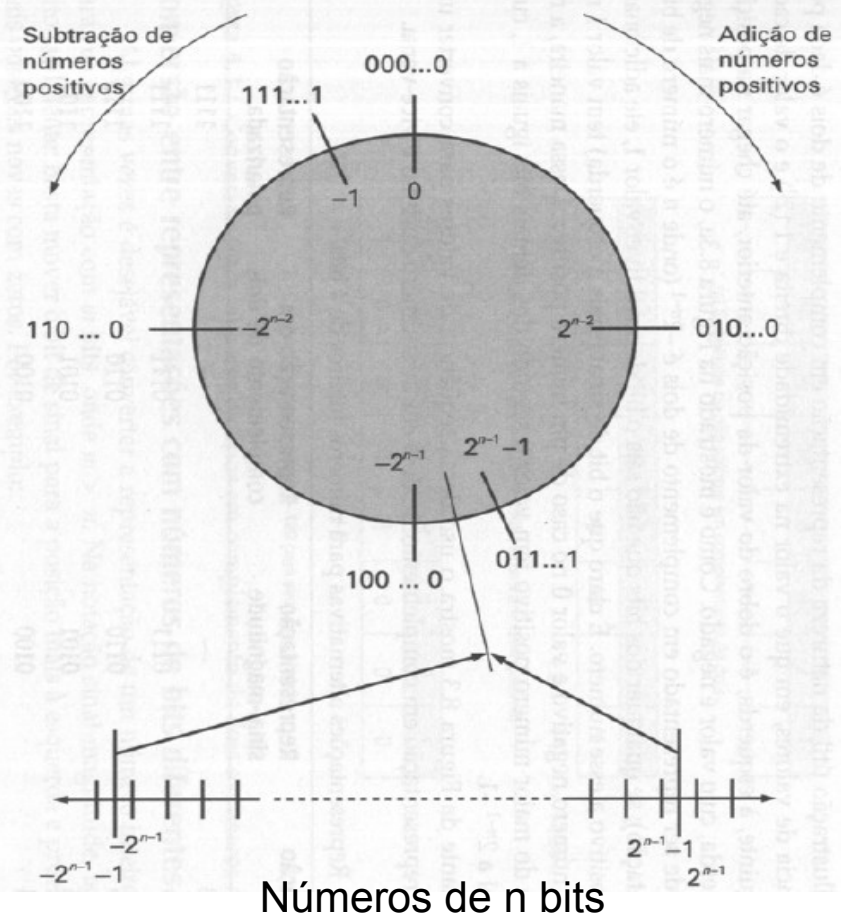
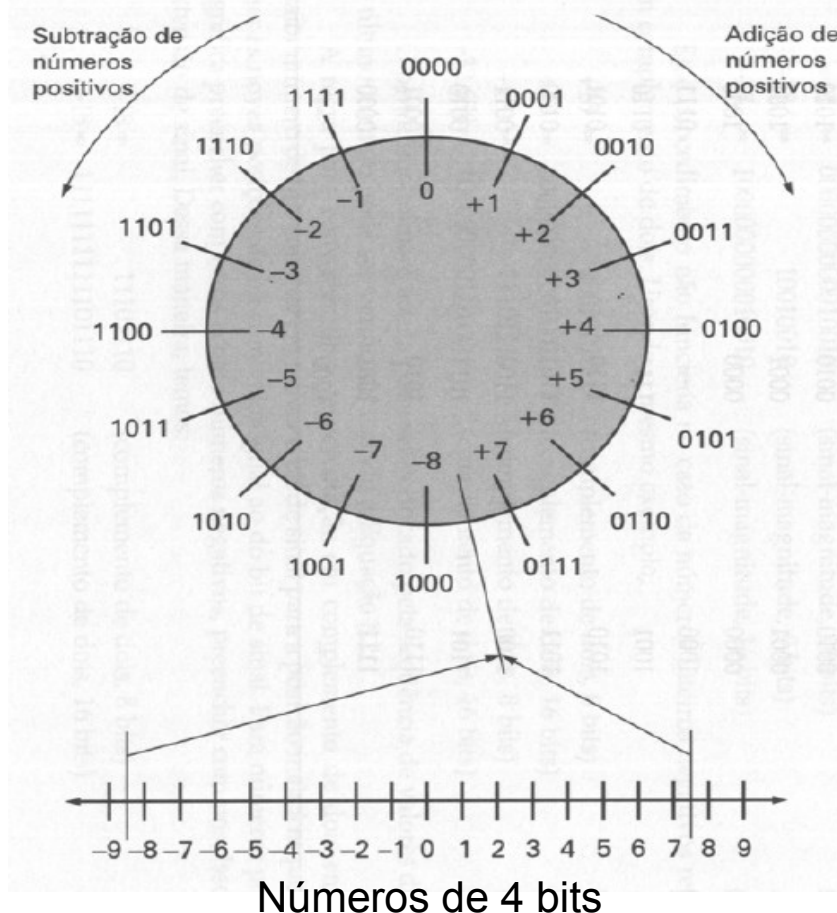
Número original	0110
Inversão	1001
Complemento de 1	$+1111$
	0110

- O zero é tratado com um número positivo.
- É possível representar $2^{n-1}-1$ números positivos.

0 a $2^{n-1}-1$

Organização e Arquitetura de Computadores I

Representação de Números Inteiros



Representação de Números Inteiros

● Representação em complemento de dois

- Representação de 2^{n-1} números negativos.

-1 a -2^{n-1}

- O número mais significativo pode ser representado por -2^{n-1} .

Representação de Números Inteiros

● Representação em complemento de dois

-128	64	32	16	8	4	2	1

(a) Sequência de oito valores em complemento de dois

-128	64	32	16	8	4	2	1
1	0	0	0	0	0	1	1

$$-128 + 2 + 1 = -125$$

(b) Conversão do valor binário 10000011 para um valor decimal

-128	64	32	16	8	4	2	1
1	0	0	0	1	0	0	0

$$-120 = -128$$

$$+8$$

Representação de Números Inteiros

● Representação em complemento de dois

- Embora a representação em complemento de dois possa parecer pouco natural, **ela torna mais simples a implementação das operações aritméticas mais importantes (adição e subtração).**

Representação de Números Inteiros

● Conversão entre representações com números de bits diferentes

- Algumas vezes é desejável converter a representação de um número inteiro com n bits para sua representação com m bits, onde $m > n$.
- Na representação sinal-magnitude é bem simples:

00001010(8 bits) 0000000000001010(16bits)

10001010(8bits) 1000000000001010(16bits)



Representação de Números Inteiros

● Conversão entre representações com números de bits diferentes

- Na representação complemento de dois, é necessário repetir o primeiro bit preenchendo as casas à esquerda necessárias:

00001010(8bits) **00000000000001010(16bits)**

11110110(8bits) **1111111111110110(16bits)**

Aritmética de Números Inteiros

● Negação

- Na representação sinal-magnitude, basta inverter o valor do bit de sinal.

10001100 Negação 00001100

- Na representação em complemento de dois, é feita a operação de complemento de dois.

11110100

00001011 (+1)

00001100 Negação

Aritmética de Números Inteiros

● Negação (casos especiais)

- Número 0

00000000

11111111 (+1)

100000000

Ignorado

Aritmética de Números Inteiros

● Negação (casos especiais – *overflow*)

- Se negarmos um padrão de bits constituído de um bit com valor 1 seguido de $n-1$ bits de valor 0, obteremos esse mesmo número.

10000000

01111111 (+1)

10000000

Aritmética de Números Inteiros

● Adição e Subtração

- A adição é feita efetuando a soma direta dos dois números:

$$\begin{array}{r} 0101 \text{ (5)} \\ +0001 \text{ (1)} \\ \hline 0110 \text{ (6)} \end{array}$$

$$\begin{array}{r} 1110 \text{ (-2)} \\ +1111 \text{ (-1)} \\ \hline 11101 \text{ (-3)} \end{array}$$

Ignorado

$$\begin{array}{r} 0100 \text{ (4)} \\ +1100 \text{ (-4)} \\ \hline 10000 \text{ (0)} \end{array}$$

Ignorado

Aritmética de Números Inteiros

● Adição e Subtração

- *Overflow* – essa situação é determinada quando dois números com o mesmo sinal forem somados e o seu resultado tiver sinal oposto.

$$\begin{array}{r} 0101 \text{ (5)} \\ +0110 \text{ (6)} \\ \hline 1011 \text{ (11)} \end{array}$$

Overflow

$$\begin{array}{r} 1001 \text{ (-7)} \\ +1010 \text{ (-6)} \\ \hline 10011 \text{ (-13)} \end{array}$$

Overflow

Aritmética de Números Inteiros

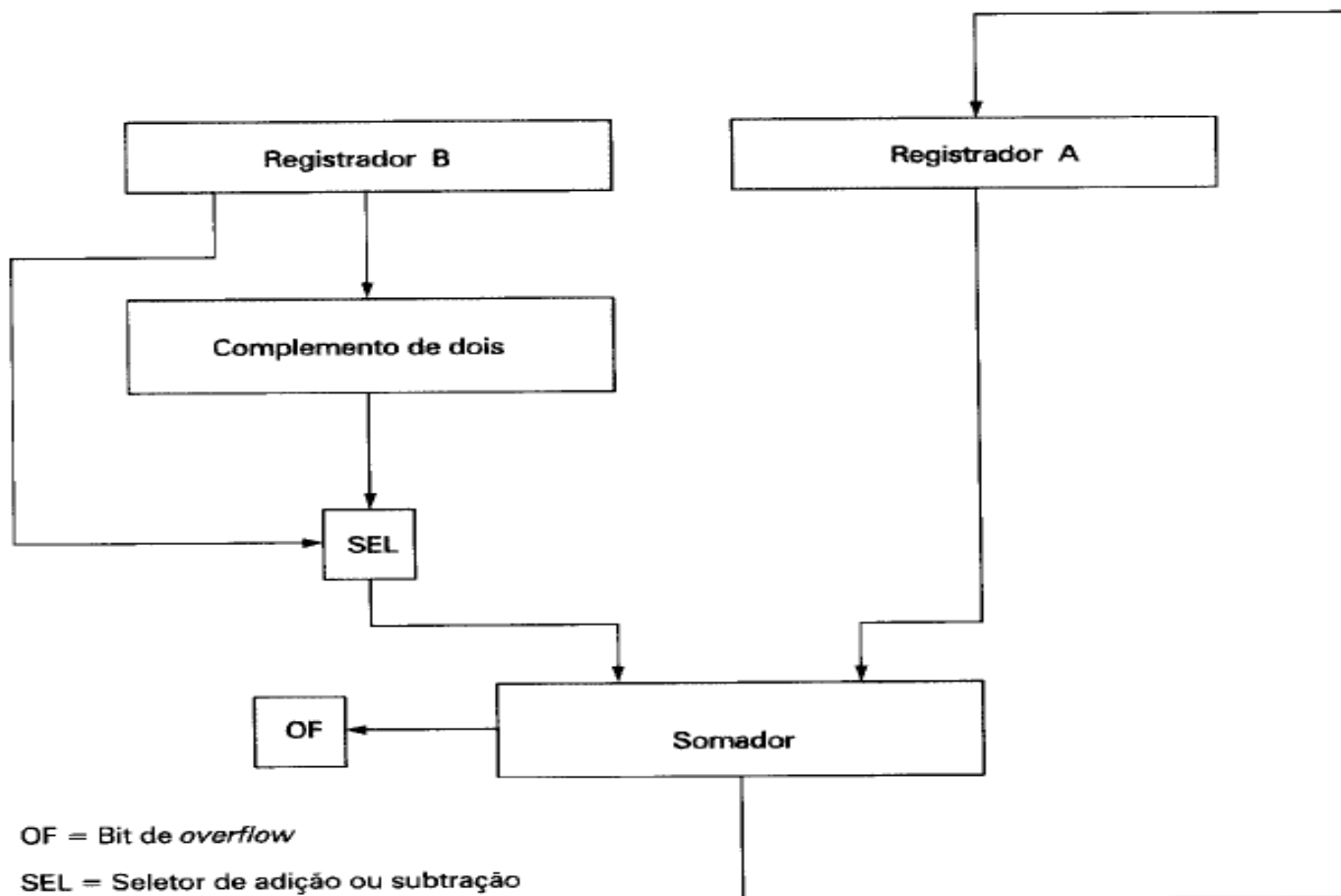
● Adição e Subtração

- Para subtrair um número M de um outro S, basta somar o M com a negação de S.

M: 0101 (5)	0101
S: 1110 (-2)	<u>+0010</u>
-S: 0010 (2)	0111 (7)

Organização e Arquitetura de Computadores I

Aritmética de Números Inteiros



Aritmética de Números Inteiros

● Multiplicação

- É uma operação complexa, seja implementada em *hardware* seja em *software*.
- Uma grande variedade de algoritmos de multiplicação tem sido usada em diversos computadores.

Aritmética de Números Inteiros

● Multiplicação – números inteiros sem sinal

- Forma mais simples de multiplicar.
- Envolve a geração de produtos parciais, que são facilmente determinados.
- O produto total é obtido através da soma dos produtos parciais.
- A multiplicação de números inteiros binários de n bits resulta em um produto com até $2n$ bits de comprimento.

Aritmética de Números Inteiros

● Multiplicação – números inteiros sem sinal

$$\begin{array}{r}
 1010 \text{ Multiplicando } 10 \\
 \times 1101 \text{ Multiplicador } 13 \\
 \hline
 \begin{array}{r}
 1010 \\
 0000 \\
 1010 \\
 + 1010 \\
 \hline
 10000010 \text{ Produto } 130
 \end{array}
 \end{array}$$

Produtos parciais

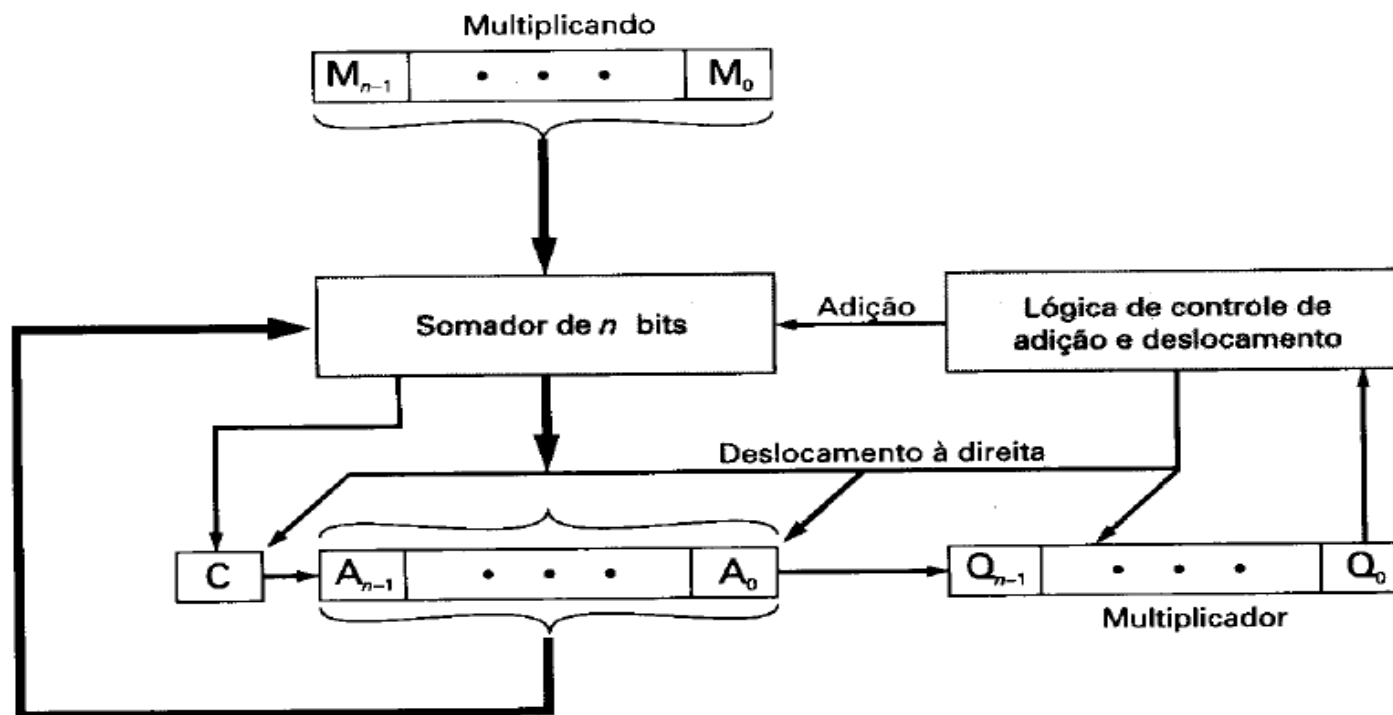
Aritmética de Números Inteiros

● Multiplicação – números inteiros sem sinal

- A multiplicação pode ser feita de modo mais eficiente do que na forma usual:
 - Os produtos parciais podem ser imediatamente acumulados
 - Para cada 1 no multiplicador, é necessário realizar uma operação de soma e um deslocamento
 - Para cada 0 no multiplicador, apenas um deslocamento é necessário

Aritmética de Números Inteiros

● Multiplicação – números inteiros sem sinal



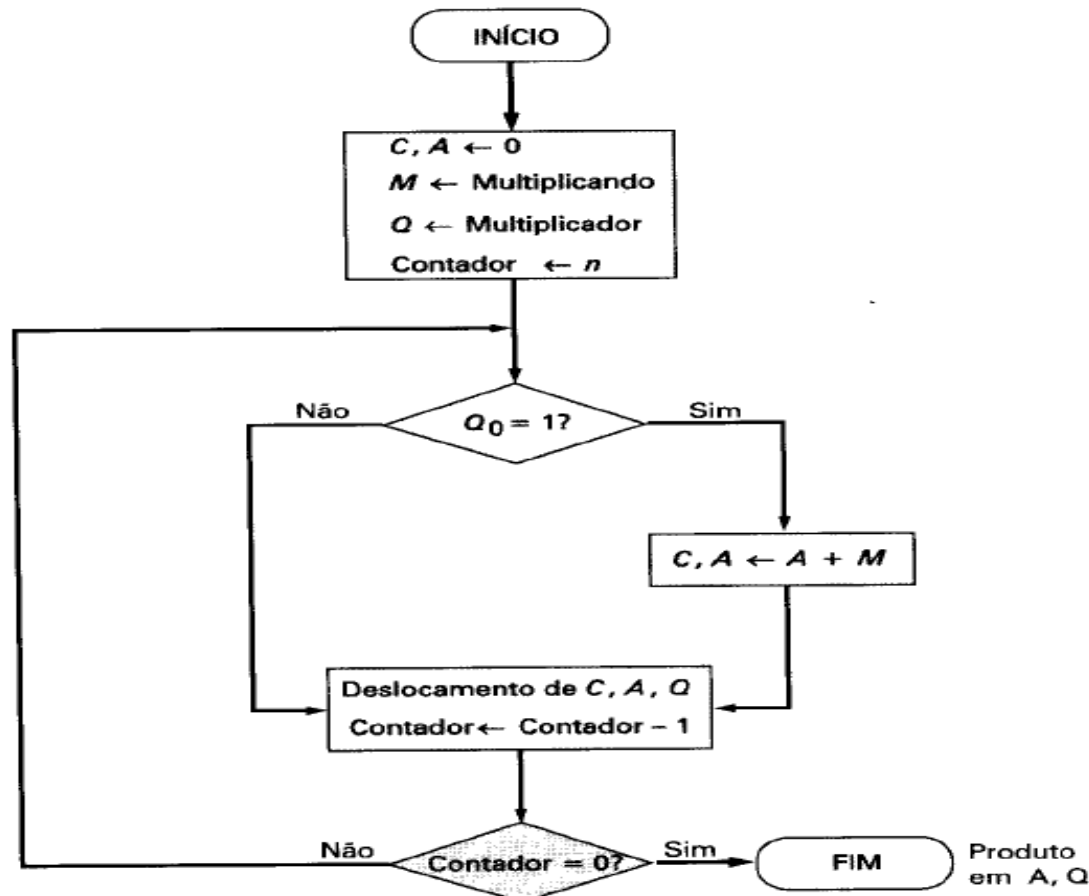
Aritmética de Números Inteiros

● Multiplicação – números inteiros sem sinal

C	A	Q	M	
0	0000	1011	1010	Valores iniciais
0	1010	101 1	1010	Adição
0	0101	0101	1010	Deslocamento
0	1111	010 1	1010	Adição
0	0111	101 0	1010	Deslocamento
0	0011	1101	1010	Deslocamento
0	1101	110 1	1010	Adição
0	0110	1110	1010	Deslocamento

Resposta

Aritmética de Números Inteiros



Aritmética de Números Inteiros

● Multiplicação – números em complemento de 2

- O algoritmo visto anteriormente não serve para números em complemento de 2:

1010	(10)	
x 1101	(13)	
10000010	(130)	Números Inteiros

1010	(-6)	
x 1101	(-3)	
10000010	(-126)	Números em complemento de 2

Aritmética de Números Inteiros

● Multiplicação – números em complemento de 2

- Um dos algoritmos mais usados é o algoritmo de Booth. Esse algoritmo tem a vantagem de efetuar a multiplicação de maneira mais rápida do que em uma abordagem mais direta.

Aritmética de Números Inteiros

● Multiplicação – números em complemento de 2

- Multiplicador e multiplicando são armazenados nos registradores Q e M.
- Utiliza-se um registrador de 1bit, posicionado logicamente à direita do bit menos significativo (Q_0) do registrador Q e designado como Q_{-1} .
- O resultado da multiplicação é dado nos registradores A e Q.
- No início, A e Q_{-1} são inicializados com 0.

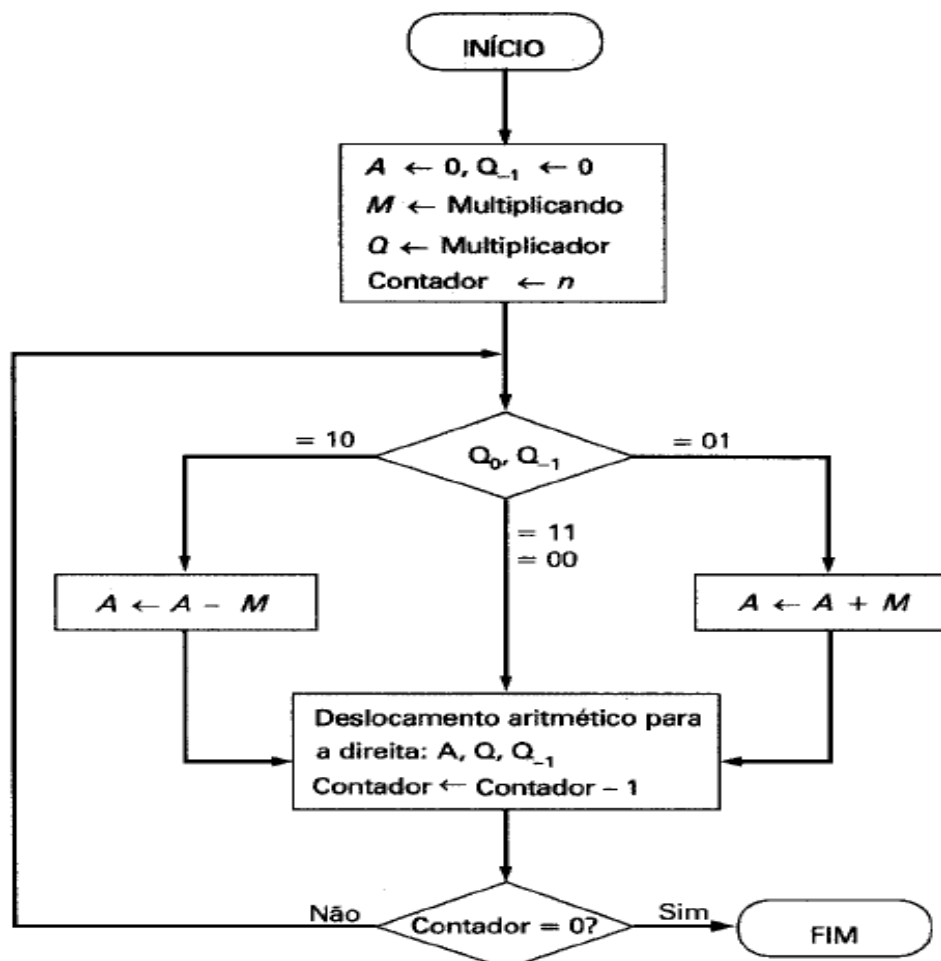
Aritmética de Números Inteiros

● Multiplicação – números em complemento de 2

- Será examinado o bit Q_0 do multiplicador (Q) e o bit de Q_{-1} :
 - Se os bits forem iguais (1-1 ou 0-0), então todos os bits dos registradores A, Q e Q_{-1} serão deslocados 1 bit para a direita (A_{n-1} é deslocado para A_{n-2} , mas sem alterar A_{n-1} – **deslocamento aritmético**).
 - Se os bits forem diferentes, o multiplicando será somado ou subtraído do registrador A (0-1 soma e 1-0 subtrai) e depois os bits serão deslocados para a direita.

Organização e Arquitetura de Computadores I

Aritmética de Números Inteiros



Aritmética de Números Inteiros

● Multiplicação – números em complemento de 2

A	Q	Q ₋₁	M	Valores Iniciais	
0000	0011	0	0111		
1001	0011	0	0111	$A \leftarrow A - M$	} Primeiro Ciclo
1100	1001	1	0111	Deslocamento	
1110	0100	1	0111	Deslocamento	} Segundo Ciclo
0101	0100	1	0111	$A \leftarrow A + M$	
0010	1010	0	0111	Deslocamento	} Terceiro Ciclo
0001	0101	0	0111	Deslocamento	
					} Quarto Ciclo