Arquitetura e Organização de Computadores

Aula - Aritmética Computacional

Prof. André D’Amato

andredamato@utfpr.edu.br

Câmpus Apucarana

UTFPR UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ CÂMPUS APUCARANA

Ministério da Educação

GOVERNO FEDERAL BRASIL PÁTRIA EDUCADORA

Adição Binária

- Exemplo:

Somar $7_{dec} + 6_{dec}$

0000 0000 0000 0111

\+

<u>0000 0000 0000 0110</u>

Adição Binária

- Exemplo:

Somar $7_{dec} + 6_{dec}$

0000 0000 0000 0111

+

<u>0000 0000 0000 0110</u>

0000 0000 0000 0001

Adição Binária

- Exemplo:

Somar $7_{dec} + 6_{dec}$

1 carry

0000 0000 0000 0111

\+

<u>0000 0000 0000 0110</u>

0000 0000 0000 0001 → Carry 1

Adição Binária

- Exemplo:

Somar 7_{dec} + 6_{dec}

1 1 carry

0000 0000 0000 0111

+

0000 0000 0000 0110

0000 0000 0000 0101 → Carry 1

Adição Binária

- Exemplo:

Somar $7_{dec} + 6_{dec}$

1 carry

0000 0000 0000 0111

+

<u>0000 0000 0000 0110</u>

0000 0000 0000 1101 $\longrightarrow$ Carry 0

Adição Binária

- Exemplo:

Subtrair $7_{dec} - 6_{dec}$

0000 0000 0000 0111

-

<u>0000 0000 0000 0110</u>

0000 0000 0000 0001 $\longrightarrow$ Carry 0

Adição Binária

- Exemplo:

Subtrair 7_{dec} - 6_{dec}

0000 0000 0000 0111

-

<u>0000 0000 0000 0110</u>

0000 0000 0000 0001 → Carry 0

Adição Binária

- Exemplo:

Subtrair 7_{dec} - 6_{dec}

0000 0000 0000 0111

-

<u>0000 0000 0000 0110</u>

0000 0000 0000 0001 → Carry 0

Subtração Binária

- Exemplo:

  Subtrair 7_{dec} - 6_{dec}

  0000 0000 0000 0111

  -

  <u>0000 0000 0000 0110</u>

  0000 0000 0000 0001 → Carry 0

Subtração Binária

- Exemplo:

  Subtrair $6_{dec} - 7_{dec}$

  0000 0000 0000 0110
  -
  <u>0000 0000 0000 0111</u>
  0000 0000 0000 0001

Subtração Binária

- Exemplo:

  Subtrair 6_{dec} - 7_{dec}

  0000 0000 0000 0110

  \-

  <u>0000 0000 0000 0111</u>

  0000 0000 0000 0001 ????? como saber que é -1

Representação de números inteiros

- Sinal magnitude:

- O primeiro bit é utilizado como bit de sinal
  - Ex:

    $0000\ 0111 \rightarrow 7_{dec}$

    $1000\ 0111 \rightarrow -7_{dec}$

Representação de números inteiros

- Complemento de dois:

- O primeiro bit é utilizado como bit de sinal
  - Ex:

0000 0111 → 7_{dec}

Inverte os bits e soma 1

$$
\begin{array}{r}
1111\ 1000 \\
+1 \\
\hline
1111\ 1001
\end{array}
\rightarrow -7_{dec}
$$

Representação de números inteiros

- Complemento de dois:

- Inverte os bits e soma 1
  - Ex:

    0000 0010 → 2_{dec}

    Inverte os bits e soma 1

    1111 1101

    $\underline{\qquad\qquad +1}$

    1111 1110 → -2_{dec}

Adição com Complemento de 2

- Ex:

  1110 (-2)

  \+

  1111 (-1)

Adição com Complemento de 2

- Ex:

    1110 (-2)

    \+

    1111 (-1)

    1

Adição com Complemento de 2

- Ex:

  1110 (-2)

  \+

  <u>1111 (-1)</u>

  01 → Carry 1

Adição com Complemento de 2

- Ex:

  1110 (-2)

  \+

  1111 (-1)

  101 → Carry 1

Adição com Complemento de 2

- Ex:

  1110 (-2)

  \+

  1111 (-1)

  1101 → Carry 1

Adição com Complemento de 2

Ex:

1110 (-2)

+

<u>1111 (-1)</u>

IGNORADO 11101 (-3) → Carry 0

Adição (Overflow)

Overflow:

- Situação em que dois números com o mesmo sinal forem somados e o seu resultado:
  - Possuir sinal oposto
  - Possuir maior quantidade de bits

- Ex:

	0 110 (6)	1001 (-7)
	+ 0 101 (5)	+ 1010 (-6)
Bit de sinal	1 011(11)	1 0011 (-13)

Exercícios Introdutórios: Soma e Subtração

Disponível no moodle

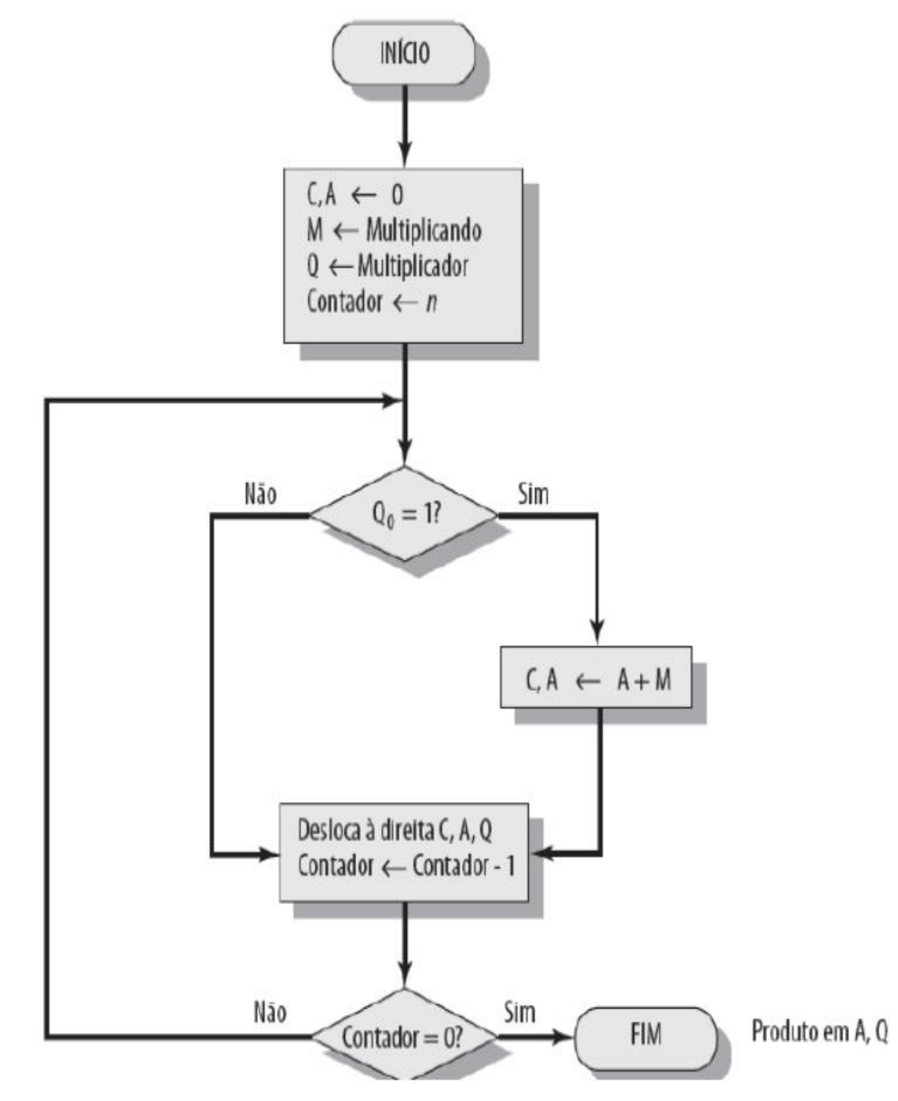
Multiplicação de Inteiros sem Sinal

- Os produtos parciais podem ser imediatamente acumulados;
  - Para cada 1 no multiplicador, é necessário realizar uma operação de adição e outra de deslocamento;
  - Para cada 0 no multiplicador, apenas um deslocamento é necessário.

Multiplicação de Inteiros sem Sinal

$$
\begin{array}{llr}
\text{Multiplicando} & & 1000_{dec} \\
\text{Multiplicador} & \times & 1001_{dec} \\
 & & \text{----}\phantom{_{dec}} \\
 & & 1000\phantom{_{dec}} \\
 & & 0000\phantom{0_{dec}} \\
 & & 0000\phantom{00_{dec}} \\
 & & 1000\phantom{000_{dec}} \\
 & & \text{---------}\phantom{_{dec}} \\
 & & 1001000_{dec}
\end{array}
$$

Multiplicação de Inteiros sem Sinal

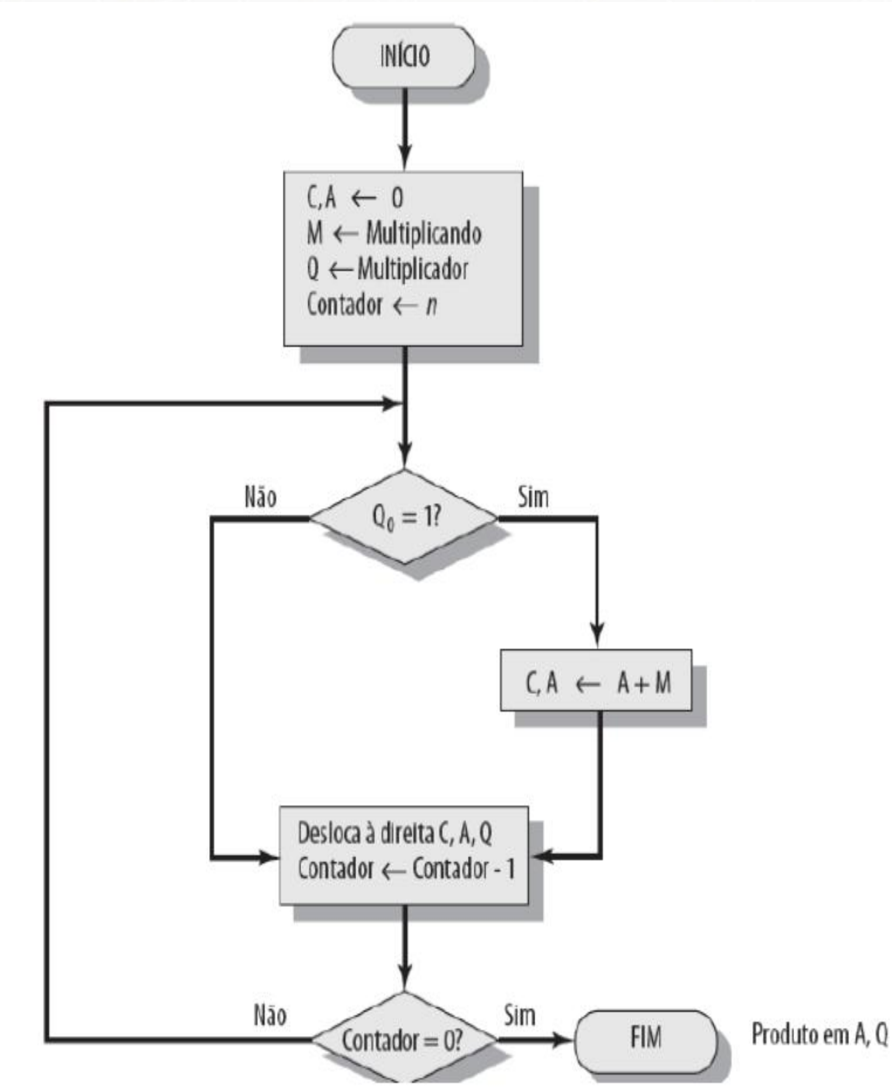


Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A	Q
0000	0101

Multiplicação de Inteiros sem Sinal

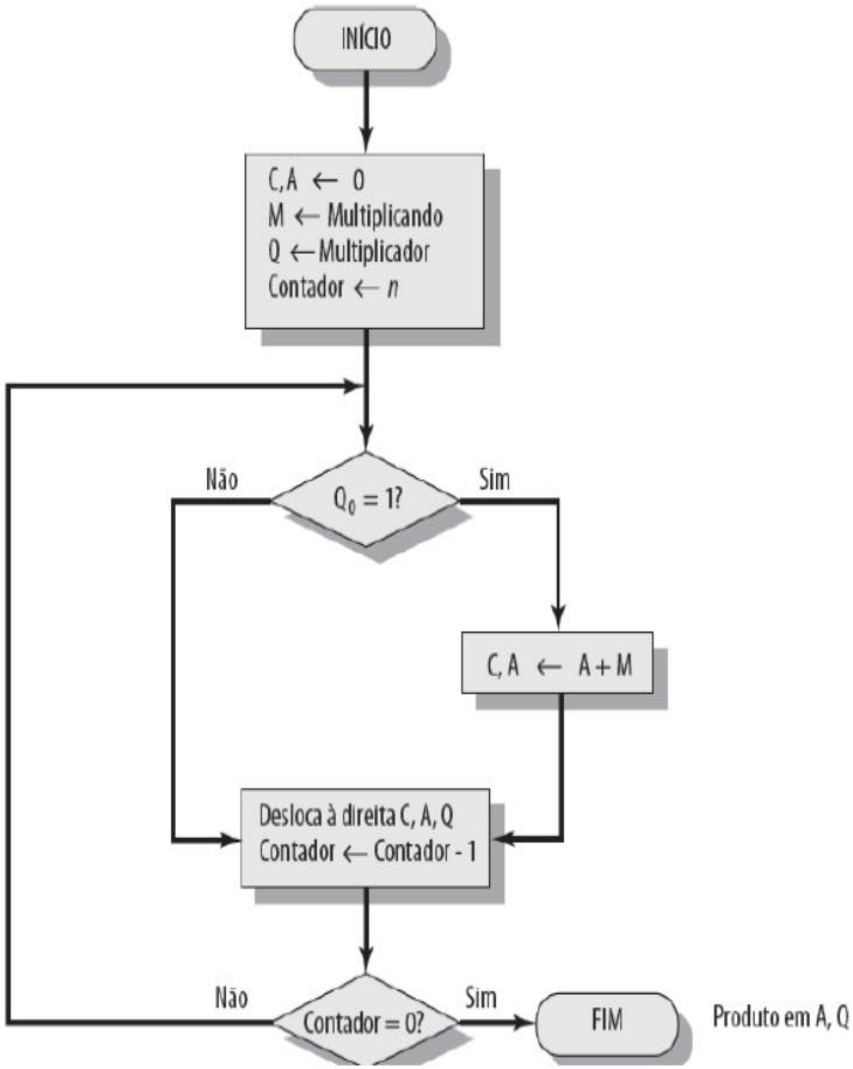


Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A Q
0000 0101
0101 0101 → soma

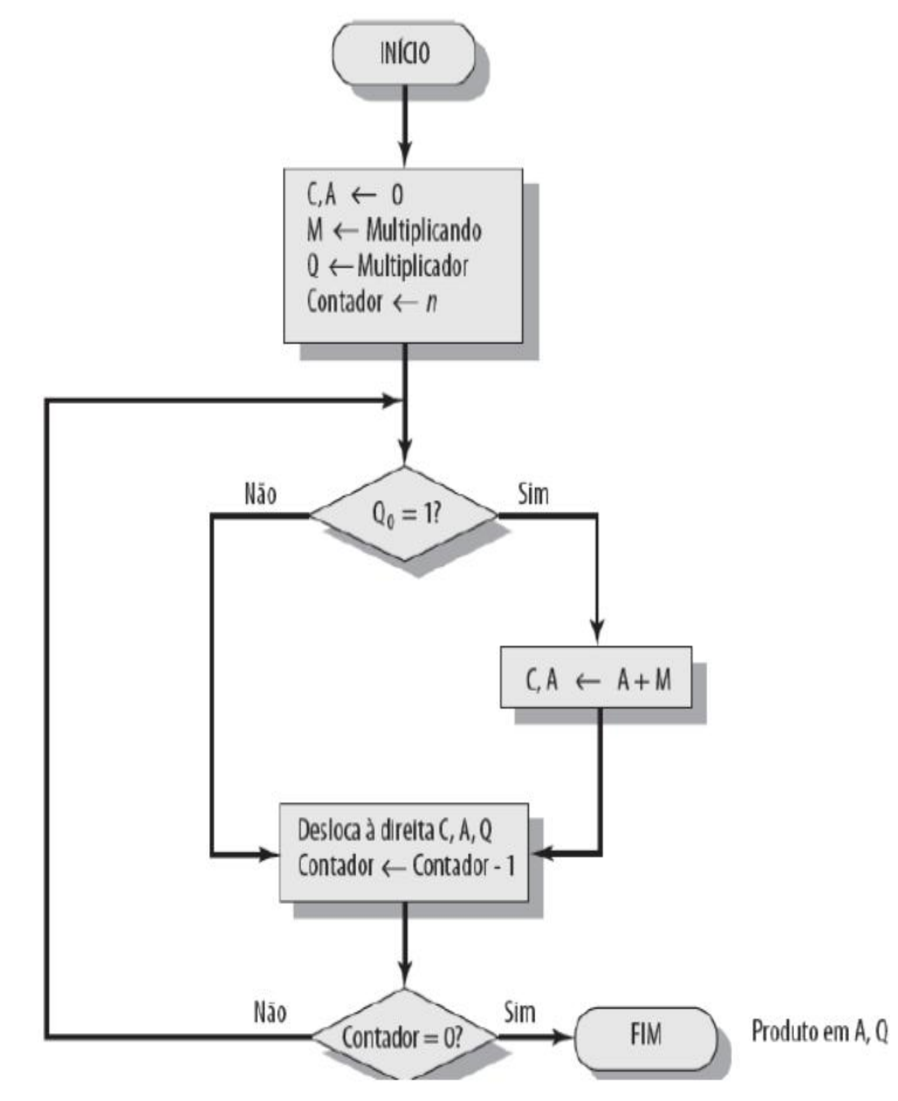
Multiplicação de Inteiros sem Sinal



Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)
  A   Q
0000 0101
0101 0101 → soma
0010 1010 → deslocamento

Multiplicação de Inteiros sem Sinal



Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A Q

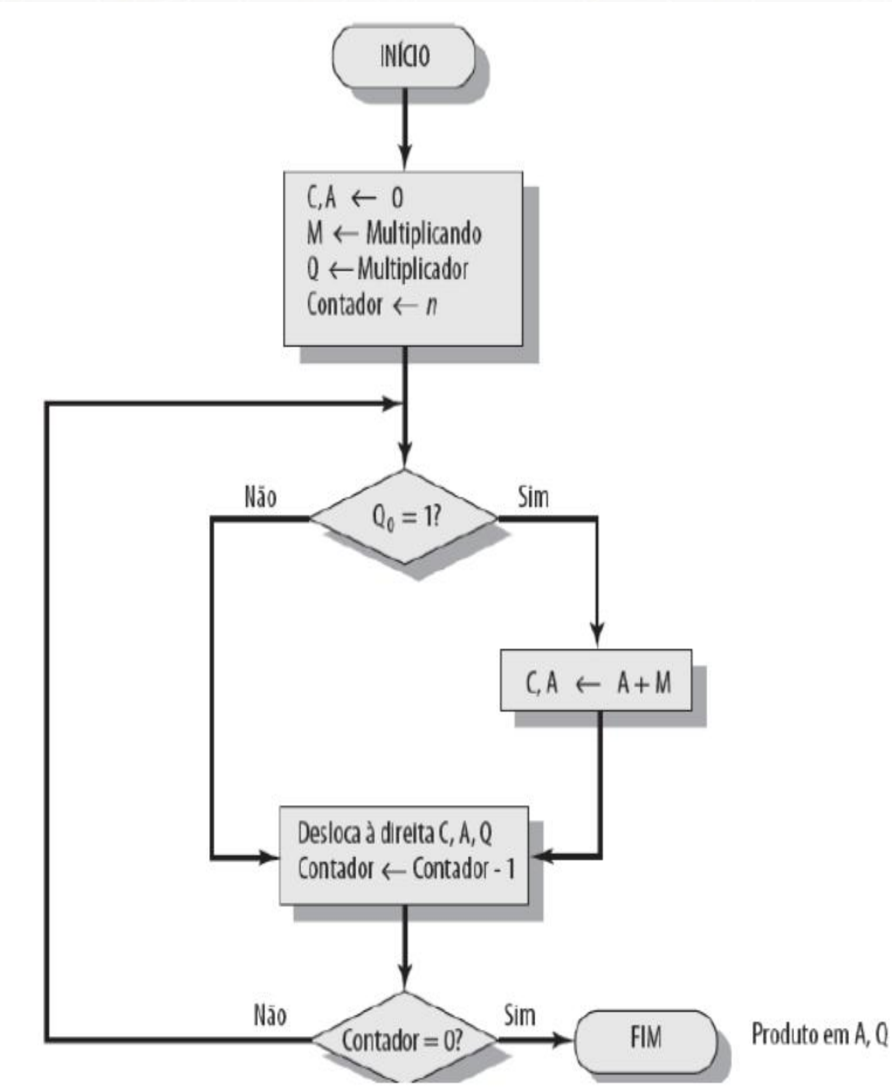
0000 0101

0101 0101 → soma

0010 1010 → deslocamento

0001 0101 → deslocamento

Multiplicação de Inteiros sem Sinal



Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A   Q

0000 0101

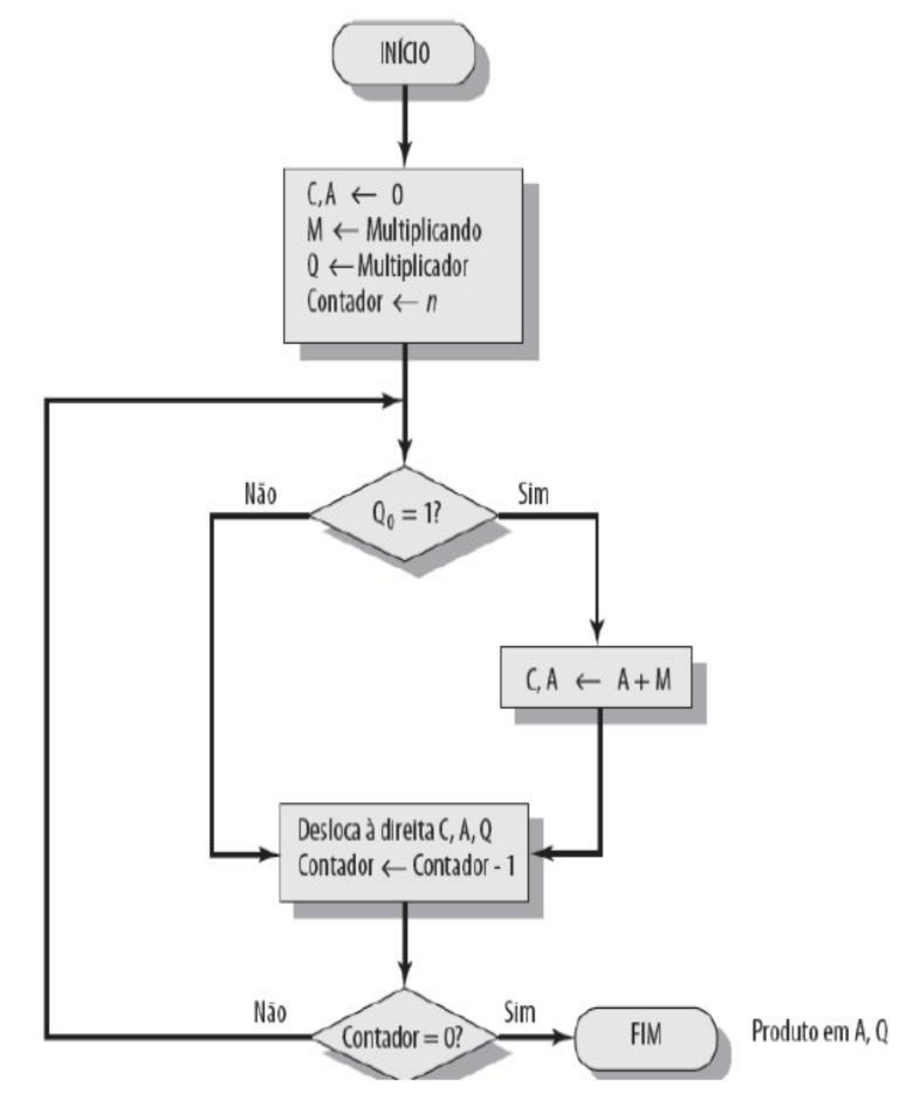
0101 0101 → soma

0010 1010 → deslocamento

0001 0101 → deslocamento

0110 0101 → soma

Multiplicação de Inteiros sem Sinal

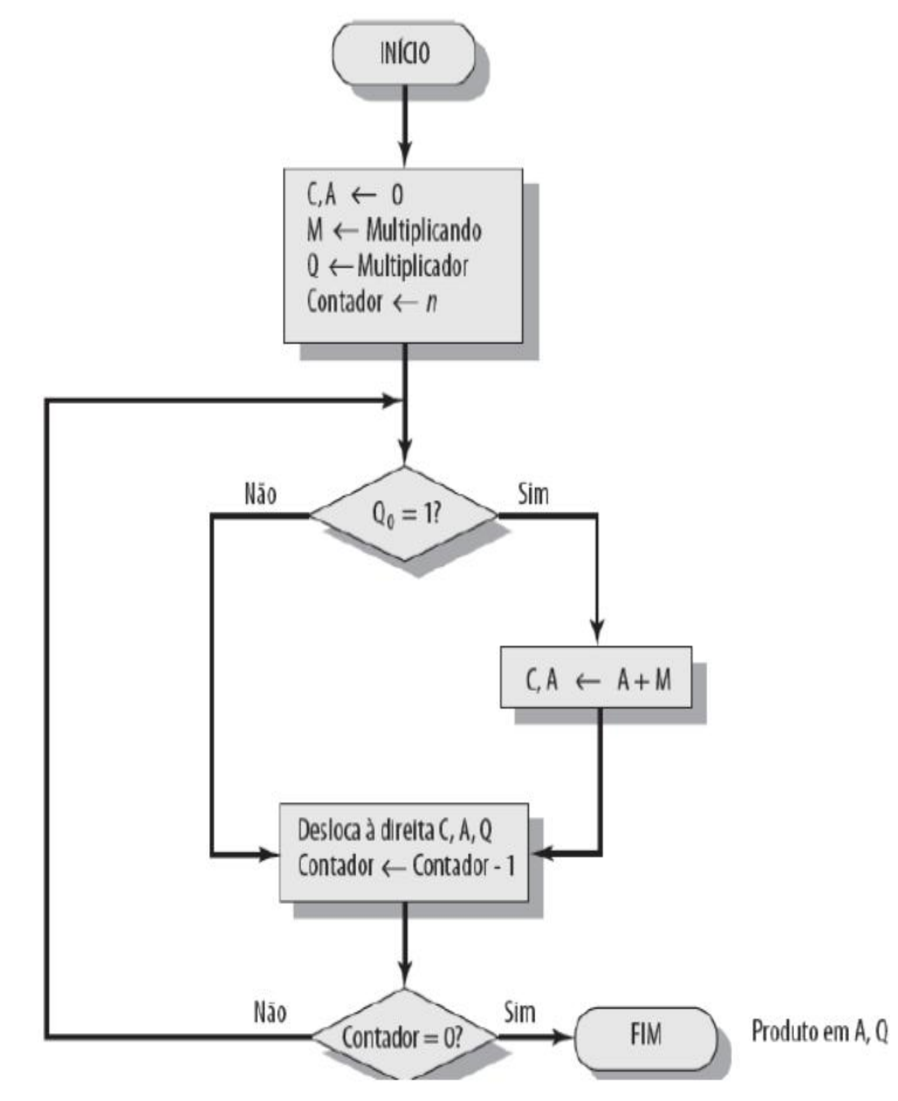


Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A	Q	
0000	0101	
0101	0101	→ soma
0010	1010	→ deslocamento
0001	0101	→ deslocamento
0110	0101	→ soma
0011	0010	→ deslocamento

Multiplicação de Inteiros sem Sinal



Utilizando o algoritmo ao lado, faça a operação aritmética abaixo:

0101 x 0101 : (5 x 5)

A Q

0000 0101

0101 0101 → soma

0010 1010 → deslocamento

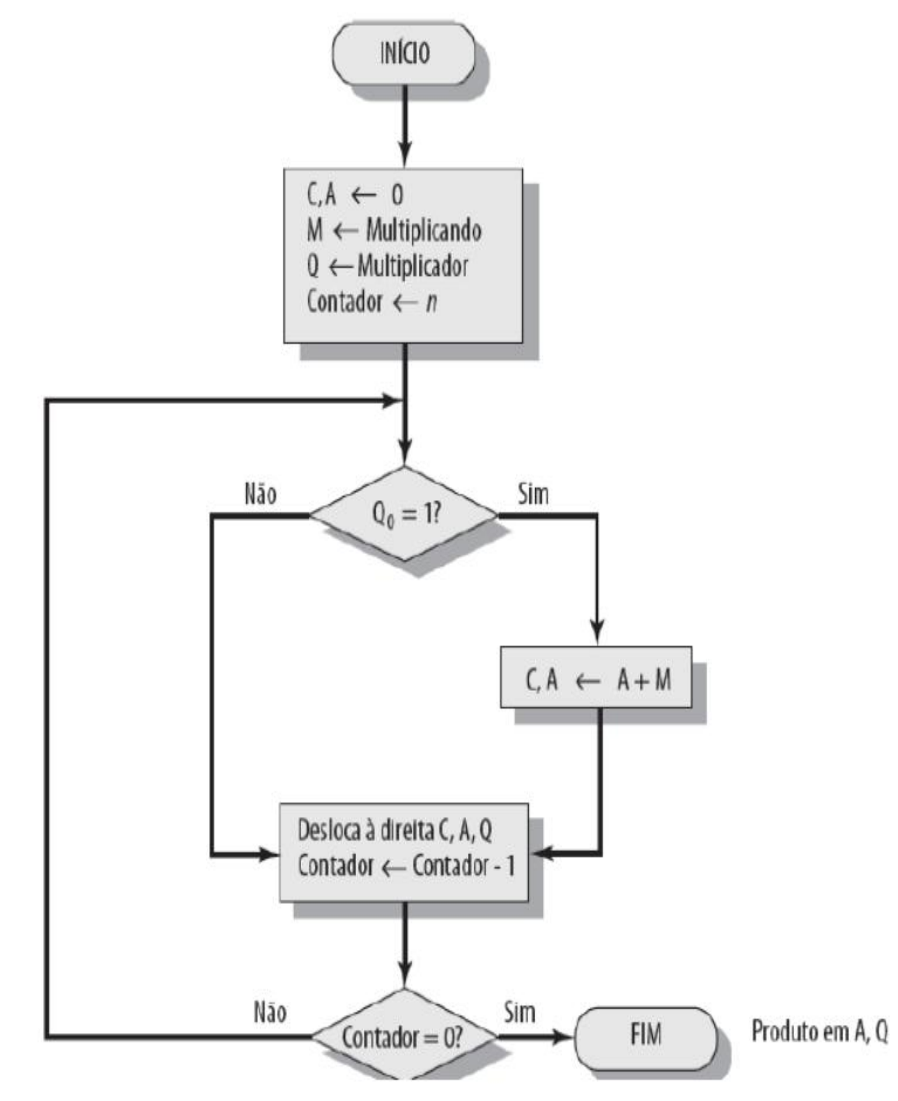
0001 0101 → deslocamento

0110 0101 → soma

0011 0010 → deslocamento

0001 1001 → deslocamento

Multiplicação de Inteiros sem Sinal

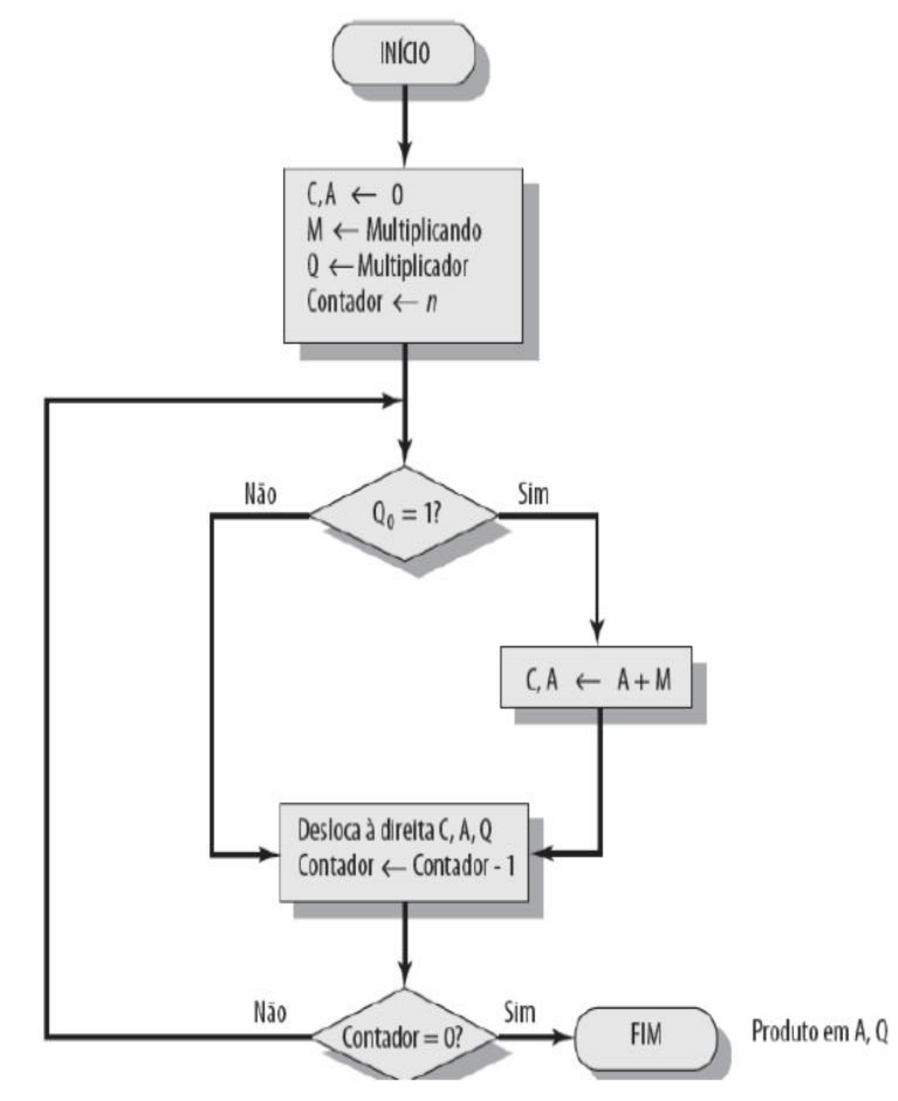


E Agora?

0100 x 1011 (4 x 11)

0000 1011

Multiplicação de Inteiros sem Sinal

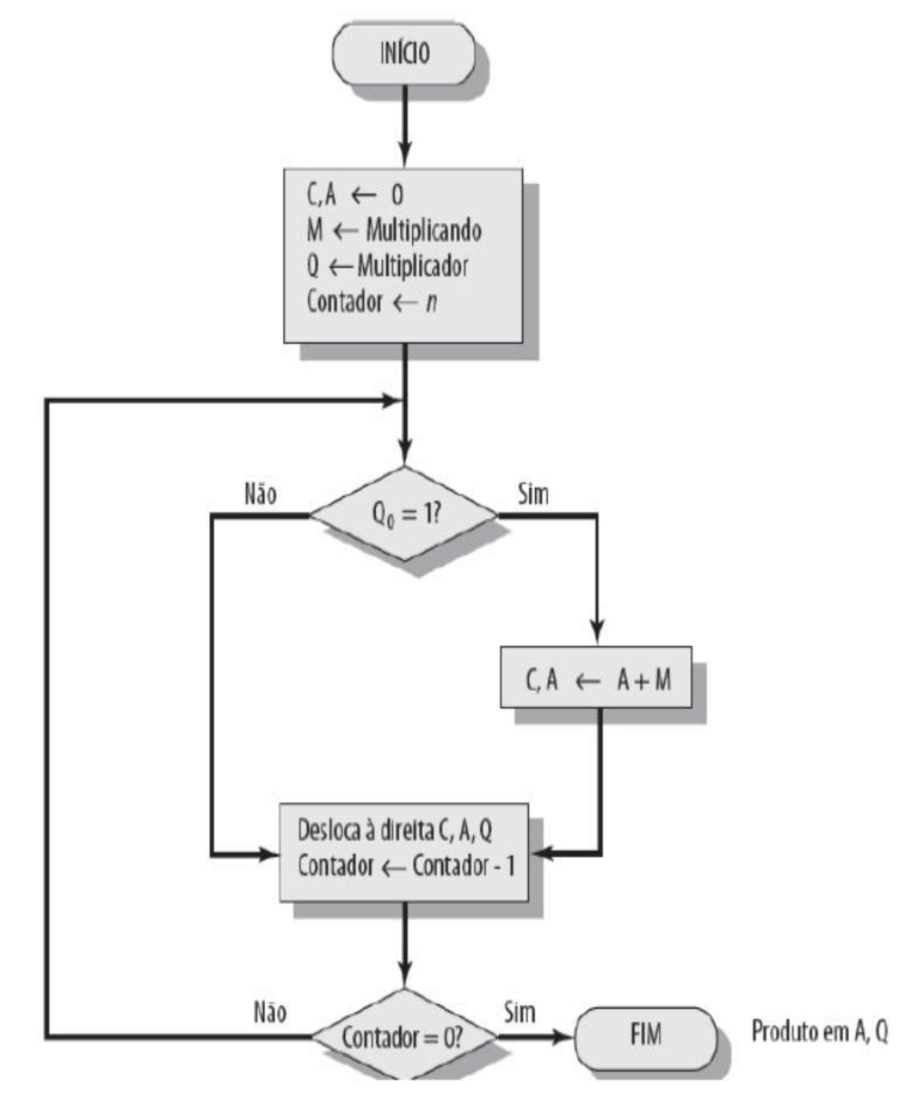


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma

Multiplicação de Inteiros sem Sinal

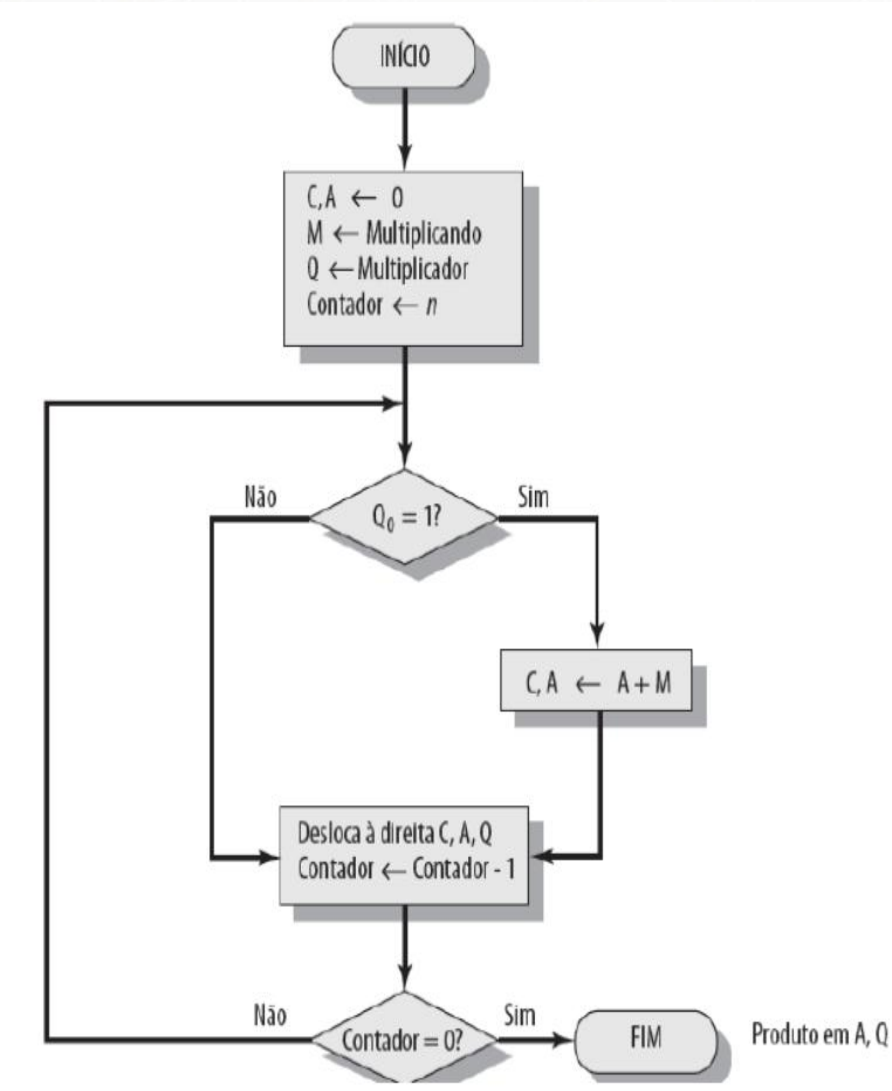


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento

Multiplicação de Inteiros sem Sinal

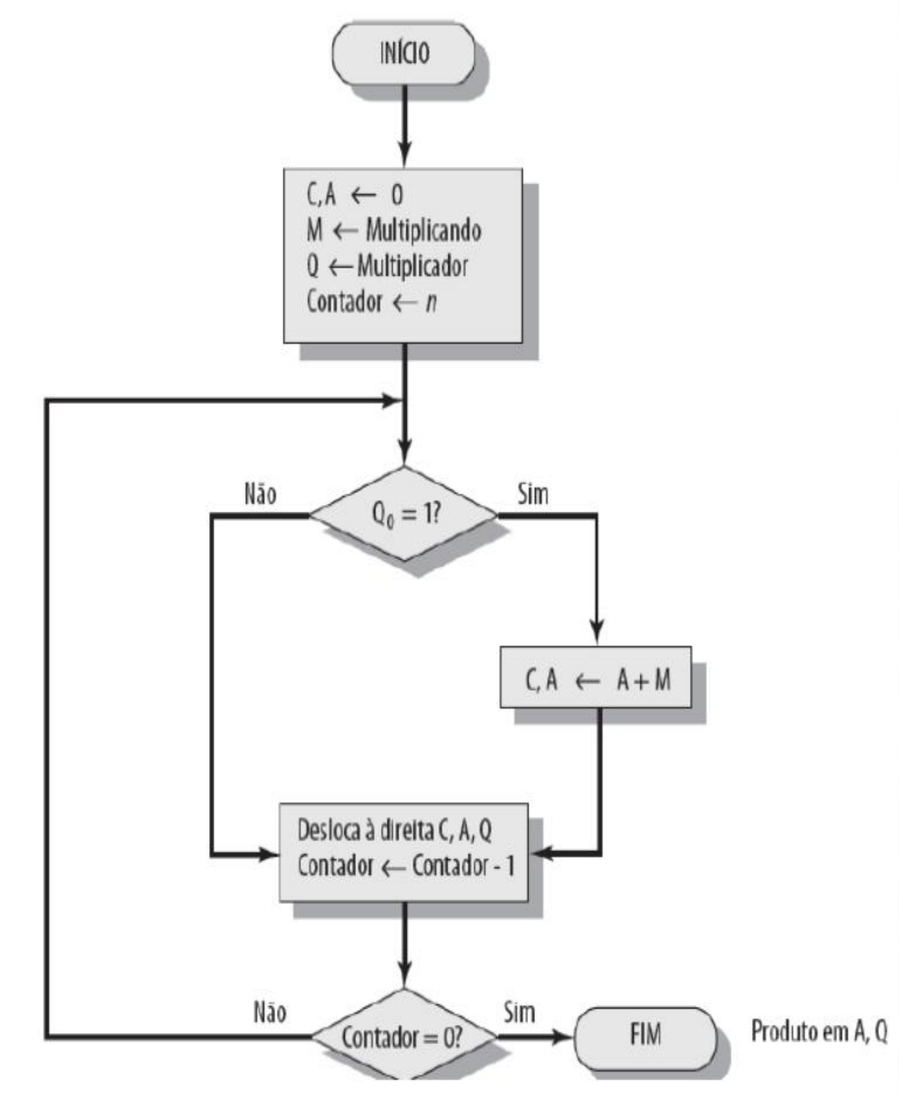


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento
0110 0101 → soma

Multiplicação de Inteiros sem Sinal

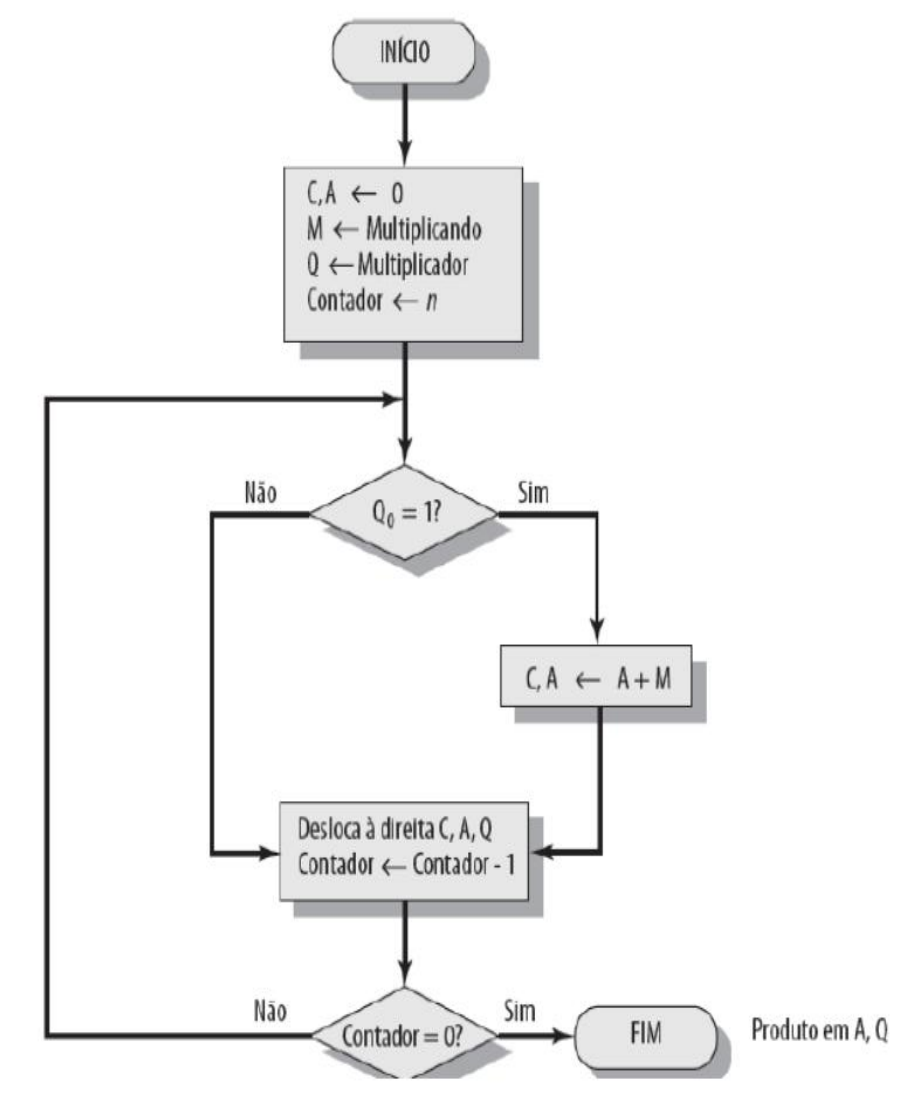


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento
0110 0101 → soma
0011 0010 → deslocamento

Multiplicação de Inteiros sem Sinal

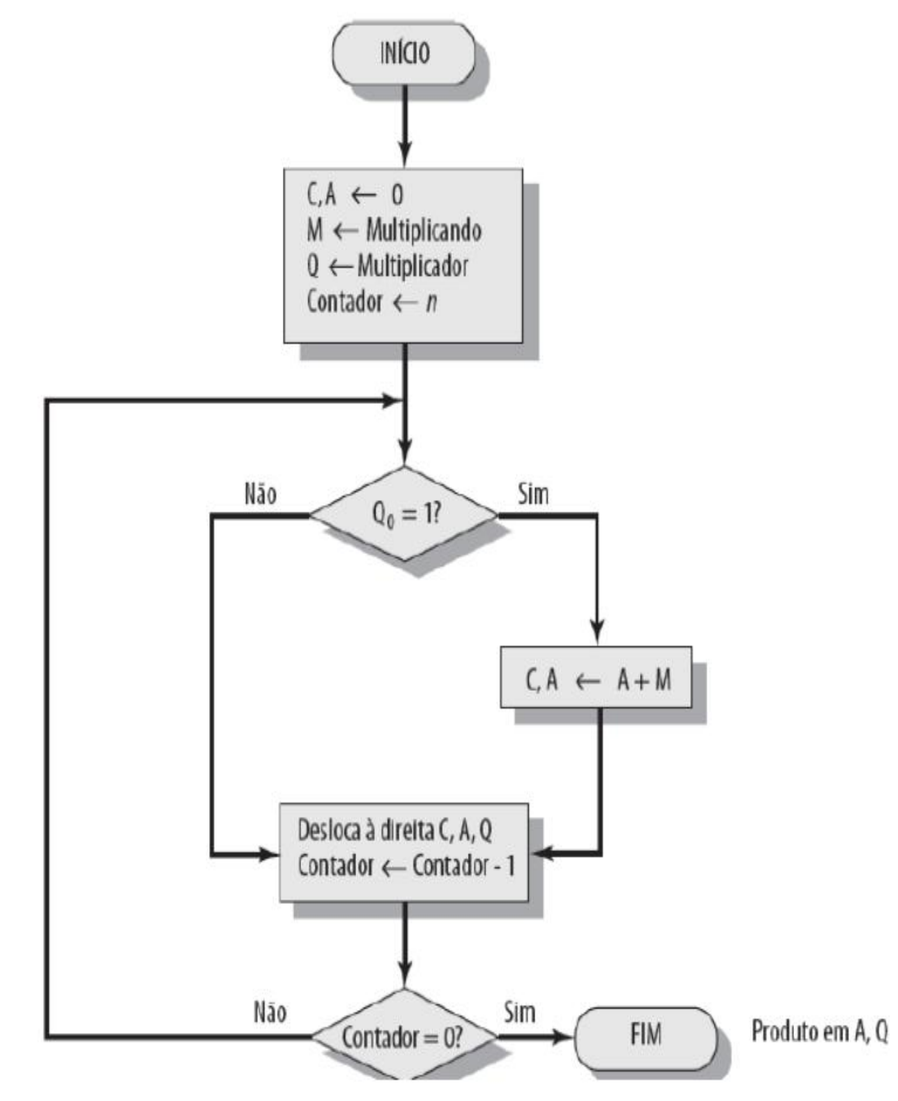


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento
0110 0101 → soma
0011 0010 → deslocamento
0001 1001 → deslocamento

Multiplicação de Inteiros sem Sinal

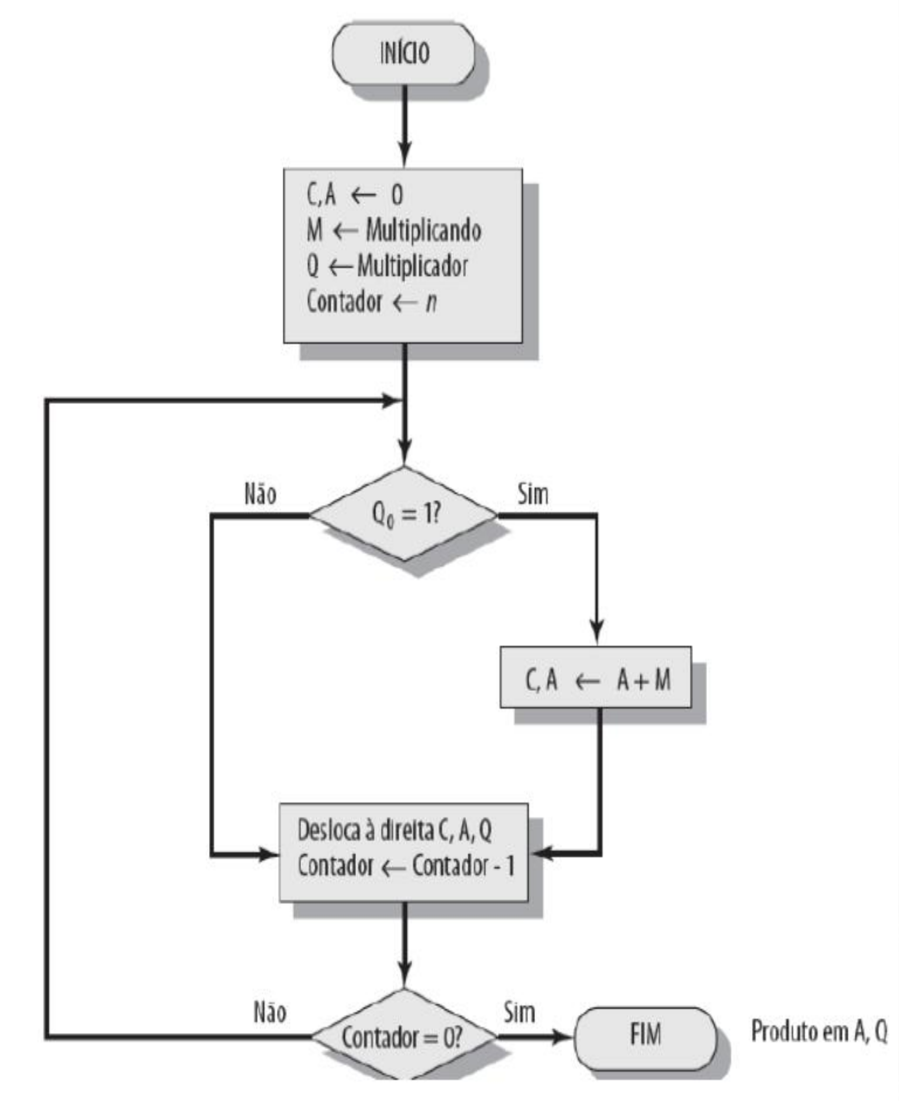


E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento
0110 0101 → soma
0011 0010 → deslocamento
0001 1001 → deslocamento
0101 1001 → soma

Multiplicação de Inteiros sem Sinal



E Agora?

0100 x 1011 (4 x 11)

0000 1011
0100 1011 → soma
0010 0101 → deslocamento
0110 0101 → soma
0011 0010 → deslocamento
0001 1001 → deslocamento
0101 1001 → soma
0010 1100 → deslocamento

Exercícios Introdutórios: Multiplicação sem Sinal

Disponível no moodle

Multiplicação – Complemento de 2

Não podemos utilizar o mesmo algoritmo para complemento de 2:

Veja:

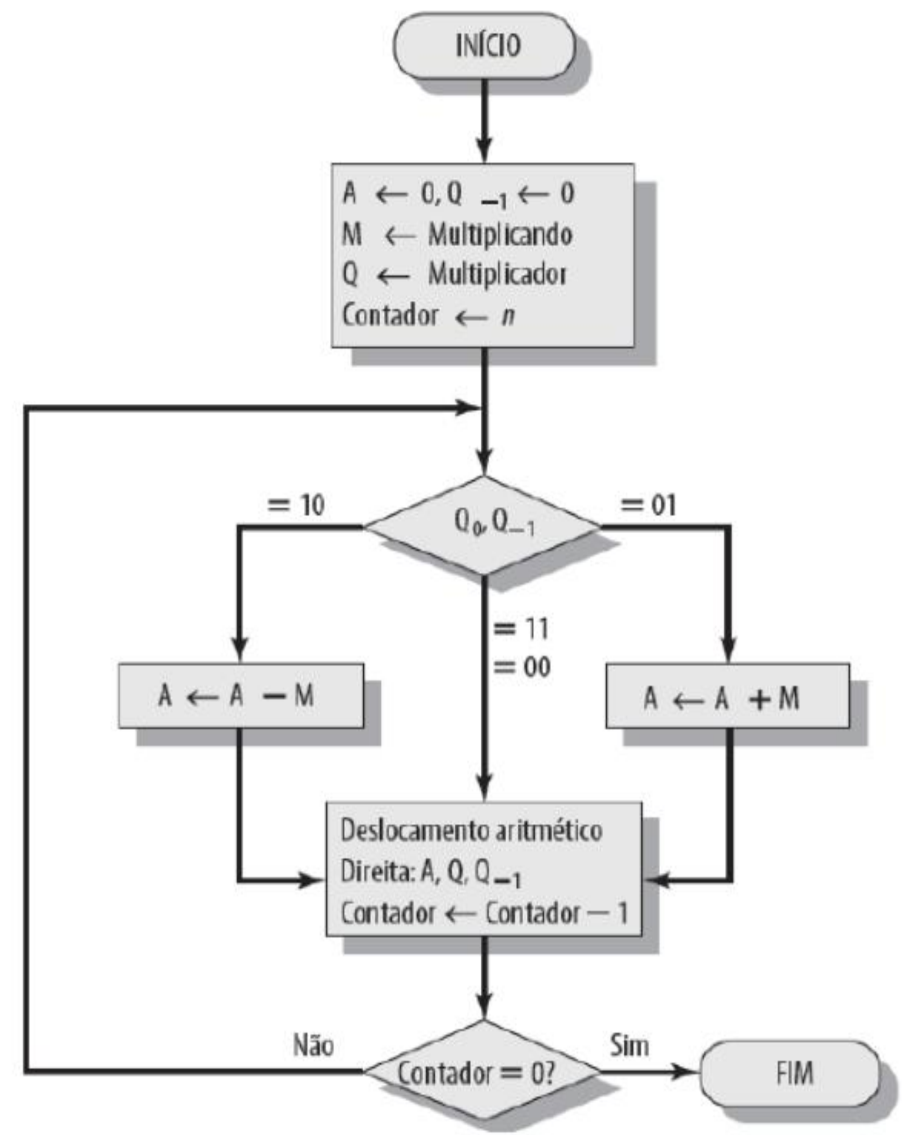
```
   1010 (-6)
x  1101 (-3)
10000010 (-126)
```

Portanto, devemos utilizar um algoritmo especial como o de Booth

Multiplicação – Complemento de 2

- Multiplicador e multiplicando são armazenados nos registradores Q e M.
- Utiliza-se um registrador de 1 bit, posicionado logicamente à direita do bit menos significativo do registrador Q, o qual é 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.
- Será examinado o bit menos significativo de 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 para a direita.
  - 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.

Multiplicação – Complemento de 2



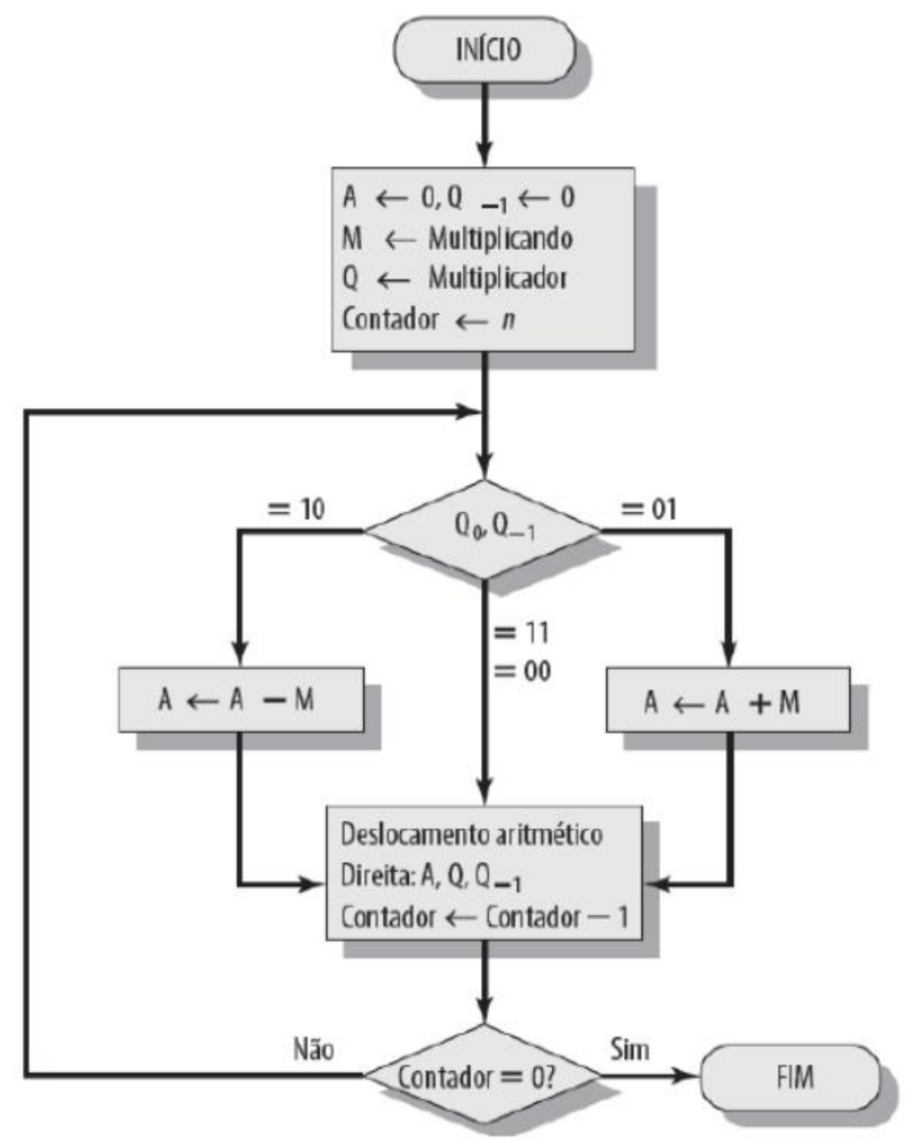
Exemplo: Faça as operações aritméticas abaixo utilizando multiplicação de números binários com complemento de 2:

1110 x 1100 (-2 x -4)

A Q Q-1

0000 1100 0 → inicial

Multiplicação – Complemento de 2

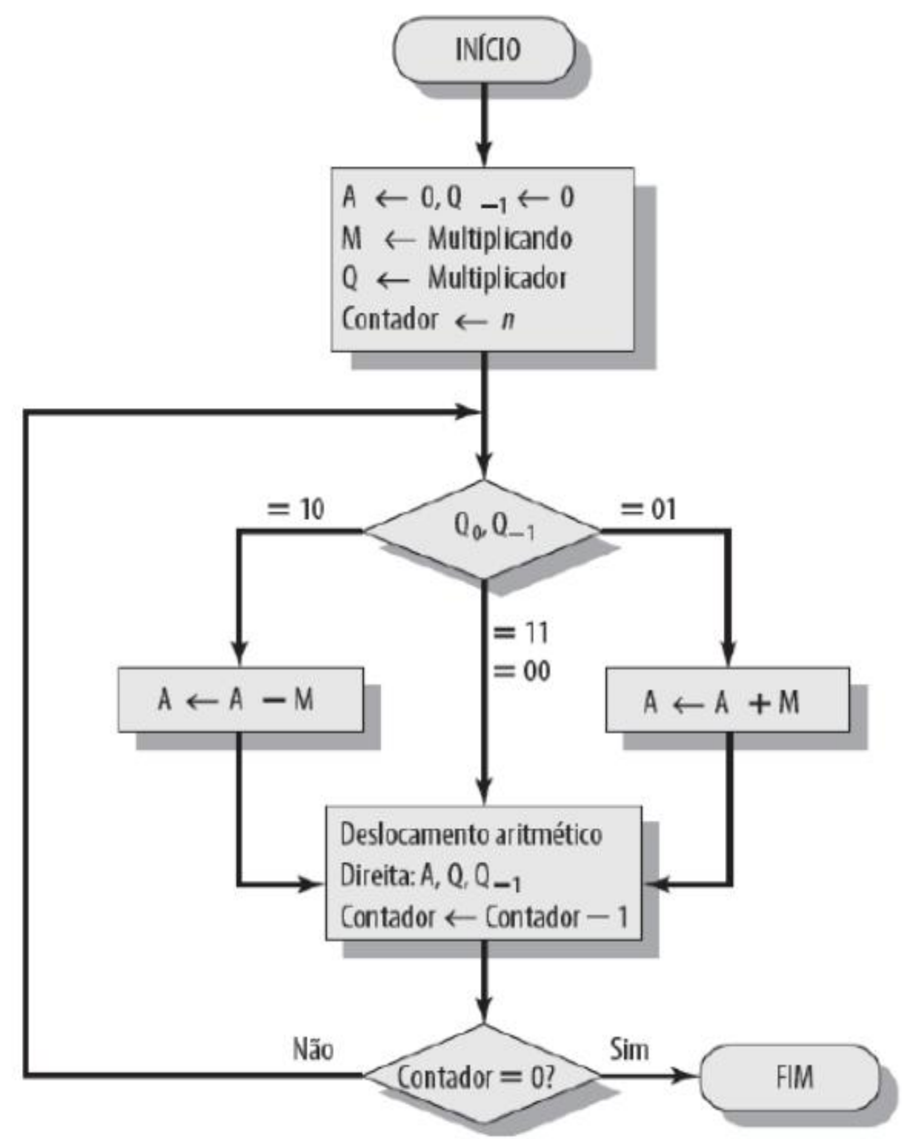


Exemplo: Faça as operações aritméticas abaixo utilizando multiplicação de números binários com complemento de 2:

1110 x 1100 (-2 x -4)

0000 1100 0 → inicial
0000 0110 0 → deslocamento
0000 0011 0 → deslocamento

Multiplicação – Complemento de 2

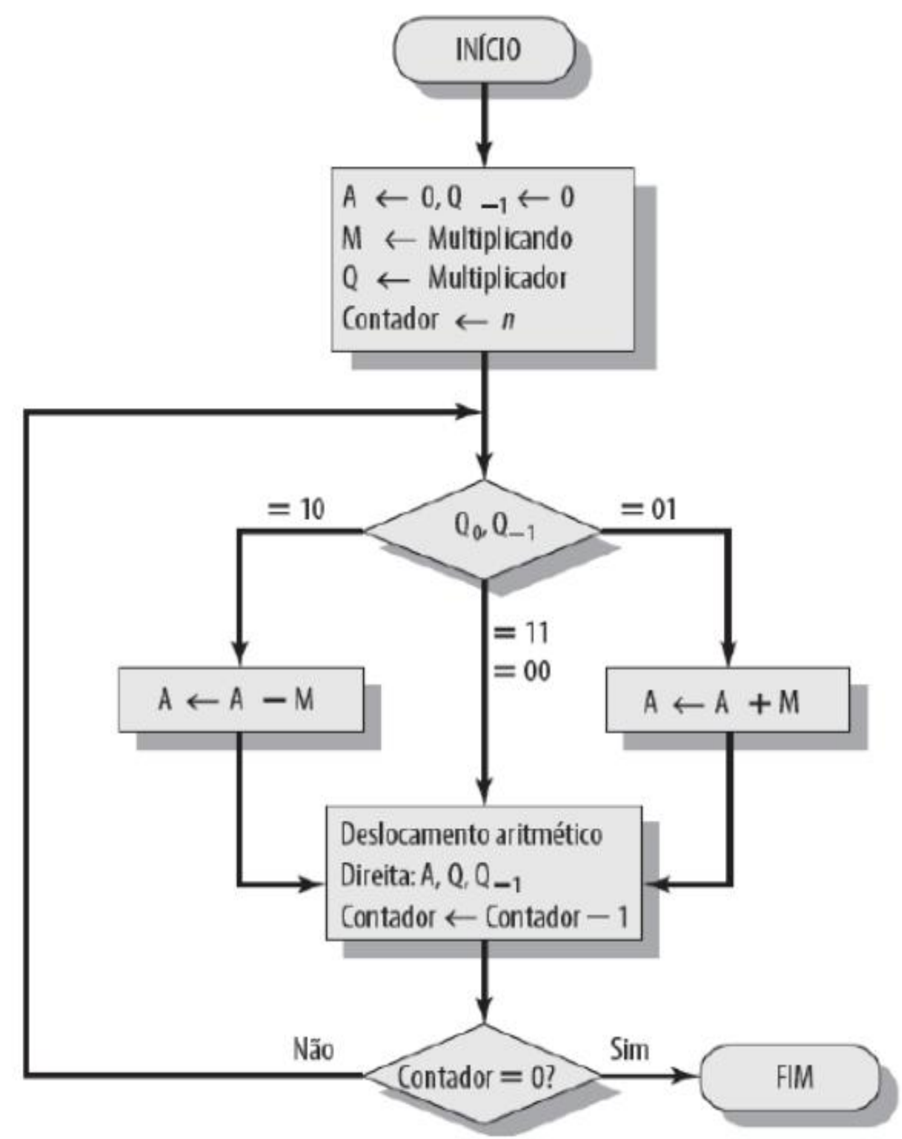


Exemplo: Faça as operações aritméticas abaixo utilizando multiplicação de números binários com complemento de 2:

1110 x 1100 (-2 x -4)

0000 1100 0 → inicial
0000 0110 0 → deslocamento
0000 0011 0 → deslocamento
0010 0011 0 → A -M
0001 0001 1 → deslocamento

Multiplicação – Complemento de 2



Exemplo: Faça as operações aritméticas abaixo utilizando multiplicação de números binários com complemento de 2:

1110 x 1100 (-2 x -4)

0000 1100 0 → inicial
0000 0110 0 → deslocamento
0000 0011 0 → deslocamento
0010 0011 0 → A -M
0001 0001 1 → deslocamento
0000 1000 1 → deslocamento
Produto

Divisão

- Realiza repetidas execuções de adição, subtração e deslocamento.
- Algoritmo
  - 1. O divisor é colocado no registrador M e o dividendo no registrador Q (sem considerar o sinal). O registrador A é iniciado com 0.
  - 2. Deslocar o conteúdo dos registradores A e Q, um bit à esquerda.
  - 3. Execute A ⟵ A - M;
  - 4. Se o sinal de A se manteve após a operação em 3, faça Q0 = 1; Senão, restaure o valor de A, ou seja A ⟵ A + M e faça Q0 = 0;
  - 5. Repita 2 a 4 de acordo com o tamanho da palavra armazenada em Q.
    - No fim:
      - O resto estará em A.
      - Se o divisor e o dividendo tiverem o mesmo sinal, o quociente estará em Q;

Divisão

$$
\begin{array}{lrl}
 & 1001_{dec} & \text{Quociente} \\
\text{Divisor } 1000_{dec} & \overline{)1001010_{dec}} & \text{Dividendo} \\
 & \underline{-1000} & \\
 & 10 & \\
 & 101 & \\
 & 1010 & \\
 & \underline{-1000} & \\
 & 10_{dec} & \text{Resto}
\end{array}
$$

Algoritmo

O algoritmo considera que o dividendo e divisor são positivos

1. O divisor é colocado no registrador M e o dividendo no registrador Q (sem considerar o sinal). O registrador A é iniciado com 0.

2. Deslocar o conteúdo dos registradores A e Q, um bit à esquerda.

3. Execute $A \leftarrow A - M$;

4. Se o sinal de A se manteve após a operação em 3, faça Q0 = 1; Senão, restaure o valor de A, ou seja $A \leftarrow A + M$ e faça Q0 = 0;

5. Repita 2 a 4 de acordo com o tamanho da palavra armazenada em Q.

   - No fim:
   - O resto estará em A.
   - Se o divisor e o dividendo tiverem o mesmo sinal, o quociente estará em Q;
   - Caso contrário, o quociente correto é o complemento de 2 do número armazenado em Q.

Divisão

- Exemplos:

Faça as operações aritméticas abaixo utilizando divisão de números binários com complemento de 2:

0111 / 0010 (7 / 2)

A Q

0000 0111 → inicial

0000 1110 → deslocamento

1110 1110 → A - M

0000 1110 → restaurar Q0 = 0

0001 1100 → deslocamento

1111 1100 → A - M

0001 1100 → restaurar Q0 = 0

0011 1000 → deslocamento

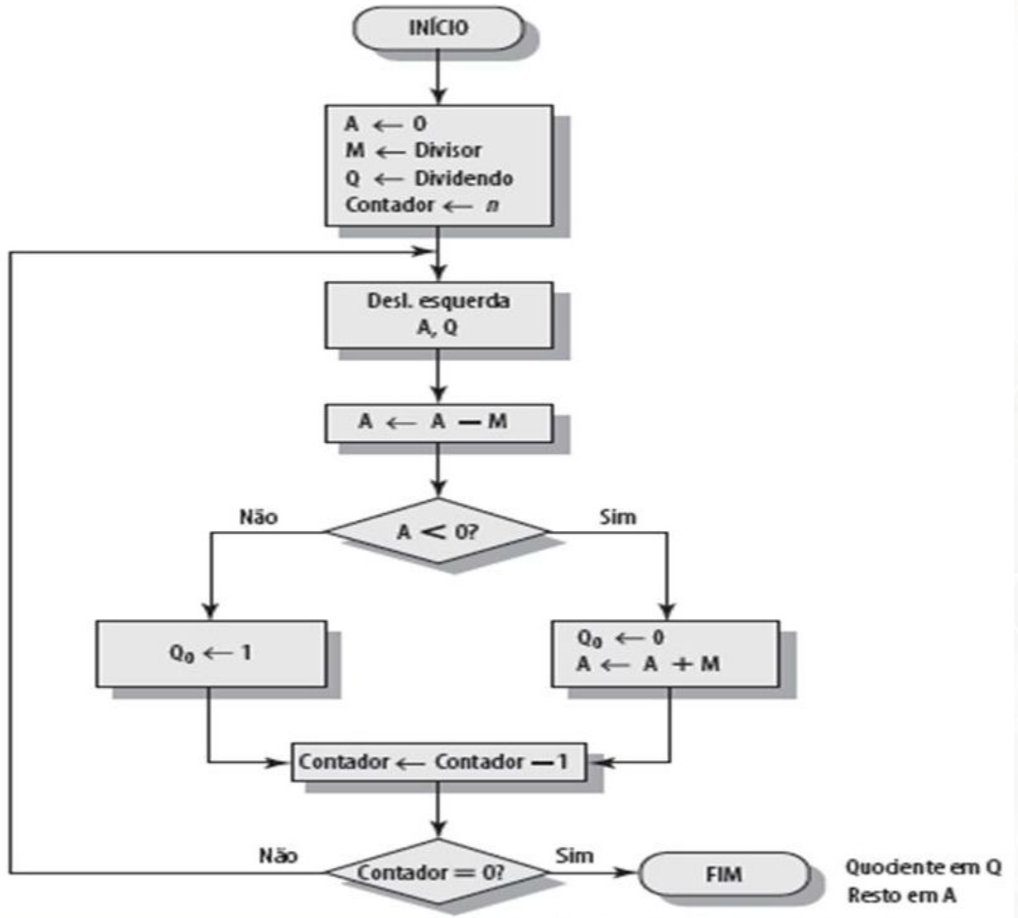
0001 1000 → A - M

0001 1001 → Q0 = 1

0011 0010 → deslocamento

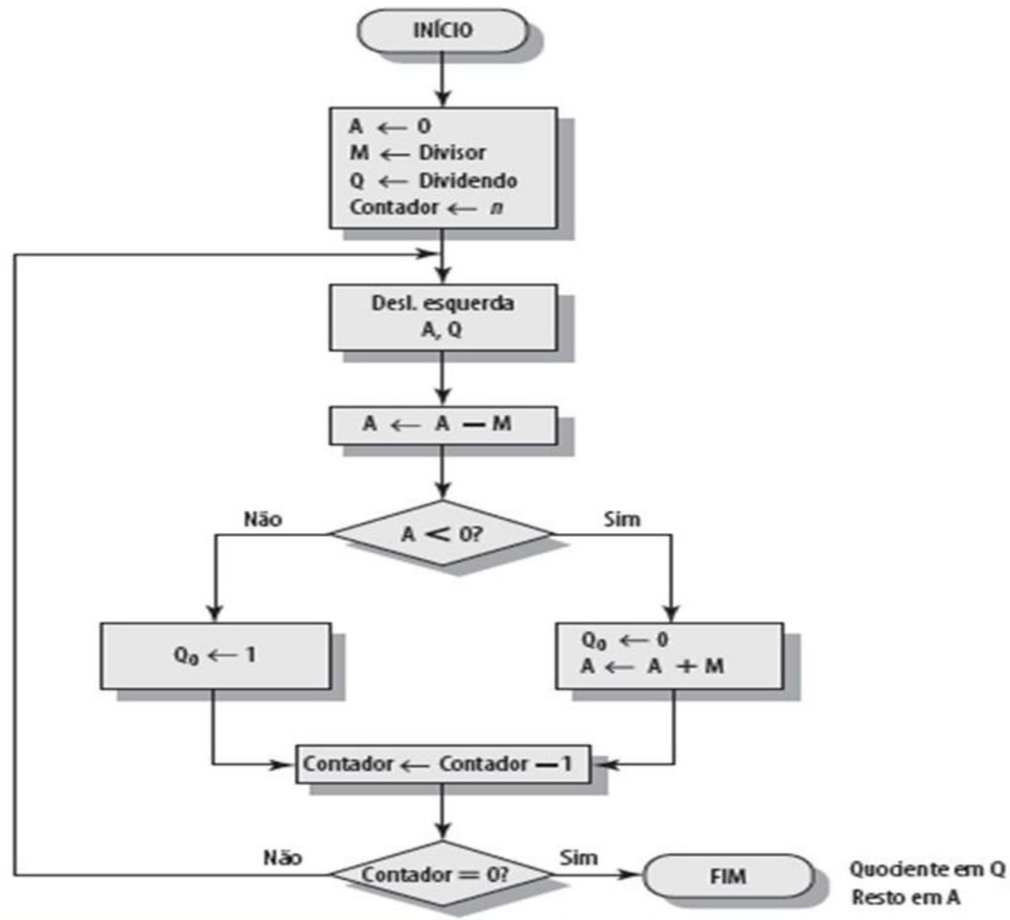
0001 0010 → A - M

0001 0011 → Q0 = 1



Divisão

1101 / 0110 (-3 / 6)
A M
0000 1101 → inicial
0001 1010 → deslocamento
1011 1010 → A - M
0001 1010 → restaurar
0011 0100 → deslocamento
1101 0100 → A - M
0011 0100 → restaurar
0110 1000 → deslocamento
0000 1000 → A - M
0000 1001 → Q 0 = 1
0001 0010 → deslocamento
1011 0010 → A - M
0001 0010 → restaurar



Exercícios Fixação: Multiplicação e Divisão

Disponível no moodle

Representação de Ponto Flutuante

- Exemplos:
  - Normalização: Um número na notação de ponto flutuante que não possui 0s à esquerda do ponto decimal.
  - Notação científica: Uma notação que apresenta números com um único dígito à esquerda do ponto decimal.
  - Exemplos:
    - $1{,}0_{dec} \times$ OK
    - $0{,}1_{dec} \times$ X
    - $10{,}0_{dec} \times$ X

Representação de Ponto Flutuante

- S: Sinal + ou -
- F: Fração (Mantissa)
- E: Expoente (Polarizado)
- B: Base

Representação de Ponto Flutuante

- S: Sinal + ou -
- F: Fração (Mantissa)
- E: Expoente (Polarizado)
- B: Base

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>s</td><td colspan="8">Expoente</td><td colspan="23">Fração</td></tr>
<tr><td>1 bit</td><td colspan="8">8 bits</td><td colspan="23">23 bits</td></tr>
</table>

Representação Binária de Ponto Flutuante

- S: Sinal + ou -
- F: Fração (Mantissa)
- E: Expoente (Polarizado)
- B: Base

- Polarizado?
    - O Intervalo de valor do expoente 0 - 255.
    - Porém a polarização subtrai do expoente o valor 127
        - Porque polarizar? Para não haver necessidade de representar expoentes negativos.
    - Intervalo verdadeiro: De -127 a +128.

Representação de Ponto Flutuante

- Exemplos:
  - 1.6328125 x 2^{20}
- Primeiro temos que descobrir quem é a mantissa 0.6328125
  - Para isso temos que usar um processo algorítmico
    - A primeira casa fracionária é o número antes da vírgula
    - Para calcular as próximas devemos subtrair da mantissa o valor fracionário e multiplicar o resultado por 2
    - Assim, achamos a próxima casa fracionária
    - Repita o processo até ZERAR a mantissa

Representação de Ponto Flutuante

- Exemplos:

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1.2656250-1) = 0.2656250 x 2 = 0.5312500

R: 1

Representação de Ponto Flutuante

- Exemplos:

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1.2656250-1) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0.5312500 - 0

R: 10

Representação de Ponto Flutuante

- Exemplos:

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1.2656250 -1) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0.5312500 - 0

0.5312500 * 2 = 1.0625000 casa fracionária = 1.0625000-1 = 0.0625000

R: 101

Representação de Ponto Flutuante

- Exemplos:

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1.2656250 -1) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0.5312500-0

0.5312500 * 2 = 1.0625000 casa fracionária = 1.0625000 - 1= 0.0625000

0.0625000 * 2 = 0.125 → 0 → 0.125 – 0 * 2 = 0.25

R: 1010

Representação de Ponto Flutuante

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1-1.2656250) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0 – 0.5312500

0.5312500 * 2 = 1.0625000 casa fracionária = 1 – 1.0625000 = 0.0625000

0.0625000 * 2 = 0.125 → 0 → 0.125 – 0 * 2 = 0.25

0,25 = 0 → 0.25 – 0 = 0.25 * 2 = 0.5

R: 10100

Representação de Ponto Flutuante

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1-1.2656250) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0 – 0.5312500

0.5312500 * 2 = 1.0625000 casa fracionária = 1 – 1.0625000 = 0.0625000

0.0625000 * 2 = 0.125 → 0 → 0.125 – 0 * 2 = 0.25

0,25 = 0 → 0.25 – 0 = 0.25 * 2 = 0.5

0.5 = 0 → 0.5 – 0 = 0.5 * 2 = 1.0

R: 101000

Representação de Ponto Flutuante

1.6328125 x 2^{20}

0.6328125 x 2 = 1.2656250 , portanto primeira casa fracionária é 1

(1-1.2656250) = 0.2656250 x 2 = 0.5312500

0.5312500 = casa fracionária = 0 →0 – 0.5312500

0.5312500 * 2 = 1.0625000 casa fracionária = 1 – 1.0625000 = 0.0625000

0.0625000 * 2 = 0.125 → 0 → 0.125 – 0 * 2 = 0.25

0,25 = 0 → 0.25 – 0 = 0.25 * 2 = 0.5

0.5 = 0 → 0.5 – 0 = 0.5 * 2 = 1.0

1.0 = 1 → 1.0 – 1 = 0(acabou)

R: 1010001 = 0,6328125

Representação de Ponto Flutuante

- Exemplos:
  - 1.6328125 x 2^{20} = + 1.1010001 x 2^{10100}

- Agora devemos encontrar o expoente
  - Para isso devemos polarizar o seu valor
    - 10100 = 20
    - 20+127 = 147
    - 147-127 = 20
    - Portanto 147 é o valor polarizado

      10010011

Representação de Ponto Flutuante

- Exemplos:
  - 1.6328125 x = + 1.1010001 x 2^{10100} = 0 10010011 10100010000000000000000

Representação de Ponto Flutuante

- Outros exemplos:

$1.6328125 \times 2^{20} = +\ 1.1010001 \times 2^{20} = 0\ 10010011\ 10100010000000000000000$

$-\ 1.6328125 \times 2^{20} = -\ 1.1010001 \times 2^{20} = 1\ 10010011\ 10100010000000000000000$

$1.6328125 \times 2^{-20} = +\ 1.1010001 \times 2^{-20} = 0\ 01101011\ 10100010000000000000000$

$-\ 1.6328125 \times 2^{-20} = -\ 1.1010001 \times 2^{-20} = 1\ 01101011\ 10100010000000000000000$

Normalização

- Normalização
  - Números de ponto flutuante geralmente são normalizados:
  - O expoente é ajustado de modo que o bit inicial da mantissa seja sempre 1.
  - Funciona da mesma forma em notação científica:
    - Os números são normalizados para um único dígito antes do ponto decimal.
    - Ex.: 3123 = 3,123 x 10^3.
  - Por ser sempre 1, não é preciso armazená-lo.

Normalização

- Exemplo:

  Normalizar 25,5

  25,5/2 = 12,75

  12,75/2 = 6,375

  6,375/2 = 3,1875

  3,1875/2 = 1,59375

  - 4 divisões

  1,59375 x 2^4 (25,5 normalizado)

Overflow e underflow

- Overflow
  - (1,110 x 2^120) x (1,010 x 2^100) = 1,00011 x 2^220
    - Não da para representar com 32 bits, por o expoente não está entre os limites -126 à 127
- Underflow
  - (1,110 x 2^-120) x (1,010 x 2^-100) = 1,00011 x 2^-220
    - No caso do underflow é possível aproximar para zero

Operações com Ponto Flutuante

- Passos para as Operações de Soma/Subtração
  - 1. Verifique zero;
  - 2. Alinhe a mantissa;
  - 3. Soma ou subtraia mantissa;
  - 4. Normalize resultado.

- A adição e a subtração são idênticas, exceto por uma mudança de sinal:
  - Se essa for uma operação de subtração, o processo começa alterando o sinal do subtraendo;
  - Em seguida, se algum operando for 0, o outro é informado como o resultado.

Operações com Ponto Flutuante

- Operações de Soma/Subtração – Alinhamento da mantissa
  - A próxima fase é manipular os números de modo que os dois expoentes sejam iguais.
  - O alinhamento é obtido deslocando repetidamente a parte de magnitude da mantissa 1 dígito para a direita, e aumentando o expoente até que os dois expoentes sejam iguais.
  - Se esse processo resultar em um valor 0 para a mantissa, então o outro número é informado como resultado.

Operações com Ponto Flutuante

- Operações de Soma/Subtração – Adição
  - As duas mantissas são somadas, podendo ocasionar:
    - 1. Resultado 0: Se as mantissas forem iguais em valor, porém diferentes em sinal.
    - 2. Overflow da mantissa por 1 dígito: A soma gerou um carry.
      - Se isso acontecer, a mantissa do resultado é deslocado para a direita e o expoente é incrementado.
    - 3. Overflow de expoente: O expoente foi incrementado e passou do valor máximo.
  - Isso seria informado e a operação encerrada.

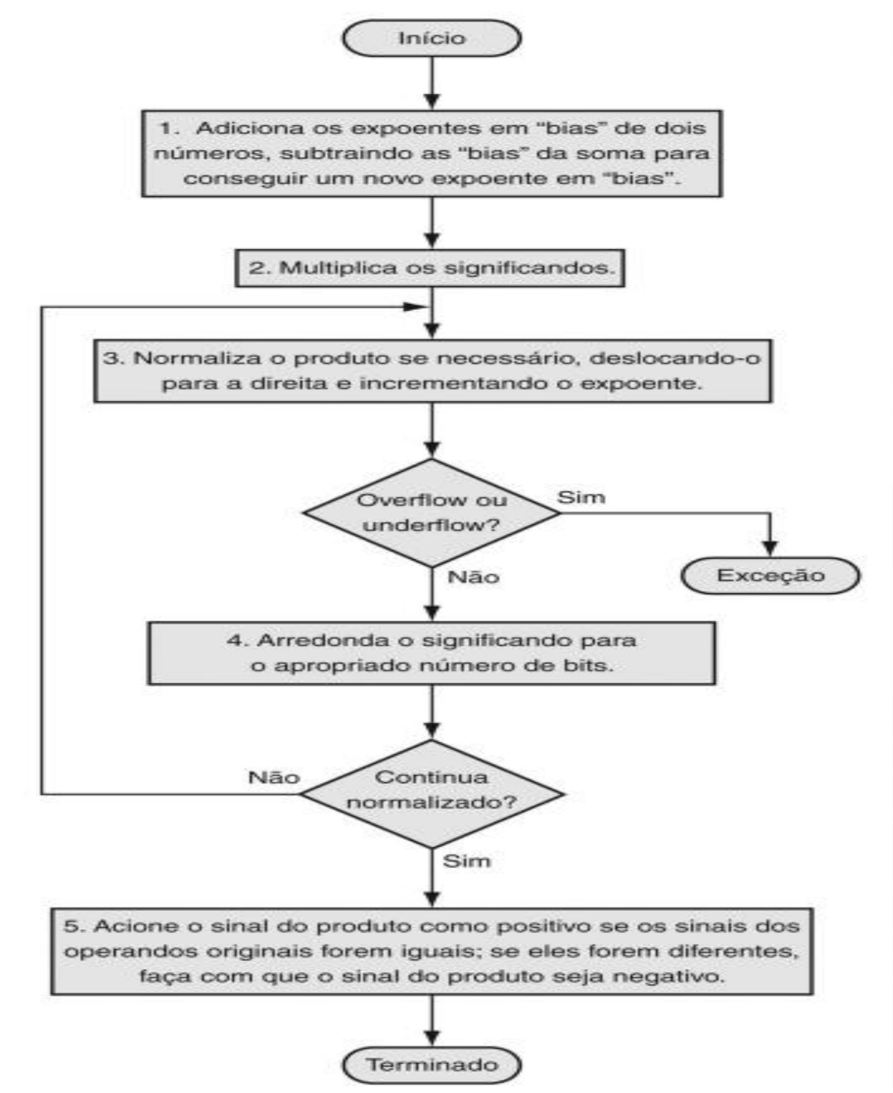
Operações com Ponto Flutuante

- Operações de Soma/Subtração – Normalização
  - A fase final normaliza o resultado.
  - Consiste no deslocamento dos dígitos da mantissa para a esquerda até que o dígito mais significativo seja diferente de zero.
  - Cada deslocamento causa um decremento do expoente e, portanto, poderá ocasionar um underflow do expoente.
  - Finalmente, o resultado poderá ser arredondado e depois informado.

Operações com Ponto Flutuante

- Passos para as Operações de Divisão/Multiplicação
  - 1. Verifique zero.
  - 2. Soma/subtraia expoentes.
  - 3. Multiplique/divida mantissa (observe sinal).
  - 4. Normalize.
  - 5. Arredonde.

Operações com Ponto Flutuante (Multiplicação e divisão)



1. Verifique zero.
2. Soma/subtraia expoentes.
3. Multiplique/divida significandos (observe sinal).
4. Normalize.
5. Arredonde.

Exercícios Fixação: Ponto Flutuante IEEE 754

Disponível no moodle

Referências

- PATTERSON, David A.; HENNESSY, John L.Organizac¸˜ao e projeto de computadores: ainterface hardware/software, 5. ed. Rio de Janeiro: Elsevier, 2017. ISBN 9788535287936.

- STALLINGS, William.; Arquitetura e Organização de Computadores. 10. ed. São Paulo:Pearson Education do Brasil, 2017. ISBN 9788543020532.

- TANENBAUM, A. S.; AUSTIN, T. Organização Estruturada de Computadores. 6. ed. SãoPaulo: Pearson Education do Brasil, 2013. ISBN 9788581435398.