

“Vocês se mudam para um lugar e se multiplicam, e se multiplicam até que todos os recursos naturais se esgotem ... [agem como] um vírus”  
(Agente Smith).

# Multiplicação em Binário

Paulo Ricardo Lisboa de Almeida

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \underline{\times 321} \end{array}$$

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \underline{\times 321} \end{array}$$

← Multiplicando  
← Multiplicador

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \times 321 \\ \hline 123 \end{array}$$

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \times 321 \\ \hline 123 \\ 2460 \end{array}$$

← Desloca uma casa para a esquerda e adiciona um zero

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \times 321 \\ \hline 123 \\ +2460 \\ \hline 36900 \\ 39483 \end{array}$$

← Desloca duas casas para a esquerda e adiciona um zero

# Relembrando

Um exemplo na base 10 para relembrar

$$\begin{array}{r} 123 \\ \times 321 \\ \hline 123 \\ +2460 \\ \hline 36900 \\ 39483 \end{array}$$

**Intuição:** O produto é **muito** maior do que o multiplicando ou o multiplicador

# Relembrando

Um exemplo na base 10 para lembrar

$$\begin{array}{r} 123 \\ \times 321 \\ \hline 123 \\ +2460 \\ \hline 36900 \\ 39483 \end{array}$$

Ao se multiplicar um valor que possui  $m$  casas por um de  $n$  casas, teremos potencialmente um resultado que ocupa  $m+n-1$  casas.

Esse raciocínio se estende para as demais bases.

**Overflows são mais prováveis!**



# Multiplicações

Precisamos compreender todas as intuições no algoritmo de multiplicação na base 10 para transferir o raciocínio para outras bases

Por que deslocamos cada resultado intermediário para a esquerda?

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline \end{array} \longrightarrow (22) * (23)$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline \end{array} \longrightarrow \begin{array}{l} (22) * (23) \\ (22) * (3+20) \end{array}$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline \end{array}$$



$$\begin{aligned} &(22) * (23) \\ &(22) * (3 + 20) \\ &(22) * (3 + (2 * 10)) \end{aligned}$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline \end{array}$$



$$\begin{aligned} &(22) * (23) \\ &(22) * (3 + 20) \\ &(22) * (3 + (2 * 10)) \\ &22 * 3 + 22 * 2 * 10 \end{aligned}$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline 66 \end{array}$$



$$\begin{aligned} & (22) * (23) \\ & (22) * (3+20) \\ & (22) * (3+(2*10)) \\ & 22*3 + 22*2*10 \end{aligned}$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline 66 \\ + 44 \\ \hline \end{array}$$



$$\begin{aligned} & (22) * (23) \\ & (22) * (3+20) \\ & (22) * (3+(2*10)) \\ & 22*3 + 22*2*10 \end{aligned}$$

# Outro exemplo

Outro exemplo na base 10

$$\begin{array}{r} 22 \\ \times 23 \\ \hline 66 \\ + 440 \\ \hline 506 \end{array}$$



$$\begin{aligned} & (22) * (23) \\ & (22) * (3+20) \\ & (22) * (3+(2*10)) \\ & 22*3 + 22*2*10 \end{aligned}$$



# Em Binário

O raciocínio em binário (ou em qualquer outra base) é o mesmo

Realizar as multiplicações, deslocar os bits, e depois somar os resultados intermediários

Esse algoritmo é o usado em nossas máquinas

O hardware **que geralmente** implementa esse algoritmo

Em hardwares mais sofisticados

As multiplicações dos bits individuais é feita em paralelo para economizar tempo

Mas o algoritmo continua sendo o mesmo

# Exemplo

Considere os valores na base 2 (base ocultada para facilitar a visualização)

$$\begin{array}{r} 11101 \\ \times 11110 \\ \hline \end{array}$$

# Exemplo

Considere os valores na base 2 (base ocultada para facilitar a visualização)

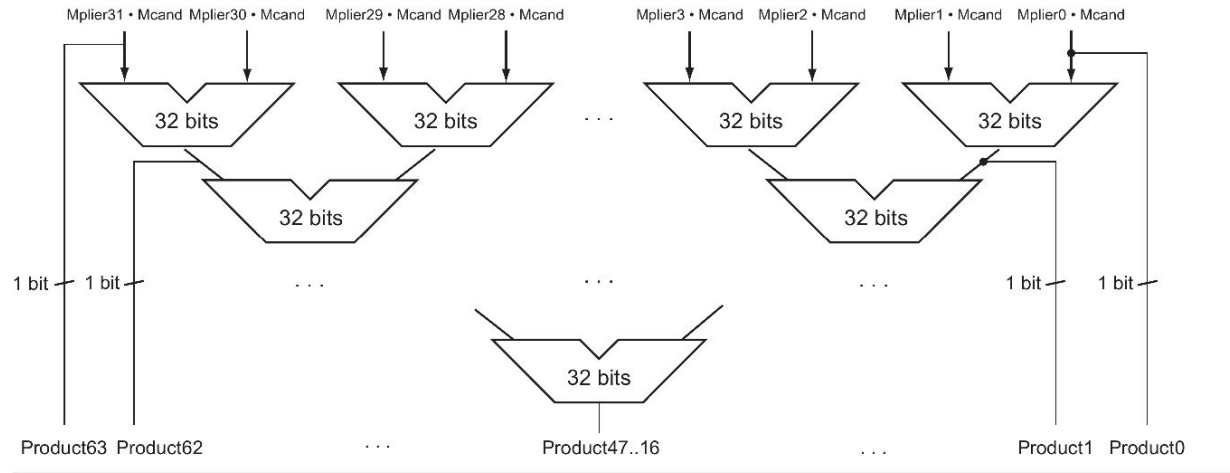
$$\begin{array}{r} 11101 \\ \times 11110 \\ \hline 00000 \\ 11101 \\ 11101 \\ 11101 \\ + \underline{11101} \end{array}$$

# Exemplo

Considere os valores na base 2 (base ocultada para facilitar a visualização)

$$\begin{array}{r} 11101 \\ \times 11110 \\ \hline 00000 \\ 11101 \\ 11101 \\ 11101 \\ + \underline{11101} \\ 1101100110 \end{array}$$

# Em Paralelo

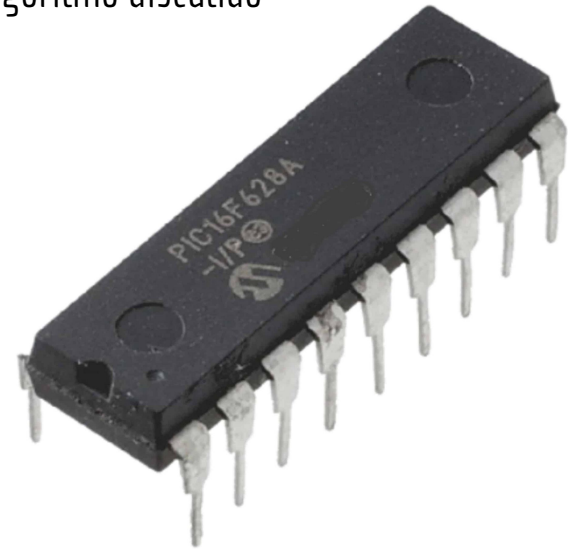


Hennessy, Patterson (2014)

# Observações

Alguns processadores, principalmente para sistemas embarcados (ex.: o PIC16F628a), não possuem instruções de multiplicação via hardware

Se você precisar multiplicar, precisa criar um programa (ex.: função) que executa o algoritmo discutido



# Observações

Realizar divisões em binário seguem raciocínios semelhantes

Implementamos os mesmos métodos utilizados na base 10

# Exercícios

1. Realize as seguintes multiplicações:

a.  $11111_2 * 1100_2$

b.  $111100_2 * 111110_2$

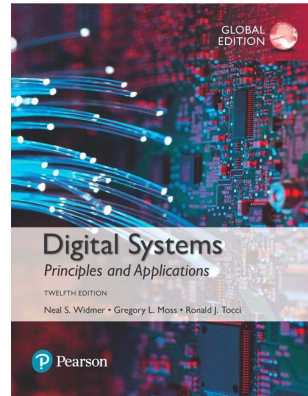
c.  $100000_2 * 11_2$

d.  $111111_2 * 1000_2$

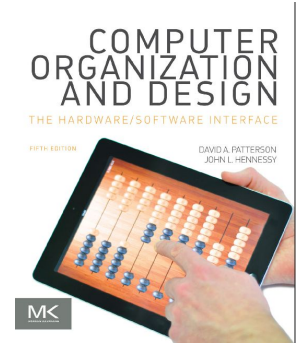


# Referências

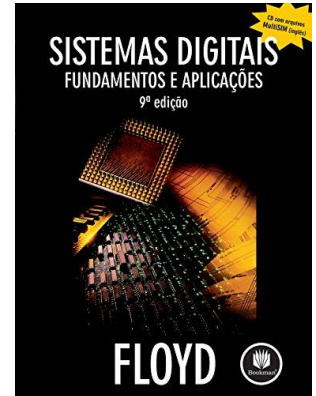
Ronald J. Tocci, Gregory L. Moss, Neal S. Widmer. Sistemas digitais. 10a ed. 2017.



Hennessy, J. L., Patterson, D. A. Computer Organization and Design: The Hardware/Software Interface. 2014.



Thomas Floyd. Widmer. Sistemas Digitais: Fundamentos e Aplicações. 2009.



# Licença

Este obra está licenciada com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).

