

Calculadora em Assembly x86  
Escola Superior de Tecnologia de Tomar  
Arquitetura de Computadores

Rúben Cardoso      Rodrigo Serra

Janeiro, 2022

# Resumo

No âmbito da unidade curricular Arquitetura de Computadores, foi solicitado um algoritmo em Assembly x86 (desenvolvido utilizando o Emu8086) que fosse capaz de calcular Divisões e Raízes Quadradas, utilizando valores fornecidos pelo utilizador através de uma interface gráfica que permitisse a escolha entre ambos os algoritmos previamente mencionados. O utilizador teria ainda a opção de optar por utilizar o teclado ou seleccionar botões na interface de forma a escolher os valores pretendidos.

***Palavras-chave*** — Assembly x86, Divisão Inteira, Raiz Quadrada, Emu8086

# Conteúdo

<b>1</b>	<b>Interface Gráfica</b>	<b>3</b>
1.1	Objetivo . . . . .	3
1.2	Pseudocódigo . . . . .	3
1.2.1	Tela inicial . . . . .	3
1.2.2	Tela da Divisão . . . . .	3
1.2.3	Tela da Raiz . . . . .	3
<b>2</b>	<b>Algoritmo da Divisão</b>	<b>5</b>
2.1	Objetivo . . . . .	5
2.2	Pseudocódigo . . . . .	5
2.3	Fluxograma . . . . .	5
<b>3</b>	<b>Algoritmo da Raiz Quadrada</b>	<b>6</b>
3.1	Objetivo . . . . .	6
3.2	Pseudocódigo . . . . .	6
3.3	Fluxograma . . . . .	7

# Capítulo 1

## Interface Gráfica

### 1.1 Objetivo

Para permitir a obtenção dos dados escolhidos pelo utilizador foi necessário utilizar uma interface gráfica capaz de o fazer. Esta interface deveria permitir a escolha entre os dois algoritmos utilizando ou o rato ou o teclado como meio de input. A interface iria ainda precisar de mostrar os resultados dos algoritmos após a sua execução.

### 1.2 Pseudocódigo

#### 1.2.1 Tela inicial

1. Imprimir um texto introdutório que indique o propósito do programa
2. Desenhar os quadrados que representam botões onde o utilizador poderá clicar
3. Ler o input do teclado que permite ao utilizador escolher se pretende utilizar o teclado ou o rato tanto para introduzir os inputs, como para escolher qual algoritmo pretende executar

#### 1.2.2 Tela da Divisão

1. Desenhar os 14 botões necessários para acomodar os inputs possíveis
2. Escrever o texto dentro dos botões todos os inputs possíveis<sup>I</sup>
3. Mostrar o texto "Dividendo", de forma a indicar ao utilizador que pode inserir o valor pretendido
4. Ler o teclado e o rato e adicionar os valores inseridos a um array
5. Mostrar o array do passo acima consoante o utilizador adiciona ou remove valores
6. Mostrar o texto "Divisor", de forma a indicar ao utilizador que pode inserir o valor pretendido
7. Ler o teclado e o rato e adicionar os valores inseridos a um array
8. Mostrar o array do passo acima consoante o utilizador adiciona ou remove valores
9. Executar o algoritmo da divisão e mostrar o resultado juntamente com o texto "Resultado"

#### 1.2.3 Tela da Raiz

- 1.

---

<sup>I</sup>Algarismos de 0-9, '-', ',', '.', backspace, confirmação

## Capítulo 2

# Algoritmo da Divisão

### 2.1 Objetivo

Este algoritmo pretende receber as variáveis, dividendo, e divisor (podendo estas ser negativas) e em seguida executar o algoritmo da divisão de forma a devolver o resultado da divisão, bem como, o resto da mesma.

### 2.2 Pseudocódigo

1. Inicialização das variáveis necessárias<sup>I</sup>
2. Retirar o primeiro HighOrder do dividendo e atribuir o seu valor à variável Resto
3. Iterar as vezes necessárias até a operação  $i^{II} \times Divisor > Resto$
4. Após a condição ser satisfeita:
  - (a) Utilizar o valor atual da variável  $i$  caso,  $i \times Divisor = Resto$
  - (b) Realizar o cálculo  $i = i - 1$ , caso,  $i \times Divisor > Resto$
5. Concatenar o valor de  $i$  à variável Quociente
6. Verificar se existem mais algarismos no dividendo
  - (a) Caso existam, voltar ao passo 3. com o novo valor retirado do dividendo
  - (b) Caso não existam, obter o valor do resultado a partir da expressão,  
 $Resto = Resto - (i \times Divisor)$

### 2.3 Fluxograma

---

<sup>I</sup>Este pseudocódigo assume que as variáveis Divisor e Dividendo já foram obtidas através da interface gráfica

<sup>II</sup>A variável  $i$  é utilizada como variável de iteração em ciclos

## Capítulo 3

# Algoritmo da Raiz Quadrada

### 3.1 Objetivo

Este algoritmo tem como objetivo receber o valor do Radicando e em seguida calcular a raiz quadrada do valor inserido, e devolver o seu resultado.

### 3.2 Pseudocódigo

1. Inicializar as variáveis necessárias<sup>I</sup>
2. Utilizar divisões consecutivas por 10 de forma a obter o número de dígitos do Radicando e armazenar o resultado em *nAlgarismos*
3. Obter o número de pares possíveis através da formula,  $nGrupos = \frac{nAlgarismos}{2}$ 
  - (a) Caso o resto da divisão do passo anterior seja igual a 1, realizar  $nGrupos = nGrupos + 1$
4. Verificar o valor de *nGrupos*
  - (a) Caso existam múltiplos grupos, ou seja  $nGrupos > 1$ , obter o grupo de HighOrder através da fórmula,  $HighOrder = \frac{Radicando}{10^{(nGrupos)}}$
  - (b) Caso exista um único grupo atribuir o valor do Radicando ao HighOrder
5. Caso seja a primeira iteração, realizar  $aux \times aux^{II}$  de forma a encontrar o maior número possível que não ultrapasse o HighOrder e armazenar a variável valor da variável *aux* na variável *Elevado*
6. Caso não seja a primeira iteração, concatenar o novo HighOrder à variável *aux*
  - (a) Caso seja a primeira iteração saltar para o passo 9.
7. Calcular  $(2 \times Elevado : a) \times a$ , incrementado o valor de *a* enquanto o resultado não for superior a *aux*
8. Concatenar o valor de *a* obtido no passo anterior à variável *Elevado*
9. Parar a execução do algoritmo caso  $nGrupos = 1$  e armazenar o valor de *Elevado* na variável *Resultado*
10. Calcular  $Aux = HighOrder - Elevado^2$
11. Calcular  $Radicando = Radicando - (HighOrder \times 10^{nGrupos})$ , de forma a remover o grupo mais à esquerda.
12. Saltar para o passo 4.

---

<sup>I</sup>Este pseudocódigo assume que a variável Radicando já foi obtida através da interface gráfica

<sup>II</sup>Variável auxiliar inicializada a 0

### 3.3 Fluxograma