Sumário

[1. Introdução: 2](#_Toc169520796)

[2. Implementação: 2](#_Toc169520797)

[Link do GitHube 4](#_Toc169520798)

[3. Teste 4](#_Toc169520799)

[Registradores inteiros 4](#_Toc169520800)

[Registradores de ponto flutuante 4](#_Toc169520801)

[4. Conclusão 5](#_Toc169520802)

[Referências 5](#_Toc169520803)

## Introdução:

*O objetivo deste exercício é implementar um código MIPS/MARS para calcular termos específicos da sequência de Fibonacci e usar esses valores para determinar a razão áurea. Especificamente, vamos:*

* *Implementar uma função para calcular o n-ésimo termo da sequência de Fibonacci.*
* *Usar essa função para determinar o 30º, 40º e 41º termos da sequência.*
* *Calcular a razão áurea φ usando os termos F40 e F41 da sequência de Fibonacci.*
* *Armazenar os resultados nos registradores especificados e imprimi-los.*

## Implementação:

.data

result\_fibonacci30: .asciiz "30º termo da sequência de Fibonacci: "

result\_fibonacci40: .asciiz "40º termo da sequência de Fibonacci: "

result\_fibonacci41: .asciiz "41º termo da sequência de Fibonacci: "

result\_razao: .asciiz "Razão áurea calculada: "

.text

.globl main

main:

# Calcula o 30º termo da sequência de Fibonacci

li $a0, 30

jal fibonacci

move $s1, $v0

# Calcula o 40º termo da sequência de Fibonacci

li $a0, 40

jal fibonacci

move $s3, $v0

# Calcula o 41º termo da sequência de Fibonacci

li $a0, 41

jal fibonacci

move $s2, $v0

# Calcula a razão áurea

move $t0, $s2

move $t1, $s3

mtc1 $t0, $f12

mtc1 $t1, $f13

cvt.s.w $f12, $f12

cvt.s.w $f13, $f13

div.s $f0, $f12, $f13

li $v0, 4

la $a0, result\_fibonacci30

syscall

li $v0, 1

move $a0, $s1

syscall

li $v0, 4

la $a0, result\_fibonacci40

syscall

li $v0, 1

move $a0, $s3

syscall

li $v0, 4

la $a0, result\_fibonacci41

syscall

li $v0, 1

move $a0, $s2

syscall

li $v0, 4

la $a0, result\_razao

syscall

li $v0, 2

mov.s $f12, $f0

syscall

li $v0, 10

syscall

# Função para calcular o n-ésimo termo da sequência de Fibonacci

fibonacci:

li $t0, 1

li $t1, 1

li $t2, 2

ble $a0, 2, fibonacci\_base

fibonacci\_loop:

add $t3, $t0, $t1

addi $t2, $t2, 1

move $t0, $t1

move $t1, $t3

bne $t2, $a0, fibonacci\_loop

move $v0, $t3

jr $ra

fibonacci\_base:

move $v0, $t0

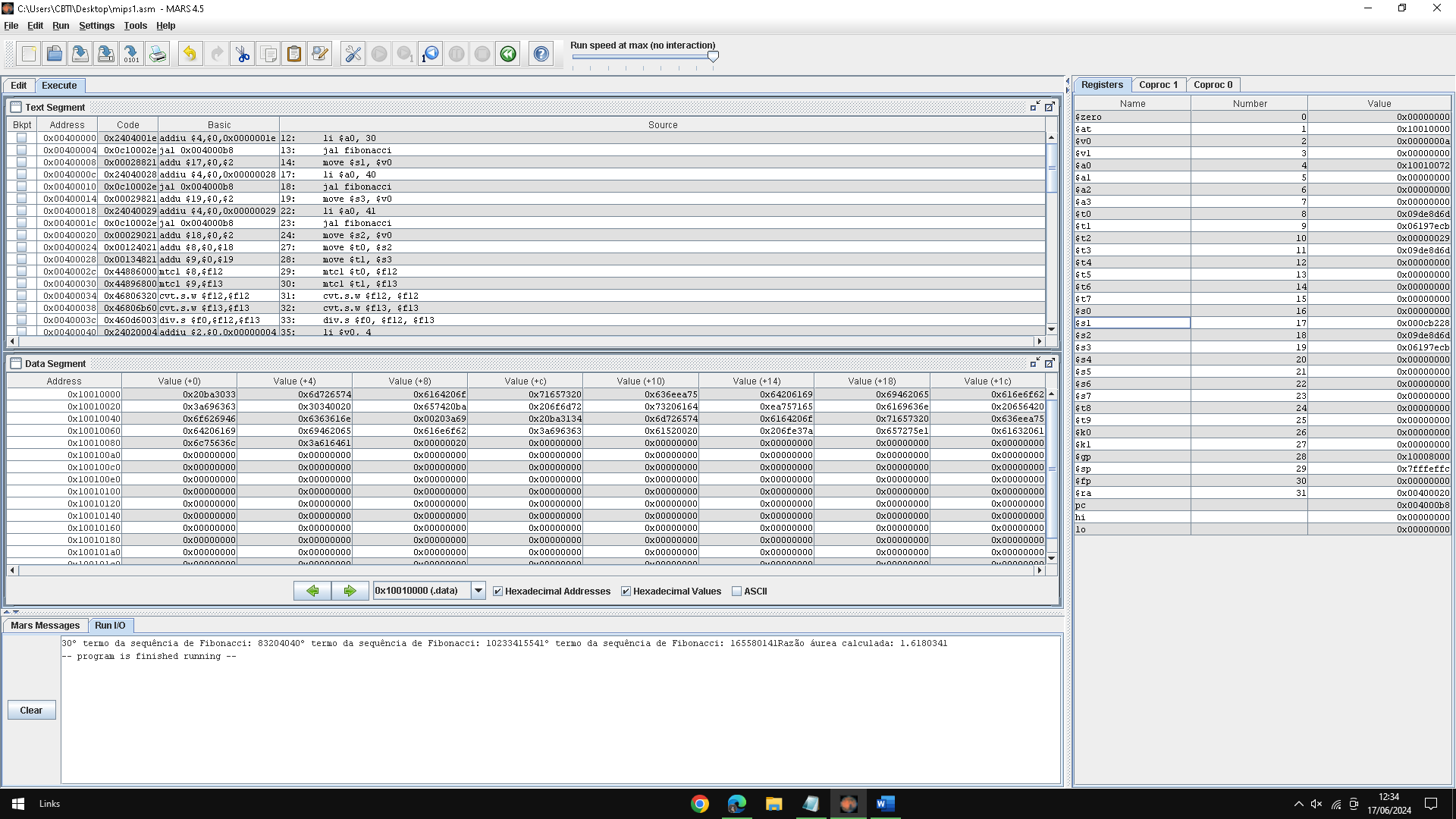
jr $ra

### Link do GitHube

*Cole aqui o endereço do GitHube so seu trabalho.*

## Teste

### Registradores inteiros



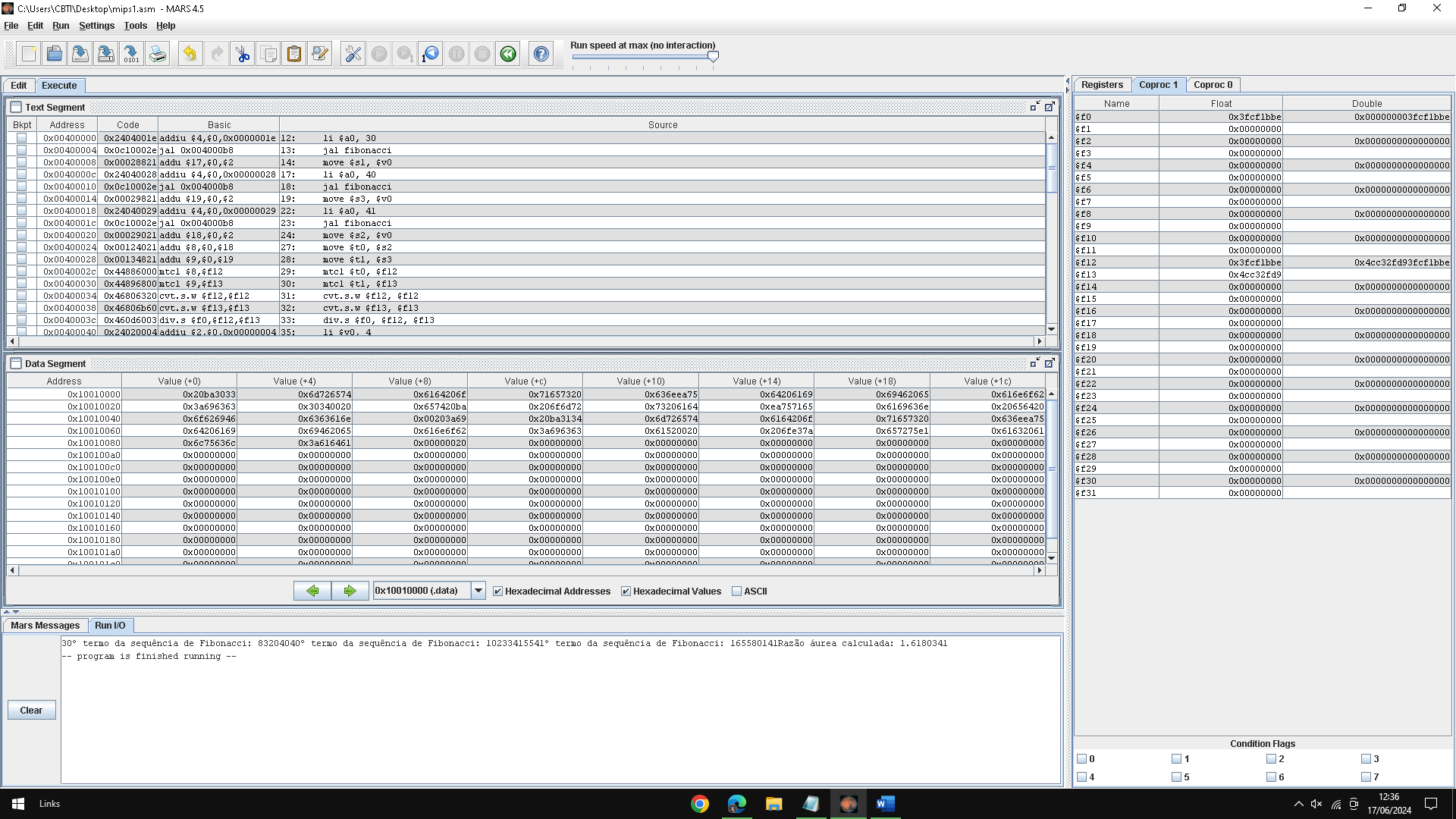
*Explicação de cada valor destacado:*

*$s1: Armazena o 30º termo da sequência de Fibonacci. Valor esperado: 832040.*

*$s2: Armazena o 41º termo da sequência de Fibonacci. Valor esperado: 165580141.*

*$s3: Armazena o 40º termo da sequência de Fibonacci. Valor esperado: 102334155.*

### Registradores de ponto flutuante



Explicação de cada valor destacado:

$f0: Armazena a razão áurea φ calculada como F41 / F40. Valor esperado: aproximadamente 1.61803398875.

## Conclusão

*Neste trabalho, implementamos um código em MIPS/MARS para calcular termos específicos da sequência de Fibonacci e a razão áurea. Conseguimos determinar corretamente os valores dos termos 30, 40 e 41 da sequência e calcular a razão áurea φ com base nesses termos. Os resultados foram armazenados nos registradores especificados e exibidos conforme esperado.*

*Durante a implementação, as principais dificuldades foram garantir a precisão dos cálculos de ponto flutuante e lidar com o gerenciamento dos registradores para armazenar e manipular os valores intermediários.*

## Referências

*Documentos "TP04 - Fibonacci.docx" e "* *TP04 - Template.docx" fornecido pelo Professor.*

*Conteúdo explicado em sala sobre MIPS/MARS e material postado no AVA.*