

# Exercício 01

## Nome do aluno:

Fábio Volkmann Coelho

## Objetivo

Executar o primeiro exemplo de programa em linguagem de montagem RISC-V.

## Instruções

1. Abra o simulador de linguagem RISC-V.
2. No editor de texto do simulador, transcreva o código abaixo:

```
# -----
# Exercício 01 - Patterson pags. 54/55/56 (versão RISC-V)
# Expressão em C: f = (g + h) - (i + j)
# -----  
  
.text  
  
main:  
    addi t1, zero, 4      # g = 4  
    addi t2, zero, 3      # h = 3  
    addi t3, zero, 2      # i = 2  
    addi t4, zero, 1      # j = 1  
  
    add  t5, t1, t2       # t5 = g + h  
    add  t6, t3, t4       # t6 = i + j  
    sub  t0, t5, t6       # f = t5 - t6 (resultado em t0)
```

## Montagem e Execução

Clique no botão **Assemble** para montar o programa.

The screenshot shows the RISC-V Assembly Debugger interface. The assembly code in the editor window is:

```
riscv1.asm
1
2
3 # Exercicio 01 - Patterson pag. 54/55/56 (versão RISC-V)
4 # Expressão em C: f = (g + h) - (i + j)
5 #
6
7 .text
8
9 main:
10    addi t1, zero, 4      # g = 4
11    addi t2, zero, 3      # h = 3
12    addi t3, zero, 2      # i = 2
13    addi t4, zero, 1      # j = 1
14
15    add t5, t1, t2      # t5 = g + h
16    add t6, t3, t4      # t6 = i + j
17    sub t0, t5, t6      # f = t5 - t6 (resultado em t0)
```

The Registers window shows the following state:

Registers	Floating Point	Control and Status
zero		Number 0 Value 0
ra		1 0
sp		2 2147479846
gp		3 268463224
tp		4 0
t0		5 0
t1		6 0
t2		7 0
t3		8 0
t4		9 0
t5		10 0
t6		11 0
t7		12 0
t8		13 0
t9		14 0
t10		15 0
t11		16 0
t12		17 0
t13		18 0
t14		19 0
t15		20 0
t16		21 0
t17		22 0
t18		23 0
t19		24 0
t20		25 0
t21		26 0
t22		27 0
t23		28 0
t24		29 0
t25		30 0
t26		31 0
pc		4194304

The Messages window is empty.

A primeira instrução destacada será **addi t1, zero, 4** localizada no endereço **0x00400000**.

Text Segment					Source
Bkpt	Address	Code	Basic		
	0x00400000	0x00400013	addi x6,x0,4	10:	addi t1, zero, 4 # g = 4
	0x00400004	0x00300393	addi x7,x0,3	11:	addi t2, zero, 3 # h = 3
	0x00400008	0x00200e13	addi x28,x0,2	12:	addi t3, zero, 2 # i = 2
	0x0040000c	0x00100e93	addi x29,x0,1	13:	addi t4, zero, 1 # j = 1
	0x00400010	0x00730f33	add x30,x6,x7	15:	add t5, t1, t2 # t5 = g + h
	0x00400014	0x01de0fb3	add x31,x28,x29	16:	add t6, t3, t4 # t6 = i + j
	0x00400018	0x41ff02b3	sub x5,x30,x31	17:	sub t0, t5, t6 # f = t5 - t6 (resultado em t0)

O registrador **PC** apontará para o endereço da instrução. Observe:

Registers	Floating Point	Control and Status	
Name	Number	Value	
zero	0	0x00000000	
ra	1	0x00000000	
sp	2	0x7ffffefffc	
gp	3	0x10008000	
tp	4	0x00000000	
t0	5	0x00000000	
t1	6	0x00000000	
t2	7	0x00000000	
s0	8	0x00000000	
s1	9	0x00000000	
a0	10	0x00000000	
a1	11	0x00000000	
a2	12	0x00000000	
a3	13	0x00000000	
a4	14	0x00000000	
a5	15	0x00000000	
a6	16	0x00000000	
a7	17	0x00000000	
s2	18	0x00000000	
s3	19	0x00000000	
s4	20	0x00000000	
s5	21	0x00000000	
s6	22	0x00000000	
s7	23	0x00000000	
s8	24	0x00000000	
s9	25	0x00000000	
s10	26	0x00000000	
s11	27	0x00000000	
t3	28	0x00000000	
t4	29	0x00000000	
t5	30	0x00000000	
t6	31	0x00000000	
pc		0x00400000	

Ao executar a instrução com **Run One Step**, o valor de **t1** muda de **0x00000000** para **0x00000004**.

Antes:

Registers	Floating Point	Control and Status	
Name	Number	Value	
zero	0	0x00000000	
ra	1	0x00000000	
sp	2	0x7ffffefffc	
gp	3	0x10008000	
tp	4	0x00000000	
t0	5	0x00000000	
t1	6	0x00000000	
t2	7	0x00000000	
s0	8	0x00000000	
s1	9	0x00000000	
a0	10	0x00000000	
a1	11	0x00000000	
a2	12	0x00000000	
a3	13	0x00000000	
a4	14	0x00000000	
a5	15	0x00000000	
a6	16	0x00000000	
a7	17	0x00000000	
s2	18	0x00000000	
s3	19	0x00000000	
s4	20	0x00000000	
s5	21	0x00000000	
s6	22	0x00000000	
s7	23	0x00000000	
s8	24	0x00000000	
s9	25	0x00000000	
s10	26	0x00000000	
s11	27	0x00000000	
t3	28	0x00000000	
t4	29	0x00000000	
t5	30	0x00000000	
t6	31	0x00000000	
pc		0x00400000	

Depois:

Registers	Floating Point	Control and Status	
Name	Number	Value	
zero	0	0x00000000	
ra	1	0x00000000	
sp	2	0x7ffffefffc	
gp	3	0x10008000	
tp	4	0x00000000	
t0	5	0x00000000	
<b>t1</b>	<b>6</b>	<b>0x00000004</b>	
t2	7	0x00000000	
s0	8	0x00000000	
s1	9	0x00000000	
a0	10	0x00000000	
a1	11	0x00000000	
a2	12	0x00000000	
a3	13	0x00000000	
a4	14	0x00000000	
a5	15	0x00000000	
a6	16	0x00000000	
a7	17	0x00000000	
s2	18	0x00000000	
s3	19	0x00000000	
s4	20	0x00000000	
s5	21	0x00000000	
s6	22	0x00000000	
s7	23	0x00000000	
s8	24	0x00000000	
s9	25	0x00000000	
s10	26	0x00000000	
s11	27	0x00000000	
t3	28	0x00000000	
t4	29	0x00000000	
t5	30	0x00000000	
t6	31	0x00000000	
pc		0x00400004	

A execução dessa primeira instrução modificou apenas o conteúdo do registrador **t1**. As demais instruções irão modificar outros registradores. Execute o programa passo-a-passo e preencha a tabela a seguir, marcando o conteúdo do registrador PC, a instrução a ser executada e o valor dos registradores após a execução da instrução. Anote o valor de um registrador apenas quando ele é modificado.

## Preencha a tabela abaixo

Antes		Depois da execução da instrução						
PC	Instrução	R5	R6	R7	R28	R29	R30	R31
		(t0)	(t1)	(t2)	(t3)	(t4)	(t5)	(t6)
		0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400000	addi t1, zero, 4	0x00000000	0x00000004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400004	addi t2, zer			0x00000003				
0x00400008	addi t3, zer				0x00000002			
0x0040000C	addi t4, zer					0x00000001		
0x00400010	add t5, t1,						0x00000003	
0x00400014	add t6, t3,							0x00000007
0x00400018	sub t0, t6,	0x00000004						

**OBS:** Salve o PDF em formato A2 e Paisagem para garantir que todas as informações da página fiquem visíveis

Salvar como PDF

**Nota:** Se for necessário reiniciar o programa, clique no botão **Reset**.

5/19/25, 9:43 PM

## Exercício 1 - Linguagem Assembly RISC-V

C:\Users\eduan\OneDrive\Área de Trabalho\riscv.lasm - RARS 1.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Reset memory and registers

Segment Registers Floating Point Control and Status

Registers

Name	Number	Value
zero	0	0x00000000
t1	1	0xfffffffffc
sp	2	0x00000000
gp	3	0x00000000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000000
t2	7	0x00000000
m0	8	0x00000000
s1	9	0x00000000
a0	10	0x00000000
a1	11	0x00000000
a2	12	0x00000000
a3	13	0x00000000
a4	14	0x00000000
a5	15	0x00000000
a6	16	0x00000000
t7	17	0x00000000
s2	18	0x00000000
s3	19	0x00000000
s4	20	0x00000000
s5	21	0x00000000
s6	22	0x00000000
s7	23	0x00000000
s8	24	0x00000000
s9	25	0x00000000
s10	26	0x00000000
s11	27	0x00000000
t3	28	0x00000000
t4	29	0x00000000
t5	30	0x00000000
pc	31	0x00040004

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010020	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010040	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010060	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010100	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010120	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010140	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010160	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x10010180	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100101a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Messages Run I/O

Reset: reset completed.

Clear

Hexadecimal Addresses Hexadecimal Values ASCII

[← Voltar ao tutorial](#)