

Exercício 07

Nome do aluno:

Fabio Volkmann Coelho

Objetivo

Consolidar o aprendizado da linguagem Assembly RISC-V e compreender como as instruções de suporte a procedimentos são executadas.

Instruções

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

```
# -----
# Exercício 07 - Versão RISC-V
# Trecho em C: int f = (g + h) - (i + j);
# -----

.text
j main

leaf_example:
    add a5, a1, a2      # a5 = g + h
    add a6, a3, a4      # a6 = i + j
    sub a0, a5, a6      # f = (g + h) - (i + j), resultado em a0
    jr ra               # retorna para o chamador

main:
    addi a1, zero, 4    # g = 4
    addi a2, zero, 7    # h = 7
    addi a3, zero, 2    # i = 2
    addi a4, zero, 1    # j = 1
    jal leaf_example    # chama a função
    nop                 # a0 agora tem o valor de retorno (f)
```

Montagem e Execução

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

The screenshot shows the RARS 1.5 assembly debugger interface. The assembly code in the editor window is:

```

1  # 
2  # Exercicio 07a - Versão RISC-V
3  # Trecho em C: int f = (g + h) - (i + j);
4  #
5
6  .text
7  j main
8
9  leaf_example:
10 add a5, a0, a1      # a5 = g + h
11 add a6, a2, a3      # a6 = i + j
12 sub a0, a5, a6      # f = (g + h) - (i + j), resultado em a0
13 jr ra              # retorna para o chamador
14
15 main:
16 addi a0, zero, 4    # g = 4
17 addi a1, zero, 7    # h = 7
18 addi a2, zero, 2    # i = 2
19 addi a3, zero, 1    # j = 1
20 jal leaf_example   # chama a função
21 nop                # a0 agora tem o valor de retorno (f)

```

The Registers window shows the following initial values:

Registers	Floating Point	Control and Status
zero		0
ra		0
sp		2147479548
gp		268468224
tp		0
t0		0
t1		0
t2		0
s0		0
s1		0
a0		0
a1		0
a2		0
a3		0
a4		0
a5		0
a6		0
a7		0
s2		0
s3		0
s4		0
s5		0
s6		0
s7		0
s8		0
s9		0
s10		0
s11		0
t3		0
t4		0
t5		0
t6		0
pc		4194304

The Messages window shows the following log:

- Reset: reset completed.
- program is finished running (dropped off bottom) --
- Reset: reset completed.

Faça a execução passo-a-passo do programa e, a cada instrução, preencha a tabela abaixo cada vez que o valor de um registrador ou posição da memória de dados for modificado.

Antes da execução da instrução		Depois da execução da instrução							
PC	Instrução	f	g	h	i	j			
		R10	R11	R12	R13	R14	R15	R16	R1
		(a0)	(a1)	(a2)	(a3)	(a4)	(a5)	(a6)	(ra)
		0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400000	j main	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400014	addi a1, zero, 4			0x00000004	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Antes da execução da instrução		Depois da execução da instrução							
0x00400018	addi a2, zero, 7		0x00000004	0x00000007	0x00000002	0x00000001	0x00000000	0x00000000	0x00000000
0x0040001C	addi a3, zero, 2		0x00000004	0x00000007	0x00000002	0x00000001	0x00000000	0x00000000	0x00000000
0x00400020	addi a4, zero, 1		0x00000004	0x00000007	0x00000002	0x00000001	0x00000000	0x00000000	0x00000000
0x00400024	jal leaf_example								0x00400028
0x00400004	add a5, a1, a2						0x0000000B		0x00400028
0x00400008	add a6, a3, a4							0x00000003	0x00400028
0x0040000C	sub a0, a5, a6	0x00000008							0x00400028
0x00400010	jr ra								0x00400028
0x00400028	nop								

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

[Adicionar linha](#) [Salvar como PDF](#)

Se desejar reiniciar o programa, clique no botão **Reset**.

C:\Users\eduardo\OneDrive\Área de Trabalho\riscv1.asm - RARS 1.5

File Edit Run Settings Tools Help

Run speed at max (no interaction)

Reset memory and registers

Text Segment

Bkpt	Address	Code	Basic	Source
	0x00400000	0x00400313	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)

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

0x10010000 (.data) Hexadecimal Addresses Hexadecimal Values ASCII

Registers Floating Point Control and Status

Name	Number	Value
zero	0	0x00000000
ra	1	0x00000000
sp	2	0x7fffffe0
gp	3	0x10008000
tp	4	0x00000000
t0	5	0x00000000
t1	6	0x00000004
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

[← Voltar ao tutorial](#)