

Exercício 05

Nome do aluno:

Fabio Volkmann Coelho

Objetivo

Consolidar o aprendizado da linguagem Assembly RISC-V e compreender como as instruções de desvio condicional são executadas.

Instruções

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

```
# -----
# Exercício 05 - Baseado em Patterson pág. 67 (versão RISC-V)
# Trecho em C: while (save[i] == k) i = i + 1;
# -----

.data
Array_save: .word 3, 3, 1, 3, 3, 1, 4, 3, 1 # save[0] até save[8]

.text
main:
    la    s6, Array_save    # endereço base de save[]
    addi s5, zero, 3        # k = 3
    addi s3, zero, 0        # i = 0

Loop:
    add  t1, s3, s3          # t1 = s3 * 2
    add  t1, t1, t1          # t1 = s3 * 4
    add  t1, t1, s6          # t1 = endereço de save[i]
    lw   t0, 0(t1)          # t0 = save[i]
    bne  t0, s5, Exit        # se save[i] != k, vai para Exit
    addi s3, s3, 1          # i = i + 1
    j    Loop               # repete o laço

Exit:
    nop                    # fim do laço
```

Montagem e Execução

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

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.

file:///home/fabio/PycharmProjects/Mestrado/Arquitetura de Computadores/RARs/tutorial-rars-main/exercicio5.html

Antes da execução da instrução			Depois da execução da instrução					
PC	Instrução	Pseudoinstrução			i	j	k	save
			R5	R6	R19	R20	R21	R22
			(t0)	(t1)	(s3)	(s4)	(s5)	(s6)
			0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x00400008	addi s5, zero, 3	k = 3					0x00000003	0x10010000
0x0040000C	addi s3, zero, zero	i = 0			0x00000000		0x00000003	0x10010000
0x00400010	add t1, s3, s3	t1 = i + i		0x00000000	0x00000000		0x00000003	0x10010000
0x00400014	add t1, t1, t1	t1 = t1 + t1		0x00000000	0x00000000		0x00000003	0x10010000
0x00400018	add t1, t1, s6	calcula offset		0x10010000	0x00000000		0x00000003	0x10010000
0x0040001c	lw t0, 0(t1)		0x00000003	0x10010000	0x00000000		0x00000003	0x10010000
0x00400020	bne t0, s5, Exit		0x00000003	0x10010000	0x00000000		0x00000003	0x10010000
0x00400024	addi s3, s3, 1	i++	0x00000003	0x10010000	0x00000001		0x00000003	0x10010000
0x00400028	j Loop	volta para o loop	0x00000003	0x10010000	0x00000001		0x00000003	0x10010000
0x00400010				0x00000002				
0x00400014				0x00000004				
0x00400018				0x10010004				
0x0040001c			0x00000003					0x10010004
0x00400020								
0x00400024					0x00000002			
0x00400028								
0x00400010				0x00000004				
0x00400014				0x00000008				
0x00400018				0x10010008				
0x0040001c			0x00000001					
0x00400020	bne t0, s5, Exit		0x00000001					
0x0040002c	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**.

[← Voltar ao tutorial](#)