|  |
| --- |
| Exercício 05 |

**Objetivo:**

Consolidar o aprendizado da interface do RARS e aprender como executam as instruções de desvio do RISC-V executando o quinto exemplo de programação na linguagem de montagem do RISC-V no livro texto.

**Instruções:**

1. Inicie o RARS.
2. No editor de texto do RARS, transcreva o código abaixo e salve o arquivo com o nome **exercicio\_05**.

####################################################################

# Exercício 05 - Patterson pags. 67

# Mostra a compilação de um laço While

####################################################################

# Trecho em C:

# while (save[i] == k)

# i = i + j;

.data # segmento de dados

# definição do array save[]. Coloca os valores de save[0]=3 até save[8]=1 na memória

Array\_save: .word 3, 3, 3, 3, 3, 1, 4, 3, 1

.text # segmento de código (programa)

main:

la s6, Array\_save # Conforme o exercício, $s6 contém o endereço-base de save[].

addi s5, zero, 3 # inicializando k=3 por causa do array que foi definido (5 iterações)

addi s4, zero, 1 # inicializando j=1 para varrer o array de 1 em 1

addi s3, zero, 0 # inicializando i=0

Loop: add t1, s3, s3 # $t1 = 2.i

add t1, t1, t1 # $t1 = 4.i

add t1, t1, s6 # $t1 = end.base + 4.i (deslocamento) = end. de save[i]

lw t0, 0(t1) # $t0 = save[i]

bne t0, s5, Exit # se save[i] != k goto Exit

add s3, s3, s4 # i = i + j

j Loop # goto Loop

Exit: nop # não faz nada

1. Para iniciar a montagem do código vá ao menu **Run** e selecione a opção **Assemble** ou pressione **F3**.
2. 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. Observe que, devido ao laço de repetição, o mesmo bloco de código será percorrido várias vezes.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Antes da execução da instrução** | | Depois da execução da  instrução | | | | | |
| **PC** | **Instrução** |  |  | **i** | **j** | **k** | **save** |
| **R5** | **R6** | **R19** | **R20** | **R21** | **R22** |
| **(t0)** | **(t1)** | **(s3)** | **(s4)** | **(s5)** | **(s6)** |
|  |  | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 |
| **00400000** | **auipc 22, 0xfc10** | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 |
| **00400004** | **addi 22, 22, 0** | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 10010000 |
| **00400008** | **addi s5, zero, 3** | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 10010000 |
| **0040000c** | **addi s4, zero, 1** | 00000000 | 00000000 | 00000000 | 00000000 | 00000003 | 10010000 |
| **00400010** | **addi s3, zero, 0** | 00000000 | 00000000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000000 | 00000000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000000 | 00000000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000000 | 00000000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000000 | 10010000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000003 | 10010000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **00400028** | **add s3, s3, s4** | 00000003 | 10010000 | 00000000 | 00000001 | 00000003 | 10010000 |
| **0040002c** | **j Loop** | 00000003 | 10010000 | 00000001 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000003 | 10010000 | 00000001 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000003 | 00000002 | 00000001 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000003 | 00000004 | 00000001 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000003 | 10010004 | 00000001 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000003 | 10010004 | 00000001 | 00000001 | 00000003 | 10010000 |
| **00400028** | **add s3, s3, s4** | 00000003 | 10010004 | 00000001 | 00000001 | 00000003 | 10010000 |
| **0040002c** | **j Loop** | 00000003 | 10010004 | 00000002 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000003 | 10010004 | 00000002 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000003 | 00000004 | 00000002 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000003 | 00000008 | 00000002 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000003 | 10010008 | 00000002 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000003 | 10010008 | 00000002 | 00000001 | 00000003 | 10010000 |
| **00400028** | **add s3, s3, s4** | 00000003 | 10010008 | 00000002 | 00000001 | 00000003 | 10010000 |
| **0040002c** | **j Loop** | 00000003 | 10010008 | 00000003 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000003 | 10010008 | 00000003 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000003 | 00000006 | 00000003 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000003 | 0000000c | 00000003 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000003 | 0000000c | 00000003 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000003 | 0000000c | 00000003 | 00000001 | 00000003 | 10010000 |
| **00400028** | **add s3, s3, s4** | 00000003 | 0000000c | 00000003 | 00000001 | 00000003 | 10010000 |
| **0040002c** | **j Loop** | 00000003 | 0000000c | 00000004 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000003 | 0000000c | 00000004 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000003 | 00000008 | 00000004 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000003 | 00000010 | 00000004 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000003 | 10010010 | 00000004 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000003 | 10010010 | 00000004 | 00000001 | 00000003 | 10010000 |
| **00400028** | **add s3, s3, s4** | 00000003 | 10010010 | 00000004 | 00000001 | 00000003 | 10010000 |
| **0040002c** | **j Loop** | 00000003 | 10010010 | 00000005 | 00000001 | 00000003 | 10010000 |
| **00400014** | **Loop: add t1, s3, s3** | 00000003 | 10010010 | 00000005 | 00000001 | 00000003 | 10010000 |
| **00400018** | **add t1, t1, t1** | 00000003 | 0000000a | 00000005 | 00000001 | 00000003 | 10010000 |
| **0040001c** | **add t1, t1, s6** | 00000003 | 00000014 | 00000005 | 00000001 | 00000003 | 10010000 |
| **00400020** | **lw t0, 0(t1)** | 00000003 | 10010014 | 00000005 | 00000001 | 00000003 | 10010000 |
| **00400024** | **bne t0, s5, Exit** | 00000001 | 10010014 | 00000005 | 00000001 | 00000003 | 10010000 |
| **00400030** | **Exit: nop** | 00000001 | 10010014 | 00000005 | 00000001 | 00000003 | 10010000 |

OBS: acrescente as linhas adicionais que forem necessárias.