



Departamento de
Sistemas e
Computação

Universidade Federal de Campina Grande
Departamento de Sistemas e Computação
Disciplina: *Organização e Arquitetura de Computadores*
Prof.: *Joseana Macêdo Fehine Régis de Araújo*

Lista de Exercícios N° 09

Observações:

- Implemente os exercícios utilizando o simulador SPIKE (RV32I);
- Comprove os resultados obtidos a partir dos resultados fornecidos pelo simulador.

1. Execute o código abaixo passo a passo (Assembly de uma máquina RISC-V, 32 bits) e responda as questões a seguir.
 - a) Para cada instrução executada, complete a tabela informando o endereço de memória onde essa será armazenada (incluindo o valor do *target*), a operação realizada e o conteúdo dos registradores envolvidos após sua execução.
 - b) Substitua *sll s1, s1, s0* por *sll s1, s1, s1* e informe o resultado final dos registradores envolvidos.
 - c) Apresente o código de máquina da instrução *add s1, s1, s0*.

Endereço de memória	Instrução	Operação realizada	Conteúdo dos registradores envolvidos
<i>0x00000200</i>	<i>addi s0, zero, 2</i>		
	<i>addi s1, zero, 1</i>		
	<i>sll s1, s1, s0</i>		
	<i>bne s0, s1,</i> <i>target</i>		
	<i>addi s1, s1, 1</i>		
	<i>sub s1, s1, s0</i>		
<i>target:</i>	<i>add s1, s1, s0</i>		

2. Execute o código abaixo passo a passo (Assembly de uma máquina RISC-V, 32 bits), informe qual a operação realizada e qual o conteúdo dos registradores *s0* e *s1* ao final da execução.

```
addi s1, zero, 0x0  
addi s0, zero, 0x0  
addi t0, zero, 0x5  
for:  
  beq s0, t0, done  
  add s1, s1, s0  
  addi s0, s0, 0x1  
j for  
done:
```

3. Execute o código abaixo passo a passo (Assembly de uma máquina RISC-V, 32 bits) e responda as questões a seguir.

a) Forneça o código de máquina da instrução `li s1, 0`.

b) Forneça o código de máquina da instrução `beq s0, t0, pc + 16`.

c) Informe o valor do Registrador pc (Contador de Programa) ao ser executada a instrução `j pc - 0xc` e indique qual é a próxima instrução a ser executada após esta.

```
0x00000200  li      s1, 0
             li      s0, 0
             li      t0, 5
             beq     s0, t0, pc + 16
             add     s1, s1, s0
             addi    s0, s0, 1
             j      pc - 0xc
```