

# Apostila de Exercícios - Assembly RISC-V

## Rastreamento de Código e Execução de Ciclos

### Instruções Gerais

- Cada questão apresenta um conjunto de registradores já inicializados
  - Execute o código RISC-V pelo número de ciclos especificado
  - Um ciclo corresponde à execução de uma linha de código
  - Registre o valor de cada registrador após cada ciclo
  - Labels (como LOOP, END) não contam como ciclos
- 

### Questão 1

Registradores iniciais:  $x1 = 4$ ,  $x2 = 8$ ,  $x3 = 3$ ,  $x4 = 1$

Executar: 10 ciclos

```
assembly

slli x1, x1, 2
add x2, x2, x1
LOOP: blt x3, x2, END
... sll x3, x3, x4
... addi x2, x2, -2
... beq x0, x0, LOOP
END: and x1, x1, x3
... xor x4, x1, x2
```

---

### Questão 2

Registradores iniciais:  $x1 = 6$ ,  $x2 = 2$ ,  $x3 = 12$ ,  $x4 = 15$

Executar: 8 ciclos

```
assembly
```

```
andi x4, x4, 7
srl x3, x3, x2
LOOP: bgt x3, x1, END
    add x1, x1, x2
    ... slt x4, x1, x3
    ... bne x4, x0, LOOP
END: srl x3, x3, 1
    ... xor x2, x2, x3
```

### Questão 3

**Registradores iniciais:**  $x1 = 10$ ,  $x2 = 5$ ,  $x3 = 20$ ,  $x4 = 8$

**Executar:** 12 ciclos

```
assembly

addi x1, x1, -2
sll x2, x2, 1
xor x3, x3, x4
LOOP: beq x1, x2, END
    ... slt x4, x1, x2
    ... bne x4, x0, SKIP
    ... addi x1, x1, -1
SKIP: addi x2, x2, 1
    ... beq x0, x0, LOOP
END: and x3, x3, x1
    ... or x4, x4, x2
```

### Questão 4

**Registradores iniciais:**  $x1 = 7$ ,  $x2 = 14$ ,  $x3 = 3$ ,  $x4 = 0$

**Executar:** 9 ciclos

```
assembly
```

```
srli x2, x2, 1
slli x3, x3, 2
LOOP: blt x4, x3, CONT
    beq x0, x0, END
CONT: add x4, x4, x1
... srl x1, x1, x3
... bne x1, x0, LOOP
END: andi x2, x2, 15
... xor x3, x2, x4
```

## Questão 5

**Registradores iniciais:**  $x1 = 16$ ,  $x2 = 1$ ,  $x3 = 31$ ,  $x4 = 2$

**Executar:** 15 ciclos

```
assembly

srl x1, x1, x4
and x3, x3, x1
LOOP: bgt x1, x2, PROCESS
... beq x0, x0, END
PROCESS: slt x4, x2, x1
... bne x4, x0, CONTINUE
... addi x2, x2, 3
CONTINUE: srli x1, x1, 1
... or x3, x3, x2
... beq x0, x0, LOOP
END: xor x1, x1, x3
... andi x2, x2, 7
... add x4, x1, x2
```

## Questão 6

**Registradores iniciais:**  $x1 = 9$ ,  $x2 = 18$ ,  $x3 = 6$ ,  $x4 = 4$

**Executar:** 11 ciclos

```
assembly
```

```
xor x1, x1, x3
slli x4, x4, 1
LOOP: beq x1, x4, END
    blt x1, x4, MENOR
    .... addi x1, x1, -1
    .... beq x0, x0, LOOP
MENOR: add x1, x1, x3
    .... srl x2, x2, 1
    .... beq x0, x0, LOOP
END: and x4, x4, x2
    .... or x3, x3, x1
```

## Questão 7

**Registradores iniciais:**  $x1 = 5$ ,  $x2 = 10$ ,  $x3 = 8$ ,  $x4 = 12$

**Executar:** 13 ciclos

```
assembly

andi x4, x4, 3
add x1, x1, x4
sll x3, x3, x4
LOOP: bne x1, x3, CHECK
    .... beq x0, x0, END
CHECK: slt x4, x1, x3
    .... beq x4, x0, GREATER
    .... slli x1, x1, 1
    .... beq x0, x0, LOOP
GREATER: srli x3, x3, 1
    .... xor x2, x2, x1
    .... beq x0, x0, LOOP
END: or x4, x1, x3
```

## Questão 8

**Registradores iniciais:**  $x1 = 24$ ,  $x2 = 3$ ,  $x3 = 15$ ,  $x4 = 6$

**Executar:** 7 ciclos

```
assembly
```

```
srl x1, x1, x2
and x3, x3, x4
LOOP: bgt x3, x1, END
    addi x3, x3, 2
    xor x4, x4, x2
    blt x4, x3, LOOP
END: slli x2, x2, 2
    or x1, x1, x2
```

## Questão 9

Registadores iniciais:  $x1 = 13$ ,  $x2 = 26$ ,  $x3 = 4$ ,  $x4 = 1$

Executar: 14 ciclos

```
assembly

srli x2, x2, 1
xor x1, x1, x2
slt x4, x1, x3
LOOP: beq x4, x0, ELSE
    add x1, x1, x3
    sll x3, x3, x4
    beq x0, x0, CHECK
ELSE: addi x1, x1, -2
    and x2, x2, x3
CHECK: slt x4, x1, x2
    bne x4, x0, LOOP
END: andi x3, x3, 31
    or x4, x1, x3
    xor x2, x2, x4
```

## Questão 10

Registadores iniciais:  $x1 = 11$ ,  $x2 = 4$ ,  $x3 = 22$ ,  $x4 = 7$

Executar: 6 ciclos

```
assembly
```

```
slli x1, x1, 1
srl x3, x3, x2
LOOP: blt x1, x3, END
    xor x1, x1, x4
    ....
    andi x4, x4, 1
    ....
    beq x0, x0, LOOP
END: add x2, x1, x3
```

## Questão 11

**Registadores iniciais:**  $x1 = 8$ ,  $x2 = 16$ ,  $x3 = 2$ ,  $x4 = 32$

**Executar:** 16 ciclos

```
assembly

and x4, x4, x1
srl x2, x2, x3
addi x3, x3, 1
LOOP: beq x1, x2, EQUAL
    ....
    bgt x1, x2, GREATER
    ....
    slli x1, x1, 1
    ....
    xor x3, x3, x4
    ....
    beq x0, x0, LOOP
GREATER: srli x2, x2, 1
    ....
    or x4, x4, x3
    ....
    beq x0, x0, LOOP
EQUAL: and x1, x1, x4
    ....
    add x2, x2, x3
    ....
    slt x4, x1, x2
    ....
    bne x4, x0, END
    ....
    xor x1, x1, x2
END: andi x3, x3, 15
```

## Questão 12

**Registadores iniciais:**  $x1 = 30$ ,  $x2 = 6$ ,  $x3 = 12$ ,  $x4 = 9$

**Executar:** 10 ciclos

```
assembly
```

```
srli x1, x1, 2
andi x4, x4, 7
LOOP: bne x2, x4, DIFF
    beq x0, x0, END
DIFF: slt x3, x2, x4
    bne x3, x0, INCREASE
    addi x4, x4, -1
    beq x0, x0, LOOP
INCREASE: add x2, x2, x1
    sll x1, x1, 1
    beq x0, x0, LOOP
END: xor x3, x1, x2
```

## Questão 13

Registadores iniciais:  $x1 = 17$ ,  $x2 = 34$ ,  $x3 = 5$ ,  $x4 = 10$

Executar: 18 ciclos

```
assembly

srli x2, x2, x3
xor x1, x1, x4
slli x3, x3, 1
OUTER: bgt x1, x2, END
    and x4, x4, x3
INNER: blt x4, x1, NEXT
    srli x4, x4, 1
    addi x3, x3, 1
    bne x4, x0, INNER
NEXT: add x1, x1, x3
    or x2, x2, x4
    slt x4, x1, x2
    bne x4, x0, OUTER
END: andi x1, x1, 63
    xor x2, x2, x3
    sll x4, x4, x3
    add x3, x1, x2
    and x4, x4, x3
```

## Questão 14

Registadores iniciais:  $x1 = 28$ ,  $x2 = 7$ ,  $x3 = 14$ ,  $x4 = 3$

Executar: 9 ciclos

assembly

```
srl x1, x1, x4
slli x2, x2, 2
and x3, x3, x2
LOOP: beq x1, x3, END
.... blt x1, x3, SMALLER
.... addi x1, x1, -4
.... beq x0, x0, LOOP
SMALLER: xor x1, x1, x2
.... or x3, x3, x4
.... beq x0, x0, LOOP
END: srli x2, x2, 1
```

## Questão 15

**Registadores iniciais:**  $x1 = 19$ ,  $x2 = 38$ ,  $x3 = 1$ ,  $x4 = 25$

**Executar:** 20 ciclos

assembly

```
srli x2, x2, 1
andi x4, x4, 15
sll x3, x3, x4
MAIN: bne x1, x2, PROCESS
.... beq x0, x0, FINAL
PROCESS: slt x4, x1, x2
.... beq x4, x0, CASE1
.... add x1, x1, x3
.... xor x2, x2, x4
.... beq x0, x0, MAIN
CASE1: addi x2, x2, -3
.... and x4, x4, x1
.... srl x3, x3, 1
.... bne x3, x0, MAIN
FINAL: or x1, x1, x2
.... slli x3, x3, 2
.... xor x4, x1, x3
.... add x2, x2, x4
.... and x1, x1, x2
.... srli x4, x4, 2
.... andi x3, x3, 31
```



## Observações Importantes

1. **Registrador x0:** Sempre contém o valor 0 (zero) e não pode ser modificado
2. **Labels:** Não contam como ciclos de execução
3. **Branches:** Quando um branch é tomado, o próximo ciclo executa a instrução no destino
4. **Immediatos:** Valores constantes nas instruções (ex: addi x1, x1, -2)
5. **Operações lógicas:** AND, OR, XOR operam bit a bit
6. **Shifts:** SLL/SLLI (shift left), SRL/SRLI (shift right logical)

## Dicas de Rastreamento

- Mantenha uma tabela com os valores dos registradores após cada ciclo
- Preste atenção aos branches condicionais e incondicionais
- Lembre-se que beq x0, x0, LABEL sempre salta (branch incondicional)
- Verifique se o número de ciclos solicitado foi atingido, mesmo se o programa chegou ao fim