

Os exercícios foram adaptados do livro

Patterson, David A. Hennessy, John L. Organização e Projeto de Computadores. Disponível em: Minha Biblioteca, (5a. edição). Grupo GEN, 2017.

1. <§2.2> Em cada código, considere os valores iniciais armazenados nos registradores como abaixo:

Reg	Valor
t0	0x0
t1	0x1
t2	0x10
t3	0x20
t4	0xFFFFFFFF

Escreva a mesma tabela para os valores alterados pelos códigos abaixo:

a.

```
Unset
and $t0, $t0, $t1
or $t1, $t1, $t2
add $t2, $t2, $t3
addi $t3, $t3, $zero
and $t4, $t4, $t0
```

b.

```
Unset
addi $t0, $t0, 0xf
andi $t1, $t1, 0xf
ori $t2, $t2, 0xf
sll $t3, $t3, 4
srl $t4, $t4, 16
```

c.

```
Unset
sll $t0, $t0, 2
sll $t1, $t1, 4
sll $t2, $t2, 5
sll $t3, $t3, 6
sll $t4, $t4, 30
```

d.

Unset

```
and $t0, $t0, $t4  
or $t1, $t1, $t4  
xor $t2, $t2, $t4  
sub $t3, $t3, $t4  
xori $t4, $t4, 0xFFF0
```

2. <§2.2> Converta as operações de alto nível abaixo para assembly do MIPS utilizando somente operações de deslocamento. Suponha que as variáveis a, b, c, d e e estejam em registradores, respectivamente em \$s0, \$s1, \$s2, \$s3 e \$s4 e que sejam inteiros de 32 bits.

Considere também que / é o operador para divisão inteira e que % é o operador de resto de divisão. Use um número mínimo de instruções assembly do MIPS.

a.

```
a = b / 2  
c = d * 2
```

b.

```
a = b / 4  
d = c / 16
```

c.

```
a = b / 4  
c = a / 32  
d = c * 4
```

d.

```
a = b % 2  
c = d % 16
```

e.

```
a = b % 32  
c = a * 64  
d = c % 256
```

f.

```
a = b * 1024  
c = d / 4096  
d = c % 16
```