Universidade Federal de São Carlos
Centro de Ciências Exatas e de Tecnologia
Departamento de Computação
Arquitetura e Organização de Computadores 1
Prof. Luciano Neris

Respostas - Exercícios 02

1. $7 = 1.

2. Endereço L1.

3.

   a) Considerando-se a variável a em t0,
   ```
   addi    t0, t0, 1
   ```

   b) Considerando-se a variável a em t0,
   ```
   addi    t0, zero, 0
   ```

   c)
   ```
   la s4, v
   li s5, 5
   sw s5, 40(s4)
   ; 40($4) = v[10], pois cada elemento do vetor é uma palavra de 32 bits, ou 4 bytes.
   ```

   d) Considerando a variável a em s2, e a variável b em s3:
   ```
   slt     s7, s2, s3  ; ou seja, s2 <s3 então s7 = 1
   bne     s7,  s0, L1 ; portanto desvia se s7 diferente de  0, como s7 =1.
   ```

   e) Considerando a variável a em s3,
   ```
   slt     s7 ,zero, s3  ; ou seja se 0 < s3 então s7 =1
   bne     s7,  zero,  L1 ; portanto desvia se s7 diferente de 0, como s7 =1.
   ```

4. 0x016a8a33 -> 00000001011010101000101000110011
   0000000 10110 10101 000 10100 0110011

funct7	rs2	rs1	funct3	rd	opcode
7 bits	5 bits	5 bits	3 bits	5 bits	7 bits
0	22	21	0	20	51
0000000	10110	10101	000	10100	0110011

5. SOLUÇÃO:
   0x0059aa03 -> 00000000010110011010101000000011
   000000000101 10011 010 10100 0000011

immediate	rs1	funct3	rd	opcode
5	19	2	20	3
12 bits	5 bits	3 bits	5 bits	7 bits
000000000101	10011	010	10100	0000011

6. SOLUÇÃO:
   0x01499463 -> 00000001010010011001010001100011
   0000000 10100 10011 001 01000 1100011

imm	rs2	rs1	funct3	Imm	opcode
0	20	19	1	8	99
0000000	10100	10011	001	01000	1100011
imm [10:5]	5	5	3	imm [4:1]	7

imm[12] imm[11]

7. SOLUÇÃO:

```
        la s4, a
        la s5, b
        li s6, 5                    # vector size

        sub   s2,s2,s2              # zera $2 -> i
Loop:
        lw    t0, 0(s4)             # $t0 = A[i]
        lw    t1, 0(s5)             # $t1 = B[i]
        add   t0, t0, t1            # $t0 = A[i] + B[i]
        sw    t0, 0(s4)             # A[i] = A[i]+B[i]

        addi  s4, s4, 4             # i = I + 1
        addi  s5, s5, 4             # i = I + 1

        addi  s2, s2, 1             # i = I +1
        slt   s7, s2, s6            # se i < n $7 = 1
        bne   s7, zero, Loop        # desvia se $7 <> 0
```

8. SOLUÇÃO:
   -126 em binário = 10000010
   -64 em binário = 11000000
   (-126) + (-64) em binário = 01000010

   Ocorre overflow pois a soma de dois números negativos (-126 e -64) deu positivo. O resultado em decimal é:
   $01000010 = 1x2^6 + 1x2^1 = 1x64 + 1x2 = 66$.

9.
   17 = 00010001

18 = 00010010
17 +18 = 00100011

10. 11111111 11000001

11.

```
lw t0, 32(s3)          # t0 recebe o valor de A[8]
add t0, s2, t0         # t0 recebe h + A[8]
sw t0, 48(s3)          # h + A[8] é armazenado em # A[12]
```

12.

```
# multiplicar o valor de 'i', que está em s4 por 4
add t1, s4, s4         # t1 recebe 2 * i
add t1, t1, t1         # t1 recebe 4 * i

add t1, t1, s3         # soma o end. base em s3 com o deslocamento de t1

lw t0, 0(t1)           # carrega A[i] em $t0
add s1, s2, t0         # g = h + A[i]
```

13.

```
swap:
        lw t0,(a0)     # t0=*xp
        lw t1,(a1)     # t1=*yp
        sw t1,(a0)     # *xp=t1
        sw t0,(a1)     # *yp=t0
        jr ra
```