Representação Binária de Instruções MIPS

1. Que instruções assembly MIPS representam as seguintes sequências de 32 bits?
    a. 00000010010100111000100000100000
    b. 10001110010100010000000001100100
    c. 00010010001100100000000000011001
    d. 00000011111000000000000000001000
    e. 00001100000000000000100111000100

2. Traduza os fragmentos de código assembly MIPS que se seguem para sequências de código máquina MIPS de 32 bits:

    a.
    ```
    8616        lw  $t0, 32($s3)
    8620        add $t0, $s2, $t0
    8624        sw  $t0, 48($s3)
    ```

    b.
    ```
    8616        bne $s3, $s4, else
    8620        add $s0, $s1, $s2
    8624        j   exit
    8628  else: sub $s0, $s1, $s2
    8632  exit: ...
    ```

    c.
    ```
    3672  proc: addiu $sp, $sp, -4
    3676        sw    $s0, 0($sp)
    3680        add   $t0, $a0, $a1
    3684        add   $t1, $a2, $a3
    3688        sub   $s0, $t0, $t1
    3692        add   $v0, $s0, $zero
    3696        lw    $s0, 0($sp)
    3700        addiu $sp, $sp, 4
    3704        jr $ra
    ```

3. Considere que se pretendia quadruplicar o conjunto de registos e o conjunto de instruções do MIPS. Como é que isso afetaria o tamanho dos campos das instruções do tipo R-format e do tipo I-format? E como é que isso contribuiria para aumentar/diminuir o tamanho do código assembly de um programa?

4. Considere que o valor 0xABCDEF12 foi guardado em memória a partir do endereço 0x1000. Indique qual seria o conteúdo das posições de memória 0x1000, 0x1001, 0x1002 e 0x1003 no caso duma arquitetura little-endian e duma arquitetura big-endian?

5. Considere que o program counter (PC) tem o valor 0x2F000000. É possível com a instrução **jump (j)** do assembly do MIPS colocar o PC no endereço 0x40000000? E com a instrução **branch-on-equal (beq)**?

6. (saber mais) Considere que o program counter (PC) tem o valor 0x00000600. É possível com a instrução **branch-on-equal (beq)** do assembly do MIPS colocar o PC no endereço 0x20014924? E se o PC tiver o valor 0x1FFFF000?