

Nome: Josiane de Sousa Alves 15/0038895

Data: 29/03/2017

Para as questões 2 a 5, considere que as variáveis 'f', 'g', 'h', 'i' e 'j' são do tipo inteiro (16 bits na arquitetura do MSP430), e que o vetor 'A[]' é do tipo inteiro.

Estas variáveis estão armazenadas nos seguintes registradores:

f: R4

g: R5

h: R6

i: R7

j: R8

A: R9

Utilize os registradores R11, R12, R13, R14 e R15 para armazenar valores temporários.

1. Escreva os trechos de código assembly do MSP430 para:

- a) Somente setar o bit menos significativo de R5.**

bis.w #1, R5

- b) Somente setar dois bits de R6: o menos significativo e o segundo menos significativo.**

bis.w #3, R6

- c) Somente zerar o terceiro bit menos significativo de R7.**

bic.w #4, R7

- d) Somente zerar o terceiro e o quarto bits menos significativo de R8.**

bic.w #Ch, R8

- e) Somente inverter o bit mais significativo de R9.**

xor.w #8000h, R9

- f) Inverter o nibble mais significativo de R10, e setar o nibble menos significativo de R10.**

xor.w #F000h, R10

bis.w #Fh, R10

2. "Traduza" o seguinte trecho de código em C para o assembly do MSP430:

```
    if(i>j) f = g+h+10;
    else f = g-h-10;

    cmp R7, R8
    jge ELSE
    add.w #10, R6
    add.w R6, R5
    mov.w R5, R4
    jmp EXIT
ELSE:
    sub.w #10, R6
    sub.w R6, R5
    mov.w R5, R4
EXIT:
    ...
```

3. "Traduza" o seguinte trecho de código em C para o assembly do MSP430:

```
    while(save[i]!=k) i++;

LOOP: mov.w R7, R12
      rla R12
      add.w R10, R12
      cmp 0(R12), R9
      jeq EXIT
      inc.w R7
      jmp LOOP
EXIT:
    ...
```

4. "Traduza" o seguinte trecho de código em C para o assembly do MSP430:

```
    for(i=0; i<100; i++) A[i] = i*2;

    mov.w #0, R7
LOOP: cmp #100, R7
      jge EXIT
      mov.w R7, R12
      rla R12
      mov.w R12, R13
      add.w R9, R13
      mov.w R12, 0(R13)
      inc.w R7
      jmp LOOP
EXIT:
    ...
```

5. "Traduza" o seguinte trecho de código em C para o assembly do MSP430:

for(i=99; i>=0; i--) A[i] = i*2;

```
mov.w #99,R7
LOOP:
  cmp #0,R7
  jl EXIT
  mov.w R7,R12
  rla R12
  mov.w R12,R13
  add.w R9,R13
  mov.w R12,0(R13)
  dec.w R7
  jmp LOOP
EXIT:
  ...
```