

Disciplina: Eletrônica Embarcada **Código:** 120871 **Turma:** A

Professor: Diogo Caetano Garcia

Aluno/Matrícula: Fábio Barbosa Pinto – 11/0116356

Questionário: 05_Cont_Assembly

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 #12,R8`

e) Somente inverter o bit mais significativo de R9.

`mov.w #128,R4`

`xor.w R4,R9`

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

`mov.w #240,R4`

`xor.w R4,R10`

`mov.w #15,R4`

`bis.w R4,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 R8,R7 ; R7 = i R8 = j  
jeq ELSE  
jl ELSE  
add.w #10,R6 ; h = h + 10  
add.w R6,R5 ; g = g + h  
mov.w R5,R4 ; f = g
```

ELSE:

```
sub.w #10 ,R6  
sub.w R6,R5  
mov.w R5,R4
```

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

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

```
WHILE:  
mov.w R7,R12 ; R7 = i, R12 = temporary register  
rla R12 ; R12 = 2*i  
add.w R10,R12 ; R12 = save, R12  
cmp 0(R12),R9 ; compara save [i] com  
jeq EXIT ; save [i] == k ?  
inc.w R7 ; i++  
jmp WHILE  
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  
FOR:  
  
cmp #100, R7 ; compare i with 100  
jge EXIT ; if i ==> 100, go to exit  
mov.w R7,R12 ; R7 = i, R12 = temporary register  
rla R12 ; R12 = 2*i  
mov.w R12,R13  
add.w R9,R13 ; R13 = ADDRESS (A) + R12  
mov.w R12,0(R13); A[i] = 2*i  
inc.w R7 ; i++  
jmp FOR  
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
FOR:

cmp #0, R7 ; compare i with 0
jl EXIT ; if i < 0, go to exit
mov.w R7,R12 ; R7 = i, R12 = temporary register
rla R12 ; R12 = 2*i
mov.w R12,R13
add.w R9,R13 ; R13 = ADDRESS (A) + R12
mov.w R12,0(R13); A[i] = 2*i
dec.w R7 ; i--
jmp FOR

EXIT:
```