



Aluna: Tatielen Rodrigues Dutra Pereira

Matricula: 12/0136074

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.

xor.w #1,R5

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

xor.w #3,R6

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

xor.w #4,R7

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

and.w #243,R8 -> decimal

(e) Somente inverter o bit mais significativo de R9.

xor.w #128,R9

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

xor #240,R10

xor.w #15,R10

Nibble -> 4 bits



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;
```

```
comp R8,R7 ; Compara R7(i) com R8(j)
jl ELSE ; Se R8<R7 pula pra ELSE
mov.w R5,R4; Como não é R8>R7, movemos o valor de R5(g) para o registrador R4(f)
sub.w R6,R4; Subtraímos R4= R6-R4 -> f=g-h
sub.w #10,R4; Subtraímos R4= R4-10 -> f=g-h-10
jmp EXIT
ELSE:
mov.w #10,R4 ; Como R7>R8, movemos o valor 10 para o registrador R4(f)
add.w R5, R4 ; Somamos R4= R5+R4 -> f=g+10
add.w R6,R4 ; Somamos R4= R6+R4 -> f=g+h+10
jmp EXIT ; Pulamos para a saída
EXIT:
```

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

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

```
Se K=R11,
LOOP_WHILE:
mov.w R7,R12
rla 12
add.w R10,R12
cmp 0(R12),11
jeq EXIT
inc.w R7
jmp LOOP_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; Foi atribuído valor ao registrador R7=99
```

```
LOOP_FOR:
```

```
cmp R7, #100; Comparou R7 com 100
```

```
jge EXIT
```

```
mov.w R7,R12; R12 recebeu R7
```

```
rla R12; Multiplicou R12*2
```

```
mov.w 0(R12),R9; A[i] = i*2
```

```
inc.w R7; R7=R7+1
```

```
jmp LOOP_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; Foi atribuído valor ao registrador R7=99
```

```
LOOP_FOR:
```

```
cmp R7, #0 ; Comparou R7 com 0
```

```
jge EXIT;
```

```
mov.w R7,R12; R12 recebeu R7
```

```
rla R12; Multiplicou R12*2
```

```
mov.w 0(R12),R9; A[i] = i*2
```

```
dec.w R7; R7=R7+1
```

```
jmp LOOP_FOR
```

```
EXIT:
```