

**MC404: Organização de Computadores e Linguagem de Montagem**  
2<sup>a</sup> Prova (16/12/2014)

**Nome:**

**RA:**

Questão	Valor	Nota
1	1,0	
2	1,0	
3	1,5	
4	4,0	
5	2,5	
Total	10,0	

**Instruções:** A duração da prova é de uma hora e quarenta minutos. Qualquer tentativa de fraude será punida com zero para todos os envolvidos.

**Questão 1.** No código abaixo, substitua os três últimos dígitos do valor atribuído a r2 pelos três últimos dígitos do seu RA e responda às seguintes questões.

```
ldr r0, =0xF8000000
ldr r1, =0x08000000
ldr r2, =0x00000___ @ <- Substitua os 3 últimos dígitos deste
                      @     valor pelos 3 últimos do seu RA
adds r6, r0, r1
adc r7, r2, r6
```

- Qual o valor da flag C após a execução da instrução `adds r6, r0, r1`?
- Qual o valor armazenado no registrador `r7` após a execução da última instrução?

**Questão 2.** Escreva um trecho de código em linguagem de montagem do ARM que aloque na pilha do programa um registro do tipo descrito a seguir e inicie o campo `b` com o valor 3.

```
struct x{
    short a;
    short b;
    int   c;
}
```

**Questão 3.** Traduza o procedimento em C abaixo para linguagem de montagem do ARM. O código deve estar de acordo com a ABI vista no curso e deve usar apenas os registradores *caller-save*.

```
int foo(int v1, int v2)
{
    return bar(10, 11, 12, 13, v1, v2);
}
```

**Questão 4.** Traduza o programa em C abaixo para linguagem de montagem do ARM. O código deve estar de acordo com a ABI vista no curso.

```
short key;

short do_something(short* array, int v1, int v2, int v3, int v4, int v5) {

    unsigned int i=0;
    short tmp;

    do {
        tmp = array[i];
        i = i+1;
    } while (tmp != key);

    check(&tmp, v2);
    return tmp+v5;
}
```

**Questão 5.** Responda às perguntas abaixo:

- a) Quais os registradores salvos automaticamente pelo *hardware* quando ocorre uma interrupção?
- b) O que acontece se o fluxo de execução atingir uma posição de memória que não contém instruções válidas?
- c) Para que serve o *tZIC*?
- d) Suponha que a pilha do programa seja uma pilha crescente e vazia. Mostre como o usuário pode empilhar os valores dos registradores r1, r4 e LR com uma única instrução. Mostre também como o usuário pode desempilhar os 3 valores da pilha e armazenálos em r1, r4 e LR com uma única instrução.
- e) Por que existem dois modos de operação distintos no ARM que provêem acesso ao mesmo conjunto de registradores? Quais são estes modos?