



INTRODUÇÃO À ARQUITETURA DE COMPUTADORES

2017/2018

1º Semestre

1º Teste (*exemplo*)

Teste tipo

Duração: 1h30m

-
- Apenas pode consultar: (a) uma página A4 escrita à mão por si, não sendo permitidas fotocópias, impressões, nem páginas escritas por outrem, e (b) o anexo que lhe deverá ter sido entregue com o teste. Por favor, não escreva nesse anexo e devolva-o no final do teste.
 - Resolva o teste no próprio enunciado, o espaço reservado para cada pergunta é suficiente para a sua resposta. Tenha em atenção que cada grupo deve ficar em folhas separadas. Utilize as costas da última folha para rascunho ou para espaço adicional de resposta, desde que claramente identificado como tal.
 - Identifique todas as folhas que entregar, **folhas não identificadas não serão corrigidas!**
-

1. Efetue a seguinte operação aritmética em binário sem sinal, usando 4 bits para os dois operandos e 5 bits para o resultado. Confirme o resultado através da conversão dos operandos e do resultado para decimal. Mostre os cálculos que efetuar.

$$1101_2 + 0100_2$$

2. Considere os seguintes valores para os registos do processador P3:

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE (flags)
A1A1h	B2B2	C3C3h	D4D4h	E5E5h	000Fh	0002h	8100h	F100h	0002h

Para as perguntas a) – d), indique quais são os novos valores, em hexadecimal, de todos (**e apenas**) os registos que são escritos na execução de cada instrução. Use ? para indicar que não tem informação suficiente para determinar o novo valor de um registo. As perguntas são independentes, isto é, assuma como **valores iniciais** para cada pergunta os indicados na tabela acima.

a) DIV R6, R7

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

b) RET

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

c) POP R1

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

d) Etiqueta1: BR.NZ Etiqueta1

R1	R2	R3	R4	R5	R6	R7	PC	SP	RE

3. Escreva uma rotina **ParOuImpar** que recebe um número inteiro que é passado como parâmetro da função e retorna 0 se esse número for par e 1 se esse número for ímpar.

Escreva também um exemplo do código de chamada à rotina.

[Notas: Os parâmetros e o valor de retorno são passados pela pilha. Para simplificar, a sua rotina pode escrever por cima de qualquer registo, sem ter de se preocupar em repor o conteúdo do registo no final da rotina.]

4. Escreva o código em Assembly que conta o número de elementos com valor 0 que estão num vetor de inteiros de 16 bits, dispostos em sequência numa região de memória contígua, cujo endereço de início está contido no registo R1, e dimensão é dada pelo valor de R2. O resultado deve ser colocado no registo R3.

5. Considere a codificação em linguagem máquina das instruções do P3.

- a) Qual a codificação em linguagem máquina da instrução **BR.NZ salto**, supondo que esta pertence à seguinte sequência de instruções. (Explique os cálculos efetuados na sua resposta.)

```
                BR.NZ  salto
                MOV    R1, M[SP+3]
salto:          XOR    R2, R2
                ...
```

- b) Use a resposta à alínea anterior para explicar uma das limitações de usar a instrução **BR.NZ** em vez do **JMP.NZ**. Para tal, escreva uma versão modificada do programa acima que faça sobressair esta limitação. A versão modificada deverá conter o número mínimo de linhas de código possível, e indicar exatamente quantas linhas são necessárias.

6. Escolha a opção correspondente à resposta correta.

(Cada pergunta certa: +1 val.; cada pergunta errada: -0,25 val.; valor mínimo do grupo: 0)

a) Qual o valor binário fracionário (sem sinal) correspondente a 5,875

☐ 101,101 ☐ 101,11 ☐ 101,111 ☐ 11,11 ☐ Nenhuma das anteriores.

b) Qual o número decimal correspondente ao número em representação em complemento para dois usando 8 bits 11100010

☐ -27 ☐ -35 ☐ -30 ☐ -2 ☐ Nenhuma das anteriores.

c) Qual o código Python que equivale ao seguinte programa em Assembly do P3, supondo que R1 contém a variável x:

```
Here:  ADD    R1, R0
        BR.Z  End
        INC   R1
        BR    Here
End:    ADD    R1, 4
```

☐ `if x!=0:`
 `x += 1`
 `x += 4` ☐ `while x!=0:`
 `x += 1`
 `x += 4` ☐ `while x==0:`
 `x += 1`
 `x += 4` ☐ `while x!=0:`
 `x += 1`
 `x += 4` ☐ Nenhum dos anteriores.

d) Considere os valores -2_{10} e -3_{10} . Calcule a sua representação binária em complemento para 2 usando 4 bits, e some os dois números, obtendo o resultado em complemento para 2 usando 4 bits. O resultado desta operação é:

☐ 0110 ☐ 1010 ☐ 1011
☐ Não é representável com 4 bits devido a *overflow*. ☐ Nenhuma das anteriores