Arquitectura de Computadores

ENGENHARIA INFORMÁTICA
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Prova Modelo
1º Parte

Nome:_____ **Número:**_____

Notas Importantes:

A fraude denota uma grave falta de ética e constitui um comportamento não admissível num estudante do ensino superior. Não serão admitidas quaisquer tentativas de fraude, levando qualquer tentativa detectada à reprovação imediata, tanto do facilitador como do prevaricador.

Durante a prova pode consultar a bibliografia da disciplina (slides, livros, enunciados e materiais de apoio aos trabalhos práticos). No entanto, não é permitido o uso de computadores/máquinas de calcular e a consulta de exercícios previamente resolvidos.

Este é um teste de escolha múltipla e deverá assinalar sem ambiguidades as respostas na tabela apresentada a baixo. Cada pergunta corretamente respondida vale cinco pontos; cada pergunta errada desconta dois pontos; cada pergunta não respondida vale zero pontos. Um total abaixo de zero, conta como zero valores.

Respostas: (indicar resposta A, B, C ou D, debaixo do número da questão)

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

1. **Indique qual dos seguintes excertos de código em *Assembly* do MIPS implementa a operação aritmética `f=a+b-c+d` (assuma que `a`→`$s0`, `b`→`$s1`, `c`→`$s2`, `d`→`$s3` e `f`→`$s4`).**

a)
```
add   $s3,$s0,$s1
sub   $t0,$s3,$s2
add   $s4,$t0,$s3
```

b)
```
add   $t0,$s0,$s1
add   $t1,$s2,$s3
sub   $s4,$t0,$s1
```

c)
```
add   $t0,$s0,$s1
sub   $t0,$s2,$t0
add   $s4,$t0,$s3
```

d)
```
add   $t0,$s0,$s1
sub   $t0,$t0,$s2
add   $s4,$t0,$s3
```

2. **Relativamente ao fragmento de programa indicado ao lado, indique qual das afirmações é VERDADEIRA.**

```
int Num = 10;

void func(){

    int temp, *pon;

    pon=(int*)malloc(Num*sizeof(int));
    …
}
```

   a. A variável ***Num*** vai ser armazenada na zona de dados estáticos, a variável ***temp*** na pilha e a variável ***pon*** no *heap*.
   b. A variável ***Num*** como é uma constante não é armazenada em lado nenhum e as variáveis ***temp*** e ***pon*** vão ser armazenadas na pilha.
   c. A variável ***Num*** vai ser armazenada na zona de dados estáticos do programa, a variável ***temp*** na pilha e a variável ***pon*** contém um endereço de uma zona de memória localizada no *heap*.
   d. As variáveis ***temp*** e ***pon*** vão ser armazenadas na pilha e a variável ***Num*** no *heap*.

3. **Considere o excerto de código em Assembly do MIPS apresentado na caixa seguinte. Indique qual das opções representa o valor correcto do registo $t3 e $t4, após a execução deste excerto de código em Assembly do MIPS:**

```
        .data
tab:    .word 200,20,220,40,200,140,120,20,100,10
        .text
        la      $t0,tab
        lw      $t1,4($t0)
        lw      $t2,8($t0)
        slt     $t3,$t1,$t2
        li      $t5,1
        beq $t3,$t5, ciclo1
        beqz $t3, ciclo2
ciclo1:
        lw $t4, 4($t0)
        j fim_ciclo
ciclo2:
        lw $t4, 8($t0)

        fim_ciclo:
```

   a. $t3=1, $t4=200
   b. $t3=0, $t4=220
   c. $t3=1, $t4=20
   d. Nenhuma das anteriores.

4. **Considere o seguinte excerto de código em Assembly do MIPS, cuja função é alterar os caracteres nas posições pares da string carregada em memória por um caracter com código ASCII dado por ASCII_novo=ASCII_antigo+1. Escolha a opção que representa a instrução em falta:**

```
        .data
str: .asciiz "Frequência de AC"
        .text
main:
        la      $a0,str
loop:
        lb      $t0,0($a0)
        beq     $t0,$0,finish
        addi    $t1,$t0,1
        sb      $t1,0($a0)
        ##instrução em falta”
        j       loop
finish:
```

   a. addi $a0, $a0, 2
   b. addi $a0, $a0, 8
   c. addi $t0, $t0, 2
   d. Nenhuma das anteriores.

5. **Considere um processador que possui um valor de CPI igual a 2.4 quando todos os acessos à memória envolvem apenas a cache. Assuma a existência de duas caches, uma para instruções e outra para dados. Considere também que 35% das instruções envolvem um acesso à memória de dados. Se a miss rate é de 5% na cache de instruções e de 20% na cache de dados e se o miss penalty é de 10 ciclos de relógio em ambas as caches qual será o CPI real deste sistema?**

   a. 3.2 **b.** 3.6 **c.** 2.4 **d.** 2.8

6. **Considere o seguinte programa em C em que a tabela "*tab*" começa no endereço** 0x62FE20**. Com base nisso, indique qual das seguintes opções é <u>VERDADEIRA</u>:**

```
#include <stdio.h>
int main(){
int tab[] ={0,2,4,6,8,10,12};
int *p1, **p2;
p1 = tab+2;
p2 = &p1;
printf("%#X \n", &tab );       // Instrução I1
printf("%#X \n", tab );        // Instrução I2
printf("%#X \n", *(*(p2)+1) );// Instrução I3
printf("%#X \n", *(p1)+1 );   // Instrução I4
return 0;
}
```

   a. As instruções I1 e I2 imprimem 0X62FE20 no ecrã, enquanto que a instrução I3 imprime 0x6 e a instrução I4 imprime 0x5 no ecrã.
   b. A instrução I1 imprime 0X62FE20, a instrução I2 imprime 0 no ecrã, a instrução I3 imprime 0x6 e a instrução I4 imprime 0x5 no ecrã.
   c. As instruções I1 e I2 imprimem 0X62FE20 no ecrã enquanto que as instruções I3 e I4 imprimem 0x5 no ecrã.
   d. As instruções I1 e I2 imprimem 0 no ecrã enquanto que as instruções I3 e I4 imprimem 0x3 no ecrã.

7. **Considerando o datapath do MIPS indique qual das seguintes afirmações é VERDADEIRA?**
   a. A instrução `beq $t0,$t1,-10` está inactiva na etapa 3 correspondente à unidade lógica e aritmética (ALU).
   b. A instrução `sb $s0,2($a0)` está activa na etapa 5 de escrita nos registos ("*register write*").
   c. A instrução `sll $t0,$t1,2` está inactiva na etapa 4 correspondente ao acesso à memória ("*memory access*").
   d. A instrução `lbu $t0,3($a0)` está inactiva na etapa 3 correspondente à unidade lógica e aritmética (ALU).

8. **Considere o seguinte excerto de código Assembly do MIPs em que é carregado em memória o array de inteiros num:**

```
      .data
num: .word 10,20,30,40,50,60,70,80,90,100
      .text
main:
      la    $a0,str
                    ...
```

Indique qual das instruções permite a leitura para o registo `$t0` do número `80` pertencente ao array `num` armazenado em memória.

a. `lb $t0,7($a0)`
b. `lw $t0,28($a0)`
c. `lb $t0,8($a0)`
d. `lw $a0,28($t0)`

9. **Considere o seguinte código em *Assembly* do MIPS, que pretende implementar o código equivalente ao programa em linguagem C descrita ao lado. Escolha das opções disponíveis aquela que correctamente representa o par de instruções <...> em falta no código MIPS:**

ASSEMBLY DO MIPS

```
li     $t1,1
li     $t2,50
li     $t3,1024

ciclo:
       beq $t2,$0,fim_ciclo
       <. . .>
       addi $t2,$t2,-1
       j      ciclo

fim_ciclo:   ...
```

LINGUAGEM C

```
...
int i,j;
i=1;
for (j=50;j>0;j--)
{
       i=i*2;
       if (i>=1024)
              break;
}

...
```

a. `sll $t1,$t1,1` e `blt $t1,$t3,fim_ciclo`.
b. `sll $t1,$t1,2` e `bge $t1,$t3,fim_ciclo`.
c. `srl $t1,$t1,1` e `bge $t1,$t3,fim_ciclo`.
d. `sll $t1,$t1,1` e `bge $t1,$t3,fim_ciclo`.

10. **Quantas vezes deve uma memória cache ser mais rápida do que a memória principal por forma a garantir que o tempo médio de acesso à memória seja de 10ns, assumindo que o tempo de acesso à memória principal é de 25ns e a hit rate na cache é igual a 80%.**
    a. 10 vezes mais rápida.
    b. 8 vezes mais rápida.
    c. 5 vezes mais rápida.
    d. 4 vezes mais rápida.

11. Qual das seguintes afirmações, relativas ao *gcc* e *gdb* que utilizou nas aulas práticas laboratoriais, é <u>FALSA</u>:

- **a.** O gdb permite, entre outras opções, correr o programa passo a passo, ver o estado das variáveis, definir breakpoints e analisar o ponto em que o programa falhou.
- **b.** O uso da flag –g com o compilador gcc permite ao gdb relacionar o código executável com o código fonte para fins de debugging.
- **c.** O gcc consegue compilar ficheiros em linguagem Assembly e em C.
- **d.** O uso da flag –o com o compilador gcc produz um ficheiro objecto.

12. No trabalho prático 3 utilizaram-se dois displays de 7 segmentos do simulador MARS. Para aceder o display da esquerda bastava escrever um byte no endereço 0xFFFF0011. Quais seriam as instruções a utilizar caso desejássemos colocar o número 9 nesse display?

a)
```
addi $s0, $s0, 0xFFFF0011
addi,$t0, $t0,0x6F
sb $t0,0($s0)
```

b)
```
addi $s0, $s0, 0xFFFF0011
addi,$t0, $t0,0x6F
sb 0($s0),$t0
```

d)
```
addi $s0, $s0, 0xFFFF0011
addi,$t0, $t0,0x6A
sb $t0,0($s0)
```

c)
```
addi $s0, $s0, 0xFFFF0011
addi,$t0, $t0,0x6C
sw $t0,0($s0)
```

13. Considere o seguinte programa em C em que a tabela "*tab*" começa no endereço 0x62FE20. Com base nisso, que valores serão impressos no ecrã?

- **a.** 0x62FE20, 22, 30.
- **b.** 0x62FE20, 7, 7.
- **c.** 0x62FE30, 22, 30.
- **d.** 0x62FE30, 22, 24.

```
#include <stdio.h>

int main(){
   int tab[]={1,5,10,15,20,25,30,35};
   int *ptr;
   ptr=tab+4;
   printf(" %#X, %d, %d. \n", ptr, *ptr+2, *(ptr+2));
   return 0;
}
```

14. Considere uma hierarquia de memória composta por uma memória principal com 4 *Gbytes* e por uma memória cache com 8 *Mbytes*. O tamanho assumido para cada bloco de memória dentro da cache é de 4 *Kbytes*. Qual das seguintes opções representa a estrutura de endereçamento adequada para uma *cache 4-way set-associative*.

- **a.** offset = 11 bits; index = 9 bits; tag = 12 bits.
- **b.** offset = 12 bits; index = 9 bits; tag = 11 bits.
- **c.** offset = 10 bits; index = 10 bits; tag = 12 bits.
- **d.** offset = 12 bits; index = 12 bits; tag = 8 bits.

15. Considere o datapath de uma CPU do tipo multiple-cycle com cinco etapas e com os seguintes tempos máximos de execução por etapa: 1 – instruction fetch (310 ps); 2 – instruction decode (290 ps); 3 – ALU (280 ps); 4 – memory access (370 ps); 5 – register write (275 ps). Indique qual das seguintes afirmações é <u>VERDADEIRA</u>?

- **a.** O período do relógio da CPU é definido pela média aritmética dos tempos de execução de cada etapa.
- **b.** O período do relógio é independente do tempo máximo de execução por etapa do *datapath*.
- **c.** O período do relógio desta CPU é 275 ps.
- **d.** O período do relógio desta CPU é 370 ps.