

# Arquitectura de Computadores I

1º Semestre de 2002/2003

## 1º MiniTeste

Nome: .....

Nº Mecanográfico:..... Turma:.....

1. Considere o seguinte trecho de código Assembly do MIPS:

```
add $s5, $t8, $t9
and $s4, $s5, 0xFF
bge $s4, $t2, prx
sub $s4, $s4, 2
prx: or $v1, $s4, $0
```

Admita também que o valor armazenado nos registos envolvidos neste programa, antes da sua execução, era o seguinte:

$\$t2 = 0x00000003$ ,  $\$t8 = 0x0003A580$ ,  $\$t9 = 0x00000082$ ,  
 $\$s4 = 0x08230012$ ,  $\$s5 = 0x00000000$ ,  $\$v1 = 0x80000000$

a) Dos registos envolvidos neste trecho de código, indique aqueles que não são alterados pelo programa.

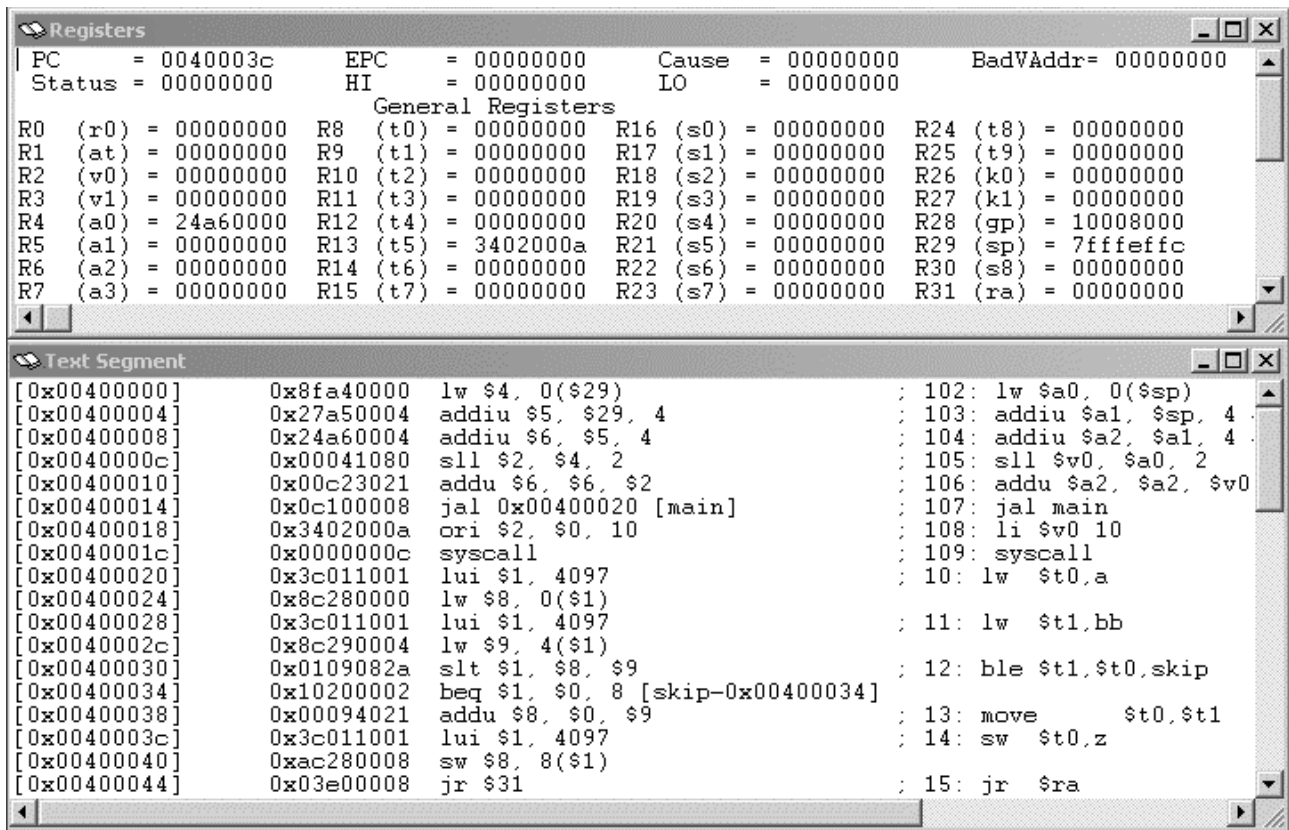
b) Determine e apresente (em hexadecimal) o valor armazenado em cada um dos restantes registos após a execução da última instrução

~~a) Os registos não alterados são os registos  $\$t2$ ,  $\$t8$  e  $\$t9$ .~~

~~b)  $\$s5 = \$t8 + \$t9 = 0x0003A580 + 0x00000082 = 0x0003A602$~~

~~$\$s4 = \$s5 \text{ and } 0xFF = 0x00000002$  (instrução 2)  $= 0x00000002 - 2 = 0$~~

~~$\$v1 = \$s4 = 0$~~



A figura representa as janelas de *Registos* e *Text Segment* do programa PCSPIM. Atendendo ao seu conteúdo responda às seguintes questões:

- ~~Indique o endereço da próxima instrução a ser executada, o seu código máquina (em hexadecimal) e a instrução Assembly respectiva (da máquina nativa).~~
- ~~Descreva, de forma sucinta, a operação efectuada por essa mesma instrução.~~
- ~~Determine, justificando, o endereço da memória externa referenciado pela penúltima instrução presente na janela de *Text Segment*.~~

- SPC = 0x0040003C. Instrução 0x3C011001. Em Assembly lui \$1, 4097**
- Coloca o valor da constante indicada nos dezasseis bits mais significativos do registo \$1. Os dezasseis bits menos significativos de \$1 tomam o valor zero.**
- Penúltima instrução: sw \$8, 8(\$1). O endereço obtém-se somando a constante 8 ao conteúdo de \$1. Este conteúdo, por sua vez é determinado pela instrução imediatamente anterior (lui \$1,4097). Logo o endereço será 0x10010000 + 0x00000008 = 0x10010008**

d) 3. Considere o seguinte trecho de código C:

```
unsigned int n, val=-0x33;  
char symbols[35];  
char prompt[] = {"\nDigite uma data: "};  
void main (void)  
{  
    ...  
}
```

Traduza, em directivas do Assembly do MIPS, o código apresentado. Assuma, para isso, que todas as variáveis residem na memória externa.

	<b>.data</b>	<b>15</b>
<b>n:</b>	<b>.space 4</b>	<b>20</b>
<b>val:</b>	<b>.word -0x33</b>	<b>20</b>
<b>symbols:</b>	<b>.space 35</b>	<b>25</b>
<b>prompt:</b>	<b>.asciiz "\nDigite uma data: "</b>	<b>20</b>

4. Considere agora o seguinte trecho de código C:

```
void main (void)
{
    int a, b, i, j;
    ...
    if (a >= b){
        j = i + a;
    }
    else {
        j = i * 2;
        b--;
    }
}
```

Traduza, em Assembly do MIPS, o código apresentado. Assuma, para isso, que as variáveis **a**, **b**, **i** e **j** residem respectivamente nos registros **\$t0**, **\$t3**, **\$t9** e **\$s0**. Comente adequadamente o seu código.

10	.text		
	.global main		
	main:		# void main(void)
	...		# { ...
25	blt \$t0, \$t3, else		# if (a >= b){
15	add \$s0, \$t9, \$t0		# j = i + a;
15	j endif		# }else {
15 else:	sll \$s0, \$t9, 1		# j = i * 2;
10	sub \$t3, \$t3, 1		# b--;
	endif:		# }
10	jr \$ra		# }