

1ª Questão)

1ª Questão) (1.0 ponto) Considere que o código em C da função `int Qx(int)` a seguir foi compilado para MIPS

<pre>int Qx (int i) { (... lógica em C ...); } Código MIPS compilado: Qx: li \$t1, 32 li \$s0, 0 li \$s1, 32 add \$s0, \$s0, \$a0 L1: addiu \$s0, \$s0, 1 sub \$s1, \$t1, \$s0 bne \$s0, \$t1, L1 jr \$ra</pre>	<p>A variável <code>i</code> foi armazenada em <code>\$a0</code> e que os resultados gerados pela função <code>Qx</code> estarão em <code>\$s0</code> e <code>\$s1</code>, valores literais (variáveis) no espaço de memória global.</p> <p>As instruções <code>li</code> e <code>addiu</code> correspondem, respectivamente, a um <i>load</i> imediato (constante) e um <i>add</i> imediato sem sinal (constante positiva).</p> <p>a) Traduza o código MIPS para C</p> <p>b) O que faz exatamente essa sequência de código em <code>Qx</code>?</p> <p>c) Quais os valores finais armazenados em <code>\$s0</code> e <code>\$s1</code>? Especifique os valores em função do parâmetro <code>i</code> da função <code>Qx</code>.</p>
--	---

```
int Qx (int i){
```

```
int t1=32, s0 = 0, s1 = 32;
```

```
s0 = s0 +i;
```

```
do{
```

```
s0 = s0 +1;
```

```
s1 = t1 - s0;
```

```
} while (s0 ~= t1};
```

b) $i < 32$

```
s1 = 0
```

```
s0 = 32;
```

$i \geq 32$

loop infinito

c)

$i < 32$

```
s1 = 0
```

```
s0 = 32;
```

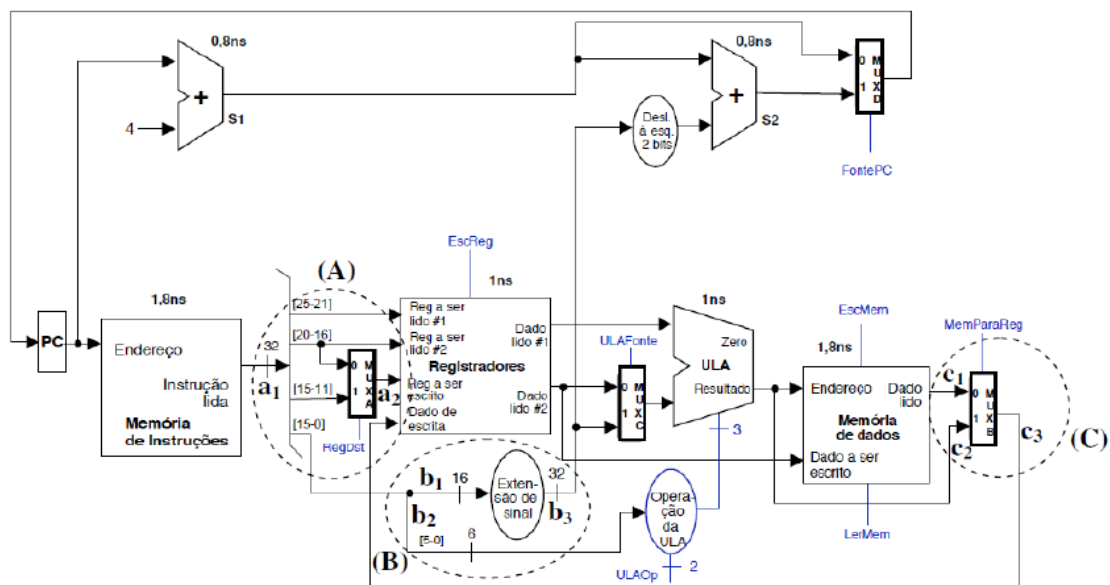
$i \geq 32$

loop infinito

$s0 = i + 1$ (vai para infinito)

$s1 = 32 - (i+1)$ (vai para $-\infty$)

Endereço	Instrução	Observações
FA00:0000	lw \$s0,60(\$t0)	Os registradores \$t0-\$t7 são numerados de 8 a 15 (temporários) e os de \$s0-\$s7 de 16 a 23. (valores salvos/armazenados). O opcode de lw é 35. O opcode da instrução sub é 0, com 34 nos bits [5-0]. O valor (em hexa) de \$t0 = 8AFF:0000. Os valores carregados em \$s0 e \$s0 após a execução da instrução lw serão, respectivamente, 12 e 25. Campos de bits não utilizados/desnecessários para a classe instrução terão valor 0.
FA00:0004	lw \$s1,64(\$t0)	
FA00:0008	sub \$s2,\$s0,\$s1	



2ª Questão)

	Opcode	Rs	Rt	Rd	Shant	Funct
lw \$s0,60(\$t0)	35	8	16	60		
lw \$s1,64(\$t0)	35	8	17	64		
sub \$s2,\$s0,\$s1	0	16	17	18	0	34
Posição	31-26	25-21	20-16	15-11	10-6	5-0

	Opcode	Rs	Rt	Rd	Shant	Funct
lw \$s0,60(\$t0)	100011	01000	10000	0000000000111100		
lw \$s1,64(\$t0)	100011	01000	10001	0000000001000000		
sub \$s2,\$s0,\$s1	000000	10000	10001	10010	00000	100010
Posição	31-26	25-21	20-16	15-11	10-6	5-0

lw \$s0,60(\$t0)

a1	100011010001000000000000000111100
a2	10000
b1	0000000000111100
b2	111100
b3	000000000000000000000000000111100
c1	0000000000000000000000000001100 – valor 12
c2	10001010111111110000000000111100
c3	0000000000000000000000000001100

c1 – valor 12

0000000000000000000000000001100

\$t0 = 8 A F F 0 0 0 0

10001010111111110000000000000000

0000000000000000000000000111100

=

10001010111111110000000000111100

lw \$s1,64(\$t0)

a1	10001101000100010000000001000000
a2	10001
b1	0000000001000000
b2	000000
b3	00000000000000000000000001000000
c1	00000000000000000000000000011001 - valor 25
c2	10001010111111110000000001000000
c3	00000000000000000000000000011001

\$t0 = 8 A F F 0 0 0 0

10001010111111110000000000000000

00000000000000000000000001000000

=

10001010111111110000000001000000

sub \$s2,\$s0,\$s1

a1	00000010000100011001000000100010
a2	10010
b1	1001000000100010
b2	100010
b3	00000000000000001001000000100010
c1	xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
c2	-13
c3	111111111111111111111111111110011

Complemento de dois

00000000000000000000000000001101 - 13

00000000000000000000000000000001

=00000000000000000000000000001100

Inverter

111111111111111111111111111110011 – 13

Transforma para real

00000000000000000000000000001100

Soma 1

00000000000000000000000000001101 - 13

Resultado -13 (Complemento dois)

00000000000000000000000000001101 - 13

111111111111111111111111111110011

00000000000000000000000000001100 12

000000000000000000000000000011001 - 25

= 12 -25

8 A F F 0 0 0 0

10001010111111110000000000000000 \$t0

00000000000000000000000001000000 \$64

=

10001010111111110000000001000000

a, b, c, d, i estão nos registrados \$t0, \$t1, \$t2, \$t3

V[] - \$s0

Enquanto i<100 v[i+1] ← v[i] +1; se v[i+1] <> 10 a++; fim se; i ← i+1; b ← c*2 -d; fim enquanto	li \$t4,100 TESTE: slt \$t5,\$t3,\$t4 beq \$t5, \$zero, FIM add \$t6,\$s0, \$t3 lw \$s1, (0)\$t6 addi \$s2, \$s1,1 sw \$s2, (1)\$t6 addi \$s3, \$zero,10 bneq \$s2, \$s3, ELSE addi \$t0, \$t0, 1 ELSE: addi \$t3, \$t3, 1 add \$t1, \$t2,\$t2 sub \$t1, \$t1, \$t2 jump TESTE FIM
---	--