

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

```
int Qx (int i) {
    (... lógica em C ...);
}
```

Código MIPS compilado:

Qx:

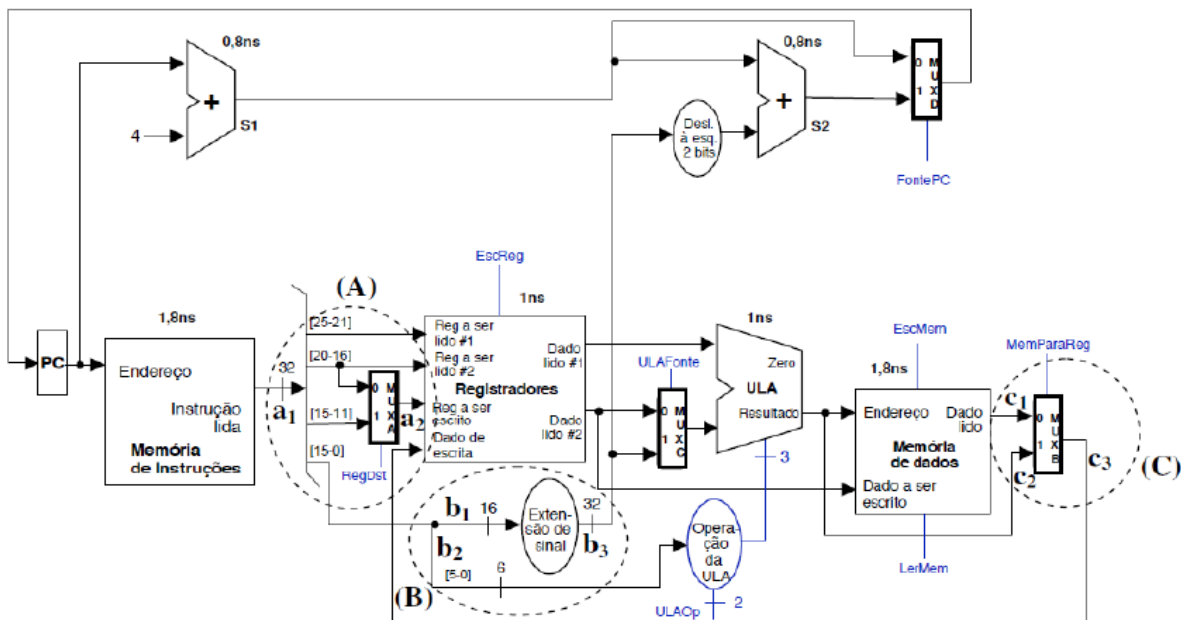
```
li $t1, 32
li $s0, 0
li $s1, 32
add $s0, $s0, $a0
L1: addiu $s0, $s0, 1
sub $s1, $t1, $s0
bne $s0, $t1, L1
jr $ra
```

A variável `i` foi armazenada em `$a0` e que os resultados gerados pela função `Qx` estarão em `$s0` e `$s1`, valores literais (variáveis) no espaço de memória global.

As instruções `li` e `addiu` correspondem, respectivamente, a um `load` imediato (constante) e um `add` imediato sem sinal (constante positiva).

- Traduza o código MIPS para C
- O que faz exatamente essa sequência de código em `Qx`?
- Quais os valores finais armazenados em `$s0` e `$s1`? Especifique os valores em função do parâmetro `i` da função `Qx`.

2ª Questão) Suponha que a seguinte sequência de instruções deve ser executada nesta arquitetura



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

Mostre quais os valores dos conjuntos de bits (a_1 a c_3) que estão sendo enviados em cada uma das partes

| | Opcode | Rs (\$t0) | Rt (\$s0) | Rd | Shant | Funct |
|---------------------------------|--------|-----------|-----------|-------|-------|-------|
| <code>lw \$s0,60(\$t0)</code> | | | | | | |
| <code>lw \$s1,64(\$t0)</code> | | | | | | |
| <code>sub \$s2,\$s0,\$s1</code> | | | | | | |
| Posição | 31-26 | 25--21 | 20-16 | 15-11 | 11-6 | 5-0 |