



1. **(1.0)** Apresente e descreva as etapas do ciclo de instrução.

2. **(3.0)** Utilizando o assembly do MIPS implemente uma função `multiplica` que recebe em `$a0` e `$a1` dois números inteiros e calcula a multiplicação utilizando o algoritmo de somas sucessivas apresentado ao lado. A função retorna em `$v0` o resultado.

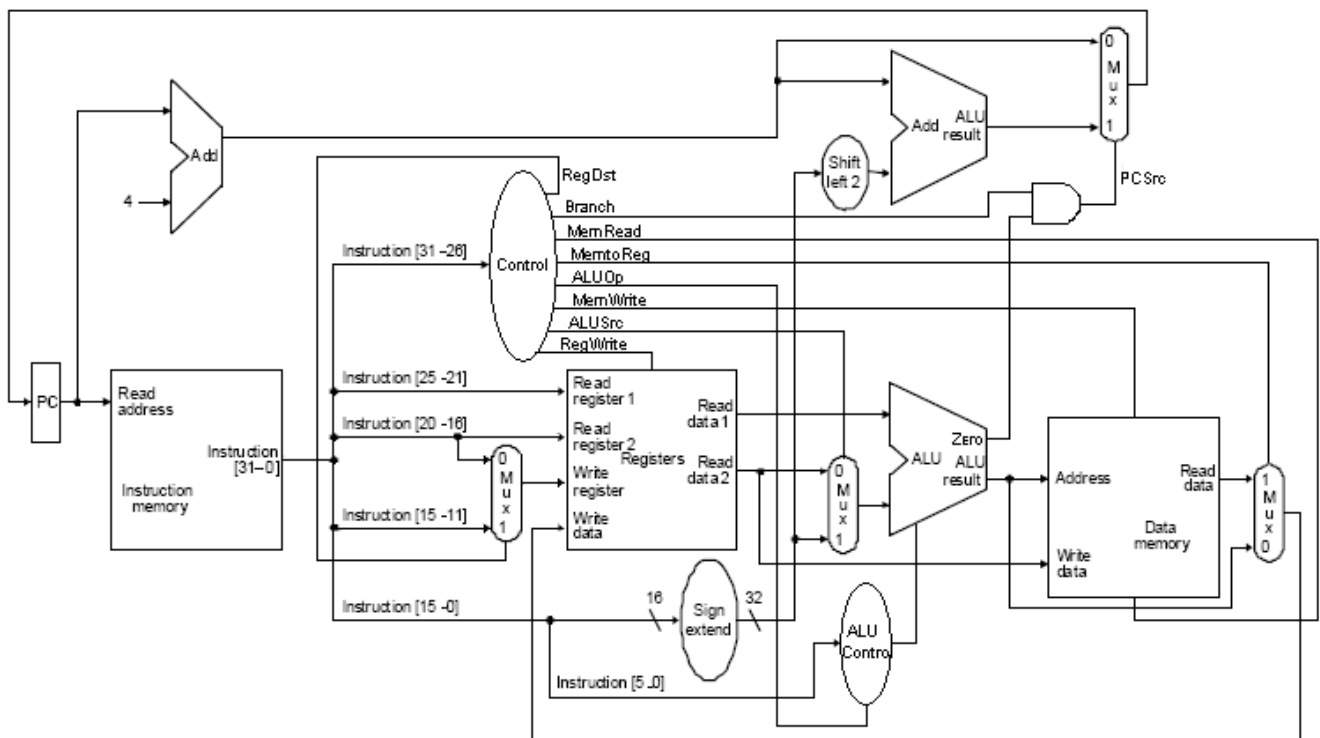
```
int multiplica(int a, int b){
    int soma = 0, i = 0;
    while ( i != a ) {
        soma = soma + b;
        i = i + 1;
    }
    return soma;
}
```

3. **(3.0)** Considerando o processador monociclo mostrado abaixo:

a) acrescente o que for necessário em termos de elementos ao caminho de dados e também sinais de controle para que o mesmo possa executar também a instrução `SLTI $rd, $rs, offset # se($rs < $offset){ $rd = 1 } else { $rd = 0 }`; Justifique.

b) acrescente o que for necessário em termos de elementos ao caminho de dados e também sinais de controle para que o mesmo possa executar também a instrução `JAL offset # $ra ← PC+4; PC ← PC[31-28]:(offset << 2)`; Justifique.

c) apresente os valores de todos os sinais de controle (`RegDst`, `Branch`, `MemRead`, `MemtoReg`, `ALUOp`, `MemWrite`, `ALUSrc`, `RegWrite`) inclusive os novos sinais de controle, para a execução das instruções providas em (a) e (b).



4. (3.0). Sabendo que o rótulo `inicio` foi armazenado no endereço de memória 65540 e considerando o formato das instruções do processador MIPS, qual o código de máquina (binário) deste programa?

```
                addi $vo, $zero, -4
inicio:         slt $t1,$v0,$s6
                beq $t1,$zero, rot
                add $t6,$zero,$t0
                lw  $t1, 20 ($s6)
                j   inicio
rot:            sw $t1, ($s7)
```

Registradores	
Nome	Número
\$zero	0
\$at	1
\$v0-\$v1	2-3
\$a0-\$a3	4-7
\$t0-\$t7	8-15
\$s0-\$s7	16-23
\$t8-\$t9	24-25
\$k0-\$k1	26-27
\$gp	28
\$sp	29
\$fp	30
\$ra	31

Convenções ALUOP: 00 - SW/LW
01 - BEQ
10 - Formato R

Convenções U.C. ULA: 000 - AND
001 - OR
010 - ADD
011 -
100 -
101 -
110 - SUB
111 - SLT