



d₀ d₁ / d₂ d₃ d₄ d₅ d₆ d₇ d₈

Nome: GABARITO Matrícula: 19/0012345

Prova 1

(4.5) 1) Dado o programa em C ao lado.

(2.5)a) Respeitando a convenção do uso de registradores, compile eficientemente para Assembly RISC-V ISA RV32I as rotinas main() e SOMAP().

Dica: Use os serviços do sistema (ecall) para escrever na tela.

(2.0)b) Para um processador RISC-V, onde as instruções de acesso à memória e os desvios (condicionais e incondicionais) possuem CPI=2 e todas as outras instruções possuem CPI=1, com frequência de clock de 1GHz, responda:

```
int SOMAP(int v[], int n){
    int i,soma=0;
    for(i=0;i<n;i++)
        if (v[i]%2==0)
            soma=soma+v[i];
    return soma;
}

void main(){
    int A[6]={1,5,4,3,2,8};
    printf("soma=%d",SOMAP(A,6));
}
```

(1.0)b.1) Qual a CPI média obtida para o seu procedimento SOMAP? _____

(1.0)b.2) Qual o tempo necessário à execução do seu procedimento SOMAP? _____

(3.0) 2) Considere o procedimento PROC abaixo sendo chamado com o argumento a0=0x10.

PROC: lui t0,0x02970

sra a0,t0,a0

bgt a0,zero,FIM

j PROC

FIM: ret

0X 02 97 02 B7

0X 40 A2 D5 33

0X 00 A0 44 63

0X FF 5F F0 6F

0X 00 00 80 67

(2.0)a) Escreva o código em linguagem de máquina, **em hexadecimal**, ao lado de cada instrução acima.

(1.0)b) Se o valor retornado no registrador a0 for tratado como uma instrução da ISA RISC-V, qual o mnemônico completo desta instrução? auipc t0, 0x00000

(2.0) 3) A ISA RV32I não possui as instruções de multiplicação, divisão e módulo (resto da divisão). No entanto, isto não significa que uma linguagem de alto nível não possa usar essas operações matemáticas neste processador. Implemente a instrução abaixo como uma pseudoinstrução da ISA RV32I:

(2.0) a) mul t1,t2,t3 # t1=Lower{t2×t3}

(1.5) 4) Responda:

(1.0) a) Por que é usada a média geométrica dos tempos de execução normalizados para resumir o desempenho de um Benchmark?

Porque é uma medida consistente e independente da máquina base.

(0.5) b) Dê sugestões para melhorar o índice de aprovação desta disciplina.

Questão 1

.data

V: .word 1,5,4,3,2,8

STRING: .string "soma="

} 0,2

STRING2: .string "soma=14" # Mais eficiente!

.text

```

MAIN: la a0,STRING      # imprime "soma="
      li a7,4
      ecall

```

} 0,2

```

      la a0,V
      li a1,6
      jal SOMAP          # retorno em a0

```

} 0,3

```

      li a7,1            # print int
      ecall

```

} 0,3

```

      li a7,10           # exit
      ecall

```

```

SOMAP: li t0,0           # i=0
      li t1,0           # soma=0

```

} 0,3

```

LOOP: bge t0,a1,SAIR     # Verifica se i>=n
      slli t2,t0,2       # i*4
      add t2,t2,a0       # endereço de v[i]
      lw t2,0(t2)        # conteúdo de v[i]
      andi t3,t2,1       # mascara o bit menos significativo
      bne t3,zero,IMPAR  # verifica se é ímpar então pula
      add t1,t1,t2       # soma se for par

```

} 1,0

```

IMPAR: addi t0,t0,1      # próximo i
      j LOOP            # volta ao LOOP
SAIR:  mv a0,t1          # coloca soma em a0
      ret               # retorna

```

} 0,2

#Questão 3

```
.text
li t2,0xffff
li t3,0xffff
mul s0,t2,t3    # resultado correto

# mul t1,t2,t3
    addi sp,sp,-12    # reserva espaco na pilha
    sw t2,0(sp)      # salva t2
    sw t3,4(sp)      # salva t3
    sw t4,8(sp)      # salva t4
    li t1,0          # zera o resultado
LOOP: beq t2,zero,FIM  # verifica se t2 eh zero
    andi t4,t3,1      # verifica bit menos significativo de t3
    srli t3,t3,1      # desloca t3 para direita
    beq t4,zero,PULA  # se lsb for zero, PULA
    add t1,t1,t2      # se lsb for 1 soma t1 ao resultado
PULA: slli t2,t2,1     # desloca t2 para esquerda
    bne t3,zero,LOOP  # verifica se t3 é diferente de zero
FIM:  lw t2,0(sp)     # recupera t2
    lw t3,4(sp)      # recupera t3
    lw t4,8(sp)      # recupera t4
    addi sp,sp,12     # libera espaco da pilha
```

Handwritten annotations:

- A bracket on the right side of the first three instructions (sw t2,0(sp), sw t3,4(sp), sw t4,8(sp)) is labeled 0,5.
- A bracket on the right side of the instructions from LOOP: andi t4,t3,1 to PULA: bne t3,zero,LOOP is labeled 1,0.
- A bracket on the right side of the instructions from FIM: lw t2,0(sp) to FIM: addi sp,sp,12 is labeled 0,5.