

Nome: GABARITO

Matrícula: 12/3456789

Prova 1

(4.0)1) Dado o programa em C ao lado:

(1.0)a) Compile-o para Assembly MIPS, respeitando a convenção do uso dos registradores.

(2.0)b) Monte o programa em linguagem de máquina (em **hexadecimal**) a partir do endereço 0x00400000

(1.0)c) Para um processador de 1GHz, em que as instruções que envolvem acesso à memória de dados exigem 3 ciclos, instruções com operações com a ULA exigem 2 ciclos e instruções de desvio incondicional 1 ciclo, qual o desempenho do algoritmo mdc para os valores de i e j dados?

```
void main(void) {  
    unsigned int i=0x0d3d4d5, j=0x00d4d5;  
    printf("%d", (int)mdc(i,j));  
}  
  
unsigned int mdc(unsigned int i, unsigned int j) {  
    while (i != j)  
        if (i > j) i -= j;  
        else j -= i;  
    return i;  
}
```

(3.5)2) As operações em ponto flutuante foram um marco na computação científica, permitindo que números reais em uma ampla faixa dinâmica fossem representados em um número finito de bits. Dada a representação em ponto flutuante IEEE 754 precisão simples:

(1.0)a) Quando comparada com a representação de números inteiros de 32 bits em complemento de 2, qual das duas representações possibilita representar a maior quantidade de números diferentes? Por quê?

(0.5)b) Dado o número 0xFF8d₃d₄d₅d₆ em IEEE754 precisão simples, qual o número decimal correspondente?

(0.5)c) Dado o número de Avogrado 6.02×10^{23} , qual a sua representação em ponto flutuante precisão simples?

(1.5)d) Projete um diagrama de blocos de um circuito combinacional que possibilite efetuar uma operação de multiplicação de dois números em ponto flutuante precisão simples. Obs.: Vc dispõe de blocos que realizem as operações de multiplicação e adição de números inteiros 32 bits representados em complemento de 2. Dica: Não se preocupe com overflows e truncamentos.

(3.5)3) As arquiteturas ARM e x86 não reservam um registrador específico para o armazenamento do endereço de retorno em chamadas de procedimentos, mas usam a pilha para o armazenamento deste endereço. Implemente na arquitetura MIPS as pseudo-instruções abaixo que visam aproxima-la daquelas arquiteturas, onde LABEL corresponde a um número de 32 bits.

(0.5)a) call LABEL # Chama o procedimento em LABEL salvando o endereço de retorno na pilha

(0.5)b) ret # recupera o endereço de retorno da pilha e retorna

(1.0)c) callz \$t1, LABEL # se \$t1=0 chama o procedimento LABEL

(1.0)d) callc1f 7, LABEL # se a flag 7 do processador C1 for false chama o procedimento LABEL

(0.5)e) Cite uma vantagem e uma desvantagem do uso dessa metodologia quando comparada à adotada pela arquitetura MIPS.

BOA SORTE!

OAC-A 1º Prova 2015/1

GABARITO

	1)		b)
0,1	a) main:	addiu \$a0, \$ZERO, 0x0d4d5	0x004000 24040375
		addiu \$a1, \$ZERO, 0x00d4d5	00400004 24050045
0,1		jal mdc	08 0c100008
0,1		add \$a0, \$ZERO, \$v0	0c 00022020
		li \$v0, 1	10 24020001
		syscall	14 0000000c
0,1		li \$v0, 10	18 2402000A
		syscall	1c 0000000c
			ciclos
0,1	mdc:	beq \$a0, \$a1, Fim	1 20 10850006
		SLTM \$t0, \$a1, \$a0	2 24 00A4402B
0,1		beq \$t0, \$ZERO, Pula	1 28 11000002
0,1		sub \$a0, \$a0, \$a1	2 2c 00852022
		j mdc	1 30 08100008
0,1	Pula:	sub \$a1, \$a1, \$a0	2 34 00A42822
		j mdc	1 38 08100008
0,1	Fim:	add \$v0, \$ZERO, \$a0	2 3c 00041020
0,1		jr \$ra	1 40 03E00008

c) Acesso memória 3 ciclos

TIPO 2 ciclos

desvios 1 ciclo

$$T_{exec} = I \times CFI \times T$$

$$\eta = \frac{1}{T_{exec}} = \frac{1}{\text{Ciclos} \times \frac{1}{16}} = \frac{1}{\text{Ciclos}} \times 10^9 =$$

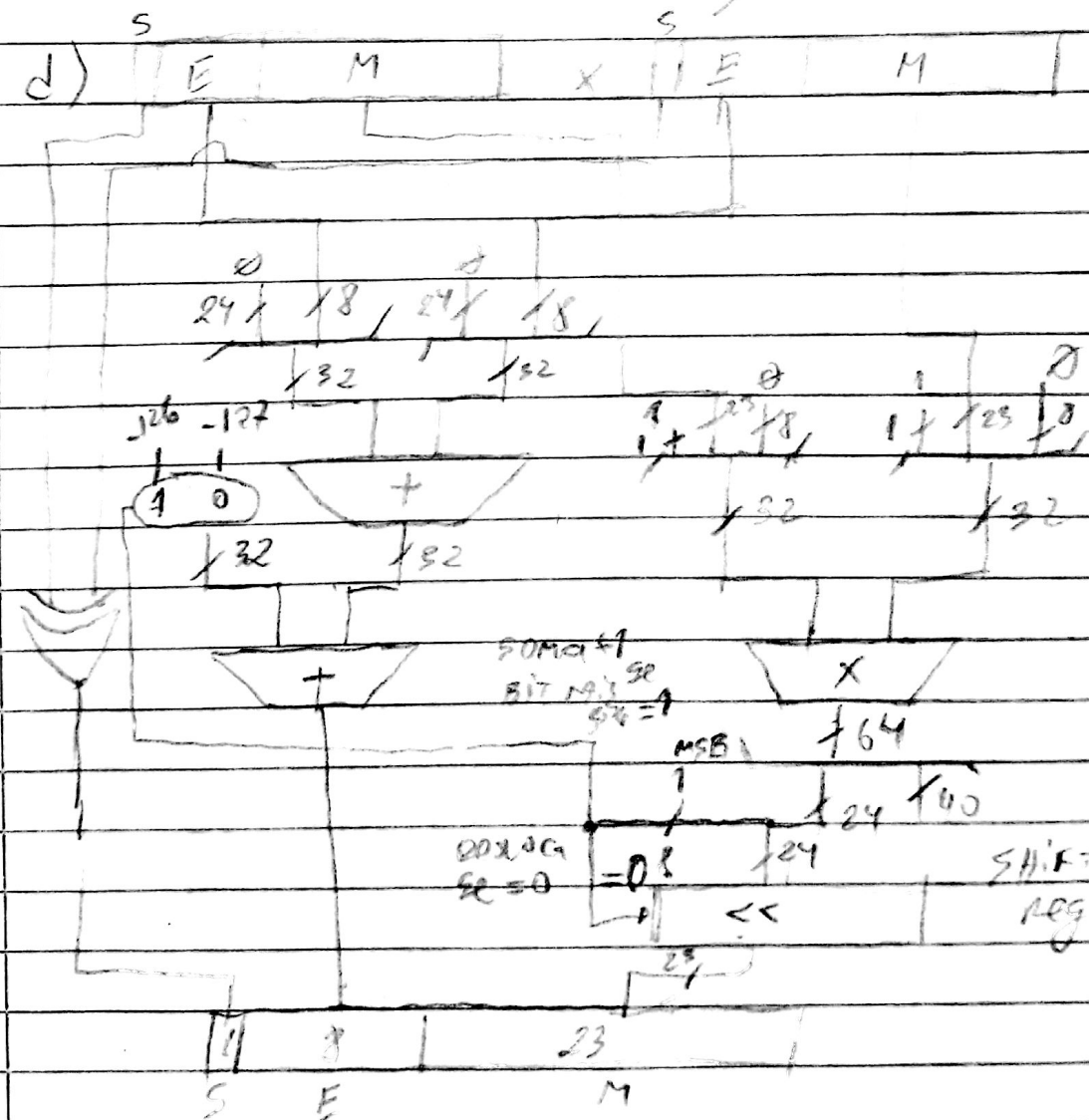
2)

a) A representação de inteiros possibilita uma maior quantidade de números diferentes, pois a IEEE754 perde representação numérica para representar os objetos ∞ e NaN.

b) $0xFF8d2d3d4d5d6$

SIGN, EXP, MANT, SE = 0 $\rightarrow -\infty$
 1 1111111 000 d2d3d4d5d6 SE = 1 \rightarrow NaN

c) $6,02 \times 10^{23} = 0x66FEF4F9$



3) CALL LABEL \Rightarrow JAL RAL

MUDA \$ra

RAL: addi \$ra, \$ra, 16

addi \$sp, \$sp, -4

sw \$ra, 0(\$sp)

J LABEL

0,5

b) RET \Rightarrow LW \$at, 0(\$sp)

addi \$sp, \$sp, 4

jr \$at

0,5

c) CALLZ \$t1, LABEL \Rightarrow bne \$t1, \$ZERO, CONT

JAL RAL

RAL: addi \$ra, \$ra, 16

addi \$sp, \$sp, -4

sw \$ra, 0(\$sp)

J LABEL

CONT: ---

1,0

d) CALLC IF 7, LABEL \Rightarrow bcc 7, CONT

JAL RAL

RAL: addi \$ra, \$ra, 16

addi \$sp, \$sp, -4

sw \$ra, 0(\$sp)

J LABEL

CONT: ---

1,0

e) -Vantagem: procedimentos que chamam outros

procedimentos e procedimentos recursivos

são implementados naturalmente

-Desvantagem: necessita acesso à memória de dados \rightarrow mais lento

0,5