



Nome: GABANITO

$d_0 \ d_1 / d_2 \ d_3 \ d_4 \ d_5 \ d_6 \ d_7 \ d_8$

Matrícula: / / / / / / / /

Prova 1

(7.0) 1) A CPU MIPS dispõe de dois coprocessadores para auxilia-la nas operações em ponto flutuante e tratamento de exceções. O coprocessador 0 do MIPS trata das exceções. A arquitetura MIPS trabalha com interrupção não-vetorizada, isto é, a rotina de tratamento para todas as exceções e interrupções está localizada a partir do endereço 0x80000180. Dado que para acesso ao C0 o MIPS possui a instrução abaixo:

`mfc0 $t0, $Rc0 # copia para o registrador $t0 o valor do registrador $Rc0 do C0. O campo Exception Code do registrador Rc0=13 (CAUSE) contém os flags que indicam a causa da exceção, e o registrador Rc0=14 (EPC) contém o endereço da memória da instrução que causou a exceção. (vide folha de referência).`

(1.0)a) Especifique a tabela verdade do circuito detector de overflow da ULA do processador MIPS desenvolvida em aula.

(2.0)b) Escreva uma rotina de tratamento de exceção que detecte apenas a exceção de overflow, escrevendo na tela “Overflow Error\nAddress: %d\nInstruction:%d \n”, onde %d é o endereço e o código de máquina, em decimal, da instrução responsável pelo overflow, e retorna para continuar executando o programa do usuário. (Se a exceção não for overflow simplesmente retorna).

Dica: a rotina de tratamento de exceção é parte do Kernel do Sistema Operacional, logo, se encontra nos segmentos .ktext (0x80000180) e .kdata(0x90000000). Considere as chamadas do sistema (syscall) implementadas normalmente.

(2.0)c) Escreva em hexadecimal no quadro em anexo o mapa da memória de dados e de código da rotina de tratamento de exceção.

(2.0)d) Com a rotina de tratamento de exceção implementada, o que é escrito na tela pelo código ao lado? Onde o label MAIN corresponde ao endereço .text (0x00400000).

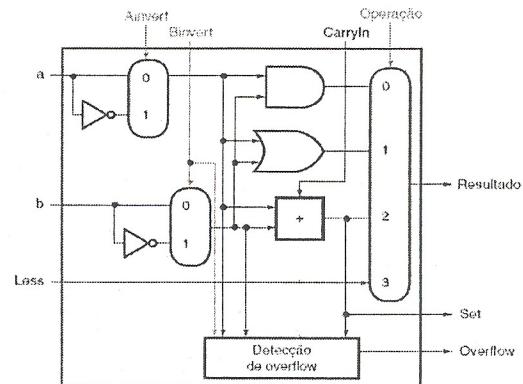
(2.0) 2) O montador Mars implementa várias pseudo-instruções Assembly que são muito úteis aos programadores. Escreva o código real das pseudo-instruções abaixo:

(1.0) a) `ror $t0,$t1,10 # rotaciona à direita a word em $t1 por 10 bits (shamt max 32) e retorna o valor em $t0`

(1.0) b) `rol $t0,$t1,10 # rotaciona à esquerda a word em $t1 por 10 bits (shamt max 32) e retorna o valor em $t0`

(1.0) 3) Nos modernos PCs com processadores baseados na arquitetura x86, a placa-mãe define o chipset. Explique o que é este chipset e porque um dos chips (se forem separados) possui um dissipador de calor maior.

(1.0) 4) Suponha que melhoramos uma máquina fazendo todas as instruções de ponto flutuante serem executadas cinco vezes mais rápido. Estamos procurando um benchmark para mostrar essa nova unidade de ponto flutuante e queremos que o benchmark geral mostre um aumento de velocidade de 3 vezes. Um benchmark que estamos considerando é executado durante 100 segundos com o hardware de ponto flutuante antigo. Quanto do tempo de execução as instruções de ponto flutuante teriam que considerar para produzir nosso aumento de velocidade desejado nesse benchmark?



MAIN:
`la $t1,0x80000000
add $t2,$t1,$t1
add $t1,$t2,$t1
addu $t3,$t1,$t1
li $v0,10
syscall`



Universidade de Brasília

Departamento de Ciência da Computação

Disciplina: CIC 116394 – Organização e Arquitetura de Computadores

Prof. Marcus Vinicius Lamar

Nome: Grabonito Matrícula: _____

Endereço	Conteúdo							
0x80000180	4	0	1	A	6	8	0	0
0x80000184	4	0	1	B	7	0	0	0
188	2	3	5	A	F	F	D	0
18C	1	7	4	0	0	0	1	2
190	3	C	0	1	9	0	0	0
194	3	4	2	4	0	0	0	0
198	2	4	0	2	0	0	0	4
19C	0	0	0	0	0	0	0	C
1A0	0	0	1	B	2	0	2	1
1A4	2	4	0	2	0	0	0	1
1A8	0	0	0	0	0	0	0	C
1AC	3	C	0	1	9	0	0	0
1B0	3	4	2	4	0	0	1	A
1B4	2	4	0	2	0	0	0	4
1B8	0	0	0	0	0	0	0	C
1BC	8	F	6	4	0	0	0	0
1C0	2	4	0	2	0	0	0	1
1C4	0	0	0	0	0	0	0	C
1C8	3	C	0	1	9	0	0	0
1CC	3	4	2	4	0	0	2	9
1D0	2	4	0	2	0	0	0	4
1D4	0	0	0	0	0	0	0	C
1D8	2	3	7	B	0	0	0	4
1DC	0	3	6	0	0	0	0	8

DAC-A

2012/1

1º Prova

Gabarito

1) a) Tabela verdade do circuito de detecção
de overflow da VLA:

BInv	A[31]	B[31]	RESULT[31]	Overflow
0	0	0	0	0
0	0	0	1	1 $A+B < 0$
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	1 $(-A)+(-B) > 0$
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1 $A-(-B) < 0$
1	1	0	0	1 $(-A)-B > 0$
1	1	0	1	0
1	1	1	0	0
1	1	1	1	0

$B_{Inv} = 0 \rightarrow$ soma \rightarrow SUB

$A[31] = 0 \quad A > 0$

$B[31] = 0 \quad B > 0$

$A[31] = 1 \quad A < 1$

$B[31] = 1 \quad B < 0$

$RESULT[31] = 0 \quad R > 0$

$RESULT[31] = 1 \quad R < 0$

b) Excessão de overflow; CODE = 12

.KDATA

STR1: .ASCII "Overflow Error in Address;"

STR2: .ASCII "in instruction;"

STR3: .ASCII "n"

.KTEXT

MFCO \$K0, \$13 # K0 = CAUSE

ANDI \$K0, \$K0, 0x007C # MISCANA

MFCO \$K1, \$14 # K1 = EPC Saida

addi \$K0, \$A0, -48 # CODE = 12? 12x4 = 48

bne \$K0, \$ZERO, SAIPA # não, sa;

* La \$A0, STR1 \rightarrow LW \$K0, 0(\$K1)

Li \$V0, 4

SYSCALL

move \$A0, \$K1

Li \$V0, 1

SYSCALL

La \$A0, STR2

Li \$V0, 4

SYSCALL

move \$A0, \$K0)

Li \$V0, 1

SYSCALL

La \$A0, STR3

Li \$V0, 4

SYSCALL

SAIDA: addi \$K1, \$K1, 4 # próxima instrução
jr \$K1 ou mtco \$K1, \$14

ERET

SE O Sistema
Permitir

mtz
correta

c) No quando

0x90000000 Over Flow YEAR ORVA

0x90000000 4F766572 666C6F77 20457272 6F720A41

DDRE SS:U 10 MIN STRU.

0x90000020 64647265 73733A20 000A496E 73747275

CTIO n:U 10 MIN

0x90000040 6374696F 6E3A2000 0A000000

d) Só a instrução: add \$t2,\$t1,\$t1 causa overflow

Ld:0.

Overflow Error

ADDRESS: 4194312 \Rightarrow 0x00400008

INSTRUCTION: 19484704 \Rightarrow 0x01295020

2) a) ROR \$t0,\$t1,10

SLL \$at,\$t1,22 \neq 32-10

SRL \$t0,\$t1,10

OR \$t0,\$t0,\$at

b) ROL \$t0,\$t1,10

SRL \$at,\$t1,22 \neq 32-10

SLL \$t0,\$t1,10

OR \$t0,\$t0,\$at

3) CHIPSET: SÃO OS CIRCUITOS QUE CONVERTEM OS DISPOSITIVOS EXTERNO AO PROCESSADOR.

PORTE NORTE → DISPOSITIVOS + NÓS PPS

PONTO SUL → DISPOSITIVOS + LEITORES

A PORTE NORTE PODE TRABALHAR A UMA FREQ. MAIOR POSSUÍ MAIOR DISSIPACAO TERMICA.

4) NOVA MÁQUINA FP → 5X

BENCHMARK AUMENTO GERAL DE 3X

T=100 s

LEITORE AMDA1:

$$\frac{100}{3} = (1-P)100 + \frac{P \times 100}{5}$$

$$33,3 = 100 - 100P + 20P$$

$$80P = 66,66 \rightarrow P = 0,83 \rightarrow 83,3\%$$