UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2023

PRIMEIRA PROVA

Escola	EACH	TURMA		Nota do aluno na PROVA
Curso	Sistemas de Informação			
Disciplina	Organização e Arquitetura de Computador II	Data da Prova	29/11/23	
Professor	Clodoaldo Aparecido de Moraes Lima			
Aluno				
No. USP				

1a Questão) (**2.0 Pontos**) Escreva um programa em linguagem de montagem (assembly language) do processador MIPS que processa um vetor de números inteiros (VET), transformando cada elemento do vetor em seu valor absoluto. Por exemplo, o valor absoluto do número -43 é +43 e do número +55 é +55. O programa deve obrigatoriamente chamar uma sub-rotina calc_abs que recebe como parâmetro um número e calcula o valor absoluto deste, retornando-o segundo as convenções do MIPS. O valor retornado pela sub-rotina deve ser armazenado, pelo programa principal, na mesma posição do array que continha o elemento original. Utilize a área de dados abaixo para escrever o seu programa.

.data
VET: .word 23 -43 55 -9 -7 21 -76 12 -45 -10
TAM: .word 10

```
main:
     la $s0,VET
     la $s1,TAM
     lw $s1,0($s1)

loop:
     beq $s1,$zero,fim
     lw $a0,0($s0)
     jal calc_abs
     sw $v0,0($s0)
     addiu $s1,$s1,-1
     addiu $s0,$s0,4
     j loop
fim:
     li $v0,10
     syscall
calc_abs:
     move $v0,$a0
     bgez $v0,fim_ca
     subu $v0,$zero,$v0
fim_ca:
     jr $ra
```

```
0,5 carrega vet, tam
0,5 codigo principal
0,5 calc_abs
0,2 uso jal
0,3 finaliza

Não carrega TAM -0.1
```

UNIVERSIDADE DE SÃO PAULO

SEGUNDO SEMESTRE LETIVO DE 2023

PRIMEIRA PROVA

2a Questão) (**2 Pontos**) Determine o número real de ciclos de clock para executar uma vez o trecho abaixo (do blez $t1,end até uma instrução que salte para trás ou até a última instrução do trecho ser executada, o que acontecer primeiro). Suponha máxima capacidade de resolução de conflitos de dados (isto inclui uma unidade de adiantamento capaz de adiantar dados da saída do terceiro estágio para a entrada do terceiro estágio, da saída do quarto estágio para a entrada do terceiro, e da saída do quarto estágio para a entrada do quarto estágio). Assuma também que está disponível uma estrutura mestre-escravo para acesso ao banco de registradores e um preditor de saltos de 2 bits que inicialmente prevê salto não-realizado. O PC é escrito no quarto ciclo de relógio de cada instrução. Detalhe a execução no diagrama pipeline abaixo, indique todos os adiantamentos de dados que ocorrerem (se ocorrerem) e mostre as bolhas nas posições adequadas, caso estas existam.

BLEZ -- Branches if the register is less than or equal to zero

Os valores iniciais dos registradores pertinentes são: $t0=0x10010000, $t1=0x44 (=68 base 10), $t2=0, $t4=0, $t3=0x100100AA

INSTRUÇÃO		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
loop:	blez $t1,end																			
	sltiu $t2,$t1,65																			
	bne $t2,$zero,nxtch																			
	sltiu $t2,$t1,91																			
	beq $t2,$zero,nxtch																			
	addiu $t4,$t4,1																			
nxtch:	addiu $t0,$t0,4																			
	lw $t1,0($t0)																			
	j loop																			
end:	sw $t4,0($t3)																			

INSTRUÇÃO		1	2	3	4	5	6	7	8	9	10	11	12	13	14
loop:	blez $t1,end	B	D	E	-	-							B		
	sltiu $t2,$t1,65		B	D	E $t2=0	-	W								
	bne $t2,$zero,nxtch			B	D	E	--	---							
	sltiu $t2,$t1,91				B	D	E $t2=1	--	W						
	beq $t2,$zero,nxtch					B	D	E	--	--					
	addiu $t4,$t4,1						B	D	E	--	W				
nxtch:	addiu $t0,$t0,4							B	D	E	----	W			
	lw $t1,0($t0)								B	D	EX	M	W		
	j loop									B	D	EX*	----	---	
end:	sw $t4,0($t3)										B	D	F	F	F

11 instruções 1,1
Forward 0.3
Estagio não usado 0.3
Jump 0.3

Convenções:
X → bolha
* → estágio em que um salto é executado (carga no PC)
- →estágio não usado

→ → adiantamento ou leitura após escrita no mesmo ciclo
Estágios do pipeline: B (Busca), D (Decodificação), E (Execução), M (Memória), W (Write-back)

3a Questão) (**2,0 Pontos**) Considere a seguinte sequência de instruções, executadas em um datapath com pipeline de 5 estágios:

```
add $5, $2, $1
lw   $3, 4($5)
lw   $2, 0($2)
or   $3, $5, $3
addi  $4,$3,4
```

a) Na ausência de forwarding ou de detecção de conflito, insira nops para garantir que o código rode corretamente

```
add $5, $2, $1
nops
nops
lw   $3, 4($5)
lw   $2, 0($2)
nops
or   $3, $5, $3
nops
nops
addi  $4,$3,4
```

cada nops 0,15

b) Repita o item anterior usando nops somente quando um conflito não puder ser evitado pela mudança ou rearranjo dessas instruções. Assuma que o registrador $7 é usado para armazenar valores temporários no seu código modificado

O único que dá para mudar é lw $2, 0($2), mas sem ganho

```
add $5, $2, $1
lw   $2, 0($2)
nops
lw   $3, 4($5)
nops
nops
or   $3, $5, $3
nops
nops
addi  $4,$3,4
```

cada nops 0,15

c) Se o processador tiver forwarding, mas esquecermos de implementar a unidade de detecção de conflitos, o que ocorrerá quando este código for executado?

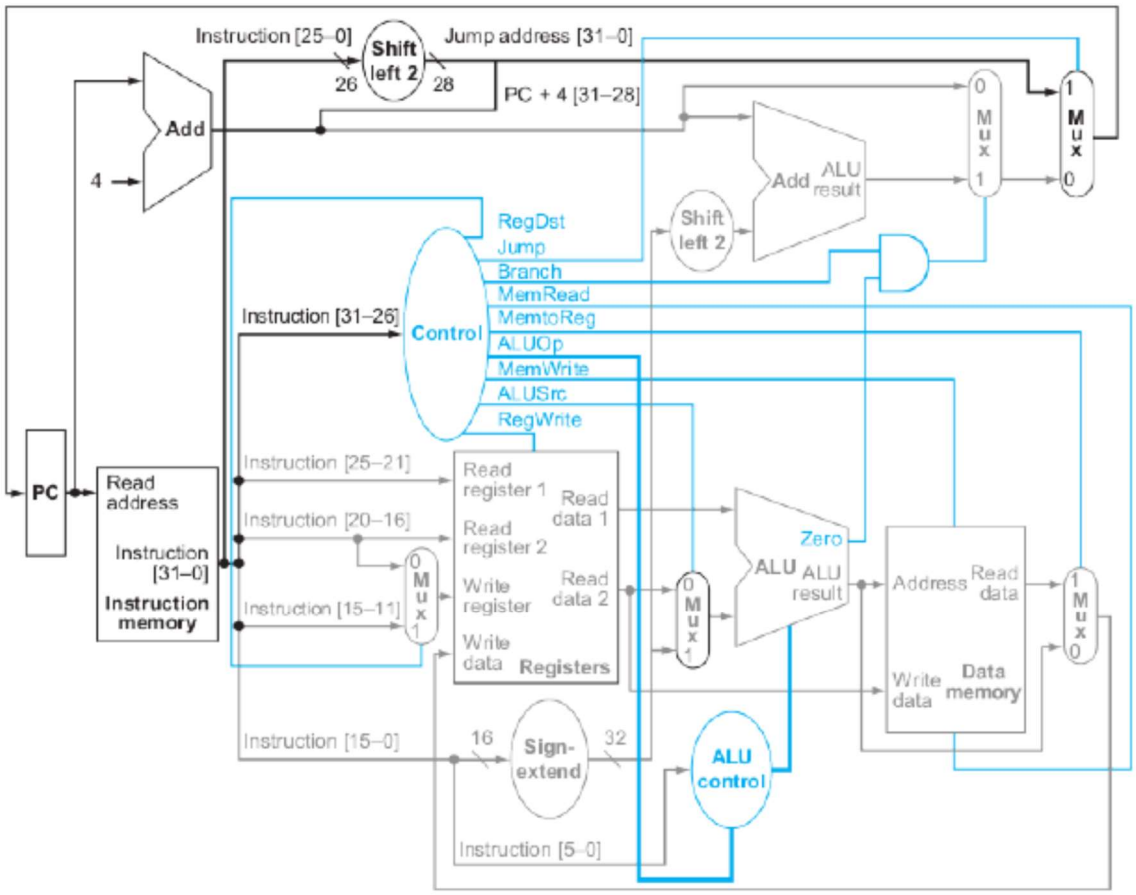
Considerando forwarding, não é preciso bolha

```
add $5, $2, $1
lw   $3, 4($5)
```

lw $2, 0($2)
or $3, $5, $3
addi $4,$3,4
O único problema que poderia ocorrer é se alguém precisasse de $3 logo após (2). Não é o caso, pois o lw em (3) deu o tempo necessário para o forwarding funcionar entre (2) e (4). Então não há problema

0,3
Jusficativa 0,2

4a Questão) (**2,0 Pontos**) Considere o datapath abaixo:



Agora suponha que, em um datapath de ciclo único, a seguinte instrução é trazida da memória: 10101100011000100000000000010100 (trata-se de um \sw rt, desl(rs)"). Assuma que a memória de dados está preenchida com zeros, e que os registradores do processador têm os seguintes valores no início do ciclo no qual a instrução acima é buscada:

r0	r1	r2	r3	r4	r5	r6	r8	r12	r31
0	–1	2	–3	–4	10	6	8	2	–16

a) 0,3 Quais as saídas da extensão de sinal e da unidade de deslocamento de jumps (\Shift left 2" na figura) para essa instrução?

101011	00011	00010	0000000000010100
opcode	rs	rt	deslocamento

A extensão de sinal usa os bits [15-0] (0000.0000.0001.0100), estendendo o bit de sinal até termos 32b: 0000.0000.0000.0000.0000.0000.0001.0100
A saída da unidade de deslocamento de jumps (shift left 2) é tão somente o deslocamento à esquerda dos bits [25-0]: 0001.1000.1000.0000.0000.0101.0000

b) 0,3 Quais os valores da entrada da unidade de controle da ALU para essa instrução?
A unidade de controle da ALU recebe os bits [5-0], ou seja, 01.0100, além dos 2 bits de ALUOp (00, conforme tabela na aula 06)

c) 0,3 Qual o novo endereço do PC após essa instrução ser executada? Mostre (na figura) o caminho que leva à definição desse valor.
Será PC+4. O caminho é PC → PC+4 → Mux do branch → Mux do jump → PC
correção
Caminho 0,2
Pc + 4 0,1
Ou
Caminho 0,3 (somente)

d) 0,3 Para cada MUX, mostre os valores de suas saídas durante a execução desta instrução com estes valores de registradores.
0,05 Mux dos Registradores se RegDst=0 00010 (2)
RegDst=1 00000
0,1 Mux da ALU (bits [0-15] com extensão de sinal, corresponde ao valor 20)

0,05 Mux da Memória X (Não há como dizer, e não importa)

0,05 Mux do Branch PC+4

0,05 Mux do Jump PC+4

e) 0,4 Para a ALU e as duas unidades de soma, quais são os valores de suas entradas de dados?
0,1 Somador do PC: PC e 4

0, 15 Somador do Branch: (PC+4) + 80 (bits [15-0] deslocados em 2 esquerda)

0, 15 ALU: -3 ($rs=-3) e 20 (saída do Mux da ALU)

f) 0,4 Quais os valores de todas as entradas para a unidade de registradores?
RegWrite: 0
Read Register 1: [25-21] = 00011 3
Read Register 2: [20-16] = 00010 2
Write Register: 2 (se RegDst=1)
ou 0 (se RegDST=0)
Write Data: Não há como dizer a partir dos dados
Cada 0,08

5a Questão) (1ponto) Limitando nossa atenção a 8 instruções: lw, sw, add, sub, and, or, slt e beq, qual o tempo médio entre 2 instruções em uma implementação de ciclo único, em que todas as instruções ocupam um único ciclo de clock, e uma implementação de pipeline, em que cada estágio ocupa um ciclo de clock? O tempo de cada instrução, tanto total quanto a cada estágio,

Instruction class	Instruction fetch	Register read	ALU operation	Data access	Register write	Total time
Load word (lw)	200 ps	100 ps	200 ps	200 ps	100 ps	800 ps
Store word (sw)	200 ps	100 ps	200 ps	200 ps		700 ps
R-format (add, sub, AND, OR, slt)	200 ps	100 ps	200 ps		100 ps	600 ps
Branch (beq)	200 ps	100 ps	200 ps			500 ps

Na implementação de ciclo único, o datapath precisa acomodar a instrução mais lenta, e uma só entra no datapath após a outra sair. Então o tempo entre uma instrução e outra _e de 800ps.
No caso da pipeline, e supondo a ausência de problemas, assim que uma instrução deixa a pipeline a seguinte já está entrando no último estágio. Então a diferença de tempo entre 2 instruções é o tempo gasto em cada estágio. Como todo estágio, em nosso modelo, leva o mesmo tempo, eles precisam ser projetados para acomodar o maior estágio. Assim, o tempo entre uma instrução e outra é de 200ps.

Ciclo único –
Tempo entre cada instrução 800 ps
Tempo de cada instrução 800ps
Tempo de cada estagio variável

Pipeline
Tempo entre cada instrução 200 ps
Tempo de cada instrução 200ps
Tempo de cada estágio igual a 200ps

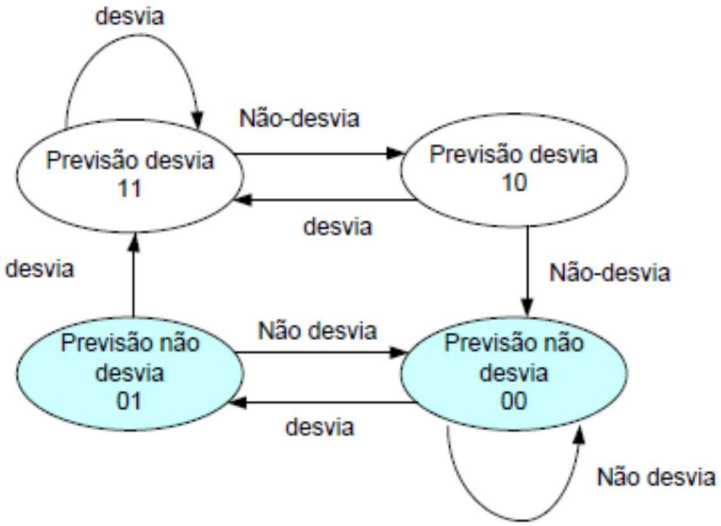
6a Questão) (1 ponto) Assuma que o seguinte código é executado sobre um processador pipeline com 5 estágio, com adiantamento e um preditor de desvio dois bits

```
Label1:  lw $1, 40 ($6)
         beq $2, $3, Label2  //tomado
         add $1, $6, $4
Label2:  beq  $1, $2, Label1// não tomado
         sw  $2, 20 ($4)
         and $1, $1, $4
```

a)Desenhe o diagrama de execução para este código, assumindo que não há slots de atraso e o que desvio executa no estágio EX

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
sw $2,20($4)	BI	DI	EX	Mem	WAR										
beq $2, $3, Label2		BI	DI	EX	MEM	WAR									
Add $1, $6, $4			BI	DI	F	F	F								
beq $1,$2,Label1				BI	F	F	F	F							
beq $1,$2,Label1					BI	DI	EX	MEM	WAR						
sw $2,20 ($4)						BI	DI	EX	MEM	WAR					
and $1, $1, $4							BI	DI	EX	MEM	WAR				

b)Qual é o speed-up alcançado ao mover a execução de desvio para o estagio DI. Assuma que a comparação no estágio DI não afeta o tempo de ciclo de clock.



Speed up -0.2

Diagrama – 0.3

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
sw $2,20($4)	BI	DI	EX	Mem	WAR										
beq $2, $3, Label2		BI	DI	EX	MEM	WAR									
Add $1, $6, $4			BI	X	X	X	X								
beq $1,$2,Label1				BI	DI	EX	MEM	WAR							
sw $2,20 ($4)					BI	DI	EX	MEM	WAR						
and $1, $1, $4						BI	DI	EX	MEM	WAR					