

Iniciado em Saturday, 27 May 2023, 11:35

Estado Finalizada

Concluída em Saturday, 27 May 2023, 11:49

Tempo 14 minutos 22 segundos
empregado

Avaliar 10,00 de um máximo de 10,00(100%)

Questão 1

Correto

Atingiu 0,50 de 0,50

Complete o código MIPS abaixo que deve adicionar o número em \$t1 com o valor decimal 15 salvando o resultado em \$t2

`li $t1, 12`

- _____
- a. `addi $t2, $t1, 0xF`
 - b. `addu $t2, $t1, $t2`
 - c. `addi $t2, $t1, F`
 - d. `add $t2, $t1, $t2`
 - e. `addui $t2, $t0, 0xF`



Sua resposta está correta.

A resposta correta é: `addi $t2, $t1, 0xF`

Correto

Notas para este envio: 0,50/0,50.

Questão 2

Correto

Atingiu 0,50 de 0,50

Considere o trecho de código C abaixo:

```
int t1 = 10;  
int t2 = 3;  
int t3 = t1 + 2 * t2;
```

Considere a seguinte relação entre variáveis em C e registradores do MIPS:

Variável em C	Registrador MIPS
t1	\$t1
t2	\$t2
t3	\$t3

Complete o trecho de código MIPS abaixo que representa o trecho de código C acima.

```
li $t1, 10  
li $t2, 3
```

- a. `mul $t2, 2`
`add $t3, $t2, $t1`
- b. `mult $t2, 2, $t2`
`add $t3, $t2, $t1`
- c. `sll $t2, $t2, 1`
`add $t3, $t2, $t1` ✓
- d. `mul $t2, 2`
`add $t1, $t2, $t3`
- e. `mult $t2, $t2, 2`
`add $t3, $t2, $t1`
- f. `mult $t2, 2`
`add $t3, $t2, $t1`

Sua resposta está correta.

A resposta correta é: `sll $t2, $t2, 1`

`add $t3, $t2, $t1`

Correto

Notas para este envio: 0,50/0,50.

Questão **3**

Completo

Atingiu 2,00 de 2,00

Traduza todo o programa em C abaixo (que contém uma função recursiva) para Assemble do MIPS

```
int sum(int x) {
    if (x == 0)
        return 0;
    return x + sum(x-1);
}

int main() {
    int a;
    scanf("%i", &a);
    a = sum(a);
    printf("%i\n", a);
    return 0;
}
```

Submeta um único arquivo **.s** ou **.asm (em texto puro)** contendo o seu programa.

 [exercicioprova1.s](#)

Comentário:

Questão 4

Correto

Atingiu 3,00 de 3,00

Considere o código assembly do MIPS32 abaixo e o respectivo endereço em memória de cada instrução.

```

.text
.globl main

main:
0x00    addiu $a0, $zero, 4
0x04    jal soma

        # imprimir resultado
0x08    addu $a0, $zero, $v0
0x0C    addiu $v0, $zero, 1
0x10    syscall

end:
0x14    addiu $v0, $zero, 10
0x18    syscall

soma:
0x1C    bne $a0, $zero, soma_corpo
0x20    addiu $v0, $zero, 0
0x24    jr $ra

soma_corpo:
        # salvar contexto
0x28    addi $sp, $sp, -8
0x2C    sw $s0, 0($sp)
0x30    sw $ra, 4($sp)

        # código principal
0x34    addu $s0, $zero, $a0
0x38    addi $a0, $a0, -1
0x3C    jal soma
0x40    add $v0, $v0, $s0

        # restaurar contexto
0x44    lw $s0, 0($sp)
0x48    lw $ra, 4($sp)
0x4C    addi $sp, $sp, 8

        # retorna
0x50    jr $ra

```

No início do programa os valores dos seguintes registradores são: **\$sp = 0xfc** e **pc = 0x00**. Os valores dos demais registradores de interesse e palavras na pilha são **0x00**.

Considere o exato momento em que a instrução em **pc = 0x20** é executada.

A) Indique os valores em hexadecimal na pilha. Cada linha da tabela indica o endereço inicial de uma palavra de 4 bytes.

Endereço	Valor
0xFC	0x00 ✓
0xF8	0x08 ✓
0xF4	0x00 ✓
0xF0	0x40 ✓
0xEC	0x04 ✓
0xE8	0x40 ✓
0xE4	0x03 ✓

0xE0	0x40	✓
0xDC	0x02	✓
0xD8	0x00	✓
0xD4	0x00	✓

B) Indique os valores dos registradores

Registrador	Valor	
\$ra	0x40	✓
\$sp	0xDC	✓
\$a0	0x00	✓
\$s0	0x01	✓

Correto

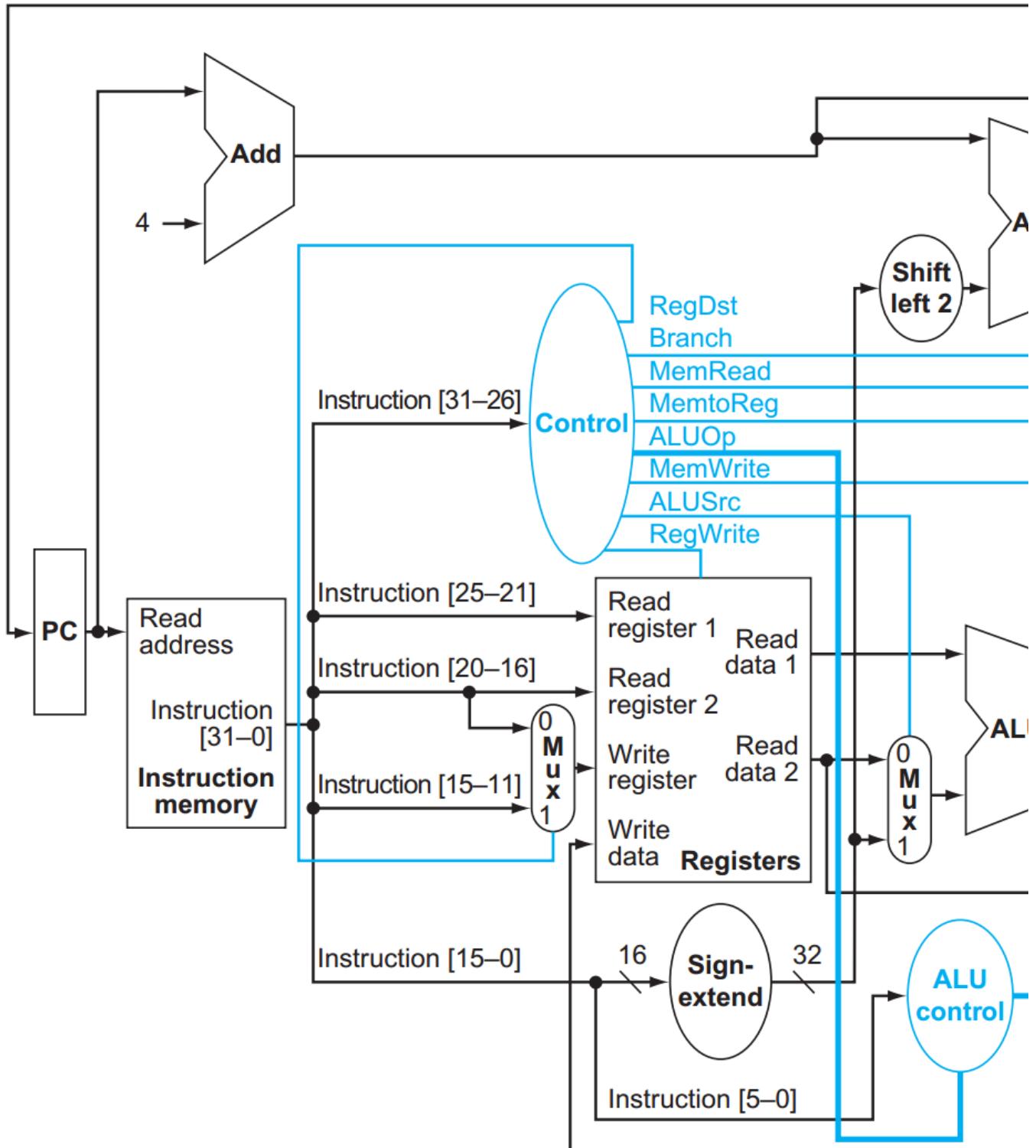
Notas para este envio: 3,00/3,00.

Questão 5

Correto

Atingiu 2,00 de 2,00

Considerando o seguinte datapath para execução de instruções com ciclo único:



E os seguintes códigos de controle da ALU:

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

Associe os valores dos sinais de controle emitidos pela unidade de controle para a instrução. O valor x indica 'tanto faz' ('doesn't care'):

add \$t1, \$t2, \$t3

Sinal de Controle	Valor
RegDst	1 ✓
Branch	0 ✓
MemRead	0 ✓
MemtoReg	0 ✓
ALUOp	10 ✓
MemWrite	0 ✓
ALUSrc	0 ✓
RegWrite	1 ✓
Saída controlador ALU	0010 ✓

Correto

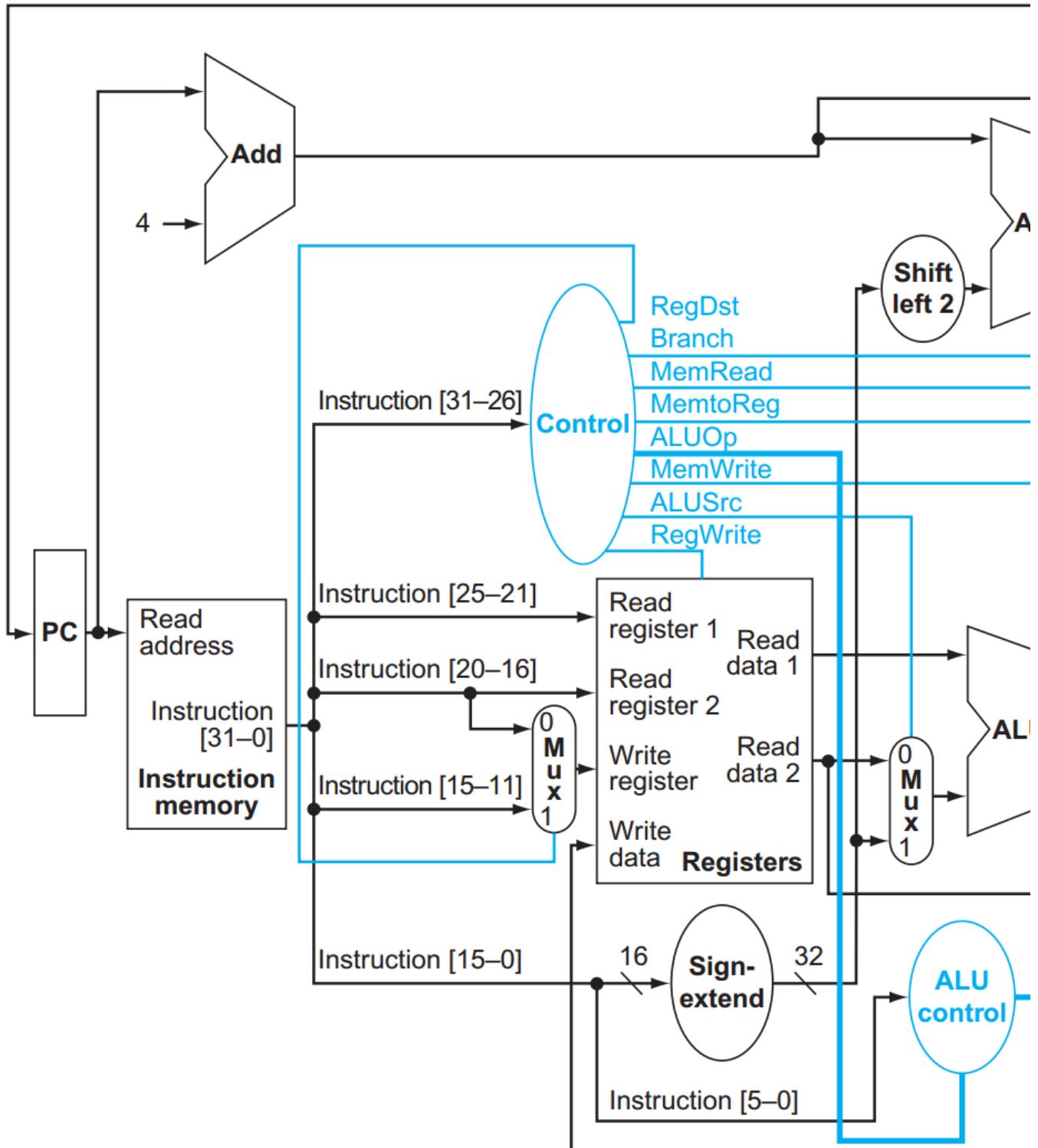
Notas para este envio: 2,00/2,00.

Questão 6

Correto

Atingiu 2,00 de 2,00

Considerando o seguinte datapath para execução de instruções com ciclo único:



E os seguintes códigos de controle da ALU:

ALU control lines	Function
0000	AND
0001	OR
0010	add
0110	subtract
0111	set on less than
1100	NOR

Associe os valores dos sinais de controle emitidos pela unidade de controle para a instrução. O valor x indica 'tanto faz' ('doesn't care'):

addi \$t1,\$t2,16

Sinal de Controle	Valor
RegDst	0 ✓
Branch	0 ✓
MemRead	0 ✓
MemtoReg	0 ✓
ALUOp	00 ✓
MemWrite	0 ✓
ALUSrc	1 ✓
RegWrite	1 ✓
Saída controlador ALU	0010 ✓

Correto

Notas para este envio: 2,00/2,00.

◀ Handout-6.3: Sinais de Controle

Seguir para...

Handout-6.4: Pipelining ►