

2022/1 CCT ARQUITETURA E ORGANIZAÇÃO DE COMPUTADORES

Painel

/

Meus cursos

/

Departamento de Ciência da Computação

/

Bacharelado em Ciência da Computação

/

2022/1 CC

/

2022/1_CCT_CCI192-04U_AOC0004

/

Semana 07

/

Revisão p/ P1

Iniciado em

Wednesday, 11 May 2022, 10:17

Estado

Finalizada

Concluída em

Friday, 13 May 2022, 18:50

Tempo empregado

2 dias 8 horas

Avaliar

8,99 de um máximo de 10,00(90%)

Questão 1

Correto

Atingiu 0,50 de 0,50

Marcar questão

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. `addui $t2, $t0, 0xF`

☒

b. `addi $t2, $t1, 0xF`

☐

c. `addu $t2, $t1, $t2`

☐

d. `add $t2, $t1, $t2`

☐

e. `addi $t2, $t1, F`

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

Marcar questão

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. `mult $t2, 2`

Navegação do questionário

1

2

3

4

5

6

Mostrar uma página por vez

Terminar revisão

```
add $t3, $t2, $t1

b. mult $t2, 2, $t2
add $t3, $t2, $t1

c. sll $t2, $t2, 1
add $t3, $t2, $t1

d. mult $t2, $t2, 2
add $t3, $t2, $t1

e. mul $t2, 2
add $t1, $t2, $t3

f. mul $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 1,99 de 2,00

🚩 Marcar questão

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.

 questao_3.s

Comentário:

Você está fazendo `$s0 = x - 1` na Linha 31, isto está complicando sua lógica.

Observe que você restaura o valor de `$s0` (valor da função chamadora) na Linha 37 e usa este valor na Linha 40.

Veja o código da questão 04 para uma alternativa mais elegante de solução.

Questão 4

Correto

Atingiu 2,40 de 3,00

🚩 Marcar questão

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: $\$sp = 0xf0$ e $pc = 0x00$. Os valores dos demais registradores de interesse e palavras na pilha são 0x00.

Considere o exato momento em que $pc = 0x20$.

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

Endereço	Valor
0xF0	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 ✓

\$ap	0xDC	✓
\$a0	0x00	✓
\$s0	0x01	✓

Correto

Notas para o envio: 3,00/3,00. De acordo com as tentativas anteriores 2,40/3,00.

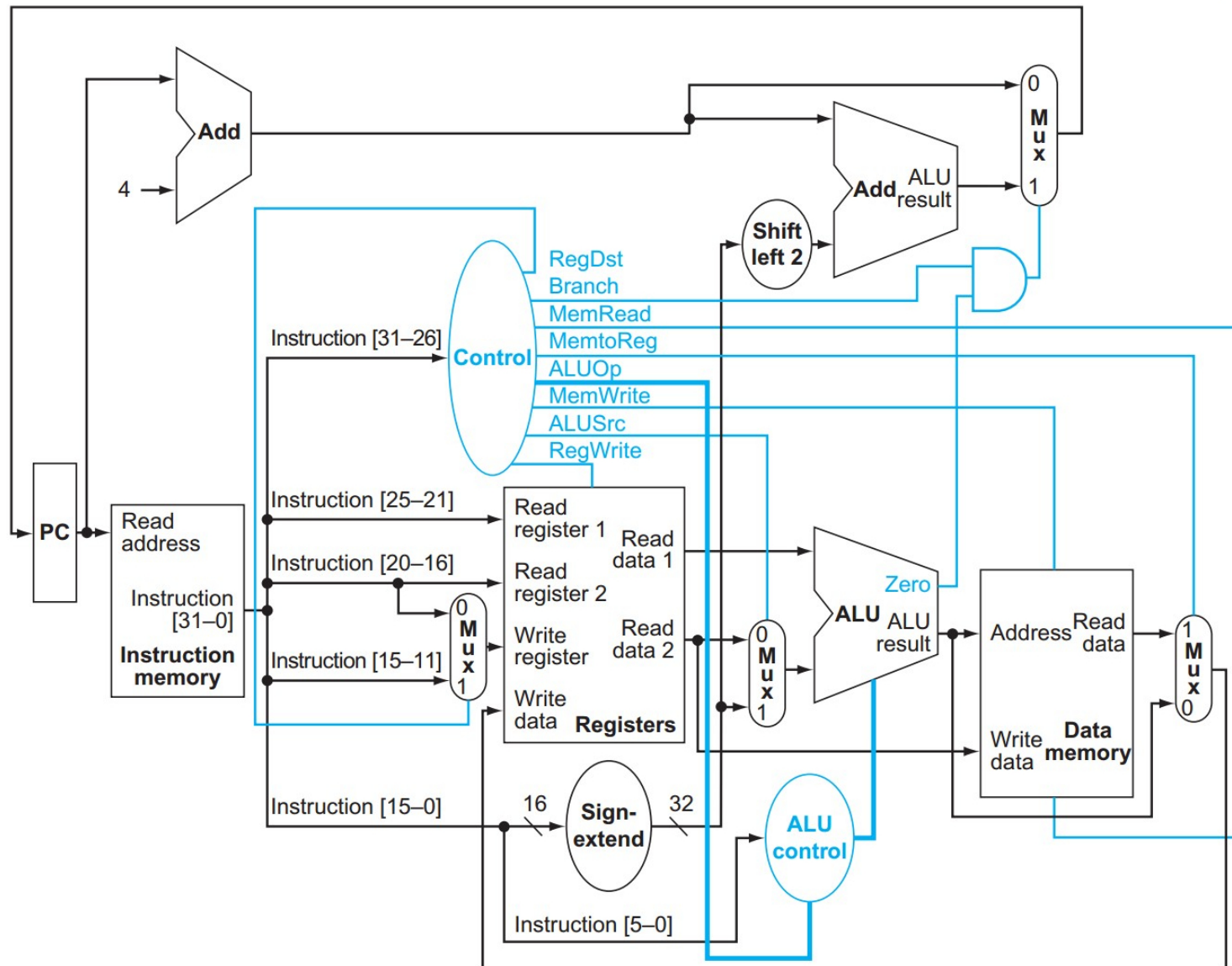
Questão 5

Correto

Atingiu 2,00 de 2,00

⚑ Marcar questão

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.

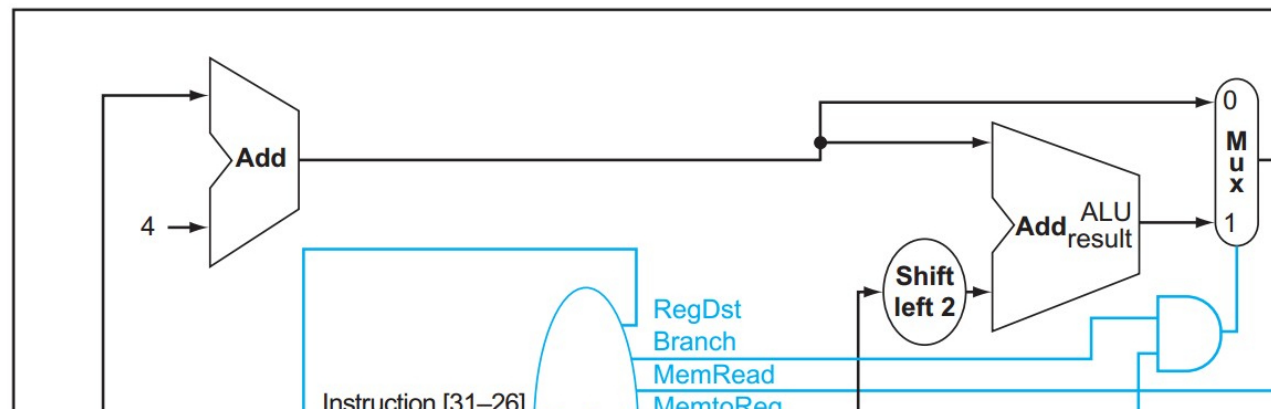
Questão 6

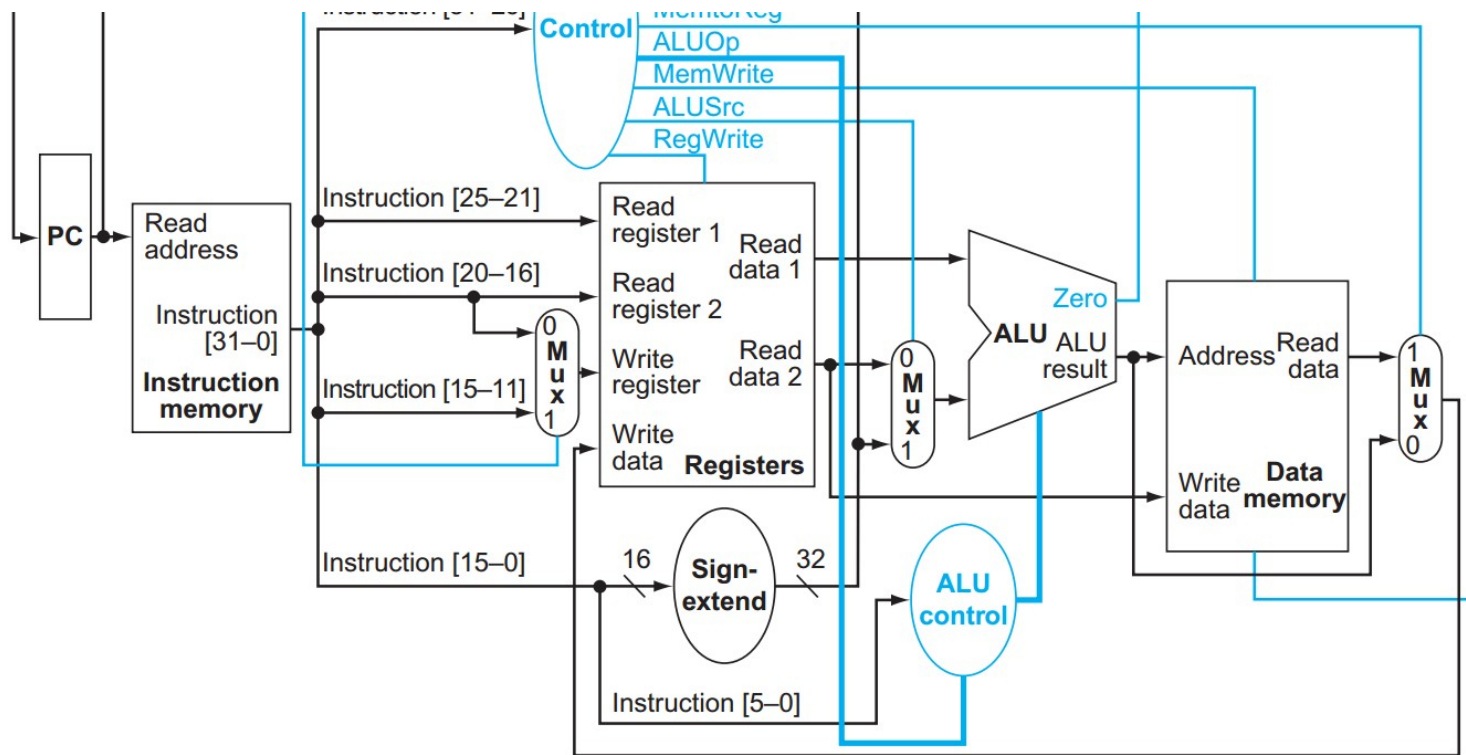
Correto

Atingiu 1,60 de 2,00

🚩 Marcar
questão

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 ✓

MemWrite	0	✓
ALUSrc	1	✓
RegWrite	1	✓
Saída controlador ALU	0010	✓

Correto

Notas para o envio: 2,00/2,00. De acordo com as tentativas anteriores **1,60/2,00**.

Terminar revisão

◀ Handout-6.3: Sinais de Controle

Seguir para...

2022-1-Especificação Recuperação P1 - Em dupla entrega na aula de 2ª, 20/06 ▶

Você acessou como VICTOR EDUARDO REQUIA (Sair)

[2022/1 CCT CCI192-04U AOC0004](#)

[Resumo de retenção de dados](#)

[Obter o aplicativo para dispositivos móveis](#)