Universidade Estadual do Rio Grande do Sul

Engenharia de Computação
Disciplina de Organização de Computadores
Professora: Débora Matos

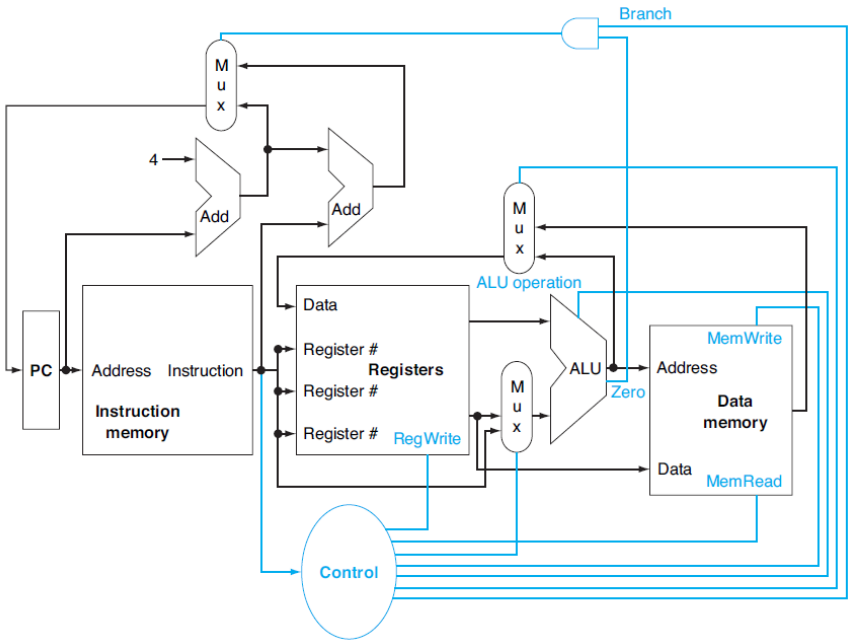
Lista de Exercícios – Conjunto de Instruções, caminho de dados e pipeline

Questão 1 (questão de prova): Um grupo de engenheiros verificou que seria interessante criar uma instrução que realize a comparação diretamente com um valor imediato (immed) passado na instrução. Quando o desvio ocorre (se os valores comparados forem iguais), o novo valor do PC recebe o valor armazenado em um registrador, conforme exemplificado a seguir.

a) (0,6) Sendo assim, acrescente as alterações necessárias no HW da organização do MIPS abaixo para que essa nova instrução possa ser executada. Explique o caminho de dados de como funcionará a nova instrução, com as modificações realizadas e com os componentes já existentes.

b) (0,5) Como essa instrução afeta os conflitos? São introduzidos novos hazards? Comente:

beqi $s0, $s1, immed # *se* $s0 == immed *faça* PC = $s1



Questão 2: Imagine que para o HW anterior, os engenheiros queiram introduzir mais algumas novas instruções. Nesse caso, serão inseridas 4 novas instruções: **ROL, ROR, SHL, SHR**, que correspondem a rotação para a esquerda, rotação para a direita, deslocamento para a esquerda, deslocamento para a direita.

a) Imagine que as instruções serão executadas a partir de um operando em um registrador e o resultado é salvo em um registrador de destino, exemplo: ROL $t0, $t1. Que modificações precisam ser feitas na organização para atender a tais instruções?

b) Agora imagine que o operando é informado de forma imediata nas instruções. Que outras modificações precisam ser adicionadas na arquitetura?

Questão 3 (questão de prova): A instrução slt $s0, $s1, $s2 tem como função setar o registrador $s0 em 1 quando $s1< $s2. No entanto, esta instrução normalmente é utilizada juntamente com uma instrução de branch (beq ou bne). Seria interessante que se pudesse utilizar apenas 1 instrução que já fizeste a comparação e executasse o desvio. Sendo assim, acrescente as alterações necessárias no Hw da organização do MIPS abaixo para que essa nova instrução possa ser executada (blt – *branch if less than*)

blt $s0, $s1, offset # if $s0 < $s1 faça PC = PC + 4 + (offset x 4), else PC=PC + 4

Questão 4 (questão de prova): Uma equipe de engenheiros verificou que para um conjunto de programas, sempre que uma subtração é realizada, o valor do cálculo precisa ser salvo em uma posição da memória. Para isso, a fim de reduzir o número de instruções do programa, uma instrução que realize esta tarefa precisa ser criada. Acrescente o hardware necessário e comente porque ele foi adicionado para atender a nova instrução, conforme consta abaixo:

sub_sw $s0, $s1, $s2 # mem[$s0] = $s1 - $s2

Nessa instrução, o valor do registrador de destino usado na instrução sub agora conterá o endereço de memória onde o valor da subtração deve ser salvo.

Questão 5: Foi verificado, que a fim de reduzir a memória de dados, seria interessante realizar operações com operandos de apenas 16 bits. Neste caso, todos os valores lidos da memória de dados seriam de 16 bits. Considere as seguintes situações:

a) Que modificações são necessárias no HW para atender a essa modificação no tamanho da palavra de dados, considerando que todas as instruções agora serão para atender 16 bits?
b) Que modificações são necessárias no HW se tivermos uma arquitetura que permita tanto operações com operandos de 16 bits, como os já utilizados, 32 bits?

Questão 6: Questões sobre *hazards* em uma organização do MIPS com pipeline:

Na sequência de instruções do MIPS a seguir, indique as situações onde ocorrem conflitos (*exemplo de indicação: linha 8 e linha 9 - conflito de dados do registrador $s3).* Quais conflitos teriam como ser solucionados com a técnica de *forwarding*?

1. add $s1, $s2, $s4
2. sll $s4, $s4, 3
3. or $t0, $s1, $s4
4. lw $s3, 0($s0)
5. sw $s5, 4($s3)
6. and $s2, $s5, $s2
7. slt $s5, $s5, $s3

Questão 7: Questões sobre *hazards* em uma organização do MIPS com pipeline:

Na sequência de instruções do MIPS a seguir,

a) Indique as situações onde ocorrem conflitos (*exemplo de indicação: linha 8 e linha 9 - conflito de dados do registrador $s3).*
b) Quais conflitos teriam como ser solucionados com a técnica de *forwarding*?
c) Reorganize a sequência de instruções utilizando outros registradores (e/ou instruções) de forma a evitar os conflitos. Reduza ao máximo o número de registradores e instruções utilizados.

1. or $s1, $s2, $s4
2. slt $s4, $s4, $s1
3. sw $s3, 0($s4)
4. lw $s4, 4($s3)
5. and $s4, $s5, $s2
6. beq $s5, $s4, $s3