UNIVERSIDADE FEDERAL DA BAHIA (UFBA)
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA (IME)
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO (DCC)

DIMITRI SANTANA MARINHO
RAFAEL FARIAS

TRABALHO PRÁTICO II

Trabalho entregue ao Prof. Dr. Marcos Ennes Barreto como requisito avaliativo parcial para composição de nota na disciplina MATA48 - Arquitetura de computadores

SALVADOR
2018

Observação. O enunciado da questão a. foi alterado, trocando o simulador WebSimple pelo simulador PS - CAS MIPS. Realizou-se a mudança do simulador de pipelining devido aos bugs presentes no simulador WebSimple que dificultavam a execução do trecho de código. Conforme o artigo fornecido no Moodle, o simulador PS - CAS MIPS é uma otimização do WebSimple, que identifica algumas dependências que o WebSimple deixa de detectar.

a. Execute o programa abaixo no simulador PS - CAS MIPS. Utilize a configuração padrão para tempo monociclo (50 ns) e para pipeline (10 ns). Não utilize forwarding (deixe desmarcadas todas as opções MEM → EX,WB → EX, WB → MEM. Indique: a) as dependências de dados existentes; b) a quantidade de ciclos ociosos no pipeline; c) o desempenho (speed up) obtido.

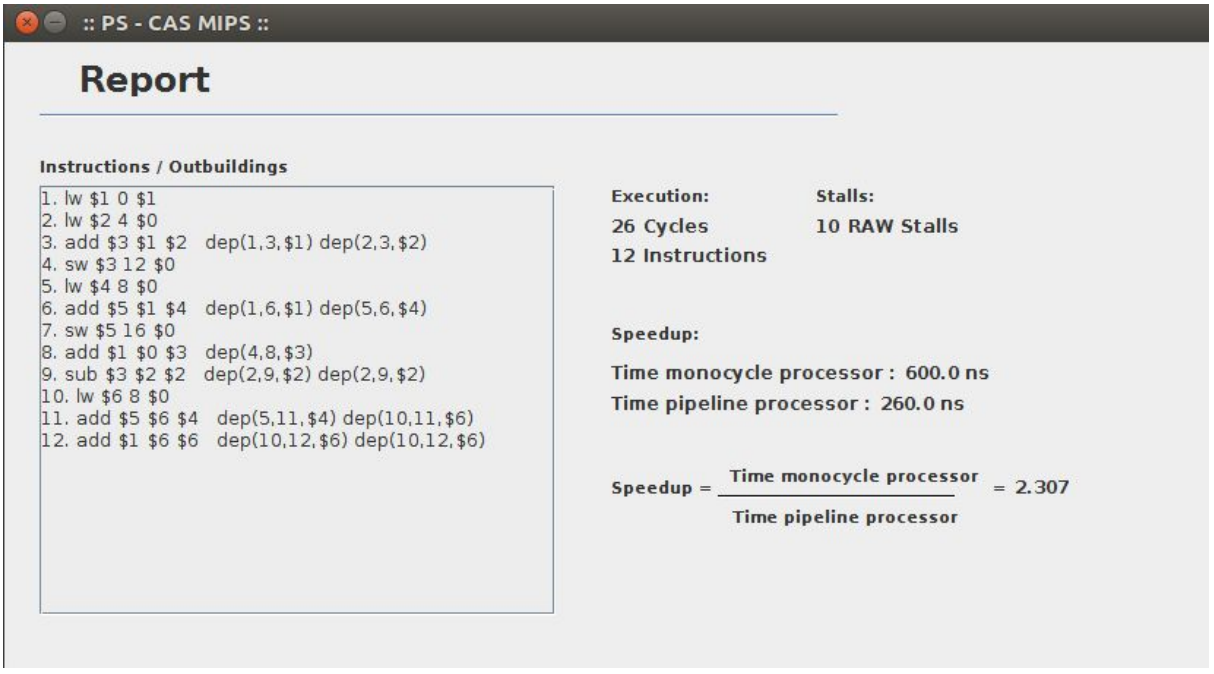
```
lw $1, 0($1)

lw $2, 4($0)

add $3, $1, $2

sw $3, 12($0)

lw $4, 8($0)

add $5, $1, $4

sw $5,16($0)

add $1, $0, $3

sub $3, $2 $2

lw $6, 8($0)

add $5, $6, $4

add $1, $6, $6
```

Dependências existentes

9 dependências (RAW) entre as operações:

dep(1,3,$1)
dep(2,3,$2)
dep(1,6,$1)
dep(5,6,$4)
dep(4,8,$3)
dep(2,9,$2)
dep(5,11,$4)
dep(10,11,$6)
dep(10,12,$6)

Quantidade de ciclos ociosos: 10 (10 RAW Stalls)
Speedup = 2.307



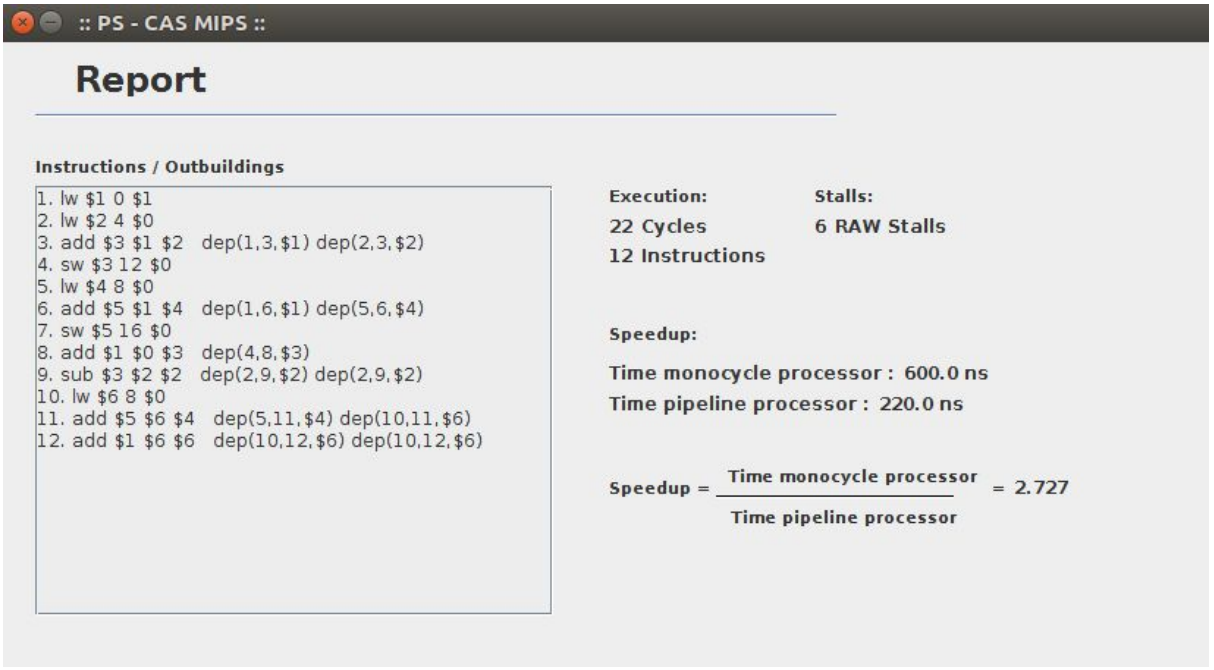
b. Modifique o exercício **a.** para considerar forwarding entre os estágios MEM → EX. Indique novamente: a) as dependências de dados existentes; b) a quantidade de ciclos ociosos no pipeline; c) o desempenho (speed up) obtido.

Dependências existentes

9 dependências (RAW) entre as operações:

dep(1,3,$1)
dep(2,3,$2)
dep(1,6,$1)
dep(5,6,$4)
dep(4,8,$3)
dep(2,9,$2)
dep(5,11,$4)
dep(10,11,$6)
dep(10,12,$6)

Quantidade de ciclos ociosos: 6 (6 RAW Stalls)
Speedup = 2.727



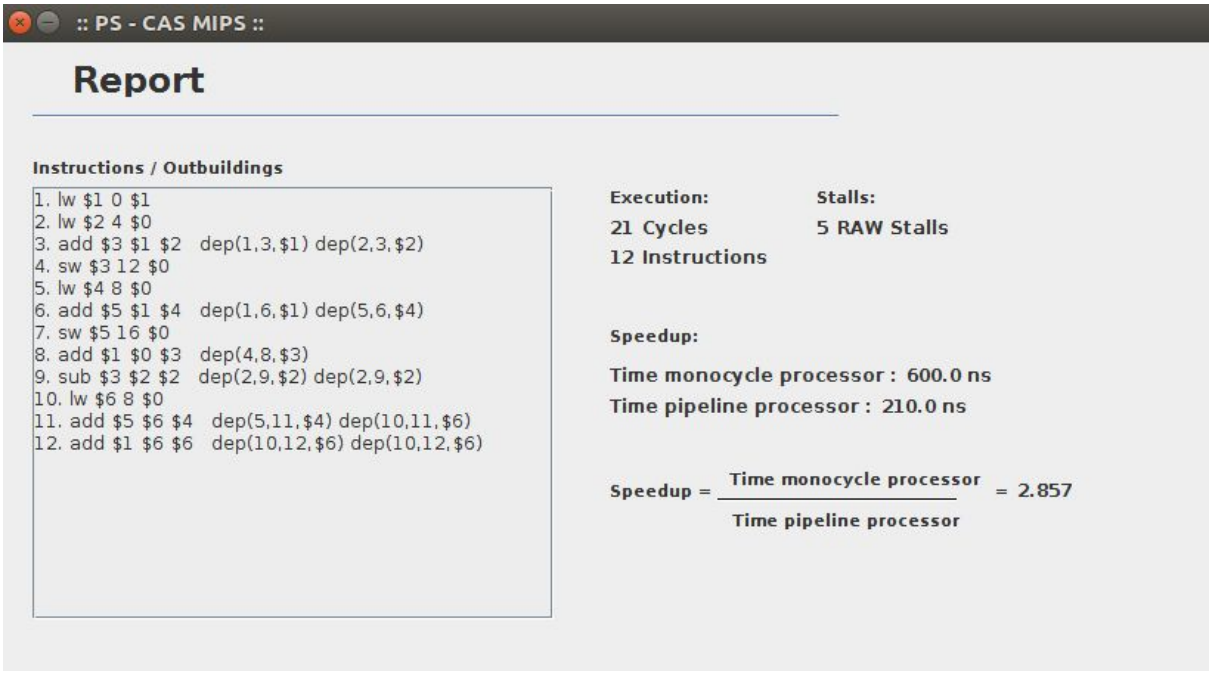
c. Modifique o exercício **a.** para considerar forwarding entre os estágios WB → EX. Indique novamente: a) as dependências de dados existentes; b) a quantidade de ciclos ociosos no pipeline; c) o desempenho (speed up) obtido.

Dependências existentes

9 dependências (RAW) entre as operações:

dep(1,3,$1)
dep(2,3,$2)
dep(1,6,$1)
dep(5,6,$4)
dep(4,8,$3)
dep(2,9,$2)
dep(5,11,$4)
dep(10,11,$6)
dep(10,12,$6)

Quantidade de ciclios ociosos: 5 (5 RAW Stalls)
Speedup = 2.857



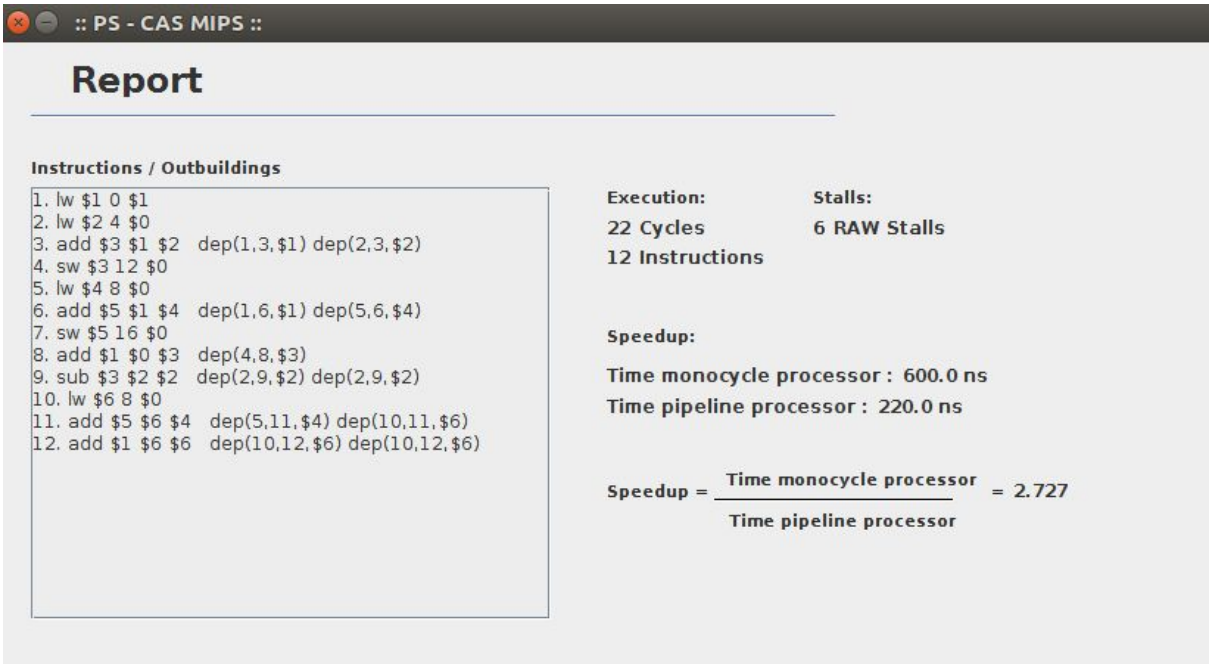
d. Modifique o exercício **a.** para considerar forwarding entre os estágios WB → MEM. Indique novamente: a) as dependências de dados existentes; b) a quantidade de ciclos ociosos no pipeline; c) o desempenho (speed up) obtido.

Dependências existentes

9 dependências (RAW) entre as operações:

dep(1,3,$1)
dep(2,3,$2)
dep(1,6,$1)
dep(5,6,$4)
dep(4,8,$3)
dep(2,9,$2)
dep(5,11,$4)
dep(10,11,$6)
dep(10,12,$6)

Quantidade de ciclos ociosos: 6 (6 RAW Stalls)
Speedup = 2.727



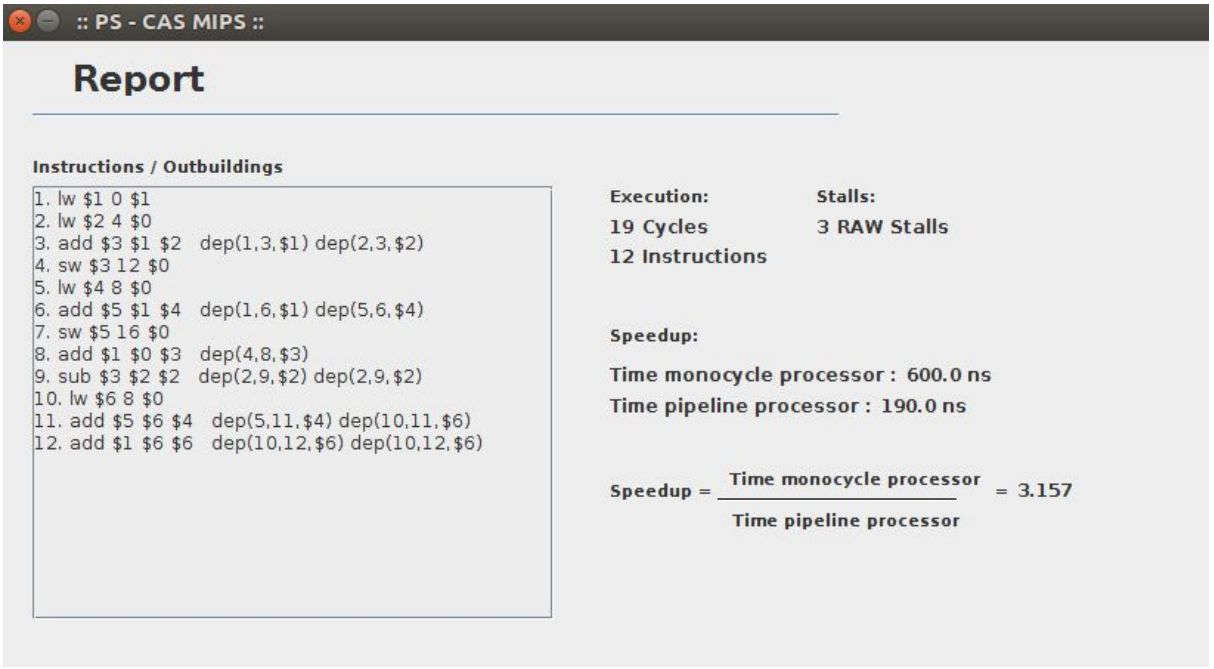
e. Modifique o exercício **a.** para considerar forwarding entre todos os estágios (MEM → EX, WB → EX, WB → MEM). Indique novamente: a) as dependências de dados existentes; b) a quantidade de ciclos ociosos no pipeline; c) o desempenho (speed up) obtido.

Dependências existentes

9 dependências (RAW) entre as operações:

dep(1,3,$1)
dep(2,3,$2)
dep(1,6,$1)
dep(5,6,$4)
dep(4,8,$3)
dep(2,9,$2)
dep(5,11,$4)
dep(10,11,$6)
dep(10,12,$6)

Quantidade de ciclios ociosos: 3 (3 RAW Stalls)
Speedup = 3.157



f. Comente sobre os resultados obtidos em cada configuração. Na sua opinião, faz sentido (seria viável tecnicamente) utilizar forwarding em todos os estágios (configuração da questão e)? Em caso negativo, qual das configurações de forwarding (questões b a d), na sua opinião, gera o melhor resultado? Por quê?

Como o trecho de código executado é o mesmo em todas as questões, alterando somente o forwarding, então é compreensível que as dependências geradas sejam as mesmas, o que muda em cada caso é a quantidade de ciclos ociosos, já que a ativação do forwarding em alguns casos ocasiona a diminuição dos ciclos ociosos, pois evita a parada do pipeline utilizando buffers internos em vez de esperar que o elemento de dado chegue nos registradores visíveis ao programador ou na memória.

Em termos de desempenho, a melhoria da execução sem forwardings para a execução:

1. com forwarding MEM → EX é de, aproximadamente, 18%;
2. com forwarding WB → EX é de, aproximadamente, 24%;
3. com forwarding WB → MEM é de, aproximadamente, 18%;
4. com todos os forwardings (MEM → EX, WB → EX, WB → MEM) é de, aproximadamente, 37 %.

A partir da análise percentual de melhora na velocidade da execução das instruções, percebe-se que a implantação de todos forwardings é viável no quesito desempenho, já que aumenta-o em 37%, no entanto, para considerar se é viável tecnicamente, deve-se levar considerar as intenções do projeto do hardware e avaliar o custo-benefício, tendo noção do custo financeiro dessa implantação.

No entanto, percebe-se que o forwarding da letra **b.** e **d.** fornecem o mesmo resultado em relação ao desempenho. Simulou-se o trecho de código excluindo o forwarding MEM → EX (**b.**) e, em seguida, sem o forwarding MEM → EX (***d.***) e ***o resultado de speedup deu igual ao da implementação de todos os forwardings***. Portanto, para esse trecho de código, é mais viável implementar dois forwardings (com o WB → EX

incluso), do que implementar todos os forwardings, já que o desempenho é o mesmo e o custo é inferior.

Com relação à simulação no PS - CAS MIPS do trecho de código executado, as configurações de forwarding das questões **b.** e **d.** geram a mesma melhora em termos de desempenho (**speedup**), mas, em termos gerais, a configuração da letra **b.** (forwarding MEM → EX) parece ser melhor que a configuração da letra **d.** (forwarding WB→ EX), pois no ciclo de instruções o *"Memory access"* (MEM) vem antes do *"Register write back"* (WB).

g. Escolha um dos trechos de código abaixo e o traduza para o MIPS 32 bits. Teste o programa traduzido com o simulador MARS (ou outro simulador MIPS) para garantir que o código está funcional. Após, insira os comandos MIPS (segmento .data) no simulador WebSimple e repita os passos executados nos exercícios a a e (habilitando as diferentes opções de forwarding). Relate, para cada opção de forwarding simulada, i) as dependências de dados encontradas, ii) a quantidade de ciclos ociosos no pipeline e iii) o desempenho obtido com a habilitação de cada opção de forwarding.