UNIVALI – Arquitetura e Organização de Computadores II Prof. Thiago Felski Pereira
Avaliação 01 – Conflitos de dados no Pipeline

Instruções:

1. Esta atividade pode ser realizada individualmente ou em **até TRÊS** alunos (no máximo).

2. A defesa da implementação será realizada dia **04/10/2023**, e a data final da postagem do relatório até às **18h59** do dia **04/10/2023**. Defesas/entregas feitas na aula seguinte terão 70% da nota.

3. A implementação poderá ser realizada em **qualquer** linguagem de programação, sendo que o código deverá ser entregue (também é válido o link para o trabalho no GitHub) e apresentado.

4. Todos os integrantes do grupo deverão estar presentes para apresentação e os códigos deverão conter identificação dos autores. Se algum integrante faltar, este terá que apresentar sozinho na semana seguinte e estará sujeito a *instrução 2*.

5. Se forem identificados trabalhos com grau de similaridade que caracterize cópia ou adaptação (autorizadas ou não pelos seus autores originais), a nota dos grupos será a nota de um trabalho dividida pelo número de grupos que entregou esses trabalhos similares.

ATIVIDADE

Premissas: Considere que o conflito estrutural está resolvido e podem ignorar conflitos de controle nessa etapa da solução, considere que o desvio nunca acontece, mas não tente reordenar instruções de desvio como branchs e jumps.

Elaborar um programa que, mediante a entrada de um arquivo de memória de instrução (ROM) em linguagem de máquina (hexadecimal ou binário), o programa deverá identificar os conflitos de dados e implementar 4 soluções distintas para o problema.

1. Considerar que não há nenhuma solução em hardware para conflitos e incluir NOPs, quando necessário, para evitar o conflito de dados.
2. Considerar que foi implementada a técnica de forwarding e inserir NOPs, quando necessário, para evitar conflito de dados.
3. Considerar que não há nenhuma solução em hardware para conflitos e quando possível reordenar as instruções e quando não for possível inserir NOPs, para evitar conflito de dados.
    a. Por exemplo, é possível que o programa não tenha nenhuma instrução, a diante no código, para ser reordenada.
4. Considerar que foi implementada a técnica de forwarding e quando possível reordenar as instruções e quando não for possível inserir NOPs, para evitar conflito de dados.

Elaborar uma análise de desempenho que avalie o sobrecusto em instruções da solução, o tempo de execução e o número de ciclos de programa da solução em Pipeline selecionada considerando um tempo de clock fornecido pelo usuário.

VALIDAÇÃO

Para a validação, será utilizado o código disponibilizado para testes além de um código extra a ser disponibilizado apenas no dia da apresentação. Será necessário ler os arquivos hexadecimais/binários manualmente usando o RARS (*dump file*).

FLUXO DE OPERAÇÃO

Passos da operação:

1. Insira o tempo de clock do Pipeline;
2. Selecione a técnica para resolução de conflitos;
3. Escolha o arquivo com o programa em binário ou hexadecimal;
4. Execute a técnica e calcule o desempenho;
5. Gere um arquivo novo com a aplicação da técnica; e
6. Exiba o resultado.

Alternativamente é possível seguir os seguintes passos da operação:

1. Insira o tempo de clock do Pipeline;
2. Escolha o arquivo com o programa em binário ou hexadecimal;
3. Execute todas as técnicas e calcule o desempenho de cada uma;
4. Gere arquivos para cada solução; e
5. Exiba todos os resultados.

Obs: Um bug comum na reordenação é tentar buscar uma instrução após a última instrução.

CÁLCULO DA NOTA

Percentual da nota para cada parte do trabalho:

- 25% Solução 1
    - Inserir uma instrução que não é um NOP válido é considerado um erro, por exemplo, 00000000000000000000000000000000, não é um NOP válido.
    - Inserir mais ou menos NOP do que o necessário é considerado um erro;
- 25% Solução 2
- 15% Solução 3
- 15% Solução 4
- 20% Desempenho
    - Essa nota considerará o cálculo de desempenho de todas as técnicas que o grupo conseguiu implementar, sendo assim, o grupo não perde nota se não calcular o desempenho das técnicas que não conseguiu fazer.