

1 Disciplina CI212-A - Organização e Arquitetura de Computadores

Projeto - Estendendo o Modelo VHDL do processador MIPS segmentado

O objetivo deste projeto é obter uma compreensão mais profunda da implementação de processador segmentado. Dado um modelo básico do projeto do processador segmentado discutido em classe e encontrado no livro texto, você simulará este modelo e irá estendê-lo. Você então testará suas extensões simulando com GHDL.

O trabalho é individual. Você também pode discutir sua solução/abordagem com os outros alunos da disciplina. Contudo, qualquer cópia do código de outro grupo será considerada uma violação de honestidade e acarretará em perda de nota para os grupos envolvidos.

Para fazer esse projeto, siga os passos abaixo:

1. Baixe os arquivos de código para o processador segmentado do site "http://www.nr2.ufpr.br/~marco/mips_pipeline". A maioria dos arquivos nesse diretório são os mesmos como no projeto do processador monociclo. Os novos arquivos incluem o arquivo de controle "control_pipeline.vhd" e o de nível superior de "mips_pipeline.vhd". Arquivos para multiplexadores de 3 e 4 entradas estão incluídos no caso de você precisar deles para suas extensões. O arquivo "rom32.vhd" foi modificado para conter instruções "no-op" entre as instruções para eliminar os hazards (riscos). Construa e simule o circuito no GHDL. Usando a janela de projeto, rastreie os sinais internos, incluindo o PC, opcode, estados da unidade de controle, etc, para verificar se o circuito funciona corretamente.
2. Estender o projeto do processador para realizar adiantamento de dados, conforme descrito na Seção 6.4, inserindo bolhas (detecção de riscos) conforme descrito na Seção 6.5. A projeto modificado será algo parecido com Fig. 6.36 na p. 416, exceto que deve incluir ligações para o campo imediato (para loads, stores e branches), bem como o hardware de desvio que está na implementação original. Você vai precisar adicionar uma unidade de transmissão, uma Unidade de Detecção de riscos, multiplexadores e sinais extra em alguns registradores de pipeline. Simule o circuito com as instruções que mostram que cada conexão de adiantamento funciona como descrito no livro. Também

mostram que riscos do load e seu uso inserem uma bolha no pipeline (20 pontos).

3. Estender o projeto para executar a instrução de desvio na fase de decodificação, como mostrado na Fig. 6.38 na p. 420. Você precisará mover o somador de destino de desvio para a fase de decodificação e adicionar um novo comparador que verifica a igualdade entre as saídas do banco de registradores. Além disso, você vai ter necessidade de fazer modificações no registradores de pipeline para remover sinais que não são mais necessários. Mais importante, você terá que alterar o projeto da unidade de controle para executar o desvio na fase de identificação. Simule o seu circuito para mostrar que ele funciona corretamente para uma sequência de instruções incluindo desvios e outras instruções (20 pontos).
4. Modifique o projeto do processador para implementar a instrução "swi". A instrução deve utilizar o opcode "011011" e funcionará da seguinte maneira: swi Rt immmed onde $M[\text{immed}] := Rt$ (20 pontos)
5. Implemente separadamente um processador superescalar-2, ou seja, um processador que faz 2 instruções por vez, estas instruções serão uma instrução alu/branch e uma load/store, presente na apresentação 19 até o slide 12. (40 pontos)
6. Faça uma listagem dos arquivos que você alterou. Destaque as mudanças que você fez e inclua os comentários que explicam como eles funcionam (dentro do código).
7. Indique a sua estimativa da quantidade de tempo gasto neste projeto, o que você considera como os pontos mais positivos e menos positivos sobre este projeto, e qualquer sugestão que possa fazer para melhorar este projeto no futuro.