

Trabalho Prático 2 – MIPS

Alexandre Alphonsos Pereira
Jerônimo Nunes Rocha
Gabriel Moraes de Souza Diniz

Introdução:

O objetivo desse trabalho é fazer com que todos os módulos que fazem parte de um processador como Fetch, Decode, etc sejam unidos e que seu comportamento final possa ser testado e validado. Criaremos um módulo Mips que une todos os outros módulos. Em seguida passaremos um conjunto de instruções através de um testbench e verificaremos o resultado obtido ao final de cada instrução para afirmar que o processador está funcionando corretamente.

Módulo implementado:

Mips.v

Perguntas:

1. O clock do MIPS deve ser mais lento que o clock da RAM porque estamos passando a instrução dividida em duas partes, 16 + 16 bits. Para que a instrução seja carregada corretamente o clock da ram deve ser duas vezes mais rápida, assim garantimos que a instrução estará pronta para os outros módulos.
2. A função do módulo Control é pegar a instrução e gerar os sinais de controle responsáveis pelo funcionamento adequado dos outros módulos
3. Multiply (mult \$1,\$2) Type:R
 Divide (div \$1,\$2) Type:R
 Move from high (mfhi \$1) Type:R
4. A necessidade do extensor de sinal do bloco Decode se deve ao fato de que certas instruções, como a de adicionar um imediato, não possui 32 bits. Nesse caso ela será estendida para que tudo funcione corretamente.
5. Para torná-lo um processador de 64 bits seria necessário modificar o memController, pois procuraríamos instruções e dados maiores, a ALU pois agora ela deve suportar os 64 bits, o banco de registradores, o decode, o fetch e todos os outros pois estes devem ser de 64 bits. A RAM não precisaria ser alterada.

Dificuldades encontradas:

Tivemos dificuldade com a sintaxe do verilog e com a falta de qualidade do log de erros, que muitas vezes nos respondia com mensagens como: I give up! Os módulos foram conectados utilizando referenciamento hierárquico e por algum motivo que não conseguimos identificar, não funcionou corretamente. Refizemos todo o trabalho criando fios. Conectar o módulo Writeback à RAM também gerou alguns problemas que foram resolvidos no final.

Conclusão:

O trabalho se mostrou bastante desafiador e com o seu desenvolvimento percebemos a complexidade existente na integração dos módulos e como os testes devem ser rigorosos para ter certeza de que tudo está se comportando como deveria. Ademais, o Trabalho Prático foi importante para frizar os conhecimentos obtidos durante a disciplina e aprofundar o funcionamento do MIPS em Pipeline.