

EE4363 / CSci 4203 – Computer Architecture

Machine Problem 2

Description

- The file `mipspipe.v` contains an incomplete behavioral Verilog description of a MIPS processor. The incomplete hardware description can execute LW, SW, and NOP instructions and does not provide support for hazard detection.
- Study the Verilog code to determine how it works. Next, modify the hardware description to add support for ADD, OR, and AND instructions.
- Compile the Verilog code for your MIPS processor. Suggestions for free Verilog compilation / simulation tools can be found in the file `Verilog_tools.rtf` on Canvas.
- Use the testbench `test_mipspipe.v` to simulate operation of the MIPS pipeline Verilog module. Test your implementation using the following code.

```
ADD    $5, $2, $1
LW     $3, 4($5)
LW     $2, 0($2)
OR     $3, $5, $3
AND    $4, $2, $3
SW     $4, 0($5)
```

Initialize the value of DMemory at 4(\$5) to “0xFFFFFFFF” and the value at 0(\$2) to “0xFFFFFFFF0”. You may need to insert NOPs into the code (IMemory) to avoid data hazards. Test the code both with and without NOPs inserted.

- Summarize your observations in a report.

Report Format

- Provide a brief summary of your code modifications.
- Provide a brief summary of your experiments.
- Compare the execution outcomes with and without insertion of NOPs to avoid data hazards.
- Attach your Verilog code for the MIPS processor and testbench with your submission.
- Submit a tarball or zip file of your files through Canvas.

Reference Materials (available on Canvas)

- Appendix-C-Computer-Organization.pdf – The Basics of Hardware Design
- Appendix-D-Computer-Organization.pdf – Mapping control to Hardware
- Section-4.12-Computer-Organization.pdf – An introduction to digital design using a hardware language to describe and model a pipeline and more pipelining illustrations