CS450/650 – W2023 – Verilog Assignment 2

Mark toward final grade: 10% + 1.5% bonus
Due: February 27th at 11:59pm
Deliverables: mips.sv
Submission: upload to Learn | Submit | Dropbox | Verilog Assignment 2
Late policy: please refer to course outline

The objective of this assignment is to augment a MIPS processor so that it can correctly run a benchmark program, SimpleAdd.c

The stub MIPS module provided (mips_stub_new.sv) can only execute one instruction, *addiu*. So you will need to first determine which other instructions are necessary to run SimpleAdd (by referring to SimpleAdd.dmp), then find out exactly what these instructions do from the MIPS manual, and finally implement the logic for these instructions. A provided testbench (p2_tb.sv) enables you to test your system.

Note: SimpleAdd uses the *jr* instruction; for this assignment you're not expected to implement the branch-delay-slot mentioned in the manual.

Relevant files:

MIPS processor	
mips_stub_new.sv -- Please rename to mips.sv	Stub MIPS module for you to modify
Testbench	
p2_tb.sv	Instantiates memory and MIPS modules, loads the benchmark program, and prints the execution outcome of each instruction.
Other modules/files needed to compile & simulate	
memory.sv	Memory module
regfile.sv	Register file module
params.sv	Memory and processor initialization parameters
Makefile	
Benchmark program	
SimpleAdd.c	Original source code
SimpleAdd.dmp	Listing generated by MIPS compiler
SimpleAdd.s	Assembly code generated by MIPS compiler
SimpleAdd.sh	Compile script
SimpleAdd.x -- Please copy to the same folder as .sv files	Object code in hexadecimal format
References & hints	
block_diagrams_corrected.pdf	Diagram of the microarchitecture
Annotated_SimpleAdd_dmp.pdf	Hints
mipsQuickRef.pdf	Summary of instruction set and registers (identifies which are pseudo instructions)
mipsPRM.pdf	MIPS Manual detailing instructions and encodings

15% Bonus

Implement additional instructions to run the SimpleIf.c benchmark.

(Please take advantage of this if you value bonus points, as there aren't bonuses in every assignment)

The toolchain

Icarus Verilog is an open-source compiler (iverilog) and simulator (vvp). It can be used with GTKWave to simulate and view waveforms. Both are available in the linux.student.cs environment, but we recommend you install them on your computer.

Icarus Verilog is available from sourceforge and from icarus.com. You need at least version 10.0. The installation instructions for Linux, FreeBSD, MacOS, and Windows are available on the wiki. Please see additional installation tips at the end of this document.

To compile: `iverilog source1.v source2.v ...` To simulate: `vvp a.out` (The man pages and the wiki has more details if you want to know more).

GtkWave is a waveform viewer and is a useful companion tool when using Icarus Verilog. The source code and binaries are available for download on sourceforge and its website.

To generate the wavefile, insert this line into the testbench file: `initial $dumpvars(0, module_tb);` where module_tb is your testbench module name, then compile and simulate as per instructions above. To view the waveform: `gtkwave dump.vcd` The wiki GtkWave page has more details.

adder_tb.v is an example that contains both a testbench and an adder module, for you to verify that your toolchain is working.

How to compile & simulate the MIPS project

Call *make* on the provided Makefile (on Linux), or directly call the tools:

```
iverilog -g2005-sv -s tb_SimpleAdd -o tb_SimpleAdd.vvp p2_tb.sv memory.sv
regfile.sv mips.sv

vvp -n tb_SimpleAdd.vvp

gtkwave dump.vcd
```

Appendix – Additional installation tips

On Windows 10

To install Icarus Verilog

- Icarus Verilog installer available from http://bleyer.org/icarus/
- most recent version: iverilog-v11-20210204-x64_setup.exe [44.1MB]
- install options: yes to minGW, no to gtkwave

To install GTKWave

- GTKWave application available from https://sourceforge.net/projects/gtkwave/files/gtkwave-3.3.100-bin-win64/
- unzip and copy gtkwave64 folder to iverilog installation location (optional)
- update "Path" in the Environment variables: need to replace the entry iverilog added (e.g. C:\iverilog\gtkwave\bin) with the new one (e.g. C:\iverilog\gtkwave64\bin)

To run the adder_tb example

- from Command Prompt
  ```
  iverilog adder_tb.v
  vvp a.out
  ```
- from Windows PowerShell
  ```
  gtkwave dump.vcd
  ```

Notes from students:

When I run ./configure to install GtkWave, this message appears:

> checking for Tcl configuration... configure: error: Can't find Tcl configuration definitions. Use --with-tcl to specify a directory containing tclConfig.sh

I installed gtkwave with apt-get and it worked!

For Mac users, I recommend using homebrew to install the two:

```
brew install gtkwave
```

```
brew install icarus-verilog
```

I also needed to follow the instructions here to get the gtkwave command working properly