Laboratory 9

Single Cycle Data Path

Due Date: beginning of class Aug. 3

Objectives:

- Complete the design of a single cycle MIPS32 data path.

Assignment

The ZIP file contains several files, explained briefly below:

- Controlunit.v – the single cycle control unit which is just a combinational logic network. It has the structure setup with several sample instructions implemented, and additional instructions may be easily added with new case statements, cutting-and-pasting the template, and setting the new control signals. Note that every instruction contains all output signals, and it explicitly sets unneeded signals to don't cares (X').
- Mips.v – this is the actual implementation of the data path. For the most part, is it an architectural file like implementing Boolean equations in lab 4: modules are instantiated and connected with wires. Note that the names of the modules in this file match those used in the data path figure, and wires were logically named. There are a few instances in which a simple adder or shifter (such as the PC+4 adder) were implemented with a Verilog statement instead of a module (see the branchMUX on line 88).
- Mips_tb.v – This file initializes the internal modules with storage (register file, instmem, datamem) and reads the sample program from a text file into instruction memory. Once the data path has been initialized, it only needs an initial reset signal and then a repeating clock. The reset signal and termination time is at the end of the file. The larger *always* loop generates a clock signal and is followed by a series of display statements that may be commented or uncommented to print out the desired set of registers.
- Prog1.asm and prog1.txt – The assembly file shows what would have been typed into Mars for the small test program. The text file is the machine code that was copied from Mars once the program was built so that it is in a format that can be read by Verilog.
- There standard set of data path modules that have been tested in the previous two labs are also present.

This data path may be simulated by building mips_tb.v.

Procedure

The data path will be completed so that the program table_copy.asm can be executed and validated. Perform the following steps.

- Build table_copy.asm and write a text file that implements the program (use the same format as prog1.txt).
- Not all the assembly commands needed for table_copy.asm have been implemented. Add the additional commands needed to controlunit.v.
- Alter mips_tb.v so that the table_copy instructions are loaded. Clearly indicate where changes were made using comments.
- Alter mips_tb.v so that the initial values in data memory from table_copy.asm are loaded. This must be done with an external text file and a loop just like the instruction memory used. Note that when data memory addresses are skipped, you may load 0xFF's into the unused memory addresses. Clearly indicate where changes were made using comments.
- Add statements in mips_tb.v to display memory addresses from 0x10010000 to 0x1001005F. The memory must be displayed in 3 rows of 32 hexadecimal bytes with rows and columns labeled. It must closely resemble the MARS data memory window. (Note that the $write operation is similar to $display, but it doesn't append a carriage return.)
- Adjust the display statements to show only the $tX registers.
- Modify the run length of the simulation to allow the entire program to be executed.
- All files created or modified MUST have appropriate header comments.

When your data path correctly executes the program, run the simulation and capture the output to a text file. The ">" operator in the command line routes output to a file instead of the screen. For example:

```
>> iverilog -o test mips_tb.v

>> vvp test > output.txt
```

Deliverables

Submit all deliverables according to the posted lab submission guidelines.

- 10 points: Compliance with lab submission guidelines
- 20 points: Text files used for initializing the instructions and data memory for table_copy.asm.
- 60 points: altered Verilog files, such as controlunit.v and mips_tb.v, graded according to the program rubric
- 10 points: Text file of data path output executing table_copy.asm

DO NOT upload the supplied module Verilog files if they have not been modified.