Introduction to CMOS VLSI Design (E158)

Harris

Lab 5: Microprocessor Assembly

In this final lab, you will assemble and simulate your entire MIPS microprocessor!

1. Top-Level Schematic

Copy your lab3_xx library to lab5_xx for this lab. Also open your controller_xx library so you can reference the alucontrol and controller cells.

Create a new schematic in the lab5_xx library named mips. Note that the Edit Facet dialog has a drop-down list of open libraries to help you navigate among the libraries. You may also find the File • Change Current Library command to be useful to switch between default libraries.

In this top-level mips cell, place your datapath, alucontrol, and controller icons. Connect the cells together. When you are finished, your system should have the following inputs and outputs:

Inputs	Outputs
ph1	adr[7:0]
ph2	writedata[7:0]
reset	memread
memdata[7:0]	memwrite

Table 1: MIPS Processor Inputs & Outputs

Be sure to label the internal wires such as those connecting the controller to the datapath. Double-click on each internal wire and give it a name; do not add exports. Names will help you when debugging your system. Also give each icon a name.

2. Simulation

To demonstrate basic functionality of your microprocessor, you will simulate the processor running a very simple program:

```
addi  $1, $0, 43   # Register 1 <- 43     instr: 20010043
addi  $2, $0, 1A   # Register 2 <- 1A     instr: 2002001A
or    $3, $1, $2   # Register 3 <- 5B     instr: 00221825
sb    $3, 38($2)   # Mem[52] <- 5B        instr: A0430038
```

Recall that you have developed a multi-cycle processor, so each instruction will require several steps. The table attached at the end of the lab lists the inputs and expected outputs on each step. The steps for the first instruction have been completed for you. Please fill out the remainder of the table.

*** If the simulator were working perfectly, you'd use the following commands to simulate your schematic. Unfortunately, two separate bugs in Electric prevent either the schematics or the layout from properly simulating. We're working on these and hope to have them fixed as you proceed to your projects. ☹

Simulate your schematics. Use the c command to set ph1 and ph2 as clocks *** explain. Check that the outputs match expectations. If they do not, track down the problem; it is likely an error in your `addi` code or your FSM synthesis. Remember that you can use the Tools • Simulation • Down Hierarchy command to descend into various cells and check that the inputs and outputs match expectation. This is easiest if you give the cells names in your schematics. Moreover, you can use Tools • Simulation • Save Vectors to Disk to save a set of input stimulus you have created.

Note that register 0 should be hardwired to contain all 0's. However, that was not done in the layout. Fortunately, the simulator defaults to all 0's in registers at the beginning of simulation, so your code should simulate correctly.

3. Top-Level Layout

Create a new layout in the lab5_xx library named mips. In this top-level cell, place your datapath, alucontrol, and controller. Wire together the modules. Don't forget to connect power and ground with fat wires and arrays of vias to handle the higher levels of current that may flow! You will avoid creating a rats nest of wiring if you systematically reserve metal2 for vertical lines and metal1 for horizontal lines.

Export all the inputs and outputs. Label the internal signals.

4. Layout Verification

Verify that your layout passes DRC, ERC, and network compares with the schematic. Fix any problems that might arise.

5. Summary

If you have successfully completed the lab, congratulations! You have designed, assembled, and tested your own microprocessor! You now are familiar with the major aspects of custom CMOS VLSI design:

- Leaf cell design
- Datapath design and assembly
- Hardware specification with Verilog
- Standard cell synthesis and place & route
- Top-level system assembly
- Switch-level simulation and logic debug
- Design Rule Checking
- Electrical Rule Checking
- Network compare and debug of mismatched networks

You will put these skills to use as you proceed with your final project!

6. What to Turn In

Please provide a hard copy of each of the following items:

1. Please indicate how many hours you spent on this lab. This will not affect your grade, but will be helpful for calibrating the workload for the future.
2. What was unclear in this lab writeup? How would you change it to run more smoothly next time?
3. A printout of your mips schematic.
4. Your table with inputs and expected outputs when simulating the processor.
5. Simulation waveforms demonstrating correct operation of the mips processor. *** not applicable in 2001
6. A color printout of your mips layout.
7. What is the verification status of your mips layout? Does it pass DRC? ERC? NCC?

Extra Credit

As you are probably aware by now, Electric has plenty of bugs and idiosyncrasies. A major goal of this class is to improve the stability and ease-of-use of Electric. Please email your bug reports directly to Prof. Harris in the format described in Lab Manual 1.

Cycle	Reset	MemData	Adr	WriteData	MemRead	MemWrite	ALUSrcA	ALUSrcB	ALUControl	State
0	1	x	x	x	x	x	x	x	x	x
1	0	20	00	x	1	0	0	01	010	0
2	0	01	01	x	1	0	0	01	010	1
3	0	00	02	x	1	0	0	01	010	2
4	0	43	03	x	1	0	0	01	010	3
5	0	x	x	x	0	0	0	11	010	4
6	0	x	x	x	0	0	1	10	010	13
7	0	x	x	x	0	0	0	00	010	14
8	0	20	04	x	1	0	0	01	010	0
9	0									
10	0									
11	0									
12	0									
13	0									
14	0									
15	0									
16	0									
17	0									
18	0									
19	0									
20	0									
21	0									
22	0									
23	0									
24	0									
25	0									
26	0									
27	0									
28	0									