

CMPE 200 – Assignment 5

Haonan Wang

Reference: Donald Hung

Computer Engineering Department, San Jose State University

Processor Design (1) – Design Code Review and Functional Verification

Purposes:

- 1) Gain hands-on processor design experience by reviewing RTL Verilog design code for the initial version of the single-cycle MIPS processor discussed in class
- 2) Learn the basic technique for functionally verifying a processor

Preparation:

- 1) Thoroughly review MIPS instruction format and Lecture 3 slides - Processor.
- 2) Download the archive “*lab5_processor_design_1*” from the class Canvas site and retrieve all the archived files, including:
 - Verilog design source code files for the initial design of the single-cycle MIPS processor, which supports the following instructions: add, sub, and, or, slt, lw, sw, beq, addi, and j.
 - A memory file “*memfile.dat*” containing machine code (in hexadecimal) of the following test program to be used as an example for processor functional verification:

Label	Assembly	Address (hex)	Machine Code (hex)
main:	addi \$2, \$0, 5	0	20020005
	addi \$3, \$0, 12	4	2003000c
	addi \$7, \$3, -9	8	2067fff7
	or \$4, \$7, \$2	c	00e22025
	and \$5, \$3, \$4	10	00642824
	add \$5, \$5, \$4	14	00a42820
	beq \$5, \$7, end	18	10a7000a
	slt \$4, \$3, \$4	1c	0064202a
	beq \$4, \$0, around	20	10800001
	addi \$5, \$0, 0	24	20050000
	slt \$4, \$7, \$2	28	00e2202a
	add \$7, \$4, \$5	2c	00853820
	sub \$7, \$7, \$2	30	00e23822
	sw \$7, 68(\$3)	34	ac670044
around:	lw \$2, 80(\$0)	38	8c020050
	j end	3c	08000011
	addi \$2, \$0, 1	40	20020001
	sw \$2, 84(\$0)	44	ac020054
end:	j main	48	08000000

- A Verilog testbench file “*tb_mips_top*” for checking the processor’s execution results, instruction by instruction
- A signal naming file “*MIPS Signal Naming_v1*”, which gives definitions of all signals appeared in the source code, as well as their relationship with the signals appeared on the lecture slides

Tasks:

- 1) Carefully study the source code you obtained from the archive, then based on the source code, draw block diagrams of the following, for the initial design of the MIPS processor:
 - a) Datapath with microarchitecture details (not including memories)
 - b) Control unit with microarchitecture details
 - c) Instruction memory and data memory
 - d) Processor core (show interconnection between the datapath and its control unit, datapath internal microarchitecture is not required)
 - e) Complete processor (show interconnections between the processor core and its memories)
- 2) Use the resources included in the archive, build the DUT (the initial design of the single-cycle MIPS) and its test bench, then functionally verify the MIPS processor based on the given sample program. Note: Since the included testbench is an eyeballing one, you need to monitor the processor’s execution on each instruction by viewing waveforms of the following signals (display with the order below):

clk, rst, pc_current, instr, alu_out, we_dm, wd_dm, rd_dm

Any other signals you would like to probe (optional) must be displayed below the above signals.

Report Contents:

Besides the essential elements of your report, you should also include:

- a) all diagrams specified in Task 1, and
- b) waveforms captured from Task 2, with necessary comments/explanations.