San Jose State University
Department of Computer Engineering

CMPE 140 Lab Report

Lab 5 Report

Title Processor Design (1): Design Code Review and Functional Verification

Semester Spring 2019 **Date** 03/13/19

by

Name Nickolas Schiffer **SID** 012279319

Name Salvatore Nicosia **SID** 012013599

Lab Checkup Record

Week	Performed By (signature)	Checked By (signature)	Tasks Successfully Completed*	Tasks Partially Completed*	Tasks Failed or Not Performed*
1	SN		100%		

*** Detailed descriptions must be given in the report.**

I. INTRODUCTION

The purpose of this lab is to gain experience in processor design by reviewing RTL Verilog code for the initial version of the single-cycle MIPS processor. The Verilog code provided was also used to learn the basic techniques for functionally verifying a processor.

II. DESIGN METHODOLOGY

The source code provided for the initial version of the single-cycle MIPS processor was carefully studied to understand the signal names and the build blocks of this processor. Based on the source code the block diagrams for the Datapath, Control Unit, Instruction Memory, Data Memory, Processor Core, and Complete Processor were drawn.

The first block diagram shown in *Figure 1* is the Datapath with the microarchitecture which shows the interconnection with the various modules including the external signals coming and going to other submodules of the MIPS Processor.

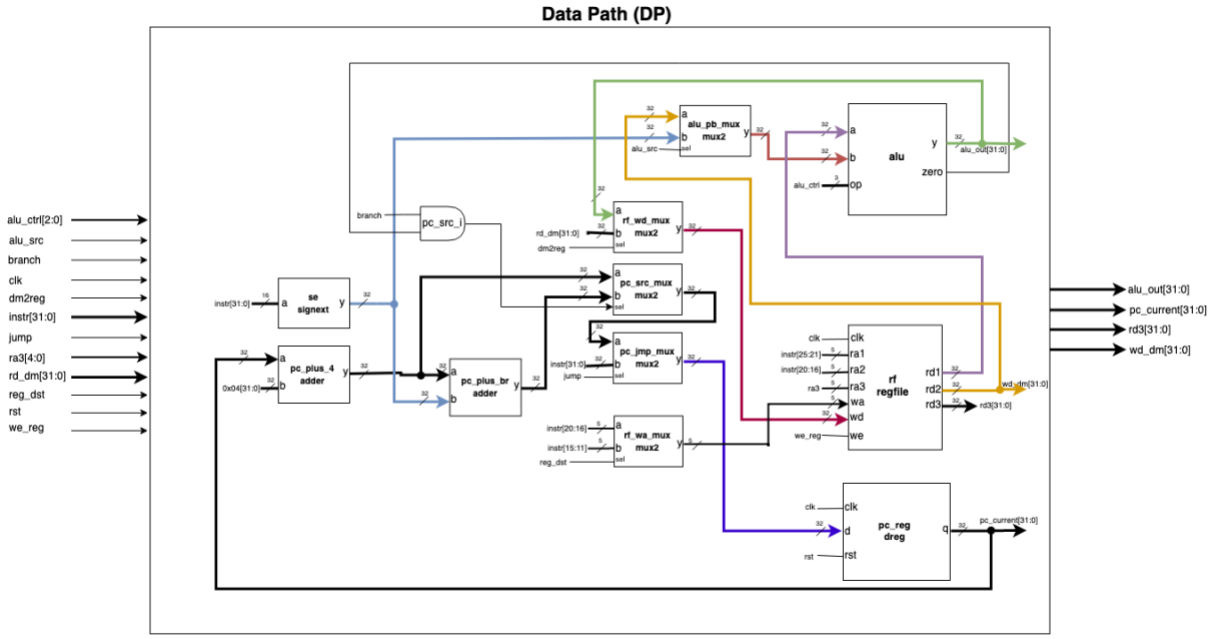

Figure 1: Datapath with Microarchitecture

The second block diagram shown in *Figure 2* is the Control Unit (CU) with microarchitecture. This diagram shows the external signals as well as the internal decoders.

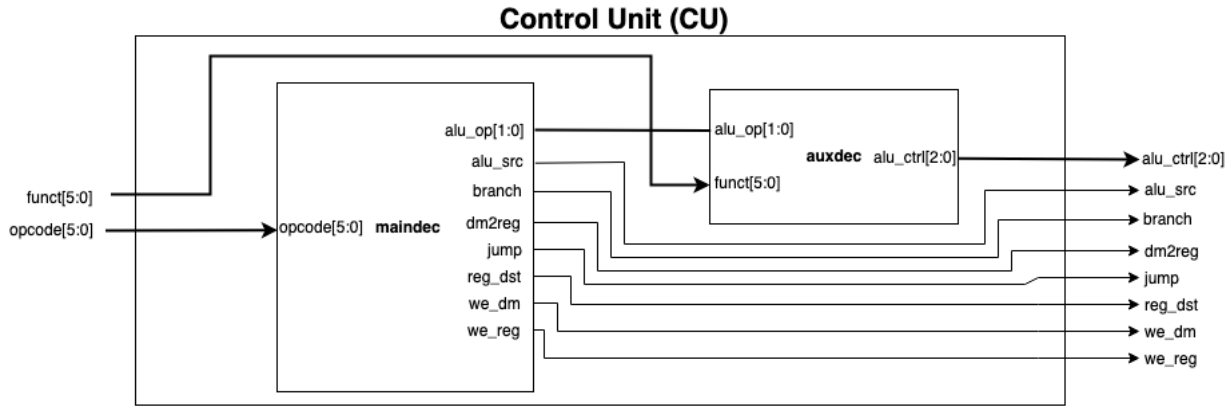

Figure 2: Control Unit with Microarchitecture

The block diagrams shown in *Figures 3 and 4* are the Instruction Memory and Data Memory used by in MIPS Processor. These memories are used by the processor core to store the instructions and data.

Figure 3: Instruction Memory

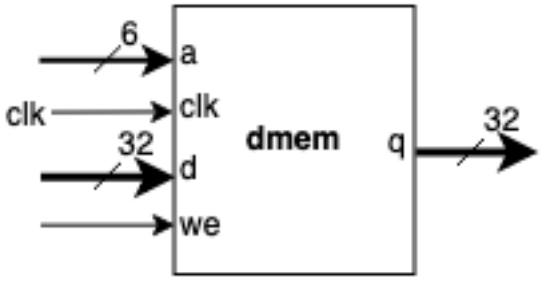

Figure 4: Data Memory

The next diagram shown in *Figure 5* MIPS Processor Core which shows the communication between the Control Unit and Datapath. The processor core receives the instructions and data from the memories described above. The processor then outputs the results from the internal register files, program counter, and ALU results back to the instruction and data memory.

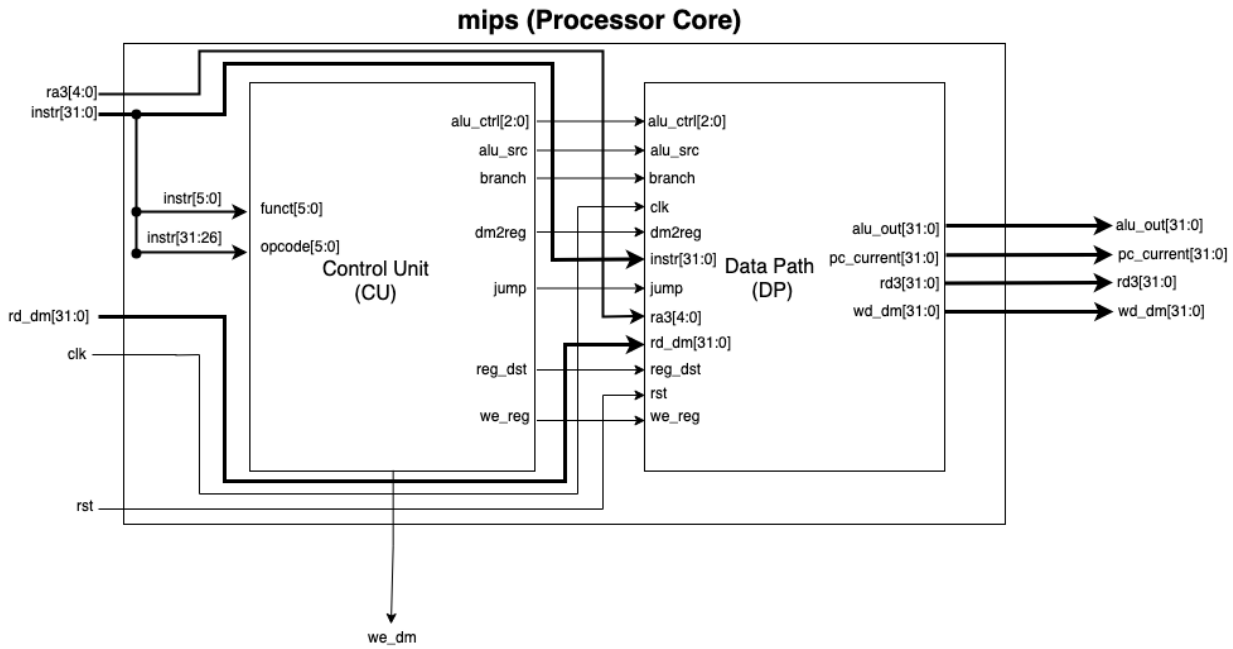

Figure 5: MIPS Processor Core

The last block diagram shown in *Figure 6* shows complete MIPS processor which include the processor core interconnection with the instruction and data memory as well as the incoming and outgoing signals.

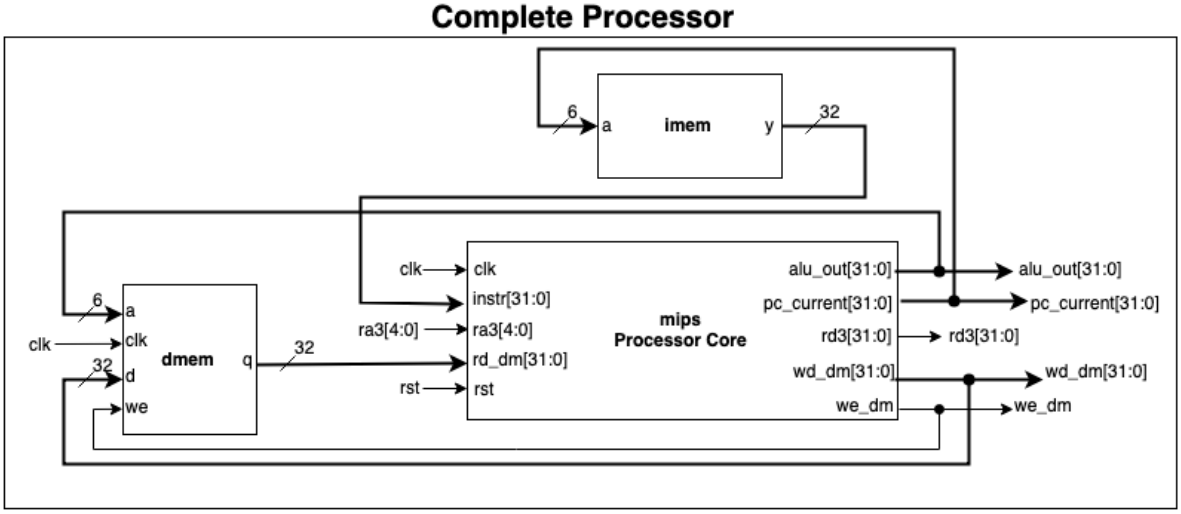

Figure 6: MIPS Complete Processor

III. TESTING PROCEDURE

The provided Verilog code of the initial version of the single-cycle MIPS processor was functionally verified through the the testbench file *"tb_mips_top"* found in the appendix. The waveforms generated by the Verilog eyeballing testbench was used to check processor's execution results for each instruction by monitoring the following signals:

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

The processor's execution results are shown in *Figures 7 and 8.* These results follow the code shown in the lab assignment. The flow of the program can be observed and validated, with the correct registers at the correct memory location being manipulated accordingly.

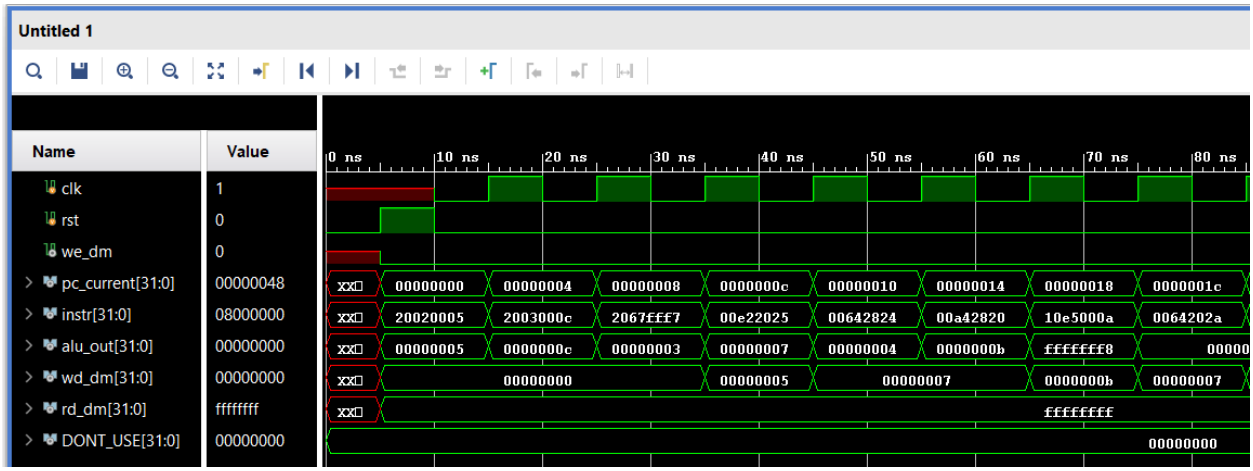

Figure 7: Processor's execution result showing the signals for each instruction.

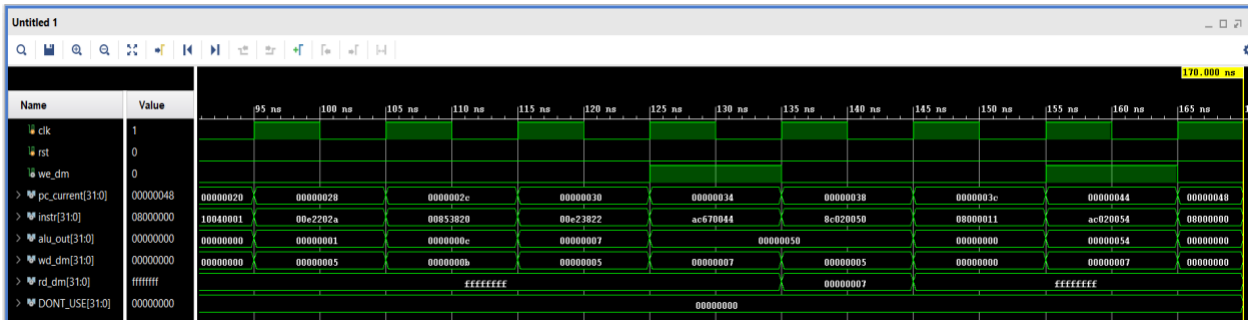

Figure 8: Processor's execution result showing the signals for each instruction.

IV. CONCLUSION

This lab helped to gain experience in processor design by reviewing the RTL code of the initial version of the single-cycle MIPS processor. The analysis of the code allowed to draw the block diagrams for the Datapath, Control Unit, Instruction Memory, Data Memory, Processor Core, and Complete Processor. In addition, the basic techniques to functionally verify a processor were learnt. This was achieved by the verifying processor's execution results through the waveforms generated by the testbench.

V. SUCCESSFUL TASKS

1. Draw block diagrams for Datapath, Control Unit, Instruction Memory, Data Memory, Processor Core, and Complete Processor.
2. Built the DUT of the initial design of the signal cycle MIPS processor and its testbench.

3. Functionally verified the MIPS processor through the Verilog Eyeballing testbench.
4. Captured waveforms generated from the testbench.

VI. APPENDIX

A. SOURCE CODE:

tb_mips_top.v

```
module tb_mips_top;

    reg         clk;
    reg         rst;
    wire        we_dm;
    wire [31:0] pc_current;
    wire [31:0] instr;
    wire [31:0] alu_out;
    wire [31:0] wd_dm;
    wire [31:0] rd_dm;
    wire [31:0] DONT_USE;

    mips_top DUT (
             .clk            (clk),
             .rst            (rst),
             .we_dm          (we_dm),
             .ra3            (5'h0),
             .pc_current     (pc_current),
             .instr          (instr),
             .alu_out        (alu_out),
             .wd_dm          (wd_dm),
             .rd_dm          (rd_dm),
             .rd3            (DONT_USE)
        );

    task tick;
    begin
        clk = 1'b0; #5;
        clk = 1'b1; #5;
    end
    endtask

    task reset;
    begin
        rst = 1'b0; #5;
        rst = 1'b1; #5;
        rst = 1'b0;
    end
    endtask

    initial begin
        reset;
        while(pc_current != 32'h48) tick;
        $finish;
    end

endmodule
```