**Lab 4 – R-type Datapath**

CECS 341 – Computer Architecture & Organization

Kenry Yu, 028210726

Olenka Bilinska, 028897191

Garret Towner, 028303091

Professor: Mandy He



California State University, Long Beach

College of Engineering

1250 Bellflower Blvd, Long Beach, CA 90840

March 18, 2022

**Goal/Objective:**

The objective of this lab is to design a MIPS Datapath for R-type instructions.

**Technical Description/Steps:**

To accomplish the objective, there are four design modules to design (program counter, program counter adder, control, and Datapath), three design modules to utilize (instruction memory, register file, and ALU), two data files to import (“imem” and “regfile32”), and a testbench to simulate the process.

Method for designing the Datapath:

To design the Datapath, all other design modules should be implemented first. After designing other modules, we initialized wires to connect the modules in the Datapath.

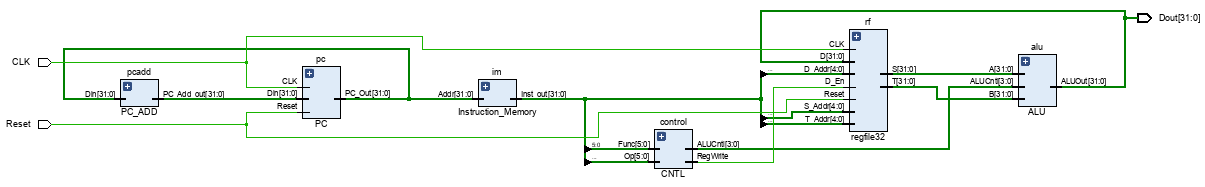
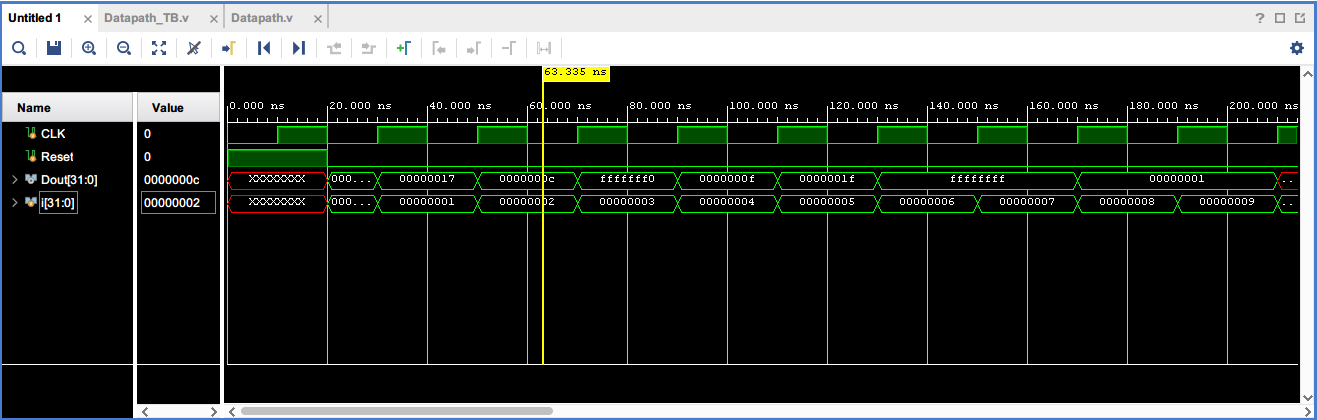
Modification of the ALU:

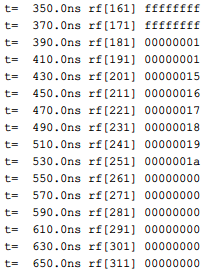
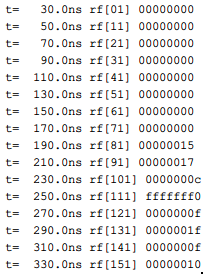
The lab required us to add two new operators to our ALU module, the SLT signed and SLT unsigned. To accomplish this feature, we used if else statement of Verilog to determine the output value. For SLT signed operator, we also added the 2’s complement to compute the correct output.

Testbench:

In the testbench, we import the instructions and register file from the data files. Then, we use a task that would utilize the data from the data files to output the results of the instruction into console log.

**Results:**

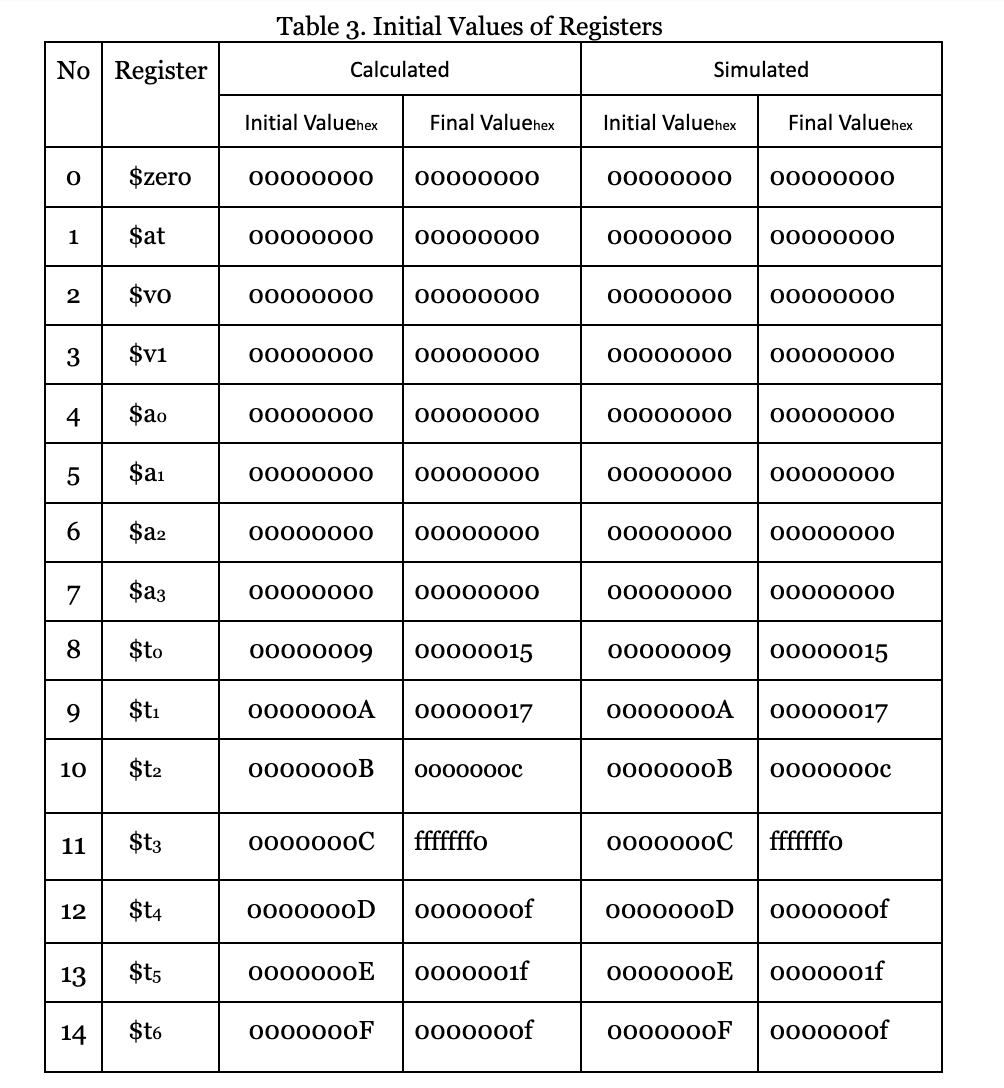
This is the schematic of the Datapath module. The schematic shows that wires have connected all the modules to work together. The waveform shows the clock is pulsing as desired. At the rising edge of the clock, the output changes according to different instructions from the data files.

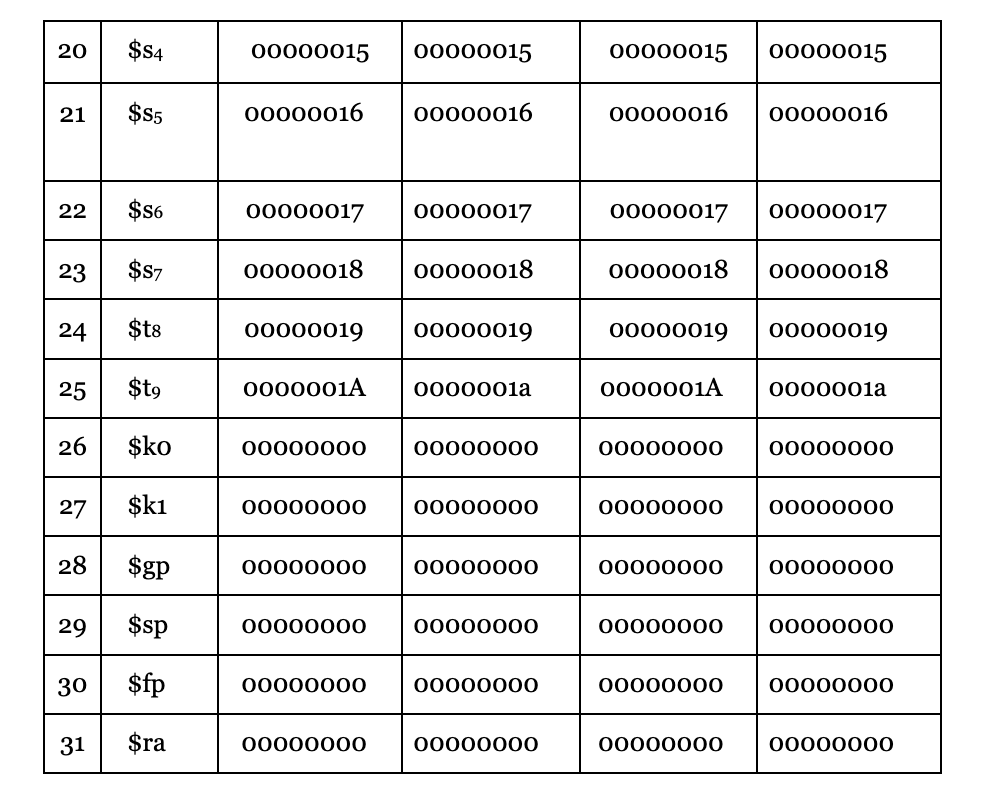
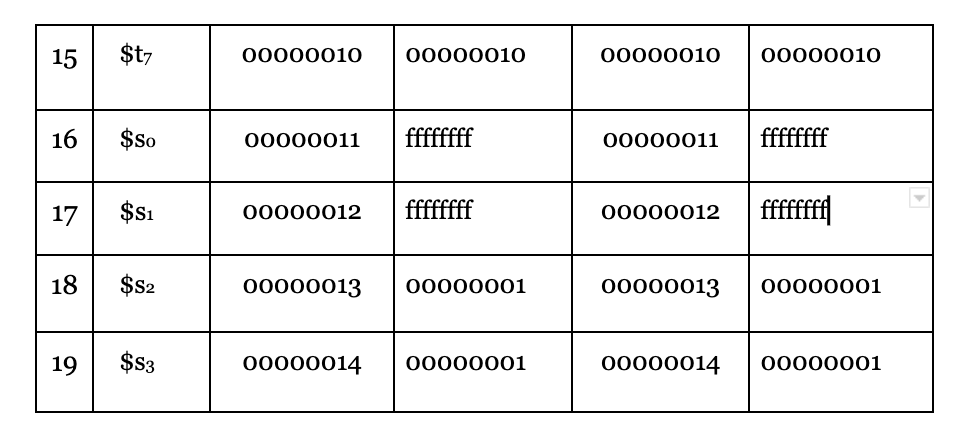


In the console log, the saved registers are displayed and proof that the Datapath program is working and generating correct output as intended.

**Conclusion:**

In this lab, we learned how to wire multiple modules together, read data from data files, and create tasks in the module. In addition, we also strengthened our knowledge in MIPS because we learned how each module worked together to compute instructions. We encounter some errors in setting the clock to pulse as intended, but we quickly notice we have the wrong symbol in the code and fix it. After the changes, the outputs are successfully displayed.





Hand solutions:

