

# CSCE 3301

Milestone 2 Report

Single-cycle RISC-V32I Processor

Jacqueline Azar: 900184062

John El Gallab: 900193761

Youssef Agiza: 900192237

## Table Of Content:

- 1- Objectives
- 2- Assumptions
- 3- Schematics and Analysis
- 4- Test Cases and Findings
- 5- List of Figure, Tools, and References Used

## **1- Objectives:**

Design a single-cycle implementation for the RISC-V32I processor using Verilog Hardware Description Language.

## **2- Assumptions:**

The processor currently only supports the base integer instruction set that was specified in RISC-V technical specifications. All 40 instructions were implemented as described in the specifications except the ECALL, EBREAK, and FENCE instructions. Instead, they were treated as a halting instruction that prevents the program counter from being updated. Two separate memories were used for the instruction ROM and Data memory.

## **3- Schematics and Analysis:**

The following schematic diagram was generated using the XILINIX VIVADO tool set to demonstrate the hardware and wiring that was described in the Verilog files. This diagram is also attached as a pdf file to the submission.

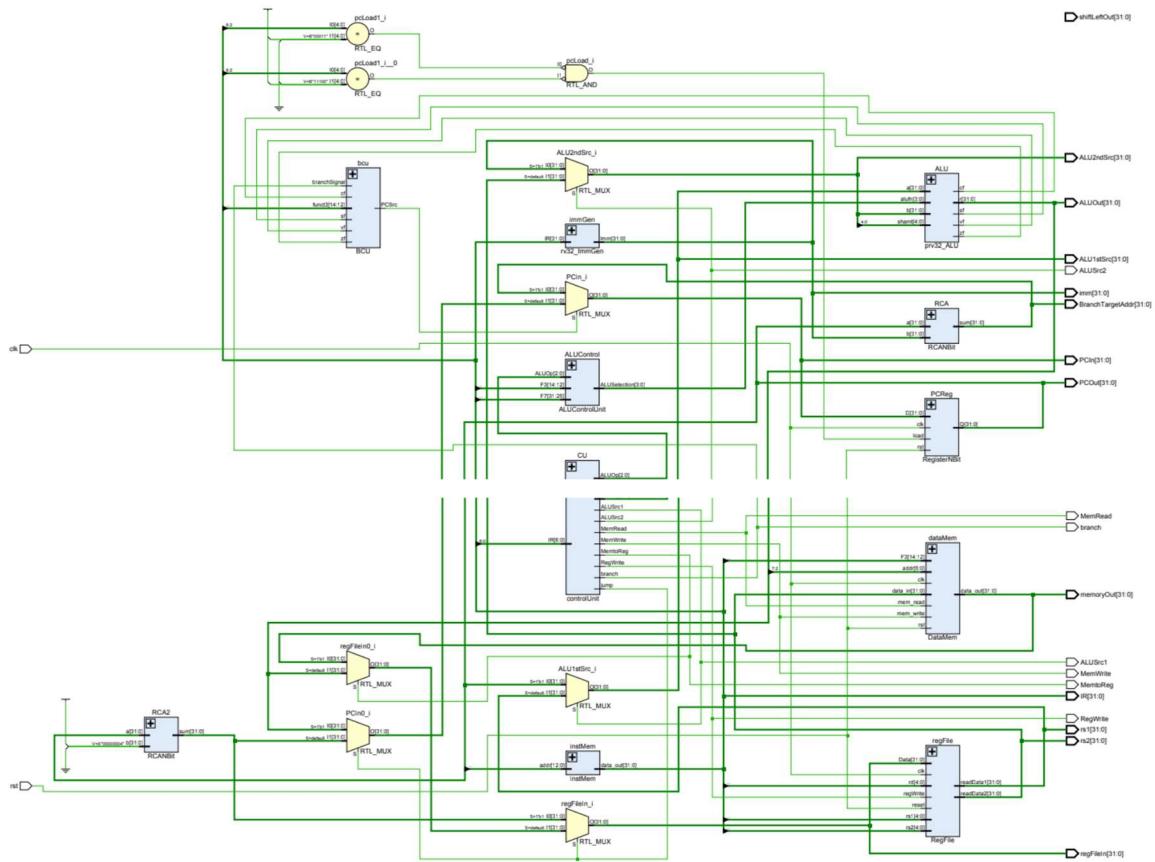


Figure 3.1: Schematic Diagram of The Processor Showing All Input/Output Ports

Another schematic (shown in figure 3.2) was drawn manually to be used as an implementation guide while designing the processor.

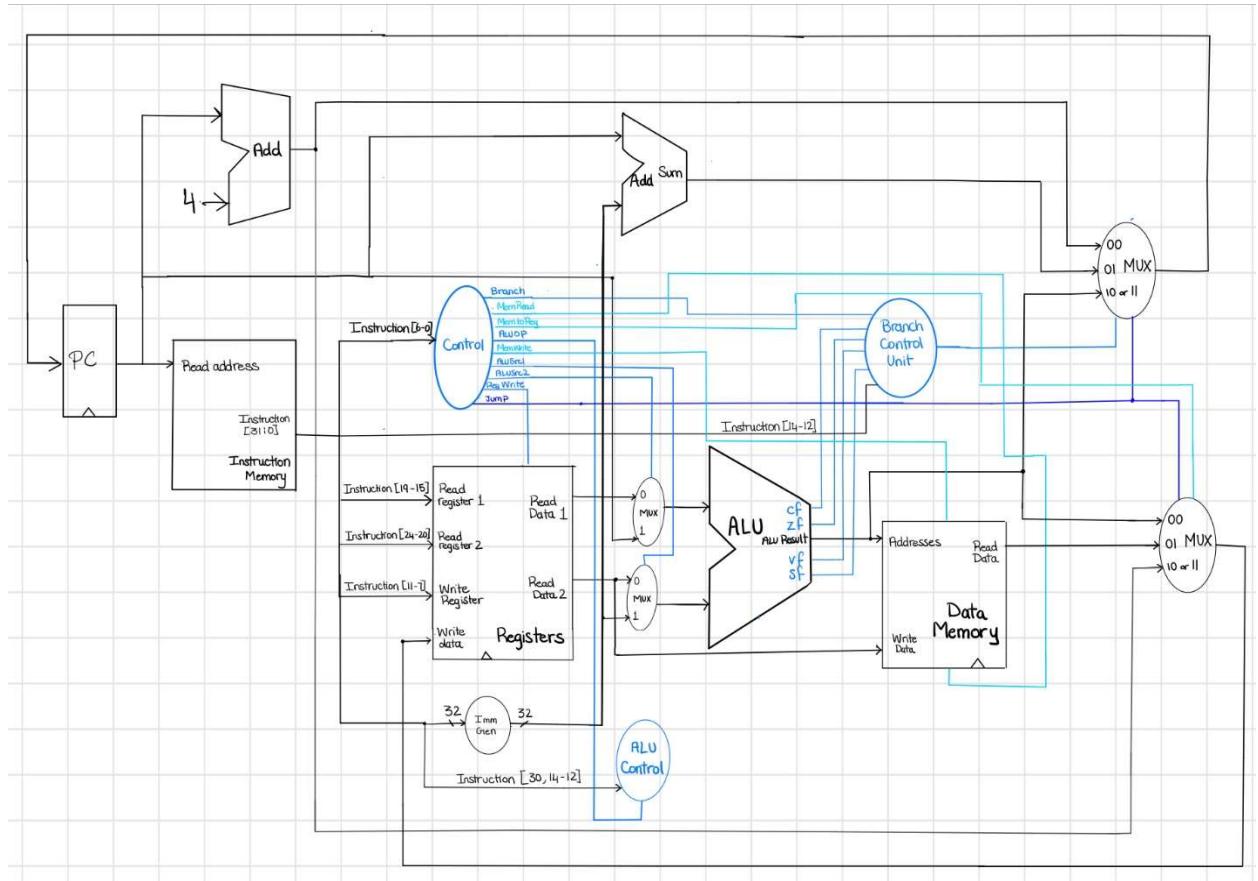


Figure 3.2: Hand drawn Schematic Diagram of The Processor

A utilization report was generated by XILINX VIVADO shown in figure 3.3.

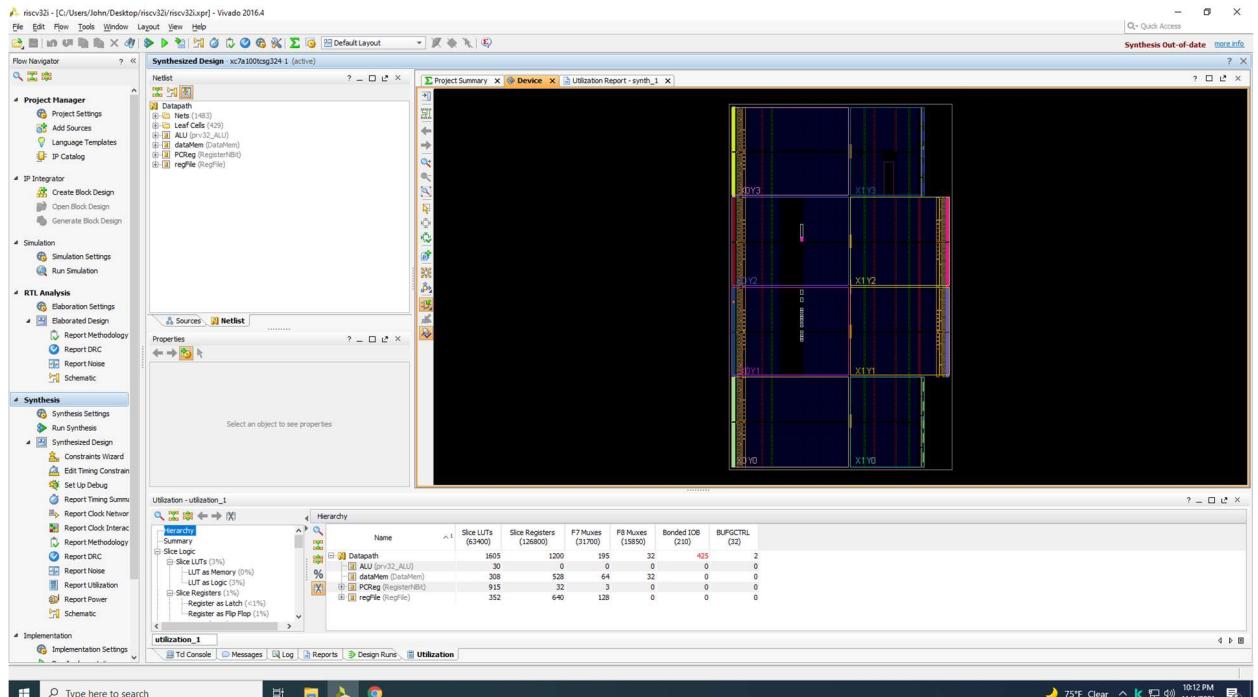


Figure 3.3: Utilization Report

A timing report was generated by XILINX VIVADO shown in figure 3.4.

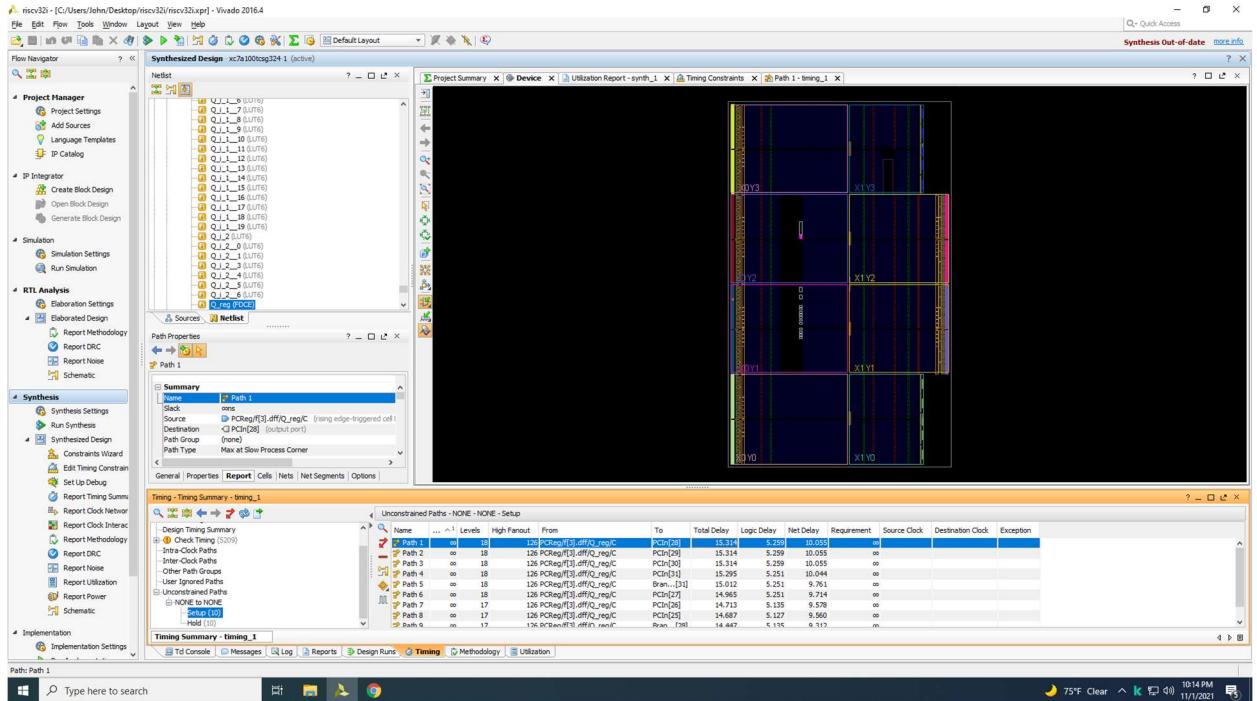


Figure 3.4: Timing Report

From this timing analysis shown in figure 3.4, we can conclude that the critical path in our design spans over 15.314 nanoseconds. This means that the maximum clock frequency that can be used with the current implementation of this processor is  $1 / (15.314 * 10^{-9}) = 65.3$  MHz.

#### 4- Test Cases and Findings:

The processor was tested rigorously using the RARS simulation test suite of instructions for a set of instructions (suite source cited in references). The remaining set of instructions that was not present in the suite were benchmarked against simple programs of our creation and were all found to result in accurate results of execution except for the SRA and SRAI instructions. Further investigations are needed to debug these 2 instructions. Below is the screenshot of all tested instructions. A value of 42 in the register x10 indicates that the instruction was completed successfully for the about

100 test cases provided for that instruction testing suite. The non-suite-based tests we compared against the predicted values on the RARS simulator to ensure accuracy.

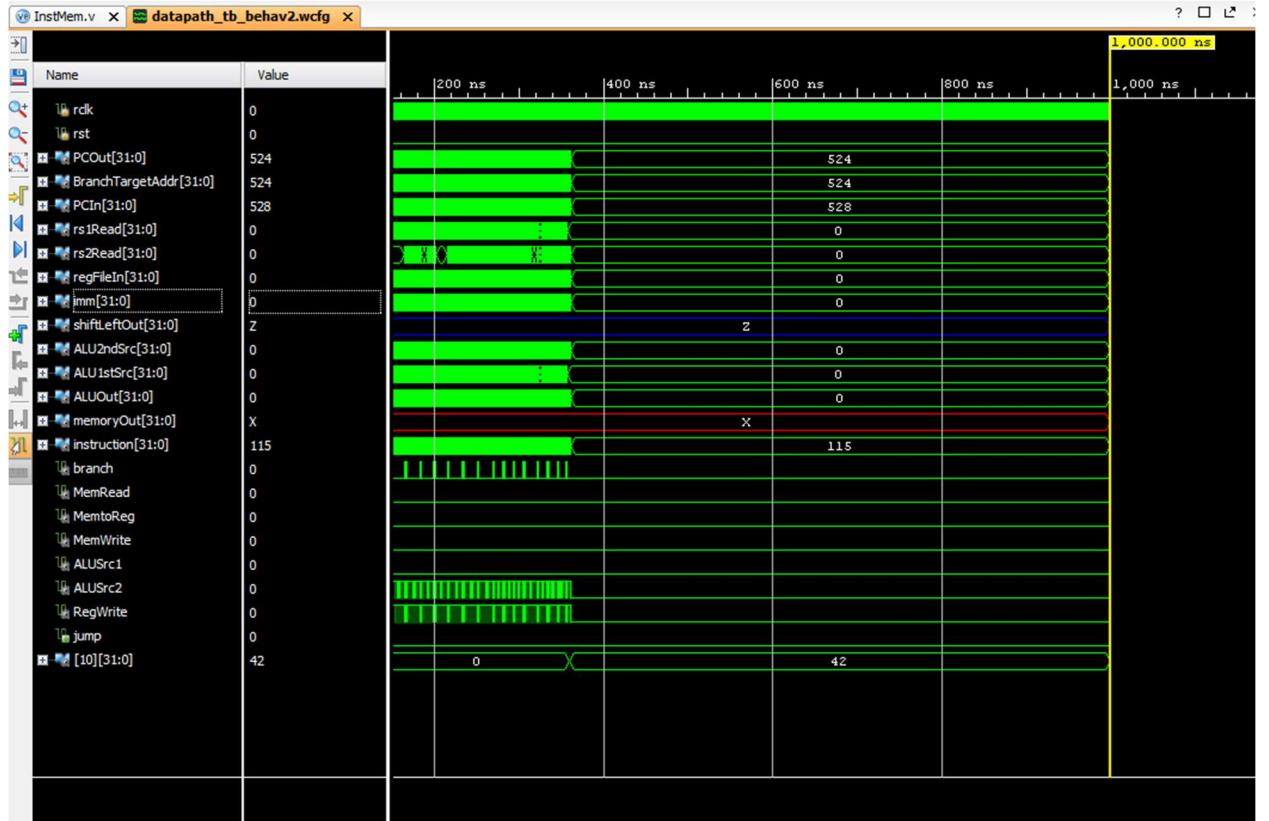


Figure 4.1 ADD Instruction

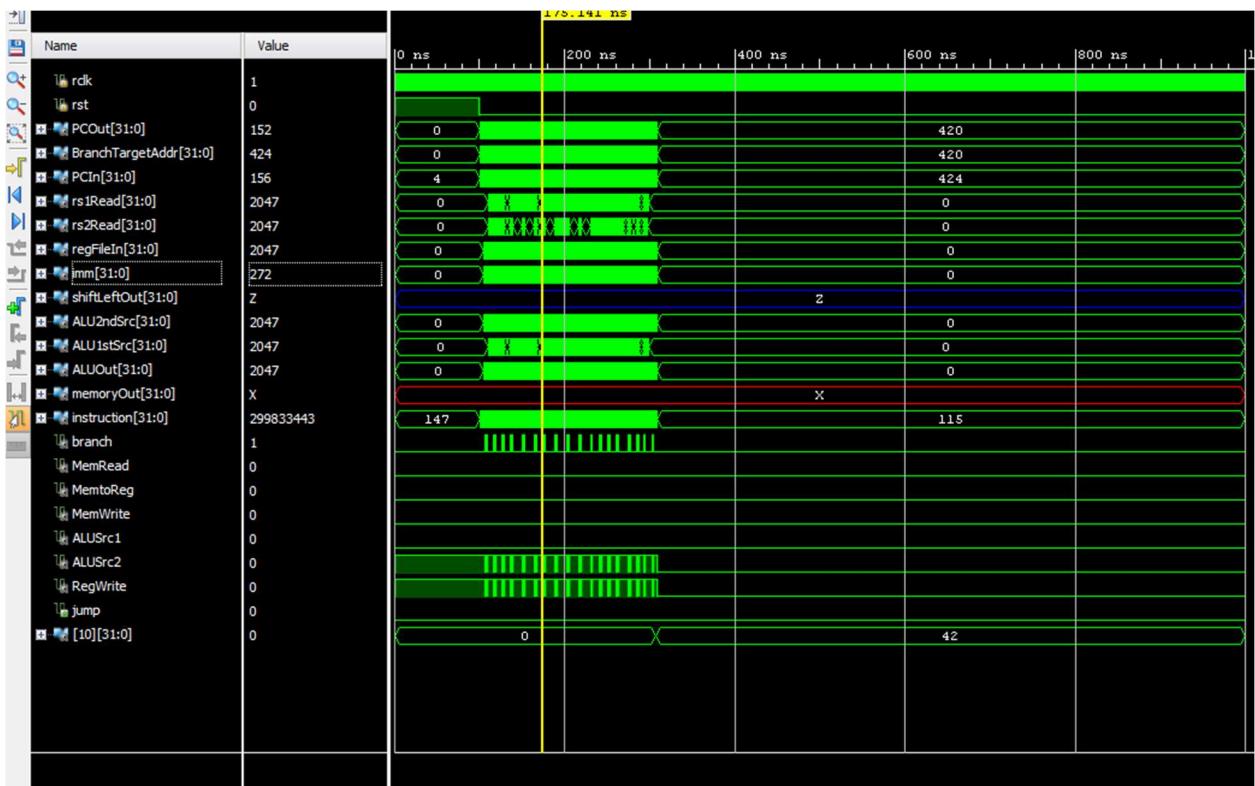


Figure 4.2 ADDI Instruction

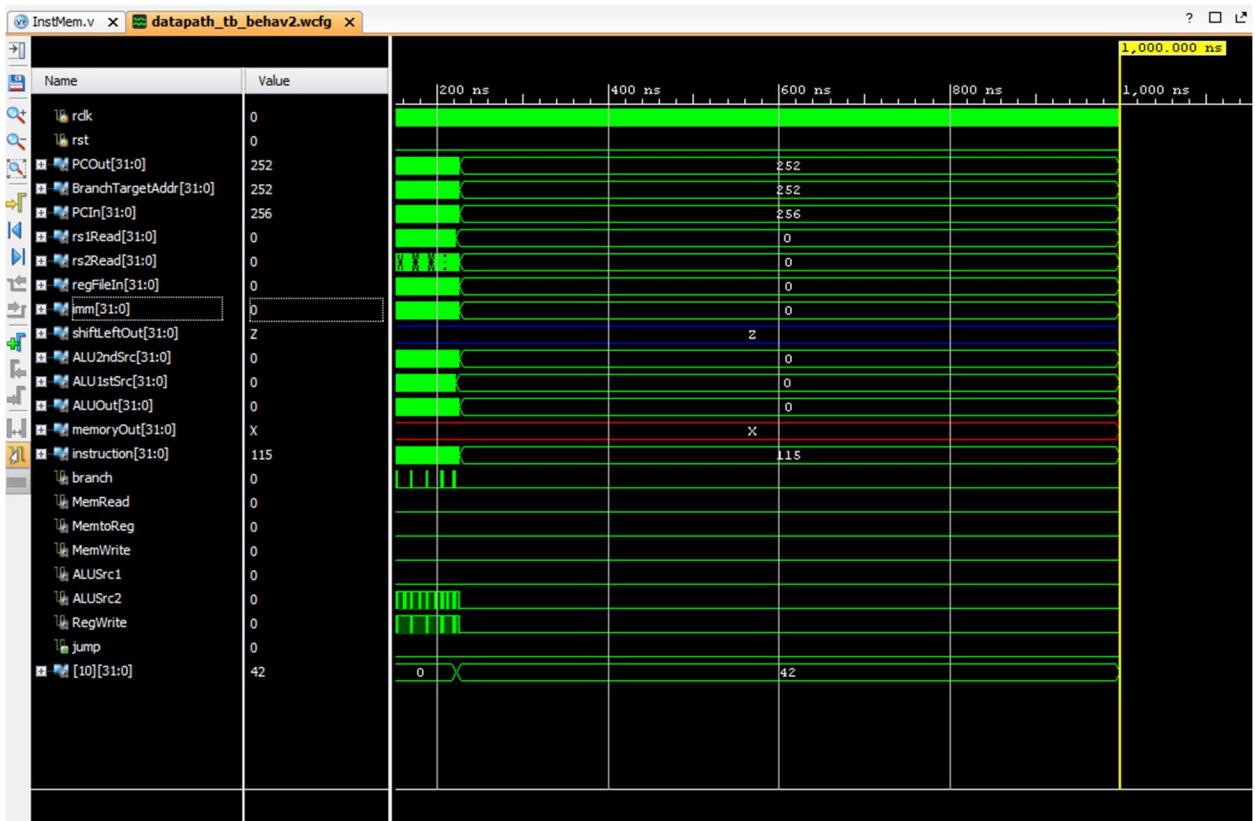


Figure 4.3 AND Instruction

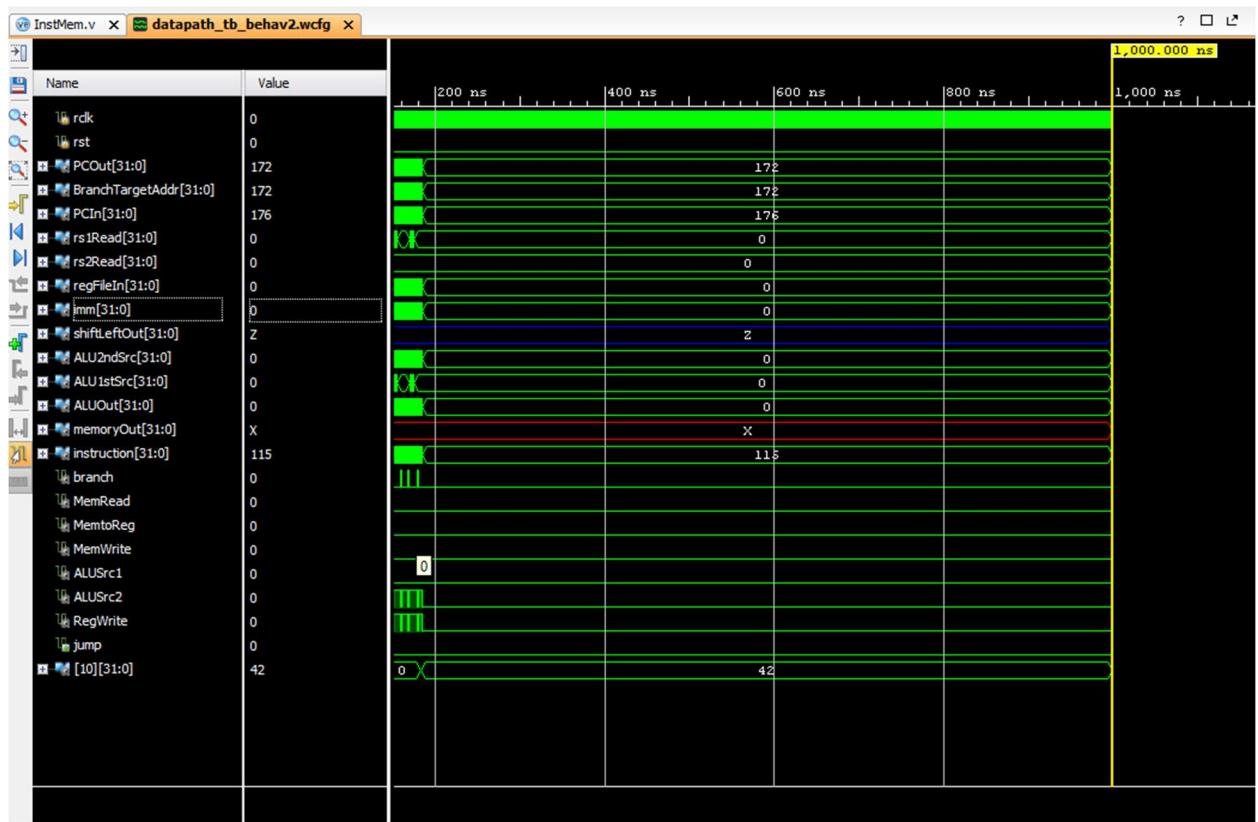


Figure 4.4 ANDI Instruction

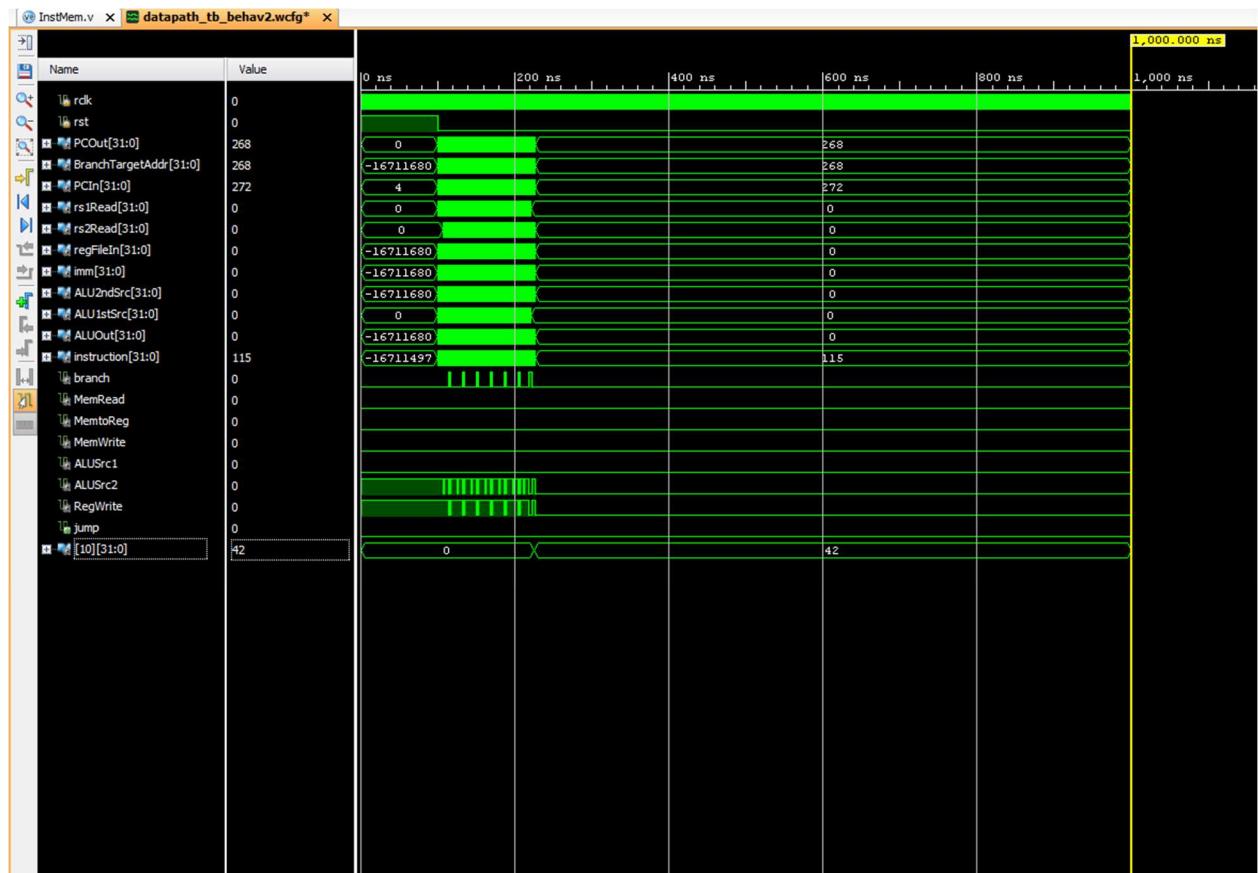


Figure 4.5 OR Instruction

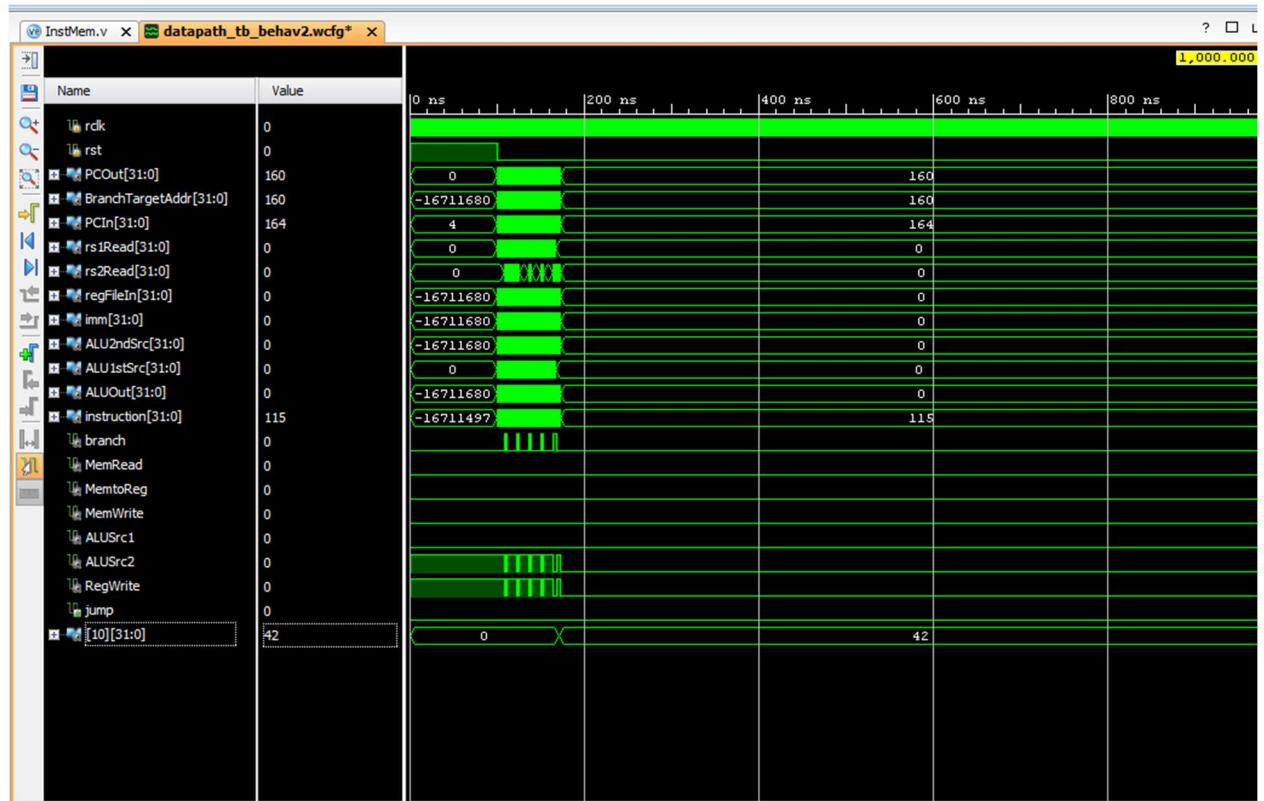


Figure 4.6 ORI Instruction

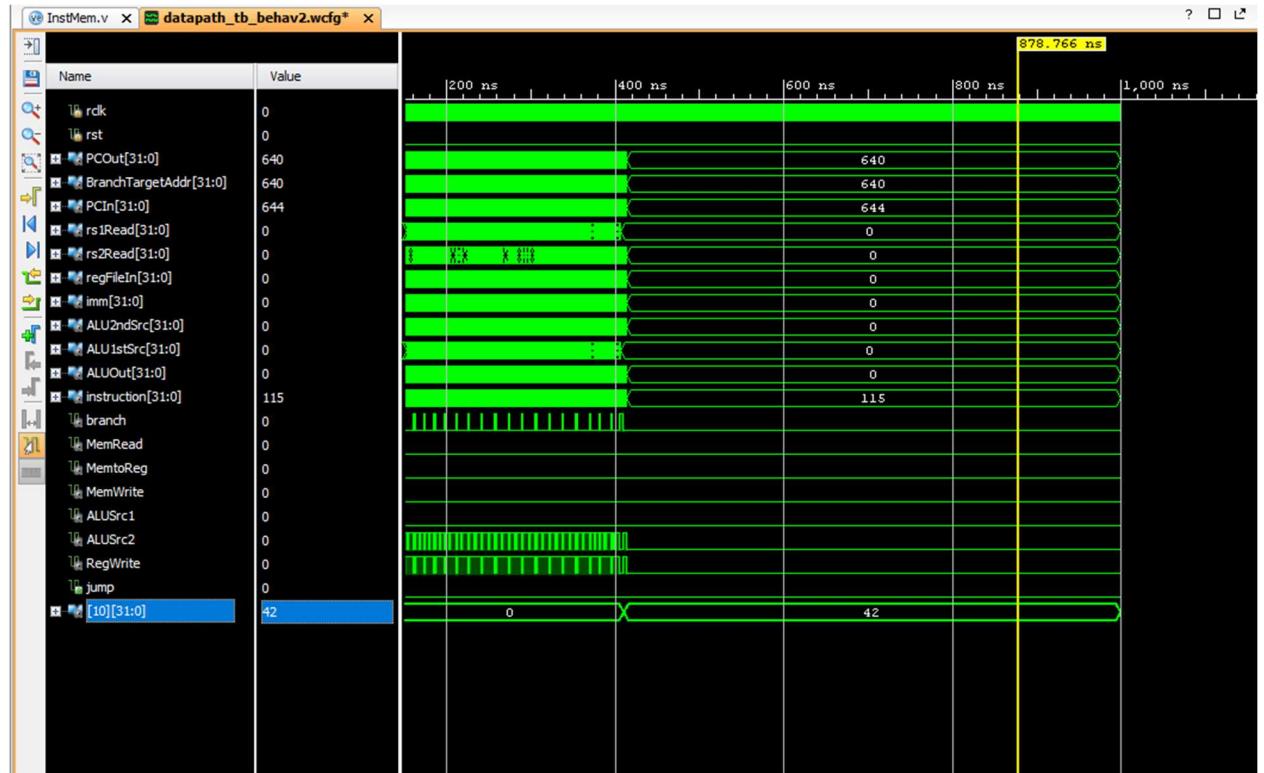


Figure 4.7 SLL Instruction

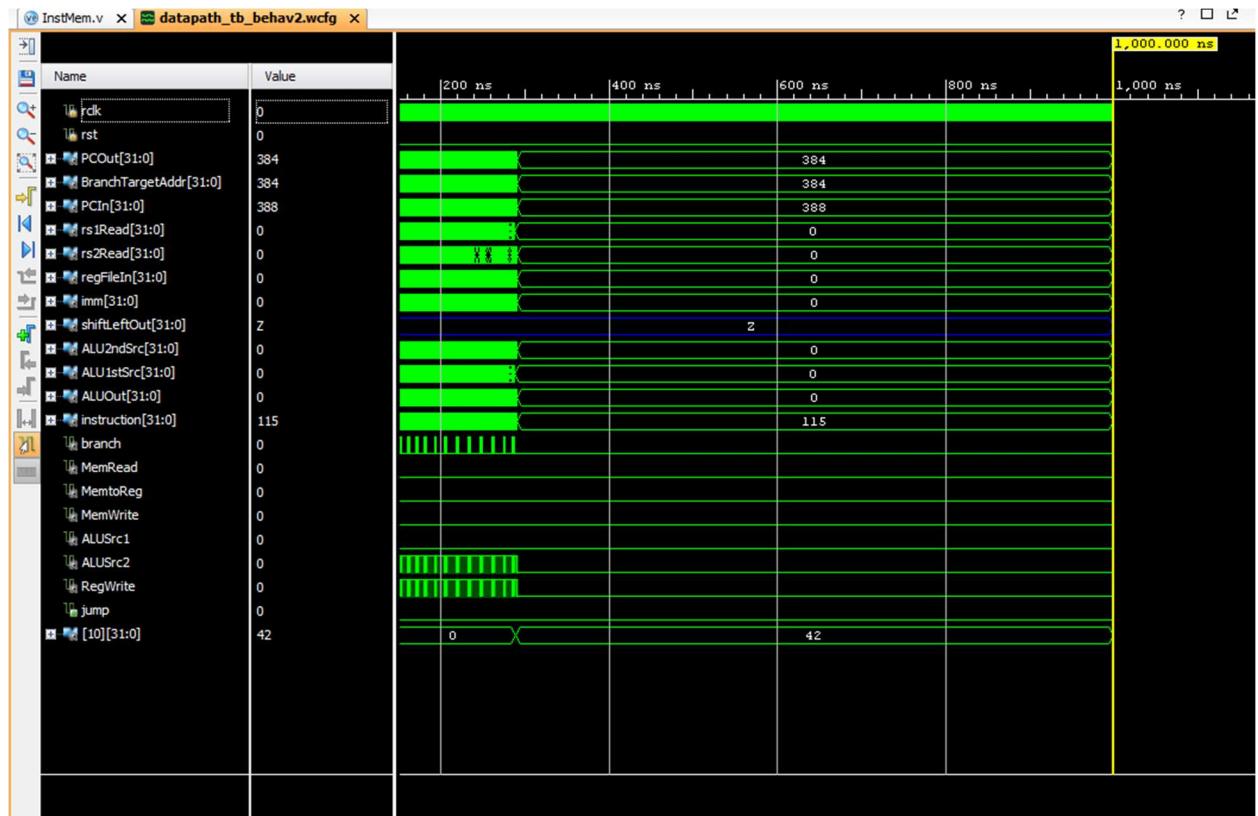


Figure 4.8 SLLI Instruction

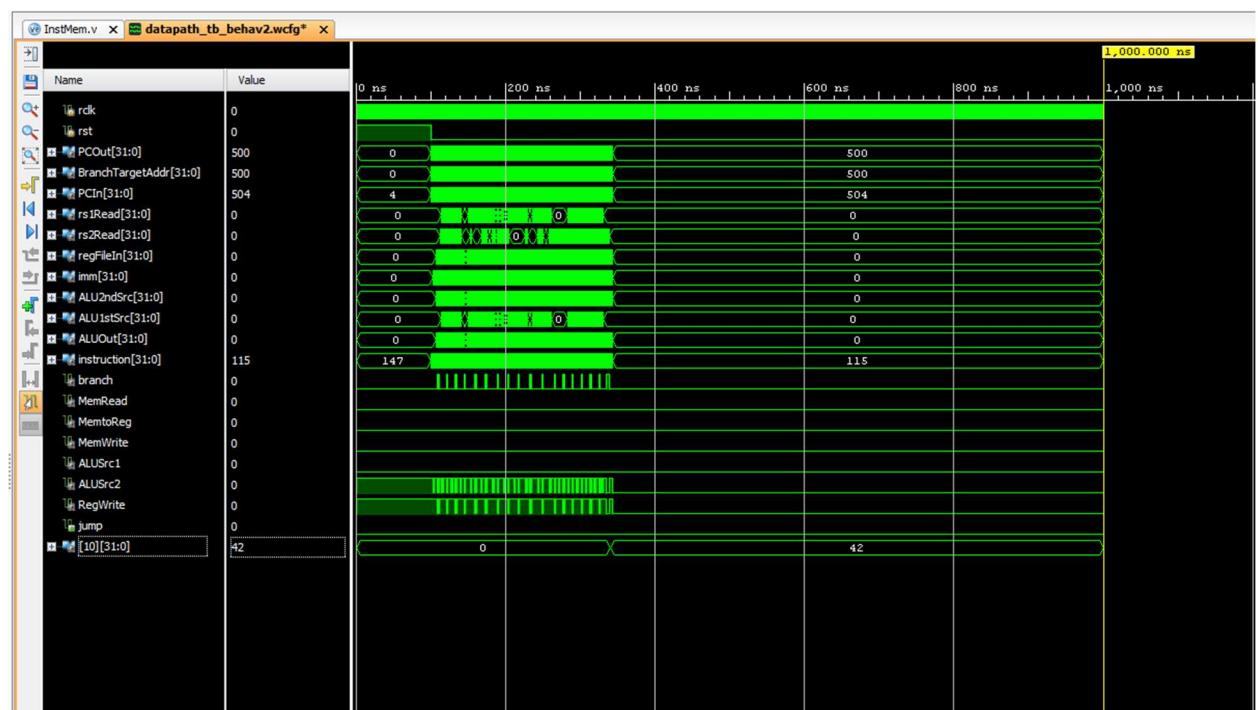


Figure 4.7 SLT Instruction

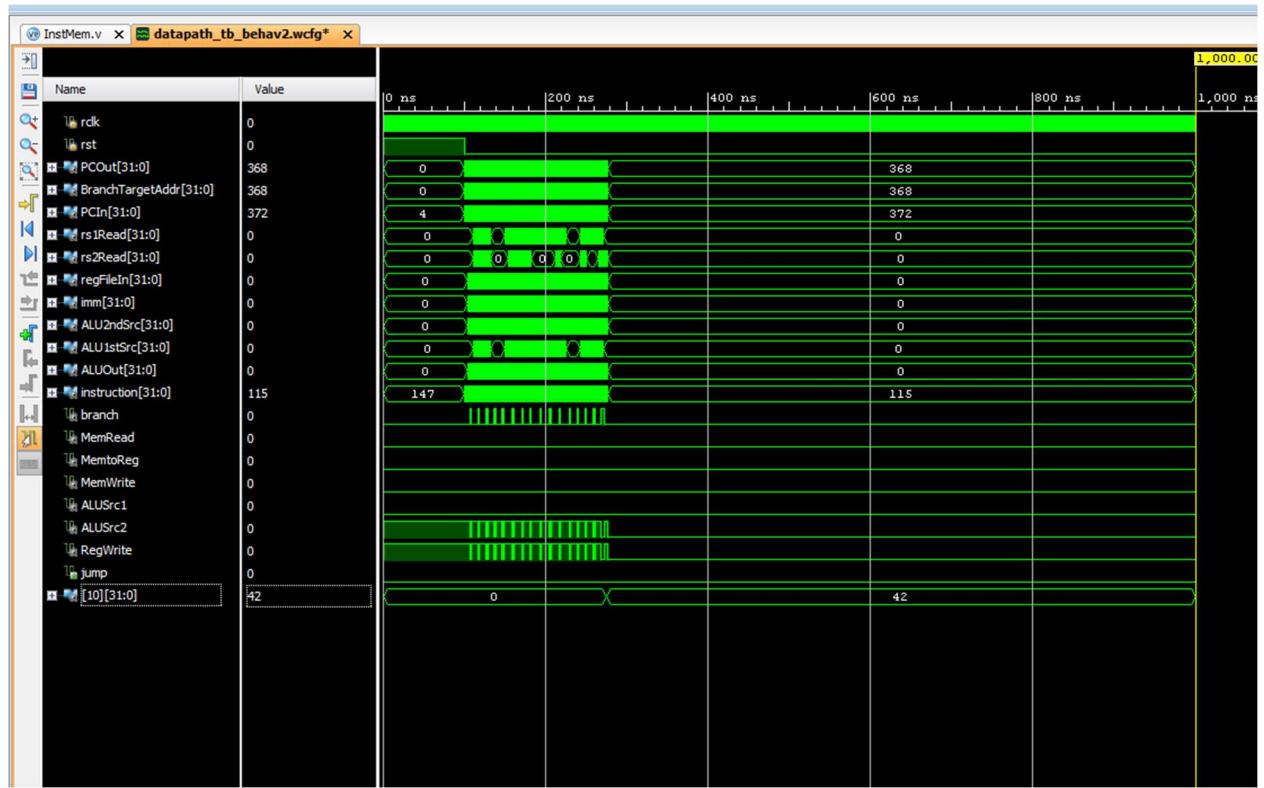


Figure 4.8 SLTI Instruction

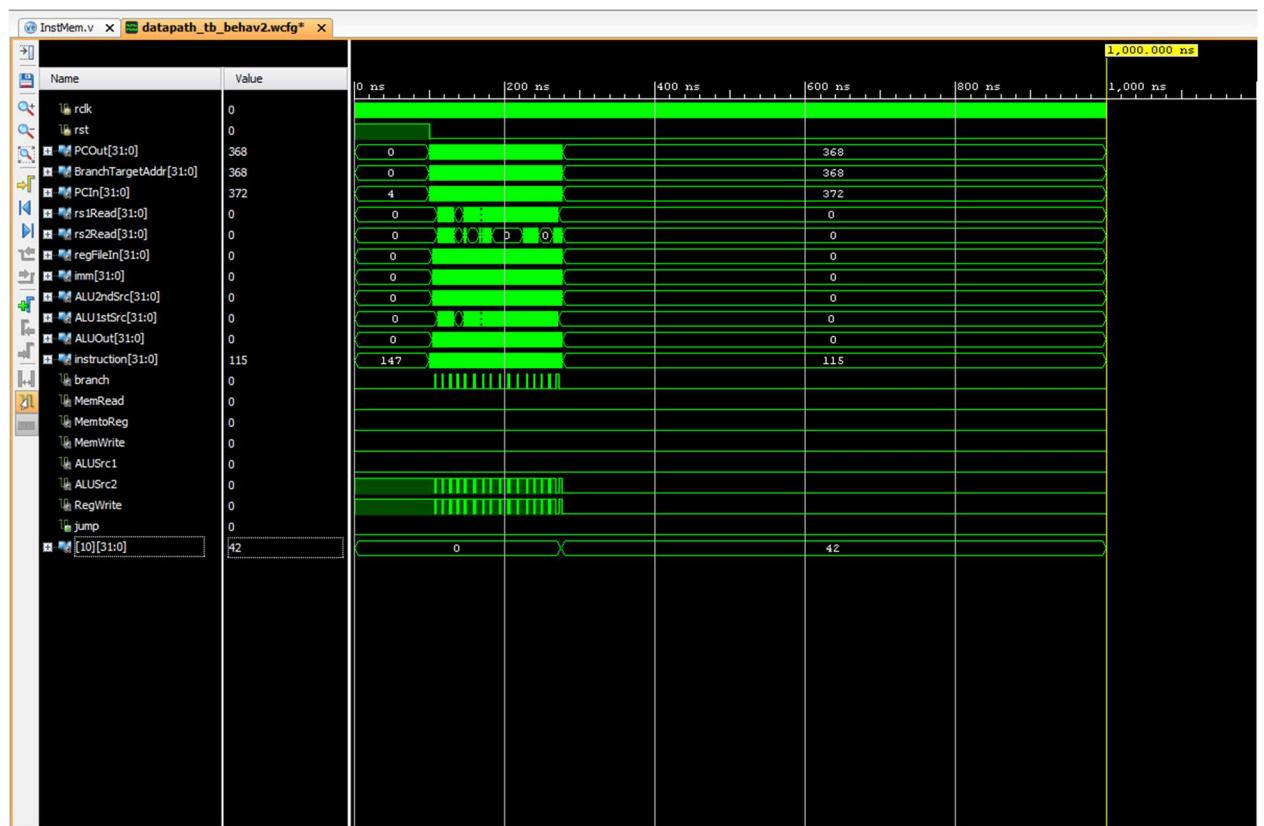


Figure 4.7 SLTIU Instruction

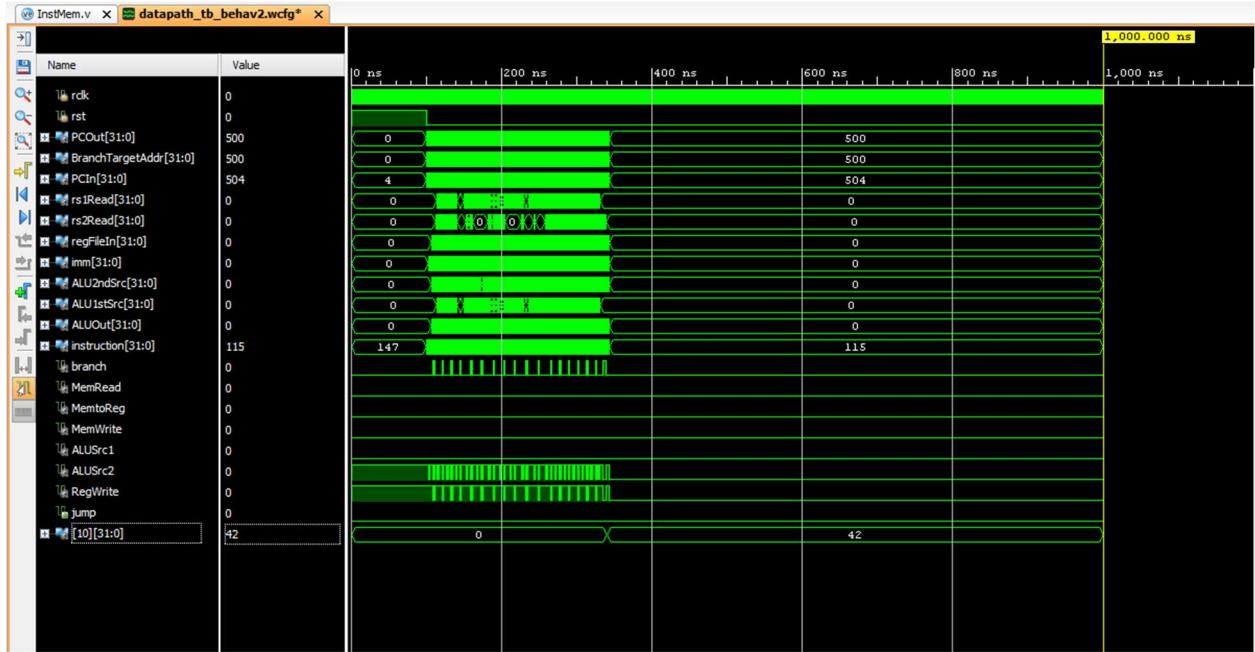


Figure 4.8 SLTU Instruction

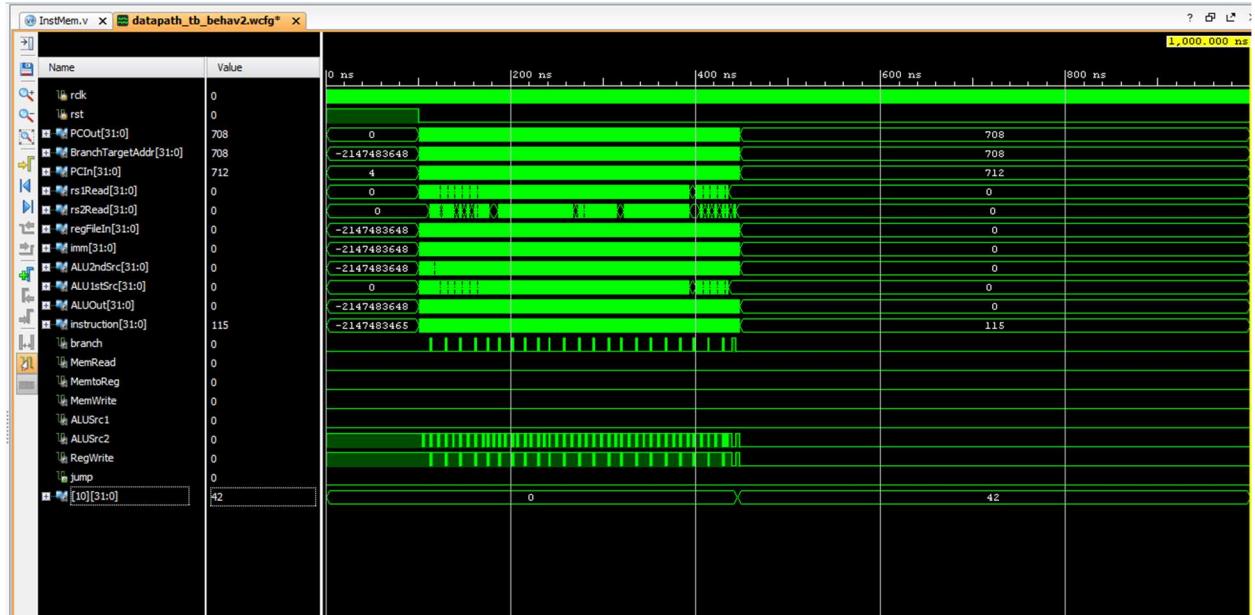


Figure 4.9 SRL Instruction



Figure 4.10 SRLI Instruction

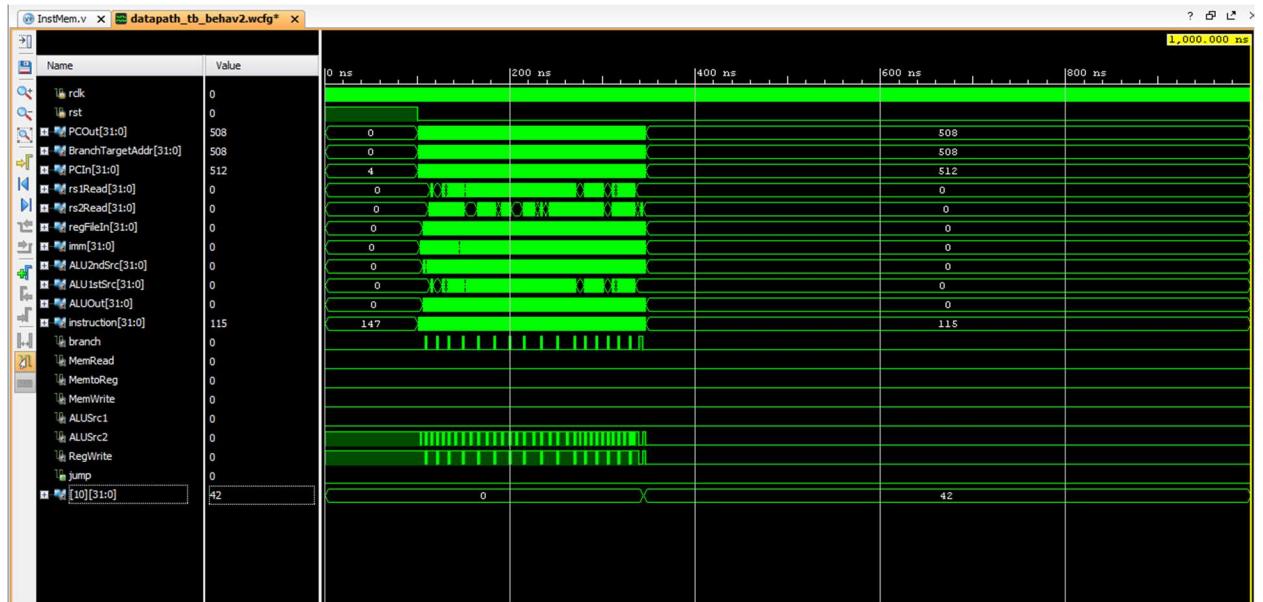


Figure 4.11 SUB Instruction

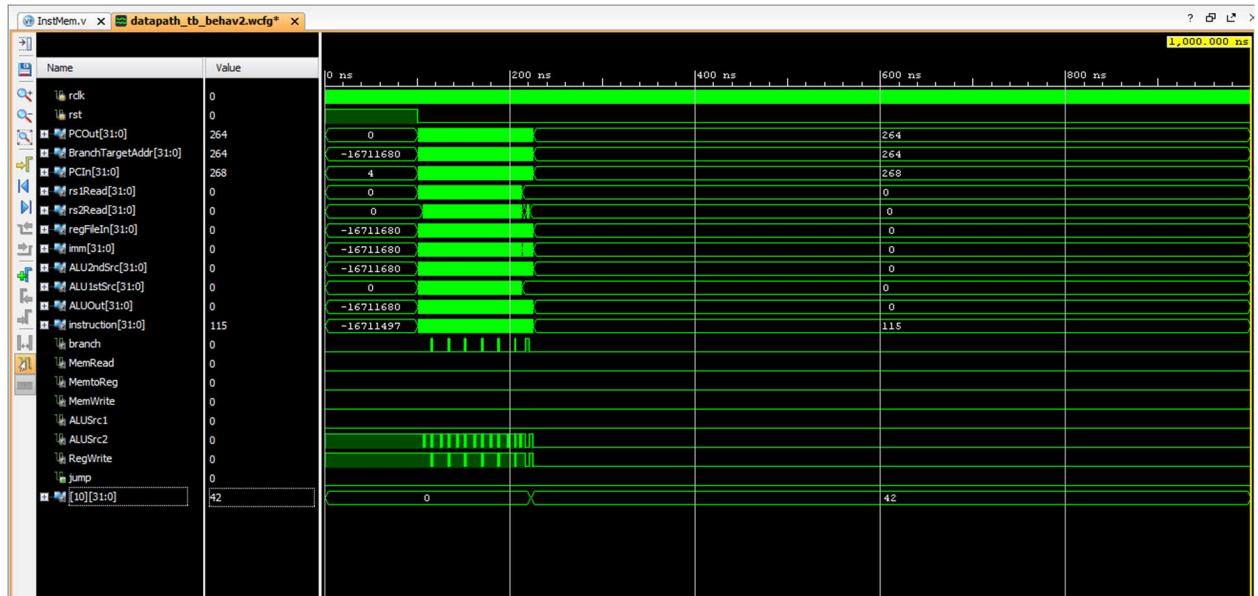


Figure 4.12 XOR Instruction

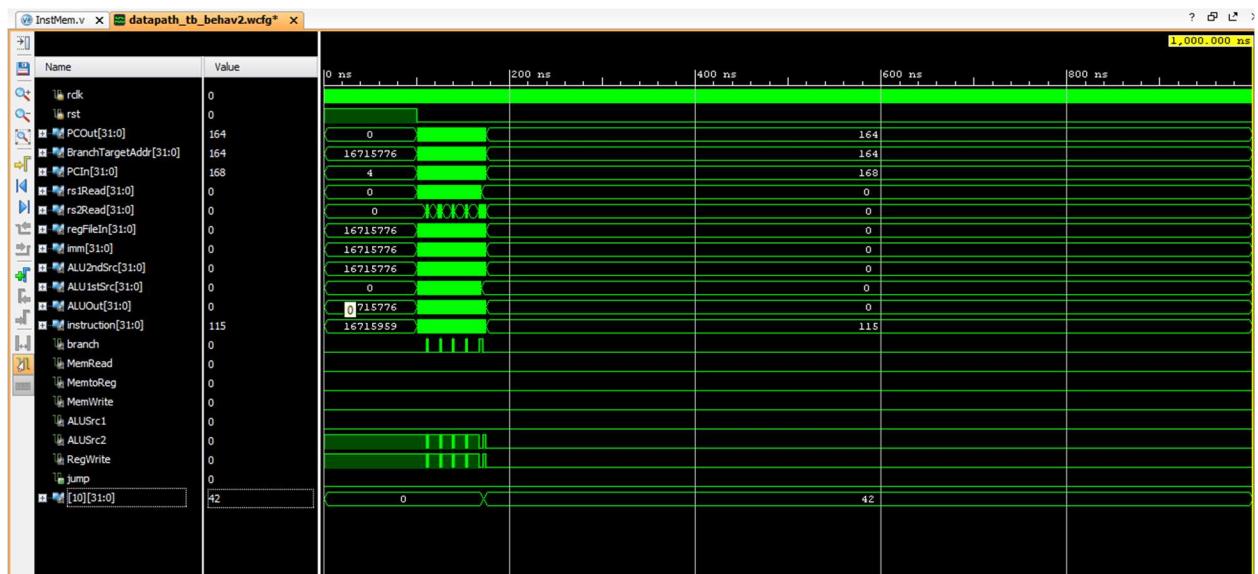


Figure 4.13 XORI Instruction

[7][31:0]	1			1
[6][31:0]	0			0
[5][31:0]	-15			-15
[4][31:0]	15			15
[3][31:0]	0			0

Figure 4.14 AND Instruction

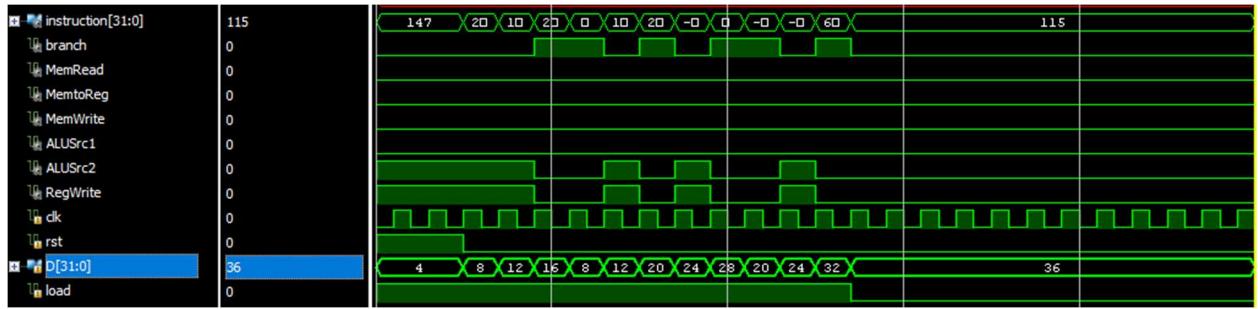


Figure 4.15 All Signed Branch Instructions



Figure 4.16 Unsigned Branch Instruction

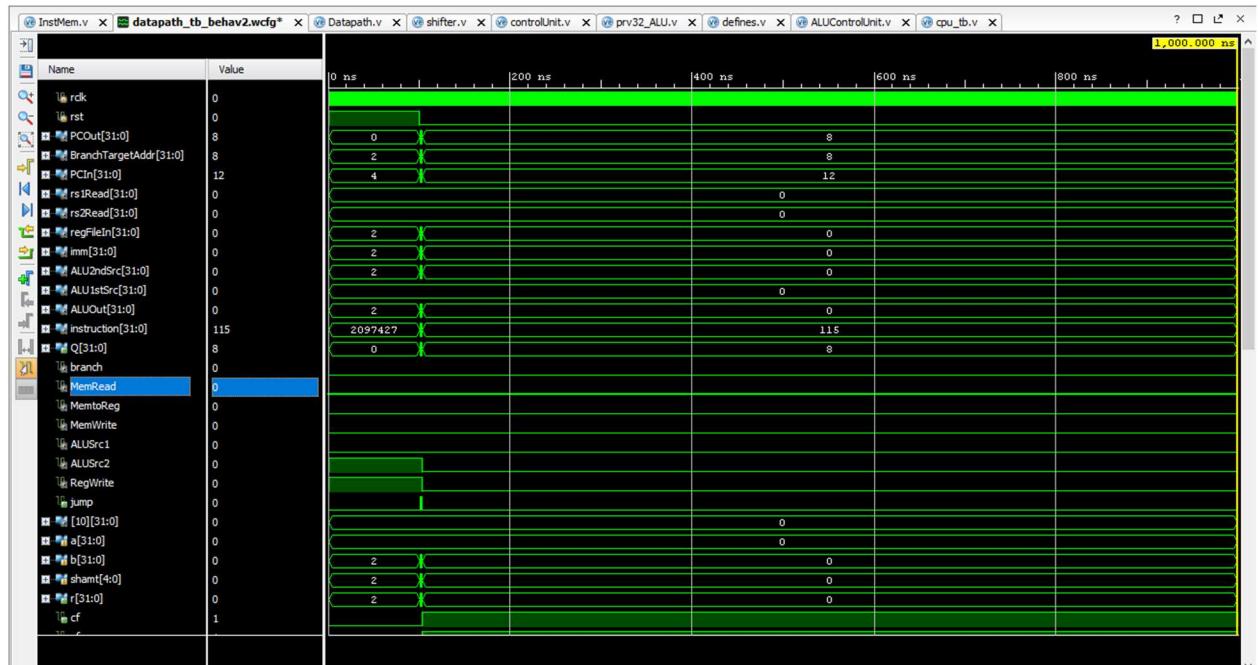


Figure 4.17 JAL/JALR Instructions

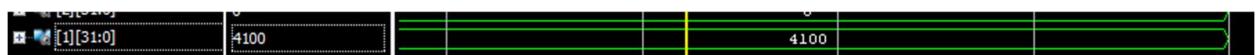


Figure 4.18 AUIPC Instruction



Figure 4.19 All Load Instruction

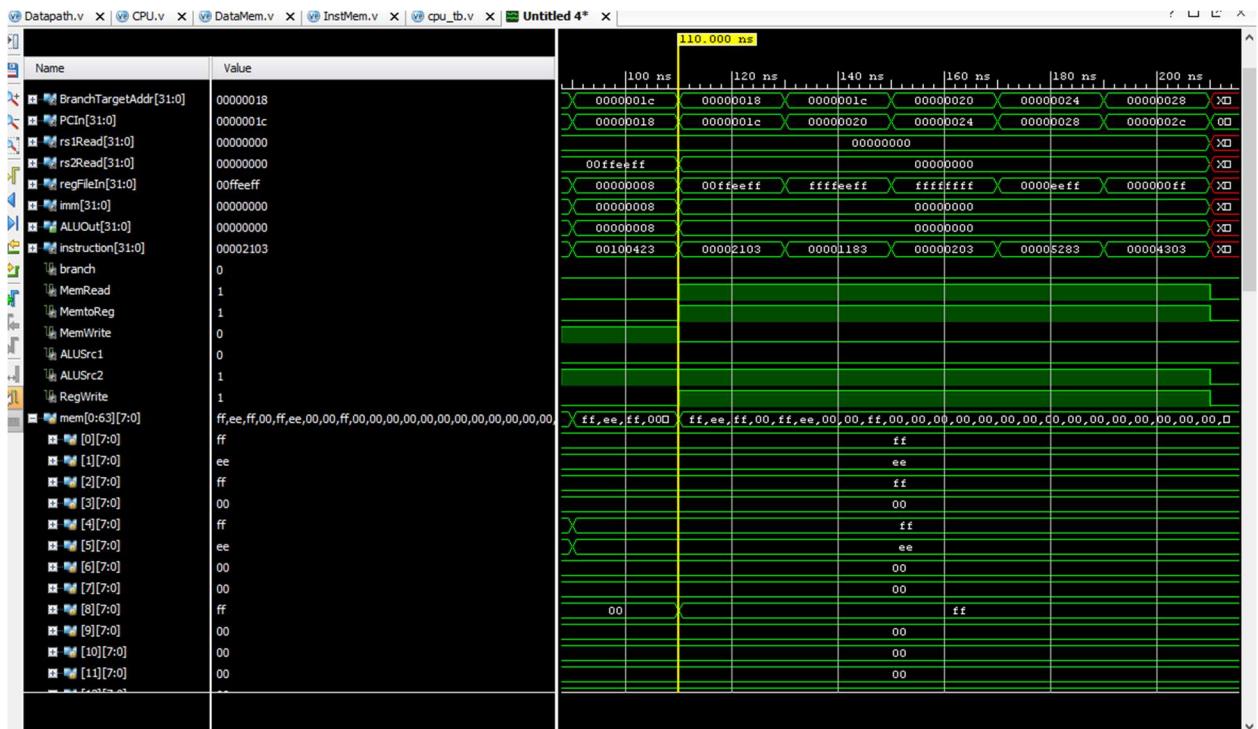


Figure 4.20 All Store Instruction

## 6- List of Figure, Tools, and References Used

- Comprehensive Testing suite link:  
<https://github.com/TheThirdOne/rars/tree/master/test/riscv-tests>
- Online Assembler: <https://www.kvakil.me/venus/>
- RISC-V specifications document : <https://riscv.org/technical/specifications/>