Milestone 2: Risc-V Processor

Computer Architecture

Prof. Cherif Salama

Hussein Elazhary 900200733

Eslam Tawfik 900215295

Milestone Breakdown

In Milestone 2 of the project, our focus will be on implementing the 40 user-level instructions specified in the RISC-V Instruction Set Manual, with exceptions for the ECALL, EBREAK and FENCE instructions. Specifically, ECALL will function as a halt instruction, stopping program execution, while EBREAK and FENCE instructions will be treated as no-operations. This phase requires us to construct a single-cycle datapath, which is a foundational step toward building a more complex, pipelined CPU architecture in subsequent milestones supporting those 40 instructions. Additionally, we'll conduct preliminary testing on these instructions to ensure their correct implementation. This will remain relatively brief for this stage, setting the stage for more extensive validation in later phases.

Implemented and Tested Instructions

Load Instructions: LB, LH, LW, LBU, LHU. These instructions are essential for reading data from memory into registers, with variations handling sign extension and different data sizes.

Store Instructions: SB, SH, SW. They complement the load instructions by handling data writing from registers back to memory, accommodating different data sizes.

Immediate Arithmetic Instructions: ADDI, SLTI, SLTIU, XORI, ORI, ANDI. These instructions perform arithmetic operations directly with an immediate value, supporting addition, bitwise operations, and set less than operations.

Immediate Shift Instructions: SLLI, SRLI. These instructions handle logical shifts to the left and right, manipulating data bits with immediates.

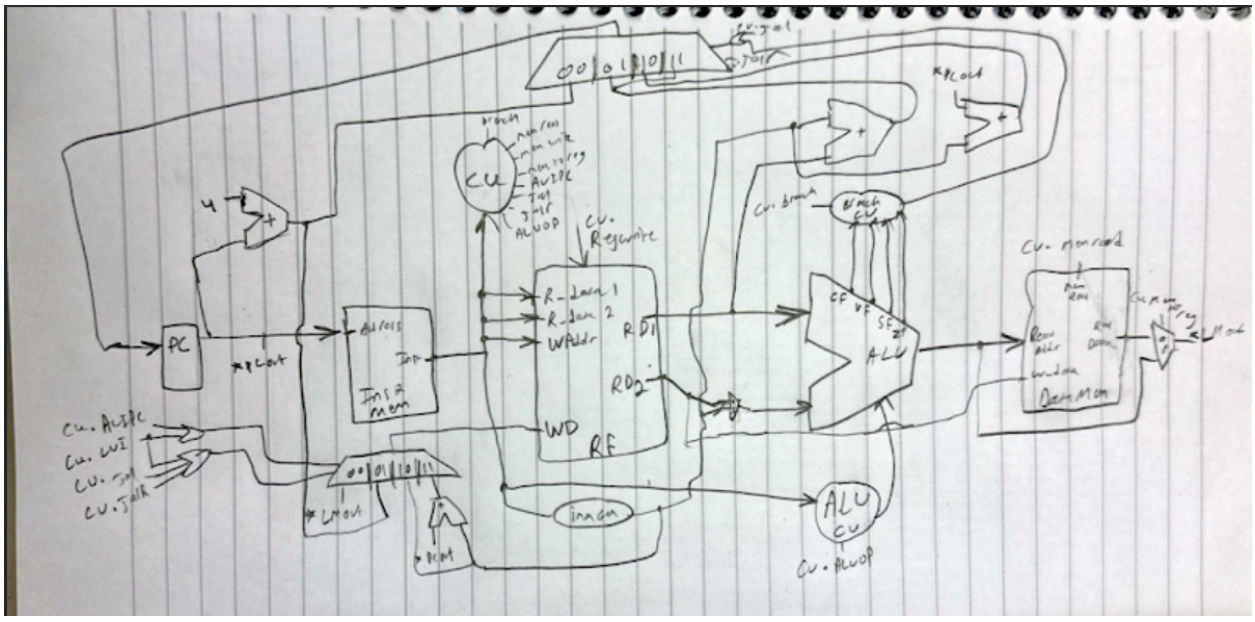
Register Arithmetic Instructions: ADD, SUB, XOR, OR, AND. Focusing on register-to-register arithmetic, these instructions allow for addition, subtraction, and various bitwise operations using data stored in registers.

Shift Instructions: SRL, SRA. Perform logical and arithmetic right shifts, manipulating data bits based on values stored in registers.

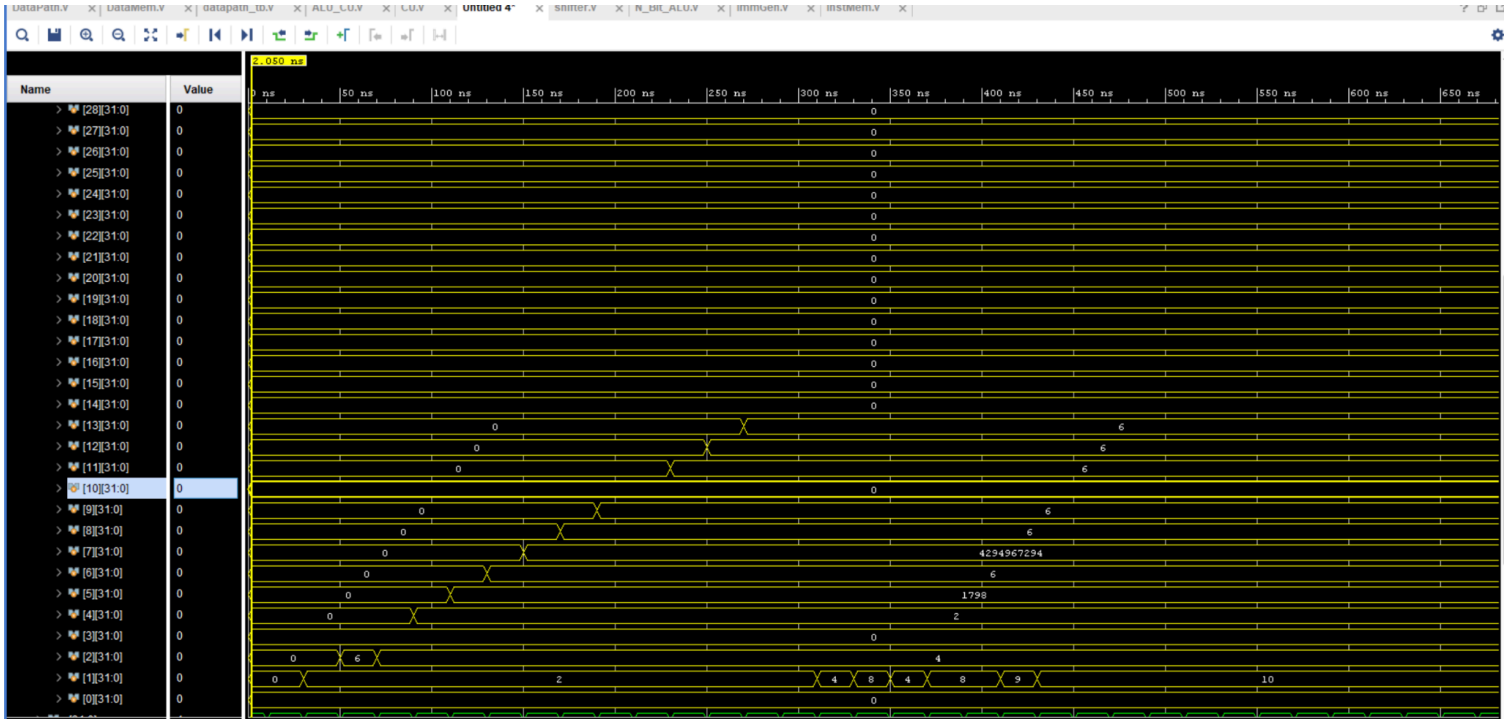
Branch Instruction: BEQ. Control flow instruction, allowing for conditional branching based on the equality of two register values which also allows for looping programs (previously tested).

Although this set of instructions represents a significant portion of the RV32I base integer instruction set, it's important to note that additional instructions are already in development. Due to the deadline, we've prioritized testing and support for these instructions, ensuring a solid foundation upon which we can build, with the aim to cover the entire instruction set and additional features in the project's next phases.

Datapath Sketch



Test Simulation Results



The following simulation results depict the results for Test Program 1 which show the implemented instructions functioning properly given simple values for easy debugging and tracing for errors, but there was a slight issue with negative numbers in the ADDI instruction which is currently being fixed in addition to supporting the remaining instructions.