

CSCI 6461 Computer System Architecture

Project Part 2 User Guide

Team 5

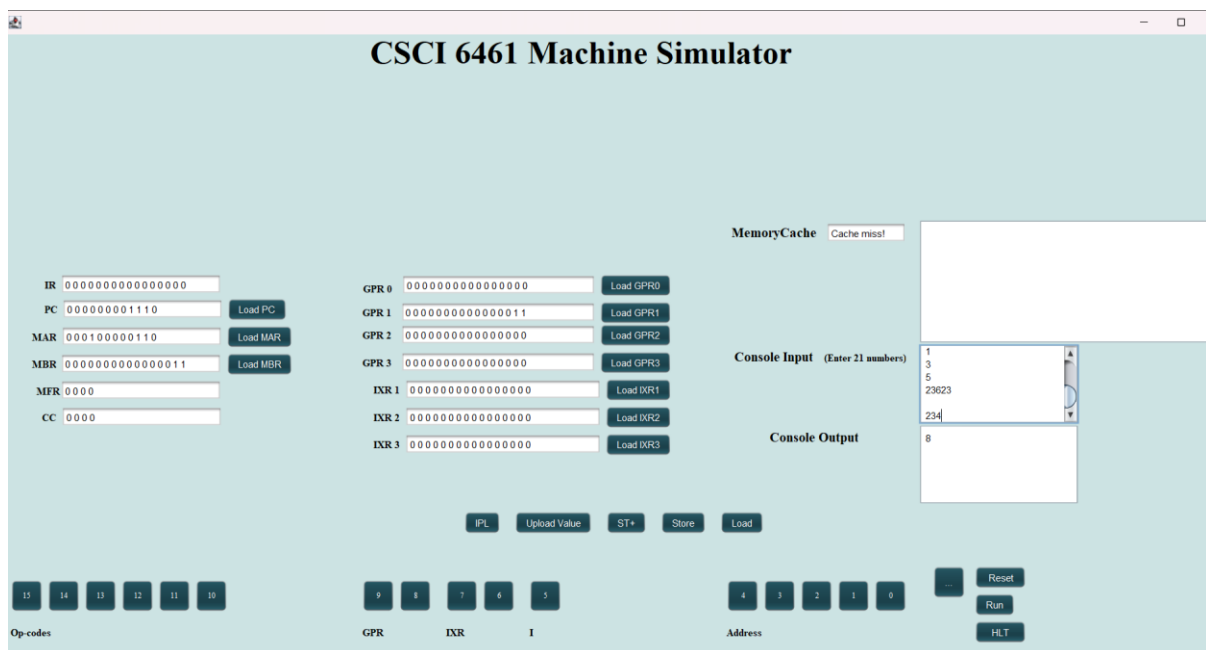
- Jaiswal Nitish
- Jadhav Yash
- Jiang Xuechun

Running the jar file

- Download the zip file and extract the zip file.
- You will find the “csaproject2.jar”.
- Run the jar file by running the command
java -jar csaproject2.jar

Operating Simulator

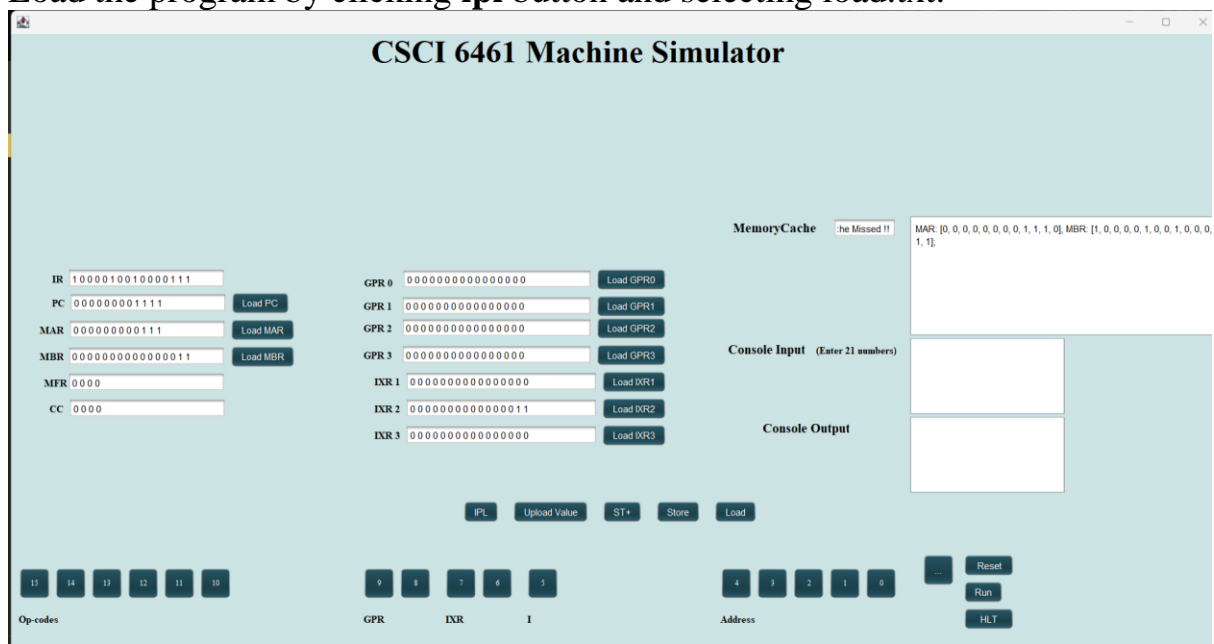
1. Click on ipl button and select demo.txt.



2. Before clicking on program 1, enter 21 numbers within the range [0, 65536) in the Keyboard Console, using ENTER to enter the number and put the next number. The last number you enter will be the search number we intend to find. If no numbers are provided, default values will be used.



- Load the program by clicking **ipl** button and selecting load.txt.



- Click the Run button to execute the program all at once. If you prefer to run the program step by step, click on the SS button.

The screenshot displays the CSCI 6461 Machine Simulator interface, which is designed to emulate a custom instruction set architecture (ISA) for educational purposes.

CSCI 6461 Machine Simulator

The interface is organized into several functional areas:

- Registers and Control:** Located at the top left, this section includes input fields and control buttons for:
 - IR (Instruction Register):** Displays the current instruction being executed.
 - PC (Program Counter):** Shows the address of the next instruction to be fetched.
 - MAR (Memory Address Register):** Holds the address of memory currently being accessed.
 - MBR (Memory Buffer Register):** Stores the data retrieved from or stored to memory.
 - MFR (Machine Frame Register):** A register used for stateful operations.
 - CC (Condition Code):** Indicates the result of the most recent operation (e.g., zero, carry).
- General Purpose Registers (GPRs):** Four registers (GPR 0-3) are shown, each with an 8-bit value field and a "Load GPR*x*" button to load data from memory.
- Dedicated Registers (DXRs):** Three registers (DXR 1-3) are provided, each with an 8-bit value field and a "Load DXR*x*" button.
- Memory Cache:** A section titled "Memory Cache" with a status indicator ("The Missed!!") and a detailed log showing the sequence of memory accesses (MAR, MBR) and their corresponding values.
- Console Input/Output:** Two large text boxes labeled "Console Input (Enter 21 numbers)" and "Console Output" for user interaction and program results.
- Execution Controls:** A row of buttons at the bottom center allows users to:
 - IPL (Initial Program Load):** Starts the simulation.
 - Upload Value:** Loads a specific value into a selected register.
 - ST+ (Store):** Stores the value in the selected register to memory.
 - Load:** Loads a value from memory into the selected register.
- Op-codes:** A row of buttons at the bottom left represents different machine instructions (e.g., 15, 14, 13, 12, 11, 10).
- Address Selection:** A row of buttons below the Op-codes allows selecting the target register (GPR, DXR, I) and the specific register number (9, 8, 7, 6, 5).
- Run Controls:** At the bottom right, there are buttons for:
 - Reset:** Resets the simulator state.
 - Run:** Executes the next instruction.
 - HLT:** Halts the execution of the program.