**CSCI 6461 Fall Semester 2021**

**Project 2 (Team 5)**

**Computer Architecture Simulator**

**Lakshmi Kesava Reddy Indla**

**Indu Goel**

**Humaira Riaz**

**Chengzheng Ye**

**II: Memory and Cache Design**

**Requirement analysis:**

Design and implement the modules for enhanced memory and cache operations.

Implement all instructions except for

* CHK and
* Floating Point/Vector operations

Extend the user interface.

Demonstrate 1st program running on your simulator.

**Goal:**

Goals of this project-II are as follows:

Design and implement enhanced memory and cache operations.

Implement all instructions.

Program\_1 take the input and outputs the target number.

**Design Notes:**

**2. Memory and Cache Design**

1. In the second phase of our CA Simulator, we implemented FIFO (Associative cache mapping) with 64 words and 16 tags. One tag for each block and 4 words in each block.
2. When the system powers up, all elements of memory sets to zero.

**Transfer Instructions:**

In order to change the control of program execution transfer instructions are used. They have the same format as the Load/store instructions.

The implemented transfer instructions are JZ, JNE, JCC, JMA, JSR, RFS (Immediate), SOB and JGE.

Conditional transfer instructions test the value of a register. Note R = 0…3.

To run transfer instructions run the txt file by name **ipl - JumpIns.txt**

**Arithmetic and Logical Instructions**:

The arithmetic and logical instructions are responsible to perform most of the computations. The arithmetic instructions define the set of operations performed by (ALU). These instructions can be further classified into binary, decimal, logical, shift/rotate, and bit/byte manipulation instructions. The instruction which are implemented in our machine simulator are AMR, SMR, AIR, and SIR.

Moreover, there are certain arithmetic and logical instructions which are register to register operations, they are also implemented. These are MLT, DVD, TRR, AND, ORR, SRC, RRC and logical not of register (NOT).

To run transfer instructions run the txt file by name **ipl-Arithematic.txt**

**I/O Operations:**

As required in the phase 2 the machine takes the input from the user, places it in the register from the input device.

In our simulator we used a text input field instead of console keyboard, in which the user is going to input data. The devid for this input text field is 0.

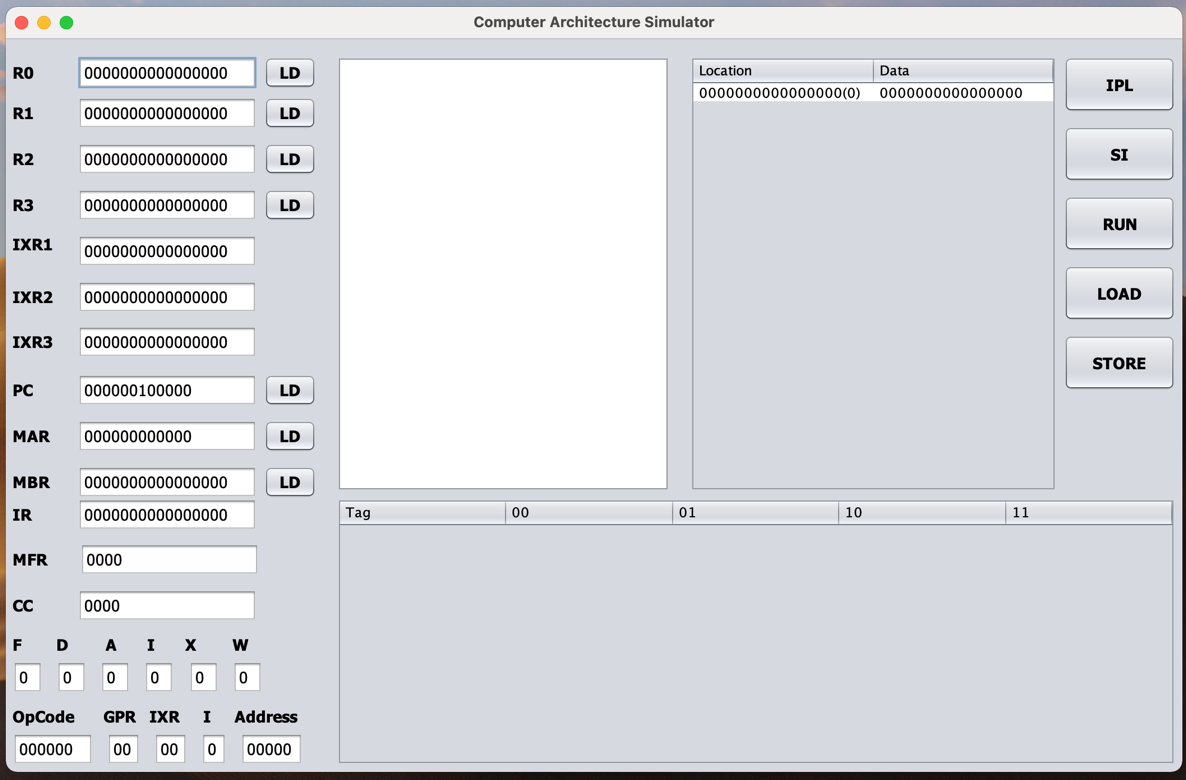
For the output, a separate window pops up, that outputs the character to device from the register.

To run transfer instructions run the txt file by name **IPL\_1.txt**

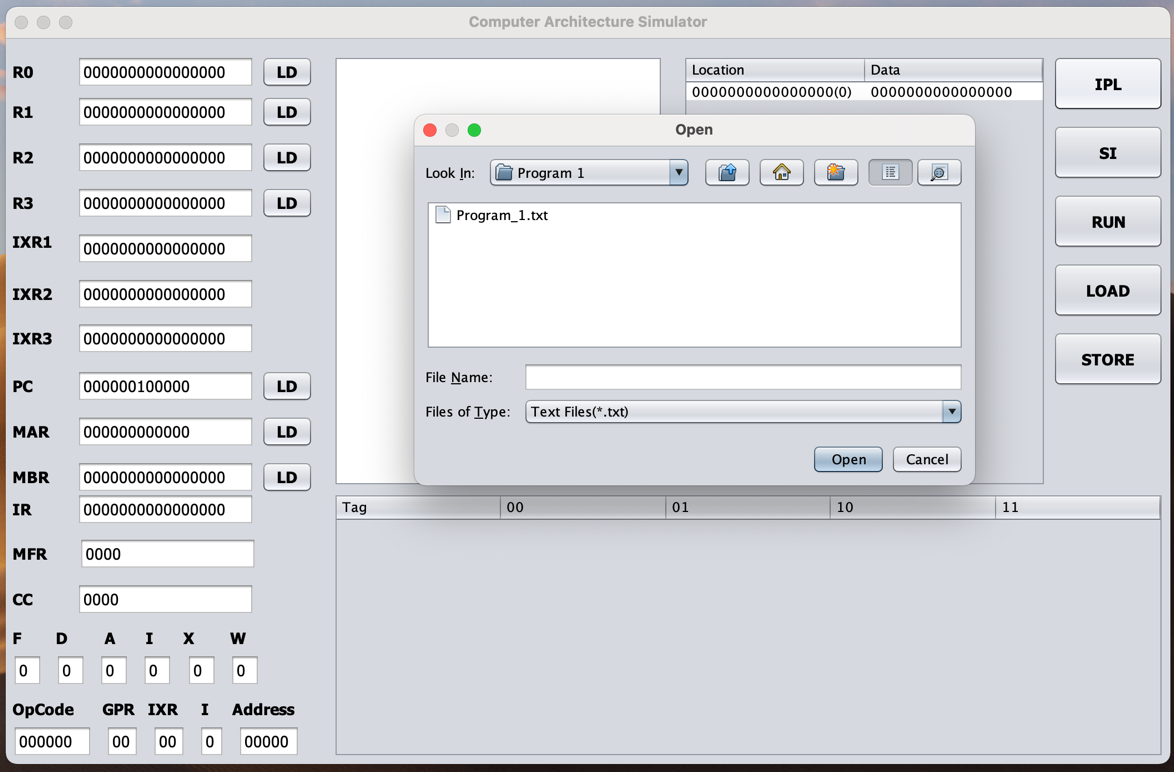
* 1. **Test Program 1**

**The program logic file has been added to the corresponding CA Simulator folder.**

1. Click on the IPL button in CA simulator to initialize the machine.

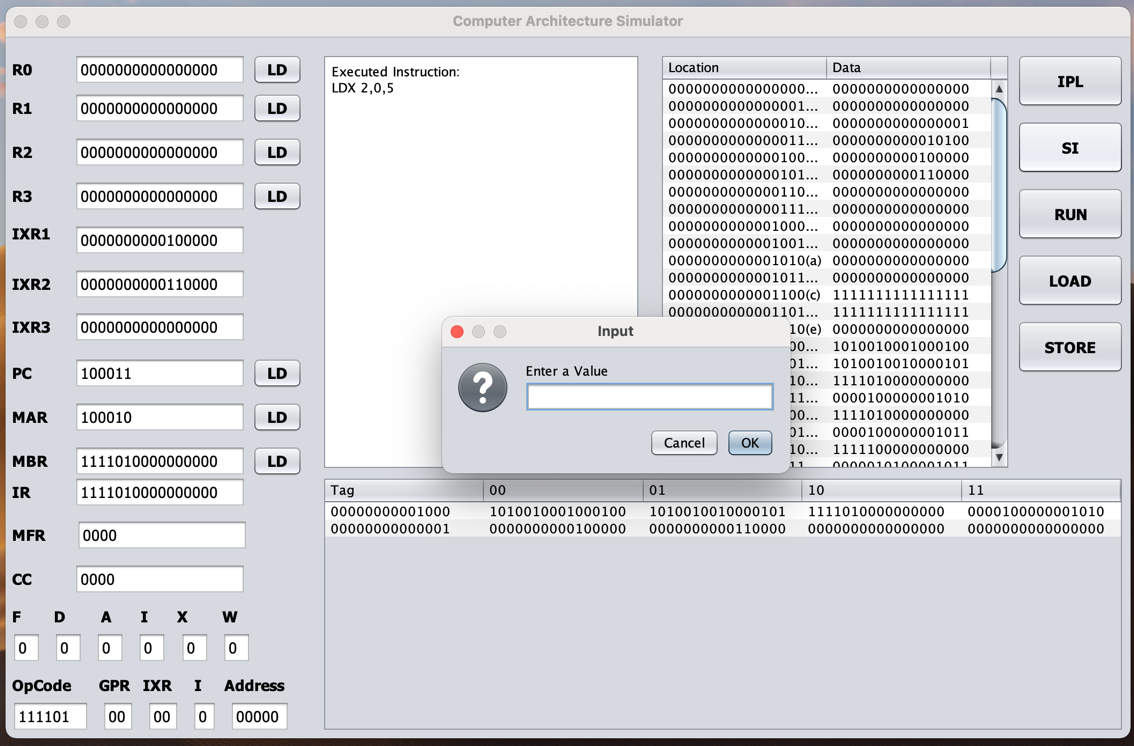
****

1. Then browse the Program\_1.txt file saved on the desktop and open it.

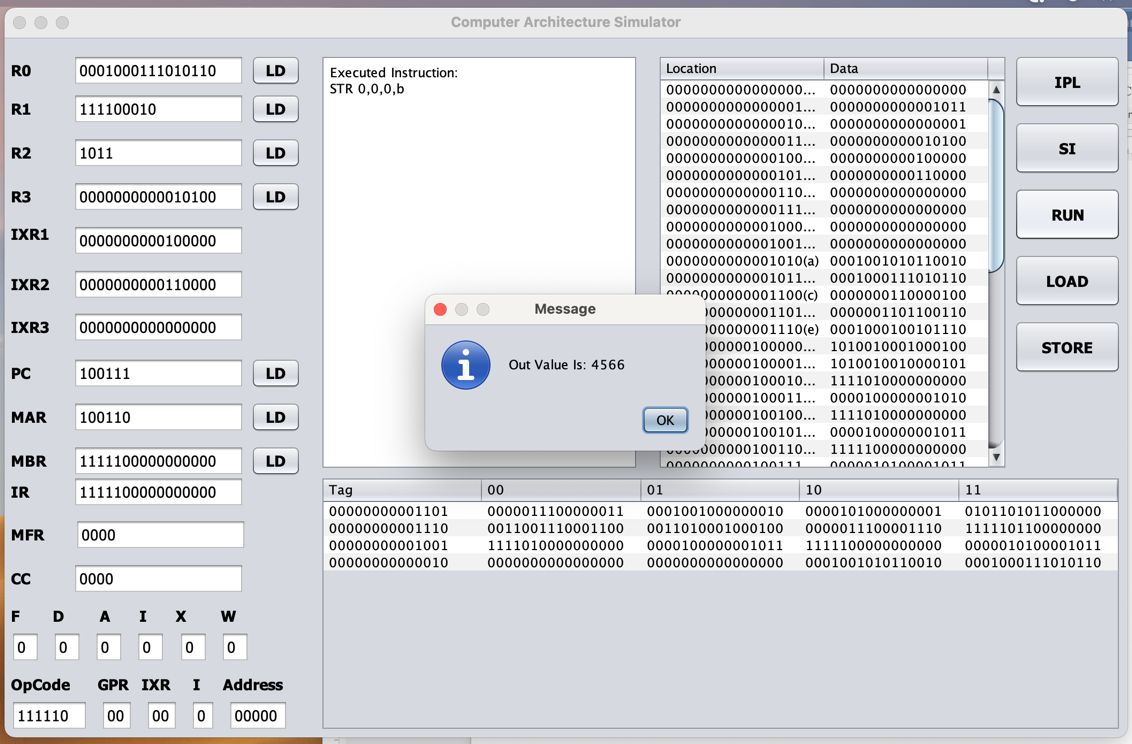
****

1. Click the SI button to run the instruction by single instruction.
2. A window is going to pop up to ask user to input the target number.

Input the target number and press ok. This first number added will be the target number. Now enter numbers for 20 times as mentioned in program 1. User can randomly enter numbers between 0 and 65535.



1. After the last number is entered by the user it will output the target number.

****

**Deliverables:**

|  |  |
| --- | --- |
| **Project Requirement** | **What we implement** |
| **Cache Design and Implementation** | Taken care of . |
| **Have all instructions working except Part IV** | Taken care of . |
| **Have your cache design at least coded out if not working** | Taken care of.(working) |
|  |  |