

CISC Simulator User Guide

Rev 3.0

Group 2

Jonathan Pritchett

Ziwei Li

Haorong Xiao

Siyuan Zhang

Table Of Contents

Table Of Contents	2
Intro	3
UI Layout	4
Overriding values	7
Changing Input Mode	7
Operator Console	7
Memory	7
Registers	8
Pipeline	8
Advancing the simulation	8
Loading the Programs	8
Program 1 test result	8
Demonstration of other instructions	10
Program 2 Test result	19

Intro

This document provides a basic guide for a user of the Simulator. As additional features are added and improved in the simulator, this document will be updated as well.

UI Layout

The screenshot displays the CISC Simulator's user interface. At the top left, the 'Input Mode' is set to 'binary'. Below this, there are three tabs: 'Registers', 'Cache', and 'Pipeline', with 'Registers' currently selected. The 'Registers' section lists various registers (R0, R1, R2, R3, X1, X2, X3, PC, CC, IR, MAR, MBR) each with a corresponding hexadecimal value field, all currently showing '0000000000000000'. To the right of the registers, the 'Memory' section shows 'Address' and 'Value' fields, both also containing '0000000000000000'. Below the memory fields are three buttons: 'Load test program', 'Load Program 1', and 'Load Program 2'. Further down, there is a checkbox for 'AutoRun?' which is unchecked, and a 'Step' button. At the bottom left, there is an 'Override All Values' button and a 'Lock Override' checkbox which is unchecked. On the right side, there is a large black rectangular area labeled 'Operator Console'.

Register	Value
R0	0000000000000000
R1	0000000000000000
R2	0000000000000000
R3	0000000000000000
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000000000
CC	0000000000000000
IR	0000000000000000
MAR	0000000000000000
MBR	0000000000000000

Memory

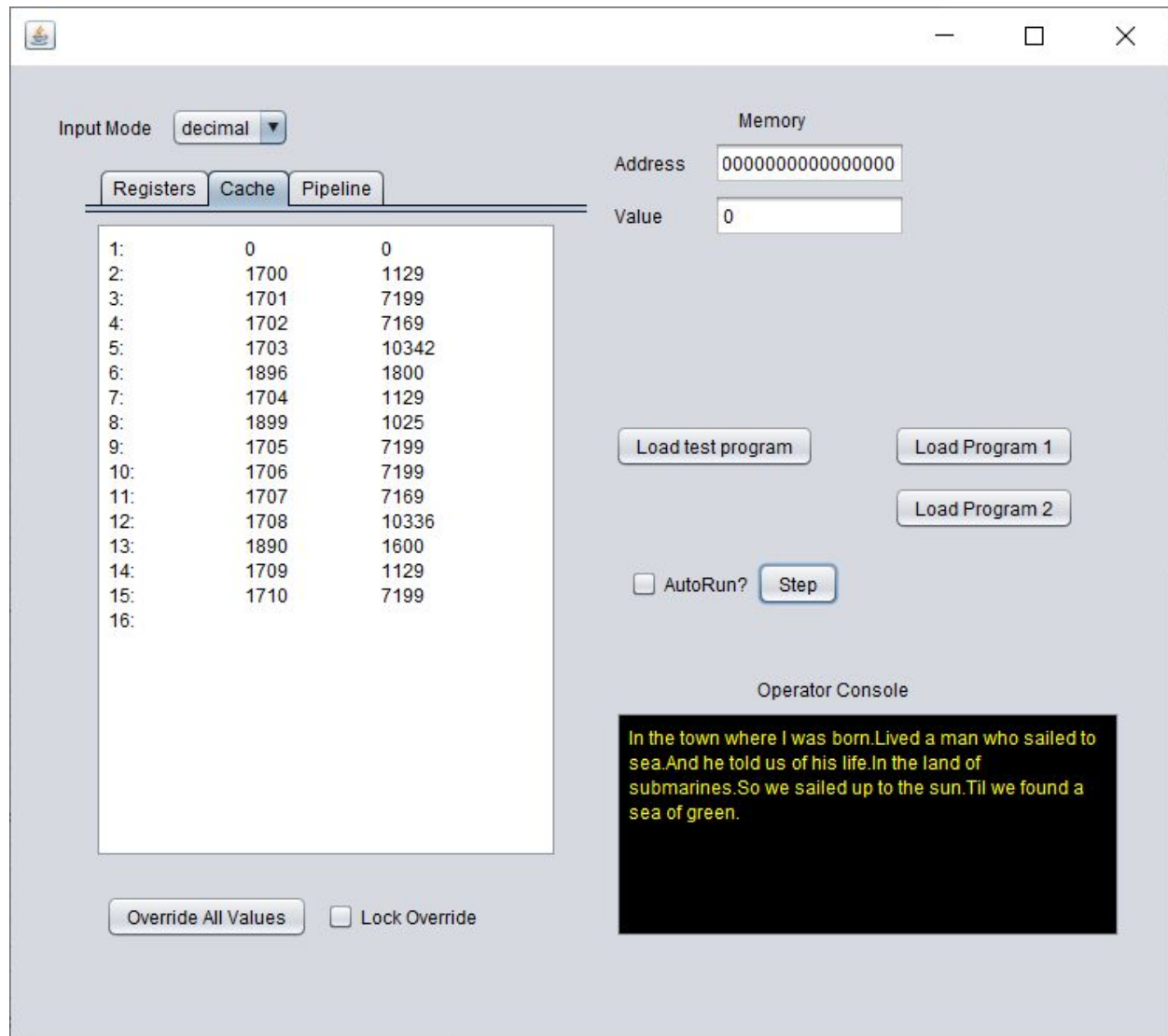
Field	Value
Address	0000000000000000
Value	0000000000000000

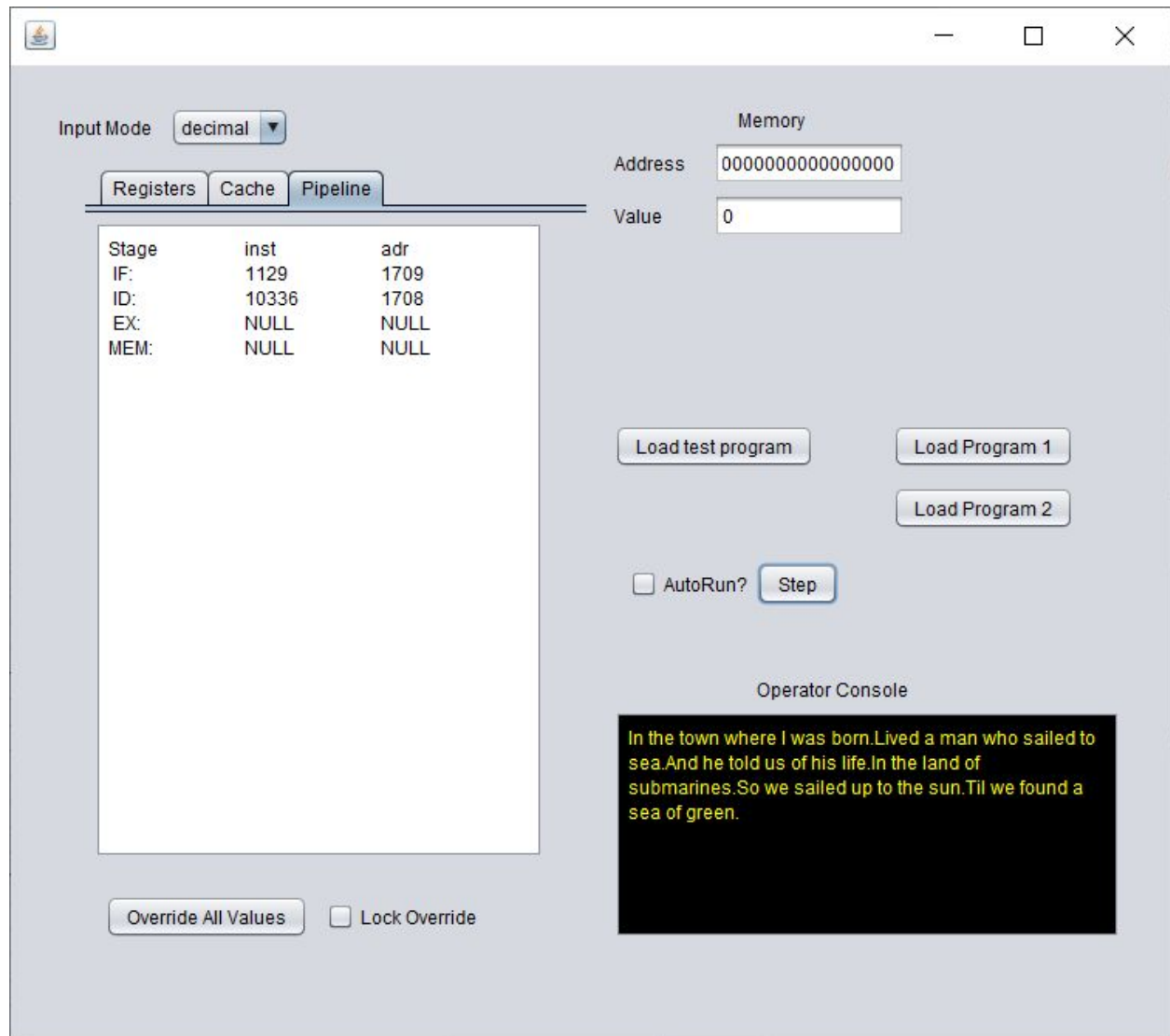
Buttons: Load test program, Load Program 1, Load Program 2

AutoRun? ☐ Step

Operator Console

Override All Values ☐ Lock Override





The screenshot shows the CISC Simulator User Interface. At the top left, there is a window title bar with a small icon and standard window controls (minimize, maximize, close). Below the title bar, the main interface is divided into several sections. On the left, there is a section for 'Input Mode' with a dropdown menu set to 'decimal'. Below this, there are three tabs: 'Registers', 'Cache', and 'Pipeline'. The 'Pipeline' tab is currently selected. Under the 'Pipeline' tab, there is a table showing the current state of the pipeline stages. The table has three columns: 'Stage', 'inst', and 'adr'. The rows are labeled 'IF:', 'ID:', 'EX:', and 'MEM:'. The values are: IF: 1129, 1709; ID: 10336, 1708; EX: NULL, NULL; MEM: NULL, NULL. To the right of the pipeline table, there is a 'Memory' section with two input fields: 'Address' (containing '0000000000000000') and 'Value' (containing '0'). Below the memory section, there are three buttons: 'Load test program', 'Load Program 1', and 'Load Program 2'. Further down, there is a checkbox labeled 'AutoRun?' which is currently unchecked, and a 'Step' button. At the bottom left, there is a button labeled 'Override All Values' and a checkbox labeled 'Lock Override' which is currently unchecked. On the bottom right, there is a section titled 'Operator Console' which contains a black box with yellow text: 'In the town where I was born. Lived a man who sailed to sea. And he told us of his life. In the land of submarines. So we sailed up to the sun. Til we found a sea of green.'

Input Mode: decimal

Registers Cache Pipeline

Stage	inst	adr
IF:	1129	1709
ID:	10336	1708
EX:	NULL	NULL
MEM:	NULL	NULL

Memory

Address: 0000000000000000

Value: 0

Load test program Load Program 1 Load Program 2

☐ AutoRun? Step

Operator Console

In the town where I was born. Lived a man who sailed to sea. And he told us of his life. In the land of submarines. So we sailed up to the sun. Til we found a sea of green.

Override All Values ☐ Lock Override

Overriding values

All of the displayed register and memory values may be freely edited by the user if the Lock override button is unchecked. Pressing the “Override All Values” buttons will force whatever values are currently in each of the fields into the appropriate register or memory address. Values entered into fields should be 16bit binary numbers.

***IMPORTANT NOTE:** Currently, these fields are unformatted, and will accept any string as input. In the future, these will be restricted to only accept binary or decimal values, but this has not been implemented yet.

Changing Input Mode

The Input Mode dropdown allows the user to select between entering (and viewing) values in Binary or Decimal. Note that this does not affect the operator console (see below).

Operator Console

The operator console allows the user to provide input and view output for the program running in the simulator. Any text output from the program will be displayed here and any input the program requires should be typed here as well.

Memory

All 2048 memory addresses may be accessed by using the memory fields. Entering a 16-bit binary address into the “Address” field and pressing enter will populate the “Value” field with the current 16-bit data stored at that address. Manually entering a 16-binary number into the “Value” field and pressing the “Override” button will force that number into the current address indicated by the “Address” field.

Clicking on the “Cache” tab will bring up a view of the memory cache. The first column indicates the cache-lines position, the second indicates the cached memory address, and the third indicates the data stored in the cache/memory.

Registers

Each of the available registers for the simulator are displayed along the left side of the panel. Each one has a label indicating which register it is, and a text field indicating its current value. Entering a 16-bit binary number into any of these fields and pressing the “Override” button will force that number into the register.

Pipeline

The pipeline used in the simulation can be viewed by clicking on the Pipeline tab. Each of the four stages of our pipeline, “Instruction Fetch (IF)”, “Instruction Decode(ID)”, “Execute Instructing (EX)”, and “Write to Memory (MEM)” are represented as a line of text. Should a stage need to be “flushed”, the value displayed will be “NULL”.

Advancing the simulation

The “Step” button allows the user to run through the simulation one cycle at a time. If there are non-zero values in the PC or Instruction registers, clicking the “Step” button will cause the simulator to execute current instructions, advance the PC register, and fetch the next instruction. Checking the “Auto-run” box will cause the simulation to run continuously until there are no more instructions to execute. Currently, Auto-run is set to advance at a rate of 1 cycle per second (this will be user-configurable in the future).

Loading the Programs

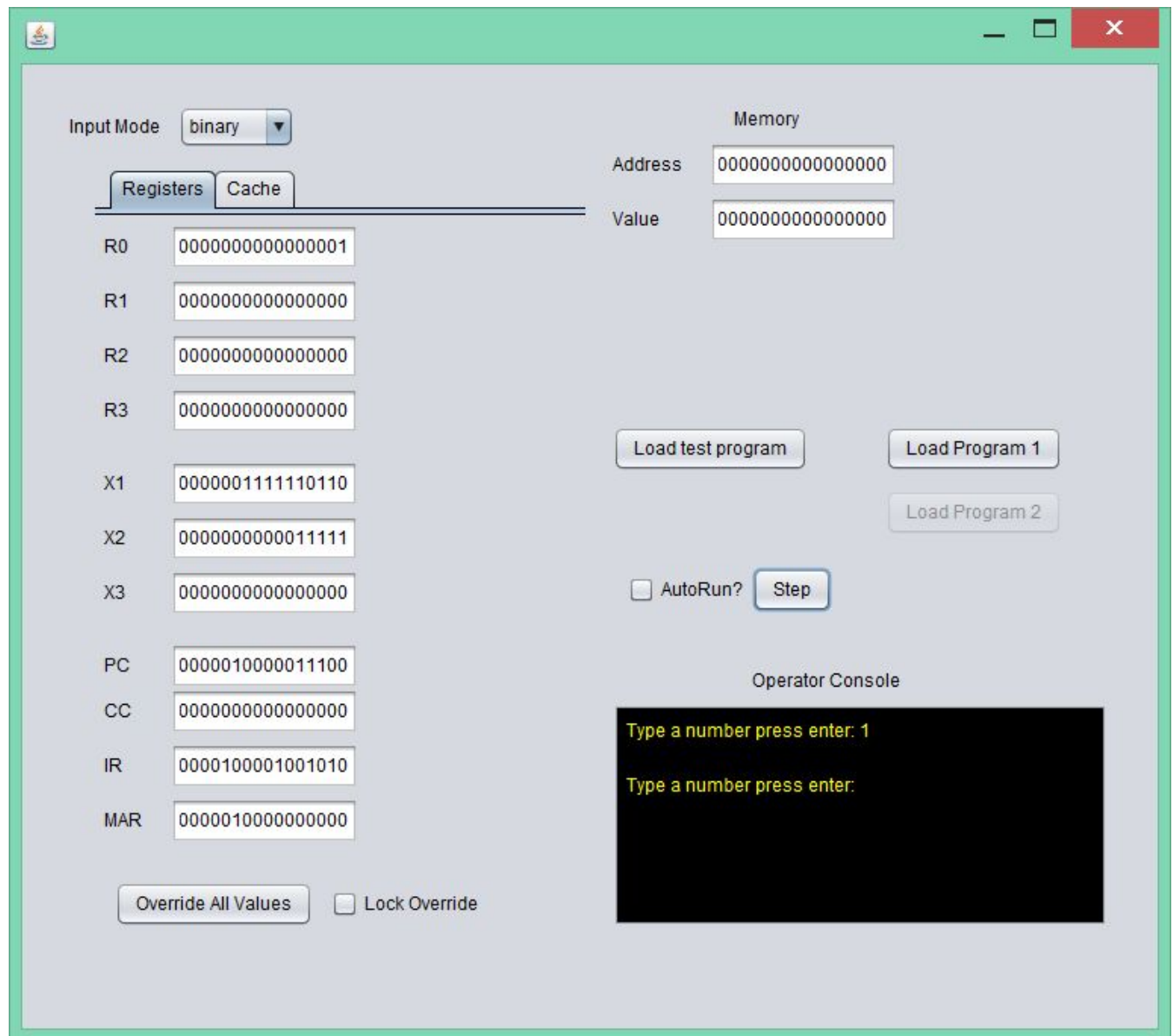
Pressing the “Load Test Program” will load a hard-coded program into the simulator’s memory. The “Step” and “Auto-run” buttons may then be used to run this test program. Likewise, clicking “Load Program 1” will load Program 1 into memory. Program 2 is currently unimplemented, and its button disabled.

Program 1 test result

This program takes 21 integer numbers from the user. Then, it will find the nearest number of those 20 numbers to the 21st number.

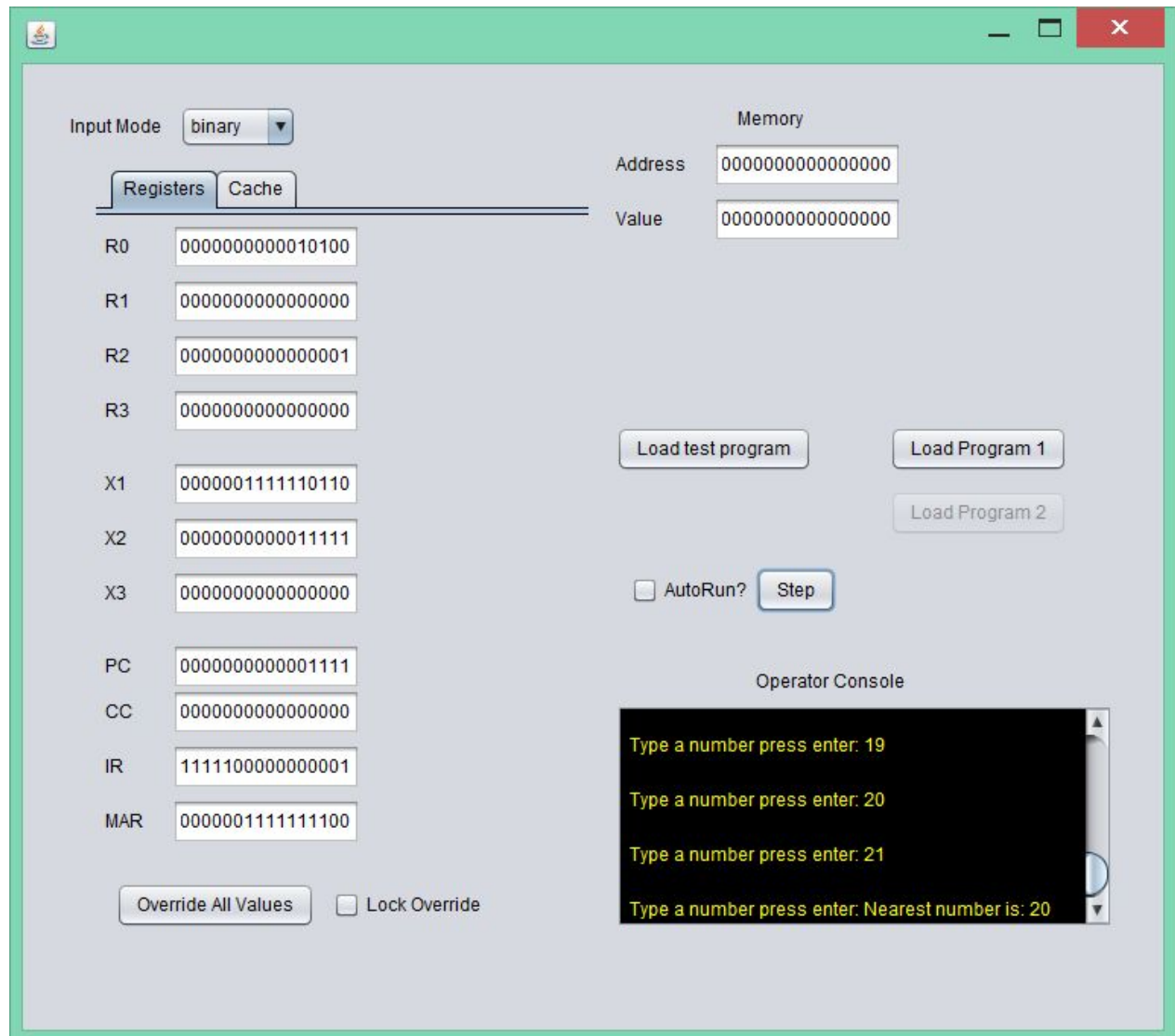
First of all, press the 'Load Program 1' button. The simulator will write the program into the memory.

To input numbers, user has to type the number and press Enter on the keyboard. Then, press the 'Step' twice to let the simulator run the instruction IN and store the number in the memory.



As we can see in the screenshot, the input number has been stored in R0 and it has been stored in memory. After having input 21 numbers, user can press 'AutoRun' checkbox to let simulator run automatically or press 'Step' button to check every step of this program.

This time, we input 1 to 20 in the simulator and the 21st number is 21. We do this is just in case of overflow and underflow situations. Here is the result of this program:



Demonstration of other instructions

Some instructions we did not use in above program. Therefore, we have some screenshots here to show that they work.

Jump if Zero:

The register R0 equals to 0. Then, Program counter is set to '1111'.

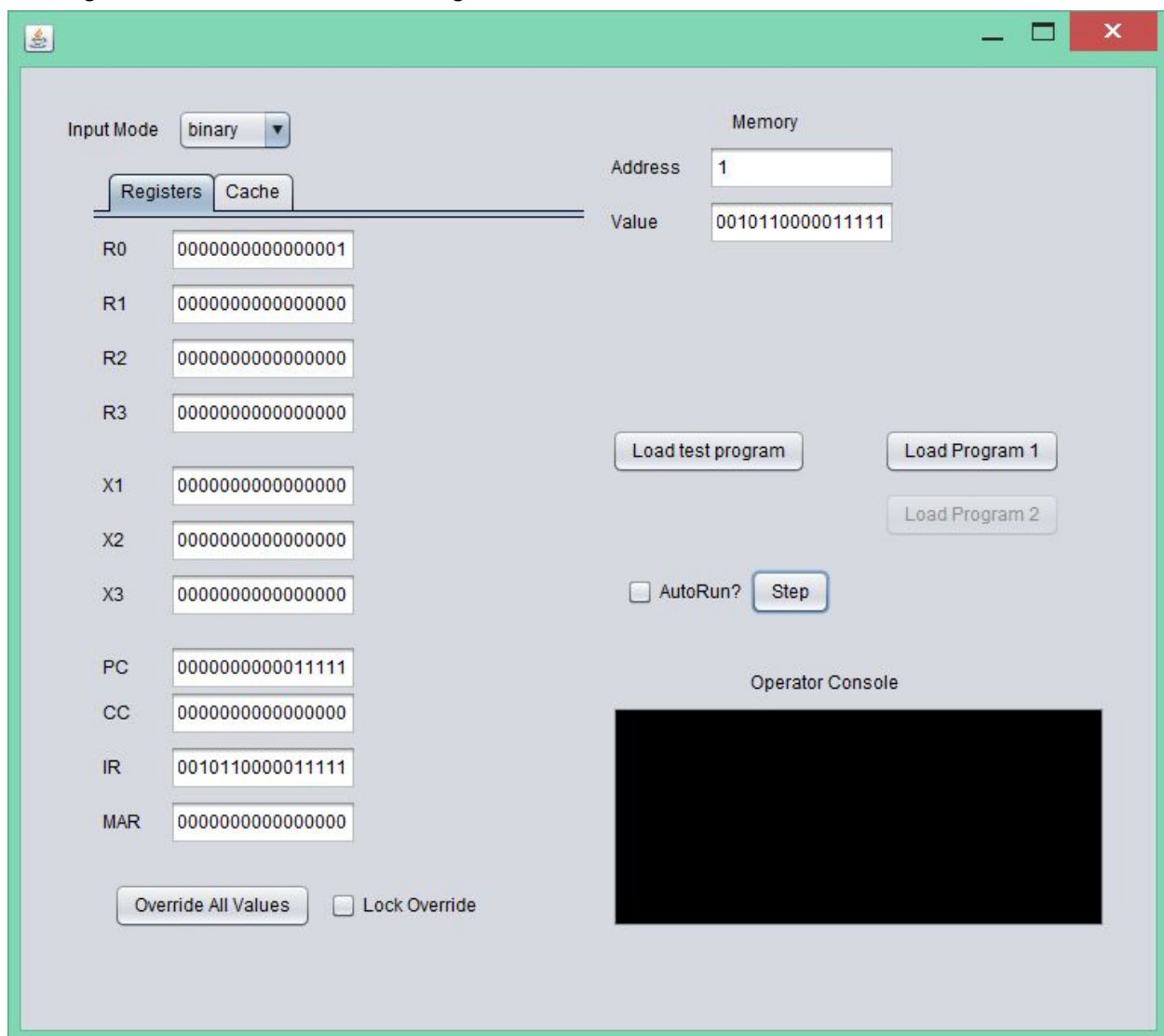
The screenshot displays the CISC Simulator interface. On the left, the 'Registers' tab is active, showing a list of registers: R0, R1, R2, R3, X1, X2, X3, PC, CC, IR, and MAR. Each register has a corresponding value field. R0, R1, R2, R3, X1, X2, X3, CC, and MAR all contain '0000000000000000'. The PC (Program Counter) contains '0000000000011111'. The IR (Instruction Register) contains '0010100000011111'. At the bottom left, there is an 'Override All Values' button and a 'Lock Override' checkbox which is unchecked.

On the right, the 'Memory' section shows 'Address' as '1' and 'Value' as '0010100000011111'. Below this are three buttons: 'Load test program', 'Load Program 1', and 'Load Program 2'. Further down is an 'AutoRun?' checkbox which is unchecked, and a 'Step' button.

At the bottom right, the 'Operator Console' is represented by a large black rectangular area.

Jump if not equal:

The register R0 is set to 1. Then, Program counter is set to '11111'.



Jump if condition code:

We set condition code here 0. Therefore, program counter adds 1 to itself.

The screenshot displays the CISC Simulator interface. On the left, the 'Registers' tab is active, showing a list of registers with their current values:

Register	Value
R0	0000000000000000
R1	0000000000000000
R2	0000000000000000
R3	0000000000000000
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000000010
CC	0000000000000000
IR	0011000000011111
MAR	0000000000000000

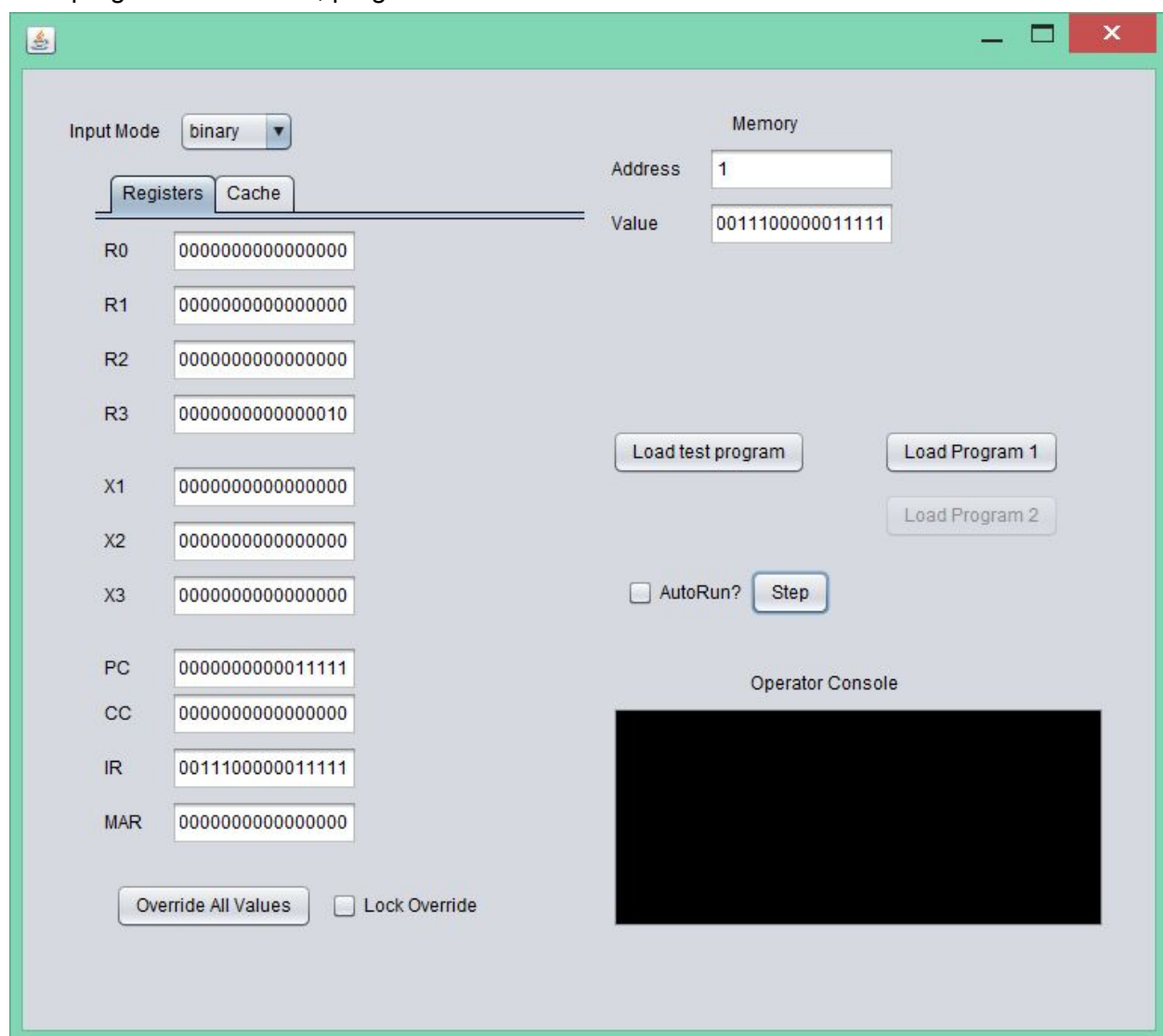
At the bottom of the registers section, there is an 'Override All Values' button and a 'Lock Override' checkbox.

On the right, the 'Memory' section shows the 'Address' as 1 and the 'Value' as 0011000000011111. Below this are buttons for 'Load test program', 'Load Program 1', and 'Load Program 2'. There is also an 'AutoRun?' checkbox and a 'Step' button.

At the bottom right, the 'Operator Console' is represented by a large black rectangular area.

Jump and Save Return Address:

$R3 = \text{program counter} + 1$, program counter = effective address.



The screenshot displays the CISC Simulator User Interface. At the top left, the 'Input Mode' is set to 'binary'. Below this, there are two tabs: 'Registers' and 'Cache'. The 'Registers' tab is active, showing a list of registers with their current values:

Register	Value
R0	0000000000000000
R1	0000000000000000
R2	0000000000000000
R3	0000000000000010
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000011111
CC	0000000000000000
IR	0011100000011111
MAR	0000000000000000

At the top right, the 'Memory' section shows the 'Address' as 1 and the 'Value' as 0011100000011111. Below the memory section, there are three buttons: 'Load test program', 'Load Program 1', and 'Load Program 2'. Further down, there is a checkbox for 'AutoRun?' which is currently unchecked, and a 'Step' button. At the bottom left, there is a button labeled 'Override All Values' and a checkbox for 'Lock Override' which is also unchecked. At the bottom right, there is a section labeled 'Operator Console' which contains a large black rectangular area.

Return From Subroutine with return code as Immed portion:

$R0 = \text{Immed}$, $PC = c(R3)$

The screenshot displays the CISC Simulator interface. On the left, a list of registers (R0, R1, R2, R3, X1, X2, X3, PC, CC, IR, MAR) is shown with their current binary values. The 'Input Mode' is set to 'binary'. The 'Memory' section shows 'Address 1' with a 'Value' of '0011110000011111'. Below the registers, there are buttons for 'Load test program', 'Load Program 1', and 'Load Program 2'. A checkbox for 'AutoRun?' is present, along with a 'Step' button. At the bottom left, there is an 'Override All Values' button and a 'Lock Override' checkbox. The 'Operator Console' area at the bottom right is currently black.

Register	Value
R0	0000000000011111
R1	0000000000000000
R2	0000000000000000
R3	0000000000000010
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000000010
CC	0000000000000000
IR	0011110000011111
MAR	0000000000000000

Input Mode:

Memory: Address: 1, Value: 0011110000011111

Buttons: Load test program, Load Program 1, Load Program 2

AutoRun? ☐ Step

Operator Console

Override All Values Lock Override ☐

AND:

We set R0 = 01, R1 = 10. The result id stored in R0.

The screenshot displays the CISC Simulator interface. On the left, a list of registers (R0, R1, R2, R3, X1, X2, X3, PC, CC, IR, MAR) is shown with their current values in binary. The IR register contains the value 0101110001000000. On the right, the Memory section shows Address 1 with Value 0101110001000000. Below the memory section are buttons for 'Load test program', 'Load Program 1', and 'Load Program 2'. There is also an 'AutoRun?' checkbox and a 'Step' button. At the bottom left, there is an 'Override All Values' button and a 'Lock Override' checkbox. The 'Operator Console' area at the bottom right is currently black.

Register	Value
R0	0000000000000000
R1	0000000000000010
R2	0000000000000000
R3	0000000000000000
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000000010
CC	0000000000000000
IR	0101110001000000
MAR	0000000000000000

Input Mode: binary

Memory Address: 1, Value: 0101110001000000

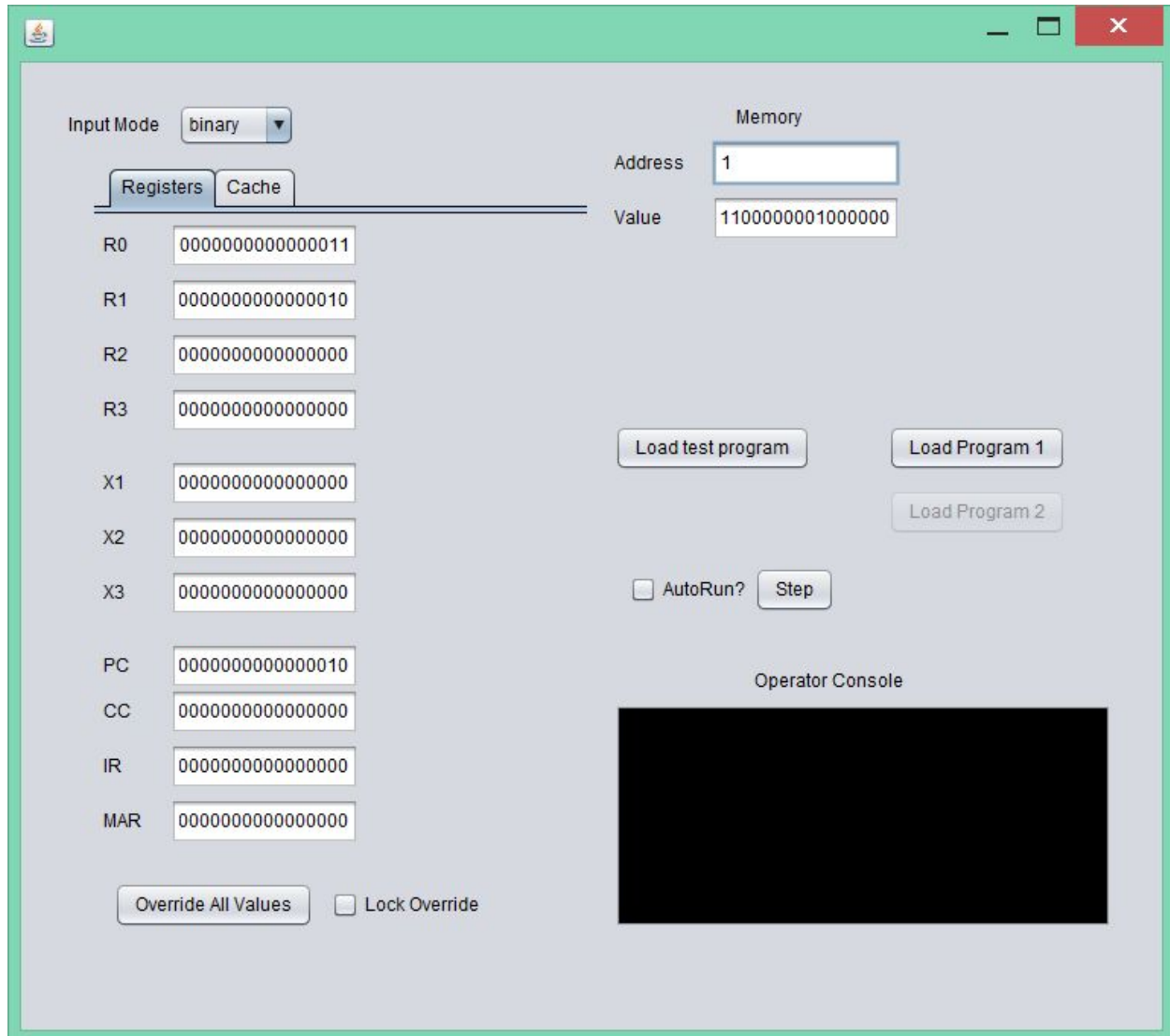
Buttons: Load test program, Load Program 1, Load Program 2, AutoRun? (unchecked), Step

Override All Values, Lock Override (unchecked)

Operator Console

OR:

We set R0 = 01, R1 = 10. The result id stored in R0.



The screenshot displays the CISC Simulator User Interface. At the top left, the 'Input Mode' is set to 'binary'. Below this, there are two tabs: 'Registers' and 'Cache'. The 'Registers' tab is active, showing a list of registers with their current values:

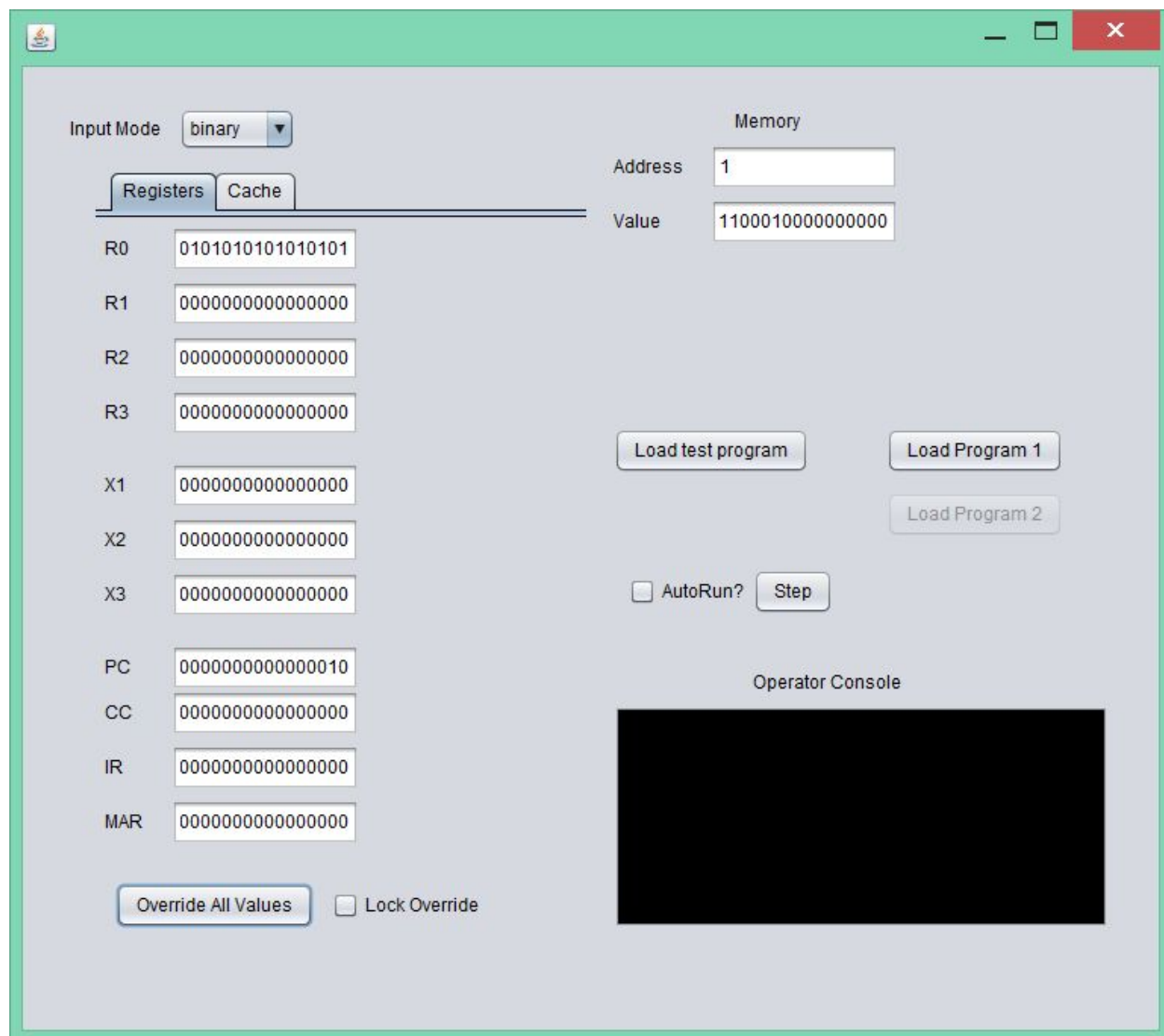
Register	Value
R0	0000000000000011
R1	0000000000000010
R2	0000000000000000
R3	0000000000000000
X1	0000000000000000
X2	0000000000000000
X3	0000000000000000
PC	0000000000000010
CC	0000000000000000
IR	0000000000000000
MAR	0000000000000000

At the bottom left, there is a button labeled 'Override All Values' and a checkbox labeled 'Lock Override' which is currently unchecked.

On the right side, the 'Memory' section shows the 'Address' as '1' and the 'Value' as '1100000001000000'. Below this, there are three buttons: 'Load test program', 'Load Program 1', and 'Load Program 2'. Further down, there is a checkbox labeled 'AutoRun?' which is unchecked, and a button labeled 'Step'.

At the bottom right, there is a section labeled 'Operator Console' which contains a large black rectangular area.

NOT:



Program 2 Test result

This program find the word required by the user in a paragraph with 6 sentences. When it finds the word, it will display the word number in the sentence and the sentence number.

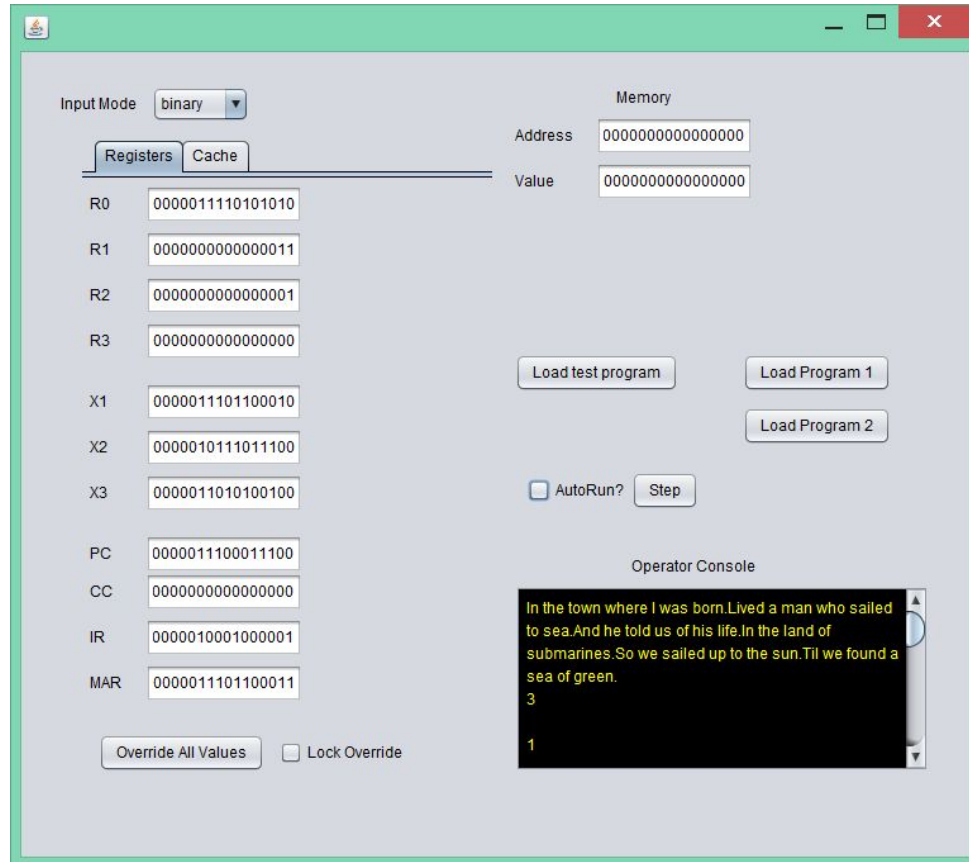
CISC Simulator User Guide

Rev

3.0

First, click the 'Load Program 2' button and a window will show up and asks user to load the file containing the paragraph. To run the program, user can choose autorun or step by step by clicking the checkbox 'AutoRun' or 'Step' button.

Below is a screenshot of the result. The word we are going to find is 'town'. And the input paragraph is 'In the town where I was born.Lived a man who sailed to sea.And he told us of his life.In the land of submarines.So we sailed up to the sun.Til we found a sea of green.' The word 'town' is in the first sentence and the it is the 3rd word in the sentence.



The R1 stores the word number and the R2 stores the line number.