

CSCI6461 - Computer System Architecture

Part - I

Instruction Manual

Our simulator works simply. All the registers (for this initial and relatively simple version of the machine) are given on the user console. The contents of any register can be modified directly by clicking on radio buttons to change individual bits. This means that there is no need to load from a main set of buttons into the register, but instead that all the registers can be directly and individually modified more easily.

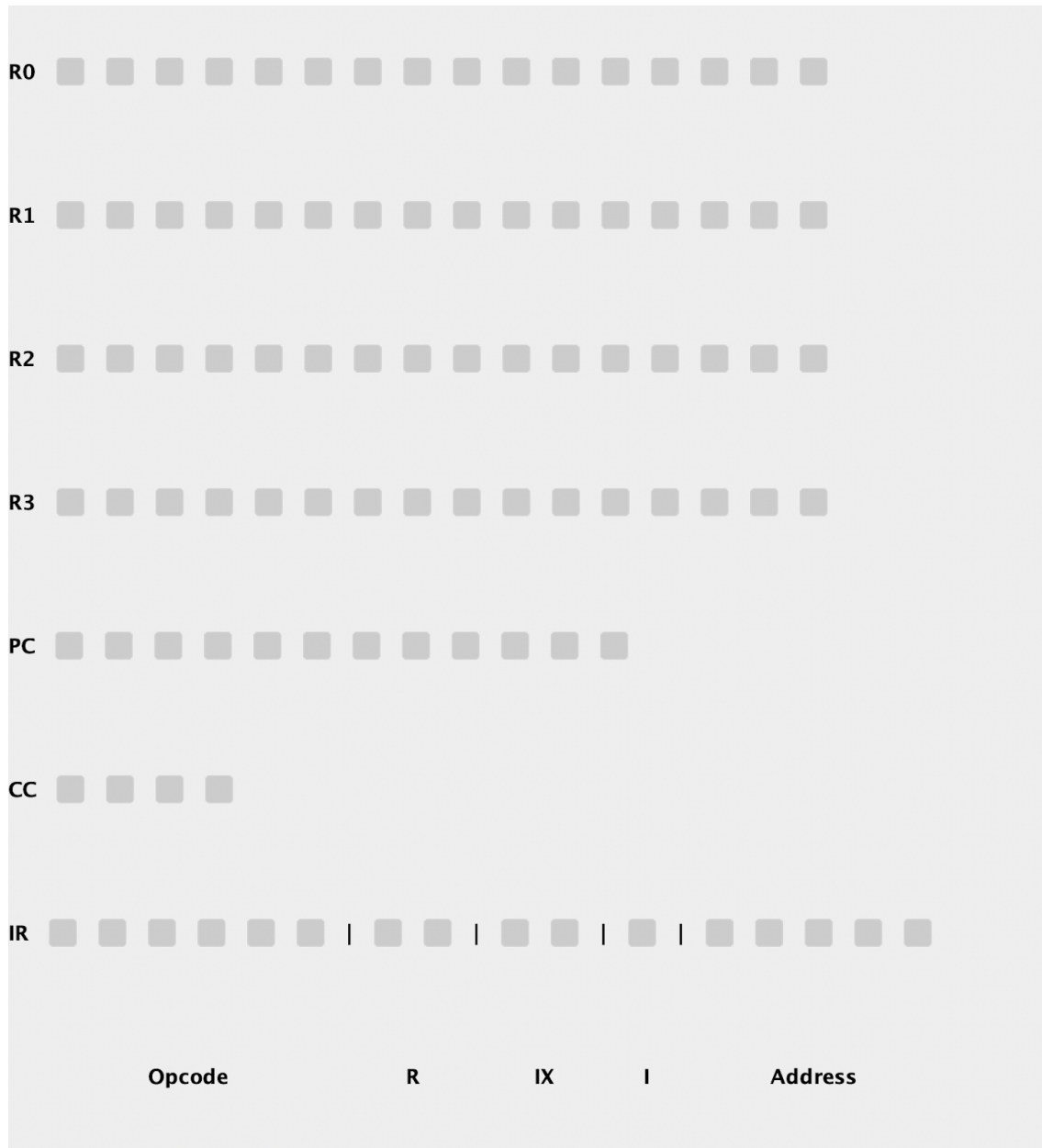
Several buttons are provided with this version of the machine. The “load” and “store” buttons work as specified for the project. The “init” button prompts the user to load a text file which is of the form where each line indicates first in hexadecimal an address and then a space and then in hexadecimal a memory contents. After selecting a file, the contents are loaded at the specified addresses.

Console Layout:

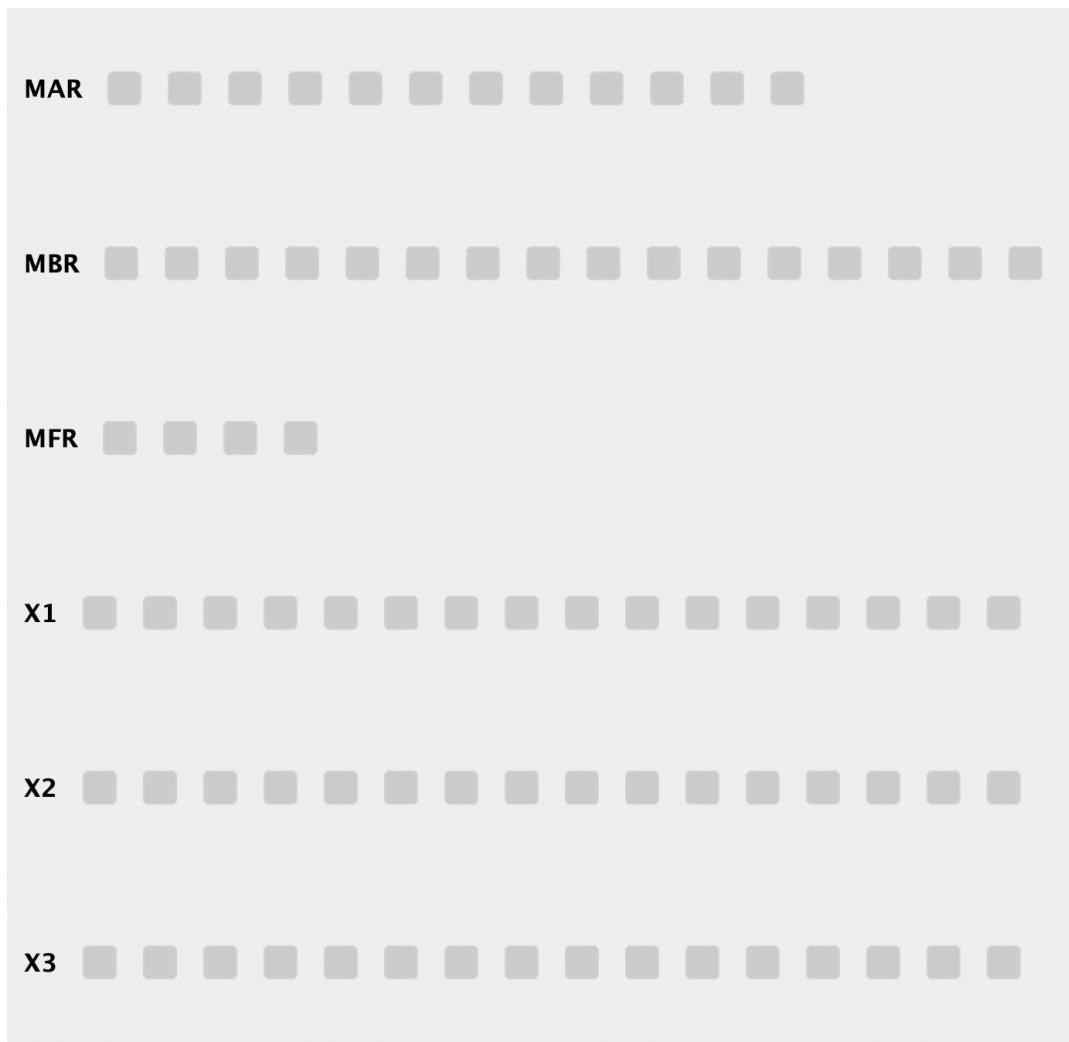
The Console is divided into 3 panels, it contains 4 GPRs (R0-R3), PC, CC, IR, MAR, MBR, MFR and 3 Index Registers (X1-X3). The console further contains radio buttons for these registers depending upon their sizes, which eliminates the need of a load button for every register and makes the process of entering values faster and allows us to utilize the space more efficiently. Lastly, the console also contains Single-step, Load, Store and Init buttons to execute instructions and perform load, store operations onto the memory.

R0	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MAR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
R1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MBR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
R2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	MFR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>																									
R3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
PC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
CC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	X3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>										
IR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>											Single Step	Load	Store	Init																
Opcode						R		IX		I		Address																																					

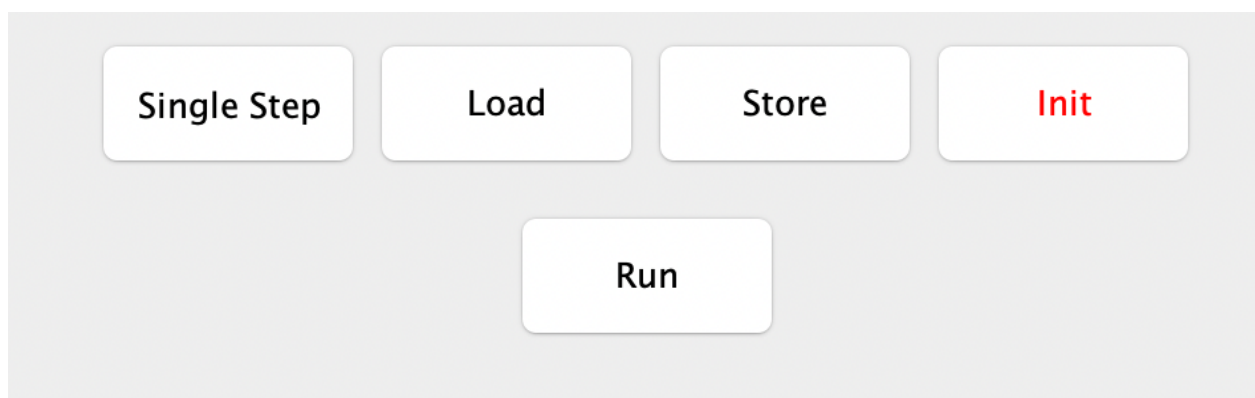
The Register panel contains all general purpose registers, the program counter, the condition code, and the instruction register, where the first 6 bits of the IR is the opcode which can be used to specify instructions, next 2 bits are for selecting the general purpose registers, next 2 bits are for selecting the index registers, next one bit is for toggling between direct and indirect addressing and the last 5 bits are for specifying the address.



The Memory panel contains memory buffer register, memory address register, machine fault register and the index registers.



The buttons panel consists of all the buttons in the frame, it houses single-step, load, store, run and init buttons, for executing instruction, performing loading/storing operations and reading a file to load its contents onto the memory to its corresponding address.



Store Button

- Used to store the value specified in MBR in the address specified in MAR
- Can be used to store instructions

Single Step Button

- Initially address from PC will be copied to MAR and then, PC will be incremented by one
- Next, contents in the address stored in MAR will be copied to MBR
- Then, contents from MBR will be copied to IR
- Now, instruction in the IR will be parsed in `executeInstruction()` function and will be executed based on the `opCode`
- If `opCode=1`, then `ldr` instruction will be executed wherein value from the given address will be copied to `targetRegister` and the display will be updated.
- If `opCode=2`, then `str` instruction will be executed wherein value from the register will be copied to target Address

Design Notes:

The source code is divided into Classes:

- Interface
- InterfaceActionListener
- Register
- RegisterListener
- Main
- Simulator
- Memory
- Utilities

Interface Class:

A Graphical User Interface class for using the simulated computer. This class inherits from the JFrame class, it contains a frame which houses 3 panels each used to house radio-buttons for all registers and buttons for different operations like single-step, load, store, init, halt and run.

InterfaceActionListener Class:

This is a simple action listener class for various buttons of the interface, which can be responded to by calling the corresponding simulator functions.

Register Class:

This is the class to represent the Register which have different names, values which are arrays of zeros and ones and their corresponding sizes which are in number of bits.

Main Class:

The Main class which initializes the simulator and gives the simulator its interface.

Simulator Class:

A class that simulates the operations of the computer. This class is responsible for creating the simulator instance with a memory size, it also contains some important functions like load, store, step, init, executeInstructions, updateRegister, incrementPC, and getRegister.

RegisterListener Class:

A listener class for the register buttons, used to update various register's values and in turn also update the simulator.

Memory Class:

This class is responsible for creating the memory of the simulator.

Utilities Class:

This class contains static functions that are useful for the simulator. It contains functions converting hexa-decimal numbers to binary numbers, binary numbers to decimal numbers and decimal numbers to binary numbers.