

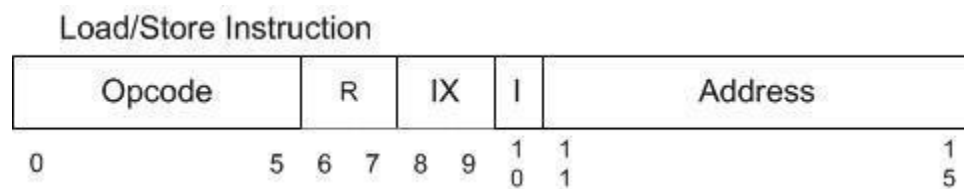
## DESIGN NOTES

### Technology and Tools Used:

1. IDE: Eclipse
2. Language: Java  
Graphic Libraries: Swing (Builder Tool)
3. Repository: Git

The design for this project and this part was determined in part by our present understanding of the Von Neumann Architecture and how the various Load and Store instructions are processed.

**Load/Store instructions Input:** The input for the simulation reflects the Input/Output structure described in the Project Description document.



Opcode:	6 bits	Specifies the instruction
R:	2 bits	Specifies the General-Purpose Register (R0-R3)
IX:	2 bits	Specifies the Index Register (IX0 - IX3)
I:	1 bit	Specifies Indirect Addressing
		If I =1, indirect addressing; otherwise, no indirect addressing.
Address:	5 bits	Specifies the location

**Simulator Window:** displays all the information about the Registers, Indicators and controllers in the computer and can be entered manually.

The screenshot displays a simulation interface for a computer architecture. At the top, a 16-bit bus is shown with bit positions 15 down to 0. Below the bus, the instruction format is defined: Opcode (bits 15-10), R (bits 9-8), IX (bits 7-6), I (bit 5), and Address (bits 4-0). The interface includes several registers and control elements:

- Registers:** GPR 0, GPR 1, GPR 2, GPR 3, PC, MAR, MBR, IR, IX1, IX2, IX3, MFR, and CC. Each register has a 'Load' button and a 16-bit value display (except for MFR and CC which are 4-bit).
- Control Buttons:** 'Store', 'Load', 'Next Step', 'Reset', 'IPL', and 'RUN'.
- Initial Values:** All registers are initialized to 0000000000000000, except for MFR and CC which are 0000.

The Register Indicators display the values of all kinds of registers.

- GPR0-GPR3: These indicate 16-bit General Purpose Registers.
- IXR1-IXR3: These are 16-bit Index Registers used for pointing to operand addresses during the run of a program.
- MAR: It is a 12-bit Memory Address Register.
- MBR: It is a 16-bit Memory Buffer Register.
- PC: It is 12-bit Program counter, a CPU register in the computer processor which has the address of the next instruction to be executed from memory.
- IR: 16-bit Instruction Register, which holds the instructions that are executed currently.
- MFR: It is a Machine Fault Register of 4-bit.

Functions of the buttons in Controller Area:

**IPL:** IPL button will load the text file with address and instructions to the memory(array).

**Next Step:** It is a Single Step button, which is used to execute the one step at a time to determine how it is functioning.

**Load:** Load button is used to execute the load operation, which loads the value of address in MAR into MBR.

**Store:** Store button is used to execute the store operation, which stores the value in MBR into the address of MAR.

**Reset:** Resets the values of the toggle buttons and all the registers to zeros.

**Run:** Run button will execute all the instructions in the input file and will provide the final output.