DESIGN NOTES

Boomika Karuppaiah

Sumanth Nandeti

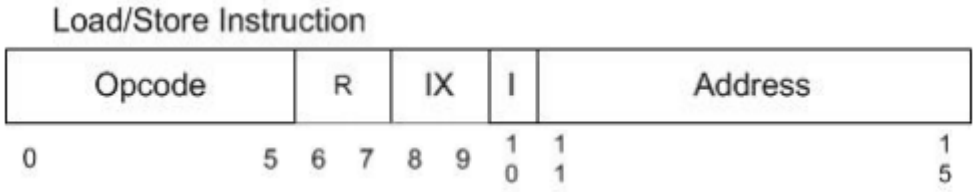
Waad Algorashy

Kamal Subedi

The project's goal is to create an assembly language-based simulator of a modest traditional CISC computer. The project's first phase will have the following features:

- 4 General Purpose Registers (GPRs) – each 16 bits in length **[R0, R1, R2, R3]**
- 3 Index Registers – 16 bits in length **[X1, X2, X3]**
- 16-bit words
- Memory of 2048 words, expandable to 4096 words
- Word addressable

Load/Store Instruction:



Opcode: 6 bits – Specifies one of 64 possible instructions; Phase 1 will have 5 Load/Store Opcodes.

R : 2 bits – R0(00), R1(01), R2(10), R3(11) General Purpose Registers.

IX : 2 bits – X1(01), X2(10), X3(11) Indexed Registers.

I : 1 bit – I=0 specifies indirect addressing, or otherwise no indirect addressing.

Address: 5 bits – Specifies one of 32 locations.

Effective address to execute various Load/Store instructions will be computed as follows:

Effective Address (EA) =

I = 0:

IX = 00: content (address field)

IX = 01 or 10 or 11: content(IX) + content of address field

I = 1:

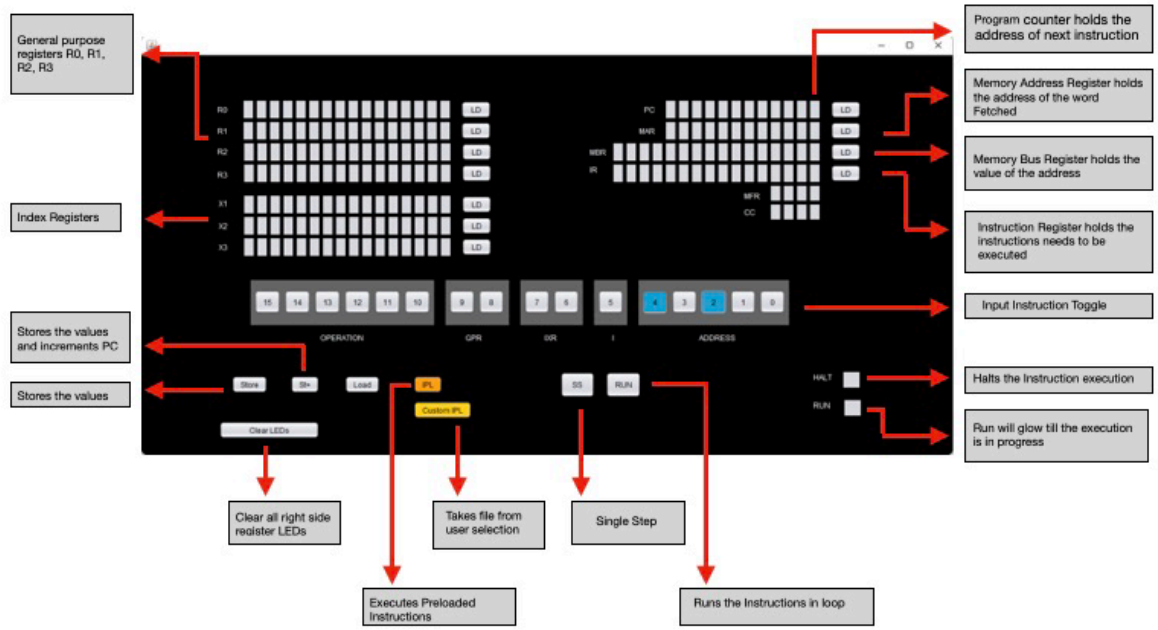
IX = 00: content (content (address field))

IX = 01 or 10 or 11: content(content(IX) + content of address field)

Following are the Load/Store instructions implemented in the Simulator (I is optional):

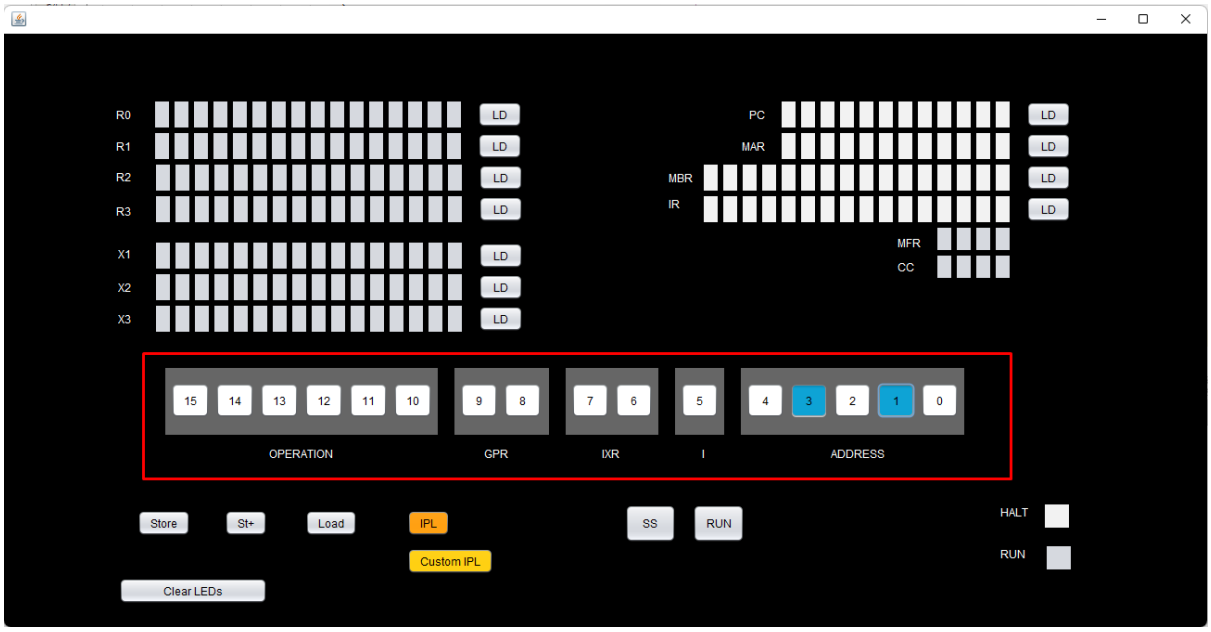
OpCode	Instruction	Description
01	LDR r x I address	Load Register From Memory, r = 0..3 r <- c(EA)
02	STR r x I address	Store Register To Memory, r = 0..3 Memory(EA) <- c(r)
03	LDA r x I address	Load Register with Address, r = 0..3 r <- EA
41	LDX x I address	Load Index Register from Memory, x = 1..3 Xx <- c(EA)
42	STX x I address	Store Index Register to Memory. X = 1..3 Memory(EA) <- c(Xx)

Simulator Design:

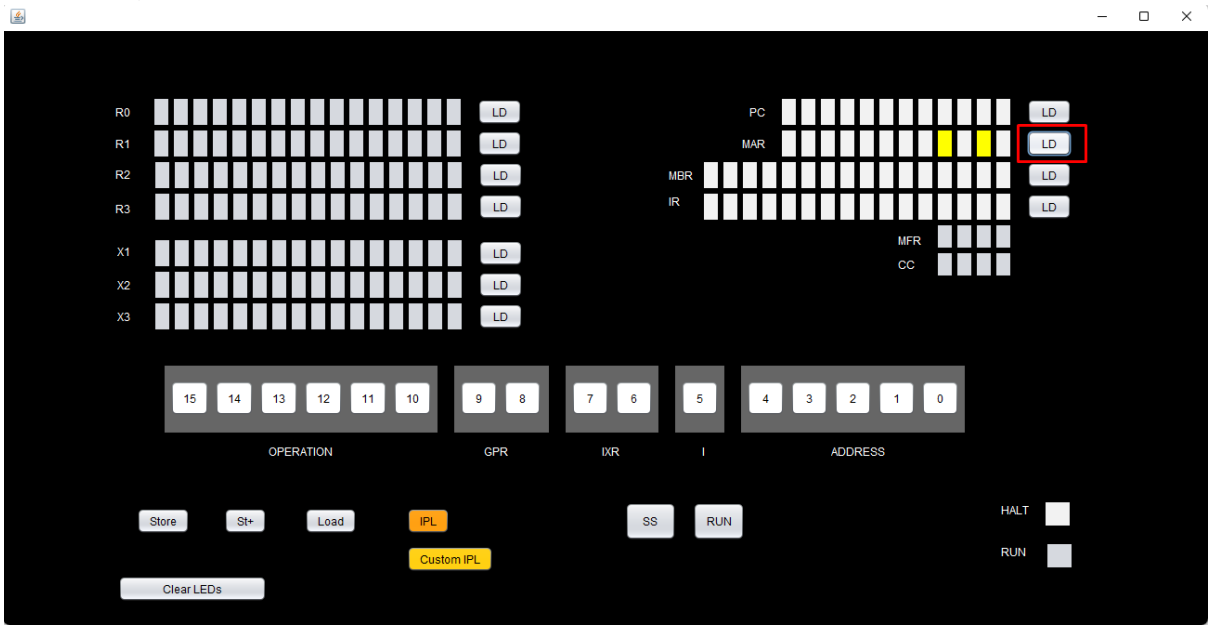


How to add instruction to memory:

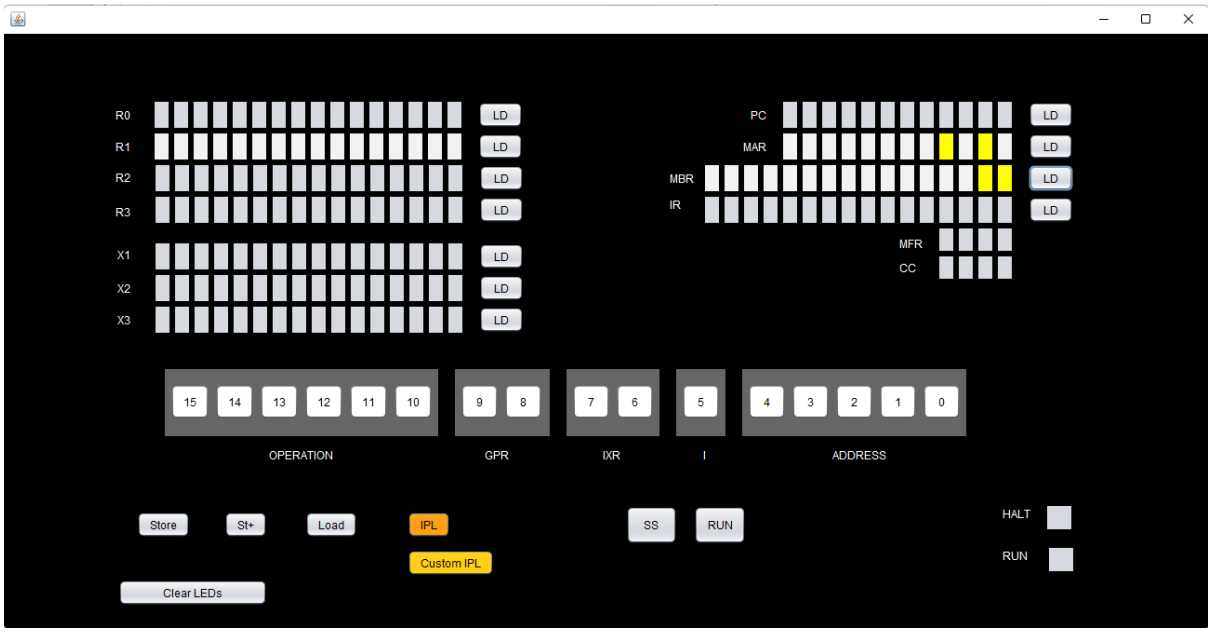
1. Input MAR instruction in Instruction Toggle, as highlighted below (White LED is 0 Blue LED is 1:



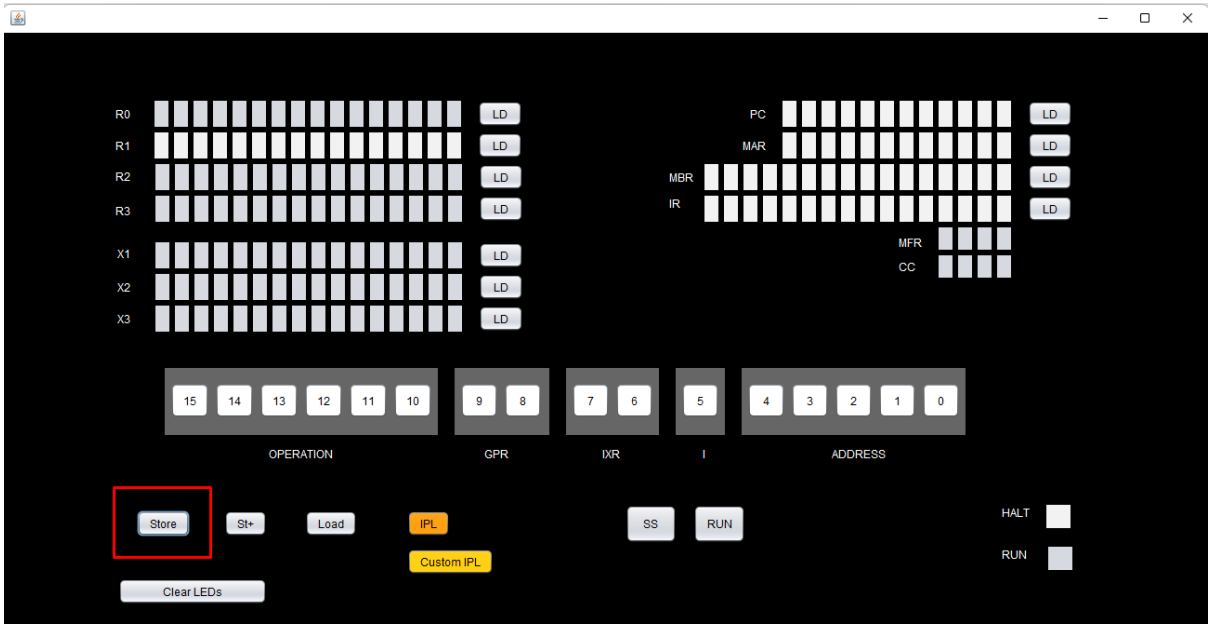
2. Click the “LD” button next to MAR to Load it to the MAR register(Yellow LED is for 1, white LED is for 0):



3. Do the above steps for MBR also:



4. Now Click the “Store” button to store the MBR instruction to the MAR memory location

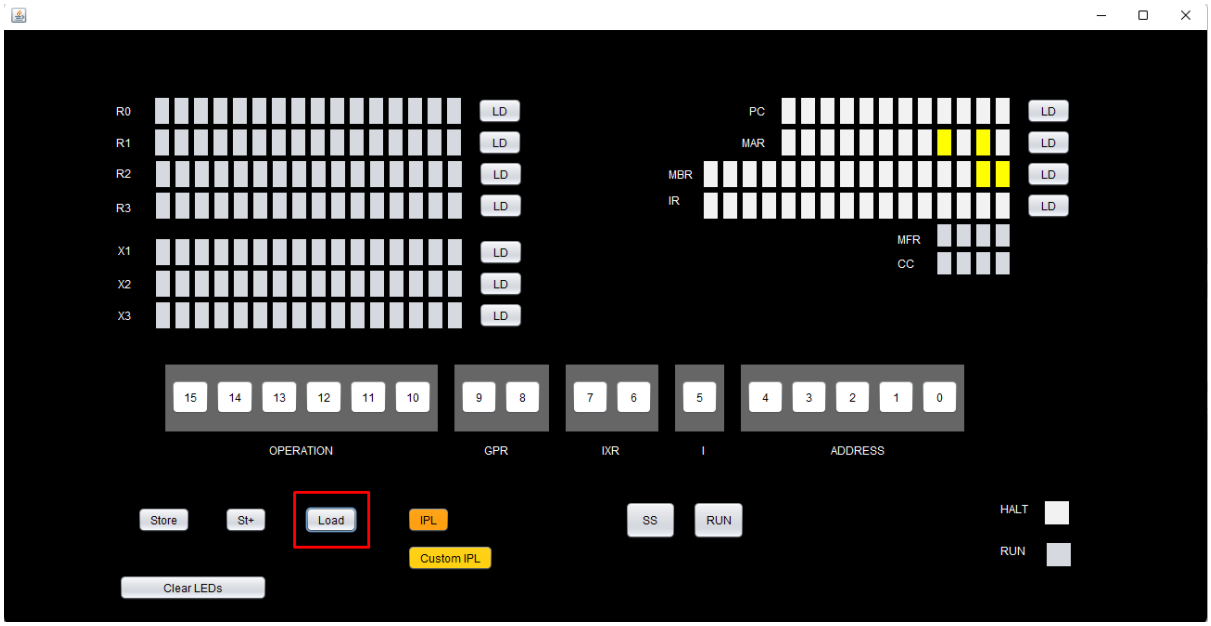


5. Store+ will work exactly like Store but with the added feature of incrementing the MAR value, so that we don’t have to input it again for the next address.

How to Load instruction from memory:

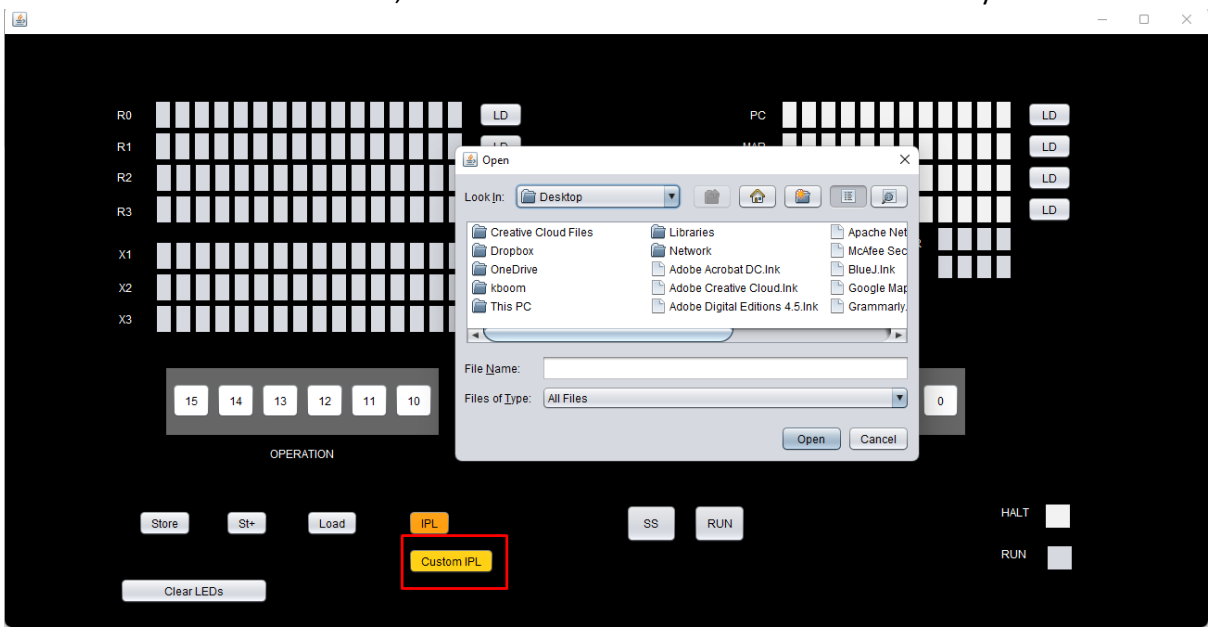
1. Click MAR instruction in the Instruction toggle and click “LD” to load it in the MAR register.

2. Click “Load” button, the instruction stored in MAR will be displayed in the MBR register:

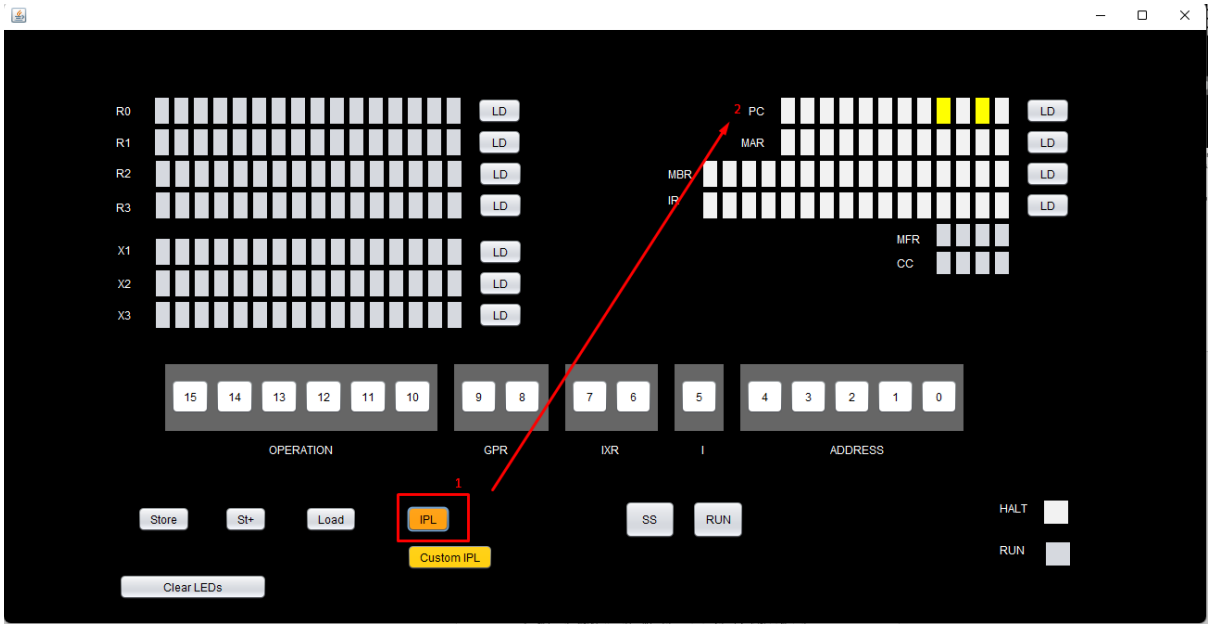


How to Load IPL file or Custom IPL file:

1. Click on the “IPL” button, this will load the contents of the default IPL.txt file to the memory.
2. Click on the “Custom IPL” button, to load the content of another txt file to memory.

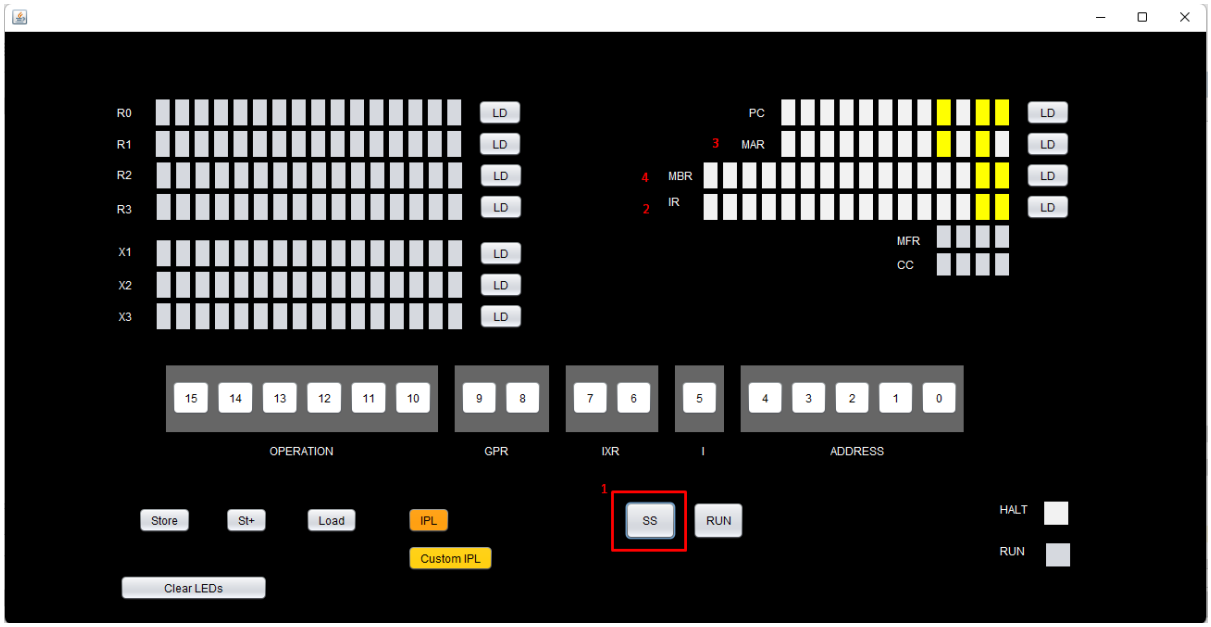


3. Once the “IPL” or “Custom IPL” button is clicked the PC will be loaded with the first instruction’s memory address:



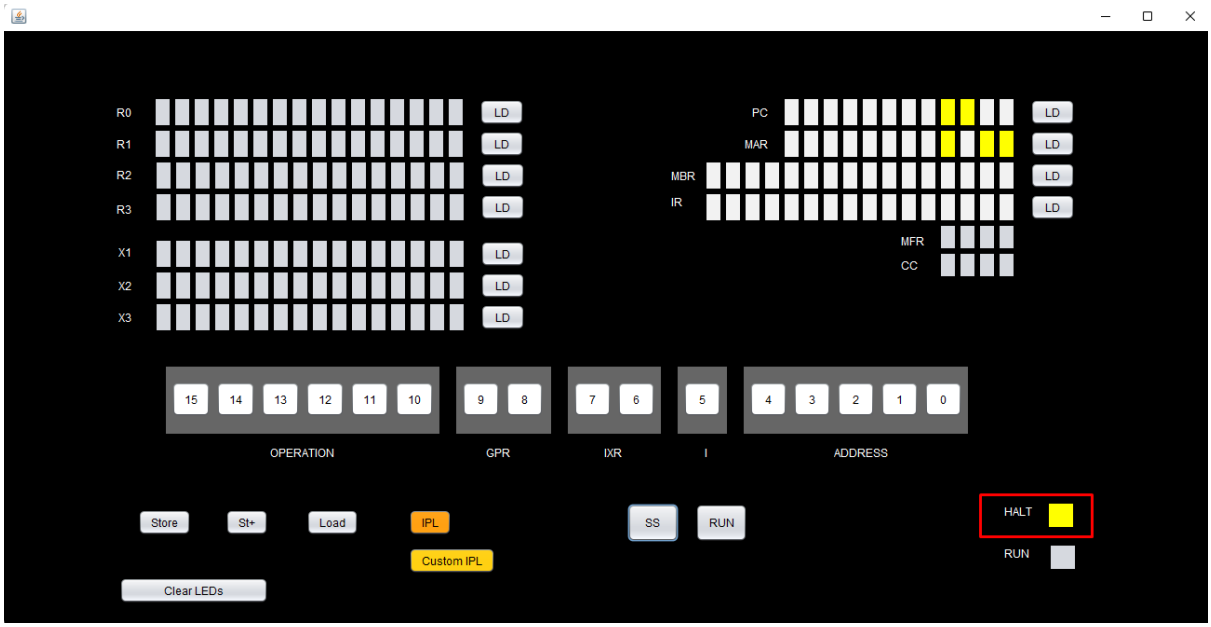
How to Single Step:

1. If “IPL” or “Custom IPL” was clicked, the PC would have automatically loaded, else if you want to try some other PC value you can click on the Instruction toggle buttons and click its “LD” to load the PC.
2. Click on the “SS” button to Single step, there will be a delay of 1 sec for registers to be loaded with their value.



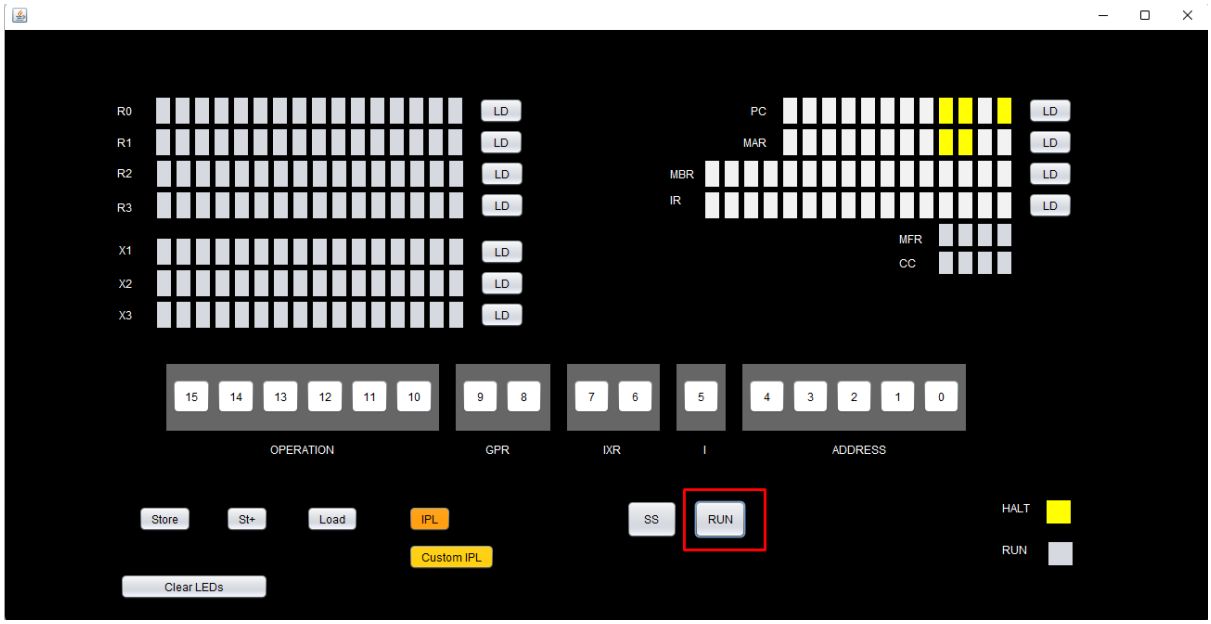
3. Once “SS” is clicked the registers will be loaded with their values in the following order: IR -> MAR -> MBR -> GPRs or IXs -> PC=PC+1.

4. If Halt is encountered, HALT LED would glow:



How to Run:

1. To do SS continuously till HALT is encountered, Click on the “RUN” button, the RUN LED will glow till the run is in progress and then turn to white when HALT is encountered.



Clear LEDs:

The Clear LEDs button will clear the LEDs of the right-side registers: