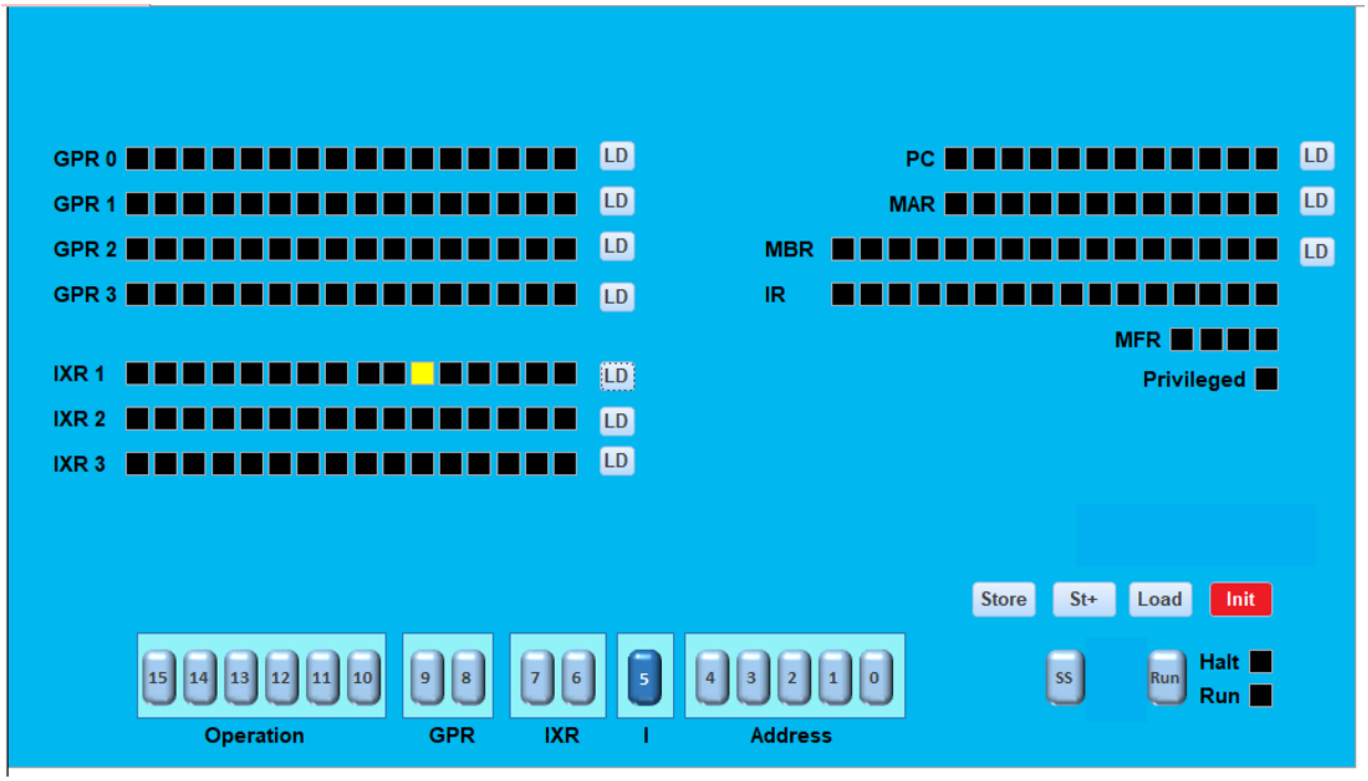
Project 1 Front Panel Requirement Clarifications

Machine Register	Function
PC	Display and Change
MAR	Display and Change
MBR	Display and Change
GPR Registers 0-3	Display and Change
Index Registers 1-3	Display and Change
CC	Display
MFR (memory fault register)	Display
IR	Display and Change

Below is a sample front panel display. Yours does not have to match exactly except that the required items above must be on the front panel. Note that the switches at the bottom were set to 0000 0000 0100 0000. The load button (LD) was pressed next to IXR 1 resulting in IXR 1 being loaded with this value and the display values set so that the appropriate light (yellow) is set.

NOTES FOR THIS PROJECT

- **You may use text boxes that display either the binary values in 0s and 1s or hexadecimal numbers. You DO NOT have to implement the lights independently.**
- **I have specified , such as SS which would go through a single cycle, that is, would go through fetch, then decode steps, etc. The St+ button is a neat, not required button that when you set the MAR and MBR and press St+, the machine stores the value in the MBR into the memory location indicated by the MAR and increments the MAR to the next machine location so that one only has to load the MBR for the next location. Useful in loading programs.**
- **The Privileged indicator is not required. It was put there to indicate that the user program tried to execute within the first 6 locations.**
- **You should have a condition code display, (after Project 1).**
- **The Init button is the IPL button, you may label it as you like.**


GPR 0
GPR 1
GPR 2
GPR 3
IXR 1
IXR 2
IXR 3
LD
PC
MAR
MBR
IR
MFR
Privileged
Store
St+
Load
Init
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
Operation
GPR
IXR
I
Address
SS
Run
Halt
Run

The machine front panel should contain an Initial Program Load (IPL) button that restarts the machine and loads a file IPL.txt into memory. The file is in a format that contains two hexadecimal numbers per line, the first being a machine address, and the second being the content at that address. Example:

```
0000    004F
000A    0009
0020    000A
```

The results of loading the above would be (in decimal)

location 0 would get 79 (004F)
location 10 would get 9 (0009)
location 32 would get 10 (000A)

This will be the means in which we load programs from a file. I will use files in this format to test the simulator. You should provide the file name that the machine loads from.

Additional note. The simulator example above numbers the bits increasing from right to left to reflect the numerical significance of the bits. Note that the project description shows bit positions increasing left to right.