#Author: Group 1 Member

#Olayinka Gbolahan

#Sidan Adi

#Mushary Alghamdi

Input:

The user can input either a hexadecimal number or a binary number in the input panel.

Store:

Store the contents of the MBR register at the address specified by the contents of the MAR register.

Store+:

Perform a "Store" operation and increment the MAR register by one.

Load:

Load the memory contents of the address specified by the contents of the MAR register into the MBR register.

IPL:

Load “IPL.txt” .

Single Step:

Run a single stage (fetch/decode/execute).

Single Instruction:

Run one single instruction.

Run:

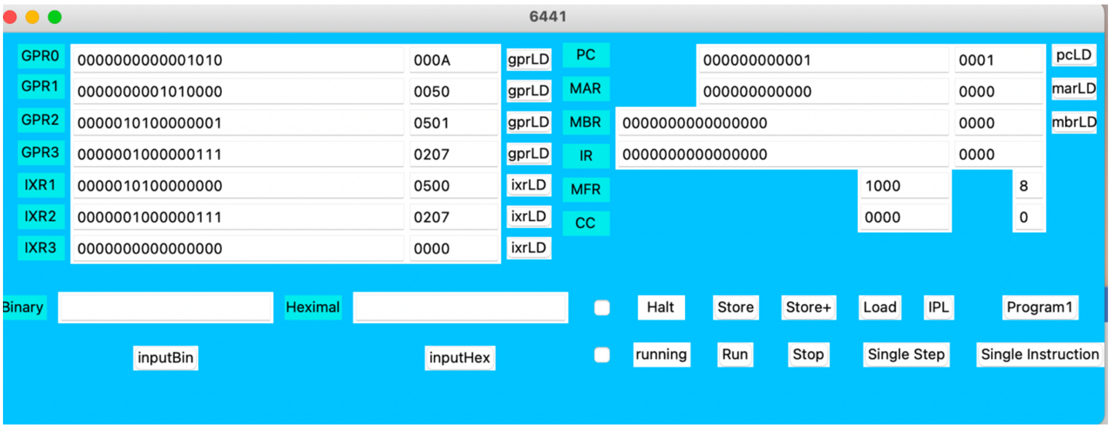
Run the emulator.

Stop:

Halt the machine by executing “halt” instruction

Program1:

This will load the contents of program1.txt for it to execute, the PC starts at 0x100. Click Program1, then click run, the output of the Debug GUI interface displays the command being executed, Enter numbers in the console. Due to program problems, you need to click the console keyboard input button every time the Plz Input characters are printing on the



GPR0	0000000000001010	000A	PC	000000000001	0001
GPR1	0000000001010000	0050	MAR	000000000000	0000
GPR2	0000010100000001	0501	MBR	0000000000000000	0000
GPR3	0000001000000111	0207	IR	0000000000000000	0000
IXR1	0000010100000000	0500	MFR	1000	8
IXR2	0000001000000111	0207	CC	0000	0
IXR3	0000000000000000	0000			

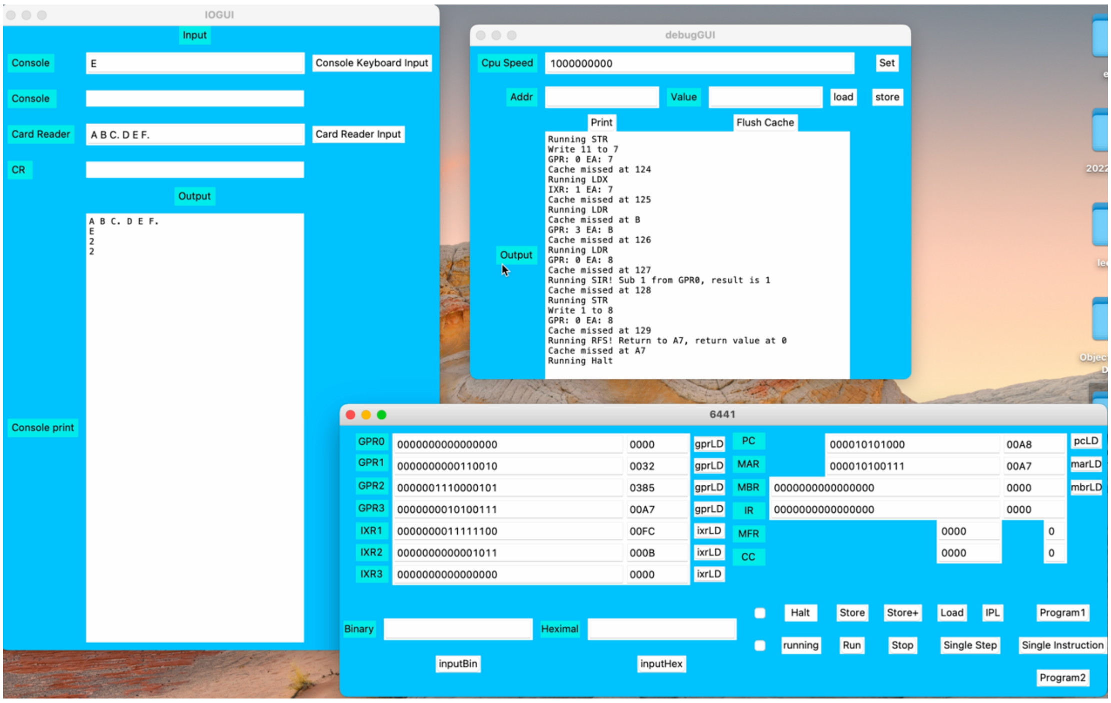
output. And finally the result is displayed in the Console Print of the IO GUI. The final result will be arranged vertically. If there is something you don't understand, please watch the demo video.

LD:

Click on LD to set its data to the data in input

Program2

This will load the contents of program 2.txt to execute, and the PC starts at 0x200. Click "Program 2", then "Run" the output of the debug GUI interface will show the command being executed, enter a number in the console, a sentence in CR, and the desired word in the console. Program 2 will print the sentence in the console and require the user to enter a word. The paragraphs in the file are then searched into memory. The console prints out words, periods, and word numbers in sentences if it contains the word.



Console: E

Console:

Card Reader: A B C. D E F.

CR:

```
A B C. D E F.
E
2
2
```

Cpu Speed: 1000000000

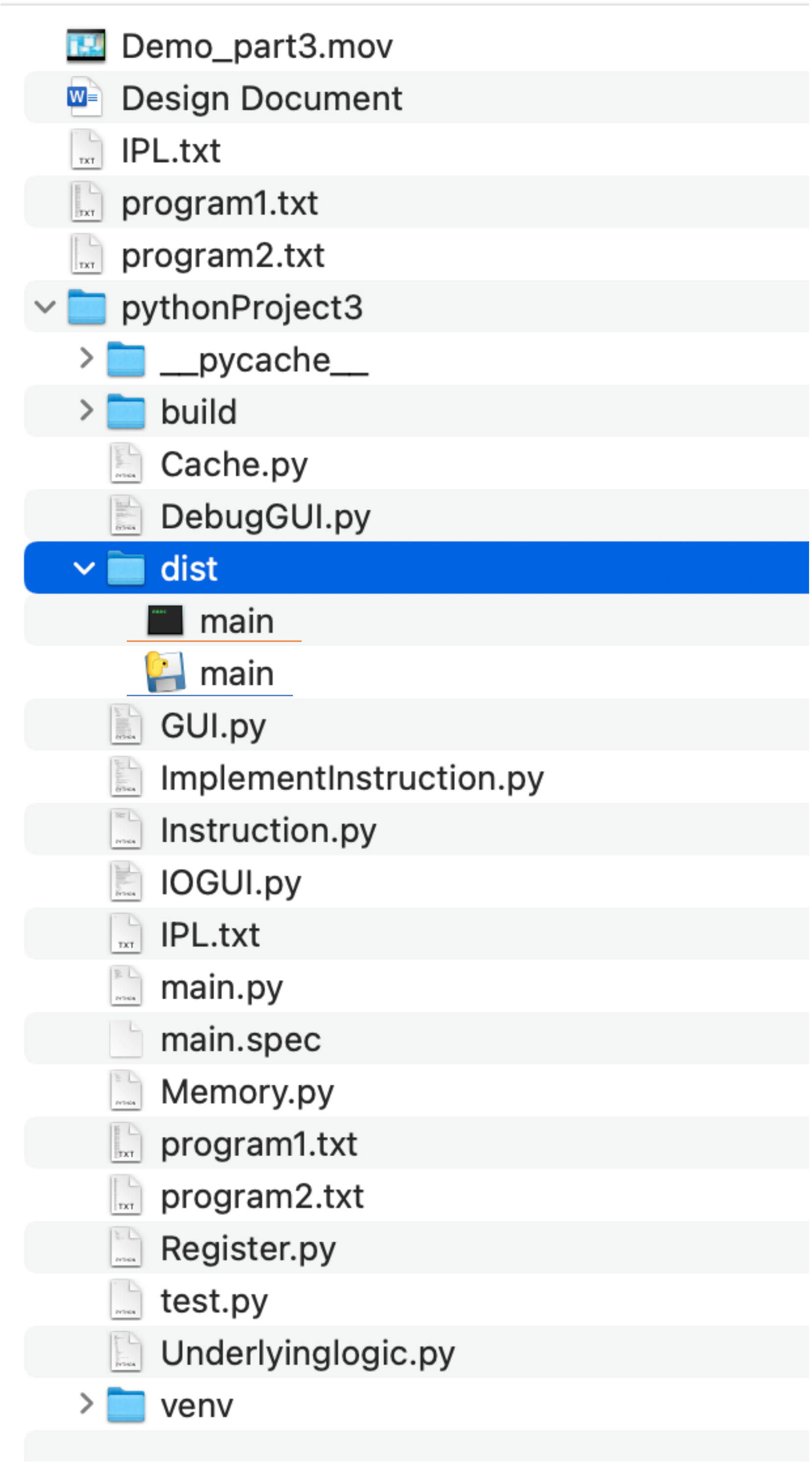
Addr:

Value:

```
Running STR
Write 11 to 7
GPR: 0 EA: 7
Cache missed at 124
Running LDX
IXR: 1 EA: 7
Cache missed at 125
Running LDR
Cache missed at B
GPR: 3 EA: B
Cache missed at 126
Running LDR
GPR: 0 EA: 8
Cache missed at 127
Running SIR! Sub 1 from GPR0, result is 1
Cache missed at 128
Running STR
Write 1 to 8
GPR: 0 EA: 8
Cache missed at 129
Running RFS! Return to A7, return value at 0
Cache missed at A7
Running Halt
```

GPR0	0000000000000000	0000	PC	000010101000	00A8
GPR1	0000000000110010	0032	MAR	000010100111	00A7
GPR2	0000001110000101	0385	MBR	0000000000000000	0000
GPR3	0000000010100111	00A7	IR	0000000000000000	0000
IXR1	0000000011111100	00FC	MFR	0000	0
IXR2	0000000000001011	000B	CC	0000	0
IXR3	0000000000000000	0000			

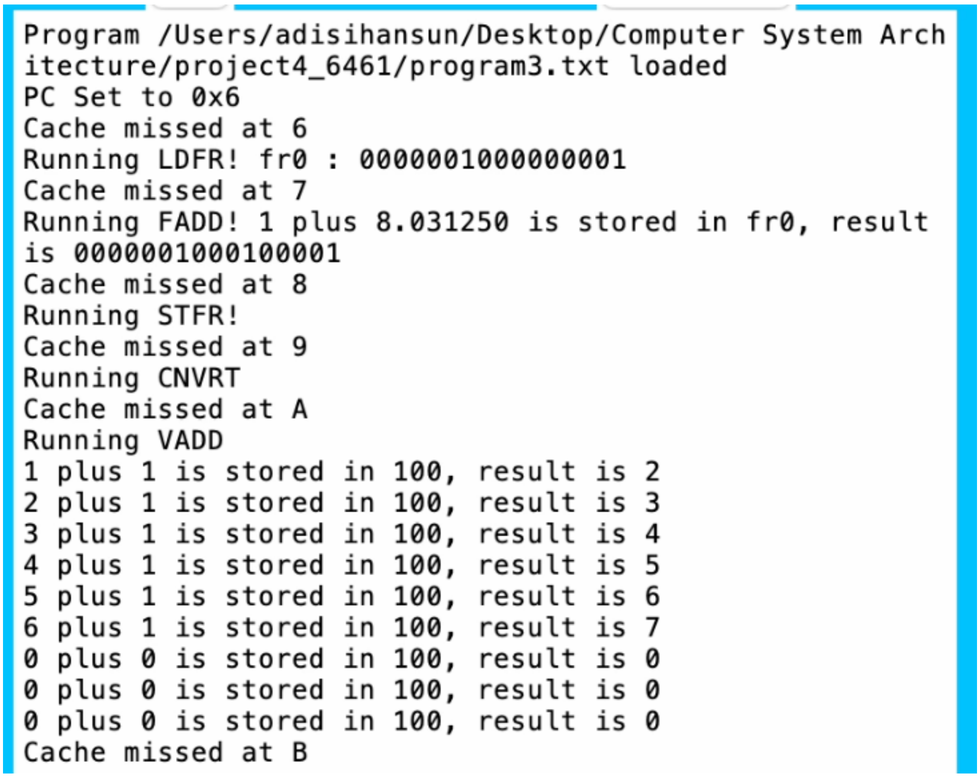
Or you can click the .exe file. When you click the button of program1, the file selection will pop up, you can select the program2.txt file.



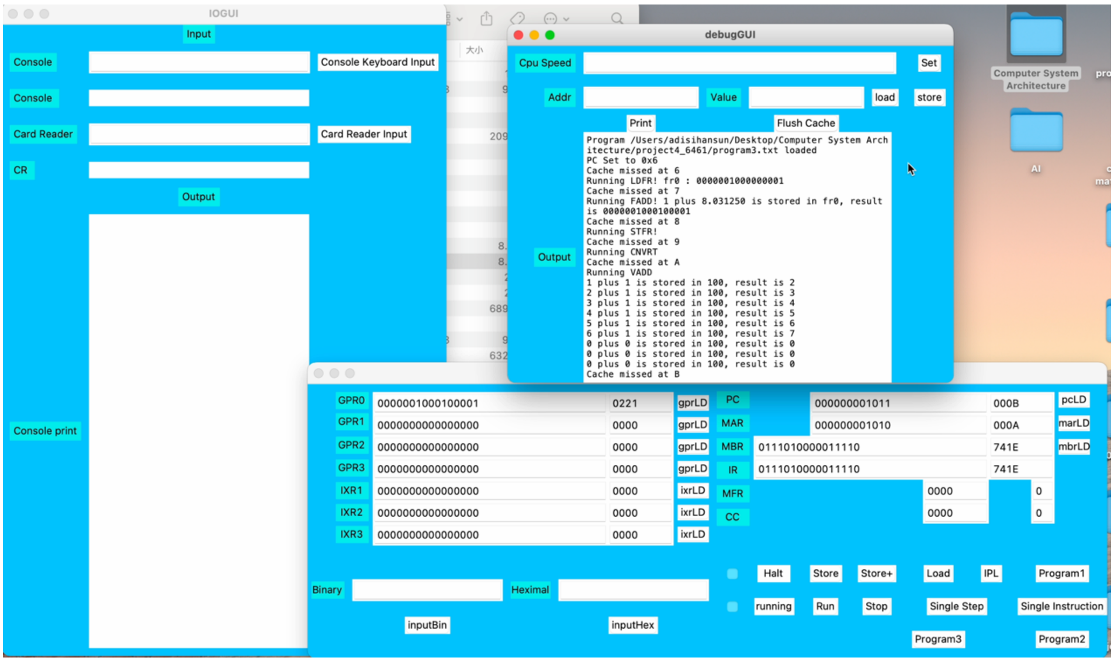
```
Demo_part3.mov
Design Document
IPL.txt
program1.txt
program2.txt
pythonProject3
  __pycache__
  build
  Cache.py
  DebugGUI.py
  dist
    main
    main
  GUI.py
  ImplementInstruction.py
  Instruction.py
  IOGUI.py
  IPL.txt
  main.py
  main.spec
  Memory.py
  program1.txt
  program2.txt
  Register.py
  test.py
  Underlyinglogic.py
  venv
```

Program3

You can directly click the program3 pair button, and select program3.txt.



```
Program /Users/adisihansun/Desktop/Computer System Arch
itecture/project4_6461/program3.txt loaded
PC Set to 0x6
Cache missed at 6
Running LDFR! fr0 : 0000001000000001
Cache missed at 7
Running FADD! 1 plus 8.031250 is stored in fr0, result
is 0000001000100001
Cache missed at 8
Running STFR!
Cache missed at 9
Running CNVRT
Cache missed at A
Running VADD
1 plus 1 is stored in 100, result is 2
2 plus 1 is stored in 100, result is 3
3 plus 1 is stored in 100, result is 4
4 plus 1 is stored in 100, result is 5
5 plus 1 is stored in 100, result is 6
6 plus 1 is stored in 100, result is 7
0 plus 0 is stored in 100, result is 0
0 plus 0 is stored in 100, result is 0
0 plus 0 is stored in 100, result is 0
Cache missed at B
```



```
Program /Users/adisihansun/Desktop/Computer System Arch
itecture/project4_6461/program3.txt loaded
PC Set to 0x6
Cache missed at 6
Running LDFR! fr0 : 0000001000000001
Cache missed at 7
Running FADD! 1 plus 8.031250 is stored in fr0, result
is 0000001000100001
Cache missed at 8
Running STFR!
Cache missed at 9
Running CNVRT
Cache missed at A
Running VADD
1 plus 1 is stored in 100, result is 2
2 plus 1 is stored in 100, result is 3
3 plus 1 is stored in 100, result is 4
4 plus 1 is stored in 100, result is 5
5 plus 1 is stored in 100, result is 6
6 plus 1 is stored in 100, result is 7
0 plus 0 is stored in 100, result is 0
0 plus 0 is stored in 100, result is 0
0 plus 0 is stored in 100, result is 0
Cache missed at B
```

GPR0	0000001000100001	0221	gprLD
GPR1	0000000000000000	0000	gprLD
GPR2	0000000000000000	0000	gprLD
GPR3	0000000000000000	0000	gprLD
IXR1	0000000000000000	0000	ixrLD
IXR2	0000000000000000	0000	ixrLD
IXR3	0000000000000000	0000	ixrLD

PC	000000001011	000B	pcLD
MAR	000000001010	000A	marLD
MBR	0111010000011110	741E	mbrLD
IR	0111010000011110	741E	
MFR	0000	0	
CC	0000	0	