User Guide of Simulator (Part IV)

Course: CSCI 6461 Computer System Architecture

Group2: Wei Lin, Xiao Xiao, Xiaojun Li

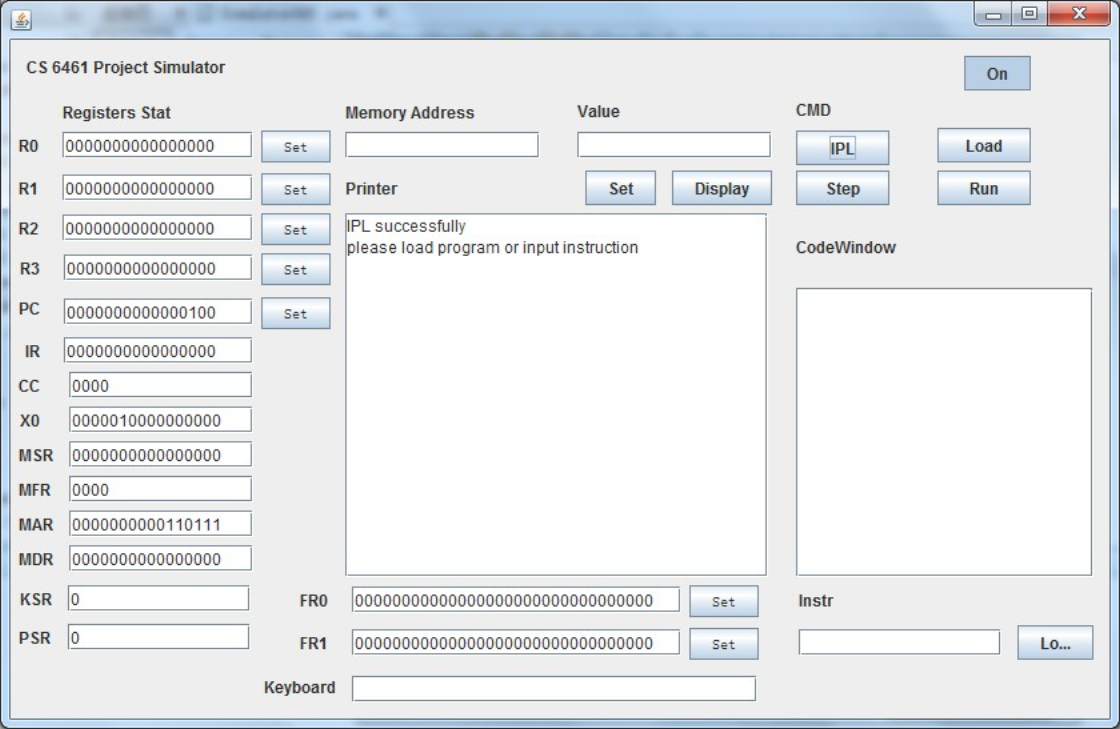
Date: December 04, 2013

Simulator Functionality (Added)

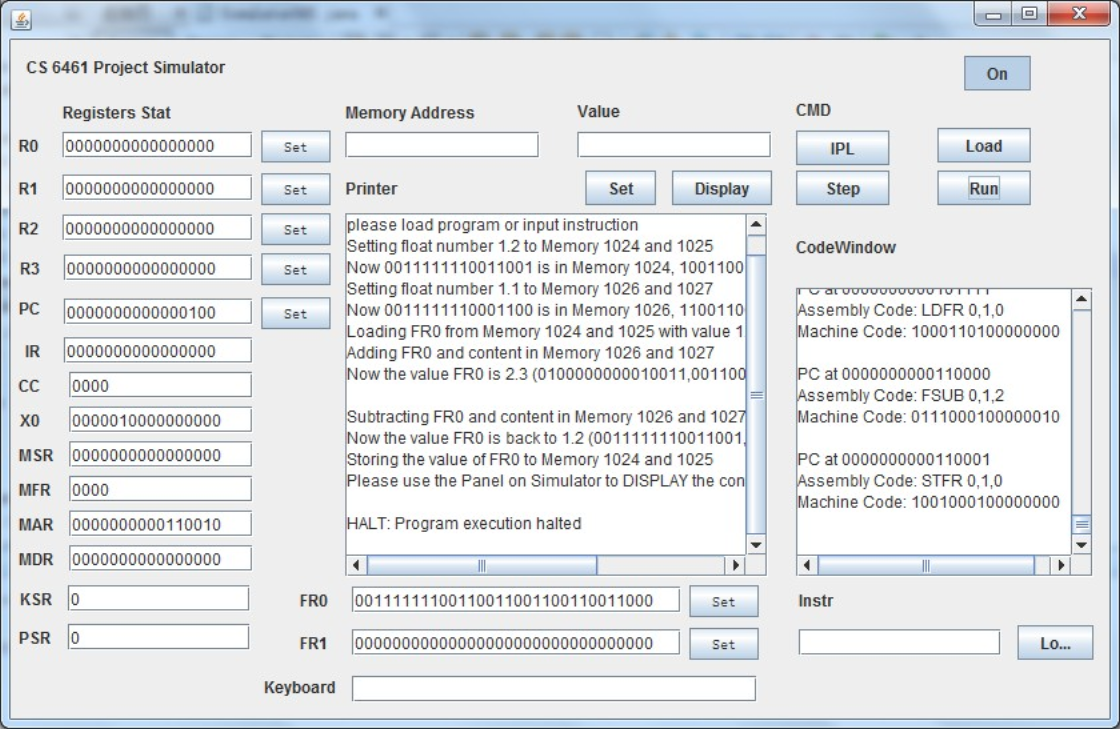
1. Increased the support of FLOAT and VECTOR operation, including float addition/subtraction, vector addition/subtraction, float point to fixed point number conversion and float number loading/storing.
2. Reinforced the functionality of IN/OUT instructions, for supporting the execution of program II.
3. Upgraded the architecture of Simulator, by using multi-thread for instruction execution, GUI value refreshing and peripheral devices input/output identification.
4. Elementary Support of build-in pipeline instruction execution
5. Increased support of TRAP instruction into 16 subroutines.

Step-by-Step Guide of Running Float Operation Test Program

1. There are three files named **float.txt**, **vector.txt** and **fixed.txt** for testing the functionality of float operation, vector operation and float to fixed conversion operation. They exist in the same directory with the runnable jar file of Simulator.
2. Firstly, double click the jar file Simulator.jar to open our program, and hit **On** to boot it, and hit the **IPL** to initial the program.



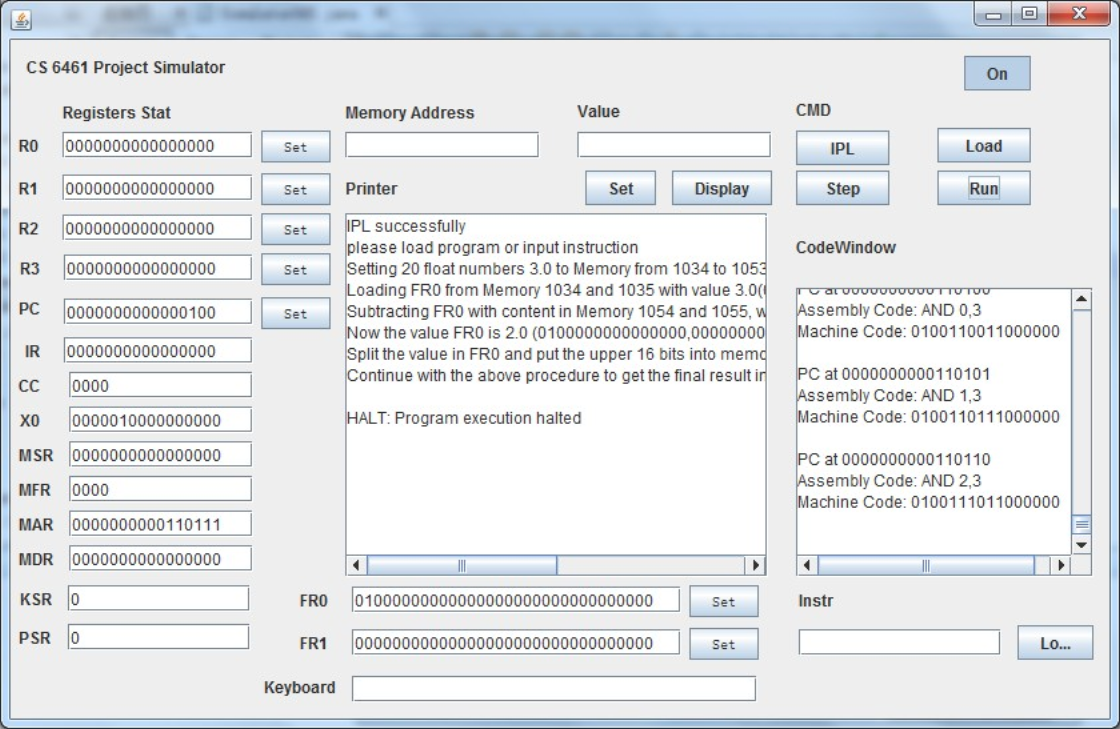
1. Then click **Load** to open a file selector, please choose **float.txt** at first, and hit the **Run** button to execute it. (If you want to run others files later, please remember to **IPL** program, then **Load** other files.)



1. (Explanation of float.txt)
2. The program is going to put a 32-bit float number 1.2 into Memory 1024 and 1025, and then put a 32-bit float number 1.1 into Memory 1026 and 1027 by using.
3. Then Load 1.2 from Memory to FD0.
4. Then Add 1.2 and 1.1 by using **FADD** and put the result 2.3 back to Memory 1024 and 1025. The float number 2.3 represents as 0100000000010011 in Memory 1024 and 0011001100110011 in Memory 1025.
5. Next, Subtract 2.3 and 1.1 by using **FSUB**, and then store back the result 1.2 into Memory 1024 and 1025.
6. You can check the content of Memory 1024 and 1025, they should be 0011111110011001 and 1001100110011001.

Step-by-Step Guide of Running Vector Operation Test Program

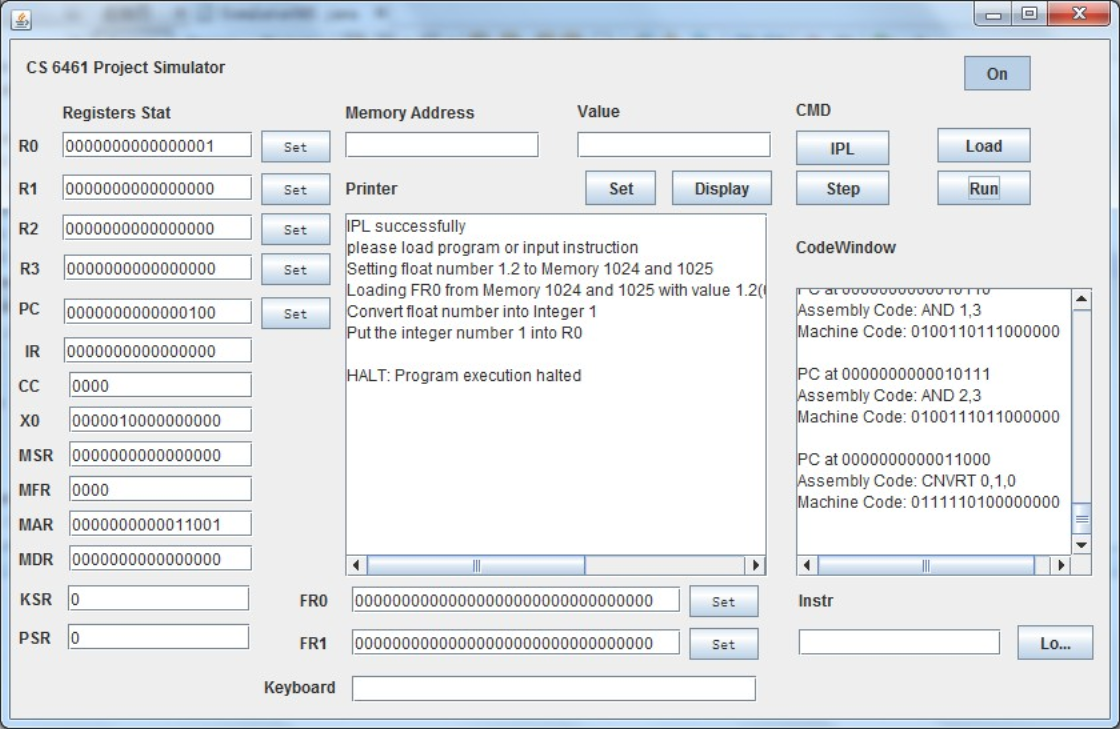
1. Please hit the **IPL** to re-initial the program, and use **Load** to **vector.txt** file, then hit **Run** button.



1. (Explanation of vector.txt)
2. This program is going to put value 1034 and 1054 at Memory 1030 (1024+6) and 1031 (1024+7), for preparation of using **VADD** later.
3. Then it puts ten float point numbers, each is 2.0, which represents as 0100000000000000,0000000000000000, into Memory from 1034 to 1053 (20 byte).
4. Then put tem float point numbers, each is 1.0, which represents as 0011111110000000,0000000000000000, into Memory from 1054 to 1073.
5. Next, the program uses **VADD** to add the vector (2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0, 2.0) and (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0), gets (3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0) and puts it back to Memory from 1034 to 1053.
6. Finally, it uses **VSUB** to subtract (3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0) and (1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0), then puts back to Memory 1034 to 1053.
7. Please use the panel to check the content of Memory.

Step-by-Step Guide of Running Conversion Operation Test Program

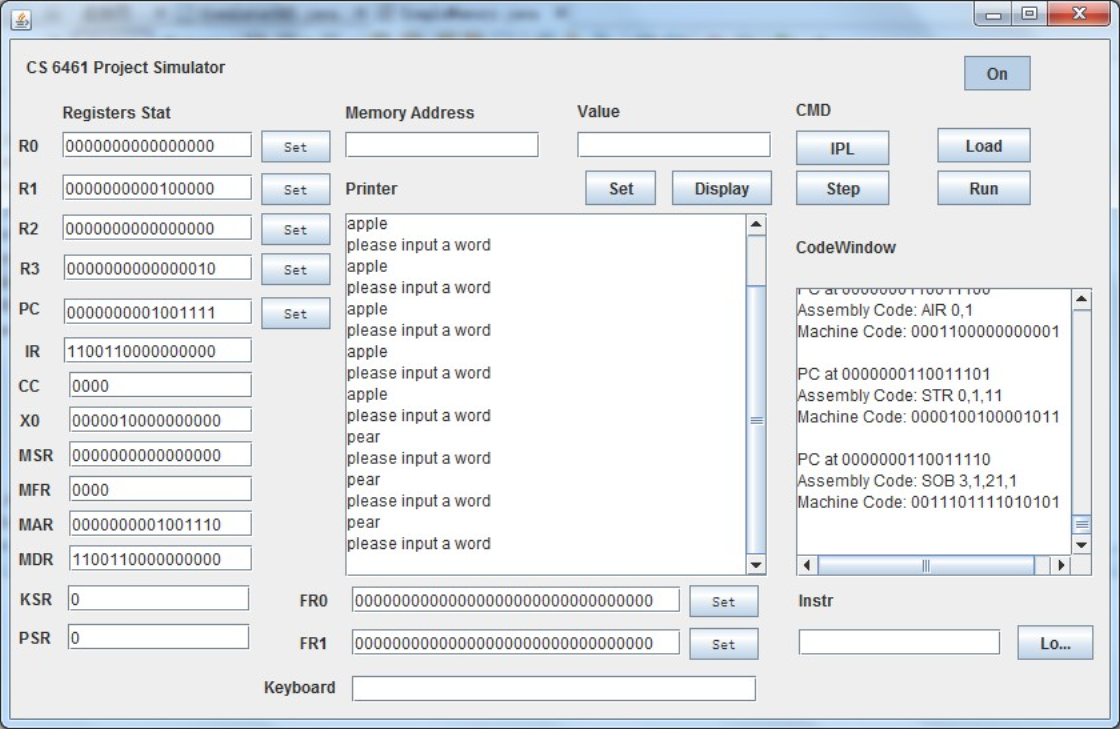
1. Please hit the **IPL** to re-initial the program, and use **Load** to **fixed.txt** file, then hit **Run** button.



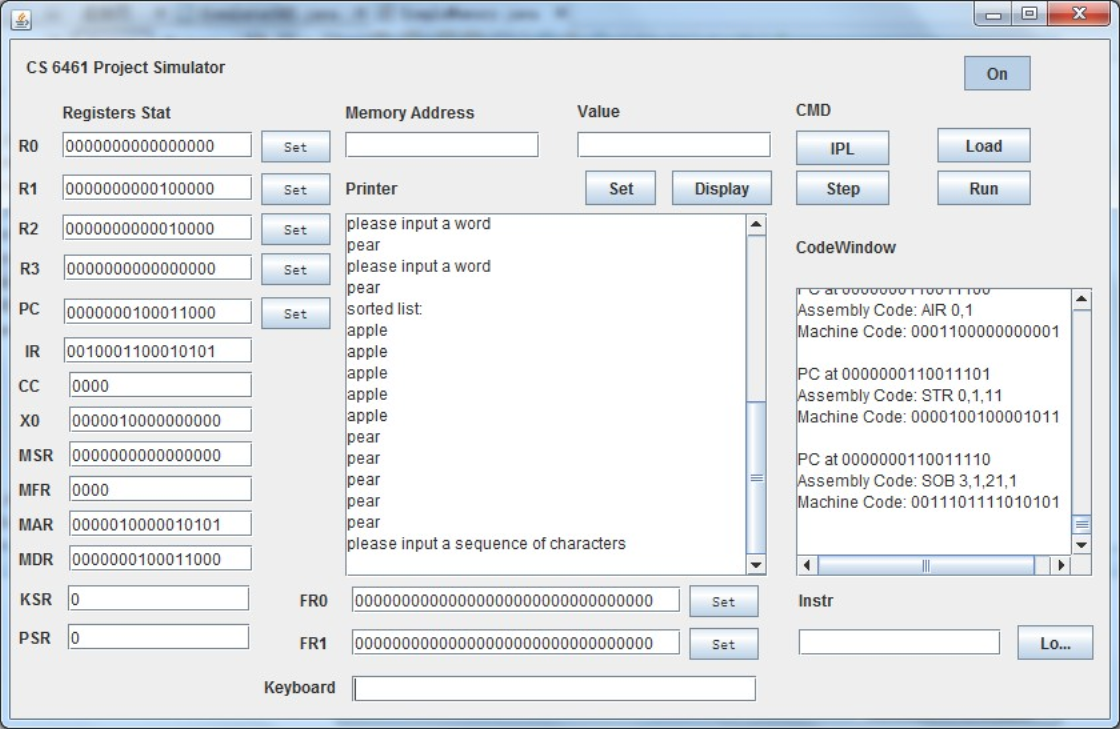
1. (Explanation of fixed.txt)
2. The program is going to put 1.2, which represents as 0011111110011001,1001100110011001, into Memory 1024 and 1025.
3. Then it uses CNVRT to convert it to 1.0 and puts it into R0. You can check the value of R0 on the panel.

Step-by-Step Guide of Running Program II

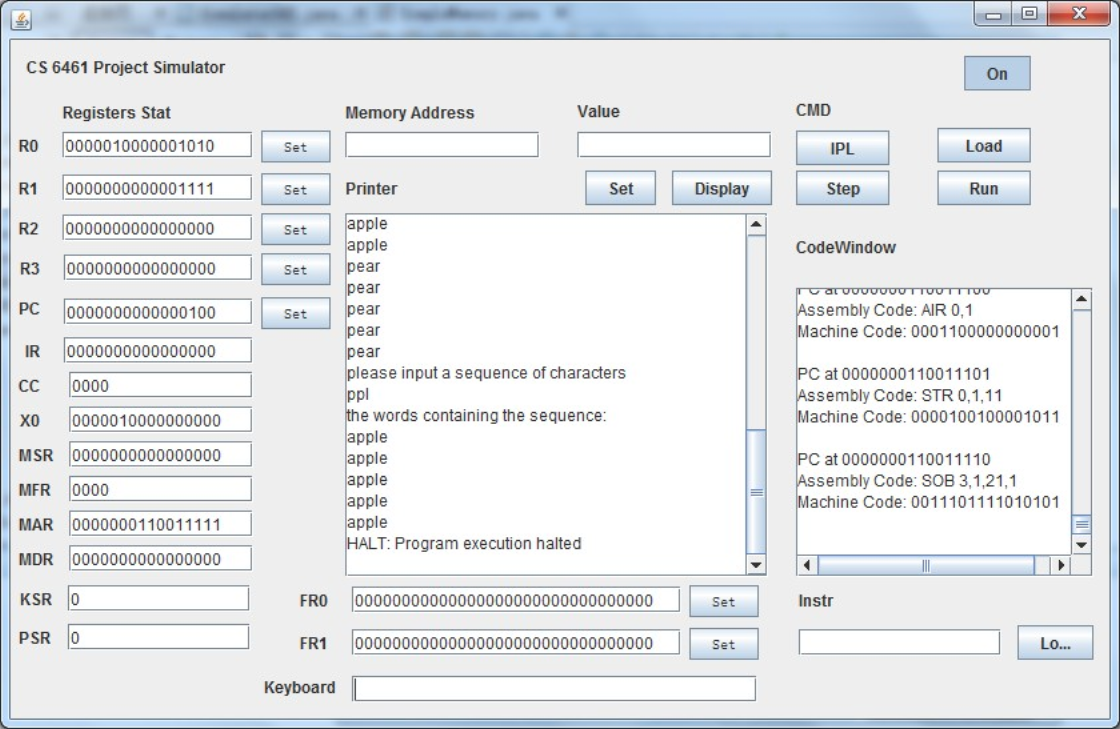
1. Please hit the **IPL** to re-initial the program, and use **Load** to **program2.txt** file, then hit **Run** button.
2. Then input 10 words in the **Keyboard** part of Panel. For test convenience, pleas input 5 “apple” and “pear”, use Enter after each word to submit into the program.



1. Then **please wait for 2-3 seconds after you input 10 words.** The program will calculate and output the 10 words according to their alphabetic order.



1. Finally, please input the comparison string “ppl”, (off course you can also input any string you like). Then the program will output the words that contain this string. In this test case, it will print out 5 “apple”s on the Panel of Simulator.



**Thanks for you HARD WORKING, dear TA~**