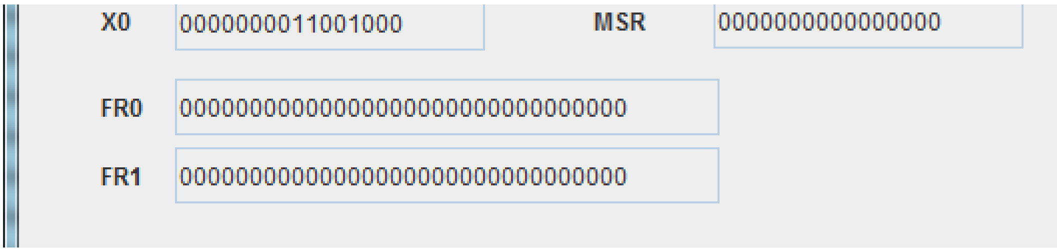
Design notes for Float & Vector

In part IV, we have implemented the float point and vector instructions, including LDFR, STFR, FADD, FSUB, CNVRT, VADD, and VSUB. In addition, we defined two 32-bit registers – FR0 and FR1, and displayed their contents on the UI as shown in *Fig.1*.



X0 0000000011001000

MSR 0000000000000000

FR0 00000000000000000000000000000000

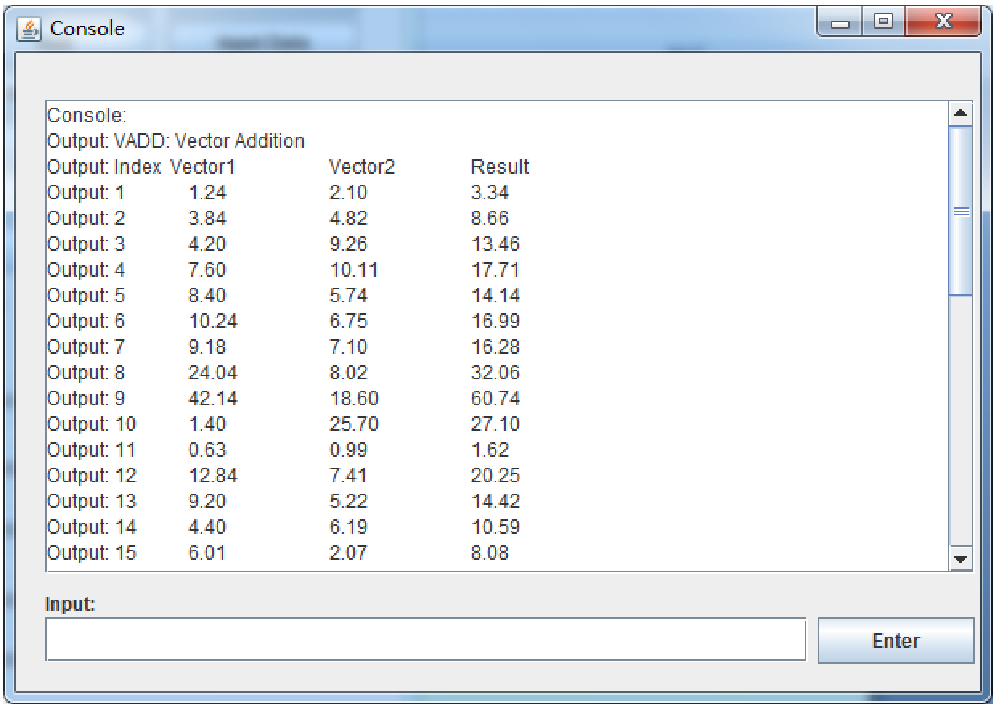
FR1 00000000000000000000000000000000

Fig.1 the UI of FR0 and FR1

LDFR is used for loading the content of addressed memory into FR0 or FR1 registers. On the contrary, STFR is used for storing the content of FR0 or FR1 into the addressed memory. FADD and FSUB can carry out the arithmetical operations of float point numbers. The float point numbers are stored in two successive memory units of 16-bit. The high byte is in the memory unit with higher address. VADD and VSUB are used for vector operation of float point numbers. The length of each vector is determined by R0.

In our program, we implemented six classes for these instructions. The memory is defined as 16-bit short integer for each unit, while FR0 and FR1 are 32-bit float point. Therefore, the key of float point operation is the transform between integer and float point. We used the

Float.intBitsToFloat() and Float.floatToIntBits () to implement the transform. Vector operation is memory-to-memory. The data of each vector is stored in the memory units with successive address. In order to print the operation process on the console UI as shown in *Fig.2*, vector1, vector2, and result are respectively stored into three Arrylist as well.



```
Console:
Output: VADD: Vector Addition
Output: Index Vector1        Vector2        Result
Output: 1       1.24           2.10           3.34
Output: 2       3.84           4.82           8.66
Output: 3       4.20           9.26           13.46
Output: 4       7.60           10.11          17.71
Output: 5       8.40           5.74           14.14
Output: 6       10.24          6.75           16.99
Output: 7       9.18           7.10           16.28
Output: 8       24.04          8.02           32.06
Output: 9       42.14          18.60          60.74
Output: 10      1.40           25.70          27.10
Output: 11      0.63           0.99           1.62
Output: 12      12.84          7.41           20.25
Output: 13      9.20           5.22           14.42
Output: 14      4.40           6.19           10.59
Output: 15      6.01           2.07           8.08
```

Fig.2 print the operation process of VADD and VSUB