Microprocessor Report

Introduction:

We have 9 classes in this project (datagui.java ,main.java, FloatRegister.java, Gui.java, igui.java, instruction.java, intRegister.java, optype.java, stations.java). Below you will find the details of each class.

* First class is datagui.java which has 2 methods. First method is initialise method which has 6 textfields inputs where three of them is how many cycles each instruction takes and the remaining three textfields are important to know how many load stations and add stations and mul stations are there. The second method is the main which is used to initialize a frame and then call the method initialize.
* Second class is igui.java which is used to write instructions that are going to be executed but in Branch not equal (BNEZ) we write the line we want to branch to not the label.
* Third class is gui.java which is the bonus. We did on the left int registers and on the right float registers and above is the instructions and down is the stations and in the middle you will find which cycle we are currently in.
* Fourth class is FloatRegister.java where we assign the register a value and it has a name that starts with an F concatenated with it the index.
* Fifth class is IntRegister.java where we assign the register a value and it has a name that starts with an R concatenated with it the index.
* Sixth class is the enum which has the optypes (subs, adds, divs, muls) that take as source 1 and source 2 two float registers while optypes (addi , subi) take int register and immediate. Also, The Instruction branch can take int register or float register. While, store and load instructions can take int register or float register. Addu and subu are like subs and adds but it is for the int register only.
* Seventh class is instruction.java which takes the optype of the instruction ,the destination , source1 , source2 or immediate based on the instruction needed to be executed. I also have 2 attributes issue and write result which takes a zero for starters. We also have an array of 2 elements named exec and Exec time variable which is the time taken by the instruction to be executed.
* Eight class is stations.java which has boolean variable named busy which lets us know if there is an instruction in the station or not. I have also the instruction I am handling at that moment, the value of source1, the value of source2, queue of source 1, queue of source 2 and destination.