Evan Smith

Code Summary for MIPS hazard detector

This code is fairly self-explanatory, although there are still a few kinks I am working out still with the display of stalls.

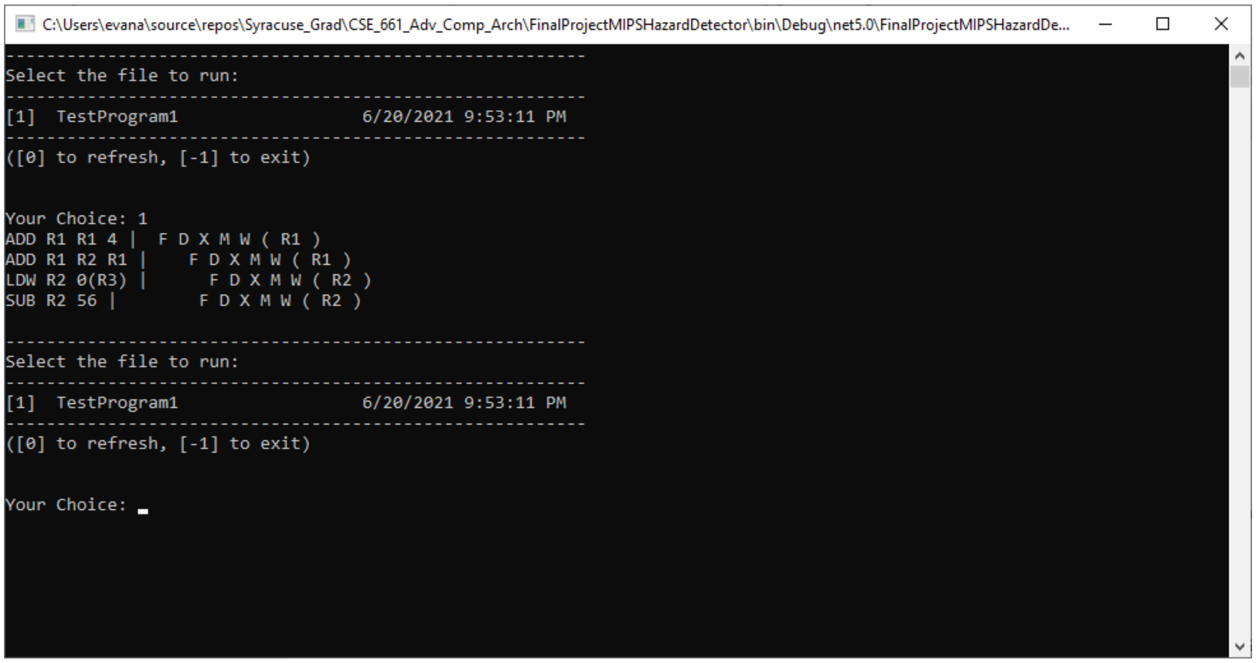
The code first prompts the user to select a code file, which must be saved in the target directory with a MIPS file extension. The program then reads in this file, attempts to parse it into instructions, and then proceeds to create Instruction objects.

These objects contain the registers that that instruction needs available, as well as tags them with a type, Read or Written, which is used to store an additional object for each that describes when exactly the register is required and when it will be made available to other instructions again. This allows for the change in logic between using a forwarding unit and not.

For each program, the code works through one instruction at a time, incrementing the stage of each instruction in turn, and injecting stalls if needed when an instruction is not able to proceed, since one or more of its required registers for that cycle were previously reserved. This system works because we work through the instructions via a queue, so the older instructions are able to maintain their hold on registers that they need, while newer instructions get stalls.

Before the demo, I will resolve the formatting issues with my GUI so that the timing diagrams look better. Have faith! 😊

An example screenshot of the fault notifying system:



```
C:\Users\evana\source\repos\Syracuse_Grad\CSE_661_Adv_Comp_Arch\FinalProjectMIPSHazardDetector\bin\Debug\net5.0\FinalProjectMIPSHazardDe...
---------------------------------------------------------
Select the file to run:
---------------------------------------------------------
[1]  TestProgram1                  6/20/2021 9:53:11 PM
---------------------------------------------------------
([0] to refresh, [-1] to exit)


Your Choice: 1
ADD R1 R1 4 |  F D X M W ( R1 )
ADD R1 R2 R1 |    F D X M W ( R1 )
LDW R2 0(R3) |      F D X M W ( R2 )
SUB R2 56 |        F D X M W ( R2 )

---------------------------------------------------------
Select the file to run:
---------------------------------------------------------
[1]  TestProgram1                  6/20/2021 9:53:11 PM
---------------------------------------------------------
([0] to refresh, [-1] to exit)


Your Choice: _
```