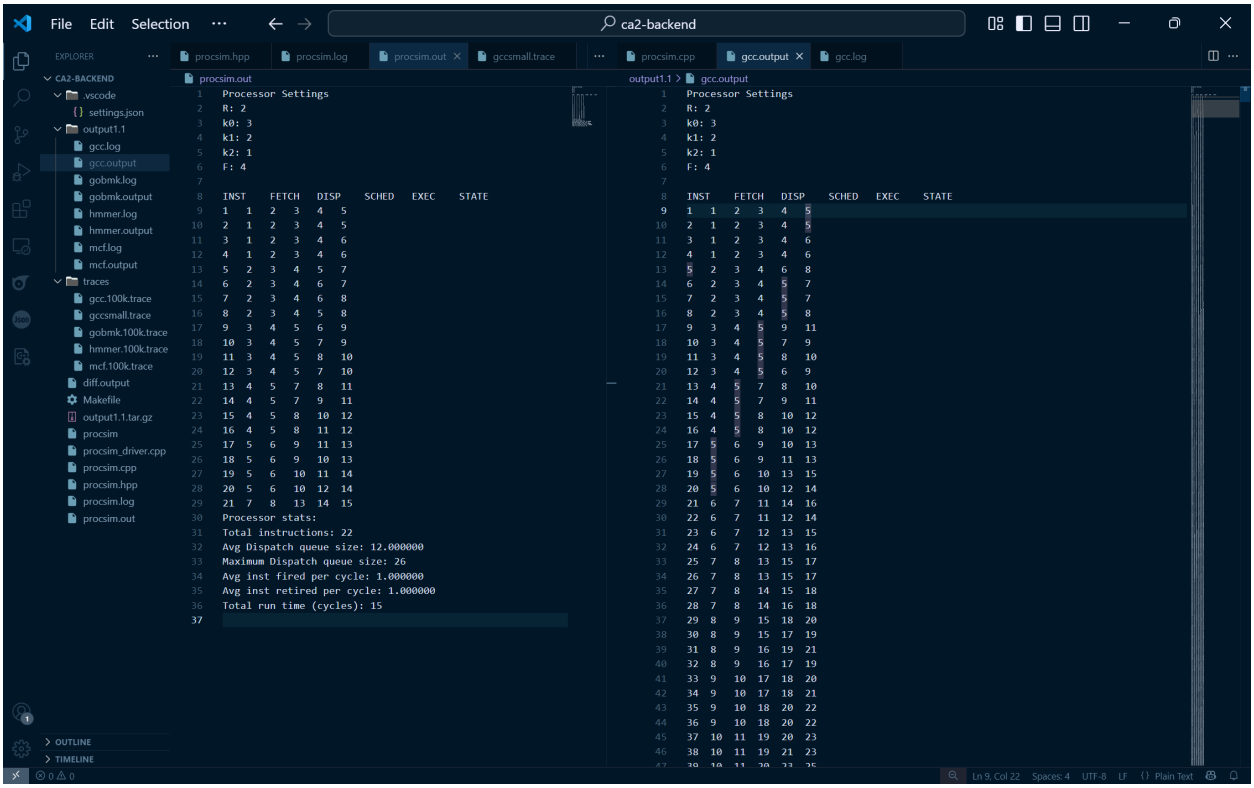
ECE 209AS CA2 Report

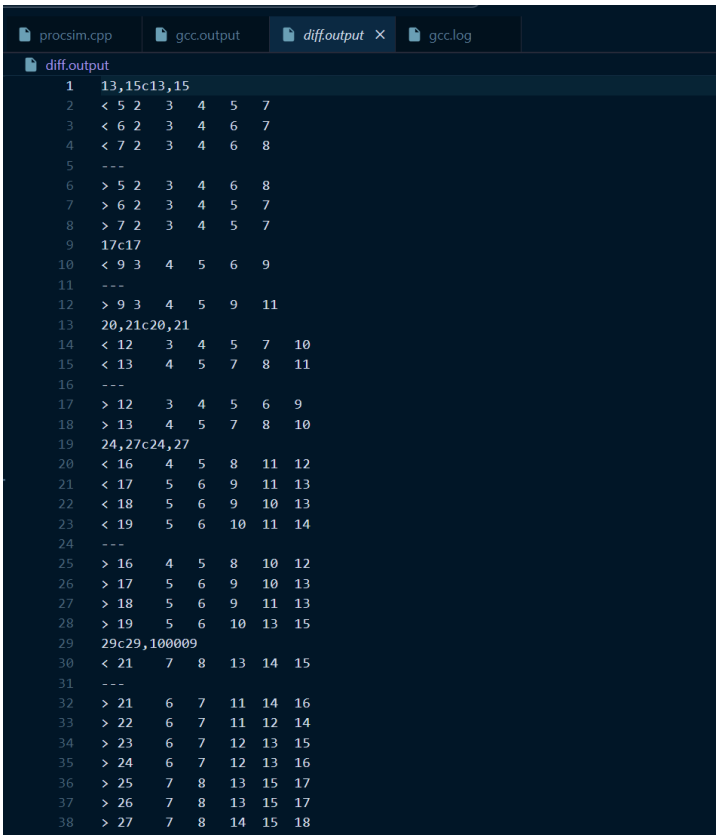
David Gotler

Experiment 1: Validate traces with output files.



This is the output of my backend implementation, just looking at the first 20 instructions of the gcc trace. It unfortunately does not match exactly. The timing does not match the output for the RAW hazard cases, for example instructions 2 and 5. There should be an extra cycle before INSTR 5 enters execution, to allow time for the reservation station values to update from the result bus.

I believe the problem has to do with the order in which scheduling and updating is done – when instructions are marked for retiring, firing and when the dependencies are updated. While I tried to follow the timing requirements of first half/second half operations, I must have missed something in the implementation.



Experiment 2

I unfortunately spent a lot of time trying to debug my code just to function properly, and didn't have enough time to run either of the full experiments, and so I did not try to optimize the architecture components.

For experiment 2, as the number of functional units and result buses are increased, the bandwidth or retire IPC will increase because these are the bottleneck in the original architecture.

Conclusion

While I didn't complete the assignment, I do feel I learned quite a lot through the process. Simulating each component of the pipeline stages solidified my understanding of the different components of the superscalar CPU architecture, as well as the Tomasulo algorithm for avoiding data flow hazards.