Disassembler Exceptions Report

by Team A: Orpal Nagra, Yuunbum Lim, Gihwan (Kris) Kwon

Project Exceptions

Despite being one of the groups that was classified as behind after the Week 2 Progress Report, we were able to finish the bulk of the project well before the deadline. This allowed us to have enough time to sufficiently test and find bugs in our program and fix them as they came up and work on putting the finishing touches on the final package to be submitted. We were able to complete a project that conforms to all of the required specifications and deliverables stated.

However there is one exception that is not graded but we acknowledge- simply stated, our disassembler could certainly be described as “fat.” As computer science students, it is painful to have so much code that could be optimized, however we were more focused with completing the project that satisfied all requirements within a timely manner and learning as much as possible along the way.

Our approach was not the same as most others- we did not split up the tasks to have one EA and one Opcode person. Rather, we all worked on Opcodes and EA. We split up the required supported functions and worked on them in parallel. This exposed everyone in our group to all the parts of the project and allowed us all to learn and practice both Opcode and EA which not only helped us with our individual understandings of the project’s components, but also gave us tons of practice for the final.

Now that we have all coded the functions in depth- in some cases with very similar code repeating- we understand the inner workings of what is going on. This means that optimization would be fairly easy- just time consuming and this is time we don’t have going into finals week.

Overall, we are very satisfied with our work- not because we met the requirements and got it done on time, rather because we all learned a TON from this project and took away lessons that we will never forget in our careers.

The project played a significant role in allowing us to implement concepts learned throughout the quarter and really see how code may work behind the scenes of seemingly simple statements such as: print(“Hello World”).

There is certainly something satisfying about accomplishing a complex project with a rather simple set of tools and available operations.

Note

- Last minute, we found a bug that affects the ADD.L and OR.L functions when the source EA is a long data value in the form #$XXXXXXXX. We were not able to fix this in time, however it if fairly minor and does not cause the program to crash.