Team 10: Norman Hansen, Konstantin Stekhov

Program Test Plan

We tested the program during each stage of its development process. For IO operations we tested the program by trying different addresses so that we could check how program validates them. First, we tried to figure out how the ASCII value is converted to HEX so when converting each character we checked the data register with the resulting HEX and compared its value to the input value. Next, we tested the input so it has the right length by saving the length in the data register and comparing it to the actual length of the input.

Finally, for testing the starting and ending addresses we looked at them so they are in the right range between 7000 and FFFFFE. Next, after testing all of the input validation we looked at A4 and A5 in order to see if the right addresses were stored.

We test the OP codes and EA modes by loading the test data into the program. In the test data file we have different instructions for each of the OP codes so we can see how all of the variations of the instructions are being decoded after running it. First, we have all the different possible MOVE instructions that need to be tested. We have a general approach of testing all of the different EA modes that the instructions could handle. After that we tested all of the different rotation instructions that the program could handle. Then we wrote the test instructions for arithmetic operations that the disassembler can handle which are ADD, ADDA, SUB, MULS, MULU, DIVS, and DIVU. When testing those we made sure that the we include both direct and indirect addressing modes. Then we moved on to the logical instructions such as OR, EOR, CMP.

All of the test instructions contain the cases where we check how they handle byte, word, and long operations depending on the instruction. We also made sure to test the cases where the instructions did not match to any of the commands that are handled by the disassembler so that it would write “DATA” after decoding them.

After we tested all of the separate parts of the instructions (OP codes, EA, and IO) we tested the program as a whole. The testing was performed by running the test data file and comparing the output in the console with the list of instructions in the test program. We made sure that the appropriate data register were used during each step of the decoding process by each part of the program so that there were no conflicts between them.

We did not use logging because we thought that it was unnecessary due to the fact that we could compare the output in the console and the list of instructions in the test file directly. Finally, we made sure that IO part of the program would be convenient for a user and added the necessary messages and formatting that we thought was appropriate.

In addition, we added comments to the subroutines that were confusing so that it would be clear how a particular operation in subroutine is performed.