Test Plan

Early Stage Testing

For EA testing, first, make a fake op_name for each different op to get in the different parsing subroutine. For each instruction, test the ea with the data register, address register direct, address register indirect, address register increment, address register decrement, immediate value with byte, word, long size, word size absolute address, and long size absolute address. Then compare the test output with the origin instruction command.

Most of my IO testing involved figuring how to read, the input file. I looked into the help file built into the simulator and read about the trap tasks #51, 53, and 56. Each opening, reading, and closes from a file respectively. Then I tested the code with several different file types. With .txt and .S68 files I could get the first 256 characters is file no problem. With a .cfg file it would append junk data that was not needed. I ended up using the address of where the read .cfg was being stored to get the starting and ending address and storing that in a variable that could be used by other subroutines.

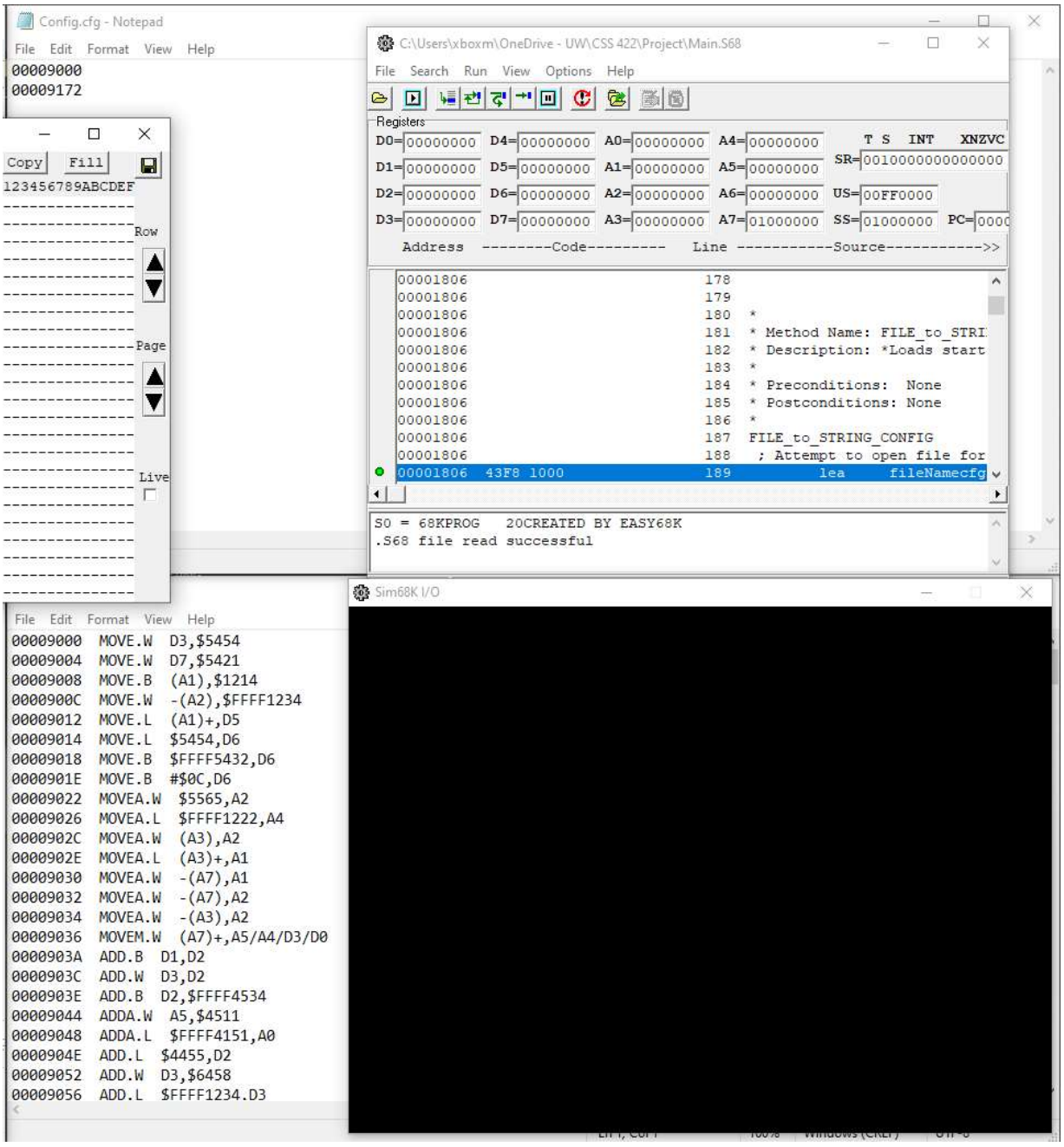
Early on, the OP portion of the code was tested by manually setting a starting address that pointed to opcodes at the end of the Main.X68 file. A loop subroutine was written which would later be called CLASSIFY_ADDRESS. This loop had a fixed increment to move to the next address to decode.

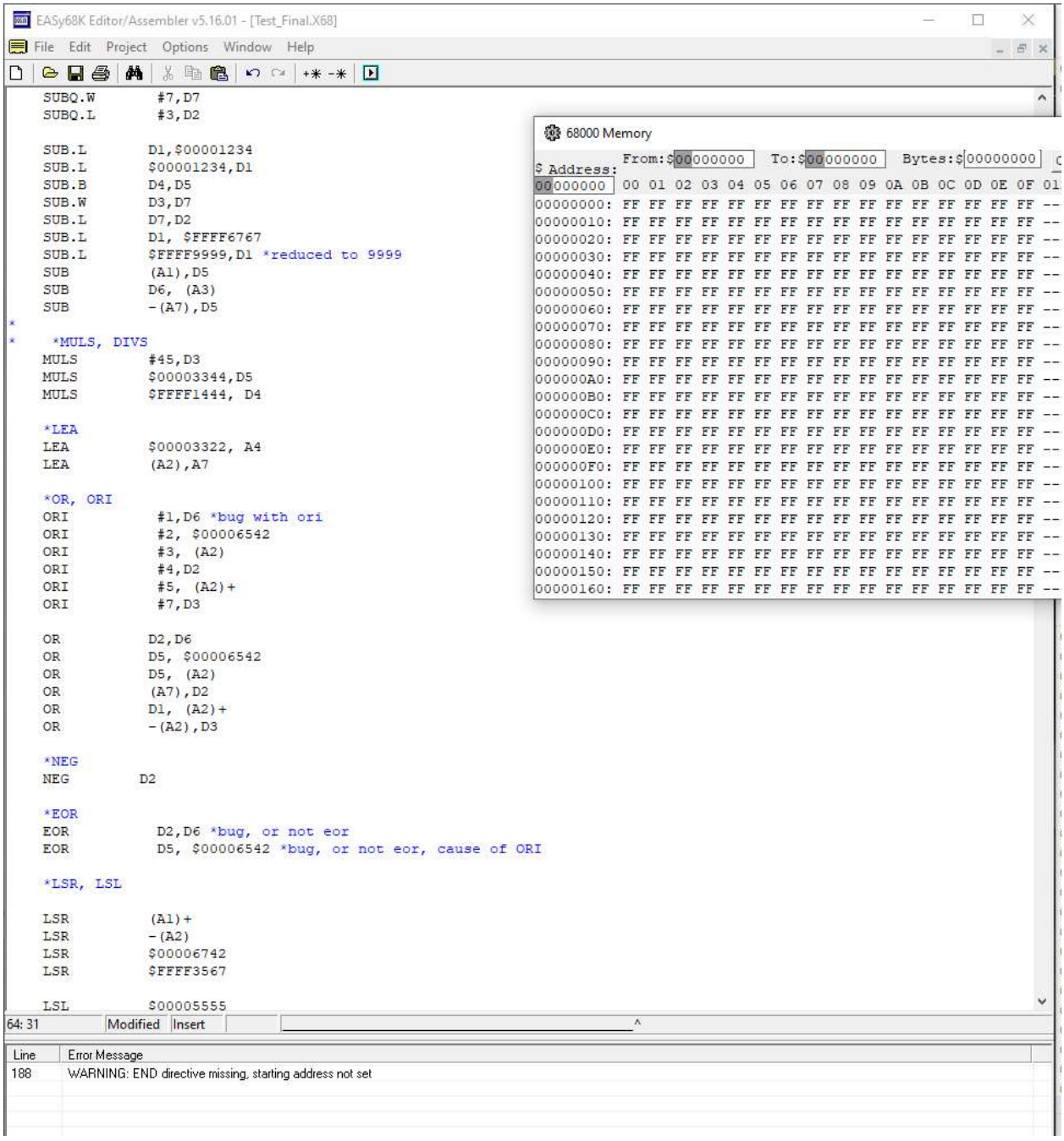
Finalized Testing

Once we successfully merged all code and had a semi-working product that just needed to be debugged, official testing had begun. We used the IO code to take in the .S68 file by loading data, and wrote a file called Test_Final.X68 (this file will be contained within our project submission). This file contains multiple versions of every opcode to be decoded. Once the file was compiled into a .S68 file, The Config.cfg file was changed so that the ending address lined up with the Test_Final final code.

When major bugs were still apparent, the code was manually stepped through to find said bugs. Then, as bugs lessened, the disassembler was ran fully and the Output.txt file was compared to the original Test_Final.X68 file. If a bug was not found, tests were commented out to make the debugging process easier. If a bug was found, a

note was written along with it’s address for further debugging. The Output.txt file would be cleared after each run. This process was repeated until no more bugs were found in the test file.




EASy68K Editor/Assembler v5.16.01 - [Test_Final.X68]
File Edit Project Options Window Help
SUBQ.W #7,D7
SUBQ.L #3,D2
SUB.L D1,$00001234
SUB.L $00001234,D1
SUB.B D4,D5
SUB.W D3,D7
SUB.L D7,D2
SUB.L D1, $FFFF6767
SUB.L $FFFF9999,D1 *reduced to 9999
SUB (A1),D5
SUB D6, (A3)
SUB -(A7),D5
* *MULS, DIVS
MULS #45,D3
MULS $00003344,D5
MULS $FFFF1444, D4
*LEA
LEA $00003322, A4
LEA (A2),A7
*OR, ORI
ORI #1,D6 *bug with ori
ORI #2, $00006542
ORI #3, (A2)
ORI #4,D2
ORI #5, (A2)+
ORI #7,D3
OR D2,D6
OR D5, $00006542
OR D5, (A2)
OR (A7),D2
OR D1, (A2)+
OR -(A2),D3
*NEG
NEG D2
*EOR
EOR D2,D6 *bug, or not eor
EOR D5, $00006542 *bug, or not eor, cause of ORI
*LSR, LSL
LSR (A1)+
LSR -(A2)
LSR $00006742
LSR $FFFF3567
LSL $00005555
64: 31 Modified Insert
68000 Memory
From:$00000000 To:$00000000 Bytes:$00000000
$ Address: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 01
00000000: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000010: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000020: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000030: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000050: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000060: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000070: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000080: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000090: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000C0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000D0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000E0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
000000F0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000100: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000110: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000120: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000130: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000140: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000150: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
00000160: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF --
Line Error Message
188 WARNING: END directive missing, starting address not set