Program Specifications

This program supports disassembly of the following instructions:

- MOVE, MOVEA, MOVEM
- ADD, ADDA
- SUB, SUBQ
- MULS, DIVS
- LEA
- OR, ORI
- NEG
- EOR
- LSR, LSL
- ASR, ASL
- ROL, ROR
- BCLR
- CMP, CMPI
- Bcc (BCS, BGE, BLT, BVC)
- BRA, JSR, RTS

And the following effective address modes:

- Data Register Direct: Dn
- Address Register Direct: An
- Address Register Indirect: (An)
- Address Register Indirect with Post incrementing: (A0)+
- Address Register Indirect with Pre decrementing: -(SP)
- Immediate Data: #
- Absolute Long Address: (xxx).L
- Absolute Word Address: (xxx).W

How to use:

To use this file, create a .cfg named Config.cfg. In this file, input the starting address of 00009000 on the first line. On the next line, input an ending address with a value great than the value of the starting address. Save the file to the same folder the disassembler is located. Run the already assembled disassembler program with the .S68 file extension. Once open, click the File drop down and click Open Data. From here, open the .S68 file you want to compile. After that is done, click the run button or press F9. Let program run to completion. After file is fully disassembled, output will be both on the console and in the Output.txt file.

Programming Standards

A3 is used to store opcodes.

D2 is used for comparisons.

A3 is used as improvised stack to print opcodes.

A6 is used to store the ending address.

A2 is used to append the output word.

foundNAME is the naming convention for decoding opcodes.

Space is reserved for variables at the top.