**Christian Sfeir**

**Lab experiment 4**

**Objectives:**

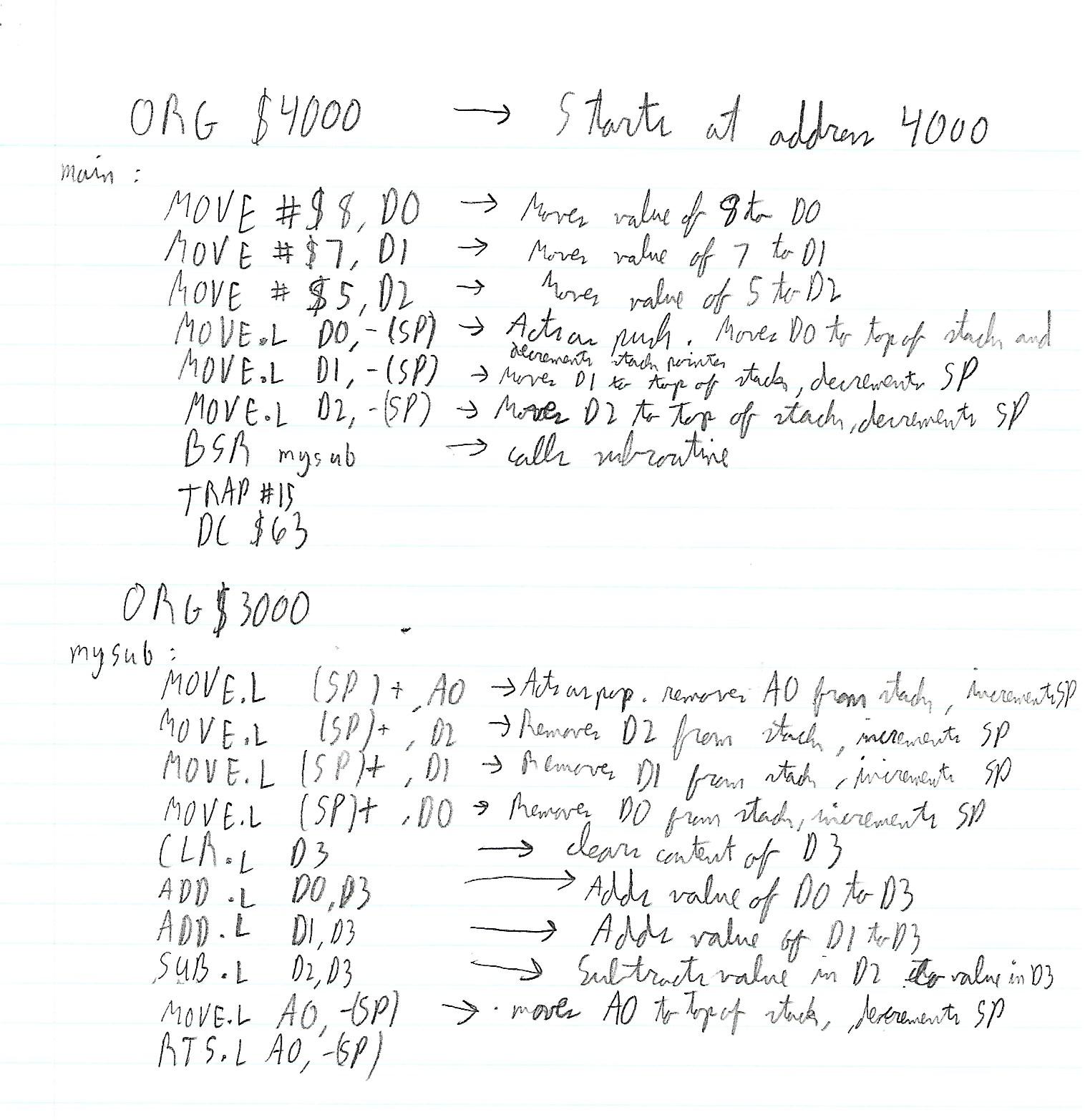
Subroutines will be investigated in this experiment.

**Introduction:**

Subroutines allow code to be easily reused multiple times in the same program without it needing to be constantly rewritten.

**Results:**

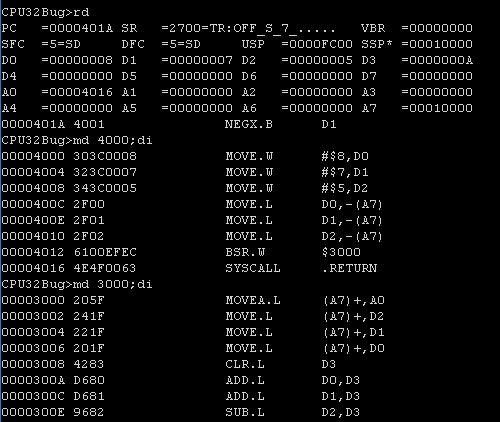
The following code was used in this experiment:



The following .lst file was obtained:

|  |
| --- |
| \* Experiment 4  \* Christian Sfeir  \* November 6th, 2019  00004000 ORG $4000  main:  00004000 303c 0008 MOVE #$8,D0  00004004 323c 0007 MOVE #$7,D1  00004008 343c 0005 MOVE #$5,D2  0000400c 2f00 MOVE.L D0,-(SP)  0000400e 2f01 MOVE.L D1,-(SP)  00004010 2f02 MOVE.L D2,-(SP)  00004012 6100 efec BSR mysub  00004016 4e4f TRAP #15  00004018 0063 DC $63    00003000 ORG $3000  mysub:  00003000 205f MOVE.L (SP)+,A0  00003002 241f MOVE.L (SP)+,D2  00003004 221f MOVE.L (SP)+,D1  00003006 201f MOVE.L (SP)+,D0  00003008 4283 CLR.L D3  0000300a d680 ADD.L D0,D3  0000300c d681 ADD.L D1,D3  0000300e 9682 SUB.L D2,D3  00003010 2f08 MOVE.L A0,-(SP)    00003012 4e75 RTS .L A0,-(SP)  ===== 0 Error(s)  ===== 0 Warning(s) |

Finally, the code produces the following output:



**Questions:**

1. A7 is assigned to the stack pointer
2. It will make the stack pointer go back to its original value (before the subroutine happened) which will overall take up less memory space
3. Using static or dynamic allocation (depending on the situation)

**Conclusion:**

The objective of this lab was to investigate subroutines. In order to accomplish this, a program which uses a subroutine that pops out the contents of a memory location, modifies it and pushes it back was designed and tested. This experiment was a success since the program ended up working correctly.