

Ethan Woycehoski

CSIS 3740

Prof. Jiang

10/22/25

Project Three Report

The objective of Project Three was to implement and test new instructions in a MARIE assembler and simulator. I implemented functions SKIPCOND, CLEAR, ADDI, and JUMPI. The CLEAR function resets the accumulator to zero. The ADDI function demonstrates indirect memory access and addition. The JUMPI function demonstrates indirect control transfer.

The SKIPCOND instruction was particularly important. It introduced conditional logic to the MARIE assembler. The SKIPCOND checks whether the accumulator is less than zero, equal to zero, or greater than zero. Depending on the condition specified, the instruction will skip the next instruction if the condition is true. For instance, when executing loop.asm, the trace shows that SKIPCOND ZE correctly skipped instructions until the accumulator reached zero, ensuring the loop executed the correct number of times.

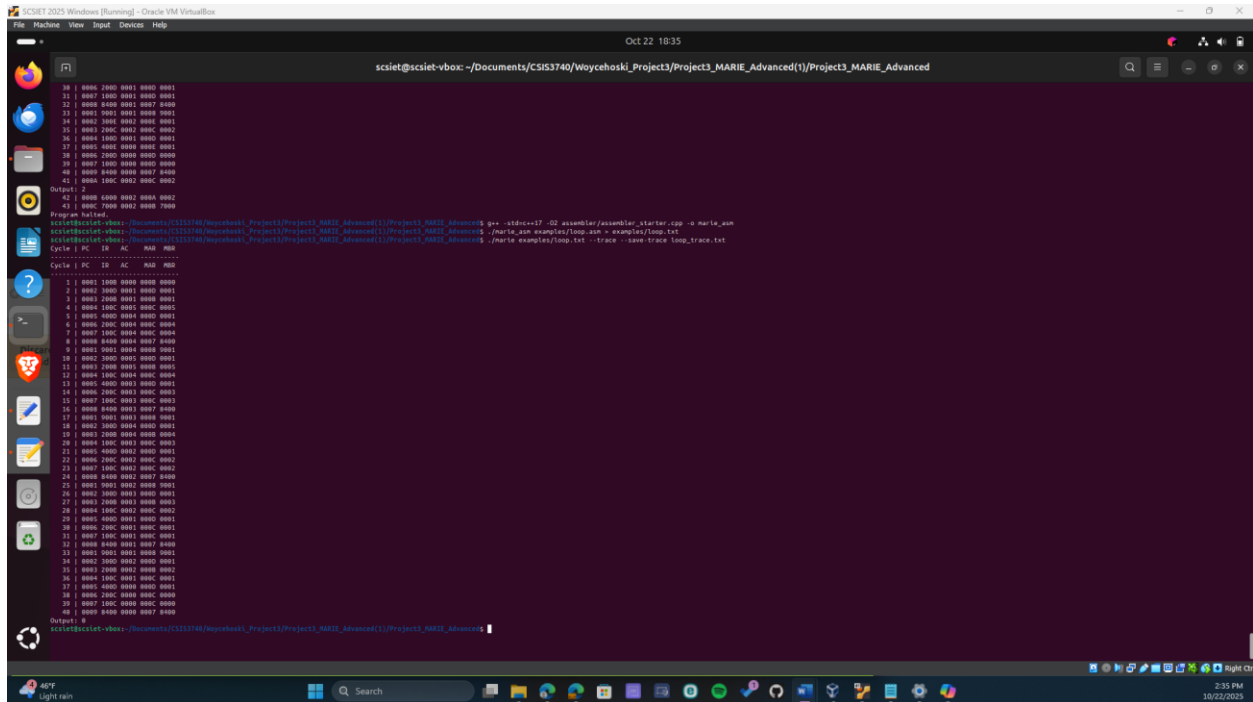
The assembler was able to accurately convert my human MARIE assembly into 16-bit hexadecimal machine code that the simulator can execute.

```
scsiet@scsiet-vbox: ~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/skipcond_test.asm > examples/skipcond_test.txt
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/skipcond_test.txt --trace --save-trace skipcond_trace.txt
xt
Cycle | PC  IR  AC  MAR MBR
-----
1 | 0001 1008 0002 0008 0002
2 | 0003 8800 0002 0001 8800
3 | 0004 1007 0000 0007 0000
Output: 0
4 | 0005 6000 0000 0004 0000
5 | 0006 7000 0000 0005 7000
Program halted.
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ g++ -std=c++17 -O2 assembler/assembler_starter.cpp -o marie_asm
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/loop.asm
100B
300D
200B
100C
400D
200C
100C
8400
9001
6000
7000
0000
0005
0001
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$
```

When running the simulator with trace enabled, the output confirmed that the SKIPCOND executed as intended.

```
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/skipcond_test.asm > examples/skipcond_test.txt
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/skipcond_test.txt --trace --save-trace skipcond_trace.txt
xt
Cycle | PC  IR  AC  MAR MBR
-----
1 | 0001 1008 0002 0008 0002
2 | 0003 8800 0002 0001 8800
3 | 0004 1007 0000 0007 0000
Output: 0
4 | 0005 6000 0000 0004 0000
5 | 0006 7000 0000 0005 7000
Program halted.
scsiet@scsiet-vbox:~/Documents/CSIS3740/Woycehoski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$
```

When testing the assembler again, this time with loop.asm, it again successfully displayed the results. The outputs and trace logs, which were saved during simulation, showed the accumulator values and instruction flow at each step of the program.



```
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced
Oct 22 18:35
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced

30 | 0005 2000 0001 0000 0001
31 | 0007 2000 0001 0000 0001
32 | 0008 0400 0001 0007 0000
33 | 0001 0001 0007 0000 0001
34 | 0002 0001 0002 0000 0001
35 | 0002 2000 0002 0000 0002
36 | 0004 1000 0001 0000 0001
37 | 0005 2000 0000 0000 0000
38 | 0006 2000 0000 0000 0000
39 | 0007 0000 0000 0000 0000
40 | 0008 0400 0007 0000 0000
41 | 000A 1000 0002 0000 0002
Output: 2
42 | 0000 0000 0002 0000 0002
43 | 0000 0000 0002 0000 0000
Program halted
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ g++ -std=c++17 -O2 assembler/assembler_starter.cpp -o marie_asm
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/loop.asm > examples/loop.txt
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$ ./marie_asm examples/loop.txt -trace -save trace loop_trace.txt
Cycle | PC | IR | AC | MAR | MBR
-----|---|---|---|---|---
1 | 0001 | 1000 | 0000 | 0000 | 0000
2 | 0002 | 2000 | 0001 | 0000 | 0001
3 | 0003 | 2000 | 0001 | 0000 | 0001
4 | 0004 | 1000 | 0001 | 0000 | 0001
5 | 0005 | 0400 | 0001 | 0000 | 0001
6 | 0006 | 0001 | 0002 | 0000 | 0001
7 | 0007 | 2000 | 0002 | 0000 | 0002
8 | 0008 | 0400 | 0007 | 0000 | 0000
9 | 0009 | 0001 | 0002 | 0000 | 0001
10 | 000A | 1000 | 0002 | 0000 | 0002
11 | 000B | 2000 | 0000 | 0000 | 0000
12 | 000C | 2000 | 0000 | 0000 | 0000
13 | 000D | 0000 | 0000 | 0000 | 0000
14 | 000E | 0400 | 0007 | 0000 | 0000
15 | 000F | 1000 | 0002 | 0000 | 0002
16 | 0010 | 0001 | 0002 | 0000 | 0001
17 | 0011 | 0002 | 2000 | 0000 | 0000
18 | 0012 | 2000 | 0000 | 0000 | 0000
19 | 0013 | 0004 | 1000 | 0003 | 0003
20 | 0014 | 0005 | 2000 | 0003 | 0003
21 | 0015 | 0007 | 1000 | 0003 | 0003
22 | 0016 | 0002 | 2000 | 0003 | 0003
23 | 0017 | 0004 | 1000 | 0003 | 0003
24 | 0018 | 0006 | 2000 | 0002 | 0002
25 | 0019 | 0007 | 1000 | 0002 | 0002
26 | 001A | 0008 | 0400 | 0007 | 0000
27 | 001B | 0001 | 0002 | 0000 | 0001
28 | 001C | 0002 | 2000 | 0000 | 0000
29 | 001D | 0004 | 1000 | 0003 | 0003
30 | 001E | 0005 | 2000 | 0003 | 0003
31 | 001F | 0007 | 1000 | 0003 | 0003
32 | 0020 | 0008 | 0400 | 0007 | 0000
33 | 0021 | 0001 | 0002 | 0000 | 0001
34 | 0022 | 0002 | 2000 | 0000 | 0000
35 | 0023 | 0004 | 1000 | 0003 | 0003
36 | 0024 | 0005 | 2000 | 0003 | 0003
37 | 0025 | 0007 | 1000 | 0003 | 0003
38 | 0026 | 0008 | 0400 | 0007 | 0000
39 | 0027 | 0001 | 0002 | 0000 | 0001
40 | 0028 | 0002 | 2000 | 0000 | 0000
Output: 8
sctiet@scsiet-vbox: ~/Documents/CSIS3740/Wojcechowski_Project3/Project3_MARIE_Advanced(1)/Project3_MARIE_Advanced$
```

In conclusion, this project demonstrated how instruction-level design affects program execution in MARIE. My added functions CLEAR, ADDI, JUMPI, and SKIPCOND were implemented and expanded the MARIE assembler to be able to deal with more instructions.