CS348-Assignment 2

Vatsal Gupta
200101105

README

20th January 2023

- **Name:** Vatsal Gupta
- **Roll No:** 200101105

Submission Contents

- README
- A2_200101105.cpp
- Assignment 2.pdf
- sample_input.txt
- sample_output.txt
- sample_intermediate.txt

Running the Code

Before “*&&*” is for compilation and after it is for running.

Command to run with generic input file

g++ -o ass2_200101105.exe A2_200101105.cpp && ass2_200101105.exe < ***(input file name)***

Example:

g++ -o ass2_200101105.exe A2_200101105.cpp && ass2_200101105.exe < sample_input.txt

```
>g++ -o ass2_200101105.exe A2_200101105.cpp && ass2_200101105.exe < sample_input.txt
```

Output files generated

- intermediate_file.txt
  The intermediate code is generated as in sample_intermediate, formatted as per the convention followed in the sample.
- machine_code_output.txt
  The final output created.
- optab.txt
  The opcode table of all relevant opcode mappings for SIC assembler are printed.
- symtab.txt
  The symbol table as generated is printed with labels/values and their corresponding addresses

  Kind Note: Name of each of these files can be changed in the function in which they are called in the code

```
1.void make_intm(string name, string start_addr, string prog_name)
//name is name of file
2.void generate_machine_code(string strFileName)
3.void print_opcodes(string filename)
4.void print_symbol_table(string filename)
```

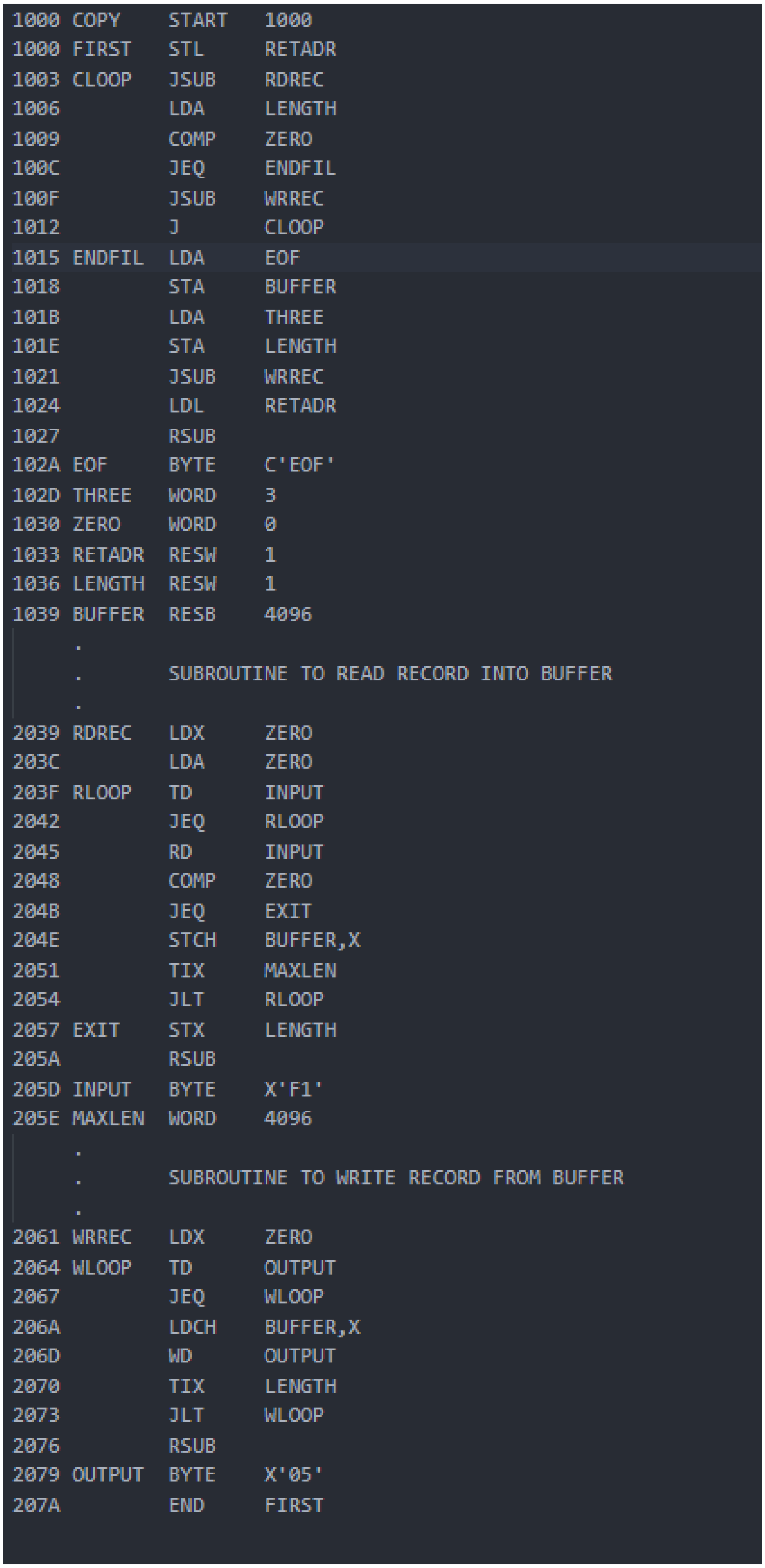
Code Sections

The code has the following major parts, marked by single comments or region tags which are collapsible with the VSCODE editor. Each of these sections are thoroughly explained in the comments of the code.

- Defining macros for conditional checking
- Hexadecimal and decimal conversions and addition handling
- Formatting functions
- Making sets for separating assembly directives from other functions
- Constructing the opcode table as a hash table
- Taking in the input of the entire file
- *Comment checking & Handling is done as BONUS.*
- Calculating loc_ctr or addressing and setting addresses
- Making the intermediate file
- Making a list of the object codes from the input file
- Printing the operation code table and the symbol table(created.)
- Generating the final machine code.

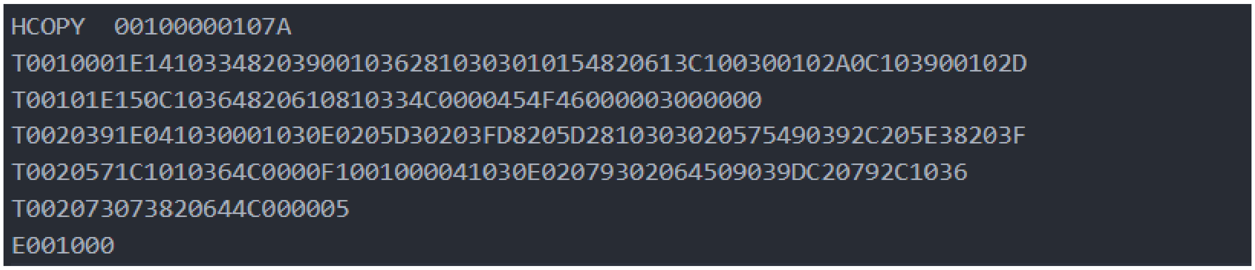
Example on sample:

1.intermediate_file.txt



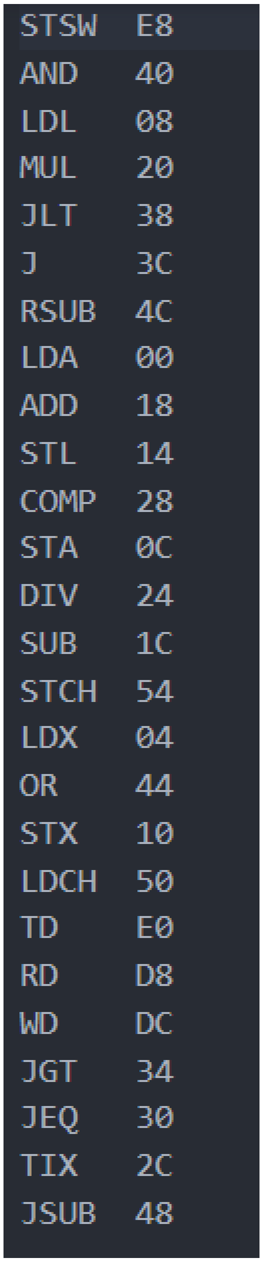
```
1000 COPY    START   1000
1000 FIRST   STL     RETADR
1003 CLOOP   JSUB    RDREC
1006         LDA     LENGTH
1009         COMP    ZERO
100C         JEQ     ENDFIL
100F         JSUB    WRREC
1012         J       CLOOP
1015 ENDFIL  LDA     EOF
1018         STA     BUFFER
101B         LDA     THREE
101E         STA     LENGTH
1021         JSUB    WRREC
1024         LDL     RETADR
1027         RSUB
102A EOF     BYTE    C'EOF'
102D THREE   WORD    3
1030 ZERO    WORD    0
1033 RETADR  RESW    1
1036 LENGTH  RESW    1
1039 BUFFER  RESB    4096
     .
     .       SUBROUTINE TO READ RECORD INTO BUFFER
     .
2039 RDREC   LDX     ZERO
203C         LDA     ZERO
203F RLOOP   TD      INPUT
2042         JEQ     RLOOP
2045         RD      INPUT
2048         COMP    ZERO
204B         JEQ     EXIT
204E         STCH    BUFFER,X
2051         TIX     MAXLEN
2054         JLT     RLOOP
2057 EXIT    STX     LENGTH
205A         RSUB
205D INPUT   BYTE    X'F1'
205E MAXLEN  WORD    4096
     .
     .       SUBROUTINE TO WRITE RECORD FROM BUFFER
     .
2061 WRREC   LDX     ZERO
2064 WLOOP   TD      OUTPUT
2067         JEQ     WLOOP
206A         LDCH    BUFFER,X
206D         WD      OUTPUT
2070         TIX     LENGTH
2073         JLT     WLOOP
2076         RSUB
2079 OUTPUT  BYTE    X'05'
207A         END     FIRST
```

2.machine_code_output.txt



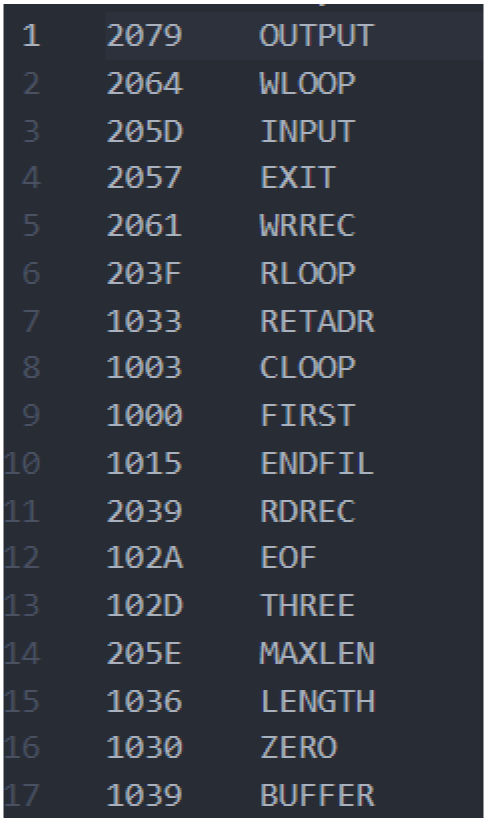
```
HCOPY  00100000107A
T0010001E141033482039001036281030301015482061 3C100300102A0C103900102D
T00101E150C1036482061081033 4C0000454F46000003000000
T0020391E041030001030E0205D30203FD8205D2810303020575490392C205E38203F
T0020571C1010364C0000F1001000041030E02079302064509039DC20792C1036
T002073073820644C000005
E001000
```

3.optab.txt



```
STSW  E8
AND   40
LDL   08
MUL   20
JLT   38
J     3C
RSUB  4C
LDA   00
ADD   18
STL   14
COMP  28
STA   0C
DIV   24
SUB   1C
STCH  54
LDX   04
OR    44
STX   10
LDCH  50
TD    E0
RD    D8
WD    DC
JGT   34
JEQ   30
TIX   2C
JSUB  48
```

4. symtab.txt



```
2079    OUTPUT
2064    WLOOP
205D    INPUT
2057    EXIT
2061    WRREC
203F    RLOOP
1033    RETADR
1003    CLOOP
1000    FIRST
1015    ENDFIL
2039    RDREC
102A    EOF
102D    THREE
205E    MAXLEN
1036    LENGTH
1030    ZERO
1039    BUFFER
```

NOTE:

- All mappings follow the SIC assembler and convention followed is as per the SIC Assembleras given in System Software An Introduction To Systems Programming (ref)
- The opcode convention has been written down for all operations which are available in the SIC assembler. This can be seen from the Appendix of the above ref.
- Code formats followed are as per the sample, exceeding the largest size of label/operand may cause this to overshoot and be misformatted.
- The assembly code provided is assumed to be correct and no error checking is employed, make sure that the input includes START and END as follows necessary conventions of the restricted SIC format.
- No extraneous lines should be present in the input assembly code file.
- ***BOUNS:*** Multi-Line Comment handling is done. To have comments in your asm code, make sure they begin with a ".", these codes appear in the intermediate file but play no role in machine code generation.

THANKS.