Alexandria University

# Final Project – Phase II

## SIC/XE assembler

Zahraa Emara – 4558
Sarah Yousry – 4582
Mayar El Mahdy – 4639
Mariam Beltagy – 5109

# I. Requirement specification

1. OBJECT FILE

```
1 HPROG    0003F00012
2 T000003F03002
3 T0000042039012
4 T00000450364
5 T000004876D617961726973
6 T000004F3FA1000
7 E0003F
```

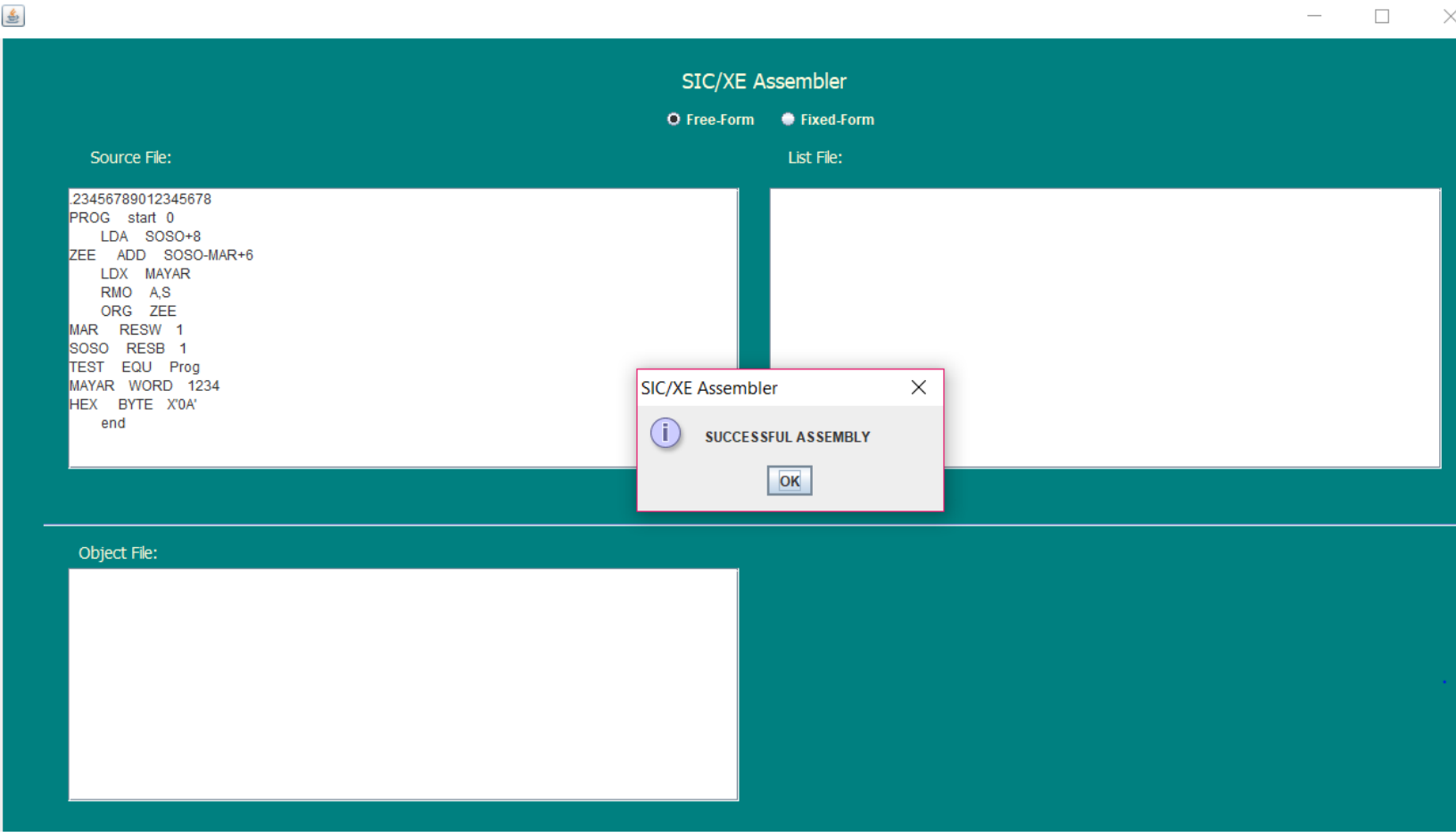
-Example of the object file

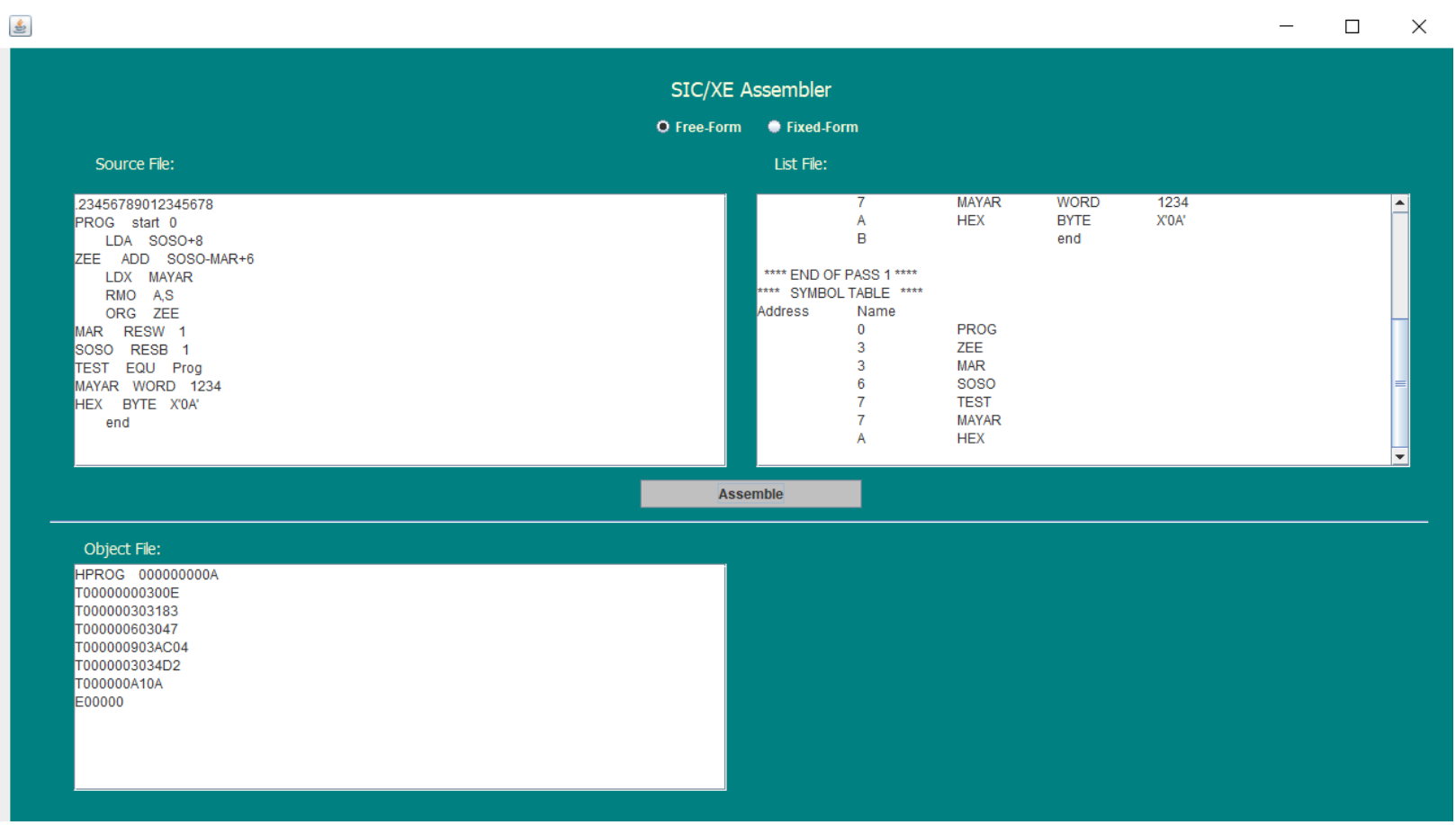
The object file is created by the three classes:

1- Header Record
2- Text Record
3- End Record

# II. Design

we used a simple GUI for the user to write the assembly code. When the user presses the Assemble button the List File and Object File are shown on the GUI.






# III. Main Data structures

- Arrays
- Adjacency Matrix

# IV. Algorithms Description

- After completing the symbol table –Which is the last thing in phase one – Phase two will begin by adding the –Header record—in the Object file.

```
if (flagError == 0) {
    bw.write("****   SYMBOL TABLE   ****");


    bw.newLine();
    Inst = "Address" + "\t\t" + "Name";
    bw.write(Inst);
    bw.newLine();
    for (int i = 0; i < symbol; i++) {
        Inst = "\t" + PCS[i] + "\t\t" + Labels[i];
        bw.write(Inst);
        bw.newLine();
    }
    if (state == 0) {
        header.WriteToFile(PCcount);
        text.WriteText(opCode, operands, index);
        end.WriteToFile(Integer.toHexString(header.PCstart).toUpperCase());
    }
}
```

as you can see here after checking that there is no error in phase one then the symbol table is written then first the header record, then every text record and then finally the end record.

- The **Header Record** has the program name –if it has one- else it just puts empty spaces from column 2 to 7 , then column 8 to 13 is the starting address of the program that the user wrote , finally column 14 to 19 is the length of the program which will be (Last address – Start address – 1) .

- The **Text Record** will be written depending on the opcode , so there is a 2D array that has all the operations and their opcodes every operation has similar ways to be written in the Object file , as first you write H following the starting address of this operation, length of this operation and then it's opcode , but –**WORD**—and –**BYTE**— are a bit different as in WORD it writes the starting address, it's length and the number defined but changed to hexadecimal, while BYTE has two cases either it is a character or a hexa-decimal string , if it is a character then each character will be converted to asci code –HEXA—, then it will be written as the WORD but the number defined will be the asci code of each character , If the BYTE is a hexa-decimal string then the hexadecimal will be added as the number defined with the same value unchanged. Operations such as RESB and RESW are skipped as they are not translated to object code. As for the operand conversion we first check if the given string contains '+' or '-' so the string is split, and the expression is evaluated using the ExpressionEvaluation() method.

- The **End Record** simply writes at the end of the object file E in Column 1 and the address of first executable instruction.

```java
public void WriteToFile(String line) {
    // TODO Auto-generated method stub
    PCstart = line;
    int length = PCstart.length();
    String l = new String();
    l = "E";
    while (length < 5) {
        l = l + "0";
        length++;
    }
    l = l + PCstart;
    try {
        if (l != null) {
            obj.newLine();
            obj.write(l);
            obj.close();
        }
    } catch (IOException e) {
        // TODO Auto-generated catch block
        e.printStackTrace();
    }
}
```

# V. Assumptions

- Assuming every operation done has its own text record – meaning that every line in the code except the start and end will have a full line of its own text record.
- Every number written in the object code is in hexa-decimal.

# VI. Sample Runs

- **Implementation of the FIXED format with some error examples:**

**Source File:**

```
.23456789012345678
PROG      START   0
           LDA    TEST
TEST      ADD     MAYAR
          RMO     A,Q
          AD      HEX
MAR       RESW    1
SOSO      RESB    1
TEST      EQU     Prog
MAYAR     WORD    1234
HEX       BYTE    X'0A'
```

**List File:**

```
**** SIC/XE Assembler ****
  0       PROG            START     0
  0                        LDA       TEST
          *****'missing or misplaced operation mnemonic '*****
  3       TEST            ADD       MAYAR
  6                       RMO       A,Q
          *****'illegal address for a register '*****
  9                       AD        HEX
          *****'unrecognized operation code '*****
  C       MAR             RESW      1
  F       SOSO            RESB      1
  10      TEST            EQU       Prog
          *****'duplicate label definition '*****
  13      MAYAR           WORD      1234
  16      HEX             BYTE      X'0A'
          *****' missing END statement '*****

**** END OF PASS 1 ****
```

- Implementation of the FIXED format (no errors):

**Source File:**

```
.23456789012345678
PROG      start   1234
          LDA     MAR-2
          ADD     SOSO+MAR+6
          LDX     MAYAR
          RMO     X,A
MAR       RESW    1
SOSO      RESB    1
TEST      EQU     Prog
MAYAR     WORD    1234
HEX       BYTE    X'0A'
          end
```

**List File:**

```
**** SIC/XE Assembler ****
  1234        PROG            start     1234
  1234                        LDA       MAR-2
  1237                        ADD       SOSO+MAR+6
  123A                        LDX       MAYAR
  123D                        RMO       X,A
  1240        MAR             RESW      1
  1243        SOSO            RESB      1
  1244        TEST            EQU       Prog
  1247        MAYAR           WORD      1234
  124A        HEX             BYTE      X'0A'
  124B                        end

**** END OF PASS 1 ****

**** SYMBOL TABLE *****
  Address     Name
  0           PROG
  1240        MAR
  1243        SOSO
  1244        TEST
  1247        MAYAR
  124A        HEX
```

**Object File:**

```
HPROG    0123400016
T00012340300123E
T000123703182489
T000123A03041247
T000123D03AC10
T0001240
T0001240
T000124010A
E01234
```

- **Implementation of the FREE format:**

**Source File:**

```
PROG     start    2233
           LDX     MAR-2
  ADD      SOSO+6
         LDT       100
         RMO         A,X
         +COMP      #5
   .THIS IS A COMMENT
         MAR         RESW    1
SOSO     RESB        1
TEST     EQU         Prog
MAYAR    WORD        1234
HEX      BYTE        X'0A'
         end
```

**List File:**

```
 **** SIC/XE Assembler ****
    2233        PROG    start   2233
    2233                LDX     MAR-2
    2236                ADD     SOSO+6
    2239                LDT     100
    223C                RMO     A,X
    223F                +COMP   #5

   .THIS IS A COMMENT
    2242        MAR     RESW        1
    2245        SOSO    RESB        1
    2246        TEST    EQU         Prog
    2246        MAYAR   WORD        1234
    2249        HEX     BYTE        X'0A'
    224A                end

  **** END OF PASS 1 ****
****   SYMBOL TABLE   ****
Address     Name
    0       PROG
    2242    MAR
    2245    SOSO
    2246    TEST
    2246    MAYAR
    2249    HEX
```

**Object File:**

```
HPROG    0223300016
T000223303042240
T000223360318224B
T00022390374
T000223C03AC01
T000223F
T000223F034D2
T000224210A
E02233
```

- An example of using ORG:

**Source File:**

```
1 .23456789012345678
2 PROG      START   10
3           LDA     MAR-2
4 MA        ADD     L
5           ORG     MA
6           ADD     SOSO+6
7 MAR       RESW    1
8 SOSO      RESB    1
9 MAYAR     WORD    1234
10 HEX      BYTE    X'0A'
11          end
```

**List File:**

```
1   **** SIC/XE Assembler ****
2
3 .23456789012345678
4     10      PROG        START       10
5     10              LDA     MAR-2
6     13      MA      ADD     L
7     16              ORG     MA
8     13              ADD     SOSO+6
9     16      MAR     RESW        1
10    19      SOSO        RESB        1
11    1A      MAYAR       WORD        1234
12    1D      HEX     BYTE        X'0A'
13    1E              end
14
15  **** END OF PASS 1 ****
16 ****   SYMBOL TABLE   ****
17 Address      Name
18    0         PROG
19    13        MA
20    16        MAR
21    19        SOSO
22    1A        MAYAR
23    1D        HEX
24
```

**Object File:**

```
1 HPROG    000100000D
2 T0000010030014
3 T000001303182
4 T000001303181F
5 T0000019034D2
6 T000001D10A
7 E00010
```