SIC XE 2-pass C++ Assembler

◆ Team members:

- **Ramy Wagdy Sobhy** [no.**21**]
- **Rofael Emil Fayez Behnam** [no.**22**]
- **Remon Hanna Wadie Youssef** [no.**23**]
- **Mohamed Ahmed Taher Mohamed Ahmed Elkholy** [no.**55**]
- **Mohamed Abd ElRahman ElFeki** [no.**59**]

◆ Requirements specifications

- implementing a 2-pass assembler for SIC XE Assembly, written in C/C++, producing code for the absolute loader used in the SIC programming assignments.

Specifications

- supports about **59** SIC XE architucture **instructions.**
- supports about **14** SIC XE architucture **pseudo-instructions** including [***EQU***, ***ORG***, ***LTORG***]
- supports about **10** registers.
- is executed through ***assemble.cpp*.**
- is to execute by entering "***assemble <source-file-name>*** " in OS shell.
- supports ***simple expression evaluation*** including operands [+, **-**, *****, **/**]
- supports **Literals** (Including ***LTORG***) in the forms [***=C'<ASCII-TEXT>'*** , ***=X'HEX-TEXT'***] [**Bonus Feature**]
- supports ***Control sections*** [**Bonus Feature**]

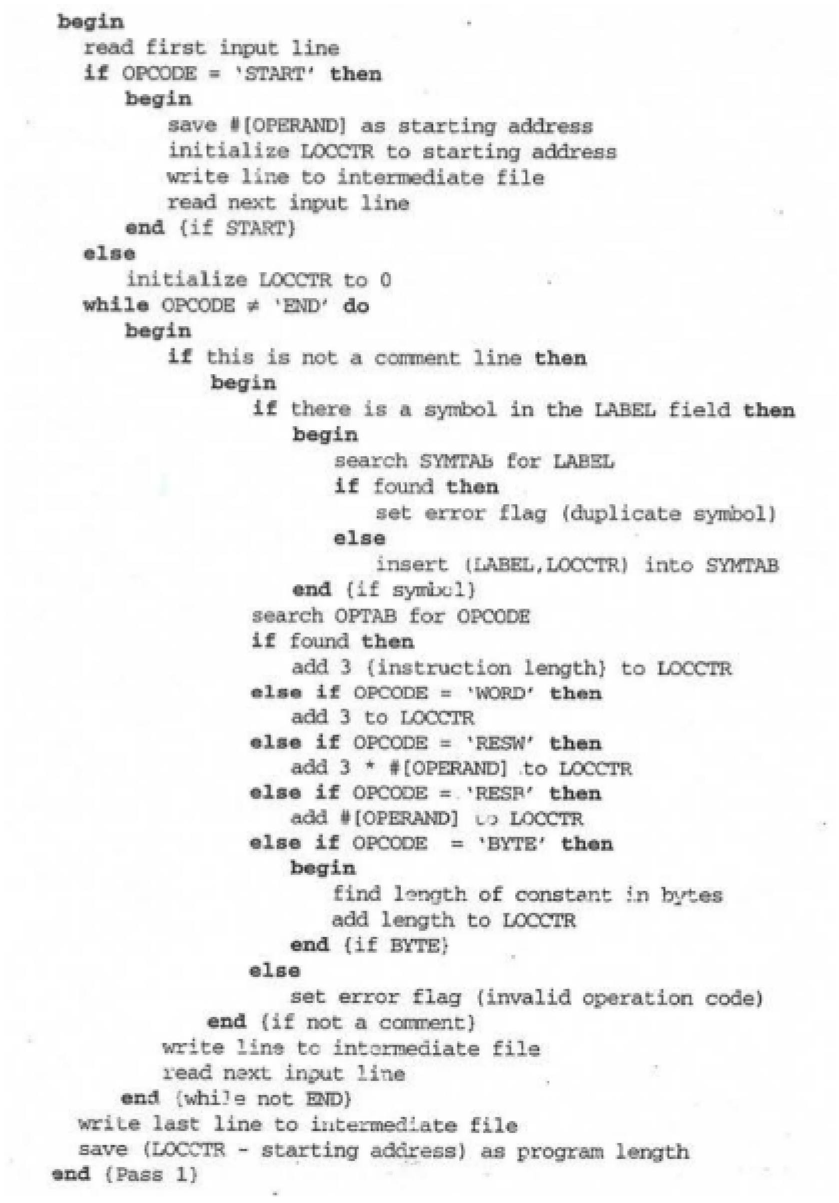
- **Design**
- ***9 C/C++*** code files
    - "***assemble.cpp***" [contains **main function** only]
    - "***SICXEPass1.cpp***" with header file "***SICXEPass1.h***"
      [performs **Pass1 algorithm**]
        - Input: ***input.txt***
        - Output: ***Intermediate_File.txt***
        - *Listing File* Array will be implemented as following:
          listingFile[0] -> **Line Number**
          listingFile[1] -> **Location Counter**
          listingFile[2] -> **Label _If found_**
          listingFile[3] -> **Operation Code**
          listingFile[4] -> **Flags**
          listingFile[5] -> **Operands**
          listingFile[6] -> **Comments _If found_**
          listingFile[7] -> **Object Code for pass2**
          listingFile[8] -> **Modification Boolean for pass2**
        - vector<string> *starts*; [address of **start of each control section**]
        - vector<string> *ends*; [address of **end of each control section**]
        - bool *pass1*(); [performs **pass1 algorithm** stated by Leland L. Beck]
        - void *generateIntermediateFile*(); [write to **O/P file**]
        - string ***evaluateExpression***(string expression);
        - bool ***isComment***(string line);
        - int *verifyLine*(string line); [checks **validity of code** statements]
        - vector<string> *split*(string s); [**parses** code statements **recursively**]
        - bool ***isLiteralFoundListingFile***(int lineNum, string literal);
        - vector<string> splitOperators(string expression);
    - "**SICXEPass2.cpp**" with its header file "**SICXEPass2.h**"
      [**pass2** implementation]
        - Input: **ListingFile** array generated by ***SICXEPass1.cpp***
        - Output: **Intermediate_File.txt**, **Object.o**
    - void *getObjectCode*(int line); [**generates Object Code** of each statement]
    - void *setFlags*(string operand,int line); [sets **n,i, x, b, p, e** instruction bits]
    - bool *pass2*(); [performs **pass2 algorithm** stated by Leland L. Beck]
    - void *flushRecord*(); [writes to **O/P file**]
    - bool *headerRecord*(string programName, string startAdd, string endAdd);
      [generates **H records**]
    - bool *flushTextRecord*(); [writes **T record to O/P file**]
    - bool *appendTextRecord*(string locationCounter, string objectCode, bool executable); [generates **T records**]

    - bool *endRecord*(); [generates **E records**]
    - bool *modificationRecord*(string address, string length, string details); [generates **M records**]
    - bool *addExternalDefinition*(string name, string address); [generates **D records**]
    - bool *addExternalReference*(string name); [generates **R record**]
- "***Tables.cpp***" associated with "***Tables.h***" [contains **main data structures**]
- "***Conversions.cpp***" associated with "***Conversions.h***" [contains ***ASCII-number-system*** conversions]
    - supports ***ASCII***, ***Binary***, ***Decimal*** and ***Hexadecimal*** 2-way conversions**.**

- **Main data structures**
    - Hash Table
- **SYMTAB:** map<string, vector<string> > symTab;
- **Registers:** map<string, int> registersTable;
- **OPTAB:** map<string, string> opTab;
    - Set
- **pseudo-instructions:** set<string> directivesSet;
    - 2-dimensional array
- **LITTAB** : string LITTAB[1000][4];

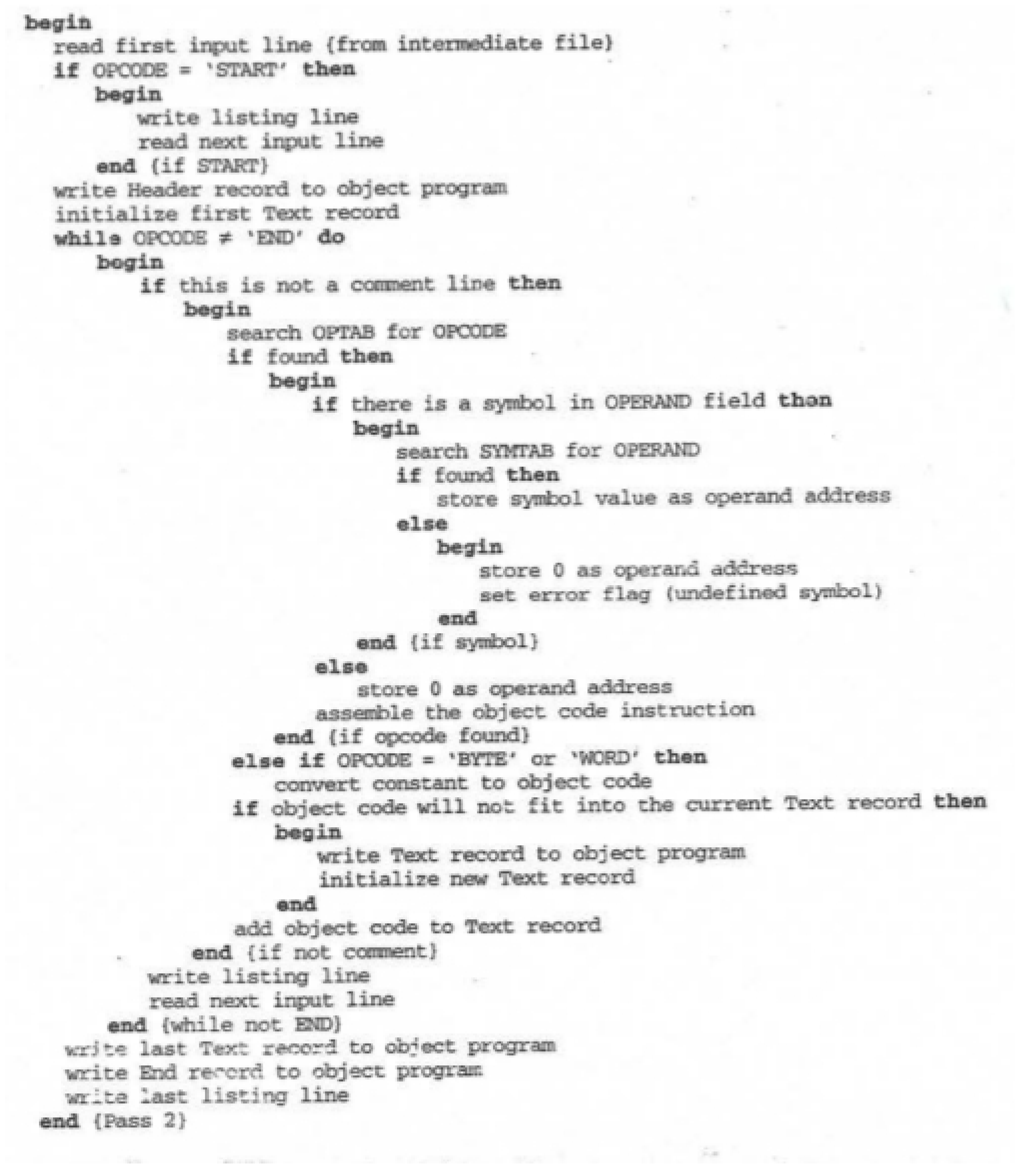
◆ **Algorithms description**
[**O(N)**, N is the count of ASCII characters in I/O file]

Pass1



```
begin
  read first input line
  if OPCODE = 'START' then
     begin
        save #[OPERAND] as starting address
        initialize LOCCTR to starting address
        write line to intermediate file
        read next input line
     end {if START}
  else
     initialize LOCCTR to 0
  while OPCODE ≠ 'END' do
     begin
        if this is not a comment line then
           begin
              if there is a symbol in the LABEL field then
                 begin
                    search SYMTAB for LABEL
                    if found then
                       set error flag (duplicate symbol)
                    else
                       insert (LABEL,LOCCTR) into SYMTAB
                 end {if symbol}
              search OPTAB for OPCODE
              if found then
                 add 3 {instruction length} to LOCCTR
              else if OPCODE = 'WORD' then
                 add 3 to LOCCTR
              else if OPCODE = 'RESW' then
                 add 3 * #[OPERAND] to LOCCTR
              else if OPCODE = 'RESB' then
                 add #[OPERAND] to LOCCTR
              else if OPCODE  = 'BYTE' then
                 begin
                    find length of constant in bytes
                    add length to LOCCTR
                 end {if BYTE}
              else
                 set error flag (invalid operation code)
           end {if not a comment}
        write line to intermediate file
        read next input line
     end {while not END}
  write last line to intermediate file
  save (LOCCTR - starting address) as program length
end {Pass 1}
```

pass2

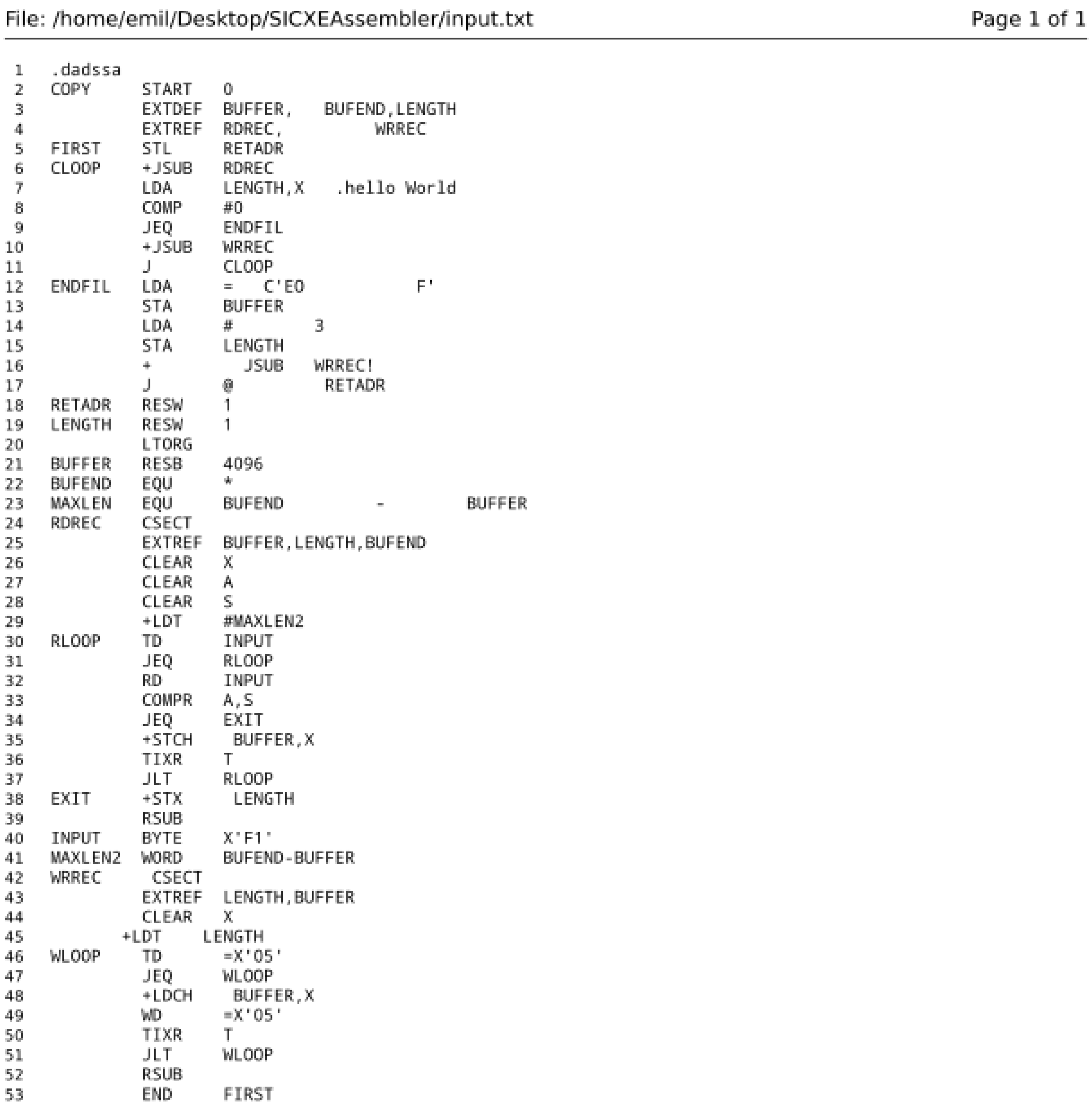


```
begin
  read first input line {from intermediate file}
  if OPCODE = 'START' then
     begin
        write listing line
        read next input line
     end {if START}
  write Header record to object program
  initialize first Text record
  while OPCODE ≠ 'END' do
     begin
        if this is not a comment line then
           begin
              search OPTAB for OPCODE
              if found then
                 begin
                    if there is a symbol in OPERAND field then
                       begin
                          search SYMTAB for OPERAND
                          if found then
                             store symbol value as operand address
                          else
                             begin
                                store 0 as operand address
                                set error flag (undefined symbol)
                             end
                       end {if symbol}
                    else
                       store 0 as operand address
                    assemble the object code instruction
                 end {if opcode found}
              else if OPCODE = 'BYTE' or 'WORD' then
                 convert constant to object code
              if object code will not fit into the current Text record then
                 begin
                    write Text record to object program
                    initialize new Text record
                 end
              add object code to Text record
           end {if not comment}
        write listing line
        read next input line
     end {while not END}
  write last Text record to object program
  write End record to object program
  write last listing line
end {Pass 2}
```

- **Assumptions**
- Free-formated I/O.
- Memory of target machine does not exceed **1 MB**.
- Hexadecimal literals must fill completely one or multiple byte-blocks.

- **TeamWord distribution**
- **Ramy Wagdy Sobhy**
    - Code statements structure verification [**pass1**]
    - geberating object code for each code statement [**pass2**]
- **Rofael Emil Fayez Behnam**
    - Free-format recursive parsing [**pass1**]
    - generating Object File Records [**pass2**]
- **Remon Hanna Wadie Youssef**
    - generating SYMTAB [**pass1**]
    - parsing EXTREF and EXTDEF operands [**pass2**]
    - QT-creator GUI with syntax highlighting feature
- **Mohamed Ahmed Taher Mohamed Ahmed Elkholy**
    - pass2 main algorithm [**pass2**]
    - QT-creator GUI with auto-compeletion feature
- **Mohamed Abd ElRahman ElFeki**
    - pass1 main algorithm [pass1]
    - generating LITAB
    - construction OPTAB [pre-design]

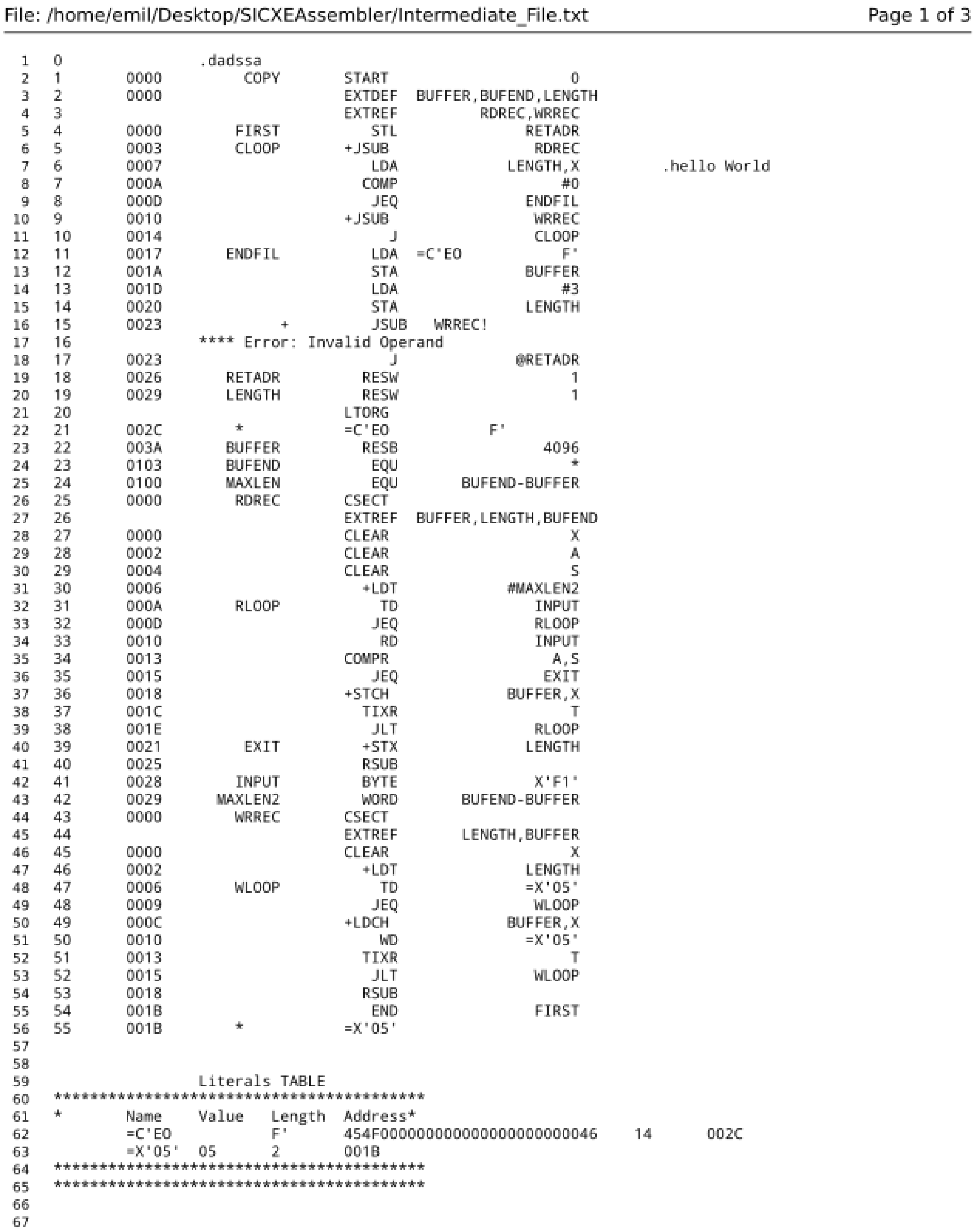
◆ Sample Runs



File: /home/emil/Desktop/SICXEAssembler/input.txt Page 1 of 1

```
.dadssa
COPY     START   0
         EXTDEF  BUFFER,   BUFEND,LENGTH
         EXTREF  RDREC,         WRREC
FIRST    STL     RETADR
CLOOP    +JSUB   RDREC
         LDA     LENGTH,X   .hello World
         COMP    #0
         JEQ     ENDFIL
         +JSUB   WRREC
         J       CLOOP
ENDFIL   LDA     =   C'EO           F'
         STA     BUFFER
         LDA     #        3
         STA     LENGTH
         +         JSUB   WRREC!
         J       @          RETADR
RETADR   RESW    1
LENGTH   RESW    1
         LTORG
BUFFER   RESB    4096
BUFEND   EQU     *
MAXLEN   EQU     BUFEND          -         BUFFER
RDREC    CSECT
         EXTREF  BUFFER,LENGTH,BUFEND
         CLEAR   X
         CLEAR   A
         CLEAR   S
         +LDT    #MAXLEN2
RLOOP    TD      INPUT
         JEQ     RLOOP
         RD      INPUT
         COMPR   A,S
         JEQ     EXIT
         +STCH    BUFFER,X
         TIXR    T
         JLT     RLOOP
EXIT     +STX     LENGTH
         RSUB
INPUT    BYTE    X'F1'
MAXLEN2  WORD    BUFEND-BUFFER
WRREC     CSECT
         EXTREF  LENGTH,BUFFER
         CLEAR   X
       +LDT    LENGTH
WLOOP    TD      =X'05'
         JEQ     WLOOP
         +LDCH    BUFFER,X
         WD      =X'05'
         TIXR    T
         JLT     WLOOP
         RSUB
         END     FIRST
```

Intermediate_File.txt



File: /home/emil/Desktop/SICXEAssembler/Intermediate_File.txt   Page 1 of 3

```
0               .dadssa
1       0000        COPY        START               0
2       0000                    EXTDEF  BUFFER,BUFEND,LENGTH
3                               EXTREF        RDREC,WRREC
4       0000       FIRST          STL          RETADR
5       0003       CLOOP        +JSUB          RDREC
6       0007                      LDA        LENGTH,X         .hello World
7       000A                     COMP              #0
8       000D                      JEQ          ENDFIL
9       0010                    +JSUB          WRREC
10      0014                        J          CLOOP
11      0017      ENDFIL          LDA   =C'EO          F'
12      001A                      STA          BUFFER
13      001D                      LDA              #3
14      0020                      STA          LENGTH
15      0023           +          JSUB   WRREC!
16              **** Error: Invalid Operand
17      0023                        J        @RETADR
18      0026      RETADR         RESW               1
19      0029      LENGTH         RESW               1
20                              LTORG
21      002C        *           =C'EO          F'
22      003A      BUFFER         RESB            4096
23      0103      BUFEND          EQU               *
24      0100      MAXLEN          EQU      BUFEND-BUFFER
25      0000       RDREC        CSECT
26                              EXTREF  BUFFER,LENGTH,BUFEND
27      0000                    CLEAR               X
28      0002                    CLEAR               A
29      0004                    CLEAR               S
30      0006                     +LDT        #MAXLEN2
31      000A       RLOOP           TD           INPUT
32      000D                      JEQ           RLOOP
33      0010                       RD           INPUT
34      0013                    COMPR             A,S
35      0015                      JEQ            EXIT
36      0018                    +STCH        BUFFER,X
37      001C                     TIXR               T
38      001E                      JLT           RLOOP
39      0021        EXIT         +STX          LENGTH
40      0025                     RSUB
41      0028       INPUT         BYTE           X'F1'
42      0029     MAXLEN2         WORD      BUFEND-BUFFER
43      0000       WRREC        CSECT
44                              EXTREF      LENGTH,BUFFER
45      0000                    CLEAR               X
46      0002                     +LDT          LENGTH
47      0006       WLOOP           TD          =X'05'
48      0009                      JEQ           WLOOP
49      000C                    +LDCH        BUFFER,X
50      0010                       WD          =X'05'
51      0013                     TIXR               T
52      0015                      JLT           WLOOP
53      0018                     RSUB
54      001B                      END           FIRST
55      001B       *            =X'05'


                Literals TABLE
*******************************************
*       Name    Value   Length  Address*
        =C'EO           F'      454F000000000000000000000046    14      002C
        =X'05'  05      2       001B
*******************************************
*******************************************


```

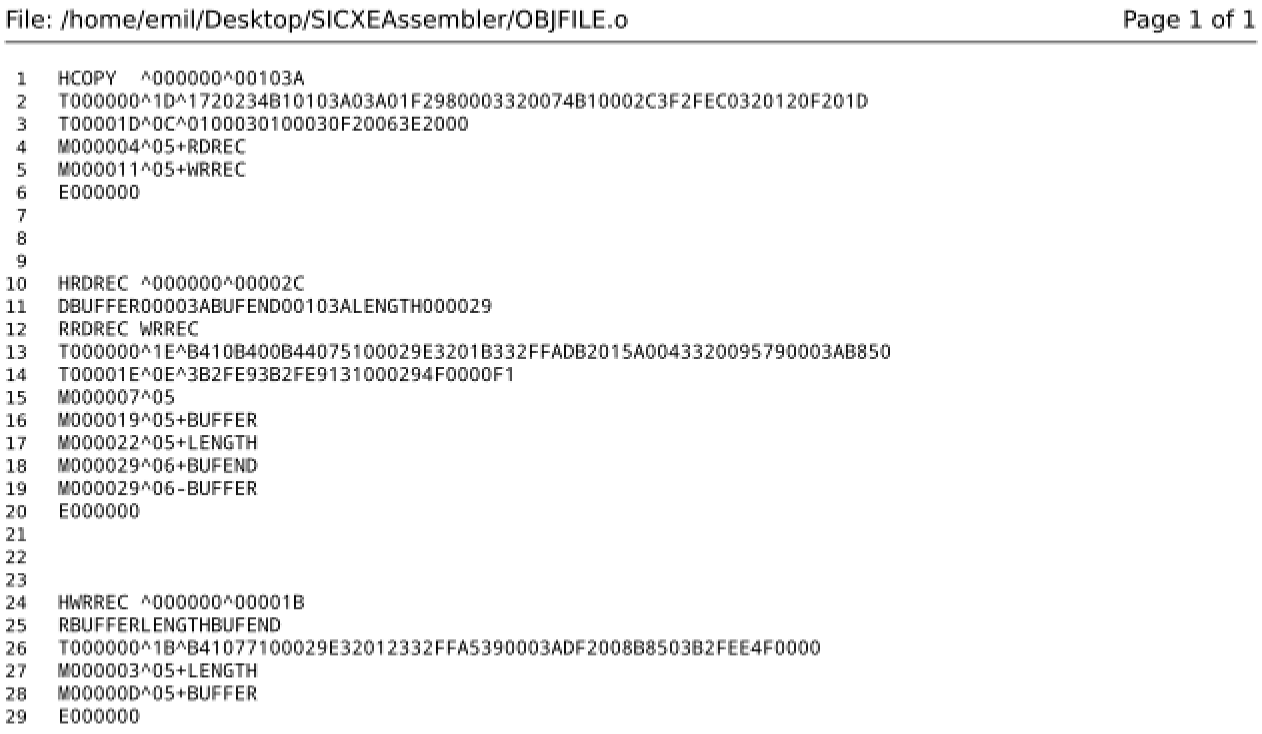
```
50      0010                    WD              =X'05'
51      0013                  TIXR                   T
52      0015                   JLT               WLOOP
53      0018                  RSUB
54      001B                   END               FIRST
55      001B        *         =X'05'


                 Literals TABLE
*****************************************
*       Name    Value   Length  Address*
        =C'EO           F'      454F000000000000000000000046    14      002C
        =X'05'  05      2       001B
*****************************************
*****************************************


        SYMBOL TABLE
**********************************************
*   SYMBOL      *    ADDRESS    *Section Numer
**********************************************
*BUFEND         |       103A |  0            *
*BUFFER         |       003A |  0            *
*CLOOP          |       0003 |  0            *
*COPY           |       0000 |  0            *
*ENDFIL         |       0017 |  0            *
*EXIT           |       0021 |  1            *
*FIRST          |       0000 |  0            *
*INPUT          |       0028 |  1            *
*LENGTH         |       0029 |  0            *
*MAXLEN         |       103A |  0            *
*MAXLEN2        |       0029 |  1            *
*RDREC          |       103A |  1            *
*RETADR         |       0026 |  0            *
*RLOOP          |       000A |  1            *
*WLOOP          |       0006 |  2            *
*WRREC          |       002C |  2            *
**********************************************
#End of pass 1 for SIC/XE Assembler#
Line    LocCtr    Labels      OpCode      Flags          Operands          Comments      ObjectCode   ModBoolean
0                .dadssa
1            0000          COPY       START                         0
2            0000                     EXTDEF                  BUFFER,BUFEND,LENGTH                            -----
3                                     EXTREF                  RDREC,WRREC                          -----
4            0000         FIRST         STL       110010         RETADR                            172023
5            0003         CLOOP       +JSUB       110001          RDREC                          4B10103A
6            0007                       LDA       111010       LENGTH,X         .hello World       03A01F
7            000A                      COMP       011000             #0                            298000
8            000D                       JEQ       110010         ENDFIL                            332007
9            0010                     +JSUB       110001          WRREC                          4B10002C

 10            0014                         J       110010          CLOOP                            3F2FEC
 11            0017        ENDFIL         LDA       110010       =C'EO           F'                             032012
 12            001A                       STA       110010         BUFFER                            0F201D
 13            001D                       LDA       010000             #3                            010003
 14            0020                       STA       110010         LENGTH                            0F2006
 15      0023            +          JSUB   WRREC!
 16              **** Error: Invalid Operand
 17            0023                         J       100010        @RETADR                            3E2000
 18            0026        RETADR        RESW                          1                            -----
 19            0029        LENGTH        RESW                          1                            -----
 20                                     LTORG       110000                                           ---
 21            002C           *         =C'EO           F'
 22            003A        BUFFER        RESB                       4096                            -----
 23            0103        BUFEND         EQU                          *                            -----
 24            0100        MAXLEN         EQU                BUFEND-BUFFER                            -----
 25            0000         RDREC       CSECT       110000                                           ---
 26                                    EXTREF                BUFFER,LENGTH,BUFEND                            -----
 27            0000                     CLEAR                          X                              B410
 28            0002                     CLEAR                          A                              B400
 29            0004                     CLEAR                          S                              B440
 30            0006                      +LDT       010001       #MAXLEN2                          75100029
 31            000A         RLOOP          TD       110010          INPUT                            E3201B
 32            000D                       JEQ       110010          RLOOP                            332FFA
 33            0010                        RD       110010          INPUT                            DB2015
 34            0013                     COMPR                        A,S                              A004
 35            0015                       JEQ       110010           EXIT                            332009
 36            0018                     +STCH       111001       BUFFER,X                          5790003A
 37            001C                      TIXR                          T                              B850
 38            001E                       JLT       110010          RLOOP                            3B2FE9
 39            0021          EXIT        +STX       110001         LENGTH                          13100029
 40            0025                      RSUB       110000                                         4F0000
 41            0028         INPUT        BYTE                     X'F1'                                F1
 42            0029       MAXLEN2        WORD                BUFEND-BUFFER                            -----
 43            0000         WRREC       CSECT       110000                                           ---
 44                                    EXTREF                LENGTH,BUFFER                            -----
 45            0000                     CLEAR                          X                              B410
 46            0002                      +LDT       110001         LENGTH                          77100029
 47            0006         WLOOP          TD       110010         =X'05'                            E32012
 48            0009                       JEQ       110010          WLOOP                            332FFA
 49            000C                     +LDCH       111001       BUFFER,X                          5390003A
 50            0010                        WD       110010         =X'05'                            DF2008
 51            0013                      TIXR                          T                              B850
 52            0015                       JLT       110010          WLOOP                            3B2FEE
 53            0018                      RSUB       110000                                         4F0000
 54            001B                       END                      FIRST                            -----
 55            001B           *        =X'05'                                                      000005
```

Object.o



File: /home/emil/Desktop/SICXEAssembler/OBJFILE.o   Page 1 of 1

```
HCOPY  ^000000^00103A
T000000^1D^1720234B10103A03A01F2980003320074B10002C3F2FEC0320120F201D
T00001D^0C^0100030100030F20063E2000
M000004^05+RDREC
M000011^05+WRREC
E000000



HRDREC ^000000^00002C
DBUFFER00003ABUFEND00103ALENGTH000029
RRDREC WRREC
T000000^1E^B410B400B44075100029E3201B332FFADB2015A0043320095790003AB850
T00001E^0E^3B2FE93B2FE9131000294F0000F1
M000007^05
M000019^05+BUFFER
M000022^05+LENGTH
M000029^06+BUFEND
M000029^06-BUFFER
E000000



HWRREC ^000000^00001B
RBUFFERLENGTHBUFEND
T000000^1B^B41077100029E32012332FFA5390003ADF2008B8503B2FEE4F0000
M000003^05+LENGTH
M00000D^05+BUFFER
E000000
```