Names:

Karim Mohamed Ali 32

Mazen Ahmed Zakaria 34

Michael Ramy Ramsis 35

Youssef Abdallah Youssef 62

Pass II: for the 2-pass assembler

Requirements specifications:

In this phase we build, we built on the previous phase and used its output to implement pass 2 of the assembler and generate object code and object program.

Design:

In this phase we added 4 additional classes to the previous phase

-Expression class: to handle simple expressions validation and evaluation.

-Utilities class: class to perform needed operations on strings from example, converting from one base to another, checking if the string is a valid hexadecimal string etc…

-Literal entry class: which will be stored in literal table, stores the value, length, name and address of a literal.

-The main module for pass 2 that keeps track of symbol table, literal table and listing table generated by pass 1 module and keeps track of the program counter and the base register. Also, responsible for executing pass 2 algorithm.

Main Data Structures used:

We extended some of phase 1 data structures such as

-The SYMTAB now contains flag for each symbol in the table to tell if it is relocatable address or absolute address.

-The Listing Table now stores a bitset representing the flags n,i,x,b,p,e for each instruction and a boolean to state if the instruction is generated by a literal such as WORD or BYTE

-The literal table for pass2 is an unordered_map of string as key and literal entry as value. The key is hexadecimal string representing the value of the literal.

-An external instance of the Utilities class to access all the helper methods in this class.

Algorithm Description

```
for each instruction in the src file {
    if (directive)  {
        if (BASE) set base flag to true
        else if (NOBASE) set base flag to false
        else if (WORD or BYTE) generate necessary object code
    }
    else if (format 1)                          // 8 bit opcode
       assemble opcode directly as 1 byte from the opcode table
    else if (format 2)                      // 8 bit opcode + 4 bit R1 + 4 bit R2
       assemble opcode + 4 bit register 1 + 4 bit register 2
    else if (format 3/4) {                  // 6 bit opcode + 6 bit flag + disp.
       if (operand is #number) {            // Immediate numeric (not #address)
          if (0 <= number <= 4095)          // Can it fit into 12 bits?
             assemble opcode + flags 010000 + 12 bit immediate value
          else if (4096 <= number <= 1048575 AND + before opcode)
             assemble opcode + flags 010001 + 20 bit immediate value
          else
             error - immediate number out of range
       } else if (operand is not blank) {  // PC, Base relative, Extended
          if (+ before opcode)              // Extended mode address requested
             assemble opcode + flags 110001 + 20 bit absolute address
          else if (-2048 <= TA-PC <= 2047) // PC AFTER instruction has been read
             assemble opcode + flags 110010 + 12 bit disp.   // 12 bit 2s comp.
          else if (BASE directive set AND 0 <= TA-B <= 4095) // Base relative
             assemble opcode + flags 110100 + 12 bit disp.   // Disp = TA - BASE
          else
             error - instruction addressing error
       } else                               // If operand is blank, as with RSUB
          assemble opcode + flags 110000 + 12 bits disp = 0  // RSUB = 4F0000
       if (# before operand)                // # is immediate mode
          set n bit to 0                    // changes nix from 110 to 010
       if (@ before operand)                // @ is indirect mode
          set i bit to 0                    // changes nix from 110 to 100
       if (line contains unexpected ,X)     // ,X is indexed mode (if unexpected)
          set x bit to 1                    // changes nix from xx0 to xx1
    }
}
```

Sample runs

test - Notepad

File Edit Format View Help

```
.23456789012345678900
COPY      START   0000
FIRST     STL     RETADR
          LDB     #LENGTH
          BASE    LENGTH
CLOOP     +JSUB   RDREC
          LDA     LENGTH
          COMP    #0
          JEQ     ENDFIL
          +JSUB   WRREC
          J       CLOOP
ENDFIL    LDA     EOF
          STA     BUFFER
          LDA     #3
          STA     LENGTH
          +JSUB   WRREC
          J       @RETADR
EOF       BYTE    C'EOF'
RETADR    RESW    1
LENGTH    RESW    1
BUFFER    RESB    4096
.
RLOOP     TD      INPUT
          JEQ     RLOOP
          TD      INPUT
          COMPR   A,S
          JEQ     EXIT
          STCH    BUFFER,X
          TIXR    T
          JLT     RLOOP
EXIT      STX     LENGTH
          RSUB
INPUT     BYTE    X'F1'
WRREC     CLEAR   X
          LDT     LENGTH
WLOOP     TD      OUTPUT
          JEQ     WLOOP
          LDCH    BUFFER,X
          WD      OUTPUT
          TIXR    T
          JLT     WLOOP
          RSUB
OUTPUT    BYTE    X'05'
          END     FIRST
```

test - Notepad

File Edit Format View Help

```
line no.    Address          Label          Mnemonic          Operands          Comment
1                          .2345678901234567890
2           0000             COPY              START             0000
3           0000             FIRST             STL               RETADR
4           0003                               LDB               #LENGTH
5           0006                               BASE              LENGTH
6           0006             CLOOP             +JSUB             RDREC
7           000A                               LDA               LENGTH
8           000D                               COMP              #0
9           0010                               JEQ               ENDFIL
10          0013                               +JSUB             WRREC
11          0017                               J                 CLOOP
12          001A             ENDFIL            LDA               EOF
13          001D                               STA               BUFFER
14          0020                               LDA               #3
15          0023                               STA               LENGTH
16          0026                               +JSUB             WRREC
17          002A                               J                 @RETADR
18          002D             EOF               BYTE              C'EOF'
19          0030             RETADR            RESW              1
20          0033             LENGTH            RESW              1
21          0036             BUFFER            RESB              4096
22                           .
23                           .SUBROUTINE TO READ INTO BUFFER
24          1036             RDREC             CLEAR             X
25          1038                               CLEAR             A
26          103A                               CLEAR             S
27          103C                               +LDT              #4096
28          1040             RLOOP             TD                INPUT
29          1043                               JEQ               RLOOP
30          1046                               TD                INPUT
31          1049                               COMPR             A,S
32          104B                               JEQ               EXIT
33          104E                               STCH              BUFFER,X
34          1051                               TIXR              T
35          1053                               JLT               RLOOP
36          1056             EXIT              STX               LENGTH
37          1059                               RSUB
38          105C             INPUT             BYTE              X'F1'
39          105D             WRREC             CLEAR             X
40          105F                               LDT               LENGTH
41          1062             WLOOP             TD                OUTPUT
42          1065                               JEQ               WLOOP
43          1068                               LDCH              BUFFER,X
44          106B                               WD                OUTPUT
45          106E                               TIXR              T
46          1070                               JLT               WLOOP
47          1073                               RSUB
48          1076             OUTPUT            BYTE              X'05'
49          1077                               END               FIRST
                          *** P A S S   1   E N D E D   S U C C E S S F U L L Y ***

          S Y M B O L   T A B L E
          ************************
          NAME                VALUE
          ************************
          OUTPUT               1076
          WLOOP                1062
          INPUT                105C
          FIRST                0000
          CLOOP                0006
          ENDFIL               001A
          RDREC                1036
          EOF                  002D
          RETADR               0030
          LENGTH               0033
          BUFFER               0036
          WRREC                105D
          RLOOP                1040
          EXIT                 1056
                          ************************************************
                          ********** S t a r t  o f  P a s s  II **********
LC        Code      Label     Opcode    Operand     Flags

0000                COPY      START     0000        n=0 i=0 x=0   b=0 p=0 e=0

0000      17202D    FIRST     STL       RETADR      n=1 i=1 x=0   b=0 p=1 e=0
```

```
000D      290000              COMP      #0          n=0 i=1 x=0   b=0 p=0 e=0
0010      332007              JEQ       ENDFIL      n=1 i=1 x=0   b=0 p=1 e=0
0013      4B10105D            +JSUB     WRREC       n=1 i=1 x=0   b=0 p=0 e=1
0017      3F2FEC              J         CLOOP       n=1 i=1 x=0   b=0 p=1 e=0
001A      032010    ENDFIL    LDA       EOF         n=1 i=1 x=0   b=0 p=1 e=0
001D      0F2016              STA       BUFFER      n=1 i=1 x=0   b=0 p=1 e=0
0020      010003              LDA       #3          n=0 i=1 x=0   b=0 p=0 e=0
0023      0F200D              STA       LENGTH      n=1 i=1 x=0   b=0 p=1 e=0
0026      4B10105D            +JSUB     WRREC       n=1 i=1 x=0   b=0 p=0 e=1
002A      3E2003              J         @RETADR     n=1 i=0 x=0   b=0 p=1 e=0
002D      454F46    EOF       BYTE      C'EOF'      n=0 i=0 x=0   b=0 p=0 e=0
0030              RETADR      RESW      1           n=0 i=0 x=0   b=0 p=0 e=0
0033              LENGTH      RESW      1           n=0 i=0 x=0   b=0 p=0 e=0
0036              BUFFER      RESB      4096        n=0 i=0 x=0   b=0 p=0 e=0
1036      B410    RDREC       CLEAR     X           n=0 i=0 x=0   b=0 p=0 e=0
1038      B400                CLEAR     A           n=0 i=0 x=0   b=0 p=0 e=0
103A      B440                CLEAR     S           n=0 i=0 x=0   b=0 p=0 e=0
103C      75101000            +LDT      #4096       n=0 i=1 x=0   b=0 p=0 e=1
1040      E32019    RLOOP     TD        INPUT       n=1 i=1 x=0   b=0 p=1 e=0
1043      332FFA              JEQ       RLOOP       n=1 i=1 x=0   b=0 p=1 e=0
1046      E32013              TD        INPUT       n=1 i=1 x=0   b=0 p=1 e=0
1049      A004                COMPR     A,S         n=0 i=0 x=0   b=0 p=0 e=0
```

```
1046      E32013              TD        INPUT       n=1 i=1 x=0   b=0 p=1 e=0

1049      A004                COMPR     A,S         n=0 i=0 x=0   b=0 p=0 e=0

104B      332008              JEQ       EXIT        n=1 i=1 x=0   b=0 p=1 e=0

104E      57C003              STCH      BUFFER,X    n=1 i=1 x=1   b=1 p=0 e=0

1051      B850                TIXR      T           n=0 i=0 x=0   b=0 p=0 e=0

1053      3B2FEA              JLT       RLOOP       n=1 i=1 x=0   b=0 p=1 e=0

1056      134000    EXIT      STX       LENGTH      n=1 i=1 x=0   b=1 p=0 e=0

1059      4F0000              RSUB                  n=1 i=1 x=0   b=0 p=0 e=0

105C      F1        INPUT     BYTE      X'F1'       n=0 i=0 x=0   b=0 p=0 e=0

105D      B410      WRREC     CLEAR     X           n=0 i=0 x=0   b=0 p=0 e=0

105F      774000              LDT       LENGTH      n=1 i=1 x=0   b=1 p=0 e=0

1062      E32011    WLOOP     TD        OUTPUT      n=1 i=1 x=0   b=0 p=1 e=0

1065      332FFA              JEQ       WLOOP       n=1 i=1 x=0   b=0 p=1 e=0

1068      53C003              LDCH      BUFFER,X    n=1 i=1 x=1   b=1 p=0 e=0

106B      DF2008              WD        OUTPUT      n=1 i=1 x=0   b=0 p=1 e=0

106E      B850                TIXR      T           n=0 i=0 x=0   b=0 p=0 e=0

1070      3B2FEF              JLT       WLOOP       n=1 i=1 x=0   b=0 p=1 e=0

1073      4F0000              RSUB                  n=1 i=1 x=0   b=0 p=0 e=0

1076      05        OUTPUT    BYTE      X'05'       n=0 i=0 x=0   b=0 p=0 e=0

1077                        END       FIRST       n=0 i=0 x=0   b=0 p=0 e=0

                    ***** S U C C E S S F U L L Y  A S S E M B L E D *****
```

test - Notepad

File Edit Format View Help

```
H^COPY   ^000000^001078
T^000000^1D^17202D^69202D^4B101036^032026^290000^332007^4B10105D^3F2FEC^032010
T^000020^1D^0F2016^010003^0F200D^4B10105D^3E2003^454F46^B410^B400^B440^75101000
T^001043^1D^E32019^332FFA^E32013^A004^332008^57C003^B850^3B2FEA^134000^4F0000^F1
T^00105F^1A^B410^774000^E32011^332FFA^53C003^DF2008^B850^3B2FEF^4F0000^05
E^000000
```