ASSEMBLER

Systems Programming

Final Project (phase-2)

Presented by:

- Monique Ehab 4928
- Haneen Ahmed 4741
- Hania Mohamed Hani 4697
- Claudia Kamal 4948
- Fagr Ahmed Refaat 4814

❖ Requirement Specification:

It is required to implement Phase-2 of a Assembler for SIC/XE machines.
Phase-2 specification requires that an entity is to be designed as follows:

1) Format:

A text file "format.txt" each line divided into 4 partitions:

a. Operation code
b. Number of bytes 1
c. Number of bytes 2
d. Memory/Register/Constant
e. Binary representation of op code.

2) Input:

A Source file's name "input.txt".

3) Process:

-The format file is read and each line partitioned and saved in a vector "record" after its category.
-The input source file is parsed in order to produce the output.

→Parsing process handles the following:

a) Source lines that are (instructions, storage declarations, comments, and assembler directives).

b) Errors and unhandled directives are handled with warnings and error messages.

c) For instructions, the parser is to minimally be capable of decoding 2, 3 and 4-byte instructions as follows:

- 2-byte with 1 or 2 symbolic register reference (e.g., TIXR A, ADDR S,A)
- RSUB (ignoring any operand or perhaps issuing a warning).
- 3-byte PC-relative with symbolic operand to include immediate, indirect, and indexed addressing.
- 3-byte absolute with non-symbolic operand to include immediate, indirect, and indexed addressing.
- 4-byte absolute with symbolic or non-symbolic operand to include immediate, indirect, and indexed addressing.
- The parser is to handle all storage directives (BYTE, WORD, RESW, and RESB).

4) Output:

a) Object code file:

A text file containing 3 records:

-Header Record: program name, starting address of program originally written in input file.

-Text Records: each contains the starting address and length of object code of this record.

- Modification Record: in case of format 4 instructions, stating location of the address field to be modified, relative to the beginning of the program and Length of address field to be modified , in half bytes.

-End Record: holds the address of first executable instruction in object code.

b) Output file:

-Symbol Table

-Source program in a format similar to a listing file.

-Extra column having the hexa representation of the

object code.

-A meaningful error message is printed below the line in which the error occurred.

❖ Design:

Pass-2 assembler:

- Source lines are read in sequence.
- The lines are passed to a parser method which supports free formatting.
- The parser method, along with other methods, is used as discussed before to output the pass-1 for the assembler and save the symbol table.
- If pass-1 was performed successfully pass-2 is then executed.
- In pass-2 every line entry is checked for producing the op-code.
- Errors in pass-2 are recorded so as to determine whether the object file will be created or not.

❖ Algorithms Description:

1) checkOrg:

- Parameters: String which is the operand for 'org' directive.
- Return type: Int which is the address to bs assigned for the current line.

2) checkEqu:

- Parameters: String which is the operand for the 'equ' directive, and a string which is the label for the 'equ' directive.
- Return type: Int which corresponds to the address to be assigned to this line.

3) formate1:

- Parameters: String which is an instruction of format 1.
- Return type: String holding the corresponding opcode.
- Functionality: A member of a set of functions designed to return an opcode for a certain instruction line according to its format used, format -1 is used in this case.

4) formate2:

- Parameters: String which is the instruction, and another string which is the operand to an instruction of format 2.
- Return type: A string holding the corresponding opcode. o Functionality: A member of a set of functions designed to return an opcode for a certain instruction line according to its format used, format -2 is used in this case, it reviews the registers -operands- and accordingly calculates the opcode.

5) formate3_4:

- Parameters: Integer which holds the index of the current line in the entries table, String which is the instruction, and

another string which is the operand to an instruction of format 2, and an integer holding the current address of the line.

- Return type: A string holding the corresponding opcode.
- Functionality: A member of a set of functions designed to return an opcode for a certain instruction line according to its format used, format -3 || 4 are used in this case, it reviews the instruction opcode, calculates the corresponding "nixbpe" values and evaluates the object code.

6) eval_address:

- Parameters: String which is the operand for a certain instruction.
- Return type: Returns integer which is address for corresponding operand.
- Functionality: Evaluates the address for all formats, it Is used to get the TA for a certain instruction whether from the symbol table or otherwise.

7) ObjectCode:

- Parameters: Void.
- Return type: Void .
- Functionality: Method responsible for assigning objectcodes for all source code lines in order to be printed out In the main method in pass-2. This is only when pass-1 is error free.

8) buildObjectFile:

- Parameters: Void.
- Return type: Void.
- Functionality: Method responsible for building up the object file if and only if pass-2 is error free.

9) is_number:

- Parameters: string to be checked.
- Return type: booolean.
- Functionality: validate that the eneterd string is a number.

10) generateObjectFile:

- Parameters: void.
- Return type: void.
- Functionality: writes the object code in object file.

11) tokenize:

- Parameters:
  -string: general expression.
  -char: delimiter.
- Return type: void.
- Functionality: separates the given string according to delimiter given.

12) initRegisters:

- Parameters: take no input.
- Return type: void.
- Functionality: gives each register its equivalent hexadecimal value.

13) modifyLocation :

- Parameters:
  -int: location of the instruction.
  -int: length of spaces it will fill in the object file.
- Return type: string.
- Functionality: changes the location of the instruction to the hexadecimal presentation and pads it with zeros to fill the corresponding spaces in the object file.

❖ Main Data Structures:

1) Maps:

a) A map data structure is used:

- For the symbol table:
  It is used to insert labels in the symbol table and to easily be able to retrieve them or check for their duplicity in constant time.
- For saving up SIC/XE machine appendix:

b) A two dimensional map is used to save up SIC/XE machine instruction set and their corresponding format.

❖ **Assumptions:**

1) Errors are produced as follows:

- In case of an invalid operand
  - "****Error: Invalid Operand".

- If line length is exceeded above limit
  - "****Error: Invalid length of the line".

- Invalid line spaces
  - "****Error: invalid spaces in this line".

- Invalid beginning
  - "****Error: invalid start of the program".

- Invalid op code
  - "****Error: Invalid OpCode".

- Duplicate symbols
  - "****Error: Duplicate Symbol".

- Invalid entry
  - "****Error: Invalid Entry".

- Invalid end program
  - "****Error: invalid end of the program".

2) Operands and Labels in the free format cannot include space characters.

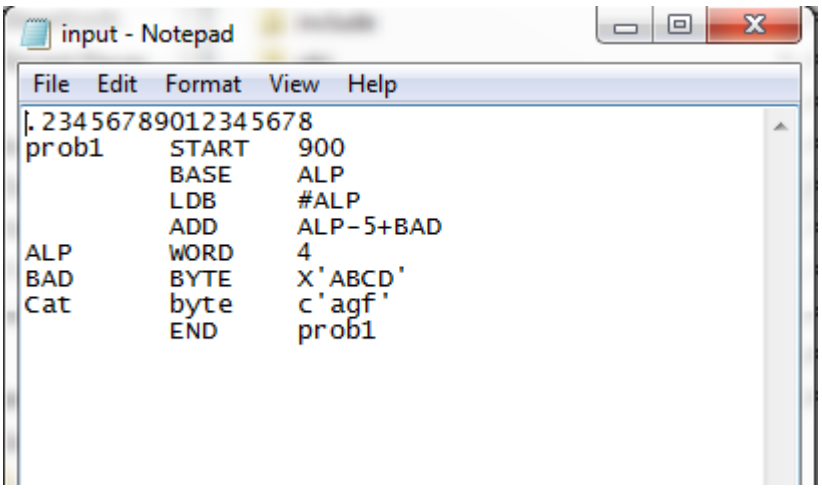
3) Dealing with a general expression:

- Assumption:
  -User must enter operators arranged according to priority of execution for the program to execute the code correctly.

- Procedure:
  -The general expression is separated using “tokenize” function, then checks the existence of the operands’ labels in the symbol table, then the expression is execute according to the input sequence.

❖ Sample Runs :

- **Input 1:**



- **Format file 1:**

format - Notepad

File Edit Format View Help

```
rmo,2,0,r,10101100
lda,3,4,m,000000
lds,3,4,m,011011
ldt,3,4,m,011100
ldx,3,4,m,000001
ldb,3,4,m,011010
ldl,3,4,m,000010
sta,3,4,m,000011
sts,3,4,m,011111
stt,3,4,m,100001
stx,3,4,m,000100
stb,3,4,m,011110
stl,3,4,m,000101
ldch,3,4,m,010100
stch,3,4,m,010101
add,3,4,m,000110
sub,3,4,m,000111
addr,2,0,r,10010000
subr,2,0,r,10010100
comp,3,4,m,001010
compr,2,0,r,10100000
j,3,4,m,001111
jeq,3,4,m,001100
jlt,3,4,m,001110
jgt,3,4,m,001101
```

- **Output file 1:**

output - Notepad

File Edit Format View Help

```
LineNo          Address          Label          Op-code                    Operand                    Comment                    ObjectCode
    0                                                                                          .23456789012345678
    1               900          prob1           start                        900
    2               900                           base                         alp
    3               900                            ldb                        #alp                                                692003
    4               903                            add                   alp-5+bad                                                1b4904
    5               906            alp            word                           4                                                000004
    6               909            bad            byte                     x'abcd'                                                  abcd
    7               90b            cat            byte                      c'agf'                                                616766
    8               90e                            end                       prob1
                 ********************pass1 errors********************




                 ********************Symbol Table********************

            Symbol                |                        Address
 ****************************|******************************
            alp                   |                          906
            bad                   |                          909
            cat                   |                          90b
```

- **Object file 1:**

objectfile - Notepad

File Edit Format View Help

```
Hprob1000090000000e
T0009000e6920031b4904000004abcd616766
E000900
```

- **Input 2:**

input - Notepad

File Edit Format View Help

```
.23456789012345678
pro       START   900
          LDS     BETA
          LDA     GAMMA
          ADDR    T,S
          SUBR    S,A
          +ADD    DELTA
ALPHA     RESW    1
DELTA     RESW    1
BETA      WORD    25
GAMMA     WORD    4
          END     pro
```

- **Output file 2:**

output - Notepad

File Edit Format View Help

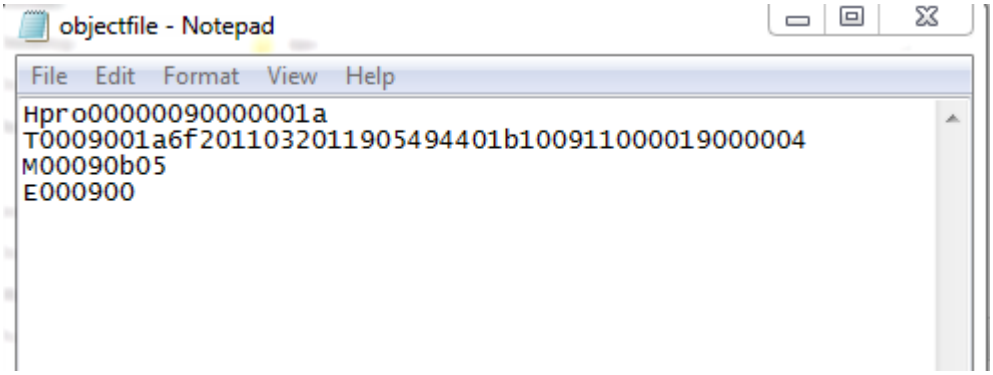
```
LineNo          Address          Label          Op-code                Operand                Comment                ObjectCode
     0                                                                                  .23456789012345678
     1              900            pro            start                    900
     2              900                             lds                   beta                                          6f2011
     3              903                             lda                  gamma                                          032011
     4              906                            addr                    t,s                                            9054
     5              908                            subr                    s,a                                            9440
     6              90a                            +add                  delta                                         1b100911
     7              90e          alpha            resw                      1
     8              911          delta            resw                      1
     9              914           beta            word                     25                                          000019
    10              917          gamma            word                      4                                          000004
    11              91a                             end                    pro
                  ********************pass1 errors********************



                  ********************Symbol Table********************

            Symbol                |                    Address
 ****************************** | ******************************
            alpha                 |                     90e
             beta                 |                     914
            delta                 |                     911
            gamma                 |                     917
```

- **Object file 2:**




```
Hpro00000090000001a
T0009001a6f20110320119054944 01b100911000019000004
M00090b05
E000900
```