phase 2

In two pass assembler, pass 2 is needed to generate data values defined by BYTE & WORD , perform processing of assembler directives not done in pass1, write the object code program and assembly listing

**Requirements specification:**

**By : Yousef Mohamed Fathy**

**Karim Atef**

**Mohamed Samy**

**Abdala ElSaman**

1. The assembler is to execute by entering: assemble <source-file-name>

2. The source file for the main program for this phase is to be named assemble.cpp

3. The output of the assembler should include (at least):

a) Object-code file whose format is the same as the one described in the text book in section

2.1.1 and 2.3.5.

b) A report at the end of pass2. Pass1 and Pass2 errors should be included as part of the

assembler report, exhibiting both the erroneous lines of source code and the error.

4. The assembler should support:

a) EQU and ORG statements.

b) Simple expression evaluation. A simple expression includes simple (A <op> B) operand

arithmetic, where <op> is one of +, -, \*, / and no spaces surround the operation, eg. A+B.

**Design :**

Class for pass 2 that take lines read from file and vector of location counters calculated in pass 1.

Two helper classes :

1) TargetAddress class : which calculate target address from operads and use symbol table formed in pass 1,

2) FlagBits class : which generate the 6 flag bits

ObjectProgramHandler class is used to write object codes in a file , he used object codes vector generated in pass2 class

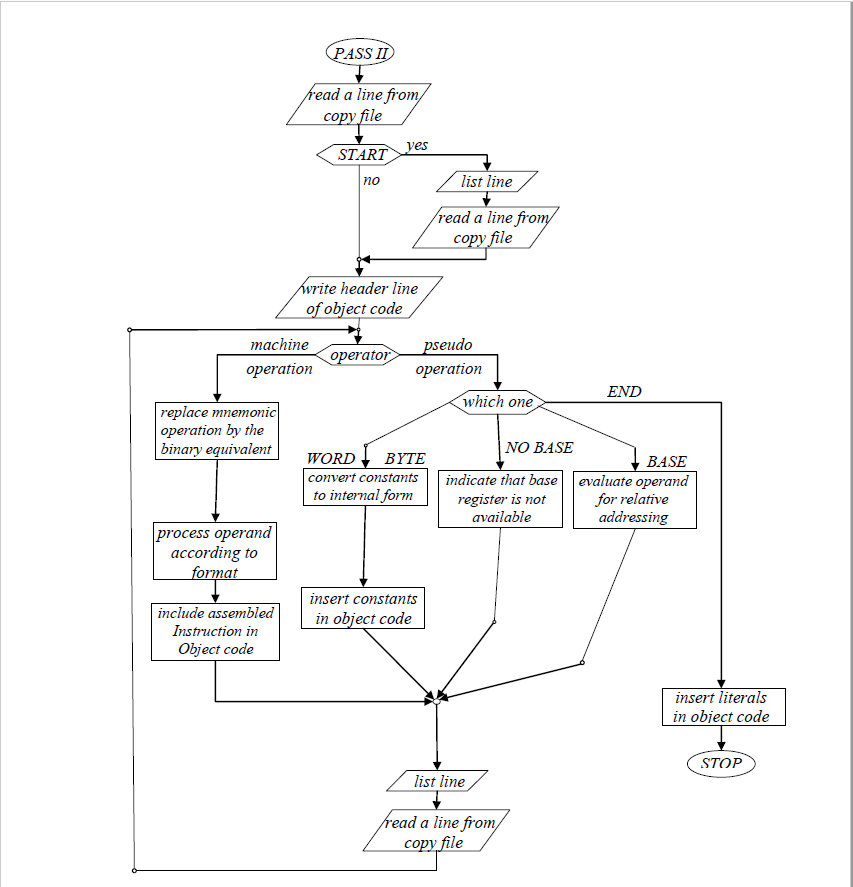
**Main data structures :**

Vector of pairs (string ,string ): used to save object code and address for each instruction

Vector program : used to get lines of program read from file in pass1 after parsing it

Vector addresses : to get addresses generated in pass1

**Algorithms description :**

****

**Assumptions (if any):**

**Sample runs :**

