Assembler

Introduce : a two-pass assembler will map the mnemonic of ELB816 machine code to Intel hex file.

Files description:

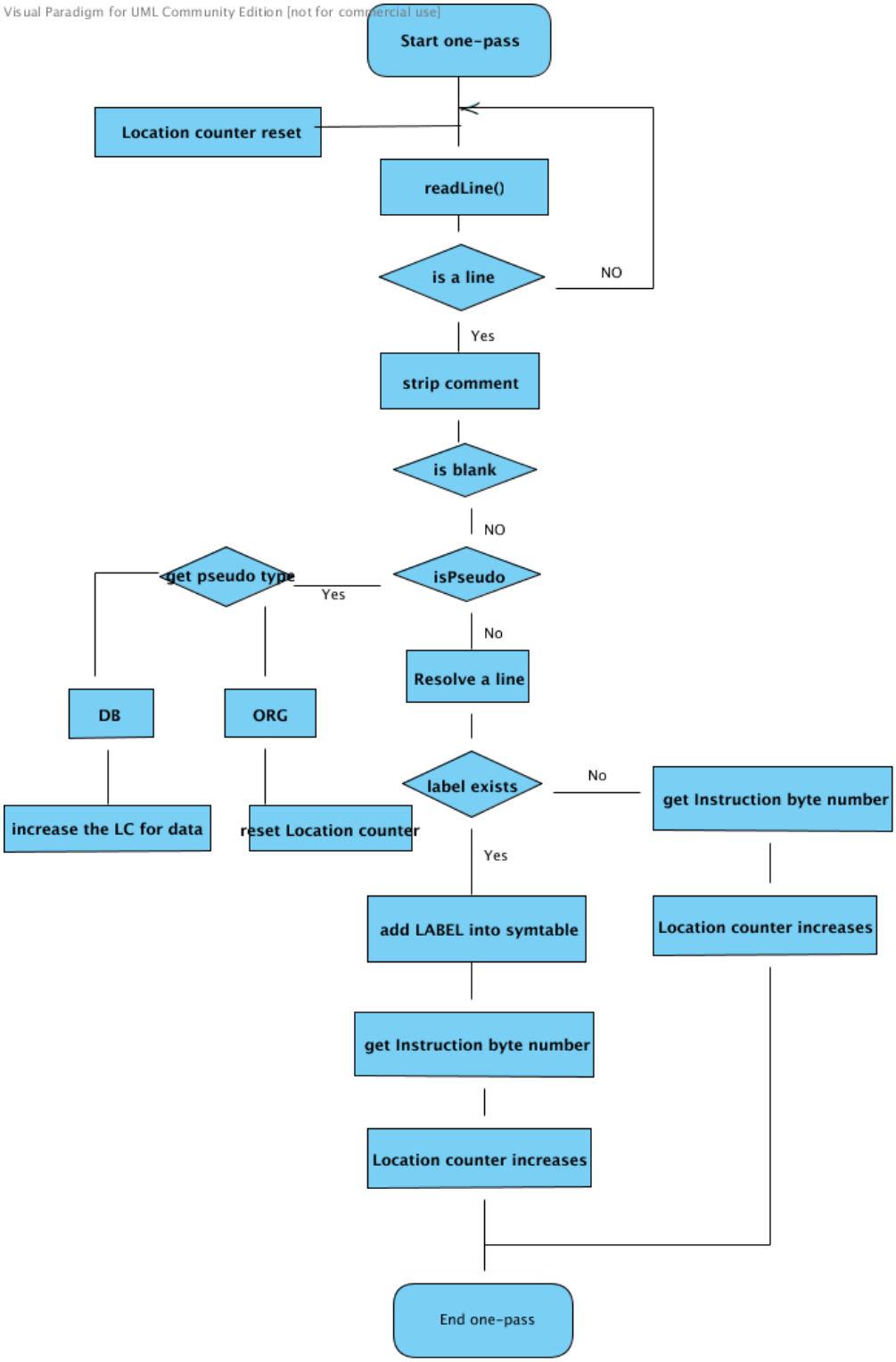
1. **assembler.py** is the main program that will do exactly to call subroutines to finish the assembly.
2. **analyzeSrc.py** is the subroutines to analyze the source code in some files like '.asm'
3. **opcode.txt** is the elb816 opcodes list in plain text version. It will be used to map some mnemonic to machine code we need.
4. **rlist** is the Reserved word list that contain all reserved word like 'A' , 'Mov', 'DB' those can't be viewed as label or data.
5. **orgsegclass.py** In the second pass of assembly, the program will use an instance of this orgsegclass to buffer 20-byte datas or opcodes so that it will make pseudo instruction easy to be processed.
6. **writehex.py** It contain some subroutines to convert the input into hex file format

Supporting instruction:

1. Real instruction : All
2. Pseudo instruction: ORG, DB, EQU, END

Main Procedures of program:

1. One-pass: assembler will create a symbol table by scanning all the source code. It will solve the forwarding reference problem.



2. Two-pass assembly