資訊三甲 10827145 黃霈昕 Document

所使用的開發平台及開發環境為Windows 10 & Dev C++

使用的程式語言為C/C++

所選擇的組合語言是SIC /SIC XE

**程式設計:**

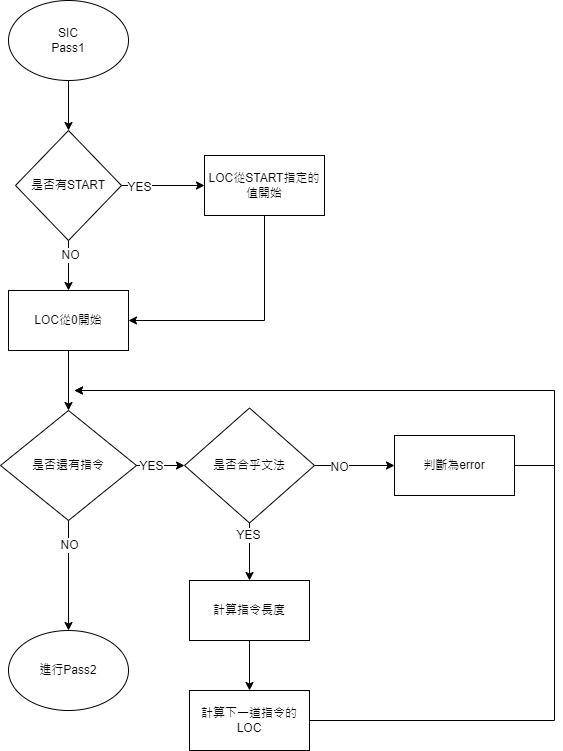
功能: 將程式作業一所切好的token進行分析、檢查文法錯誤，實作含指定pseudo instruction的Assembler，在pass1時給予合法的指令對應的位址，再經由pass2將指令轉為對應的object code ，最後將每行指令所對應的資訊寫成output 檔。

流程:

**SIC**

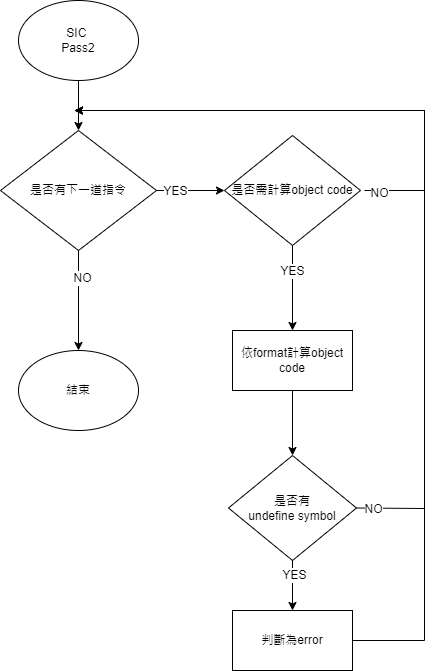
Pass1:

判斷第一行指令是否有START 這個pseudo instruction ，有便使用START 後的十進制數字作為程式起始位址，沒有則由0開始。接著依序讀入指令一邊判斷是否合法一邊計算指令長度得到下一道指令的起始位址，並且將symbol 及所對應的address 存入symbol table ，若遇到需要用到尚未定義symbol 的EQU時，先填入0x0。



Pass2:

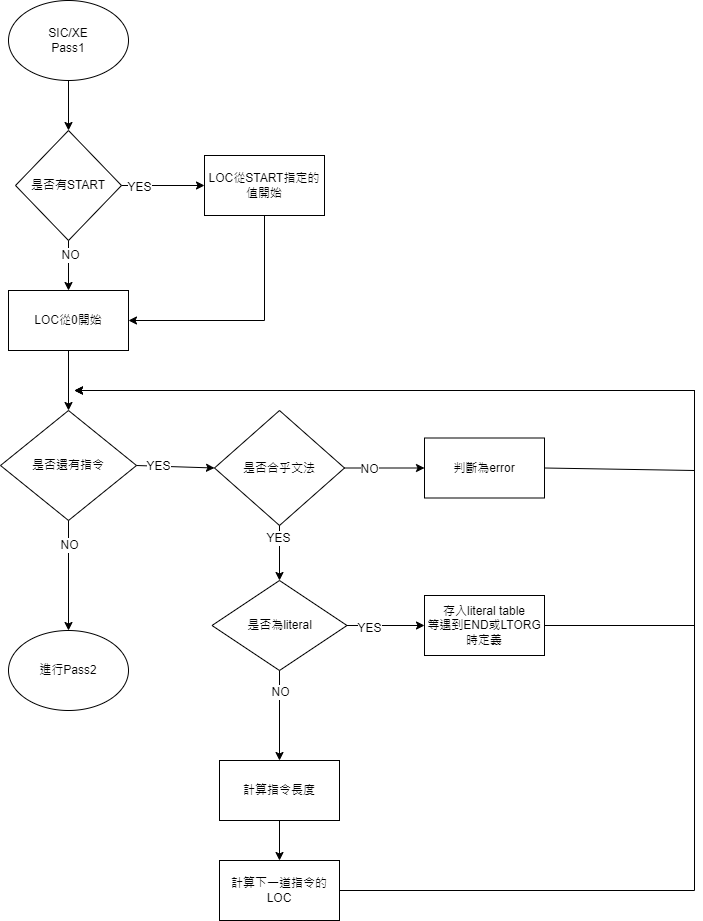
將有object code 的指令依format 及已定義的symbol 轉換為object code，若有未定義的symbol 則判斷為錯誤，若遇到pass1未完成定義的EQU時，將其定義，若symbol還是未定義則為錯誤。



**SIC/XE**

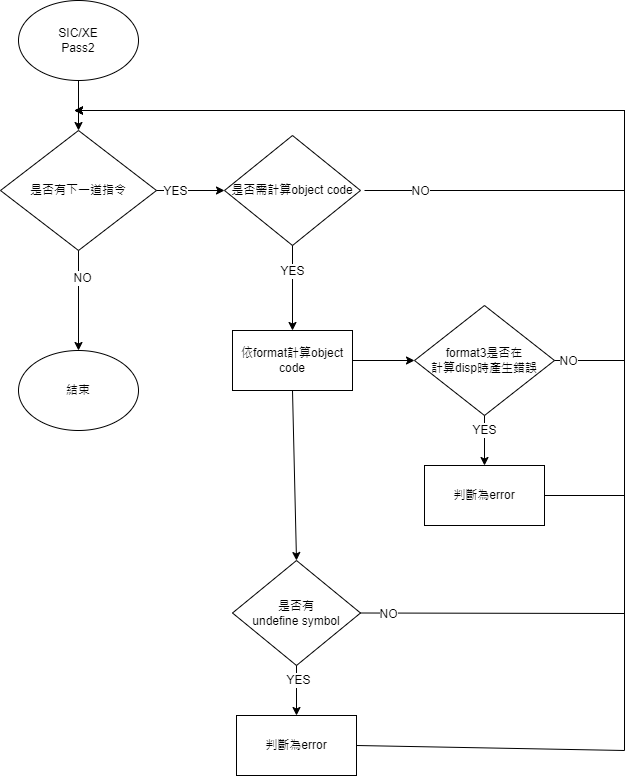
Pass1:

判斷第一行指令是否有START 這個pseudo instruction ，有便使用START 後的十進制數字作為程式起始位址，沒有則由0開始。接著依序讀入指令一邊判斷是否合法一邊判斷計算指令長度得到下一道指令的起始位址，並且將symbol 及所對應的address 存入symbol table ，若遇到需要用到尚未定義的symbol 的EQU時，先填入0；若遇到literal 則將literal放入literal table，當遇到END或LTORG時再給予literal table中尚未定義的literal位址；當出現BASE時將BASE後的symbol 設定為base register ，以供pass2時format 3需要用到base register 時使用。



Pass2:

將有object code 的指令依format 及已定義的symbol 轉換為object code，若為format 3指令，要先算disp，也就是pc-後面symbol 的位址，若disp小於-2048或者大於2047就要以base register 計算，若沒有設定base register 則判斷為錯誤，若用base register 計算的disp小於0或者大於4095也是為錯誤；若有未定義的symbol 則判斷為錯誤；若遇到pass1未完成定義的EQU時，將其定義，若symbol 還是未定義則為錯誤。

****

data structure:

TokenPtr為程式作業一切好的Token，存放每個Token的資訊(table, num)

Table1-4為指定Table，Table5放Symbol，Table6放Integer/Real，Table7放String，Table5-7皆以mod100的hash function放入範圍為0-99的array中。

Symbol Table(symtab)為放置Symbol及其對應位址的array，放置方法與Table5相同

Literal Table(literTab)為放置literal及其對應位址的array，放置方法與Table5-7相同

Literal Buffer(literBuf)為放置尚未定義之literal的vector

line為存放每一行指令資訊(line, location, object code, 是否錯誤, SICXE指令的format)的vector

//本次作業指定功能皆已完成