

SIC/SICXE 侯昱宏

設計辦法:建一個 struct 陣列，每一行都是一個 Struct，在 Pass1 時判斷這行是屬於什麼類型，有無定義，並且計算記憶體位址，且建立一個 addresstable(map 型別)，來當 cross reference，另外每個指令的 op code 用 map 來存放。Pass2 時根據 pass1 解析的東西來翻機器碼。SICXE 則是要再做判斷是哪個 type，PC 算距離是否能用 12 個 bit 表達，如果不能要轉 BASE reg，再不能跳 error。

struct alldata

```
{
    int line ;
    string Loc = "";
    string Sourcstatement = "";
    string def = "";
    string table1 = "";
    string table2 = "";
    string table3 = "";

    string table3_2 = "";    //type2 會有兩個暫存器

    string table4 = "";
    string table5 = "";    // table5
    string table6 = "" ;    // table6
    string table7 = "" ;    // table7
    string Objectcode = "" ;

    string annotation = "" ;//註解

    string type = "" ;
};
```

資料結構如上

```

char DecToHex (unsigned int n)
{
    char hex[] = {'0', '1', '2', '3', '4', '5',
                  '6', '7', '8', '9', 'A', 'B', 'C',
                  'D', 'E', 'F'};
    return hex[n] ;
}

string toHex (int num ) // 10轉16
{
    unsigned int unNum = (int) num ;
    string result ;

    do
    {
        result = DecToHex(unNum % 16) + result ;
        unNum /= 16 ;
    } while (unNum>0);
    return result ;
}

int toDec( string str ) //16轉10
{
    return strtol( str.c_str(), NULL,16 ) ;
}

```

進位轉換

op code

```

char op_TAB[59][7] = { "ADD", "ADDF", "ADDR", "AND", "CLEAR", "COMP", "COMPF", "COMPR", "DIV", "DIVF", "DIVR",
    "FIX", "FLOAT", "HIO", "J", "JEQ", "JGT", "JLT", "JSUB", "LDA", "LDB", "LDCH", "LDF", "LDL", "LDS",
    "LDT", "LDX", "LPS", "UML", "MULF", "MULR", "NORM", "OR", "RD", "RMO", "RSUB", "SHIFTL", "SHIFTR",
    "SIO", "SSK", "STA", "STB", "STCH", "STF", "STI", "STL", "STS", "STSW", "STT", "STX", "SUB", "SUBF",
    "SUBR", "SVC", "TD", "TIO", "TIX", "TIXR", "WD" };

char opCode[59][4] = { "18", "58", "90", "40", "B4", "28", "88", "A0", "24", "64", "9C", "C4", "C0", "F4", "3C",
    "30", "34", "38", "48", "00", "68", "50", "70", "08", "6C", "74", "04", "E0", "20", "60", "98", "C8",
    "44", "D8", "AC", "4C", "A4", "A8", "F0", "EC", "0C", "78", "54", "80", "D4", "14", "7C", "E8", "84",
    "10", "1C", "5C", "94", "B0", "E0", "F8", "2C", "B8", "DC" };

map< string, string > opcode ;
for( int i =0 ; i <= 58 ; i++ )
    opcode.insert( pair<string,string>( op_TAB[i], opCode[i] ) ) ;

```