PA1_report 110521167 曹寓恆

1. How to compile and execute?

一、正常模式

g++ -std=c++11 110521167_PA1.cpp -o PA1

./PA1 [input name].isc [output name].v

```
[110521167@eda359_forclass ~/PA1]$ g++ -std=c++11 110521167_PA1.cpp -o PA1
[110521167@eda359_forclass ~/PA1]$ ./PA1 c17.isc c17.v
```

二、偵錯模式(print execution time)

g++ -std=c++11 -DDEBUG_MODE 110521167_PA1.cpp -o PA1

./PA1 [input name].isc [output name].v

```
[110521167@eda359_forclass ~/PA1]$ g++ -std=c++11 -DDEBUG_MODE 110521167
_PA1.cpp -o PA1
[110521167@eda359_forclass ~/PA1]$ ./PA1 c6288.isc c6288.v
Excution time = 0.04sec
```

2. The completion of the assignment.

一、 主架構

```
int main(int argc, char **argv)
{
    const string caseName = "test";
    string inFile = argc == 1 ? caseName + ".isc" : argv[1];
    string outFile = argc == 1 ? caseName + ".v" : argv[2];

    Parser ps(inFile, outFile);
    vector<gate *> *gateList = ps.inputISC();
    ps.outputVerilog(gateList);
    delete gateList;

    return 0;
}
```

二、 Gate-structure

```
struct gate;
typedef vector<gate *> gatePtrVec;
struct gate
{
    string name, type, from;
    gatePtrVec input, output;

    gate(string _name, string _type) : name(_name), type(_type) {}
};
```

三、 Parser

inputISC():

呼叫 gotoFirstInfo 跳至第一行 gate 敘述句，接著重複呼叫 loadGateInfo()讀入電路直到讀完檔案，再呼叫 linker 將各個邏輯閘接線成方便 verilog 輸出的資料結構，最後回傳 gate list。

outputVerilog():

先呼叫 classify()將 inputISC()回傳的 gate list 分類成 input, interior, output 三種類型，接著呼叫 rename 處理 verilog 的命名規則，最後完成輸出。

```
class Parser
{
private:
    string inFile, outFile, caseName;
    Linker lk;
    ifstream ifs;
    ofstream ofs;

    // sub functions of inputISC()
    string gotoFirstInfo();
    void loadGateInfo(string &);

    // sub functions of outputVerilog()
    tuple<gatePtrVec, gatePtrVec, gatePtrVec> classify(gatePtrVec &gateList);
    void rename(gatePtrVec &, string);
    void outputDeclare(gatePtrVec &input, gatePtrVec &interior, gatePtrVec &output);
    void outputGate(gatePtrVec &interior, gatePtrVec &output);

public:
    Parser(string &, string &);
    ~Parser();
    bool openFile();
    gatePtrVec inputISC();
    void outputVerilog(gatePtrVec &gateList);
};
```

四、 Linker

connectGate():

根據 private 儲存的資訊，若 gate’ s input 連接到類別為”from”的 gate structure 則再往前追朔訊號來源，否則不需要更動。

getSortedGateList():

map index2gate 本身已經依照 index 進行排序，所以只要遍歷 index2gate 並回傳就能得到所求。

```
class Linker
{
private:
    vector<pair<gate *, vector<int>>> gate_index; // store input gates' index
    unordered_map<int, gate *> index2gate;        // index to gate
    unordered_map<string, gate *> name2gate;      // gateName to gate

public:
    ~Linker();

    // operations of private variable
    void pushVec_gate_index(gate *, vector<int> &);
    void insertMap_index2gate(int, gate *);
    void insertMap_name2gate(string &, gate *);

    // gate linker
    void connectGate();
    gatePtrVec getSortedGateList();
};
```

3. The hardness of this assignment and how you overcome it.

我覺得這次作業蠻簡單的沒有遇到什麼問題。

4. Any suggestions about this programming assignment.

我覺得或許可以增加 bonus，例如給 delay 參數找 critical path 之類的。