//add/sub->2 cycle

//mul->10 cycle

//div->40 cycle

#include<iostream>

#include<string>

#include<vector>

#include<sstream>

#include<fstream>

#include <iomanip>

using namespace std;

struct inst\_reg\_struct//建立儲存inst的struct

{

string d = " ";//目的

string r1 = " ";//來源1

string r2 = " ";//來源2

};

struct regnameval //建立儲存reg的名稱、值

{

string name = "";

int value;

};

vector<string> inst\_type;//讀inst的type

vector<inst\_reg\_struct> inst\_reg;//讀inst的reg

vector<regnameval>newreg;

int main() {

vector<string> input\_inst;

ifstream ifs("input.txt");

string in;

if (!ifs.is\_open()) {

cout << "Failed to open file.\n";

}

else {

while (getline(ifs, in)) {

// cout << in << "\n";

input\_inst.push\_back(in);

}

ifs.close();

}

for (int i = 0; i < input\_inst.size(); i++)

{

for (int j = 0; j < input\_inst[i].size(); j++)

{

if (input\_inst[i][j] == ' ')

{

string buffer;

buffer.assign(input\_inst[i].substr(0, j));

inst\_type.push\_back(buffer);

for (int k = j; k < input\_inst[i].size(); k++)

{

if (input\_inst[i][k] == ',')

{

inst\_reg\_struct buffer2;

buffer2.d = (input\_inst[i].substr(k - 2, 2));

buffer2.r1 = (input\_inst[i].substr(k + 2, 2));

buffer2.r2 = (input\_inst[i].substr(k + 6, 3));

inst\_reg.push\_back(buffer2);

break;

}

}

break;

}

}

}

int\*\* time\_table;//先算何時issue,dispatch,write\_back

time\_table = new int\* [8];

for (int i = 0; i < 8; i++) {

time\_table[i] = new int[3];

}

//issue

for (int i = 0; i < input\_inst.size(); i++) {

time\_table[i][0] = i + 1;

}

//dispatch

for (int i = 0; i < input\_inst.size(); i++)

{

int statue = -1;

for (int k = 0; k < i; k++) {

if (inst\_reg[i].r1 == inst\_reg[k].d || inst\_reg[i].r2 == inst\_reg[k].d)//後面inst來源等於前面inst的結果(RAW關係)

statue = k;//記錄其位置

}

int buffer;

if (statue != -1) {//有RAW

buffer = time\_table[statue][2] + 1;//有RAW->等Write Back

}

else {//沒有RAW

buffer = time\_table[i][0] + 1;//沒有RAW->dispatch+1

}

time\_table[6][0] = 55;

string tempinsttype = inst\_type[i];//目前inst的type

int s2 = -1;

for (int w = 0; w < i; w++) { //如果一樣類型要dispatch

if (tempinsttype == "ADD" || tempinsttype == "SUB" || tempinsttype == "ADDI")

{

if (time\_table[w][2] > buffer)

if (inst\_type[w] == "ADD" || inst\_type[w] == "ADDI" || inst\_type[w] == "SUB")

s2 = w;

}

else if (tempinsttype == "MUL" || tempinsttype == "DIV")

{

if (time\_table[w][2] > buffer)

if ((inst\_type[w] == "MUL" || inst\_type[w] == "DIV"))

s2 = w;

}

}

if (s2 != -1)

buffer = time\_table[s2][2];

//dispatch

time\_table[i][1] = buffer;

//write\_back

if (inst\_type[i] == "ADD" || inst\_type[i] == "ADDI" || inst\_type[i] == "SUB") {

time\_table[i][2] = time\_table[i][1] + 2;//add/sub->2

}

else if (inst\_type[i] == "MUL") {

time\_table[i][2] = time\_table[i][1] + 10;//mul->10

}

else if (inst\_type[i] == "DIV") {

time\_table[i][2] = time\_table[i][1] + 40;//div->40

}

}

int mc = 0;//最大的cycle

for (int i = 0; i < 8; i++)

{

if (mc < time\_table[i][2])

mc = time\_table[i][2];

}

//將會執行cycle的設成true

bool\* cyclestatue = new bool[mc];

for (int i = 0; i < mc; i++)//先全部給false

cyclestatue[i] = false;

for (int i = 0; i < input\_inst.size(); i++)

{

for (int j = 0; j < 3; j++)

{

cyclestatue[time\_table[i][j]] = true;

}

}

struct RS

{

string sign = "";

string R1;

string R2;

};

vector<bool>RS\_status = { false,false, false, false, false };

vector<RS>RS(5);//紀錄RS

vector<int>RS\_pos = { -1,-1,-1,-1,-1 };//紀錄RS中放inst的位置

string RS\_add\_buffer;//紀錄add\_buffer

string RS\_mul\_buffer;//紀錄mul\_buffer

string t[5] = { "F1","F2","F3","F4","F5" };

for (int i = 0, k = 0; i < 5; i++, k += 2)

{

regnameval temp;

temp.name = t[i];

temp.value = k;

newreg.push\_back(temp);//初始化F1-F5

}

vector<bool>RAT\_status = { false,false, false, false, false };

vector<string>RAT(5);//紀錄RAT

for (int i = 1; i <= mc; i++)

{

if (cyclestatue[i])

{

int issue = -1;//記錄誰要issue

int dispatch[2] = { -1,-1 };//記錄ADD,MUL的dispatch

int write\_back = -1;//記錄誰要write\_back

for (int j = 0; j < input\_inst.size(); j++)

{

if (time\_table[j][0] == i)//找要issue

issue = j;

if (time\_table[j][1] == i) {//找要dispatch

for (int k = 0; k < 2; k++) {

if (dispatch[k] == -1) {

dispatch[k] = j;

break;

}

}

}

if (time\_table[j][2] == i)//找要write\_backh

write\_back = j;

}

if (issue != -1)//代表有要issue的inst

{

if (inst\_type[issue] == "ADD")

{

for (int k = 0; k < 3; k++)//有三個位置可暫放

{

if (!RS\_status[k])//看是否還有位置 (true代表有人了/false代表有空位)

{

RS\_pos[k] = issue;//把要issue的位置儲存下來

RS[k].sign = '+';

for (int j = 0; j < 5; j++)

{

if (inst\_reg[issue].r1 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R1 = to\_string(newreg[j].value);

else

RS[k].R1 = RAT[j];

}

if (inst\_reg[issue].r2 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R2 = to\_string(newreg[j].value);

else if (RAT\_status[j])

RS[k].R2 = RAT[j];

else

{

RS[k].R2 = inst\_reg[issue].r2;

}

}

}

RS\_status[k] = true;//代表RS有人了

for (int j = 0; j < 5; j++) {

if (newreg[j].name == inst\_reg[issue].d && inst\_reg[issue].r1 != newreg[j].name && inst\_reg[issue].r2 != newreg[j].name)

{

{

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

}

}

break;

}

}

}

else if (inst\_type[issue] == "ADDI")

{

for (int k = 0; k < 3; k++)//有三個位置可暫放

{

if (!RS\_status[k])//看是否還有位置 (true代表有人了/false代表有空位)

{

RS\_pos[k] = issue;//把要issue的位置儲存下來

RS[k].sign = '+';

if (issue == 0)

RS[k].R2 = "1";

else if (issue == 5)

RS[k].R2 = "2";

for (int j = 0; j < 5; j++)

{

if (inst\_reg[issue].r1 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R1 = to\_string(newreg[j].value);

else

RS[k].R1 = RAT[j];

}

}

RS\_status[k] = true;//代表RS有人了

for (int j = 0; j < 5; j++) {

if (newreg[j].name == inst\_reg[issue].d && inst\_reg[issue].r1 != newreg[j].name && inst\_reg[issue].r2 != newreg[j].name)

{

{

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

}

}

break;

}

}

}

else if (inst\_type[issue] == "SUB")

{

for (int k = 0; k < 3; k++)//有三個位置可暫放

{

if (!RS\_status[k])//看是否還有位置 (true代表有人了/false代表有空位)

{

RS\_pos[k] = issue;//把要issue的位置儲存下來

RS[k].sign = '+';

for (int j = 0; j < 5; j++)

{

if (inst\_reg[issue].r1 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R1 = to\_string(newreg[j].value);

else

RS[k].R1 = RAT[j];

}

if (inst\_reg[issue].r2 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R2 = to\_string(newreg[j].value);

else if (RAT\_status[j])

RS[k].R2 = RAT[j];

else

{

RS[k].R2 = inst\_reg[issue].r2;

}

}

}

RS\_status[k] = true;//代表RS有人了

for (int j = 0; j < 5; j++)

{

if (newreg[j].name == inst\_reg[issue].d && inst\_reg[issue].r1 != newreg[j].name && inst\_reg[issue].r2 != newreg[j].name)

{

{

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

}

}

break;

}

}

}

else if (inst\_type[issue] == "MUL")

{

for (int k = 3; k < 5; k++)

{

if (!RS\_status[k])

{

RS\_pos[k] = issue;

RS[k].sign = '\*';

for (int j = 0; j < 5; j++)

{

if (inst\_reg[issue].r1 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R1 = to\_string(newreg[j].value);

else

RS[k].R1 = RAT[j];

}

if (inst\_reg[issue].r2 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R2 = to\_string(newreg[j].value);

else

RS[k].R2 = RAT[j];

}

}

RS\_status[k] = true;//代表RS有人了

for (int j = 0; j < 5; j++) {

if (newreg[j].name == inst\_reg[issue].d && inst\_reg[issue].r1 != newreg[j].name && inst\_reg[issue].r2 != newreg[j].name) {

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

else if (newreg[j].name == inst\_reg[issue].d)

{

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

}

break;

}

}

}

else if (inst\_type[issue] == "DIV")

{

for (int k = 3; k < 5; k++)

{

if (!RS\_status[k])

{

RS\_pos[k] = issue;

RS[k].sign = '/';

for (int j = 0; j < 5; j++)

{

if (inst\_reg[issue].r1 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R1 = to\_string(newreg[j].value);

else

RS[k].R1 = RAT[j];

}

if (inst\_reg[issue].r2 == newreg[j].name)

{

if (!RAT\_status[j])

RS[k].R2 = to\_string(newreg[j].value);

else

RS[k].R2 = RAT[j];

}

}

RS\_status[k] = true;//代表RS有人了

for (int j = 0; j < 5; j++) {

if (newreg[j].name == inst\_reg[issue].d && inst\_reg[issue].r1 != newreg[j].name && inst\_reg[issue].r2 != newreg[j].name) {

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

else if (newreg[j].name == inst\_reg[issue].d)

{

RAT\_status[j] = true;//代表RAT有人了

string t = to\_string(k + 1);

RAT[j] = "RS" + t;

}

}

break;

}

}

}

}

for (int j = 0; j < 2; j++)

{

if (dispatch[j] != -1)//要dispatch

{

for (int k = 0; k < 5; k++) {//設定dispatch的buffer

if (RS\_pos[k] == dispatch[j])

{

if (RS\_status[k])

{

if (k < 3)

{

string num = to\_string(k + 1);

RS\_add\_buffer = "(RS" + num + ") " + RS[k].R1 + " " + RS[k].sign + " " + RS[k].R2;

}

else

{

string num = to\_string(k + 1);

RS\_mul\_buffer = "(RS" + num + ") " + RS[k].R1 + " " + RS[k].sign + " " + RS[k].R2;

}

}

}

}

}

}

int pos = 0;

if (write\_back != -1) //要write\_back

{

//RS

for (int k = 0; k < 5; k++)

{

if (RS\_pos[k] == write\_back)

{

if (RS\_status[k])

{

for (int j = 0; j < 5; j++) {

if (newreg[j].name == inst\_reg[RS\_pos[k]].d)

pos = j;

}

int count;//計算

if (k < 3)//add,sub

{

if (RS[k].sign == "+")

{

count = stoi(RS[k].R1) + stoi(RS[k].R2);

}

else

{

count = stoi(RS[k].R1) - stoi(RS[k].R2);

}

RS\_add\_buffer = "";

}

else //mul,div

{

if (RS[k].sign == "\*")

{

count = stoi(RS[k].R1) \* stoi(RS[k].R2);

}

else

{

count = stoi(RS[k].R1) / stoi(RS[k].R2);

}

RS\_mul\_buffer = "";

}

string temp = to\_string(k + 1);

string str = "RS" + temp;

for (int i = 0; i < 5; i++)

{

if (RAT\_status[i] && RAT[i] == str) //把RAT中要wb的消除

{

RAT\_status[i] = false;

RAT[i] = "";

newreg[pos].value = count;//給值 F2,F4,F2

}

}

for (int i = 0; i < 5; i++)//RS中有沒有要值

{

if (RS[i].R1 == str) {

RS[i].R1 = to\_string(count);

}

if (RS[i].R2 == str) {

RS[i].R2 = to\_string(count);

}

}

newreg[pos].value = count;//給值

RS\_status[k] = false;//清RS

RS\_pos[k] = -1;

RS[k].sign = "";

RS[k].R1 = "";

RS[k].R2 = "";

}

}

}

}

cout << endl;

cout << "----------------------------------------------------------------------------" << endl;

cout << endl;

//輸出第幾個Cycle

cout << "Cycle : " << i << endl;

cout << endl;

//輸出RF表格

cout << setw(4) << "" << left << setw(5) << "----RF----" << endl;

for (int i = 0; i < 5; i++) {

cout << left << setw(4) << newreg[i].name <<

"|" << right << setw(8) << newreg[i].value

<< "|" << endl;

}

cout << setw(4) << "" << "----------" << endl;

cout << endl;

//輸出RAT表格

cout << setw(4) << "" << left << setw(5) << "----RAT----" << endl;

for (int i = 0; i < 5; i++) {

cout << left << setw(4) << newreg[i].name <<

"|" << right << setw(8) << RAT[i]

<< "|" << endl;

}

cout << setw(4) << "" << "----------" << endl;

cout << endl;

//輸出RS(ADD)表格

cout << setw(4) << "" << left << setw(8) << "---------RS-------------" << endl;

for (int i = 0; i < 3; i++) {

cout << "RS" << left << setw(2)

<< i + 1 << "|" << right << setw(7) << RS[i].sign <<

"|" << right << setw(7) << RS[i].R1 <<

"|" << right << setw(7) << RS[i].R2 << "|" << endl;

}

cout << setw(4) << "" << "-------------------------" << endl;

if (RS\_add\_buffer == "")//如果buffer為空的->empty

RS\_add\_buffer = "empty";

if (RS\_mul\_buffer == "")

RS\_mul\_buffer = "empty";

cout << "BUFFER : " << RS\_add\_buffer << endl;//輸出RS\_add\_buffer

cout << endl;

//輸出RS(MUL)表格

cout << setw(4) << "" << left << setw(8) << "------------------------" << endl;

for (int i = 3; i < 5; i++) {

cout << "RS" << left << setw(2)

<< i + 1 << "|" << right << setw(7) << RS[i].sign <<

"|" << right << setw(7) << RS[i].R1

<< "|" << right << setw(7) << RS[i].R2 << "|" << endl;

}

cout << setw(4) << "" << "-------------------------" << endl;

cout << "BUFFER : " << RS\_mul\_buffer << endl;//輸出RS\_mul\_buffer

}

}

}