#include<iostream>

#include<fstream>

#include<string>

#include<vector>

#include<sstream>

#include <cmath>

#include<algorithm>

#include <cstdlib>

#include <iomanip>

using namespace std;

struct contentofRS//每個RS中的內容

{

bool used;

string operand;

string rs1;

string rs2;

};

struct inst//每個instuction的內容

{

string type;

string rd;

string rs1;

string rs2;

int imm;

};

struct buffer//buffer內容

{

int cycle;

int RS;

bool empty; // 0 代表空的，1 代表有東西

};

vector<inst> instruction; //大小為input.txt總行數

vector<int> RF = { 0,0,2,4,6,8 }; //初始化RF，RF[0]不會用到

string RAT[6];//RAT[0]不會用到

contentofRS RS[6];//RS[0]不會用到

int cycle = 1; //目前進行到的Cycle編號

bool changedCycle; //是否有變化

int C\_ADDandSUB, C\_MUL, C\_DIV; //四種type的ALU Cycle time

buffer bufferADD;

buffer bufferMUL;

void input()

{

ifstream inFile("input.txt", ios::in);

if (!inFile) {

cout << "File could not be opened!\n";

system("pause");

exit(1);

}

stringstream ss;

string buffer;

while (getline(inFile, buffer)) {

ss.str("");

ss.clear();

ss << buffer;

inst insttemp;

string temp;

//將type寫入

ss >> temp;

insttemp.type = temp;

//將rd寫入

ss >> temp;

temp.pop\_back();

insttemp.rd = temp;

//將rs1寫入

ss >> temp;

temp.pop\_back();

insttemp.rs1 = temp;

//判斷type為何種，若為ADDI就將temp轉乘int型態，寫入imm，其他則是寫入rs2

ss >> temp;

if (insttemp.type == "ADDI")

insttemp.imm = stoi(temp);

else

insttemp.rs2 = temp;

instruction.push\_back(insttemp);//將本次做好的instuction push進去

}

}

void Issue()

{

if (!instruction.empty())//判斷還有沒有instruction等待issue

{

if (instruction[0].type == "ADD" || instruction[0].type == "SUB" || instruction[0].type == "ADDI")

{

for (int i = 1; i <= 3; i++)//ADD的ALU

{

if (!RS[i].used)//判斷RS[i]有沒有空間

{

if (instruction[0].type == "SUB")

RS[i].operand = "-";

else

RS[i].operand = "+";

int rs1 = instruction[0].rs1[1] - '0'; //找出本instruction的rs1 register編號

if (RAT[rs1] == "")//RAT[rs1]為空

RS[i].rs1 = to\_string(RF[rs1]);

else//RAT[rs1]有東西

RS[i].rs1 = RAT[rs1];

if (instruction[0].type == "ADDI")//若是"ADDI"，imm直接放到rs2

RS[i].rs2 = to\_string(instruction[0].imm);

else

{

int rs2 = instruction[0].rs2[1] - '0';//找出本instruction的rs2 register編號

if (RAT[rs2] == "")//RAT[rs2]為空

RS[i].rs2 = to\_string(RF[rs2]);

else//RAT[rs2]有東西

RS[i].rs2 = RAT[rs2];

}

int rd = instruction[0].rd[1] - '0';//找出本instruction的rd register編號

RAT[rd] = "RS" + to\_string(i);// 更新 RAT 的值

RS[i].used = 1;//將RS[i]設為"正在使用中"

changedCycle = 1;//本Cycle有變化

instruction.erase(instruction.begin()); //將issue進來的instruction刪除

break;

}

}

}

else if (instruction[0].type == "DIV" || instruction[0].type == "MUL")

{

for (int i = 4; i <= 5; i++)//MUL的ALU

{

if (!RS[i].used)//判斷RS[i]有沒有空間

{

RS[i].used = 1;//將RS[i]設為"正在使用中"

if (instruction[0].type == "MUL")

RS[i].operand = "\*";

else

RS[i].operand = "/";

int rs1 = instruction[0].rs1[1] - '0'; //找出本instruction的rs1 register編號

if (RAT[rs1] == "")//RAT[i]為空

RS[i].rs1 = to\_string(RF[rs1]);

else//RAT有東西

RS[i].rs1 = RAT[rs1];

int rs2 = instruction[0].rs2[1] - '0';//找出本instruction的rs2 register編號

if (RAT[rs2] == "")//RAT[i]為空

RS[i].rs2 = to\_string(RF[rs2]);

else//RAT有東西

RS[i].rs2 = RAT[rs2];

int rd = instruction[0].rd[1] - '0';//找出本instruction的rd register編號

RAT[rd] = "RS" + to\_string(i);//更新RAT的值

changedCycle = 1;//本Cycle有變化

instruction.erase(instruction.begin()); //將issue進來的instruction刪除

break;

}

}

}

}

}

void Dispatch()

{

if (!bufferADD.empty)//判斷bufferADD是否有東西

{

for (int i = 1; i <= 3; i++)//從RS找出數值都準備好的

{

if (RS[i].used)

{

if (RS[i].rs1[0] != 'R' && RS[i].rs2[0] != 'R')

{

bufferADD.RS = i;

bufferADD.empty = 1;

bufferADD.cycle = cycle + C\_ADDandSUB;

changedCycle = 1;

}

}

}

}

if (!bufferMUL.empty)//判斷bufferMUL是否有東西

{

for (int i = 4; i <= 5; i++)//從RS找出數值都準備好的

{

if (RS[i].used)//要分成乘法和除法兩項，因為ALUcycle數不同

{

if (RS[i].rs1[0] != 'R' && RS[i].rs2[0] != 'R')

{

bufferMUL.RS = i;

bufferMUL.empty = 1;

if (RS[i].operand == "\*")

bufferMUL.cycle = cycle + C\_MUL;

else

bufferMUL.cycle = cycle + C\_DIV;

changedCycle = 1;

}

}

}

}

}

void WriteResult(buffer& thisbuffer)

{

int result;

if (cycle == thisbuffer.cycle)

{

//計算buffer的值

string operand = RS[thisbuffer.RS].operand;

int rs1 = stoi(RS[thisbuffer.RS].rs1);

int rs2 = stoi(RS[thisbuffer.RS].rs2);

if (operand == "+")

result = rs1 + rs2;

else if (operand == "-")

result = rs1 - rs2;

else if (operand == "\*")

result = rs1 \* rs2;

else

result = rs1 / rs2;

//更新RAT和RF的值

for (int i = 1; i <= 5; i++)

if (RAT[i] == "RS" + to\_string(thisbuffer.RS))

{

RAT[i] = "";

RF[i] = result;

}

//將result寫進RS

for (int i = 1; i <= 5; i++)

{

if (RS[i].rs1 == "RS" + to\_string(thisbuffer.RS))

RS[i].rs1 = to\_string(result);

if (RS[i].rs2 == "RS" + to\_string(thisbuffer.RS))

RS[i].rs2 = to\_string(result);

}

//freeRS

RS[thisbuffer.RS].operand = "";

RS[thisbuffer.RS].rs1 = "";

RS[thisbuffer.RS].rs2 = "";

RS[thisbuffer.RS].used = 0;

thisbuffer.empty = 0;

changedCycle = 1;

}

}

void printCycleStatus()

{

cout << "Cycle : " << cycle << endl << endl;

cout << " \_\_RF\_\_\_" << endl;

for (int i = 1; i <= 5; ++i)

cout << " F" << i << " | " << setw(3) << RF[i] << " |" << endl;

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

cout << " \_\_RAT\_\_" << endl;

for (int i = 1; i <= 5; ++i)

cout << " F" << i << " | " << setw(3) << RAT[i] << " |" << endl;

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

cout << " \_RS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

for (int i = 1; i <= 3; i++)

{

cout << " RS" << i << " | " << setw(3) << RS[i].operand << " | " << setw(3) << RS[i].rs1 << " | " << setw(3) << RS[i].rs2 << " | " << endl;

if (i == 3)

{

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

cout << " BUFFER: ";

if (!bufferADD.empty)

cout << "empty" << endl << endl;

else

cout << "(RS" << bufferADD.RS << ") " << RS[bufferADD.RS].rs1 << " " << RS[bufferADD.RS].operand << " " << RS[bufferADD.RS].rs2 << endl << endl;

}

}

cout << " \_RS\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

for (int i = 4; i <= 5; i++)

{

cout << " RS" << i << " | " << setw(3) << RS[i].operand << " | " << setw(3) << RS[i].rs1 << " | " << setw(3) << RS[i].rs2 << " | " << endl;

if (i == 5)

{

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

cout << " BUFFER: ";

if (!bufferMUL.empty)

cout << "empty" << endl << endl;

else

cout << "(RS" << bufferMUL.RS << ") " << RS[bufferMUL.RS].rs1 << " " << RS[bufferMUL.RS].operand << " " << RS[bufferMUL.RS].rs2 << endl << endl;

}

}

}

int main()

{

input();//讀檔

//輸入ALU的Cycle Time

cout << "Please input cycle time of ADD/SUB, MUL, DIV." << endl;

cin >> C\_ADDandSUB >> C\_MUL >> C\_DIV;

while (true)

{

if (instruction.empty()) //判斷instruction及RS還有沒有東西

{

int i;

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

if (RS[i].used != 0)

break;

if (i == 5)

break;

}

changedCycle = 0;

WriteResult(bufferADD);//看有沒有ADD ALU做完

WriteResult(bufferMUL);//看有沒有MUL ALU做完

Dispatch();

Issue();

if (changedCycle)

printCycleStatus();

cycle++;

}

}