VLSI Testing and Design for Testability

Assignment2

B103526 中央電機黃聖倫

■ Logic simulator on circuits c17.bench and c7552.bench

指令:./atpg -logicsim -input <input_pattern_filename> <絕對路徑電路位置>

Ex: ./atpg -logicsim -input c17.input
"/home/Student113/s110305504/VLSI_Testing/Assignment2/circuits/iscas85/c17.bench"

修改"sim.cc"中的「PrintIO()」使輸出格式與助教給的格式一樣。

再使用 diffchecker 確認了兩者輸出一樣

```
[s110305504@cad podem]$ ./atpg -logicsim -input c17.input "/home/Student113/s110305504/VLSI_Testing/Assignment2/circuits/iscas85/c17.bench
"
Start parsing input file
Finish reading circuit file
Run logic simulation
PI: 01110     PO: 00
PI: 10101     PO: 11
PI: 00101     PO: 01
PI: 01000     PO: 11
PI: 10001     PO: 01
PI: 00011     PO: 01
PI: 00111     PO: 00
PI: 00000     PO: 00
total CPU time = 0
```

PI: 01110 PO: 00
PI: 10101 PO: 11
PI: 00101 PO: 01
PI: 01000 PO: 11
PI: 10001 PO: 01
PI: 00011 PO: 01
PI: 00111 PO: 00
PI: 00000 PO: 00

■ Single Pattern Logic Simulator

(a) Generate pattern

```
else if (option.retrieve("pattern")) {
    string pattern_num=option.retrieve("num");
    int No_pattern;
    stringstream ss(pattern_num);
    ss >> No_pattern;
    string file_name=option.retrieve("output");
    //cout<<file_name<<" "<<pattern_num<<endl;
    if(option.retrieve("unknown")){
        pattern_generate(file_name,No_pattern,1);
    }
    else{
        pattern_generate(file_name,No_pattern,0);
    }
}
```

沒有 X 的輸入: ./atpg -pattern -num {integer} -output [output_name] [絕對路徑 circuit_name]

有 X 的輸入:./atpg -pattern -num {integer} -unknown -output [output_name] [絕對路徑 circuit_name]

根據 pattern 產生需求使用不同指令

使用 srand(time(0))隨機生成亂數，並根據是否有 unkown 決定最後結果要取餘數 3 還是取餘數 2。再使用兩個 for loop 生成所需 pattern。

```
void pattern_generate(string file_name,int number,bool check)
{
    int signal=0;
    unsigned i = 0;
    GATE* gptr;
    ofstream outfile;
    outfile.open(file_name.c_str());
    if (!outfile.is_open()) {
        cerr << "Can't open file!" << endl;
    }
    for (;i < Circuit.No_Gate();i++) {
        gptr = Circuit.Gate(i);
        if(gptr->GetFunction()==G_PI)
        {
            outfile<<"PI "<<gptr->GetName()<<" ";
            signal++;
        }
    }
    outfile<<endl;
    srand(static_cast<unsigned>(time(0)));
    for (int i=0;i<number;i++)
    {
        for (int j=0;j<signal;j++)
        {
            outfile<< random_value(check);
        }
        outfile<<endl;
    }
}
```

```
char random_value(bool check) {
    int random_num = 0;
    if(check) random_num = rand() % 3;
    else random_num = rand() % 2;
    switch (random_num) {
        case 0:
            return '0';
        case 1:
            return '1';
        case 2:
            return 'X';
        default:
            return '0';
    }
}
```

c17.bench

number	CPUtime	maximum memory usage(kb)	Average memory usuage
100	0	2108	0
1000	0	2112	0
10000	0.06	2112	0

c7552.bench

number	CPUtime	maximum memory usage(kb)	Average memory usuage
100	0.06	4328	0
1000	0.29	4328	0
10000	2.09	4324	0

包含測量 MEMORY 指令:

/usr/bin/time -f "Average memory usage: %K \nMaximum memory usage: %M\n" ./atpg -logicsim -input <pattern_name> <絕對路徑電路檔案位置>

Average memory usage 皆為 0 推測為程式運行時間太少，沒有足夠的樣本來統計平均記憶體使用。

(b) Modified logic simulator

透過修改 sim.cc 中的 LogicSimVectors()、LogicSim()、Evaluate(GATEPTR gptr)來完成這個部分的程式。

```
void CIRCUIT::LogicSimVectorsmod()
{
    cout << "Run logic simulation" << endl;
    //read test patterns
    while (!Pattern.eof()) {
        Pattern.ReadNextPattern();
        SchedulePI();
        LogicSim_MOD();
        PrintIO();
    }
    return;
}
```

```
void CIRCUIT::LogicSim_MOD()
{
    GATE* gptr;
    VALUE new_value;
    for (unsigned i = 0;i <= MaxLevel;i++) {
        while (!Queue[i].empty()) {
            gptr = Queue[i].front();
            Queue[i].pop_front();
            gptr->ResetFlag(SCHEDULED);
            new_value = Evaluate_MOD(gptr);
            if (new_value != gptr->GetValue()) {
                gptr->SetValue(new_value);
                ScheduleFanout(gptr);
            }
        }
    }
    return;
}
```

```
VALUE CIRCUIT::Evaluate_MOD(GATEPTR gptr) {
    GATEFUNC fun(gptr->GetFunction());
    VALUE cv(CV[fun]);
    VALUE value(gptr->Fanin(0)->GetValue());
    VALUE temp_value;

    // 根據邏輯函數進行計算
    switch (fun) {
        case G_AND:
        case G_NAND:
            for (unsigned i = 1; i < gptr->No_Fanin() && value != cv; ++i) {
                logic_and(temp_value, value, gptr->Fanin(i)->GetValue());
                value = temp_value;
            }
            break;

        case G_OR:
        case G_NOR:
            for (unsigned i = 1; i < gptr->No_Fanin() && value != cv; ++i) {
                logic_or(temp_value, value, gptr->Fanin(i)->GetValue());
                value = temp_value;
            }
            break;

        default:
            break;
    }

    // NAND NOR NOT
    if (gptr->Is_Inversion()) {
        logic_not(value, value);
    }

    return value;
}
```

並新增如 setLogicValue、getLogicValue 等函式讓字體 switch 到數字，數字 switch 到文字。

並將 0 視為 00、1 視為 11、X 視為 01 或是 10 來做 BITWISE 的計算。

```
// AND
void logic_and(VALUE &result, VALUE a, VALUE b) {
    int a_val, b_val;
    setLogicValue(a_val, a);
    setLogicValue(b_val, b);
    int fin = (a_val & b_val);
    getLogicValue(fin, result);
}

// OR
void logic_or(VALUE &result, VALUE a, VALUE b) {
    int a_val, b_val;
    setLogicValue(a_val, a);
    setLogicValue(b_val, b);
    int fin = (a_val | b_val);
    getLogicValue(fin, result);
}

// NOT
void logic_not(VALUE &result, VALUE a) {
    int a_val;
    setLogicValue(a_val, a);
    int fin = ~a_val & 0b11;
    getLogicValue(fin, result);
}
```

指令為

EX:./atpg -mod_logicsim -input [input_name] [絕對路徑+circuit_name]

./atpg -mod_logicsim -input c7552.input
"/home/Student113/s110305504/VLSI_Testing/Assignment2/circuits/iscas85/c7552.bench"

通過與助教給的 c17.output、c7552.output 比較驗證了結果的正確性

```
[s110305504@cad podem]$ ./atpg -mod_logicsim -input c17.output "/home/Student113/s110305504/VLSI_Testing/Assignment2/circuits/iscas85/c17.bench"
Start parsing input file
Finish reading circuit file
Run logic simulation
PI: 11121 PO: 12
PI: 20202 PO: 22
PI: 01111 PO: 00
PI: 00012 PO: 02
PI: 11102 PO: 11
PI: 10221 PO: 22
PI: 01110 PO: 00
PI: 20121 PO: 22
PI: 12100 PO: 12
PI: 20121 PO: 22
total CPU time = 0

[s110305504@cad podem]$ ./atpg -logicsim -input c17.output "/home/Student113/s110305504/VLSI_Testing/Assignment2/circuits/iscas85/c17.bench"
Start parsing input file
Finish reading circuit file
Run logic simulation
PI: 11121 PO: 12
PI: 20202 PO: 22
PI: 01111 PO: 00
PI: 00012 PO: 02
PI: 11102 PO: 11
PI: 10221 PO: 22
PI: 01110 PO: 00
PI: 20121 PO: 22
PI: 12100 PO: 12
PI: 20121 PO: 22
total CPU time = 0
```

Build:

make