

VLSI Testing Assignment 0

311510173 魏子翔

1. First, I enroll the option ass0 in main.cc.

```
option.enroll("ass0",GetLongOpt::NoValue,  
             "do assiment 0", 0);
```

2. I create the function void GetCircuitInfo() under class Circuit in circuit.h and
descript it in detailed in circuit.cc.

Then I can implement this function to get the circuit information in main.cc.

```
else if(option.retrieve("ass0"))  
{  
    Circuit.GetCircuitInfo();  
}
```

3. The detail of function of GetCircuitInfo().

- A. I use the existing function No_PI(), No_PO(), and No_Gate() in circuit.h to
get the information of number of inputs and outputs, and total number of
gates respectively.

```
cout << "Number of inputs: " << No_PI() << endl;  
cout << "Number of outputs: " << No_PO() << endl;  
cout << "Total number of gates: " << No_Gate() << endl;
```

- B. To get the others of information, I need to code by myself. Using the loop to
count the number of gates of each type, and we can get the results by
applying the function called GATE* Gate(unsigned index) in circuit.h and
the structure of class GATE in gate.h. Simultaneously, I utilize No_Fanout()
of each gate to accumulate the branch, stem, and fanout.

```
cout << "Number of gates for each type*****" << endl;  
string gate_type[12]={ "PI", "PO", "PPI", "PPO", "NOT", "AND", "NAND", "OR", "NOR", "DFF", "BUF", "BAD"};  
unsigned EachGateNum[12] = {0};  
unsigned Total_Fanout = 0;  
unsigned Total_Stem = 0;  
unsigned Total_Branch = 0;  
for(unsigned i=0; i<No_Gate(); i++)  
{  
    EachGateNum[GATEFUNC(Gate(i)->GetFunction())]++;  
    if(Gate(i)->No_Fanout() > 1)  
    {  
        Total_Branch += Gate(i)->No_Fanout();  
        Total_Stem++;  
    }  
    Total_Fanout += Gate(i)->No_Fanout();  
}  
for(int i=0; i<12; i++)  
{  
    if(gate_type[i] == "DFF")  
        i++;  
    cout << gate_type[i] << ": " << EachGateNum[i] << endl;  
}  
cout << "*****" << endl;  
cout << "Number of flip-flops: " << EachGateNum[2] << endl;
```

- C. Finally, having the total number of branch, stem, and fanout from above, I can obtain the rest of answers by implementing the code like below.

```
cout << "Total number of signal nets: " << Total_Fanout + Total_Stem << endl;
cout << "Number of branch nets: " << Total_Branch << endl;
cout << "Number of stem nets: " << Total_Stem << endl;
cout << "Average number of fanouts of each gate: " << double(Total_Fanout)/No_Gate() << endl;
```

- D. Result of execution

```
[311510173@mseda03 podem]$ ./atpg -ass0 ../circuits/iscas89_seq/s35932_seq.bench
Start parsing input file
Finish reading circuit file
Number of inputs: 35
Number of outputs: 320
Total number of gates: 19876
Number of gates for each type*****
PI: 35
PO: 320
PPI: 1728
PP0: 1728
NOT: 3861
AND: 4032
NAND: 7020
OR: 1152
NOR: 0
BUF: 0
BAD: 0
*****
Number of flip-flops: 1728
Total number of signal nets: 39068
Number of branch nets: 21240
Number of stem nets: 7023
Average number of fanouts of each gate: 1.61225
total CPU time = 0.13
```