VLSI Testing and Design for Testability

Assignment1

B103526 中央電機黃聖倫

Method:

In order to find all paths that contain a certain input and output, I modified the function 'SetupOption' to add some new arguments.

```
int SetupOption(int argc, char ** argv)
{
    option.usage("[options] input_circuit_file");
    option.enroll("help", GetLongOpt::NoValue,
            "print this help summary", 0);
    option.enroll("logicsim", GetLongOpt::NoValue,
            "run logic simulation", 0);
    option.enroll("plogicsim", GetLongOpt::NoValue,
            "run parallel logic simulation", 0);
    option.enroll("fsim", GetLongOpt::NoValue,
            "run stuck-at fault simulation", 0);
    option.enroll("stfsim", GetLongOpt::NoValue,
            "run single pattern single transition-fault simulation", 0);
    option.enroll("transition", GetLongOpt::NoValue,
            "run transition-fault ATPG", 0);
    option.enroll("input", GetLongOpt::MandatoryValue,
            "set the input pattern file", 0);
    option.enroll("output", GetLongOpt::MandatoryValue,
            "set the output pattern file", 0);
    option.enroll("bt", GetLongOpt::OptionalValue,
            "set the backtrack limit", 0);
    /*************************************************************************/
    option.enroll("path", GetLongOpt::NoValue,
            "list and count all possible paths connecting the given PI and PO", 0);
    option.enroll("start", GetLongOpt::MandatoryValue,
            "set the input path", 0);
    option.enroll("end", GetLongOpt::MandatoryValue,
            "set the output path", 0);
    /*************************************************************************/
    int optind = option.parse(argc, argv);
    if ( optind < 1 ) { exit(0); }
    if ( option.retrieve("help") ) {
        option.usage();
        exit(0);
    }
    return optind;
}
```

Fig1. Modify Function ‘SetupOption’

To accelerate the code execution time, I used the function `findtrack` to get the addresses of the input and output signals, and called the function `backward` to propagate a flag from the output signal back to the input signal through certain gates.

By adding these flags, after finding the path from the input signal to the output signal, we can avoid gates that do not affect the result, thereby reducing unnecessary paths.

Additionally, I added a data member called `flag1` to the `Gate` class, which is of type `bool`. This flag indicates whether a gate can propagate the signal to the output.

```
158  void findtrack(string PI,string PO,GATE*& PI_PTR, GATE*& PO_PTR)
159  {
160      unsigned i = 0;
161      int signal=0;
162      GATE* gptr;
163      string out= PO;
164      for (;i < Circuit.No_Gate();i++) {
165          gptr = Circuit.Gate(i);
166          if(gptr->GetName()==PI)
167          {
168              signal++;
169              PI_PTR=gptr;
170          }
171          else if (gptr->GetName()==out)
172          {
173              signal++;
174              PO_PTR=gptr;
175          }
176          if (signal==2) break;
177      }
178      backward(PO_PTR);
179  }
```

Fig2. Function ‘findtrack’

```
void backward(GATE* PO_PTR)
{
    GATE* now=PO_PTR;
    now->Setflag1();
    if(now->GetFunction()==G_PI) return;
    else {
        for (unsigned j = 0;j < now->No_Fanin();j++) {
            if(now->Fanin(j)->Getflag1())continue;
            now->Fanin(j)->Setflag1();
            //cout<<now->Fanin(j)->GetName()<<endl;
            backward(now->Fanin(j));
        }
    }
}
```

Fig3. Function ‘backward’

I used recursion (by calling the recursive function) to find all the paths from the input signal to the output signal. By using the existing function `No_Fanout`, I retrieve the output of a specific gate and traverse the path until the output signal is found.

An important aspect is that I use `flag1` (as previously set by the `findtrack` and `backward` functions) to check whether a gate can propagate the signal to the output, which helps accelerate the process.

```
194  void recursive(GATE* PI_PTR, GATE* PO_PTR, vector<string>& namelist,unsigned &count)
195  {
196      if(PI_PTR==PO_PTR)
197      {
198          count++;
199          for (unsigned i=0;i<namelist.size();i++)
200          {
201              cout<<namelist[i]<<" ";
202          }
203          cout<<PO_PTR->GetName();
204          cout<<endl;
205          return;
206      }
207      else{
208          for (unsigned j = 0;j < PI_PTR->No_Fanout();j++) {
209              if(PI_PTR->Getflag1()){
210                  namelist.push_back(PI_PTR->GetName());
211                  recursive(PI_PTR->Fanout(j),PO_PTR,namelist,count);
212                  namelist.pop_back();
213              }
214          }
215      }
216  }
```

Fig4. Function ‘recursive’

Test Case Result:

CASE1:

./atpg -path -start G3 -end PO_G16 /path/circuits/iscas85/c17.bench

(path is depend on user workstation)

```
[s110305504@cad podem]$ ./atpg -path -start G3 -end PO_G16 /home/Student113/s110305504/VLSI_Testing/Assignment0/circuits/iscas85/c17.bench
Start parsing input file
Finish reading circuit file
G3 net14 net18 G16 PO_G16
G3 net17 G16 PO_G16
The paths from G3 to PO_G16: 2
total CPU time = 0
```

CASE2:

./atpg -path -start 126GAT_30 -end PO_863GAT_424 /path/circuits/iscas85/c880.bench

```
[s110305504@cad podem]$ ./atpg -path -start 126GAT_30 -end PO_863GAT_424 /home/Student113/s110305504/VLSI_Testing/Assignment0/circuits/iscas85/c880.bench
Start parsing input file
Finish reading circuit file
126GAT_30 517GAT_227 517GAT_227b 543GAT_236 581GAT_250 581GAT_250b 654GAT_270 734GAT_287 773GAT_351 773GAT_351b 788GAT_367 788GAT_367b 802GAT_372 808GAT_377 808GAT_377b 826GAT_391 837G
AT_396 846GAT_407 855GAT_418 863GAT_424 PO_863GAT_424
126GAT_30 517GAT_227 517GAT_227b 543GAT_236 581GAT_250 581GAT_250b 654GAT_270 734GAT_287 773GAT_351 789GAT_368 789GAT_368b 802GAT_372 808GAT_377 808GAT_377b 826GAT_391 837GAT_396 846GA
T_407 855GAT_418 863GAT_424 PO_863GAT_424
126GAT_30 517GAT_227 517GAT_227b 543GAT_236 581GAT_250 651GAT_271 722GAT_295 763GAT_337 773GAT_351 773GAT_351b 788GAT_367 788GAT_367b 802GAT_372 808GAT_377 808GAT_377b 826GAT_391 837GA
T_396 846GAT_407 855GAT_418 863GAT_424 PO_863GAT_424
126GAT_30 517GAT_227 517GAT_227b 543GAT_236 581GAT_250 651GAT_271 722GAT_295 763GAT_337 773GAT_351 789GAT_368 789GAT_368b 802GAT_372 808GAT_377 808GAT_377b 826GAT_391 837GAT_396 846GAT
_407 855GAT_418 863GAT_424 PO_863GAT_424
The paths from 126GAT_30 to PO_863GAT_424: 4
total CPU time = 0
```

CASE3:

./atpg -path -start 307GAT_18 -end PO_2548GAT_840 /path/circuits/iscas85/c6288.bench

```
476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2326GAT_742 2326GAT_742b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2326GAT_742 2326GAT_742b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2477GAT_823 2477GAT_823b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2326GAT_742 2326GAT_742b 2362GAT_763 2362GAT_763b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2477GAT_823 2477GAT_823b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
269GAT_716 2269GAT_716b 2327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2414GAT_782 2414GAT_782b 2477GAT_823 2477GAT_823b 2548GAT_840 PO_2548GAT_840
307GAT_18 549GAT_285 549GAT_285b 1763GAT_500 1763GAT_500b 1826GAT_543 1826GAT_543b 2005GAT_605 2005GAT_605b 2037GAT_622 2037GAT_622b 2085GAT_640 2085GAT_640b 2151GAT_682 2151GAT_682b 2
327GAT_745 2327GAT_745b 2362GAT_763 2362GAT_763b 2476GAT_816 2476GAT_816b 2548GAT_840 PO_2548GAT_840
The paths from 307GAT_18 to PO_2548GAT_840: 468
total CPU time = 0.03
```

Build:

make

./atpg -path -start <PI> -end <PO> <absolute path>

Ex:

```
./atpg -path -start 307GAT_18 -end PO_2548GAT_840 /home/Student113/s110305504/VLSI_Testing/Assignment0/circuits/iscas85/c6288.bench
```