VLSI Testing and Design for Testability Assignment 2

311510173 魏子翔

1. How to compile codes

   Go to the folder /podem, then enter “make” in command line to compile all codes.

2. The algorithm and idea of my codes

   A. Generate random pattern

   I apply the current time as the seed to get the rand(). And according to having unknown or not, if with unknown, num would be 3, otherwise num would be 2. Then rand_value could be 0, 1, 2 or 0, 1 respectively. Finally, I can output the pattern 0, 1, X(rand_value = 2) to the output file.

```
srand(time(0));
for(int i=0; i<num; i++)
{
    for(unsigned j=0; j<PIlist.size(); j++)
    {
        int rand_value = rand() % num_signal;
        if(rand_value == 2)
            ofile << "X";
        else
            ofile << rand_value;
    }
    ofile << endl;
}
```

   B. Logic simulation

   Let 00, 11, 10 represent the logic 0, 1, X in LogicEX CIRCUIT::Mod_Encode that can expand the logic 0, 1, X. Then, I can imply VALUE CIRCUIT::Mod_Evaluate(GATEPTR gptr) by the codes below.

```
case G_AND:
case G_NAND:
    for (unsigned i = 1;i<gptr->No_Fanin() && cv!=Mod_Decode(ec_value);++i) {
        ec_value_temp = Mod_Encode(gptr->Fanin(i)->GetValue());
        ec_value.leftbit = ec_value.leftbit & ec_value_temp.leftbit;
        ec_value.rightbit = ec_value.rightbit & ec_value_temp.rightbit;
    }
    break;
case G_OR:
case G_NOR:
    for (unsigned i = 1;i<gptr->No_Fanin() && cv!=Mod_Decode(ec_value);++i) {
        ec_value_temp = Mod_Encode(gptr->Fanin(i)->GetValue());
        ec_value.leftbit = ec_value.leftbit | ec_value_temp.leftbit;
        ec_value.rightbit = ec_value.rightbit | ec_value_temp.rightbit;
    }
    break;
```

After evaluating, the result 00, 11 , {10, 01} can be transformed into the logic 0, 1, X in LogicEX CIRCUIT::Mod_Decode.

3. Several case results

A. Generate random pattern

In command line, I entered

```
./atpg -pattern -num 10 -output ../c17_case1.input ../circuits/iscas85/c17.bench
```

and

```
./atpg -pattern -unknown -num 10 -output ../c17_case2.input ../circuits/iscas85/c17.bench
```

to get the random pattern w/wo unknown of c17.bench as below.

```
PI G1 PI G2 PI G3 PI G4 PI G5
11111
11100
01101
11001
10000
00111
11010
10101
01011
10100
```

```
PI G1 PI G2 PI G3 PI G4 PI G5
X1010
0XXX1
0XX1X
X11XX
00X11
X1X00
111XX
0X1XX
00000
X0110
```

Fig. the random pattern w/wo unknown of c17.bench

```
./atpg -pattern -num 15 -output ../c432_case1.input ../circuits/iscas85/c432.bench
```

and

```
./atpg -pattern -unknown -num 15 -output ../c432_case2.input ../circuits/iscas85/c432.bench
```

to get the random pattern w/wo unknown of c432.bench as below

```
PI 1GAT_0 PI 4GAT_1 PI 8GAT_2 PI 11GAT_3 PI 14GAT_4 PI 17GAT_5 PI 21GAT_6 PI 24GAT_7 PI 27GAT_8 PI 30GAT_9 PI 34GAT_10 PI 37GAT_11 PI 40GAT_12 PI 43GAT_13 PI 47GAT_14 PI 50GAT_15
001101011101110110110101101100110110111
010100001010110110101001000011111001
001001010001001001101100101001101101
001011111011110110011011000011111010
000101100010001010000100011101001010
100111111001001111110010110110100101
011101001111000111100011000010011111
010111010101001101001001000011111110
111100001010000001011110110001010101
100100100101011101011000011100100000
001011011100100110100011010111111010
010100100110100101010101010101101111
001010000011110100001011011100000011
011100010010011100101010110111101011
110111111001111000011101011100011100
```

```
PI 1GAT_0 PI 4GAT_1 PI 8GAT_2 PI 11GAT_3 PI 14GAT_4 PI 17GAT_5 PI 21GAT_6 PI 24GAT_7 PI 27GAT_8 PI 30GAT_9 PI 34GAT_10 PI 37GAT_11 PI 40GAT_12 PI 43GAT_13 PI 47GAT_14 PI 50GAT_15
X0X11XX011XXX1100X0X0101110XX0XXX100
101XX10X01011011XX110101X011XXX10110
0XX1X0X01100011XX0110X1X1010X10X010X
1100X011X10111011111X11XX1XXX00111XX
X101X10000X101X0010X00XX0X1X11X0001X
110100X1101X1X1XXXX0X1010X10X0X1X0X0
1X1001000XX01X11110111X1XXX00X0X1X00
01011XX0X0X1XXX0X1X10X111011XX101XX0
1X11X1XXXX0111X00XXXX10X1X0X1X101001
1X10XX110X11X1010101101X0X0XX00X1X01
10XXX11101000111X01X0X100XX1X0X10X10
0XX0100X1X100X01XX1XXX1X01111XX1X101
1100X10011100011X00X11010X0X0X10X1X
1XX0111X100X1000X000101110X10X1110XX
XXX1XX001X00X1X000X11001X110X10100X0
```

Fig. the random pattern w/wo unknown of c432.bench

B. Logic simulation

The output of c17.bench is same as the file provided on the website.

```
Start parsing input file
Finish reading circuit file
Run ASS2 Mod logic simulation
01110 00
10101 11
00101 01
01000 11
10001 01
00011 01
00111 00
00000 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
```

Fig. the output of c17.bench

The output of c7552.bench is also same as the file provided on the website.

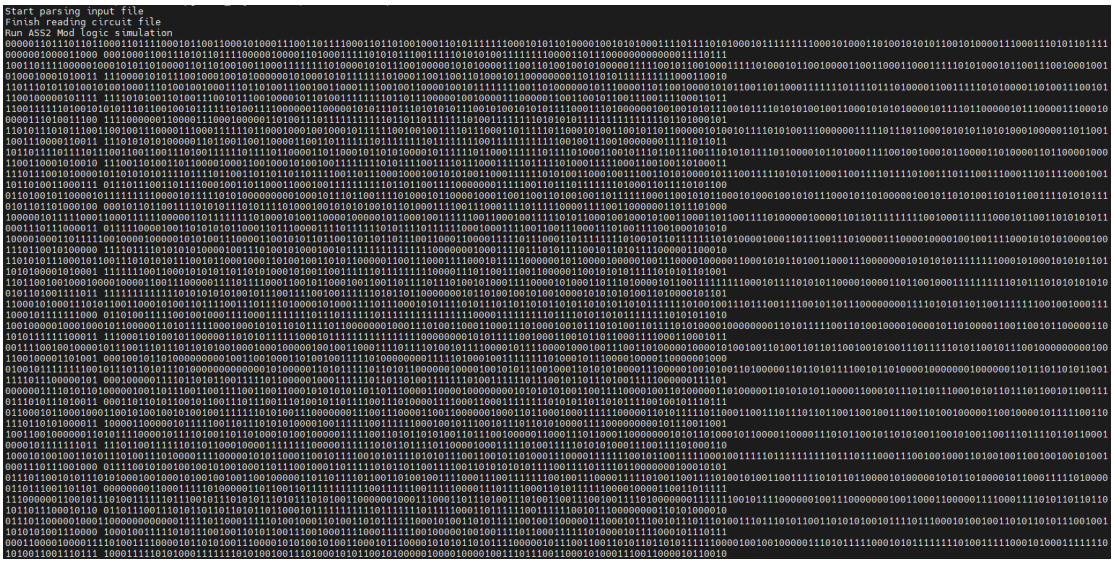

Fig. the output of c7552.bench

C. Discuss the results

I compared the result of -mod_logicsim to the result of -logicsim.

- c17.bench

  pattern

```
PI G1 PI G2 PI G3 PI G4 PI G5
X1010
0XXX1
0XX1X
X11XX
00X11
X1X00
111XX
0X1XX
00000
X0110
```

-mod_logicsim

```
[311510173@mseda03 podem]$ ./atpg -mod_logicsim -input ../c17_case2.input  ../circuits/iscas85/c17.bench
Start parsing input file
Finish reading circuit file
Run ASS2 Mod logic simulation
21010 11
02221 22
02212 22
21122 22
00211 02
21200 11
11122 12
02122 22
00000 00
20110 20
total CPU time = 0
```

-logicsim

```
[311510173@mseda03 podem]$ ./atpg -logicsim -input ../c17_case2.input ../circuits/iscas85/c17.bench
Start parsing input file
Finish reading circuit file
Run logic simulation
21010 11
02221 22
02212 22
21122 22
00211 02
21200 11
11122 12
02122 22
00000 00
20110 20
total CPU time = 0
```

- c432.bench

pattern

```
PI 1GAT_0 PI 4GAT_1 PI 8GAT_2 PI 11GAT_3 PI 1
X0X11XX011XXX1100X0X0101110XX0XXX100
101XX10X01011011XX110101X011XXX10110
0XX1X0X01100011XX0110X1X1010X10X010X
1100X011X10111011111X11XX1XXX00111XX
X101X10000X101X0010X00XX0X1X11X0001X
110100X1101X1X1XXXX0X1010X10X0X1X0X0
1X1001000XX01X11110111X1XXX00X0X1X00
01011XX0X0X1XXX0X1X10X111011XX101XX0
1X11X1XXXX0111X00XXXX10X1X0X1X101001
1X10XX110X11X1010101101X0X0XX00X1X01
10XXX11101000111X01X0X100XX1X0X10X10
0XX0100X1X100X01XX1XXX1X01111XX1X101
1100X10011100011 0X00X11010X0X0X10X1X
1XX0111X100X1000X000101110X10X1110XX
XXX1XX001X00X1X000X11001X110X10100X0
```

-mod_logicsim

```
[311510173@mseda03 podem]$ ./atpg -mod_logicsim -input ../c432_case2.input ../circuits/iscas85/c432.bench
Start parsing input file
Finish reading circuit file
Run ASS2 Mod logic simulation
20211220112221100202010111022022 2100 0001000
10122102010110112211010120112221 0110 0001000
02212020110001122011021210102102 0102 0001000
11002011210111011111211222202 2011122 0001000
21012100002101200102002202121 1200012 0001000
11010021101212122220210102102 0212020 0001000
12100100022012111101112122200 2021200 0001000
01011220202122202121021110112 2101220 0001000
12112122220111200222210212021 2101001 0001000
12102211021121010101101202022 0021201 0001000
10222111010001112012021002212 0210210 0001000
02201002121002012212221201111 2212101 0001000
11002100111000110200211010202 0210212 0001000
12201112100210002000101110210 2111022 0001000
22212200120021200021100121102 1010020 0001000
total CPU time = 0
```

-logicsim

```
[311510173@mseda03 podem]$ ./atpg -logicsim -input ../c432_case2.input ../circuits/iscas85/c432.bench
Start parsing input file
Finish reading circuit file
Run logic simulation
202112201122211002020101110220222100 0001000
101221020101101122110101201122210110 0001000
022120201100011220110212101021020102 0001000
110020112101110111112112212220011122 0001000
210121000021012001020022021211200012 0001000
110100211012121222202101021020212020 0001000
121001000220121111011121222002021200 0001000
010112202021222021210211101122101220 0001000
121121222201112002222102120212101001 0001000
121022110211210101011012020220021201 0001000
102221110100011120120210022120210210 0001000
022010021210020122122212011112212101 0001000
110021001110001102002110102020210212 0001000
122011121002100020001011102102111022 0001000
222122001200212000211001211021010020 0001000
total CPU time = 0
```