International Journal of Engineering &Technology, 7 (3.12) (2018) 354-356

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Input Bridging Fault and Non Feedback Bridging Fault Detection and Test Vector Generation by Boolean Satisfiability Method

K.Sarvani[1], B .Viswanathan[2]

[1]Mtech-VLSI Design, [2]Assistant professor (S.G)
[1,2]Department of Electronics and Communications Engineering, SRM University, India
**Corresponding Author Email: [1]Sarvani.kanduri@gmail.com, [2] viswanathan.b@ktr.srmuniv.ac.in*

Abstract

The input bridging faults and non feedback bridging fault(short between lines) occurs due to improper soldering.This paper enumerates the Fault detection and high quality test vector generation for these faults by using Boolean Satisfiability method with the help of JAVA code.This method can be adapted to deal with many faults in the combinational circuits.

Keywords: *Boolean Satisfiability, Input Bridging Fault, Non Feedback Bridging Fault, Test Vectors.*

1. Introduction

Bridging faults are most familiar in CMOS ICs [1].Bridging fault can be explained as shorting of two input lines or output lines or input and output lines of a circuit. These faults are different from 'Stuck at Faults'. The bridging faults are sorted into three types

- Input Bridging Fault
- Non feedback Bridging Fault
- Feedback Bridging Fault.

The input bridging faults occurs when two input lines of a logic gate are getting shorted. Then there is a clash between two input lines, if the lines are set to two different logics (If the gates belongs to TTL family then the input line is brought down to logic 0, in other case, if the gates belong to CMOS family then logic is brought up to 1 [2].

Non Feedback bridging faults arises when two Output lines or Input-Output lines of different circuits are short. The variation in input line logic depends on the family of gates it belong to.

In order to detect a fault and to generate test vectors Boolean Satisfiability method can be used. This method cannot be considered as a structural or algebraic method[3].This method is based on Boolean algebra. The equations expressed in Conjunctive normal form (POS) can be solved using this. This method is utilised to detect many faults like stuck at faults, open faults,delay faults both in regular and large circuits.

Boolean satisfiability is mainly depends on Davis-Putnam-Logemann-Loveland algorithm (DPLL or DLL). But by using this algorithm minimal test set cannot be generated. So to get minimal test set of high quality test vectorscode was generated in JAVA. This JAVA code generates all possible test vectors for the fault in a combinational circuit. So with the help of this code test vectors are generated by injecting fault for each and every path in circuit and finally minimal test set for the circuitis obtained. Initially the Boolean satisfiability was used by the computer science engineers but in recent times its usage extended into various fields.

This Boolean satisfiability method is highly adaptable and can be used instead of other traditional methods to detect and generate test vectors for both input bridging and non feedback bridging faults. This is the main advantage of Boolean satisfiability method and it sufficiently deals with all faults [4].

The programs generally used to check the Boolean Satisfiability are called SAT solvers. Checking the satisfiability is a NP-complete problem but by using a SAT solver one can have accurate results in less time. This Boolean satisfiability method is a three step method and can be explained as following.

2. Procedure for Detecting Input Bridging Fault

Boolean satisfiability method is sorted into 3 steps –

- Getting fault free or good circuit equations
- Getting faulty circuit
- Activation of propagation path

These steps play an important role in detection and this method requires the conjunctive normal form equation [5]. For example for NAND gate CNF (conjunctive normal form) is written as follows:

$$C1 = (A + C)\,(B + C)\,(-A - B - C)$$

The first two steps are same for input bridging fault and non feedback bridging fault.

A. Step-1 (fault free circuit equations)

In the first step equations are drawn out from the complete circuit. The CNF equation should be written involving each and every gate which is present in the combinational circuit and this step provides complete behavior of the circuit.Each and every line in the circuit is numbered by node numbers starting from 1.

To explain this procedure half adder circuit can be considered, which provides a better understanding.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This circuit is having bridging at i1 and i2 nodes and that fault should be propagated to primary outputs. For this circuit the fault free circuit equation is given as

$$CG = (-i3 + i1)(-i3 + i2)(i1 + i2 + i3)$$
$$(i6 - i4)\ (i6 - i5)(-i4 - i5 - i6)(-i7 + i1)(-i7 + i2)$$
$$(i8 - i5)\ \ (-i8 - i4 - i5)\ (-i9 + i2)(i9 - i2)(i10 -$$
$$i5)(-i10 - i5)(-i10 - i9 - i2)\ (i8 - i5)$$

In the above equations negative values (of the nodes) indicates the logic 0 and positive values indicates the logic 1 value.
For example, Half adder circuit with input bridging fault can be considered, as it contains input bridging in the input side for nodes i1 and i2. Then, the affected node is represented with another node as i11 and the effect of thefault should be propagated to output through i12 to i13.

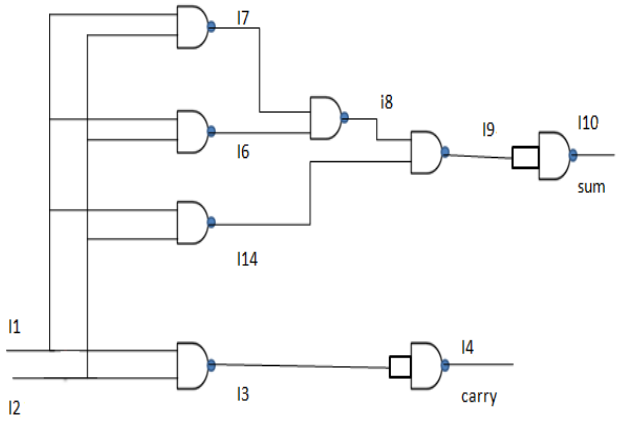

Fig. 1: Half adder circuit with NAND gates

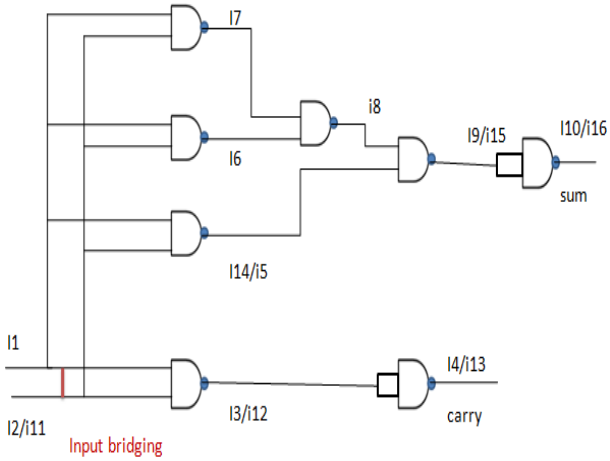

Fig. 2: Half adder circuit with an input bridging fault

B. Step-2 (faulty circuit equations)

The logic gates which are affected with input bridging fault may give faulty values which are different from fault free values. For these gates alone CNF equations have to be written by giving different node numbers to the affected lines. Faulty behaviour of the circuit is determined by this CNF equations. The equations are as follows

$$CF = \ (i11 + i12)(i1 + i12)(-i1 - i12 - i11)$$
$$(i12 + i13)\ (-i12 - i13)$$
$$(i1 + i14)(i11 + i14)(-i1 - i11 - i14)(i8 + i15)$$
$$(i15 + i14)(-i8 - i15 - i14)(i1 + i16)$$
$$(-i15 - 16)(i10 + i16)\ (-i10 - i16)$$

Where CF represents equation for faulty circuit and CG represents equation for good circuit.

C. Step-3 (Activation of Propagation Path)

This is the most important step in the procedure. In order to activate the path, CNF equation of the gates which are involved between the fault location and output are considered. The propagation path considered should be shortest one. If testing time is less, costs associated with testing are proportionally less. For example consider half adder circuit with input bridging fault. The effect of the fault is propagated to primary outputs. XOR operation has been done to the faulty values and fault-free values.CA indicates activation of path

$$CA = \ (i1 - i11)(i11 - i1)(i13 + i12)(-i13 - i12)(-i14 -$$
$$i15)\big(-i8 - i15 - i14\big)(i1 + i16)(-i7 + i2)(i7 + i1 + i2)(i8 -$$
$$i4)\ (i10 - i5)(-i10 - i5)$$

Let Ø represents whole half adder circuit operation [6].

Ø = CG.CF.CA =1

Depend on Ø value the circuit is satisfiable or unsatisfiable. If Ø gives value 1 then it is satisfiable else it is unsatisfiable. If the circuit is satisfiable then JAVA code gives test vector.
The test vector for the half adder circuit with input bridging fault is: 10. Since there are only two primary inputs we cannot generate minimal test set. The input combinations which detect the fault are considered as test vectors [7].

3. Procedure for Detecting Non Feedback Bridging Fault

The non feedback bridging faults are different from the feedback bridging faults. This occurs when two logic gate outputs or input of one gate and output of another gate are short. These faults can also be detected by using Boolean satisfiability method .The first two steps are same for input bridging and non feedback bridging fault.
The steps are as follows

A. Step-1 (fault free circuit equations)

For this circuit also node numbers are given in order to identify the faulty value and fault free value, here i1,i2,i3,i4,i5,i6 are primary inputs. I15 is primary output. For example non feedback bridging is between i8 and i7 (i.e the output of AND gate and XOR gate) likewise the non feedback bridging is injected and tested at each and every point in the circuit except at primary inputs

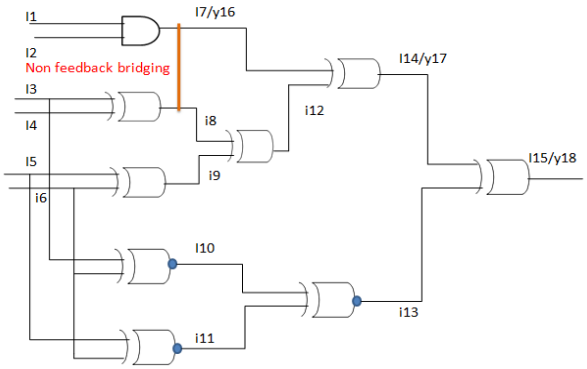

Fig. 3: Error detecting and correcting circuit with non-feedback bridging fault

To propagate the faulty value only one path should be considered. The fault free equations are given as

$$CG = (i9 - i7)(-i3 - i7 - i9)(-i3 + i13)$$
$$(-i12 + i13)(-i3 - i13 - i12)(-i15 + i10)(-i14 + i6)$$
$$(-i14 + i12)(i14 + i12 + i6)(i10 + i13 + i15)$$
$$(-i11 + i16)\ \ (-i14 + i16)\ \ (-i11 - i14 - i16)$$
$$(i7 + i14 + i12)(i5 - i6 - i11)(i10 - i11 - i13)$$

$$(i14 + i15 - i13)(i14 - i15 + i13)(-i5 - i6 - i11)$$
$$(i7 + i12 - i14)\ (-i14 - i13 - i15)\ \ (i14 + i15)$$
$$(i12 + i14 + i11)\ (-i15 + i10)$$

B. Step-2 (faulty circuit equations)

Here the fault is in between output lines of a gate but not in the input lines. This fault is promoted to the output through the gates and the equations for the gates which are affected by the bridging faults are:

$$\boldsymbol{CF} = \ (i8 + i15)(i15 + i14)(-i8 - i15 - i14)$$
$$(i15 + i16)(-i15 - 16)(i3 + i1 + i2)(-i4 + i6)(-i5 + i6)$$
$$(-i15 + i10)(i10 + i13 + i15)(-i11 + i16)(-i14 + i16)$$
$$(-i11 - i14 - i16)(i17 + i7 + i22)\ \ (-i14 + i12)$$
$$(i14 + i12 + i6)\ \ (-i15 + i10)\ \ (i10 + i13 + i15)$$

C. Step-3 (Activation of Propagation Path)

This step involves the activation condition. Primary inputs i1 and i3 should be equal and i2,i4 should be equal to detect the fault. XOR operation has been done to the fault free gates and the faulty gates such that both should not be equal. Shortest path is selected for activation. So, y16-y17-y18 nodes are selected to propagate the effect of the fault to the output. One can test any circuit using the same three steps.

The equations are

$$CA\ = \ (-i24 + i30)(i24 + i10 + i30)(-i24 + i27)$$

$$(i24 + i27 + i13)(-i14 + i12)(i14 + i12 + i6)$$
$$(-i15 + i10)(-i25 - i14 - i20)(-i26 + i15 + i21)$$
$$(-i26 - i15 - i21)\ \ (i27 + i3 - i16)\ \ (i27 + i16 - i3)$$
$$(i28 + i16 - i5)\ \ (i28 - i26 - i5)\ \ (-i228 - i26$$
$$- i5)\ \ (-i128)\ (i22)\ \ (i23)\ \ (i24)\ \ (i25)\ \ (i25)\ (i26)$$

Let Ø represents whole error detecting and correcting circuit operation

Ø = CG.CF.CA =1

The minimal test set of error detecting and correcting circuit for non feedback bridging fault which is injected at every point in the circuit is:

{010100,111100,111111,111110,101000,010100,101011,010111, 011010,000101,111001}

The minimal test set of error detecting and correcting circuit for input bridging fault is:

{1011100,100001,1001000,101011,101001,101100,111100,11111 0,110110,000110,010110}

4. Results

By using Boolean satisfiability method input bridging and non feedback bridging faults are detected for Bench Mark circuits C432, B02, C499, C880 and B01. Fault coverage estimates for total number of faults and faults detected by Boolean satisfiability method. The fault coverage equation is given by

Fault coverage= (faults detected / total number of faults injected) x 100

These results are obtained for non feedback bridging faults and input bridging faults by injecting faults at every node.

The results are obtained by using JAVA code. This JAVA code is used to solve Boolean satisfiability problem in less time. For Boolean satisfiability method conjunctive normal form equations are used.

Table 1: Input bridging fault detection and test vector generation for benchmark circuits

Bench mark circuit	No of primary inputs	No of primary outputs	No of gates	Faults injected	Faults detected	Fault coverage (%)
B02	5	5	26	18	18	100
C432	36	7	152	40	39	97.5
C499	41	32	320	70	70	100
C880	60	26	311	75	74	98.66
B01	7	7	51	12	12	100

Table I and Table II are used to estimate the fault coverage for the Benchmark circuits. The fault coverage for the input bridging fault and non feedback bridging fault is above 95%.

It proves that Boolean satisfiability method is best method for detecting faults and generating test vectors than traditional methods.

Table 2: Non feedback bridging fault detection and test vector generation for benchmark circuits

Bench mark circuits	Faults injected	Faults detected	Fault coverage (%)
B02	22	21	95.45
C432	67	65	97.01
C499	75	75	100
C880	80	79	98.75
B01	20	20	100

5. Conclusion

The Boolean satisfiability method is not limited to detect stuck-at-faults, open faults, delay faults only. It detects bridging faults i.e. input bridging fault and non feedback bridging faults also.

It proves that single method is sufficient to detect different types of faults. It is good achievement compared to conventional methods. The traditional methods has different procedure for each type of fault. Some of those procedures are quite complex. Moreover Boolean satisfiability method produces quality test vectors by using JAVA code. The fault coverage is achieved by this Boolean satisfiability method for ISCAS bench mark circuits B02, B01, C432, C499 and C880.

References

[1] Michele Favalli and Marcello Dalpasso "Boolean and PseudoBoolean Test Generation for Feedback Bridging Faults" IEEE Trans on Computers, VOL. 65, NO. 3, MARCH 2016

[2] Soumitrakumarmandal,"Digital electronics principles and applications" Tata Mc GrawHill private limited ,New Delhi

[3] T. Larrabee, "Test pattern generation using Boolean satisfiability"IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 11,no. 1, pp. 4–15, Jan. 1992.

[4] R. Drechsler, et al., "Test Generation Using Boolean Proof Engines".New York, NY, USA: Springer Verlag, 2009.

[5] M. Favalli and M. Dalpasso, "How many test vectors we need to detect a bridging fault?" J. Electron. Testing, vol. 25, no. 1, pp. 79–95, 2009.

[6] Chih-Ang Chen and Sandeep K. Gupta, "Satisfiability-Based Test Generatorfor Path Delay Faults in Combinational Circuits "Design Automation Conference Proceedings 1996.

[7] A. Biere, M. Heule, H. V. Mareen, and T. Walsh, Eds.,"Handbook of Satisfiability." Amsterdam, The Netherlands: IOS Press, 2009.