Finding False Paths for Sequential Circuits Using Operations on ROBDDs

Anzhela Matrosova
Institute of Applied Mathematics and Computer Science
Tomsk State University
Tomsk, Russsia
mau11@yandex.ru

Sergei Ostanin
Institute of Applied Mathematics and Computer Science
Tomsk State University
Tomsk, Russsia
sergeiostanin@yandex.ru

Semen Chernyshov
Institute of Applied Mathematics and Computer Science
Tomsk State University
Tomsk, Russsia
semen.cher@mail.ru

***Abstract*—Performance of VLSI is, first of all, its high operation speed determined by a clock frequency. Developing of VLSI is oriented to maximal possible clock frequency under correct functioning. Clock frequency estimation is reduced to finding paths with maximal delays (critical paths) among logical components of VLSI. But some of the selected paths may be false. It means that the path has no impact on component functioning. It is necessary to find such paths in order to exclude them from consideration when we determine clock frequency. Detecting false paths may increase VLSI operation speed. The precise method of finding false paths in a sequential circuit based on finding test pairs for non-robust path delay faults (PDFs) is developed. The length of a transfer sequence delivering the test pair from the initial internal state is not more the given value *l*. The method is based on applying operations on ROBDDs extracted from the combinational part of a sequential circuit. Experimental results illustrate the suggested method.**

Keywords—sequential circuit, transfer sequence, Reduced Ordered Binary Decision Diagram (ROBDD), false path, Path Delay Fault (PDF)

I. Introduction

If changing of input signal of the path does not generate changing of output signal of the path for any input Boolean vector of a circuit, the path is false.

It is known that the number of false paths among critical paths may reach 80% [1]. Determining clock frequency without seeking for false paths may decrease operation speed.

There are heuristic methods of finding false paths in combinational circuits [2-5].

In [6] a precise method of detecting false paths in a sequential circuit is suggested. It is based on finding test pairs (v_1, v_2) for non-robust path delay faults (PDFs) and confirming existence of transfer sequence delivering such test pairs from the initial internal sate. The length of the transfer sequence is not more given value *l*. A disadvantage of this method consists in applying an equivalent normal form (ENF) in order to find test patterns v_2 of test pairs (v_1, v_2) that detect non-robust PDFs in the combinational part of a sequential circuit.

Here we overcome this disadvantage applying only operations on ROBDDs. The ROBDDs are obtained for fragments of combinational parts of sequential circuits. We derive all test patterns v_2 for rising and fallings transition of a path considered in a combinational part of a sequential circuit. These test patterns are compactly represented by two ROBDDs (for rising and falling transitions separately). We find out existence of transfer sequences (with length not more *l*) delivering test pairs from the initial internal state. We don't find the transfer sequence itself, but only confirm its existence. It allows executing some operations on ROBDDs depending on only state variables of a combinational part of a sequential circuit.

For finding of false paths we suggest considering only non-robust PDFs. If there is no test pair detecting non-robust PDF neither for rising, nor falling transitions in a combinational circuit (in the combinational part of a sequential circuit), the path is false for the combinational circuit and for the sequential circuit. For detecting false paths in a combinational circuit it is enough to find only vector v_2 for one of the test pairs. If we have v_2 , any vector that differ from v_2 by the variable marking the beginning of the path considered is v_1. For sequential circuit we have to find both vectors (v_1, v_2) for test pairs for rising or falling transitions. In this case, we consider a path as false one if there is no a test pair detecting non-robust PDF neither for rising nor falling transition under condition that this test pair is reachable from the initial internal state q_0 by a transfer sequence with length not more *l*.

Note that the values of the variable corresponding to the beginning of the path must be mutually inverse in test pairs (v_1, v_2): vector v_1 is previous one and vector v_2 is next one. After getting test pairs (v_1, v_2) it is necessary to confirm existence of transfer sequence that delivers the proper test pair from the initial internal state q_0.

II. Deriving Boolean difference for path α

Represent path α of one output combinational circuit *C* with inputs $x_1,\ldots, x_n$ by sequence of symbols: x_i, u_1, $u_2,\ldots, u_{(r-1)}$, u_r. Here *r* is path α length, x_i is the path input (circuit *C* input), u_1, $u_2,\ldots, u_{(r-1)}$, u_r – the path element (gate) outputs. Output u_r is the circuit *C* output. Circuit *C* may belong to a combinational part of a sequential circuit.

Let u_i, $u_{(i-1)}$ be outputs of neighbor elements of path α. Consider sub-circuit C_{u_i} of circuit *C* with output u_i and inputs $x_1,\ldots, x_n$, $u_{(i-1)}$. Here $u_{(i-1)}$ is input variable of sub-circuit C_{u_i} and at the same time is input variable of the element with output u_i.

The reported study was partially supported by Russian Science Foundation, research project № 14-19-00218.

Note as $D_{u_i} / D_{u_{(i-1)}}$ Boolean difference of the function implemented by sub-circuit C_{u_i} with respect to variable $u_{(i-1)}$.

To calculate $D_{u_i} / D_{u_{(i-1)}}$ it is necessary to execute the following steps.

1. Derive ROBDD $R(f_{u_i})$ representing the function of sub-circuit C_{u_i}. The ROBDD depends on variables $x_1,\ldots, x_n, u_{(i-1)}$. The ROBDD is obtained by using variable $u_{(i-1)}$ as the first one of Shannon decomposition.

2. Obtain from $R(f_{u_i})$ two ROBDDs: $R(f_{u_i}^{u_{(i-1)}=0})$, $R(f_{u_i}^{u_{(i-1)}=1})$ which roots are children nodes of ROBDD $R(f_{u_i})$ root. These ROBDDs implement functions that are obtained from function f_{u_i} by changing variable $u_{(i-1)}$ for constant 0, 1, accordingly.

3. Multiplications $f_{u_i}^{u_{(i-1)}=0}$, $\overline{f}_{u_i}^{u_{(i-1)}=1}$, and $f_{u_i}^{u_{(i-1)}=1}$, $\overline{f}_{u_i}^{u_{(i-1)}=0}$ are executed and results are merged being represented by ROBDD $R(D_{u_i} / D_{u_{(i-1)}})$:

$$R(D_{u_i} / D_{u_{(i-1)}}) = R(f_{u_i}^{u_{(i-1)}=0}) \,\&\, R(\overline{f}_{u_i}^{u_{(i-1)}=1}) \vee$$

$$\vee\, R(f_{u_i}^{u_{(i-1)}=1}) \,\&\, R(\overline{f}_{u_i}^{u_{(i-1)}=0}) \qquad (1)$$

Note that ROBDD operations have a polynomial complexity.

Execute multiplications:

$R(D_{u_r} / D_{u_{(r-1)}})$ & $R(D_{u_{(r-1)}} / D_{u_{(r-2)}})$ &…& $R(D_{u_2} / D_{u_1})$ & $R(D_{u_1} / D_{x_i})$.

As a result we obtain Boolean difference D_{path} for path α representing it by $R(D_{path})$.

Boolean vector μ turning $R(D_{path})$ into 1and circuit C into 1 is vector v_2 for rising transition. If vector μ turns circuit C into 0, it is vector v_2 for falling transition [7].

If $R(D_{path})$ is empty, path α is false as for combinational circuit and for sequential circuit.

If $R(D_{path})$ is not empty, path α is not false in combinational circuit. However, for sequential circuit it is necessary to deliver test pair (v_1, v_2) from the initial internal state q_0 of a sequential circuit by a transfer sequence with length not more l. The values of the variable corresponding to the beginning of path α must be mutually inverse in the test pairs (v_1, v_2). It is not always possible.

Note that vectors (v_1, v_2) represent full states of sequential circuits. Parts of their full states correlated with input variables of a sequential circuit represent last but one and last elements of the transfer sequence from initial internal state q_0, correspondingly.

III. Deriving all test patterns for rising and falling transitions from ROBDD $R(D_{PATH})$

Thus, ROBDD $R(D_{path})$ presents test patterns for rising and falling transitions all together. It is necessary to divide them. ROBDD for rising transition call R_{rise}:

$R_{rise} = R(D_{path})\&x_i\,(R(D_{path})\&\,\overline{x}_i)$.

ROBDD for falling transition call R_{fall}:

$R_{fall} = R(D_{path})\&\,\overline{x}_i\;\;(R(D_{path})\&x_i)$.

Consider the combinational circuit in Fig. 1.

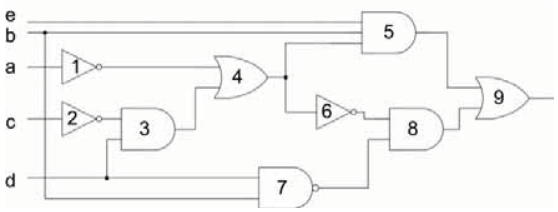

Fig. 1. Combinational circuit C

Let path α includes gates with numbers 3, 4, 6, 8, 9. The beginning of the path is marked by variable d. The path α may be represented by the sequence of symbols: $d, u_3, u_4, u_6, u_8, u_9$.

Calculate D_{path} for α representing results by SoPs.

$$D_{u_9} / D_{u_8} = (u_5 \vee (u_8 = 1)) \oplus (u_5 \vee (u_8 = 0)) = \overline{u}_5$$

$$= \overline{b} \vee \overline{e} \vee ac \vee a\overline{d}\,.$$

$$D_{u_8} / D_{u_6} = (u_6 = 1)\,\&\,u_7 \oplus (u_6 = 0)\,\&\,u_7 = u_7 = \overline{b} \vee \overline{d}\,.$$

$$D_{u_4} / D_{u_3} = (\overline{a} \vee 1) \oplus (\overline{a} \vee 0) = a\,.$$

$$D_{u_3} / D_d = (\overline{c}\,\&\,1) \oplus (\overline{c}\,\&\,0) = \overline{c}\,.$$

$$D_{path} = (\overline{b} \vee \overline{e} \vee ac \vee a\overline{d})\,\&\,(\overline{b} \vee \overline{d})\,\&\,a\,\&\,\overline{c} = a\overline{c}\overline{b} \vee a\overline{c}\overline{d}\,.$$

Find SoP$_{rise}$, representing all vectors v_2 for rising transition and SoP$_{fall}$ representing all test patterns for falling transition.

$$\mathrm{SoP}_{rise} = (a\overline{c}\overline{b} \vee a\overline{c}\overline{d})\,\&\,\overline{d} = a\overline{c}\overline{d}\,.$$

$$\mathrm{SoP}_{fall} = (a\overline{c}\overline{b} \vee a\overline{c}\overline{d})\,\&\,d = a\overline{c}\overline{b}d\,.$$

IV. Revealing of transfer sequence existence

Pick out from ROBDD R_{rise} internal states and form from them the sum of products (SoP). Obtain ROBDD R^{s_0} using this SoP. The ROBDD represents all internal states that are contained in full states represented by vectors v_2 among test pairs (v_1, v_2) for rising transition of path α. Note that we have to deliver test pair (v_1, v_2), first vector v_1, and then vector v_2 from initial internal state

q_0. Full states represented by these vectors surely differ by inverse value of the variable marking the beginning of path α. The variable may be either internal or input one.

Thus, we have to reveal existence of the transfer sequence with the length not more l under the above mentioned restriction.

Procedure of revealing existence of the transfer sequence from initial state q_0 to some state from a set of internal states of a sequential circuit is suggested in [8]. A correction of this procedure oriented to the restriction is proposed in [6].

If there is no the transfer sequence for internal states generated by R_{rise}, it is necessary to reveal existence of the transfer sequence for internal states generated by R_{fall}. If there is no the transfer sequence in this case, path α is considered as false one.

V. Experimental results

Experiments on MCNC benchmark sequential circuits were performed. For each circuit, 100 paths of maximum length from primary inputs to primary outputs were chosen. If the number of paths is less than 100 then all paths were considered. The test patterns for rising and falling transitions (R_{rise} and R_{fall}) for each path were obtained. Existence of transfer sequences for each example with the length not more 100 is identified. Results are presented in Table I. Here are the name of benchmark circuit (**circuit**), the number of inputs (**num_i**), the number of outputs (**num_o**), the number of flipflops (**num_ff**), the number of false paths in percent among of all chosen paths (**num_f_p(%)**), the maximum length of transfer sequences (**MAX_length**), the average length of transfer sequences (**AVG_length**), the number of transfer sequences with length equal to 1 in percent (**num_t_s_1(%)**).

TABLE I. Experimental results

circuit	num_i	num_o	num_ff	num_f_p (%)	MAX_length	AVG_length	num_t_s_1 (%)
S01494	8	19	8	1	20	10.82	12.4
bbara	4	2	4	3	6	2.5	30.8
bbsse	7	7	4	7	9	3.44	25.2
bbtas	2	2	3	2.63	5	2.09	40.3
beecount	3	4	3	6.5	3	1.3	74.8
cse	7	7	4	10	7	2.81	43.3
dk14	3	5	3	1	3	1.27	72.88
dk15	3	5	2	0	2	1.24	76.1
dk16	2	3	5	7	4	2.44	8.33
dk17	2	3	3	3	4	1.93	42.28
donfile	2	1	5	8	4	2.4	20
ex1	9	19	5	9	8	2.21	45.38
ex2	2	2	5	4	6	2.66	23.68
ex3	2	2	4	2	3	1.84	31.36
ex4	6	9	4	4	14	6.86	13.12
ex6	5	8	3	3	2	1.41	58.82
ex7	2	2	4	3	4	2.45	19.73
keyb	7	2	5	6	7	2.19	57.03
kirkman	12	6	4	17	9	4.03	14.29
lion	2	1	2	0	4	1.71	53.66
lion9	2	1	4	15.7	4	2.33	31.94
mark1	5	16	4	15	6	3.09	18.92
mc	3	5	2	3.7	3	1.54	58.43
opus	5	6	4	6	6	2.18	53.49
planet	7	19	6	29	22	10.51	14.44
s1	8	6	5	1	5	2.97	18.92
s1a	8	6	5	0	5	2.43	43.82
s8	4	1	3	0	3	1.82	29.58
sand	11	9	5	6	4	1.86	37.25
scf	27	56	7	15	15	10.43	4.28
sse	7	7	4	7	9	3.44	25.18
styr	9	10	5	11	7	3.13	39.16
tav	4	4	2	0	3	1.25	84.06
tbk	6	3	5	0	3	1.56	50.57

Experiments show that up to 29% of the paths are false (**planet**). The average length of transfer sequence among all circuits is 3.12. The number of transfer sequences with length equal to 1 is up to 84.06% (**tav**).

VI. Conclussion

The precise method of finding false paths in a sequential circuit is developed. It is based on getting test pairs for non-robust PDFs. The test pair is delivered by the transfer sequence from the initial internal state of a sequential circuit. The sequence length is not more given value l. All test patterns v_2 from test pairs are derived separately for rising and falling transitions of the path with using operations on ROBDDs. Test patterns are compactly represented by two ROBDDs. Transition sequences are also found by using operations on ROBDDs.

References

[1] D. H. C. Du, S. H. C. Yen, and S. Ghanta, "On the General False Path Problem in Timing Analysis," in 26th ACM/IEEE Design Automation Conference, 1989, pp. 555–560.

[2] Jing Zeng, M. Abadir, and J. Abraham, "False Timing Path Identification Using ATPG Techniques and Delay-Based Information," in Proc. of the 39th Design Automation Conference, New Orleans, LA, USA, 2002, pp. 562–565.

[3] F. S. Marques, R. P. Ribas, S. Sapatnekar, and A. I. Reis, "A New Approach to the Use of Satisfiability in False Path Detection," in Proceedings of the 15th ACM Great Lakes Symposium on VLSI, New York, NY, USA, 2005, pp. 308–311.

[4] M. A. Kacou, F. Ghaffari, O. Romain, and B. Condamin, "Error rate estimation of a design implemented in an FPGA based on the operating conditions," in Proc. of the 2017 IEEE East-West Design & Test Symposium (EWDTS), Novi Sad, Serbia, 2017, pp. 1–7.

[5] A. Y. Matrosova, S. A. Ostanin, and V. Singh, "Detection of false paths in logical circuits by joint analysis of the AND/OR trees and SSBDD-graphs," Automation and Remote Control, vol. 74, no. 7, pp. 1164–1177, Jul. 2013.

[6] A. Y. Matrosova, V. V. Andreeva, S. V. Chernyshov, S. V. Rozhkova, and D. V. Kudin, "Finding False Paths in Sequential Circuits," Russian Physics Journal, vol. 60, no. 10, pp. 1837–1844, Feb. 2018.

[7] A. Y. Matrosova and V. B. Lipskii, "Properties of pairs of test vectors detecting path delay faults in high performance VLSI logical circuits," Automation and Remote Control, vol. 76, no. 4, pp. 658–667, Apr. 2015.

[8] A. Matrosova, V. Andreeva, and A. Melnikov, "ROBDDs application for finding the shortest transfer sequence of sequential circuit or only revealing existence of this sequence without deriving the sequence itself," in Proc . of the 2016 IEEE East-West Design Test Symposium (EWDTS), 2016, pp. 1–4.