2015 IEEE 18th International Symposium on Design and Diagnostics of Electronic Circuits & Systems

Increasing Manufacturing Yield Using Partially Programmable Circuits with CLB implementation of Incompletely Specified Boolean Function of the Corresponding Sub-circuit*

A. Matrosova, S. Ostanin, I. Kirienko
Tomsk State University
Tomsk, Russia
mau11@yandex.ru, sergeiostanin@yandex.ru, irina.kirienko@sibmail.com

***Abstract*—The new approach to partially programmable circuit design that allows masking arbitrary gate faults of a logical circuit is considered. It is supposed that only one gate may be fault. There are reserved Configurable Logic Blocks (CLBs) based on Look Up Tables (LUTs) that may mask a gate fault. The suggested approach in comparison with the currently in use ones allows masking any gate fault but not the certain stuck-at faults at the gate poles.**

Keywords—stuck-at faults, test patterns, incompletely specified Boolean functions, partially programmable circuit.

I. INTRODUCTION

The growing size, density and complexity of modern VLSI chips are contributing to increase hardware faults and design errors in the silicon, decreasing the manufacturing yield and increasing the design cycle [1]. There has been investigated several practical techniques to increase the yield of VLSIs [2-4]. A general technique is to add space redundancy, e.g., Triple Modular Redundancy (TMR). If we are considering only manufacturing defects, another possible way to increase the yield is Double Modular Redundancy (DMR). In this case we select a module without a defect between identical modules after the VLSI test. Embedded FPGAs have also been used for yield improvement [5-6]. However, the above methods have disadvantages: area overhead and/or performance degradation. In [7, 8] the different approaches to increase the yield with lower overhead have been proposed. There has been used a circuit obtained from a conventional logic circuit from gates by replacing its sub-circuits with Look-Up Tables (LUTs) and Multiplexers (MUXs). Then, if we detect some defects (by the VLSI test) in it, we reconfigure the functionality of some LUTs and MUXs to bypass the defects (single stuck-at faults). This kind of circuits is called Partially Programmable Circuits (PPCs) since some of their parts are programmable. The approach is based on the representation of the sub-circuit behavior of a combinational circuit by the incompletely specified Boolean function.

In this paper we compactly (with two ROBDDs – Reduced Ordered Binary Decision Diagrams) present an incompletely specified Boolean function and then use one of its implementation. Arbitrary gate faults of a combinational circuit are considered. Only one gate may be fault. The suggested method guarantees masking any gate fault. We suggest using a multiplexer in each combinational circuit branch point. When the fault of a gate is detected, the proper replacing sub-circuit from reserved CLBs is constructed. The sub-circuit inputs are connected with the combinational circuit inputs. The sub-circuit output is connected with the circuit through the closest multiplexer (MUX) in the branch point. MUXs select lines are controlled by programmable memory cells. The sub-circuit masks either the arbitrary fault of the gate which output is the branch point or the arbitrary fault of any gate for which this branch-point is the closest one but its own output is not a branch point. If the combinational circuit considered is fault free its performance does not degrade in contrast to [7]. The circuit degrades only by linking-up sub-circuit.

In Section 2 the possibility of describing behavior of a sub-circuit of a combinational circuit with the incompletely specified Boolean function represented by two ROBDDs is considered. In Section 3 the ways of masking faults are discussed. The experimental results are presented in Section 4.

II. ABOUT IMPLEMENTATIONS OF THE INCOMPLETELY SPECIFIED BOOLEAN FUNCTION

Let v be a branch point of a combinational circuit C consisting of gates and having single output. Correlate each branch point to a multiplexor. The output (pole v) of a fault free gate corresponding to a branch point is connected with one MUX input. If the gate with output pole v is fault then the output of the special masking sub-circuit is connected with another MUX input. The sub-circuit consisting of CLBs masks arbitrary fault of the gate. In both cases the MUX output becomes the branch point.

Describe a derivation of a masking sub-circuit. Correlate the stuck-at fault 0 ($\sigma = 0$) and the stuck-at fault 1 ($\sigma = 1$) to the pole v. We use these faults to represent the incompletely specified Boolean function describing sub-circuit behavior (the sub-circuit is included into circuit C). The real fault of gate with the output pole v may be arbitrary. In the papers [9, 10] the method of finding all test patterns for the fault $\sigma = 0$ ($\sigma = 1$) and presenting them by the ROBDD is suggested. It is based on results obtained in the paper [11].

Let f be the function that is implemented by the fault free sub-circuit with the output pole v. Let f_v^0 be the function that represents all test patterns for the fault $\sigma = 0$ at the pole v. Let f_v^1 be the function that represents all test patterns for the fault $\sigma = 1$ at the pole v. Let $\varphi(X)$ be the function implemented by the circuit C and $\varsigma(X, v)$, $X = \{x_1,\ldots, x_n\}$, be the function implemented by the sub-circuit of circuit C in which the pole v is considered as the input

*The reported study was partially supported by RSCF, research project № 14-19-00218.
Participation in DDECS2015 is partially supported within the "Program of increase of the international competitiveness of Tomsk State University for 2013-2020".

978-1-4799-6780-3/15 $31.00 © 2015 IEEE
DOI 10.1109/DDECS.2015.64

variable along with a set X and the output coincides with circuit C output: $\varsigma(X, f) = \varphi(X)$.

We conclude that any test pattern for the fault $\sigma = 0$ ($\sigma = 1$) at the pole v turns the function f ($\bar{f}$) into 1 and turns the function $\varphi(X)$ either into 1 or into 0. Moreover changing the value at the pole v from 1 to 0 (from 0 to 1) for the fault $\sigma = 0$ ($\sigma = 1$) generates changing the value at the circuit output from 1 to 0 if the test pattern turns fault free circuit C into 1 and from 0 to 1 if the test pattern turns fault free circuit C into 0. Note that if a Boolean vector of the length n turns the function f into 0 (into 1), then it cannot be a test pattern for the fault $\sigma = 0$ ($\sigma = 1$) at the pole v.

They say f includes g (g is an implicant of f), $g \le f$, if for any Boolean vector α of the length n from the condition $g(\alpha) = 1$ follows the condition $f(\alpha) = 1$.

For the fault $\sigma = 0$ we have: $f_v^0 \le f$, for the fault $\sigma = 1$: $f_v^1 \le \bar{f}$.

Let the condition $f_v^0 \le \varepsilon \le \bar{f}_v^1$ for the function ε takes place.

Theorem 1. $\varsigma(X, f) = \varsigma(X, \varepsilon)$, that is changing the function f for the function ε does not alter the function $\varphi(X)$.

Let the function γ be as follows: $f_v^1 \le \gamma \le \bar{f}_v^0$.

Theorem 2. $\varsigma(X, f) = \varsigma(X, \bar{\gamma})$, that is changing the function f for the function $\bar{\gamma}$ does not alter the function $\varphi(X)$.

Explain Theorems 1, 2 using the Venn diagrams in Fig. 1. Here all spots of the rectangle represent all Boolean vectors of the length n. These spots are divided into the on-set M_f^1 and the off- set M_f^0 of the completely specified function f by wavy line. The oval inside of M_f^1 area presents the on-set of the function f_v^0 corresponding to a set of all test patterns for the fault $\sigma = 0$ at the pole v of circuit C. The oval inside of M_f^0 area presents the on-set of the function f_v^1 corresponding to a set of all test patterns for the fault $\sigma = 1$ at the pole v of circuit C. The dashed part of the rectangle is don't care area of the incompletely specified Boolean function. The on-set and the off-set of this function are represented with the on sets of the functions f_v^0, f_v^1, correspondingly.

Taking into consideration Theorems 1, 2 we may choose as a masking circuit the circuit that implements either the function ε, or the function $\bar{\gamma}$. Both these functions may be represented with the corresponding ROBDDs. The chosen ROBDD may be covered with CLBs (to get the circuit whose behavior is represented by this ROBDD) in the way described in the paper [12].

Proposition. The behavior of any sub-circuit of a combinational circuit C is described with the incompletely specified Boolean function f_i. The on-set of this incompletely specified function is represented by the on-set of the function f_v^0 *and the off- set of f_i is represented by the on-set of the function f_v^1.*

This proposition takes place for any internal pole of a single output circuit C. The proposition gives additional possibilities for the circuit C minimization. For finding a minimum masking circuit realization we can use one of the heuristic algorithm BDD minimization for incompletely specified Boolean functions, for instance [13]. Let $f^*(X)$ be the completely specified function that realizes the incompletely specified function (presented by f_v^0 and f_v^1) and it has minimum BDD size. Use ROBDD of the function f^* to mask the fault of the gate with the output pole v.

The example of circuit C is given in Fig. 1. Extract functions f_v^0, f_v^1, f, f^* for pole v_1. Represent incompletely specified function with K-map in Fig. 2a. Here the on-set is represented with 1s, the off-set is represented with 0s and the don't care set with asterisks. Mark elements of the on-set for function f^* by circles. For the pole v_1 the function f^* coincides with f. The ROBDD of f has minimum number of nodes (Fig. 2b).

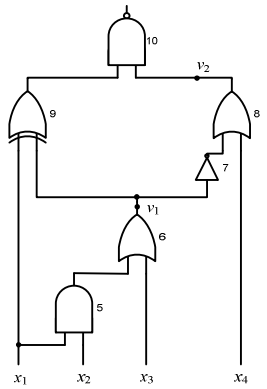
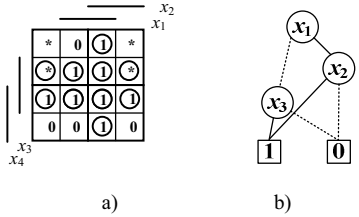


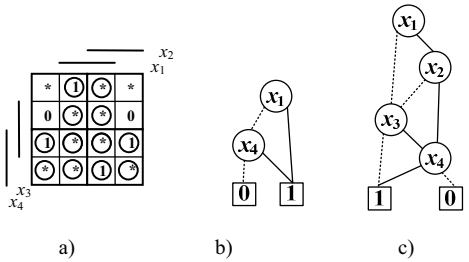
Figure 1. The combinational circuit C.



a) b)

Figure 2. a) K-maps of the functions for pole v_1; b) ROBDD for $f^* = f$.

Extract functions f_v^0, f_v^1, f, f^* for pole v_2. Represent the incompletely specified function with K-map in Fig. 3a. In this case the size of the ROBDD for the function f^* is less in comparison with the ROBDD for f (Fig. 3b-c).



a) b) c)

Figure 3. a) K-maps of the functions for pole v_2; b)ROBDD for f^*; c) ROBDD for f.

When replacing, we don't provide connections of the replacing sub-circuit with the gates of circuit C. The replacing sub-circuit is connected with the proper multiplexor input and the inputs of the circuit C.

Remind that two completely specified functions f_v^0, f_v^1 corresponds to the pole *v*.

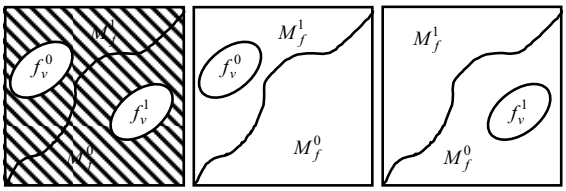
Let one of this function be equal to 0.

The function $f_v^1 = 0$. This situation is represented by the Venn diagram of Fig. 4b. This means that the function ε may be changed for the constant 1. As a result we may mask the considered gate fault by a connection of the proper MUX input with a power source.

The function $f_v^0 = 0$. This situation is represented by the Venn diagram of Fig. 4c. This means that the function γ may be changed for the constant 1. As a result we may mask the considered gate fault by a connection of the corresponding MUX input with a ground.

If $f_v^1 = 0$ and $f_v^0 = 0$ we have: ς(*X*, 1) = ς(*X*, 0) that is the variable *v* is unessential. In this case the fault gate does not change the function φ(*X*). There is no need to mask such fault.

In all above mentioned particular cases there is no need to construct a sub-circuit.



a) b) c)

Figure 4. a) Representation of the incompletely specified Boolean function for pole *v*; b) $f_v^1 = 0$; c) $f_v^0 = 0$.

III. MASKING FAULTS

Consider a multi output circuit *C*. If all paths of the branch point *v* are connected with the same output we have the situation considered in Section 2.

Let the paths of branch-point *v* be connected with the different outputs of circuit *C*. In this case we may obtain the ROBDD for the function *f* of the pole *v* and cover the ROBDD with CLBs to mask the fault gate with the output pole *v*. Take into consideration that in this paper we consider any logical fault of the gate with the output pole *v*.

Let a fault gate with the output pole *w* be not a branch point. We also admit any logical fault of the gate with the output pole *w*. Then we find the closest branch point *v* for the pole *w*. (There is only one branch point). Change the sub-circuit implementing the function *f* corresponding to the pole *v* for the proper masking sub-circuit in above mentioned way. As the sub-circuit with the output pole *w* is a part of the sub-circuit with the output pole *v*, then the fault gate with the output pole *w* will be masked. Note that there are no branch points between poles *v* and *w*. This means that the fault of the gate with output pole *w* does not affect any gates beyond the path connecting *v* and *w*. Masking the fault of the gate with output pole *w* at the expense of use of masking sub-circuit with output pole *v* gives the opportunity to apply MUXs only for the branch points of a combinational circuit.

Thus if a fault gate in circuit *C* is found, a developer decides either this fault may be masked by the CLB sub-circuit or circuit *C* must be redesigned. The decision depends on a complexity of the masking sub-circuit and its other characteristics.

Let us demonstrate the fault masking on the example. The partially programmable circuit that masks faults on pole v_1 is presented in Fig. 5.

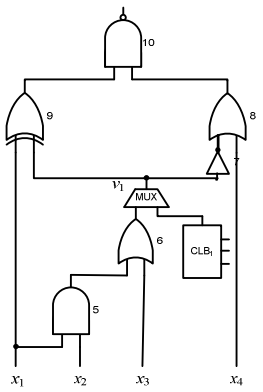


Figure 5. Partially programmable circuit.

IV. EXPERIMENTAL RESULTS

We have used MCNC Benchmark circuits for experiments. The experimental results are presented in Table 1. Here is: N_in – a number of inputs; N_out – a number of outputs: N_BP – a number of branch points; BDD_f_avg – average value of the size of the BDD for *f* for all internal lines; ROBDD_f*_avg – average value of the size of the ROBDD for *f** for all internal lines, *f**(*x*) is the one of the best implementation of the incompletely specified Boolean function, ROBDD for *f** was produced by using function Cudd_bddMinimize from CUDD package; Diff_max – maximal difference between size of ROBDD for *f* and size of ROBDD for *f** among all internal lines.

Note that during experiment it was found out that the functions corresponding to some poles of circuits may be changed for the constant 0(1) (for instance, in circuits too_large, lal and others). In this case there is no need to construct the proper sub- circuits.

Experimental results showed that for many circuits ROBDDs representing either function *f* or function *f** may be not large. An overhead for fault free circuits is the number of MUXs in branch-points. An overhead for masking sub-circuit is connected with the number of ROBDD internal nodes i.e. the number of 3-input LUTs.

V. CONCLUSION

The original approach to masking arbitrary gate faults of a combinational circuit based on inserting fault free sub-circuit through MUX is developed. The sub-circuit is derived by covering the proper ROBDD by CLBs. The

TABLE I. EXPERIMENTAL RESULTS

Circuit	N_in	N_out	N_BP	BDD_f_avg	BDD_f*_avg	Diff_max
b9	41	21	10	4.063	3.426	11
ttt2	24	21	6	5.891	4.749	20
count	35	16	14	9.258	9.258	0
example2	85	66	23	5.958	4.082	17
too_large	38	3	40	10.45	2.108	12
comparator	8	3	11	2.846	1.808	10
C17	5	2	2	2.5	2.375	1
cm82a	5	3	1	2.667	2.667	0
t	5	2	1	2	2	0
cm42a	4	10	3	2.333	2.333	0
cm138a	6	8	1	3	3	0
decod	5	16	2	2	2	0
cmb	16	4	1	12.4	5.35	20
cm163a	16	5	3	3.636	3.418	2
pcle	19	9	7	4.286	4.286	0
cm162a	14	5	5	4.357	3.4	6
x2	10	7	6	2.6	1	4
cu	14	11	2	6.083	3.197	17
pcler8	27	17	8	4.286	3.874	3
cc	21	20	1	4.692	4.692	0
sct	19	15	1	6.88	6.88	0
9symml	9	1	8	14	10.442	30
lal	26	19	6	6.923	6.923	0

ROBDD represents one of the realizations of the incompletely specified Boolean function. The incompletely specified Boolean function corresponds to a branch point which is either output of the fault gate or the closest branch point to the fault gate. When the fault of a gate is detected, the proper replacing sub-circuit from reserved CLBs is constructed. Its output is connected with the circuit through the closest multiplexer in the branch point. The sub-circuit masks either arbitrary fault of the gate which output is the branch point or arbitrary fault of the gate for which this branch-point is the closest one but its own output is not a branch point. If a combinational circuit considered is fault free its performance does not degrade in contrast to [7]. The circuit degrades only by linking-up the sub-circuit when a combinational circuit is fault.

REFERENCES

[1] International technology roadmap for semiconductors, 2007.

[2] V. K. R. Chiluvuri and I. Koren: “Layout-synthesis techniques for yield enhancement”, IEEE Transactions on Semiconductor Manufacturing, 8, 2, pp. 178–187 (1995).

[3] A. Nardi and A. L. Sangiovanni-Vincentelli: “Logic synthesis for manufacturability”, IEEE Design & Test of Computers, 21, 3, pp. 192–199 (2004).

[4] C. He and M. Jacome: “Defect-aware high-level synthesis targeted at reconfigurable nanofabrics”, IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, 26, 5, pp. 817–833 (2007).

[5] M. Abramovici, C. Stroud, and M. Emmert, “Using embedded FPGAs for SoC yield improvement,” in Design Automation Conf., 2002, pp. 713–724.

[6] A. Doumar and H. Ito, “Detecting, diagnosing, and tolerating faults in sram-based field programmable gate arrays: a survey,” IEEE Trans. on VLSI Systems, vol. 11, no. 3, pp. 386–405, 2003.

[7] S. Yamashita, H. Yoshida, and M. Fujita, “Increasing yield using partially-programmable circuits,” in Workshop on Synthesis And System Integration of Mixed Information technologies (SASIMI), 2010, pp. 237– 242.

[8] S. Yamashita, H. Sawada, A. Nagoya, “SPFD: A new method to express functional permissibilities,” IEEE Trans. On CAD, Vol. 19, N 8, pp. 840-849, 2000.

[9] A. Matrosova, S. Ostanin and I. Kirienko, “All Stuck-at Fault Test Patterns and Incompletely Specified Boolean Functions,” Proceeding of the 11th International Workshop on Boolean Problems. Freiberg, Germany, 2014, pp. 165-170.

[10] A.Yu. Matrosova, S.A. Ostanin, A.I. Bucharov, I.E. Kirienko, “Generating all test patterns for a given stuck-at fault of a logical circuit and its ROBDD implementation,” Tomsk State University, Journal of Control and Computer Science, №2 (27), 2014. (In Russian)

[11] A.Yu. Matrosova, “A fault-detection method for a synchronous device,” Automation and Remote Control, Vol. 38, No.12, pp.1849–1857, 1978.

[12] A. Matrosova, E. Loukovnikova, S. Ostanin, A. Zinchuck, "Test Generation for Single and Multiple Stuck-at Faults of a Combinational Circuit Designed by Covering Shared ROBDD with CLBs," IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT '07), Sept. 2007, pp. 206-214.

[13] T. Shiple, R. Hojati, A. Sangiovanni-Vincentelli, and R. K. Brayton, “Heuristic minimization of BDD’s using don’t cares,” in Proc. Design Automation Conf., 1994, pp. 225–231.