Simplification of the scheme of the self-tested detector (*m*, *n*)-code*

N. Butorina
Tomsk State University, Tomsk, Russia
nnatta07@mail.ru

Abstract

The article is devoted to the improvements of a self-tested checker (m, n) - codes, constructed within FPGA technology

1. Introduction

The increasing complexity of digital systems and critical application in which they are used demands their high reliability. Temporary faults (both transient and intermittent) and single event upset faults (SEU faults) become more and more probable. It is often necessary an error should be detected as soon as it is produced by the failure before it propagates through the system. This is achieved by using concurrent error detection (CED) techniques which allow to detect both permanent and temporary faults during normal operation.

CED techniques are based on implementing digital circuits as self-checking. These free fault circuits as a rule produce error detecting codes (valid codewords). A circuit fault changes valid codewords for invalid ones. The latter are just detected with a checker. The self-testing means that within the considered class of faults for each fault there exists a test vector among the set of all codewords for the checker.

Here we assume that V is a set of faults for self-testing checker which consists of all multiple stuck-at faults on input and output poles of configurable logic blocks (CLBs). It is supposed that only one CLB in the checker can be faulty, and multiple faults appear only on CLBs inputs. The faults on outputs of one CLB are the faults on inputs of other CLBs.

We assume that either a checker or a circuit can be faulty in a self-checking system but not both of them can.

The following demands for the self-testing checker are formulated:

- the checker has to give out a signal about fault if not codewords appear on circuit outputs (or checker inputs) in some moment of time t;
- a fault from the considered set of faults V can appear in the checker and it has to be detectable in working area of functioning of the checker.

A self-testing checker has two outputs:

a) Its outputs manifest wrong signals (00, 11) if either not codeword appears on self-checking circuit outputs or the checker is faulty.

b) Otherwise the checker outputs manifest right signals (01, 10).

We suppose that the checker is constructed by one- and two-outputs CLB in the frame of FPGA technology.

One output block realizes a function of 7 and less variables, two output block does functions of 6 and less variables.

Many synthesis techniques of self-testing checkers for m-out-of-n codes are presented [1-5]. In this paper the improvements of the self-tested checker (*m*, *n*) - codes constructed within CLBs are offered.

2. Derivation of Formula *A*

Special formula $A(X)$ is proposed in [4] for a compact description of all *m*-out-of-*n* codewords where $1 \le m \le n$. Denote the disjunction of conjunctions representing all the *q*-out-of-*p* codewords, $p < n,\ q \le p,\ X^r \subseteq X$, $X = \{x_1,\dots, x_n\}$, as $D^q_p(X^r)$. Further we consider a case $n = 2m$.

Divide set X into two subsets X^1, X^2, where

$X^1 = \{x_1,\dots, x_g\}$, $X^2 = \{x_{g+1},\dots, x_n\}$.

Theorem 1.

$$D^m_{2m}(X) = \sum_{i=0}^{m} D^i_g(X^1)D^{m-i}_s(X^2) \quad (1)$$

We call *k – basis of the decomposition*, D^i_g, D^{m-i}_s – *decomposition functions*. The initial set of variables X of the cardinality n is divided into two subsets X^1 and X^2: $|X^1| = g = \lceil n/2 \rceil$, $|X^2| = s = n - g$. If $g > k$ and $s > k$, then formula (1) is used again for every decomposition function $D^i_g, D^{m-i}_s, i = \overline{0,m}$, etc. As a result we have formula *A* specifying all the *m*-out-of-*n* codewords.

Note that the number of all the *m*-out-of-*n* codewords is equal to C^m_n, that is the number of combinations of *n* things *m* at a time.

*The reported study was partially supported by Russian Science Foundation, research project № 14-19-00218.
Participation in EWDTS2015 is partially supported within the "Program of increase of the international competitiveness of Tomsk State University for 2013-2020".

978-1-4673-7776-8/15/$31.00 ©2015 IEEE

For example we obtain formula (1) for D_{14}^7, $k = 7$. In this case the set X is divided into the subsets $X^1=\{x_1,x_2,x_3,x_4,x_5,x_6,x_7\}$, $X^2=\{x_8,x_9,x_{10},x_{11},x_{12},x_{13},x_{14}\}$.

The decomposition for D_{14}^7 is represented in the following form

$$D_{14}^7 = D_7^0(X^1)D_7^7(X^2)\vee D_7^1(X^1)D_7^6(X^2)\vee$$
$$\vee D_7^2(X^1)D_7^5(X^2)\vee D_7^3(X^1)D_7^4(X^2)\vee D_7^4(X^1)D_7^3(X^2)\vee$$
$$\vee D_7^5(X^1)D_7^2(X^2)\vee D_7^6(X^1)D_7^1(X^2)\vee D_7^7(X^1)D_7^0(X^2).$$

Hence

$$X^{11}=\{x_1,x_2,x_3,x_4\}, X^{12}=\{x_5,x_6,x_7\},$$
$$X^{21}=\{x_8,x_9,x_{10},x_{11}\}, X^{22}=\{x_{12},x_{13},x_{14}\}.$$

Consequently, for the following functions

$$D_7^0(X^1)=D_4^0(X^{11})D_3^0(X^{11}),$$
$$D_7^1(X^1)=D_4^0(X^{11})D_3^1(X^{11})\vee D_4^1(X^{11})D_3^0(X^{11}),$$
$$D_7^2(X^1)=D_4^0(X^{11})D_3^2(X^{11})\vee D_4^1(X^{11})D_3^1(X^{11})\vee$$
$$D_4^2(X^{11})D_3^0(X^{11}),$$
$$D_7^3(X^1)=D_4^0(X^{11})D_3^3(X^{11})\vee D_4^1(X^{11})D_3^2(X^{11})\vee$$
$$D_4^2(X^{11})D_3^1(X^{11})\vee D_4^3(X^{11})D_3^0(X^{11}),$$
$$D_7^4(X^1)=D_4^1(X^{11})D_3^3(X^{11})\vee D_4^2(X^{11})D_3^2(X^{11})\vee$$
$$D_4^3(X^{11})D_3^1(X^{11})\vee D_4^4(X^{11})D_3^0(X^{11}),$$
$$D_7^5(X^1)=D_4^2(X^{11})D_3^3(X^{11})\vee D_4^3(X^{11})D_3^2(X^{11})\vee$$
$$D_4^4(X^{11})D_3^1(X^{11}),$$
$$D_7^6(X^1)=D_4^3(X^{11})D_3^3(X^{11})\vee D_4^4(X^{11})D_3^2(X^{11}),$$
$$D_7^7(X^1)=D_4^4(X^{11})D_3^3(X^{11}),$$

$$D_7^0(X^2)=D_4^0(X^{21})D_3^0(X^{21}),$$
$$D_7^1(X^2)=D_4^0(X^{21})D_3^1(X^{22})\vee D_4^1(X^{21})D_3^0(X^{22}),$$
$$D_7^2(X^2)=D_4^0(X^{21})D_3^2(X^{22})\vee D_4^1(X^{21})D_3^1(X^{22})\vee$$
$$D_4^2(X^{21})D_3^0(X^{22}),$$
$$D_7^3(X^2)=D_4^0(X^{21})D_3^3(X^{22})\vee D_4^1(X^{21})D_3^2(X^{22})\vee$$
$$D_4^2(X^{21})D_3^1(X^{22})\vee D_4^3(X^{21})D_3^0(X^{22}),$$
$$D_7^4(X^2)=D_4^1(X^{21})D_3^3(X^{22})\vee D_4^2(X^{21})D_3^2(X^{22})\vee$$
$$D_4^3(X^{21})D_3^1(X^{22})\vee D_4^4(X^{21})D_3^0(X^{22}),$$
$$D_7^5(X^2)=D_4^2(X^{21})D_3^3(X^{22})\vee D_4^3(X^{21})D_3^2(X^{22})\vee$$
$$D_4^4(X^{21})D_3^1(X^{22}),$$
$$D_7^6(X^2)=D_4^3(X^{21})D_3^3(X^{22})\vee D_4^4(X^{21})D_3^2(X^{22}),$$
$$D_7^7(X^2)=D_4^4(X^{21})D_3^3(X^{22}).$$

we get

$$A=D_{14}^7=D_4^0(X^{11})D_3^0(X^{11})D_4^4(X^{21})D_3^3(X^{22})\vee$$
$$(D_4^0(X^{11})D_3^1(X^{11})\vee D_4^1(X^{11})D_3^0(X^{11}))$$
$$(D_4^3(X^{21})D_3^3(X^{22})\vee D_4^4(X^{21})D_3^2(X^{22}))\vee$$
$$\vee(D_4^0(X^{11})D_3^2(X^{11})\vee D_4^1(X^{11})D_3^1(X^{11})\vee D_4^2(X^{11})D_3^0(X^{11}))$$
$$(D_4^2(X^{21})D_3^3(X^{22})\vee D_4^3(X^{21})D_3^2(X^{22})\vee D_4^4(X^{21})D_3^1(X^{22}))$$
$$\vee(D_4^0(X^{11})D_3^3(X^{11})\vee D_4^1(X^{11})D_3^2(X^{11})\vee D_4^2(X^{11})D_3^1(X^{11})\vee$$
$$D_4^3(X^{11})D_3^0(X^{11}))$$
$$(D_4^1(X^{21})D_3^3(X^{22})\vee D_4^2(X^{21})D_3^2(X^{22})\vee D_4^3(X^{21})D_3^1(X^{22})\vee$$
$$D_4^4(X^{21})D_3^0(X^{22}))$$
$$\vee(D_4^1(X^{11})D_3^3(X^{11})\vee D_4^2(X^{11})D_3^2(X^{11})\vee D_4^3(X^{11})D_3^1(X^{11})\vee$$
$$D_4^4(X^{11})D_3^0(X^{11}))$$
$$(D_4^0(X^{21})D_3^3(X^{22})\vee D_4^1(X^{21})D_3^2(X^{22})\vee D_4^2(X^{21})D_3^1(X^{22})\vee$$
$$D_4^3(X^{21})D_3^0(X^{22}))\vee$$
$$\vee(D_4^2(X^{11})D_3^3(X^{11})\vee D_4^3(X^{11})D_3^2(X^{11})\vee D_4^4(X^{11})D_3^1(X^{11}))$$
$$(D_4^0(X^{21})D_3^2(X^{22})\vee D_4^1(X^{21})D_3^1(X^{22})\vee D_4^2(X^{21})D_3^0(X^{22}))$$
$$\vee(D_4^3(X^{11})D_3^3(X^{11})\vee D_4^4(X^{11})D_3^2(X^{11}))(D_4^0(X^{21})D_3^1(X^{22})\vee$$
$$D_4^1(X^{21})D_3^0(X^{22}))$$
$$\vee D_4^4(X^{11})D_3^3(X^{11})D_4^0(X^{21})D_3^0(X^{21}).$$

3. A self-testing property for the checker of m-out-of-n code

Special types of functions keeping self-testing properties are considered in [6].

Definition 1. Call function D_p^q, 0 <q <p as the function of type 1.

Table 1. Function D_4^2 – function of type 1.

x_1	x_2	x_3	x_4
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	0
1	0	1	0
1	1	0	0

Theorem 2. Let CLB realize the function of type 1. Then for a multiple fault on inputs CLB either the test pattern representing by the one of conjunctions from D_p^q exists, or a multiple fault is detected on output CLB as constant 1.

Definition 2. The function of type 2 is called the function which has to meet the following conditions: there is only one single component at each column and the number of single components is identical at every line.

The example of the function of type 2 is given in table 2.

Table 2. Function of type 2.

x_1	x_2	x_3	x_4
1	1	0	0
0	0	1	1

Theorem 3. Let the CLB realize the function of type 2. Then for a multiple fault on inputs CLB either there exists the test from the area of a single values of this function, or the multiple fault detected on CLB output as constant 1.

Theorem 4. Let on one of the outputs of two output block function of type 1 or function of type 2 realize. Let on the second output any function realize. Then for a multiple fault on input CLB of this block either there exists the test from area of single values of function of type 1 (function of type 2) or the multiple fault is detected on output CLB as constant 1.

Theorem 5. Let the subcircuit realize mutually inverse sets. Then for a multiple fault on input CLB either there exists the test from area of single values of function , or the multiple fault is detected on output CLB as constant 1.

Consider a subcircuit 1 (figure 1). The lower level of this subcircuit consists of one - and the two-output CLB realizing functions D_p^q, D_{s-p}^t, 0 ≤ q ≤ p, 0 ≤ t ≤ s – p (function of type 1), and outputs of these CLBs are inputs to the CLB, realizing the function of type 2. At the subcircuit there are at least two CLBs the sets of variables of which coincide. If in subcircuit 1 there is CLB realizing functions D_p^0 (D_p^p), then in a subcircuit there are CLB realizing functions D_p^1 (D_p^{p-1}). Set of single values of a system of Boolean functions of the lower level of subcircuit 1 (the system consisting of the function of type 1) we call set M_1.

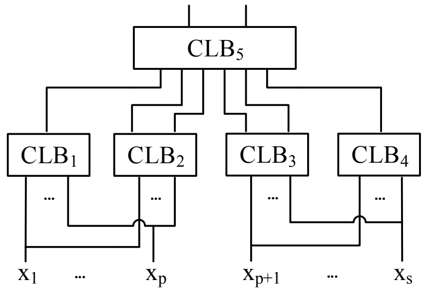

Figure 1. Subcircuit 1.

Theorem 6. For a multiple fault on input CLB (subcircuit 1) either there exists the test pattern representing by one of the conjunctions from set M_1, or the multiple fault is detected on output CLB as constant 1.

Proof.

1) Let CLB realize function of type 1, that is function D_p^q, $0 < q < p$. Then this case has been considered in theorems 2 and 4.

2) Let on one of the outputs of two-output CLB function be realized D_p^0. Then on the other output function is realized D_p^q, $0 < q \le p$. And according to theorem 4 and 5 there is a test presented by one of conjunctions from set M_1, or the multiple fault is detected on output CLB as constant 1.

3) Let the CLB realize function D_p^0. As k is a number of inputs to CLB, we consider that CLB realizes function D_k^0, that is $p = k$. The area of single values of this function consists of one vector 00 … 0 (the number 0 is equal to k). Let a fault be represented by ternary vector β and vector β has r of the defined component, $r \le k$. Consider all possible cases.

1. Let $r = k$, that is all components β are defined. Then

a) if vector β is a vector from area of single values of the function, the multiple fault is detected on output CLB as constant 1;

b) if vector β is not a vector from area of single values of the function, the multiple fault is detected on output CLB by Boolean vector 000000;

2. Let $r < k$, s is a number of single components vector β. Then the following options are possible:

a) if $0 < s \le k$, that is there is at least one single component, multiple fault is detected on output CLB by Boolean vector 000000;

b) if $s = 0$, then this fault is not detected by Boolean vector 000000. But in subcircuit 1 there is CLB_2 which realizes function D_k^1 the set of variables of which coincides with a set of variables of function D_k^0. Let α

be the code word CLB_2 in which any zero component of vector β coincides with a single component of vector α (such vector obligatory exists). As a consequence of the fault which arises in CLB_1, codeword CLB_2 becomes codeword CLB_1. But outputs of CLB_1 and CLB_2 are inputs to CLB_5, realizing the function of type 2. Therefore, on output CLB_5 instead of 1 there is 0, so, fault of CLB_1 is detected.

4) Let CLB realize function D_p^p. In this case reasonings are similar to reasonings in point 3. Fault of CLB_1 is detected on the output of subcircuit 1 by the code word CLB_2 realizing function D_p^{p-1}.

The theorem is proved.

Earlier [7] for providing a self-testing of CLB realizing functions D_p^0 (D_p^p) the special subcircuits are offered. These subcircuits demand additional splitting of a set of codewords of this CLB into subsets. Theorem 6 allows to simplify the scheme of the self-tested checker and to reduce the quantity of CLB in a checker. For example, there is an old scheme of the checker D_{14}^7 in figure 2 and a new scheme of the same checker in figure 3.

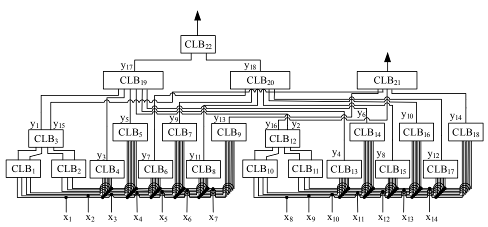

Fig. 2. Checker for (7,14) code words (old scheme).

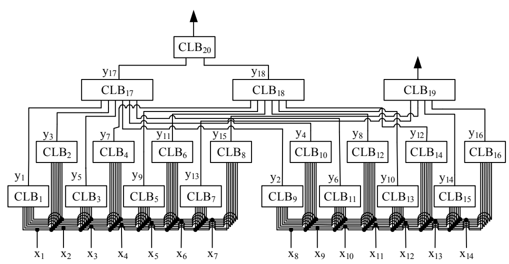

Fig. 3. Checker for (7,14) code words (new scheme).

3. Conclusion

The improvements of the self-tested checker (m, n) – codes are offered. They allow to simplify the scheme of the checker and to reduce the quantity of CLBs.

6. References

[1] Parag K. Lala. "Self-Checking and Fault-Tolerant Digital Design" Morgan Kaufmann Pub (June 26, 2000).

[2] D.A. Anderson, G. Metze. "Design of totally self-checking check circuits for m-out-of-n codes". IEEE Trans. Computers, C-22 (March 1973), pp. 263-269.

[3] M.A. Marouf, A.D. Friedman. "Efficient design of sel-checking checker for any m-out-of-n code". IEEE Trans. Computers, C-27 (June 1978), pp. 482-490.

[4] A. Matrosova, V. Ostrovsky, K. Nikitin. "Designing FPGA based Self-testing Checkers for m-out of n Codes", y IEEE On-line Testing Symposium. Greece, Kos, July 2003, pp. 49-53.

[5] A. Matrosova, K. Nikitin. "Designing Self-testing Checker with CLBs". ISSN, 1561-7793, Tomsk State University, N6, September 2003, pp. 124-136 (in Russian).

[6] Butorina N. Obosnovanie samotestiruemosti detectora (m, n)-coda // ICAM'2012. Tomsk. NTL, 2012. P. 55-55.

[7] Butorina N. Obosnovanie samotestiruemosti detectora (m, n)-coda // Izvestiya VUZov. Phisic. 2013. T. 56, № 9(2). P. 156-159.