Efficient Highly Testable Borden Code Checkers

D. Nikolos & X. Kavousianos
Department of Computer Engineering and Informatics
University of Patras, 26500, Patras, Greece &
Computer Technology Institute, Kolokotroni 3, Patra, Greece
e-mail : nikolosd@cti.gr, kabousia@ceid.upatras.gr

Abstract

Borden codes are optimal t-unidirectional error detecting (t-UED) codes. The main contribution of this paper is to offer for first time in the open literature Borden code checkers which are Totally Self Checking (TSC) under a realistic fault model including node stuck-at, transistor stuck-on, resistive bridgings, breaks, and several transistor stuck-open faults. Single output as well as double output checkers are given. No other single output TSC Borden code checker is already known even under the inadequate single stuck-at fault model. Apart from the above the proposed checkers are more efficient, with respect to area and speed, than the already known TSC Borden code checkers.

I. Introduction.

A circuit consisting of a functional circuit, whose output words belong to a certain code, and a checker that monitors the output of the functional circuit and indicates if it is a code or a non-code word is called Self Checking Circuit (SCC) [1]. These circuits can provide concurrent error detection and thus can detect transient, intermittent as well as permanent faults. Since transient faults have become increasingly dominant in VLSI circuits [2-4], providing protection against them has become very important. The reliability of a SCC depends on the ability of its checker to behave correctly despite the possible occurrence of internal faults. It has been shown that this is achieved when the checker satisfies either the Totally Self Checking (TSC) [5] or the Strongly Code Disjoint (SCD) [6] property. In this paper we will take into account the TSC property. A circuit is a TSC checker if it is self-testing, fault-secure and code disjoint [5, 7].

It is well known that the most common errors in VLSI circuits are unidirectional in nature [8-11]. Unidirectional errors have also been observed in compact laser disks [11, 12]. Berger [13] and m-out-of-n [14] codes are well known codes that detect unidirectional errors of any multiplicity. However, in many cases, it is reasonable to assume that the number t of unidirectional errors is limited. The value of t, of course, will depend on the bit organization, layout, etc. Borden codes [15] are optimal, with respect to redundancy, t-unidirectional error detecting (t-UED) codes. A t-UED Borden code with length n consists of the code words of all the m-out-of-n codes with m congruent to $\lfloor n/2 \rfloor$ mod (t+1).

TSC Borden code checkers were proposed recently in [16-21]. Their common characteristic is that they are TSC only with respect to single stuck-at faults. However, the conventional stuck-at fault model has been found to be inadequate for CMOS circuits [22]. CMOS is the current dominant technology for manufacturing VLSI circuits, thus new TSC checker designs are required that will take into account a more realistic fault model including apart from stuck-at, transistor stuck-open, transistor stuck-on, resistive bridging and break faults.

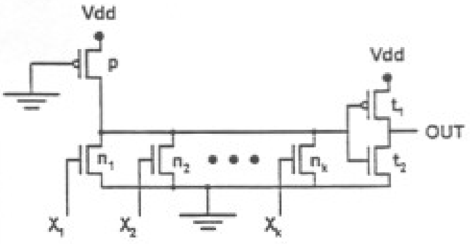

Figure 1. m-ones threshold circuit

In this paper a new method for designing TSC checkers for Borden codes is proposed. The checkers designed according to this method are TSC with respect to stuck-at, transistor stuck-on, resistive bridging faults, breaks and several transistor stuck-open faults. The proposed checkers are also more efficient, with respect to area and speed, than the already known TSC Borden code checkers.

There are cases that a single output TSC checker with its output two-rail encoded in time may have some advantages over the double output checker [23, 33]. No single output TSC checker for Borden codes is up today known from the open literature. To this end, apart from double output we also present single output TSC checkers for Borden codes.

Throughout this paper the following notations are used :

- $W^1(X)$ denotes the number of ones of the vector X.
- V_{OHMIN} (V_{OLMAX}) is the minimum HIGH (maximum LOW) voltage at the output of a circuit.
- V_{tn} (V_{tp}) is the threshold voltage of n (p) - transistor.
- β_n, (β_p) is the gain factor of n (p) - transistors.
- KP_n (KP_p) is the Spice parameter for $\mu_n C_{ox}$ ($\mu_p C_{ox}$).
- W_{ni}/L_{ni} (W_{pi}/L_{pi}) is the ratio of nmos (pmos) transistor i.

II. Design Method

A. *Threshold circuits*

Definition 1. A circuit M with k inputs, $X_1, X_2, \ldots, X_k$ and one output, OUT, is called m-ones threshold circuit, if it

ok

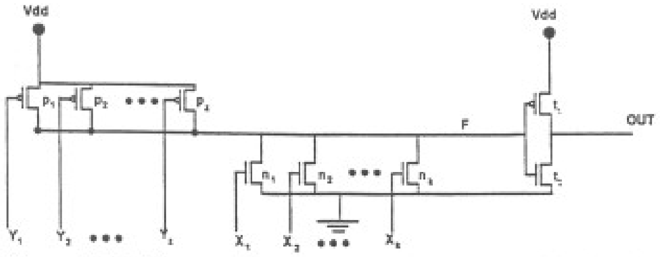

Figure 2 $(V_1, V_2, \dots, V_z)$ aggregate-ones threshold circuit.

operates as follows: when $W^1(X_1, X_2, \dots, X_k) \geq m$ then OUT is High else OUT is Low.

The circuit of figure 1 is an m-ones threshold circuit if the following relations are satisfied [24] :

a. $W_{n_1}/L_{n_1} = W_{n_2}/L_{n_2} = \cdots = W_{n_k}/L_{n_k} = (W/L)_n$

b. $(m-1) \cdot Q_1 \cdot (W/L)_n \leq W_p/L_p \leq m \cdot Q_2 \cdot (W/L)_n$ (1),

$$Q_1 = KP_n/KP_p \cdot \left(2(V_{dd} - V_{tn})V_{OHMIN} - V^2_{OHMIN}\right) / \left(V_{dd} + V_{tp}\right)^2$$

$$Q_2 = KP_n/KP_p \cdot \left(2(V_{dd} - V_{tn})V_{OLMAX} - V^2_{OLMAXN}\right) / \left(V_{dd} + V_{tp}\right)^2$$

The ones-weight, $TW^1(M)$, of an m-ones threshold circuit is by definition equal to m.

Definition 2. A circuit M with k+z inputs $X_1, X_2, \dots, X_k$ and $Y_1, Y_2, \dots, Y_z$ and one output, OUT is called an $(V_1, V_2, \dots, V_z)$ aggregate-ones threshold circuit, with $V_1, V_2, \dots, V_z \in N^*$ if for each vector $Y_1, \dots, Y_z$ the circuit operates as follows: when $W^1(X_1, \dots, X_k) \geq \overline{Y}_1V_1 + \overline{Y}_2V_2 + \cdots + \overline{Y}_zV_z$ then OUT is High else it is Low.

The sum $\overline{Y}_1V_1 + \overline{Y}_2V_2 + \cdots + \overline{Y}_zV_z$ is called the aggregate-ones-weight, $AW^1_M(V, Y)$, of the circuit for the vector $Y_1, Y_2, \dots, Y_z$.

It is easily proven that the circuit of Figure 2 is an $(V_1, V_2, \dots, V_z)$ aggregate- ones threshold circuit, if the following relations are satisfied:

a. $W_{n_1}/L_{n_1} = W_{n_2}/L_{n_2} = \cdots = W_{n_k}/L_{n_k} = (W/L)_n$

b. $\left(V_i - 1/z\right) \cdot (W/L)_n Q_1 \leq W_{p_i}/L_{p_i} \leq V_i (W/L)_n Q_2$, $i \in [1, z]$ (2)

Figure 3 shows the symbols for the m-ones threshold circuit (a) and $(V_1, V_2, \dots, V_z)$ aggregate-ones threshold circuit (b).

B. ***TSC Checkers for the Borden codes.***

Let C be a t-UED Borden code with length k. Depending on the length k of the code and the value of t, the code word 0-out-of-k and/or the code word k-out-of-k belong to Borden code. As in the checkers given in [16] we do not include the code words 0-out-of-k and k-out-of-k in the code space. The only price we pay is that two less code words are available for encoding purposes, the unidirectional error capability of the code remains the same.

Consider the circuit of figure 4, where $m = \lfloor k/2 \rfloor \bmod (t+1)$, $m+f(t+1) \leq k$ and $m+(f+1)(t+1)>k$. Based on definition 2,

$$AW^1_A(V,Y) = (t+1)\overline{O}_{f-1} + \cdots + (t+1)\overline{O}_0 + m + clk \qquad (3)$$

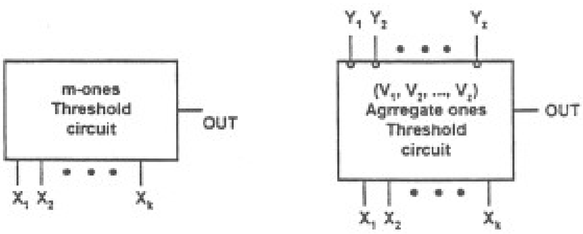

Figure 3. Symbols for: (a) m-ones threshold circuit; (b) $(V_1, V_2, \dots, V_z)$ aggregate-ones threshold circuit.

where $Y=(O_{f-1}, \dots, O_0, 0, clk)$, and $V=((t+1), \dots, (t+1), m, 1)$. Consider that the input clk is driven by a signal with frequency equal to the frequency of the system clock and $I = (I_1, \dots, I_k)$ is the received word. In the sequel we will prove that the circuit of fig. 4 is a code disjoint circuit with respect to the Borden code, that is, it is a Borden code checker.

a. Consider that the circuit of figure 4 receives as input a code word of C. Then according to the definition of Borden codes $W(I) = m+r(t+1)$, with $0 \leq r \leq f$. Then according to the function of the m-ones threshold circuit (Definition 1) we have $O_i = 0$ for $i=0, 1, \dots, r-1$ and $O_j = 1$ for $j=r, r+1, \dots, f-1$. Then from relation (3) we get $AW^1_A(V, Y) = r(t+1)+m+clk$ therefore for clk = 0 we have $W(I) = AW^1_A(V, Y)$ and OUT = 1, while for clk = 1 we have $W(I) < AW^1_A(V,Y) = m+r(t+1)+1$, hence OUT = 0. Therefore, when the circuit of figure 4 receives code words of the Borden code then its output is two-rail encoded in time, specifically, during a period of the signal clk its output takes the values (1,0).

b. Consider that the circuit of figure 4 receives a word I that is not a code word of C. We have two cases.

b.1. $W(I) = m+r(t+1)+a$, where $0 \leq r \leq f$ and $0 < a \leq t$. Then according to the function of the m-ones threshold circuit we have $O_i = 0$ for $i \in [0, r]$ and $O_j = 1$ for $j \in [r+1, f-1]$ and from relation (3) we get $AW^1_A(V, Y) = m+(r+1)(t+1)+clk$. Since $W(I) = m+r(t+1)+a$, with $0<a\leq t$, we have $W(I) < AW^1_A(V,Y)$ for either clk=0 or clk=1, therefore OUT = (0, 0).

b.2. $W(I) = a$, with $0 \leq a < m$. Then $O_i = 1$ for $i \in [0, f-1]$, thus from (3) we get $AW^1_A(V,Y) = m+clk$. So either for clk=0 or 1 we have $W(I)=a<m\leq AW^1_A(V,Y)$ hence we get OUT=(0,0).

From the above it is obvious that when the input vector I is a code word of the Borden code and the circuit is fault free then during a period of the signal clk the output OUT gets the values (1, 0). When the input vector is not a code word, then during a period the output OUT gets the values (0, 0). In other words the circuit of figure 4 is a single output Borden code checker. As in the case of the single output comparator given in [23] the output of the checker can be simply checked using a flip flop. The flip flop is triggered by a clock signal identical to the system clock, but delayed with respect to system clock, by a suitably chosen time interval (taking into account the checker input/output delay and the

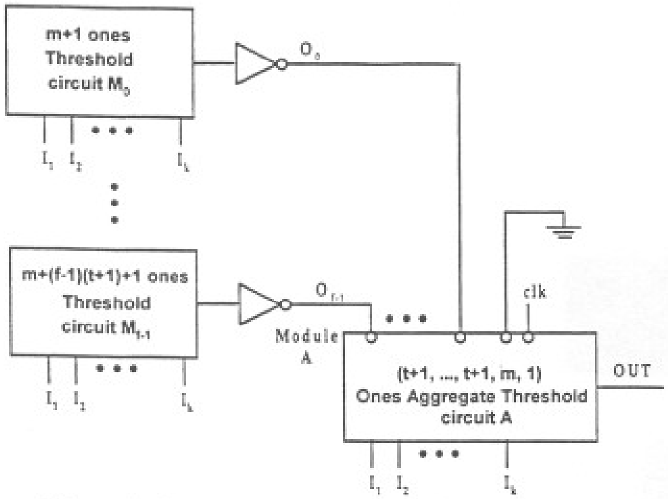

Figure 4. Single Output Borden code checker.

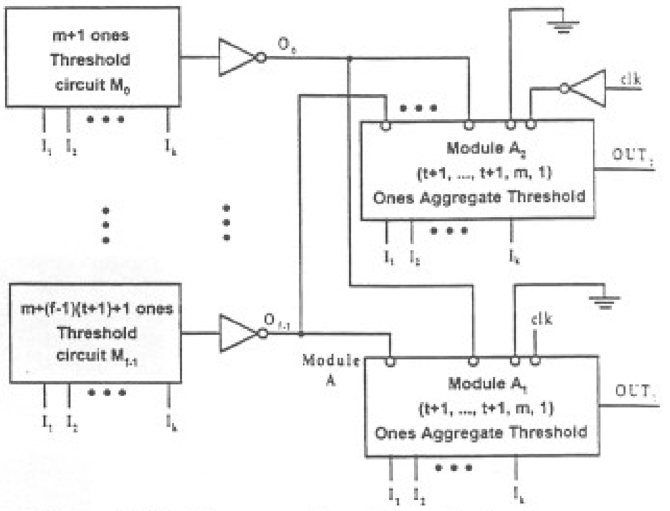

Figure 5. Double output Borden code checker.

flip flop setup time). The output of the checker is sampled on both the triggering signal rising and falling edges (as the flip flop presented in [25]).

From the above it is easy to see that the checker input/output delay, t_d, plus the flip flop setup time t_s, must be smaller than the half of the period of the system clock. This implies that the single-output TSC Borden code checker can be used only in systems with period greater than $2\cdot(t_d+t_s)$ (the same comment concerns the single output comparators given in [23]). However as we will see the delay of the proposed single output checkers is very small, thus they can be used in most applications.

Figure 5 presents the double output checker for the Borden code. In this checker, modules A_1, A_2 are identical to module A of Figure 4. The input clk is driven by a clock signal with the half frequency of feeding inputs to the checker. The feeding frequency is usually equal to the frequency of the system clock, therefore the signal driving input clk can be easily obtained from the system clock using a T flip flop. So, when the checker receives a code word of the Borden code , ($Output_1$, $Output_2$) = (0, 1) for clk=1 and ($Output_1$, $Output_2$) = (1, 0) for clk=0. When the received word is not a code word ($Output_1$, $Output_2$) is either (0, 0) or (1, 1), thus it is not two-rail encoded.

The manufacturability of the proposed checkers depends on the manufacturability of the m-ones and aggregate-ones threshold circuits. A problem of these ratioed circuit is that their correct operation depends on the conductance values of nmos and pmos transistors as well as the other circuit parameter's values. It is well known that fluctuations in integrated circuit manufacturing processes cause deviations on the actual values of the parameters from their nominal values. Designing the m-ones and aggregate-ones threshold circuits we choose the values of W_p, and L_p so that the value of W_p/L_p to be in the middle of the ranges given by relations (1) and (2). Then due to statistical variations of the device characteristics the range can be shortened or shifted to the left or to the right but the value of W_p/L_p will remain within the range, therefore the manufactured IC will operate correctly. As the value of f become greater (we remind that in figures 4 and 5 the value of f is such that $m+f(t+1) \leq k < m+(f+1)(t+1)$) the range defined by relation (1) and (2) become shorter and the yield of the manufacturing process will become smaller. With the improvement of a manufacturing process the circuit parameters deviation becomes smaller and the m-ones threshold circuits as well as aggregate-ones threshold circuits for larger values of f can be constructed. However, given the quality of a manufacturing process there exist a maximum value of f for which Borden code checkers can be constructed following our method.

III. Testability Analysis.

All the inverters used in these circuits are designed with n-dominate logic.

A. Single output checker.

We use the notation $OUT_{0,1}$=Q, R which means that during a period of the signal clk, OUT is equal to Q for clk=0 and OUT is equal to R for clk=1.

1. Stuck-on fault on the pmos transistor p or the pmos transistor t_1 of the circuit M_i, $0 \leq i \leq f-1$, or the pmos transistor of the inverter with output O_i, or on the transistor p_m of circuit A. Any one of these faults is not detected, but after its occurrence the checker remains code disjoint. Furthermore if this fault is followed by one of the other considered faults, the resulting fault is detectable.

2. Stuck-open fault on transistor p or transistor t_2 of circuit M_i, $0 \leq i \leq f-1$, or stuck-open fault on the pmos transistor of the inverter with output O_i. These faults are detected when the checker receives two successive code words, I_1, I_2, with $W(I_1) = m+f(t+1)$, $W(I_2)=m$. Then for both code words $O_i=0$ and the output of the checker for the second code word will be $OUT_{0,1}$ =(0, 0).

3. Stuck-open fault on transistor t_1 of circuit M_i, $0 \leq i \leq f-1$, or stuck-open fault on the nmos transistor of the inverter with output O_i. These faults are detected when the checker

Table 1. Resistive Bridging Faults.

Bridging Fault	Maximum Resistance	Bridging Fault	Maximum Resistance
I_i, I_{i+1}	5.2 KΩ	I_i, F	6 KΩ
O_0, O_1	600 Ω	F, Vdd	6 KΩ
I_i, Gnd	5.6 KΩ	F, Gnd	6 KΩ
I_i, F_0	1,2 KΩ	clk, F	6 KΩ
F_0, Gnd	1,65 KΩ	clk, Vdd	6 KΩ
F_0, Vdd	750 Ω	O_0, F	6 KΩ
I_i, F_1	2,3 KΩ	O_0, Vdd	6 KΩ
F_1, Gnd	2,75 KΩ	O_1, F	6 KΩ
F_1, Vdd	750 Ω	O_1, Vdd	6 KΩ

receives two successive code words, I_1, I_2, with $W(I_1)=m$ and $W(I_2)=m+f(t+1)$. Then for both inputs $O_i=1$ and the output of the checker for the second code word will be $OUT_{0,1}=(1, 1)$.

4. Stuck-open fault on transistor n_i , $i\in[1, k]$, of M_r, with $0\leq r\leq f-1$. These transistor stuck-open faults are not detected during the fault free operation of the checker and might make the checker fail to reveal the presence of a non code word at the inputs of the checker. However it should be reminded that transistor stuck-open faults have been verified to be less likely to occur than the other kinds of faults considered in this work [22, 26, 27]. Moreover, to reduce further on the likelihood of stuck-open faults, the checker layout could be suitably designed [28-30]. In addition, these faults could be off-line detected by applying non Borden code words to the inputs of the checker as will be shown in the next few lines. Consider a non code word I with $I_i = 1$ and $W(I) = m+r(t+1)+1$, $0\leq r\leq f-1$. When the checker receives the input I and is fault free then $O_j = 0$ for j=0, 1, …, r, and $O_s= 1$ for s=r+1, r+2, …, f-1. Hence $AW_A(V, Y) = m+(r+1)(t+1)+clk$ and $AW_A(V, Y) > W(I)$, therefore, $OUT_{0,1} =(0, 0)$. When a stuck-open fault has occurred on transistor n_i , $i\in[1, k]$, of the circuit M_r, $0\leq r\leq f-1$, then $O_j = 0$ for j=0, 1, …, r-1, and $O_s= 1$ for s=r, r+1, …,f-1. Hence $AW_A(V, Y) = m+r(t+1)+clk$ which implies that $W(I) \geq AW_A(V, Y)$, therefore, $OUT_{0,1} =(1, 1)$. The conclusion is that applying to the checker a non code word I with $I_i = 1$, $W(I) = m+r(t+1)+1$ and $0\leq r\leq f-1$, if a stuck-open fault has occurred on transistor n_i , $i\in[1, k]$, then $OUT_{0,1} =(1, 1)$ else $OUT_{0,1} =(0, 0)$. To detect the stuck -open fault on transistor n_i of M_r for i=1, 2, …, k we have to apply $\lceil k/(m+r(t+1)+1)\rceil$ words, with $W(I) = m+r(t+1)+1$, such that each input to take the value one. Therefore to detect a stuck-open fault on transistor n_i of M_r for $i\in[1, k]$ and $r\in[0, f-1]$ we need $\sum_{r=0}^{f-1}\lceil k/(m+r(t+1)+1)\rceil$ test vectors. For example for the 3-UED Bose code with k=12 only 6 test vectors are required.

All the other stuck-at, transistor stuck-open and transistor stuck-on faults are detected by a code word.

Resistive bridging faults (RBFs) between two transistor terminals or between two inputs have been considered. All RBFs with connecting resistance $R \in [0,R_{max}]$ are detected, where R_{max} depends on the sizing of the transistors and the information length. We are interested for R_{max}=6KΩ [27]. For the 8 bit Borden code checker of figure 4 and an implementation in λ=1μm technology with transistor aspect ratios $(W/L)n_i=4/1$, $(W/L)_p=20/1$ (module M_0), $(W/L)_p=50/1$ (M_1), $(W/L)_m=9/1$ - $(W/L)_{clk}=9/1$ - $(W/L)_0=(W/L)_1=30/1$ (A), the value of R_{max} is given in Table 1. During the simulation the inputs of the checker are driven by standard cell inverters with $(W/L)_p=12/1$ and $(W/L)_n=6/1$.

All breaks are detectable except those at the gate, drain or source of the transistor n_i of circuit M_r, for $r\in[0, f-1]$. These faults are equivalent to the stuck-open fault on transistor n_i of M_r, we have already been referred to.

We can easily see that applying to the checker :

a. $\lceil k/m\rceil$ code words of the form (0…001010…100…0), with weight equal to m, so that each input to take the value one (since $m \leq \lfloor k/2\rfloor$ each input takes also the value zero),

b. $\lceil k / (k-(m+r(t+1)))\rceil$ code words, with weight m+r(t+1), for r=1, 2, …, f, so that each input of the checker to take the value zero for each value of r,

c. a pair of code words I_1, I_2 with $W(I_1)=m+f(t+1)$ $W(I_2)=m$

d. a pair of code words I_1, I_2, with $W(I_1)=m$ $W(I_2)=m+f(t+1)$

all the above mentioned detectable node stuck-at, transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks are detected. That is, the test set consists of :

$\lceil k/m\rceil + \sum_{r=1}^{f}\lceil k/(k-(m+r(t+1)))\rceil$code words + two pairs of code words. For example for the 3-UED Borden code with k=12 the test set consists of 14 code words plus two pairs of code words, that is, the test set is very small.

B. Double output checker.

The testability analysis of the double output checker of figure 5 is similar to that of figure 4 except of the undetectable stuck-at 0 or 1 fault on line clk. The problem can be overcame in two ways. The first is the detection of these faults using a checker for periodic signals [31]. The other is to reduce the occurrence probability of these faults by suitably designing the circuit layout [30] and to detect them doing periodic off-line testing [32], similar to what is typically done to reveal the occurrence of faults on the system clock signal.

IV. Comparisons and Conclusions.

In this paper we presented a novel method for designing single and double output TSC checkers for Borden codes. The proposed checkers are the first known checkers, for these codes, that are TSC under a realistic fault model including stuck-at, transistor stuck-on, resistive bridging faults, and several transistor stuck-open faults. The corresponding already known from the open literature checkers [16-21] are TSC only under the inadequate stuck-at fault model. The proposed single output TSC checkers for Borden codes are the only known in the open literature.

Among the TSC checkers already proposed [16-21], the checkers given in [16] and [17] are excessively complex.

The checkers proposed in [18] are significantly less complex than the checkers given in [16, 17], but their complexity grows quickly with t. The most recent checkers from [19] and [20] are efficient, but they can be designed for very narrow ranges of k and t. Following the method proposed recently in [21], efficient checkers for any value of t and k can be designed which for the practical values of t and k are more efficient than the checkers given in [18] and in some cases than those given in [19, 20]. The majority of the checkers designed by the method proposed in this paper are more efficient, with respect to the area and speed, than the already known Borden code checkers [16-21]. For example, we have implemented the single and double output 2-UED Borden code checker with k=8 following our method and the checker with the method given in [21] with λ=1μm technology. The improvement in delay is 2,4% for the single and 37% for the double output checker and the improvement in area is 78,8% and 69,44% respectively. The checkers oposed in [21] are double output checkers. We have to note that the area has been estimated as the sum of WxL of the transistors, that is, the routing has not been taken into account. We can see that the proposed single and double output checkers are impressively more efficient, with respect to area than the checkers given in [21], while the proposed double output checkers are also significantly faster. Another advantage of the proposed checkers is their small test set.

References.

[1] W. C. Carter and P. F. Schneider. "Design of dynamically checked computers", in Proc 4th Cong. IFIP, Edinburg, Scotland, vol 2 pp 878-883, Aug 5-10, 1968.

[2] X. Castillo, S. R. McConnel, and D. P. Siewiorek, "Derivation and calibration of a transient error reliability model." IEEE Trans. Comp., vol. C-31, July 1982.

[3] Y. Savaria, N. Rumin, J. Hayes, and V. Agarwal, " Soft-error filtering: A solution to the reliability problem of future VLSI digital circuits" Proc IEEE, vol 74, no 5, 669-683, May 1986

[4] M. M. Yen, W. K. Fuchs, and J. A. Abraham, " Designing for concurrent error detection in VLSI: Application to a microprogram control unit", IEEE J. Solid-State Cir., vol. SC-22, pp. 595-605, Aug. 1987.

[5] D. A. Anderson "Design of Self-Checking digital network, using coding techniques" Coordinated Sci. Lab. Univ. Illinois, Urbana-Champaign, Rep R-527, Sept. 1971.

[6] M. Nicolaidis and B. Courtois, "Strongly Code Disjoint Checkers", IEEE Trans. Comp., Vol. 37, June 1988

[7] T.R.N. Rao, E. Fujiwara, "Error-Control coding for computer systems." Prentice-Hall, Englewood Cliffs, NJ, 1989.

[8] D. Pradhan and J. Stiffler, "Error correcting codes and self-checking circuits in fault tolerant computers", Computer, pp.27-37, Mar. 1980.

[9] G. Mak, J. Abraham, and E. Davidson, " The design of PLAs with concurrent error detection", in Proc. Int. Symp. Fault-Tolerant Comp., Santa Monica, CA, June 1982, pp 303-310.

[10] M. Nicolaidis and M. Boudjit, "New Implementations, Tools, and Experiments for Decreasing Self-Checking PLAs Overhead', Proc. ICCD'91, pp. 275-281.

[11] E. Fujiwara and D. K. Pradhan, "Error-Control Coding in Computers", Computer , vol. 23, pp. 63-72, July 1990.

[12] E. L. Leis, " Data Integrity in Digital Optical Disks", IEEE Trans. Computers, vol. 33, no. 9, pp. 818-827, Sept. 1984.

[13] J. M. Berger, "A note on error detection codes for asymmetric binary channels" Inform. Contr., vol. 4, pp 68-73, Mar. 1961.

[14] C. V. Freiman, "Optimal Error Detection Codes for Completely Asymmetric Binary Channels", Information Control, vol. 5, pp. 64-71, Mar. 1962.

[15] J. M. Borden, " Optimal Asymmetric Error Detecting Codes", Information Control, vol. 53, pp. 66-73, Apr. 1982.

[16] N. Jha, "A Totally Self-Checking Checker for Borden's Code" IEEE Trans. Comp. Aid. Design, vol 8, p 731-736, July 1989

[17] D. Nikolos, et al., " Totally self-checking checkers for optimal t-unidirectional error detecting codes', in Proc. 13th Int. Conf. Fault-Tolerant Syst. Diagn., Varna, Bulgaria, June 20-22, 1990, pp. 326-331.

[18] E. Fujiwara and M. Yoshikawa, " A Design Method for Cost Effective Self-Testing Checker for Optimal d-Unidirectional Error Detecting Codes", IEICE Trans. Information and systems, vol. E75-d,pp. 771-777, Nov. 1992

[19] T. Haniotakis, D. Nikolos, A. Paschalis and D. Gizopoulos, "Totally self-checking checkers for Borden codes", Int. J. Electronics, vol. 76, pp. 57-64, Jan. 1994

[20] T. Haniotakis, A. Paschalis and D. Nikolos, " Efficient Totally self-checking checkers for a Class of Borden codes", IEEE Trans. Computers, vol.44, no. 11, Nov. 1995, pp. 1318-1322.

[21] S. Piestrak, "Design of self-testing checkers for Borden codes", IEEE Trans. Comput., vol. 45, no. 4, Apr. 1996, pp. 461-469.

[22] J. Shen,W. Maly and F. Ferguson, "Inductive Fault Analysis of MOS Intergrated Circuits", IEEE Design & Test of Computers, pp.26-33, December 1985.

[23] C. Metra, M. Favalli, B. Ricco "Highly Testable and Compact Single Output Comparator", Proc. of 15th IEEE VLSI Test Symp. April 27-May 1,1997,Monterey California pp.210-215

[24] X. Kavousianos, D. Nikolos "Self-Exercising, Self-Testing k-order Comparators" 15th IEEE VLSI Test Symposium, pp. 216-221, April 27-May 1, 1997 Monterey California.

[25] M. Afghahi and J. Yuan, " Double edge triggered D flip flop for high speed CMOS circuits", IEEE J. of Solid State Circuit, SC-26, pp. 1168-1170, Aug 1991.

[26] F. Ferguson, and J. Shen, "Extraction and simulation of realistic CMOS faults using inductive fault analysis", Proc. of Int. Test Conf., pp. 475-484, 1988.

[27] H. Hao and E. J. McCluskey, "Resistive Shorts Within CMOS Gates", in Proc. of IEEE Int. Test Conf., pp. 292-301, 1991.

[28] S. Koeppe, "Optimal layout to avoid stuck-open faults", Proc. of Design Automation Conf., pp. 829-835, 1987.

[29] J. Soden, K. Treece, M. Taylor and C. Hawkins, "CMOS IC stuck-open fault electrical effects and design considerations" Proc. of IEEE Int. Test Conf., pp. 423-430, 1989

[30] J. Teixeira, I. Teixeira, C. Almeida, F. Goncalves and J. Goncalves " A Methodology for Testability Enhancement at Layout Level" J. of Electronic Testing: Theory and Application, 1:287 - 299, 1991.

[31] A. M. Usas, "A Totally Self-Checking Checker Design for the Detection of Errors in Periodic Signals", IEEE Trans. Comput. Vol. C-24 No. 5, May 1975, pp. 483-488.

[32] M. Nicolaidis, "Self - Exercising Checkers for Unified Built - In Self - Test (UBIST), IEEE Trans. on CAD, vol. 8, No 3, March 1989, pp. 203 - 218.

[33] S. Kundu, E. S. Sogomonyan, M. Goessel and S. Tarnick "Self-Checking Comparator with One Periodic Output", IEEE Trans. Comput., 45:379-380, March 1996.