Novel Single and Double Output TSC Berger Code Checkers

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

Abstract

This paper presents a novel method for designing type-I and type-II single and double output TSC Berger code checkers taking into account a realistic fault model including stuck-at, transistor stuck-open, transistor stuck-on, resistive bridging faults and breaks. A benefit of the proposed type-I single and double output checkers is that all faults are testable by a very small set of code words the number of which does not increase with the information length, that is, the checkers are C-testable. The proposed double output checkers are two-times faster than the corresponding single output checkers, but require for their implementation twice as many transistors as the single output checkers. The proposed single output checkers are the first known TSC Berger code checkers in the open literature, while the type-I single output checkers are near optimal with respect to the number of the transistors required for their implementation. The checkers of this paper, with either, single or double output are significantly more efficient, with respect to the implementation area and speed, than the already known from the open literature Berger code checkers.

Index Terms

Self-Checking Circuits, Totally Self-Checking Circuits, Berger Codes, Unidirectional Errors.

I. Introduction

Self Checking Circuits (SCC) [1] are widely used in applications with high reliability requirements, due to their ability to detect errors on line during the normal system operation. The errors covered include those caused by permanent, transient as well as intermittent faults. A SCC consists 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.

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) [2] or the Strongly Code-Disjoint (SCD) [3] property. In this paper we will take into account the TSC property. The TSC checker is a circuit which satisfies the self-testing, fault secure and code disjoint properties [2, 4].

It has been observed for many years that a large number of errors in VLSI circuits and compact laser disks are of unidirectional type [5-7]. Berger codes [8] are the least redundant separable codes among the All Unidirectional Error Detecting (AUED) codes [9]. For k information bits, the check part has length $r = \lceil \log_2(k+1) \rceil$. There are two different encoding schemes for Berger code: B_0 and B_1. The B_0 encoding scheme uses the binary representation of the number of 0's in the information bits as the check symbol, whereas the B_1 encoding scheme uses the ones complement of the number of 1's in the information bits.

Due to the wide use of Berger codes, several design methods of Berger code checkers were proposed in the open literature [10-17]. The checkers given in [10-15] are TSC only with respect to single stuck-at faults; only the checkers designed in [16, 17] are TSC with respect to a realistic fault model [24] including stuck-at, transistor stuck-open, transistor stuck-on and resistive bridging faults. Apart from the above the checkers given in [17] have the advantage that require less area and feature higher speed than the checkers proposed in [14-16]. However, the checkers proposed in [17] have static power consumption.

In this paper a new method for designing TSC Berger code checkers is proposed. The checkers designed according to this method are TSC with respect to stuck-at, transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks. The proposed checkers are significantly more efficient, with respect to area and speed, than the already known TSC Berger code checkers. Apart from this the proposed checkers have significantly lower power consumption than the checkers given in [17]. We use the B_1 encoding scheme, but the modifications for B_0 scheme are straightforward.

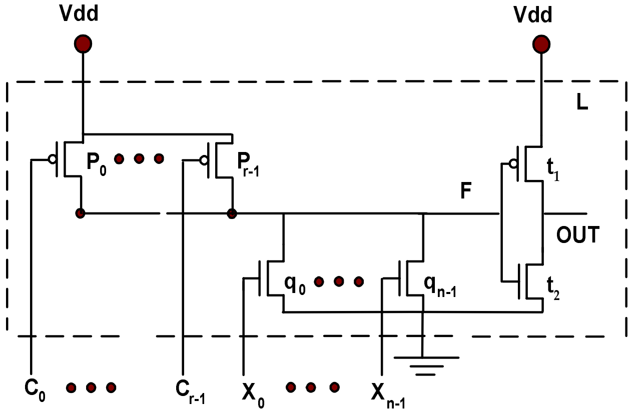

Figure 1. (r, n) aggregate threshold circuit.

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 [19, 20]. To this end, apart from double output we also present single output TSC Berger code checkers.

Throughout this paper the following notations are used :

- 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 nmos (pmos) transistor.
- β_n, (β_p) is the gain factor of nmos (pmos) transistors.
- KP_n (KP_p) is the Spice parameter for $\mu_n \cdot C_{ox}$ ($\mu_p \cdot C_{ox}$).
- W_{ni}/L_{ni} (W_{pi}/L_{pi}) is the ratio of nmos (pmos) transistor i.
- r is the number of check bits of the Berger code.
- n is the number of information bits of the Berger code.
- W(X) denotes the Hamming weight of the vector X, that is, the number of ones.

II. Design Method

The design of the proposed in this paper Berger code checkers is based on a circuit called (r, n) aggregate-weight threshold circuit.

A. (r, n) aggregate-weight threshold circuit.

We call the circuit of figure 1 an (r, n) aggregate-weight threshold circuit and its operation is given by the following definition.

Definition 1. A circuit with r+n inputs C_{r-1}, C_{r-2}, …, C_0 and X_{n-1}, X_{n-2}, …, X_0, and one output, OUT, is an (r, n) aggregate-weight threshold circuit if for

$$W(X) \geq \overline{C}_{r-1} 2^{r-1} + \overline{C}_{r-2} 2^{r-2} + \ldots + \overline{C}_0$$

gives OUT=1 else OUT=0.

In the circuit of figure 1 the nmos transistors q_0, q_1, …, q_{n-1} have been chosen to have the same width and the same length. Therefore the circuit of figure 1 will be an (r, n) aggregate-weight threshold circuit if the sizes of the pmos transistors satisfy some relations. To derive those relations the following definitions are necessary.

Definition 2. The weight $TW(P_a)$ of a transistor P_a, a∈[0, r-1], in an (r, n) aggregate-weight threshold circuit is equal to k if the circuit operates as follows :
When among the transistors P_i , for i = 0, 1, …, r-1, only the transistor P_a is conductive, that is C_a=0 and C_i=1 for i∈[0, r-1] and i≠a, then we have OUT=1 if at least k of the inputs X_j, j∈[0, n-1], are equal to one, else we have OUT=0.

Definition 3. The aggregate weight $AW(P_{a1}, P_{a2}, …,, P_{am})$, of a combination of transistors P_{a1}, P_{a2},…, P_{am} of the (r, n) aggregate-weight threshold circuit, with a1, a2, …, am ∈[0, r-1], is equal to k if the circuit operates as follows :
when among the transistors P_i, for i = 0, 1, …, r-1, only the transistors P_{a1}, P_{a2}, …, P_{am} are conductive then we have OUT=1 if at least k of the inputs X_j, j∈[0, n-1], are equal to one, else we have OUT=0.

We notice that when W(X)=k, then k of the transistors q_0, q_1, …, q_{n-1} are conductive. Consequently, taking into account the definition 3, the definition of the (r, n) aggregate weight threshold circuit is equivalent to the following definition.

Definition 4. The circuit of figure 1 is an (r, n) aggregate-weight threshold circuit if for each input vector C_{r-1}, C_{r-2}, …, C_0, the aggregate weight of the conductive pmos transistors $P_{a0}, P_{a1}, \ldots, P_{ai}$, with a0, a1, …, ai ∈ [0, r-1], is equal to $\overline{C}_{r-1} \cdot 2^{r-1} + \overline{C}_{r-2} \cdot 2^{r-2} + \ldots + \overline{C}_0$.

In the sequel we present the design procedure of the (r, n) aggregate weight threshold circuit.

Design Procedure.

Design the circuit of figure 1 so that the aspect ratio W_{Pa}/L_{Pa} of each P_a for a= 0,1,2, …, r-1, to satisfy the relation

$$(2^a - 1/(r+1))Q_1 \leq W_{Pa}/L_{Pa} \leq 2^a Q_2 \qquad (1)$$

where

$$Q_1 = \frac{KP_n}{KP_p} \cdot \frac{2(V_{dd} - V_{tn})V_{OHMIN} - V_{OHMIN}^2}{\left(V_{dd} + V_{tp}\right)^2} \cdot \frac{W_n}{L_n}$$

and

$$Q_2 = \frac{KP_n}{KP_p} \cdot \frac{2\left(V_{dd} - V_{tn}\right)V_{OLMAX} - V_{OLMAX}^2}{\left(V_{dd} + V_{tp}\right)^2} \cdot \frac{W_n}{L_n}$$

To prove that the circuit designed by the above Design Procedure is an (r, n) aggregate-weight threshold circuit the following theorem is necessary.

Theorem 1. Let W_n/L_n be the aspect ratio of the nmos transistors q_0, q_1, …, q_{n-1}, and W_{Pa}/L_{Pa} be the aspect ratio of the transistor P_a of the (r, n) aggregate-weight threshold circuit, with a∈[0, r-1]. If $(k-1)Q_1 \leq W_{Pa}/L_{Pa} \leq k \cdot Q_2$ then the weight of transistor P_a is $TW(P_a)$ = k.

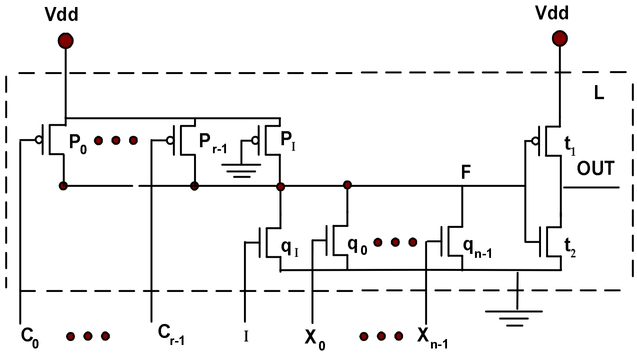

Figure 2. Type-I Single Output TSC checker.

Proof. Consider the (r, n) aggregate-weight threshold circuit of figure 1 and that among the transistors P_i , with $i\in[0, r-1]$, only the transistor P_a is conductive, that is $C_a=0$ and $C_i=1$ for $i\in[0, r-1]$ with $i\neq a$ (then the circuit of figure 1 is equivalent to module D of the circuit of fig. 2 in [18]). Then according to the definition 2 the weight of transistor P_a is $TW(P_a) = k$ if :
when k or more inputs X_i of the circuit of figure 1 are high then the output OUT is high else OUT is low. Such a circuit will be called a k-weight threshold circuit and a systematic method to design such a circuit has been given in [18]. In [18] we have shown that such a circuit is a k-weight threshold circuit as long as the ratio of the transistor aspect ratios satisfy the following relation :

$$(k-1)Q_1 \leq W_{Pa}/L_{Pa} \leq k\cdot Q_2$$

where V_{OHMIN}, V_{OLMAX} refer to the output of the threshold circuit at the point F. ■

When we connect two transistors in parallel the total conductance is the sum of the conductance of each transistor. The conductance of a transistor is approximated in [21, p. 61] for the linear and saturation regions respectively, by the following relations:

$$G_{linear} = KP\cdot W/L\cdot V_{drain\text{-}source} \text{ and}$$

$$G_{saturation} = KP\cdot W/L\cdot(V_{gate\text{-}source} - V_{threshold})$$

Assuming that all the conductive pmos (or nmos) transistors in parallel have the same gate voltage, they are all in the same region of operation, so we can simplify the above equation into the $G = f\cdot W/L$, where f is a factor that depends on the region of operation of the transistors. So for two transistors connected in parallel we have $G_{total} = G_1 + G_2 = f(W_1/L_1 + W_2/L_2)$. Therefore, if we connect in parallel two pmos (or nmos) transistors of ratios W_1/L_1 and W_2/L_2 then when they are both conductive they have the same conductance with one pmos (or nmos) of ratio $W_1/L_1+W_2/L_2$ which is also conductive.

Theorem 2. The circuit designed by Design Procedure is an (r, n) aggregate-weight threshold circuit.

Proof. According to Design Procedure for each transistor of a combination P_{a1} P_{a2}, …, P_{am} with $a_1, a_2, \ldots, a_m \in[0, r-1]$, we have:

$$(2^{ai}-1/(r+1))\cdot Q_1 \leq W_{Pai}/L_{Pai} \leq 2^{ai\cdot} Q_2, \text{ for } i=1, 2, \ldots, m.$$

If we add the above relations from i=1 to i=m, we get

$$(2^{a1}+\ldots+2^{am}-m/(r+1))\cdot Q_1 \leq W_{Pa1}/L_{Pa1} +\ldots+ W_{Pam}/L_{Pam} \leq (2^{a1}+\ldots+2^{am})\cdot Q_2$$

As we have already shown, the parallel connection of conductive transistors P_{a1} P_{a2}, …, P_{am} is equivalent to a single transistor P_{total} which has ratio

$$W_{ptotal} / L_{Ptotal} = W_{Pa1}/L_{Pa1} +\ldots+ W_{Pam}/L_{Pam}$$

Then taking into account the above and the fact thar m<r+1 we have

$$(2^{a1}+\ldots+2^{am}-1)\cdot Q_1 \leq W_{Ptotal}/L_{Ptotal} \leq (2^{a1}+\ldots+2^{am})\cdot Q_2$$

The above relation implies that the transistor P_{total} satisfies Theorem 1 and the weight of P_{total} is

$$TW(P_{total}) = \sum_{i=1}^{m} 2^{ai} \text{ which means, } AW(P_{a1}, \ldots, P_{am}) = \sum_{i=1}^{m} 2^{ai} .$$

Then the above relation implies that each combination of conductive transistors P_{a1}, P_{a2}, …, P_{am} with $a_1, a_2, \ldots, a_m\in[0, r-1]$ has aggregate weight $2^{a1}+2^{a2}+\ldots+2^{am}$. Taking into account that the transistors P_{a1}, P_{a2}, …, P_{am} are conductive when the inputs C_{a1}, C_{a2}, …, C_{am} are equal to zero we conclude that for each input vector $C_{r-1}C_{r-2}\ldots C_0$ the aggregate weight of the conductive transistors is equal to $\overline{C}_{r-1}\cdot 2^{r-1} + \overline{C}_{r-2}\cdot 2^{r-2}+\ldots+\overline{C}_0$. ■

B. Single and Double Output Berger Code Checkers.

Consider the circuit of figure 2 which results from the circuit of figure 1 if we append the pmos transistor P_I and the nmos transistor q_I. Transistor P_I has the same width and length with transistor P_0, while transistor q_I, is identical to the transistors $q_0, q_1, \ldots, q_{n-1}$. Consider for the time being that I=0. Then, according to definition 1, $TW(P_I)=TW(P_0)=2^0=1$. P_I is permanently conductive so we have $AW(P_{a1}, P_{a2}, \ldots, P_{am}, P_I)=$

$$1+\sum_{i=1}^{m} 2^{ai} = \overline{C}_{r-1}\cdot 2^{r-1} + \overline{C}_{r-2}\cdot 2^{r-2}+\ldots+\overline{C}_0 +1$$

Then taking into account Definition 3 we have :
if $W(X) \geq \overline{C}_{r-1}\cdot 2^{r-1}+\overline{C}_{r-2}\cdot 2^{r-2}+\ldots+\overline{C}_0+1$ then OUT=1, else OUT= 0. Now taking into account the value of input I we conclude that the operation of the circuit of figure 2 is described by the following relations:
if $W(X, I) \geq \overline{C}_{r-1}\cdot 2^{r-1} + \overline{C}_{r-2}\cdot 2^{r-2}+\ldots+\overline{C}_0+1$ then OUT=1, else OUT=0 or equivalently if
$W(X)+W(I) \geq \overline{C}_{r-1}\cdot 2^{r-1} + \overline{C}_{r-2}\cdot 2^{r-2}+\ldots+\overline{C}_0+1$ then OUT=1 , else OUT=0.

Table 1 presents the value of the output of the circuit of figure 2 for all possible input combinations. Consider that the input I in figure 2 is driven by the system clock, $X=(X_0, X_1, \ldots, X_{n-1})$ is the information part and $C= (C_0, C_1, \ldots, C_{r-1})$ is the Berger code symbol corresponding to X. When the input vector XC is a Berger code word and the circuit is fault free then during a period of the signal I the output OUT gets the values (0, 1). When the input vector is not a

Table 1: Operation of the checker.

Condition	Figure 2		Figure 3	
	OUT		OUT_0, OUT_1	
	I=0	I=1	I=0	I=1
$W(X) < \overline{C}_{r-1} \cdot 2^{r-1} + \ldots + \overline{C}_0$	0	0	0, 0	0, 0
$W(X) = \overline{C}_{r-1} \cdot 2^{r-1} + \ldots + \overline{C}_0$	0	1	1, 0	0, 1
$W(X) > \overline{C}_{r-1} \cdot 2^{r-1} + \ldots + \overline{C}_0$	1	1	1, 1	1, 1

code word then during a period the output OUT gets the values (1, 1) or (0, 0). In other words the circuit of figure 4 is a single output Berger code checker. As in the case of the single output comparator given in [20] 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 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 [26]).

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 Berger 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 [20]). However as we will see the delay of the proposed single output checkers is very small, thus they can be used in most applications.

The implementation of the single-output TSC Berger code checker requires n+r+4 transistors, where n+r is the number of inputs. Taking into account that the implementation of a function requires at least as many transistors as the number of its inputs, we conclude that the proposed checker is near optimal with respect to the number of the required for its implementation transistors.

The double output Berger code checker is given in Figure 3. Module L_0 as well as module L_1 is identical to module L of figure 2. The input I is driven by a clock signal with the half frequency of feeding inputs to the checker. This feeding frequency is usually equal to the frequency of the system clock, therefore the signal driving input I can be easily obtained from the system clock using a T flip flop. The operation of the circuit is described in Table 1.

The manufacturability of the proposed checkers depends on the manufacturability of the (r, n) aggregate-weight threshold circuit. This circuit is a ratioed circuit. A problem of a ratioed circuit is that its 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 (r, n) aggregate-weight threshold circuit 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 range given by relation (1). 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 r becomes greater the range defined by relation (1) becomes 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 (r, n) aggregate-weight threshold circuits for larger values of r can be constructed. However, given the quality of a manufacturing process there exist a maximum value of r for which Berger code checkers can be constructed following our method.

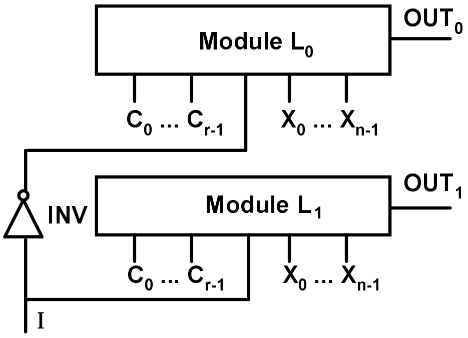

Figure 3. Type-I Double output checker.

In the sequel we will modify the proposed Berger code checker with r-1 check bits, in order to implement the checker for the Berger code with r check bits. That is, we increase by one the maximum value of r for which we can construct Berger code checkers following our method. (Note that increasing by one the value of r we double the length of the information word for which we can construct Berger code checkers.) Let's assume that we have to design the Berger code checker for the case of n information bits $X_0, X_1, \ldots, X_{n-1}$ and r check bits $C_0, C_1, \ldots, C_{r-1}$ ($2^{r-1} \le n \le 2^r-1$). Then for a code word

$W(X) = \overline{C}_{r-1} \cdot 2^{r-1} + \overline{C}_{r-2} \cdot 2^{r-2} + \ldots + \overline{C}_0 \cdot 2^0$ or

$W(\overline{X}) = n - (\overline{C}_{r-1} \cdot 2^{r-1} + \overline{C}_{r-2} \cdot 2^{r-2} + \ldots + \overline{C}_0 \cdot 2^0)$ or

$W(\overline{X}) = n - 2^r + 1 + (C_{r-1} \cdot 2^{r-1} + \ldots + C_0 \cdot 2^0)$ or

$W(\overline{X}) + 2^r - n - 1 = (C_{r-1} \cdot 2^{r-1} + \ldots + C_0 \cdot 2^0)$.

In other words, $X_0, X_1, \ldots, X_{n-1}$, $C_0, C_1, \ldots, C_{r-1}$ is a code word if the following equivalent relations are satisfied:

$W(X) = \overline{C}_{r-1} \cdot 2^{r-1} + \overline{C}_{r-2} \cdot 2^{r-2} + \ldots + \overline{C}_0 \cdot 2^0$

and $W(\overline{X}) + 2^r - n - 1 = (C_{r-1} \cdot 2^{r-1} + \ldots + C_0 \cdot 2^0)$.

Since these two conditions are equivalent, we choose to check the validity of the first for C_{r-1}=1 and the validity of the second for C_{r-1}=0, so the relations become:

If C_{r-1}=1 then $W(X) = \overline{C}_{r-2} \cdot 2^{r-2} + \overline{C}_{r-3} \cdot 2^{r-3} + \ldots + \overline{C}_0 \cdot 2^0$ (2)

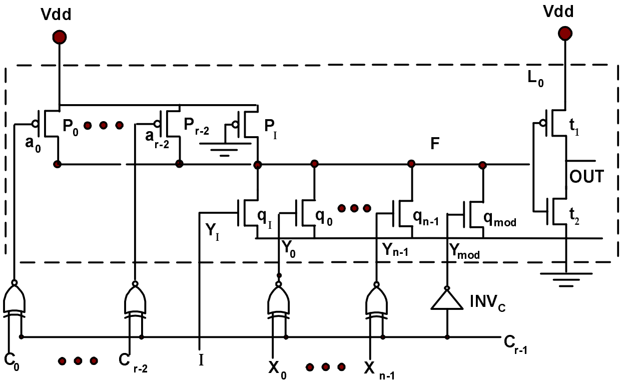

Figure 4. Type II Single output checker.

and if C_{r-1}=0 then $W(\overline{X}) + 2^r - n - 1 = C_{r-2} \cdot 2^{r-2} + ... + C_0 \cdot 2^0$ (3)
The meaning of relations (2) and (3) is that depending on the value of C_{r-1} we can count zeros or ones. These relations lead us to the circuit of figure 4, where module L has been designed following the design method of section II for r-1 check bits and n information bits and the aspect ratio of transistor q_{mod} is given by the relation W_{mod}/L_{mod} = $(2^r$-n-1)W_{qi}/L_{qi}. Obviously, for n=2^r-1 (complete Berger code) transistor q_{mod} is not appended. For C_{r-1}= 1 and C_{r-1}=0 the circuit of figure 4 checks respectively the validity of relations (2) and (3). We can easily verify that the circuit of figure 4 is a single output Berger code checker working similar to that of figure 2. The design of the corresponding double output checker is obvious.

III. Testability Analysis.

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

A. Type-I single output checkers.

The transistor P_I or transistor t_1 stuck-on fault 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. All the other stuck-at, transistor stuck-open and transistor stuck-on faults are detected by a codeword.

The self-checking capability with respect to resistive bridging faults and break faults on device terminals has been evaluated with extensive circuit-level simulations. Resistive bridging faults (RBFs) between two transistor terminals or between two inputs have been considered. All RBFs with connecting resistance R ∈ [0,R_{max}] are detected, where R_{max} depends on the sizing of the transistors and the information length. We are interested for resistances in the range [0, 6KΩ] [23]. For the 8 bit Berger code checker of figure 2 and an implementation in λ=1μm technology with transistor aspect ratios $(W/L)pm_0$=5/1, $(W/L)pm_1$=10/1, $(W/L)pm_2$=20/1, $(W/L)pm_3$=41/1, $(W/L)pm_I$=5/1 and $(W/L)nm_i$=2/1, for i=1 to n, the value of R_{max} is 6KΩ for all cases except the following faults: a. R_{max} = 4,9KΩ for the gate - source bridging fault of transistor q_i, i∈[1, n], b. R_{max} = 3,2KΩ for the gate - source bridging fault of transitor p_0, c. The gate - source bridging fault of transistor p_I is undetectable. During the simulation the inputs of the checker are driven by standard cell inverters with aspect ratios $(W/L)_p$=12 and $(W/L)_n$=6. For the 16bit checker we found negligible differences in the values of R_{max}.

The proposed checkers are Self Testing for all break faults on device terminals.

The test set of the checker consists of a) the vectors that are required so as to apply 0 and 1 at each input X_i with i∈[0, n-1] and at each input C_j with j∈[0, r-1] and b) the vectors that are required so as to apply the pattern 01 or 10 to any adjacent inputs $X_{i,}$ X_{i+1} with i∈[0, n-2] and C_j, C_{j+1} with j∈[0, r-2]. When n=2^r-1 the test set consists of 4 code words. For example, when n=7 the test vectors are ($C_2C_1C_0$, $X_0X_1X_2X_3X_4X_5X_6$) = (000, 1111111), (111, 0000000), (101, 0000011) and (011, 1010101). When n<2^r-1, a code word with all the check bits equal to 0 does not exist. Therefore we have to apply two code words with check parts respectively 011…1 and 100…0, which always exist, so that each input C_j to take the value 0. Therefore in this case the test set consists of 5 vectors. The above implies that the type-I single output checkers are C-testable, that is, the cardinality of the test set does not increase with the information length.

B. Type-I double output checkers.

The testability analysis of the double output checker of figure 3 is similar to that of figure 2 except of a stuck-at 0 or 1 fault on line I, which is undetectable. The problem can be overcame reducing the occurrence probability of these faults by suitably designing the circuit layout [25] and to detect them doing periodic off-line testing [22], similar to what is typically done to reveal the occurrence of faults on the system clock signal. The test set of this double output checker is the same with that of the single output checker of figure 2 but these vectors should be applied for I=0 and I=1.

C. Type-II single output checkers.

The testability analysis of the single output checker of figure 4 is similar to that of figure 2 with only one difference, the testability of the XNOR gates. We have to note that in this case for some values of n, some XNORs that are driven by some inputs C_i, i∈[0, r-1] do not take the combination (0, 0) at their inputs. Single stuck-at faults on the terminal lines of each XNOR gate are always testable. However the testability of transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks depend on the implementation of the XNOR gate.

IV. Comparisons and Conclusions.

In this paper we presented a novel method for designing single and double output TSC Berger code checkers under a realistic fault model including stuck-at, transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks. The proposed single output TSC Berger code checkers are the first known in the open literature, and they are near optimal with respect to the number of the transistors required for their implementation. Another benefit of the proposed type-I checkers is that all faults are testable by a very small set of code words the number of which does not increase with the information length.

The checkers designed following our method are significantly more efficient with respect to area and speed in comparison to the corresponding already known checkers. Among the already known TSC checkers for Berger codes the checkers proposed in [17] are the most efficient with respect to the required area and speed as well as the fault model. These Berger code checkers take into account apart from single stuck-at faults, transistor stuck-on, transistor stuck-open and resistive bridging faults too.

We have implemented the proposed Berger code checkers as well as that given in [17] for 8 information bits with λ=1μm technology. The reductions achieved by the proposed checkers in delay, area and power consumption (100MHz), against the checkers given in [17] are respectively 43%, 85% and 82% for the single output type I checker and 72%, 68% and 65% for the double output type I checker. We can see that the proposed checkers are impressively more efficient with respect to speed, area and power consumption than the checkers given in [17]. 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 easily see that the routing in the proposed design is significantly less than the routing required for the implementation of the checkers given in [17]. The above implies that taking into account the routing the reduction of the required area becomes even larger in favor to the proposed checkers. An improvement of the design presented in [17] was recently proposed in [27]. It is ease to see that the checkers proposed here are more efficient with respect to area, speed and power consumption than the checkers given in [27].

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] D. A. Anderson “Design of Self-Checking digital network, using coding techniques” Coordinated Sci. Lab. Univ. Illinois, Urbana-Champaign, Rep R-527, June 1984

[3] M. Nicolaidis and B. Courtois, "Strongly Code Disjoint Checkers", IEEE Trans. Comp., pp 751-756, June 1988.

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

[5] D. K. Pradhan and J. I. Stiffler, “Error correcting codes and self-checking circuits in fault tolerant computers”, Computer, pp.27-37, Mar. 1980

[6] E. L. Leiss, “Data Integrity in Digital Optical Disks”, IEEE Trans.Computers, vol. 33, no. 9, pp. 818-827, Sept. 1984.

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

[8] J. M. Berger, “A note on error detection codes for asymmetric binary channels”, Inform. Contr., pp 68-73 , Mar. 1961.

[9] C. V. Freiman , “Optimal error detection codes for completely asymmetric binary channels” Inform. Contr., vol. 5, pp.64-71, Mar. 1962.

[10] M. A. Marouf and A. D. Freidman. “Design of self-checking checkers for Berger codes.” In Proc. Of Int. Symp. Fault-Tolerant Comput., pp. 179-184, 1978

[11] S. Piestrak, “Design of Fast Self-Testing Checkers for a Class of Berger Codes”, Trans. On Computers, vol. C-36, pp. 629 -634, May 1987.

[12] J. C. Lo and S. Thanawastien, “The design of fast totally self-checking Berger code checkers based on Berger code partitioning”, in Proc. FTCS-19, pp.226-231 June 1988.

[13] S. J. Piestrak, “The minimal test set for sorting networks and the use of sorting networks in self-testing checkers for unordered codes”, Proc. of FTSC-20, 1990, pp. 457-464.

[14] T. R. N. Rao ,G. Feng, M. S.Kolluru, and J. C. Lo, “Novel Totally Self-Checking Berger Code Checker Designs Based on Generalized Berger Code Partitioning”, IEEE Trans. on comp., vol. 42, pp. 1020 - 1024, August 1993.

[15] D. Pierce and P. K. Lala, “Efficient Self-Checking Checkers for Berger Codes”, in Proc. Of 1st IEEE Int. On-Line Testing Workshop. ,pp. 238 - 242 ,1995.

[16] C. Metra, M. Favalli, and B. Ricco, “Novel Berger Code Checker”, in Proc. IEEE Int. Work. On Defect and Fault Tolerance in VLSI Systems, pp. 287 - 295, 1995.

[17] C. Metra and J. C. Lo, “Compact and High Speed Berger Code Checker”, 2nd IEEE Int. On-Line Testing Workshop, Biarritz, Franze, July 8-10, 1996, pp.144-149.

[18] 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.

[19] 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.

[20] C. Metra, M. Favalli, B. Ricco “Highly Testable and Compact Single Output Comparator” 15th IEEE VLSI Test Symposium pp. 210-215, April 27-May 1, 1997.

[21] Neil H. E. Weste, Kamran Eshraghian, *Principles of CMOS VLSI Design, A systems Perspective*, 2nd ed., Add. Wesley.

[22] 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

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

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

[25] 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.

[26] 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.

[27] Y. Y. Guo, J.JC. Lo and C. Metra, "Fast and Area-Time Efficient Berger Code Checkers", Proc. IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems, Oct. 20-22, 1997, Paris, Fr., pp. 110-118.