VLSI DESIGN
2000, Vol. 11, No. 1, pp. 35–45
Reprints available directly from the publisher
Photocopying permitted by license only

Novel Single and Double Output TSC CMOS Checkers for *m*-out-of-*n* Codes*

X. KAVOUSIANOS[a,b], D. NIKOLOS[a,b,†] and G. SIDIROPOULOS[a,b]

[a]*Department of Computer Engineering and Informatics, University of Patras, 26500, Patras, Greece;*
[b]*Computer Technology Institute, Kolokotroni 3, Patra, Greece*

(Received 1 April 1999; In final form 5 October 1999)

This paper presents a novel method for designing Totally Self-Checking (TSC) *m*-out-of-*n* code checkers taking into account a realistic fault model including stuck-at, transistor stuck-on, transistor stuck-open, resistive bridging faults and breaks. The proposed design method is the first method in the open literature that takes into account a realistic fault model and can be applied for most practical values of *m* and *n*. Apart from the above the proposed checkers are very compact and very fast. The single output checkers are near optimal with respect to the number of transistors required for their implementation. Another benefit of the proposed TSC checkers is that all faults are tested by a very small set of single pattern tests, thus the probability of achieving the TSC goal is greater than in checkers requiring two-pattern tests. The single output TSC checkers proposed in this paper are the first known single output TSC checkers for *m*-out-of-*n* codes.

Keywords: Self-Checking Circuits, Totally Self-Checking circuits, unidirectional errors, *m*-out-of-*n* codes

I. INTRODUCTION

A variety of error control codes has been proposed and many of them have been used to enhance the reliability of computer systems [1–3]. 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) [4]. 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, providing

*Based on "Design of Compact and High Speed Totally Self-Checking CMOS Checkers for *m*-out-of-*n* Codes" by X. Kavousianos, D. Nikolos and G. Sidiropoulos, which appeared in *Proceedings of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, Paris, France, pp. 128–136. © 1997 IEEE.

†Corresponding author. Tel.: (+3061) 997752, Fax: (+3061) 991909, e-mail: nikolosd@cti.gr

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 [1, 5].

DEFINITION 1 A circuit is *self-testing* for a set of faults F, if for every fault in F, the circuit produces a non-code output for at least one code input.

DEFINITION 2 A circuit is *fault-secure* for a set of faults F, if for every fault in F, the circuit never produces an incorrect code output for any code input.

DEFINITION 3 A circuit is *code-disjoint* if during fault free operation, code inputs map into code outputs and non-code inputs map into non-code outputs.

Some authors believe that the fault secure property is meaningless for checkers [7, 8] while some others believe that it is useful for all others except for the final checker [21]. However, most known self-testing/code-disjoint checkers are also fault secure. Thus the addition of the fault secure property to such a checker does not require any extra area or delay overhead.

It has been shown that a large number of errors in VLSI circuits and compact laser disks are of unidirectional type [9–11]. This means that in any given data word the errors can be either $0 \rightarrow 1$ type or $1 \rightarrow 0$ type, but not both. Many codes have been developed to detect unidirectional errors, among the most known are the m-out-of-n codes [12].

Apart from the low redundancy (small number of check bits) of a code, its suitability for use in a computer system heavily depends also on the existence of a simple and fast TSC checker for this code. Unless the hardware needed to implement the checker is relatively simple compared with the hardware monitored, a fault-prone checker could increase rather than decrease the likelihood of erroneous information propagation.

The problem of designing TSC checkers for m-out-of-n codes or special classes of m-out-of-n codes as 1-out-of-n and m-out-of-$2m$ codes under the assumption of the single stuck-at fault model has been extensively studied in the literature [13–28]. However, the conventional stuck-at fault model has been found to be inadequate for CMOS circuits [29]. 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 [29].

TSC CMOS checkers under single stuck-at and transistor stuck-open faults have been proposed in [30] for m-out-of-$2m$, m-out-of-$2m+1$, $(m-1)$-out-of-$(2m-1)$ and $(m+1)$-out-of-$(2m+1)$ codes and in [31] and [32] for m-out-of-$2m$ codes. Also TSC CMOS checkers for a subset of m-out-of-$2m$ codes under stuck-at, stuck-open, stuck-on, breaks and some bridging faults have been given in [33]. TSC checkers are used to achieve the Totally Self-Checking goal (*i.e.*, the first erroneous output of a functional block is signaled by the checker). The achievement of the TSC goal is based on two assumptions: (a) faults occur one at a time, and (b) there is a sufficient time interval between the occurrence of any two faults so that all required code inputs can be applied to the circuit. The stuck-open faults in the checkers proposed in [30–33] require two-pattern tests to be detected. The probability that the checker will receive, during the normal operation, all the required two-pattern tests in a short period of time is much smaller than the probability to receive a test set of equal length consisting of single pattern tests. Therefore the checkers proposed in [30–33] have very small probability of achieving the TSC goal, which is the target of their use. Apart from the above drawback the checkers proposed in [30, 31] are fully unstructured and thus they are not suitable for VLSI implementation.

Recently PLA Self-testing checkers for incomplete m-out-of-n codes and 1-out-of-n codes were proposed in [34]. Metra proposed in [35] a novel method for designing TSC 1-out-of-n code checkers under a realistic fault model including stuck-at, resistive bridging faults, breaks, transistor stuck-on and the majority of transistor stuck-open faults. It was shown that the TSC 1-out-of-n checkers proposed in [35] require impressively less area than the corresponding already known TSC checkers.

From the above it is evident that no method for designing TSC m-out-of-n code checkers, under realistic faults, has been proposed yet in the open literature. In this paper we give such a method.

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. For example the routing of the error signals coming from different checkers would be simplified. There are also applications where the system poses particular constraints on the number of possibly used input/output signals [39]. No single output TSC m-out-of-n code checker is up today known from the open literature. To this end, apart from double output we also present single output TSC checkers for m-out-of-n codes.

The rest of the paper is organized as follows. In Section II we give the design method while in Section III we present the testability analysis. Discussion and comparisons are given in Section IV.

II. DESIGN METHOD

The design of the proposed m-out-of-n code checker is based on the circuit of Figure 1, which in the sequel we will call "m-weight threshold circuit". When m or more inputs X_i of the m-weight threshold circuit are high then the output OUT is low else OUT is high. The circuit of Figure 1 is the same with the threshold function generator used in [38]. However a systematic method for designing such a circuit has not been given in [38]. In the sequel we will give a systematic method for designing such a threshold circuit.

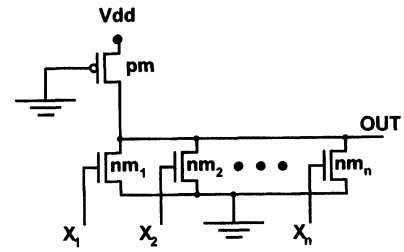

FIGURE 1 m-weight threshold circuit.

The following notations are used in the paper

- V_{OHMIN} (V_{OLMAX}) is the minimum HIGH (maximum LOW) voltage at the output of the threshold circuit.
- V_{tn} (V_{tp}) is the threshold voltage of $n$mos ($p$mos) transistor.
- $\beta_n(\beta_p)$ is the gain factor of $n$mos ($p$mos) 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 $n$mos ($p$mos) transistor i.
- $W(X)$ denotes the Hamming weight of the vector X, that is, the number of ones.

The m-weight threshold circuit (Fig. 1) operates as follows: if less than m of the transistors nm_1, $nm_2, \ldots, nm_n$ are conductive, then OUT = 1 (or $V_{\text{out}} \geq V_{\text{OHMIN}}$) else if at least m of the transistors $nm_1, nm_2, \ldots, nm_n$ are conductive then OUT = 0 (or $V_{\text{out}} \leq V_{\text{OLMAX}}$).

Let us say now that λ of the transistors nm_1, $nm_2, \ldots, nm_n$ are conductive. In the region of our interest [V_{OLMAX}, V_{OHMIN}] transistor pm is in saturation region, the λ conductive transistors of the $nm_1, nm_2, \ldots, nm_n$ are in the linear region while the rest of them are in cut-off region. Let I_{dsp} be the current flowing through transistor pm, I_{dsn} the current flowing through nm_j, one of the conductive transistors among nm_1, $nm_2, \ldots,$ nm_n. We consider all the $n$mos transistors nm_1, $nm_2, \ldots, nm_n$ having the same sizes W_n (width)

and L_n (length). Then we have

$$I_{dsp} = \lambda \cdot I_{dsn} \text{ or equivalently}$$

$$\beta_p \cdot \frac{(V_{dd} + V_{tp})^2}{2} = \lambda \cdot \beta_n \cdot \left[(V_{dd} - V_{tn}) \cdot V_{\text{out}} - \frac{V_{\text{out}}^2}{2}\right]$$

or equivalently

$$\frac{\beta_p}{\lambda \cdot \beta_n} = \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{out}} - V_{\text{out}}^2}{(V_{dd} + V_{tp})^2} \tag{1}$$

The function $f(x) = (-x^2 + 2 \cdot (V_{dd} - V_{tn}) \cdot x)/(V_{dd} + V_{tp})^2$ is maximized at the point $x = V_{dd} - V_{tn}$ and $f(x)$ is monotone increasing for $x < V_{dd} - V_{tn}$. We are interested in the region $[V_{\text{OLMAX}}, V_{\text{OHMIN}}]$ so we assume that $V_{\text{out}} \leq V_{dd} - V_{tn}$. Then taking into account the monotony of the function we conclude that for $V_{\text{out}} \geq V_{\text{OHMIN}}$ we have $f(V_{\text{out}}) \geq f(V_{\text{OHMIN}})$. When $m-1$ of the transistors nm_1, $nm_2, \ldots, nm_n$ are conductive the output voltage must satisfy the condition $V_{\text{out}} \geq V_{\text{OHMIN}}$. Setting $\lambda = m - 1$ in Eq. (1) we get

$$\frac{\beta_p}{\beta_n} \geq (m-1) \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OHMIN}} - V_{\text{OHMIN}}^2}{(V_{dd} + V_{tp})^2} \tag{2}$$

For $V_{\text{out}} \leq V_{\text{OLMAX}}$ we get $f(V_{\text{out}}) \leq f(V_{\text{OLMAX}})$. When m of the transistors nm_1, $nm_2, \ldots, nm_n$ are conductive the output voltage must satisfy the condition $V_{\text{out}} \leq V_{\text{OLMAX}}$. Setting $\lambda = m$ in Eq. (1) we get

$$\frac{\beta_p}{\beta_n} \leq m \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OLMAX}} - V_{\text{OLMAX}}^2}{(V_{dd} + V_{tp})^2} \tag{3}$$

From (2) and (3) we get

$$(m-1) \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OHMIN}} - V_{\text{OHMIN}}^2}{(V_{dd} + V_{tp})^2} \leq \frac{\beta_p}{\beta_n} \leq m \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OLMAX}} - V_{\text{OLMAX}}^2}{(V_{dd} + V_{tp})^2} \tag{4}$$

Taking into account that $\beta_p = KP_p \cdot W_p/L_p$ and $\beta_n = KP_n \cdot W_n/L_n$, from relation (4) we get that in order to design an m-weight threshold circuit the transistor sizes must satisfy the following relation:

$$(m-1) \cdot \frac{KP_n}{KP_p} \cdot \frac{2(V_{dd} - V_{tn})V_{\text{OHMIN}} - V_{\text{OHMIN}}^2}{(V_{dd} + V_{tp})^2} \leq \frac{W}{L} \leq m \cdot \frac{KP_n}{KP_p} \cdot \frac{2(V_{dd} - V_{tn})V_{\text{OLMAX}} - V_{\text{OLMAX}}^2}{(V_{dd} + V_{tp})^2} \tag{5}$$

where $W = W_p/W_n$ and $L = L_p/L_n$.

Figure 2 presents an $m/(m+1)$ programmable weight threshold circuit. When the input I is equal to one, the circuit of Figure 2 behaves identically to the circuit of Figure 1, that is, it is an m-weight threshold circuit, while when the input I is equal to zero then the circuit of Figure 2 behaves as an $(m+1)$-weight threshold circuit. Relation (5) implies that the aspect ratios of the transistors pm_m, pm_1 and nm_1, $nm_2, \ldots, nm_n$ of the $m/(m+1)$ programmable weight threshold circuit (Fig. 2), which are W_{pm_m}/L_{pm_m}, W_{pm_1}/L_{pm_1} and W_{nm}/L_{nm} respectively, must satisfy the following relations:

$$(m-1) \cdot \frac{KP_n}{KP_p} \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OHMIN}} - V_{\text{OHMIN}}^2}{(V_{dd} + V_{tp})^2} \leq \left(\frac{W}{L}\right)_m \leq m \cdot \frac{KP_n}{KP_p} \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OLMAX}} - V_{\text{OLMAX}}^2}{(V_{dd} + V_{tp})^2} \tag{6}$$

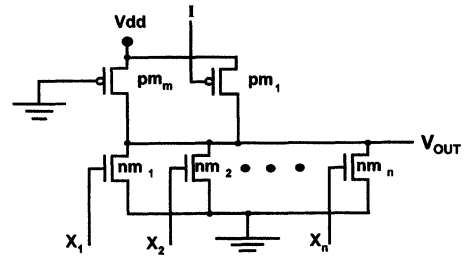

FIGURE 2 $m/(m+1)$ programmable weight threshold circuit.

and

$$m \cdot \frac{KP_n}{KP_p} \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OHMIN}} - V_{\text{OHMIN}}^2}{(V_{dd} + V_{tp})^2} \leq \left(\frac{W}{L}\right)_m + \left(\frac{W}{L}\right)_1 \leq (m+1) \cdot \frac{KP_n}{KP_p} \cdot \frac{2 \cdot (V_{dd} - V_{tn}) \cdot V_{\text{OLMAX}} - V_{\text{OLMAX}}^2}{(V_{dd} + V_{tp})^2} \quad (7)$$

where

$$\left(\frac{W}{L}\right)_m = \frac{W_{pm_m}}{L_{pm_m}} \cdot \frac{L_{nm}}{W_{nm}} \quad \text{and}$$

$$\left(\frac{W}{L}\right)_1 = \frac{W_{pm_1}}{L_{pm_1}} \cdot \frac{L_{nm}}{W_{nm}}$$

The proposed single output m-out-of-n code checker is shown in Figure 3. Module L is a $m/(m+1)$-weight threshold circuit. Specifically, when CLK = 0 the module behaves like an $m+1$-weight threshold circuit while when CLK = 1 it behaves like an m-weight. In this design the input CLK is driven by the system clock. We can easily verify that the design of Figure 3 is a single output m-out-of-n code checker. When the input vector is a code word and the checker is fault free then during a period of CLK the output Q gets the values (1, 0). When the input vector is not a code word then during a period the output Q gets the values (1, 1) or (0, 0). As in the case of the single output comparator given in [39] 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 [40]).

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 m-out-of-n code checker can be used only in systems with period greater than 2 $(t_d + t_s)$ (the same comment concerns the single output comparators given in [39]). 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 m-out-of-n code checker requires $n+2$ transistors, where n 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 transistors required for its implementation.

A similar design for single-output TSC code checkers, but only for $m=1$, was proposed in [41]. In the design given in [41] transistors pm_m and pm_1 of Figure 3, have been substituted by one transistor with resistance equivalent to the sum of the resistances of pm_m and pm_1.

A slightly different checker is shown in Figure 4. This design has two outputs (Q_0, Q_1) which

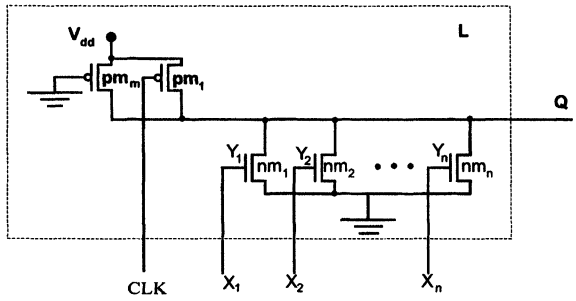

FIGURE 3 Single output m-out-of-n checker.

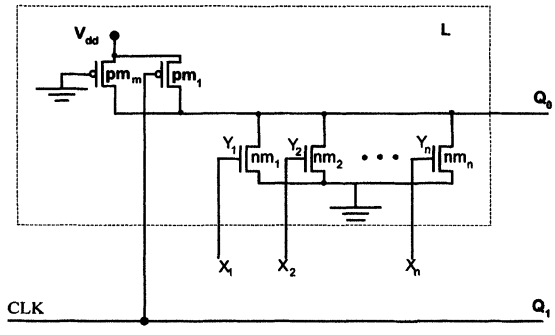

FIGURE 4 Double output m-out-of-n checker with clock as one output.

are both two rail encoded in time. The frequency of the clock signal CLK should be equal to the frequency of the system clock. When the checker receives a code word under error free operation, the outputs (Q_0, Q_1) get the values (1, 0) and (0, 1) for CLK = 0 and 1 respectively. The outputs Q_0, Q_1 of the checker are sampled twice per period of the system clock. The limitations concerning the single output checker of Figure 3 are valid also for this double output checker.

The double output m-out-of-n code checker is given in Figure 5. Module L_0 as well as module L_1 is a $m/(m+1)$-weight threshold circuit and they are both similar to module L of Figure 3. We can easily see that for $I=0$ the module L_0 behaves like an m-threshold circuit and the module L_1 as an $(m+1)$-threshold circuit, while for $I=1$ we have the opposite. The input I is driven by a clock signal with the half frequency of feeding inputs to the checker, which is usually equal to the frequency of the system clock. Thus 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 I. W denotes the Hamming weight (number of ones) of the vector $(X_1, X_2, \ldots, X_n)$.

The circuits of Figures 3–5 have obviously the code disjoint property, because for each m-out-of-n encoded input, they produce the 2-rail encoded outputs and for each non code word input they produce a non-2-rail encoded output.

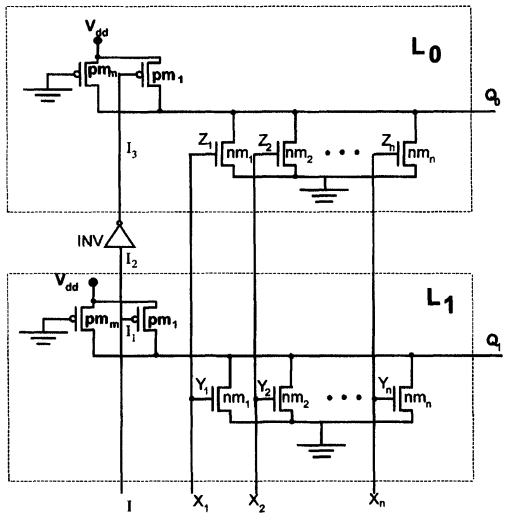

FIGURE 5 Double output m-out-of-n checker.

TABLE I Operation of the circuit of Figure 5

I	Weight of input vector (W)	Output Q_0	Output Q_1
0	$W < m$	1	1
	$W = m$	0	1
	$W > m$	0	0
1	$W < m$	1	1
	$W = m$	1	0
	$W > m$	0	0

The manufacturability of the proposed checkers depends on the manufacturability of the m-weight threshold circuit. The m-weight threshold circuit is a ratioed circuit. A problem of a ratioed circuit is that its correct operation depends on the conductance values of $n$mos and $p$mos 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-weight threshold circuit we choose the values of W_p, L_p, W_n and L_n so that the value of W/L lies in the middle of the range given by relation (5). 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/L will remain within the range, therefore the manufactured IC will operate correctly. As the value of m becomes greater the range defined by relation (5) becomes shorter and the yield of the manufacturing process will become smaller. With the improvement of the manufacturing process the circuit parameters deviation becomes smaller and m-weight threshold circuits for larger values of m can be constructed. In this point we have to note that for m-out-of-n codes with $m \geq \lceil n/2 \rceil$ we can use a checker for the $(n-m)$-out-of-n code simply by inverting the outputs of the functional circuit. We have run Monte Carlo simulations for circuit parameter deviations up to 10% and verified the correct operation of the m-out-of-n code checkers for $m < 5$. Apart from the above we have verified that for all cases the noise margins are above 0.7 volts.

Evidently the value of m depends on the technology used for manufacturing the circuit. In the case of deep or very deep submicron technology it may be necessary to limit m to 3 or 2. We have to note that, independently of the used technology, for $m = 1$ the checkers can always be manufactured.

III. TESTABILITY ANALYSIS

In the following we prove firstly that the proposed checkers are self-testing for single stuck-at, transistor stuck-on and transistor stuck-open faults. The testability analysis has also been verified with extensive simulations.

III.1. Single-output Checker

We use the notation $Q_{0,1} = Z, W$ which means that in a clock period, when CLK = 0, we have $Q = Z$ and when CLK = 1, $Q = W$. When the checker receives m-out-of-n code words and is fault free we get $Q_{0,1} = 0, 1$. When the output of the checker is $Q_{0,1} = 0, 0$ or $Q_{0,1} = 1, 1$ and its input is an m-out-of-n code word a fault has occurred in the checker itself.

a. X_i stuck-at 0 or transistor nm_i stuck-open with, $i \in [1, n]$. When the checker receives a code word with $X_i = 1$ then $Q_{0,1} = 1, 1$.
b. X_i stuck-at 1 or transistor nm_i stuck-on, with $i \in [1, n]$. When the checker receives a code word with $X_i = 0$ then $Q_{0,1} = 0, 0$.
c. CLK stuck-at 0 or transistor pm_1 stuck-on. When the checker receives a code word then $Q_{0,1} = 1, 1$.
d. CLK stuck-at 1 or transistor pm_1 stuck-open. When the checker receives a code word then $Q_{0,1} = 0, 0$.
e. Transistor pm_m stuck-on. This 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.
f. Transistor pm_m stuck-open. When the checker receives a code word then $Q_{0,1} = 0, 0$.
g. Q stuck-at 0. When the checker receives a code word then $Q_{0,1} = 0, 0$.
h. Q stuck-at 1. When the checker receives a code word then $Q_{0,1} = 1, 1$.

The above analysis implies that the test set consists of the m-out-of-n code words that apply both 0 and 1 values at each input X_i, $i \in [1, n]$, of the checker. We need $\lceil n/m \rceil$ code words to apply the value 1 at each input, and $\lceil n/(n-m) \rceil$ code words to apply the value 0 at each input. We can easily see that applying max $\{\lceil n/m \rceil, \lceil n/(n-m) \rceil\}$ specific code words each input receives both values, therefore the test set consists of max $\{\lceil n/m \rceil, \lceil n/(n-m) \rceil\}$ code words.

III.2. Double-output Checker

The testability analysis of the checker of Figure 4 is similar to that of the checker of Figure 3. The only one difference is a stuck at 0 or 1 fault on line CLK which can be detected using a checker for periodic signals [37]. In the sequel we will consider the testability of the double-output code checker of Figure 5.

1. Faults affecting both modules L_0, L_1.

   Such faults are only stuck-at faults at the primary inputs $X_1, X_2, \ldots, X_n$ and line I.

   a. X_i stuck-at 0, with $i \in [1, n]$. When the checker receives a code word with $X_i = 1$ then $Q_0 = Q_1 = 1$.
   b. X_i stuck-at 1, with $i \in [1, n]$. When the checker receives a code word with $X_i = 0$ then $Q_0 = Q_1 = 0$.
   c. Line I stuck-at 0 or 1. These faults are detected with a checker for periodic signals [37].

2. Faults affecting only module L_0.

   a. Line Z_i stuck-at 0 or transistor nm_i stuck-open, with $i \in [1, n]$. When the checker receives a code word with $X_i = 1$ and $I = 0$ then $Q_0 = Q_1 = 1$.

b. Line Z_i stuck-at 1 or transistor nm_i stuck-on, with $i \in [1, n]$. When the checker receives a code word with $X_i = 0$ and $I = 1$ then $Q_0 = Q_1 = 0$.
c. Line I_2 stuck-at 0. When the checker receives a code word and $I = 1$ then $Q_0 = Q_1 = 0$.
d. Line I_2 stuck-at 1. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 1$.
e. Line I_3 stuck-at 0 or transistor pm_1 stuck-on. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 1$.
f. Line I_3 stuck-at 1 or transistor pm_1 stuck-open. When the checker receives a code word and $I = 1$ then $Q_0 = Q_1 = 0$.
g. Line Q_0 stuck-at 0. When the checker receives a code word and $I = 1$ then $Q_0 = Q_1 = 0$.
h. Line Q_0 stuck-at 1. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 1$.
i. Transistor pm_m stuck-open. When the checker receives a code word and $I = 1$ then $Q_0 = Q_1 = 0$.
j. Transistor pm_m stuck-on. This 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.
k. Transistor *n*mos of the inverter INV stuck on. We construct the inverter with *n*-dominate logic so when the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 1$.
l. Transistor *p*mos of the inverter INV stuck-on. This fault is undetectable, but does not affect the operation of the circuit, the circuit remains code disjoint. Furthermore, if this fault is followed by a detectable fault, the resulting fault is detectable.
m. Transistor *n*mos of the inverter INV stuck-open. When the checker receives two successive code words with $I = 0$ and 1 respectively, then the second code word will give $Q_0 = Q_1 = 0$.
n. Transistor *p*mos of the inverter INV stuck-open. When the checker receives two successive code words with $I = 1$ and 0 respectively then at the second code word we have $Q_0 = Q_1 = 1$.

Since the frequency of the clock signal driving input I is equal to half of the frequency of feeding input vectors to the checker, for two successive inputs we have $I = 0$ and $I = 1$ respectively or $I = 1$ and $I = 0$ respectively. This is totally different to the case that pairs of test vectors are needed to test stuck-open faults in [30–33].

3. Faults affecting only module L_1.

a. Line Y_i stuck-at 0 or transistor nm_i stuck-open, with $i \in [1, n]$. When the checker receives a code word with $X_i = 1$ and $I = 1$ then $Q_0 = Q_1 = 1$.
b. Line Y_i stuck-at 1 or transistor nm_i stuck-on, with $i \in [1, n]$. When the checker receives a code word with $X_i = 0$ and $I = 0$ then $Q_0 = Q_1 = 0$.
c. Line I_1 stuck-at 0 or transistor pm_1 stuck-on. When the checkers receives a code word and $I = 1$ then $Q_0 = Q_1 = 1$.
d. Line I_1 stuck-at 1 or transistor pm_1 stuck-open. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 0$.
e. Line Q_1 stuck-at 0. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 1$.
f. Line Q_1 stuck-at 1. When the checker receives a code word and $I = 1$ then $Q_0 = Q_1 = 0$.
g. Transistor pm_m stuck-open. When the checker receives a code word and $I = 0$ then $Q_0 = Q_1 = 0$.
h. Transistor pm_m stuck-on. This 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.

It is evident that the test set TS of the double output checker consists of the union of the test sets TS_0, TS_1 of modules L_0 and L_1 respectively. These two test sets are obviously equivalent and consist of $\max \{\lceil n/m \rceil, \lceil n/(n-m) \rceil\}$ code words, as shown in the testability analysis of the single output checker, which must be applied to the checker for both values 0, 1 of input I. Therefore, the test sets TS_0, TS_1 and consequently the test set TS consist of $2 \max \{\lceil n/m \rceil, \lceil n/(n-m) \rceil\}$ code words.

III.3. Resistive Bridging and Break Faults

The self-checking capability of the proposed designs 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 $R \in [0, R_{max}]$ are detected, where R_{max} depends on the sizing of the transistors. For the checker of Figure 5 and an implementation in 1 μm technology with transistor aspect ratios $(W/L)pm_m = 2/1$, $(W/L)pm_1 = 2/1$ and $(W/L)nm_i = 1/1$, for $i = 1$ to n, the value of R_{max} for the various RBFs are given in Table II. During the simulation the inputs of the checker are driven by standard cell inverters with aspect ratios $(W/L)p = 12$ and $(W/L)n = 6$. The data of Table II are also valid for the checkers of Figures 3 and 4 with the difference that in these checkers the value of R_{max} for a resistive bridging fault between the gate and source of the transistor pm_1 is equal to 0.9 KΩ.

The proposed checkers are Self Testing for all break faults on device terminals except a break on the gate terminal of the transistor pm_m in module L (single output checker) and in modules L_0 and L_1 (double output checkers). This break does not affect the operation of the circuits and the circuits remain code disjoint. Furthermore, if this fault is followed by a detectable fault, the resulting fault is detectable.

It is very easy to verify that for any of the above considered faults the output of the checker is the correct code output or a non-code word, therefore the checker is fault secure.

TABLE II Resistive bridging faults

Transistor	Drain–Source	Gate–Drain	Gate–Source
nm_i	16 k	16 k	3.4 k
pm_m	18 k	16 k	–
pm_1	18 k	16 k	2.8 k

Bridging resistance between two inputs: 3 k.

IV. COMPARISONS

This is the first method for designing TSC m-out-of-n code checkers for most practical values of m and n, that takes into account a realistic fault model.

The checkers proposed in [13–28] take into account only stuck-at faults thus they are unsuitable for CMOS VLSI implementations. The PLA design given in [34] is valid for incomplete m-out-of-n codes and 1-out-of-n codes. Checkers only for m-out-of-2m codes taking into account apart from stuck-at faults, stuck-open faults too, were proposed in [30–32]. Their test set includes a large number of code input pairs, that increases with the value of m. For example for the 3-out-of-6 (6-out-of-12) code, the checkers given in [30–32] require 103(7,364), 34(500) and 16(49) code input pairs respectively, while the proposed single-output m-out-of-2m code checkers require 2 vectors and the double-output m-out-of-2m code checkers require 4. The large number of code input pairs, as we have explained in the introduction, reduces significantly the probability that the TSC goal will be achieved by these checkers.

Checkers for some m-out-of-2m codes under realistic faults were recently given in [33]. The checkers given in [33] have significantly greater area overhead and delay than the checkers proposed here. For example the 6-out-of-12 code checker given in [33] require 786 transistors and its delay is equal to 23.67 ns in 0.8 μm implementation.

For the special case of 1-out-of-n codes, TSC checkers for realistic faults were proposed in [35]. As it is shown in Table III the checkers designed

TABLE III Comparisons

	Checker	Area reduction	Delay reduction	Average power consumption reduction
Single Output	1-out-of-8	67%	45%*	74%
	1-out-of-16	65%	58%*	71%
Double Output	1-out-of-8	19%	59%	47%
	1-out-of-16	35%	66%	42%

*The setup time of the flip flop has been considered equal to zero.

following the proposed method compare favorably to the checkers given in [35], achieving significant reductions with respect to area, delay and average power consumption.

V. CONCLUSION

In this paper we presented a new systematic method for designing TSC checkers for m-out-of-n codes including the 1-out-of-3 case. The checkers designed according to the proposed method have many benefits. They are TSC with respect to realistic faults: stuck at, transistor stuck-on, transistor stuck-open, resistive bridging and break faults, the probability of achieving the TSC goal is greater than in other checkers and they are very compact and fast. The only undesirable characteristic of the proposed checkers is that they exhibit static power consumption. However, it was shown in [35] that the power consumption of the full static CMOS 1-out-of-n code checkers compensates the static power consumption of the checkers proposed in [35] above a frequency threshold of operation. Taking into account that the power consumption of the checkers proposed in this paper is significantly lower (from 42% up to 74%) than the power consumption of the checkers given in [35] (Tab. III) we conclude that this frequency threshold is even lower.

References

[1] Rao, T. R. N. and Fujiwara, E., "*Error control coding for computer systems*", Englewood Cliffs, NJ: Prentice Hall, 1989.

[2] Iacopini, M. J. and Vail, D. K., "The fault tolerance approach of the advanced architecture on-board processor", In: *Dig. Papers 19th Int. FTCS*, Chicago, IL, 21–23 July, 1989, pp. 6–12.

[3] Slegel, T. J. and Veracca, R. J., "Design and Performance of the IBM enterprise system/9000 type 9121 vector facility", *IBM J. Res. Develop.*, **35**, 367–381, May, 1991.

[4] Carter, W. C. and Schneider, P. R., "Design of dynamically checked computers", *IFIP Congress*, **2**, 878–883.

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

[6] Nicolaidis, M. and Courtois, B., "Strongly Code Disjoint Checkers", *IEEE Trans. Comput.*, **37**, 751–756, June, 1988.

[7] Tamir, Y. and Sequin, C. H., "Design and Application of Self-testing Comparators Implemented with MOS PLAs", *IEEE Trans. Comput.*, **C-33**, 493–506, June, 1984.

[8] Nicolaidis, M., "Fault Secure Property *Versus* Strongly Code Disjoint Checkers", *IEEE Trans. on CAD*, **13**(5), 651–658, May, 1994.

[9] Pradhan, D. K. and Stiffler, J. I., "Error correcting codes and self-checking circuits in fault tolerant computers", *Computer*, pp. 27–37, March, 1980.

[10] Leiss, E. L., "Data Integrity in Digital Optical Disks", *IEEE Trans. Computers*, **33**(9), 818–827, September, 1984.

[11] Fujiwara, E. and Pradhan, D. K., "Error-control Coding in Computers", *Computer*, **23**, 63–72, July, 1990.

[12] Freiman, C. V., "Optimal Error Detection Codes for Completely Asymetric Binary Channels", *Information Control*, **5**, 64–71, March, 1962.

[13] Anderson, D. A. and Metze, G., "Design of totally self-checking check circuits for m-out-of-n codes", *IEEE Trans. Computers*, **22**, 263–269, March, 1973.

[14] Smith, J. E., "The design of totally self-checking check circuits for a class of unordered codes", *J. Design Automation Fault-tolerant Computing*, **2**, 321–342, October, 1977.

[15] Marouf, M. A. and Friedman, A. D., "Efficient design of self-checking checkers for any m-out-of-n code", *IEEE Trans. Computers*, **27**, 482–490, June, 1978.

[16] Gaitanis, N. and Halatsis, C., "A new design method for m-out-of-n TSC checkers", *IEEE Trans. Computers*, **32**, 273–283, March, 1983.

[17] Efstathiou, C. and Halatsis, C., "Modular realization of totally self-checking checkers for m-out-of-n codes", *Proc. 13th FTCS*, Milan, pp. 154–161, June, 1983.

[18] Piestrak, S., "Design method of totally self-checking checkers for m-out-of-n codes", *Proc. 13th FTCS*, Milan, pp. 162–168, June, 1983.

[19] Nanya, T. and Tohma, Y., "A 3-level realization of totally self-checking checkers for m-out-of-n codes", *Proc. 13th FTCS*, Milan, pp. 173–176, June, 1983.

[20] Efstathiou, C. and Halatsis, C., "Efficient modular design of m-out-of-$2m$ TSC checkers, for $m = 2^k - 1$, $k > 2$", *Electronics Letters*, **21**, 1083–1084, November, 1985.

[21] Paschalis, A., Nikolos, D. and Halatsis, C., "Efficient modular design of TSC checkers for m-out-of-$2m$ codes", *IEEE Trans. Computers*, **37**, 301–309, March, 1988.

[22] Piestrak, S., "The minimal test-set for sorting networks, and the use of sorting networks in self-testing checkers for unordered codes", *Proc. 20th FTSC*, pp. 457–464, June, 1990.

[23] Paschalis, A. M., Efstathiou, C. and Halatsis, C. (1990). "An efficient TSC 1-out-of-3 code checker", *IEEE Trans.*, **C-39**, 407–411.

[24] Lo, J. and Thanawastien, S. (1990). "On the design of combinational totally self-checking 1-out-of-3 code checkers", *IEEE Trans.*, **C-39**, 387–393.

[25] Haniotakis, T. N., Pashalis, A. M. and Nikolos, D., "Fast and low-cost TSC checkers for 1-out-of-n and $(n-1)$-out-of-n codes in MOS transistor implementations", *Int. J. Electronics*, **71**, 781–791, November, 1991.

[26] Tao, D. L., Hartmann, C. R. P. and Lala, P. K. (1992). "A general technique for designing totally self-checking checker for 1-out-of-n code with minimum gate delay", *IEEE Trans. Computers*, **C-41**, 881–886.

[27] Dimakopoulos, V. V., Sourtziotis, G., Paschalis, A. and Nikolos, D., "On TSC Checkers for *m*-out-of-*n* codes", *IEEE Trans. Computers*, Vol. 44, No. 8, August, 1995.
[28] Piestrak, S. J., "Modular Design of Self-testing checkers for *m*-out-of-*n* codes, *2nd IEEE Int. On-line Testing Workshop*, Biarritz France, 8–10 July, 1996, pp. 132–135.
[29] Shen, J., Maly, W. and Ferguson, F., "Inductive fault analysis of MOS integrated circuits", *IEEE Design and Test*, December, 1985, pp. 26–33.
[30] Kundu, S. and Reddy, S. M. (1989). "Design of TSC checkers for implementation in CMOS technology", *Proc. of Inter. Conference on Computer Design*, pp. 116–119.
[31] Manjunath, S. and Radhakrishnan, D. (1991). "Efficient design of CMOS TSC checkers", *International Journal of Electronics*, **71**(1), 67–79.
[32] Th., Haniotakis and Paschalis, A., "Efficient Structured Design of Robustly Testable CMOS TSC *m*-out-of-2*m* code checkers", *Proc. of 1st IEEE On-line Testing Workshop*, Nice, France, 4–6 July, 1995, pp. 233–237.
[33] Lala, P., Busaba, F. and Zhao, M., "Transistor-level Implementation of Totally Self-Checking Checkers for a Subset *m* out of 2*m* codes", *Proc. of 2nd IEEE Int. On-line Testing Workshop*, 8–10 July, 1996, pp. 124–131.
[34] Piestrak, S. J. (1996). "Design of Minimal-level PLA Self-testing Checkers for *m*-out-of-*n* codes", *IEEE Trans. on VLSI Systems*, **4**(2), 264–272.
[35] Metra, C., Favalli, M. and Ricco, B., "Novel 1-out-of-*n* CMOS checker", *Electronic Letters*, 18 August, 1994, **30**(17), 1398–1400.
[36] Kavousianos, X. and Nikolos, D., "Self-exercising, Self-testing *k*-order Comparators", *Proc. of 15th IEEE VLSI Test Symposium*, Monterey, California, USA, 27 April–1 May, May, 1997, pp. 216–221.
[37] Usas, A. M., "A Totally Self-Checking Checker Design for the Detection of Errors in Periodic Signals", *IEEE Trans. Comput.*, May, 1975, **C-24**(5), 483–488.
[38] Metra, C. and Lo, J. C., "Compact and High Speed Berger Code Checker", *Proc. of 2nd IEEE Int. On-line Testing Workshop*, Biarritz, France, 8–10 July, 1996, pp. 144–149.
[39] Metra, C., Favalli, M. and Ricco, B., "Highly Testable and Compact Single Output Comparator", *Proc. of 15th IEEE VLSI Test Symposium*, 27 April–1 May, 1997, Monterey California, pp. 210–215.
[40] Afghahi, M. and Yuan, J., "Double edge triggered D flip flop for high speed CMOS circuits", *IEEE J. of Solid State Circuit*, **SC-26**, 1168–1170, August, 1991.
[41] Metra, C., Favalli, M. and Ricco, B., "Highly Testable and Compact 1-out-of-*n* code Checker with Single Output", *Proc. of DATE*, 23–26 February, 1998, Paris, France, pp. 981, 982.

Authors' Biographies

Xrysovalantis I. Kavousianos is currently pursuing his Ph.D. at the University of Patras, Greece in the area of On-line Testing. He holds a diploma degree in Computer Engineering from the same Department. In 1997 he was awarded a scholarship of the Technical Chamber of Greece due to his excellent student records. He is also a student member of IEEE. His other research interests include VLSI design and Low-power Testing.

Dimitris Nikolos received the B. Sc. degree in physics in 1979, the M. Sc. degree in electronics in 1981, and the Ph.D. degree in computer science in 1985, all from the University of Athens. Dr. Nikolos is currently full professor in the Department of Computer Engineering and Informatics at the University of Patras and head of the Technology and Computer Architecture Laboratory. He served as program co-chairman of the last three IEEE International On-line Testing Workshops. He also served on the program committees for the IEEE International On-line Testing Workshop in 1995 and 1996, for the 1997, 1998 and 1999 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems and for the Third European Dependable Computing Conference. His main research interests are fault-tolerant computing, computer architecture, VLSI design, test, and design for testability. He is a member of the IEEE.

George I. Sidiropoulos is currently pursuing his M. Sc. at the University of Patras, Greece. He received the Diploma in Computer Engineering from the University of Patras, Greece in 1998. From 7/1999 he is a post-graduate scholar of the Computer Technology Institute of Greece. His other research interests include VLSI design and On-line Testing.