High Speed and Highly Testable Parallel Two-Rail Code Checker

Martin Omaña Daniele Rossi Cecilia Metra
D.E.I.S. University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

Abstract

In this article we propose a high speed and highly testable parallel two-rail code checker, which features a compact structure and is Totally-Self-Checking or Strongly Code-Disjoint with respect to a wide set of realistic faults. The proposed checker is also particularly suitable to implement embedded two-rail code checkers, as it requires only two input codewords for fault detection. Our checker can be employed to check the correct operation of a connected functional block using the two-rail code, to implement the output two-rail code checker of "normal" checkers for unordered codes, or to join together the error messages produced by various checkers (possibly using different codes) present within the same self-checking system. The behavior of our checker has been verified by means of electrical level simulations (performed using HSPICE), considering both nominal values and statistical variations of electrical parameters.

1. Introduction

Self-Checking Circuits (SCCs) [2] have always been largely used in systems for high reliability applications (like transport, space, avionics, medicine, etc.), to allow the on-line testing of both permanent and transient faults. Because of the recognized relevance of transient faults for any electronic component implemented using very deep-submicron technology [1], SCCs are todays' attracting a significant interest in an ever increasing set of application fields.

They consist of a functional block (F), whose outputs are directly connected to a checker (C). In particular, F is designed in order to produce output words belonging to a chosen error detecting code, when it is fault-free, and words not belonging to the code, in case of internal faults. Instead, C should verify whether or not the words produced by F belong to the chosen error detecting code, giving an output error message when this is not the case. Such a behavior must be ensured despite the occurrence of faults within C (i), or an error indication must be given by C in case of internal faults (ii). Requirements (i) and (ii) have been formalized by the Strongly Code-Disjoint (SCD) [13] and Totally Self-Checking (TSC) [2] properties, respectively, which ensure that the checker behaves correctly from the whole SCC point of view.

Consequently, a significant effort has been devoted to the design of TSC or SCD checkers for various kinds of error detecting codes, which satisfy either the TSC or the SCD property with respect to a set of realistic faults as wider as possible.

Among existing error detecting codes, the two-rail one (whose codewords consist of bits that are two-by-two complementary) is frequently used in practical applications. Two-rail code checkers (TRCs) verifying the correctness of an input $2n$-bit word, generally referred to as n-variable two-rail code checkers (TRC_n), are usually designed as a tree of 2-variable two-rail code checkers (TRC_2).

In order to guarantee that a tree-structured TRC_n satisfies the TSC property with respect to possible stuck-at faults, it is generally necessary that all possible four codewords of the 2-variable two-rail code are applied to the inputs of each component TRC_2. This condition may be not satisfied in practical applications, where the TRC_n is embedded with the considered SCC. Additionally, this condition may significantly constraint the design of checkers for different kinds of codes which present a TRC as output stage, as it is generally the case for "normal" checkers for unordered codes [3, 18, 7, 12]. Consequently, specific design techniques have been proposed to deal with the derived problems (e.g., those in [8, 17, 14]). Moreover, a possible non-tree structured single output TRC_n which is suitable to implement embedded $TRCs$ was proposed in [6]. It presents a 3 level structure, including the series of n pMOS and nMOS transistors which, for high values of n, might significantly slow down the checker operation. A faster, more compact non-tree structured TRC, which is also suitable to implement embedded checkers, has been recently proposed in [9]. However, because of the parasitic capacitance it involves, also this checker may present some limitations in terms of operating frequency.

A novel TRC is here proposed which is faster than those in [6] and [9], while featuring nearly the same advantages as [9] in terms of area overhead over previous solutions.

1530-1591/03 $17.00 © 2003 IEEE

Additionally, similarly to the checkers in [9] and [6], it requires only two input codewords to satisfy the TSC or SCD property with respect to a wide set of realistic faults, thus being suitable to implement embedded $TRCs$. Finally, it is suitable to implement an error indicator, by simply adding a few feedback transistors.

This article is organized as follows. In Section 2, we describe the proposed n-variable two-rail code checker. In Section 3, we report some results of the electrical level simulations that we have performed to verify its behavior. In Section 4, we analyze the circuit self-checking ability and its possible use for embedded TRCs' implementation. In section 5, we report the costs of the proposed circuit and we compare them with those of alternative solutions. In Section 6, we briefly describe how to implement an error indicator based on the proposed checker. Finally, some conclusive remarks are drawn in Section 7.

2. The Proposed Two-Rail Code Checker

The electrical structure of the proposed n-variable two-rail code checker is shown in Fig. 1, where: (i) CK denotes the system clock signal; (ii) X_i and Y_i (i=1, ..., n) are the checker inputs which, in case they are two-railed, satisfy the condition $X_i = \overline{Y_i}$; (iii) O_1 and O_2 are the checker internal nodes which, in the fault-free case, assume alternating complementary logic values in each clock semi-period, thus guaranteeing that the checker is TSC with respect to possible output stuck-at faults [6, 9]. As can be seen, it consists in two identical subcircuits, each one evaluating half of the inputs. One subcircuit is driven by the CK signal, while the other by $\overline{CK}$, so that nodes O_1 and O_2 always carry complementary values in the fault-free case.

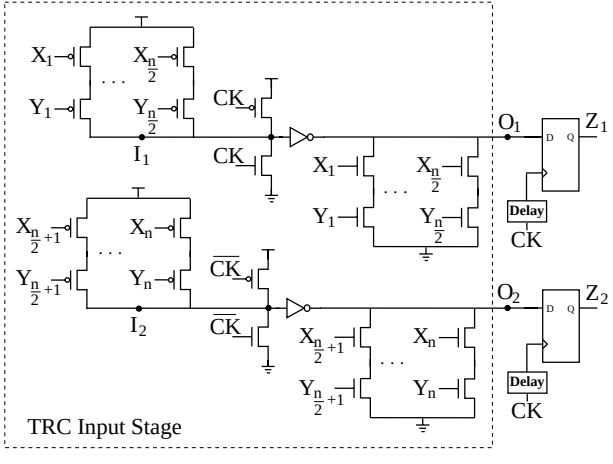

Figure 1. Electrical structure of the proposed n-variable two-rail code checker (TRC).

The inputs of the checker are supposed to be synchronous, or additional flip-flops are inserted at the checker inputs. Moreover, the series of two n- or p-channel transistors driven by the input bits are supposed to be more conductive (*dominant*) than the circuit single transistors driven by the clock signal and those composing the inverters.

The checker nodes O_1 and O_2 are sampled on both the clock rising and falling edges by means of two flip-flops. These flip-flops are triggered by a signal equal to CK, but delayed with respect to it of a suitably chosen time interval (taking into account the checker input/output delay, and the flip-flop setup time). As regards the flip-flop implementation, we have modified the circuit presented in [19] as shown in Fig. 2(a), so that it is triggered on both the rising and falling edges, as illustrated in Fig. 2(b).

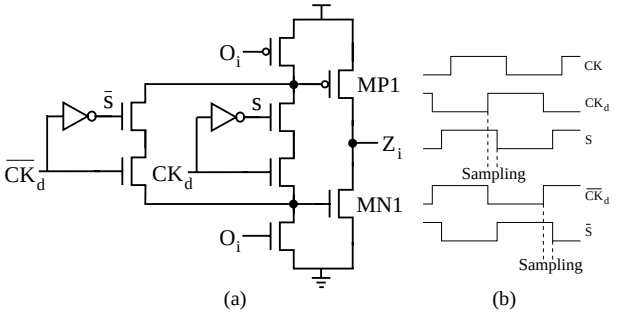

Figure 2. (a) Electrical structure of the double-edge triggered flip flop D used at the checker output. (b) Flip-flop sampling instants.

Indications of correct operations are recognized if alternately Z_1 and Z_2 are complemented at the end of each clock semi-period. Otherwise, the presence of an error indication is recognized. It is easy to verify that, in the fault-free case: 1) when CK=0, low and high logic values are given to the circuit outputs Z_1 and Z_2, respectively; 2) when CK=1, a high logic value is given on Z_1 and a low logic value on Z_2.

If a non two-rail word is present on the checker input, a wide range of possibilities might occur. Let us denote by N_{00} and N_{11} the number of (X_i, Y_i) couples equal to (00) and (11), respectively, within the same non-two-railed input word. In particular, the following cases of non-two-railed input words might occur: (i) $N_{00} > 0$ and $N_{11} = 0$; (ii) $N_{00} = 0$ and $N_{11} > 0$; (iii) $N_{00} > 0$ and $N_{11} > 0$ (included the case with $N_{00} = N_{11}$).

In case (i), one or more series of two p-channel transistors (in one or both subcircuits, depending on the input non code-word) begin to conduct. Therefore, an electrical conflict with the CK (or $\overline{CK}$) driven transistors is originated in one of the clock semi-periods. This electrical conflict is won by the two p-channel conductive transistors driven by the input bits, which force the outputs Z_1 and/or Z_2 to remain at a low logic value, thus producing an output error

indication during such a clock semi-period.

Similarly, in case (ii), one or more series of two n-channel transistors (in one or both subcircuits) begin to conduct, producing an electrical conflict with one inverter transistor in one of the clock semi-periods. Again, this electrical conflict is won by the n-channel conductive transistors driven by the input bits, and the output Z_1 and/or Z_2 remain at a low logic value, thus producing an error indication in such a clock semi-period.

Case (iii) is a combination of the previous cases. In particular, one or more series of two p-channel transistors begin to conduct, generating an electrical conflict with the CK or $\overline{CK}$ driven transistors. As in case (i), this electrical conflict is won by the p-channel conductive transistors driven by the input bits, forcing the nodes I_1 and/or I_2 to remain at a high logic value, which produces a low logic value at inverters outputs (nodes O_1 and O_2). Furthermore, these values are confirmed by the series of two n-channel transistors that also begin to conduct. Therefore, no conflict is generated between these transistors.

It should be also noted that, similarly to [6, 9], as the outputs assume both a low and a high logic value within a clock period under fault-free operations, both stuck-at-0 and stuck-at-1 faults affecting the checker outputs can be detected. To fulfill this purpose, similarly to [6, 9], the checker outputs should be sampled twice within a clock period. Alternatively, our proposed checker could be modified as shown in section 6, in order to provide, upon the generation of an error indication on nodes (O_1, O_2), an output error indication which remains latched until activation of a proper reset signal.

From the power consumption point of view, this checker has static power dissipation only when an output error message is given, which is expected to be an unlikely event.

The circuit also constitutes a relatively small additional load to the clock signal, which can be neglected compared with conventional loads applied to the system clock.

3. VLSI Implementation and Verification

The proposed n-variable two-rail code checker has been implemented considering, as an example, the case of n=32 and a standard 0.18μm CMOS technology with 2.5 V power supply. Furthermore, the following transistor aspect ratios have been considered: (i) $(W/L)_n = 1$ and $(W/L)_p = 2$, for the transistors driven by the CK and $\overline{CK}$ signals and for the inverter transistors; (ii) $(W/L)_n = 2.5$ and $(W/L)_p = 7.5$, for the transistors driven by the checker input bits. In particular, as for transistors in (ii), we have chosen the minimun aspect ratio that makes them dominant with respect to the transistors in (i), thus reaching the maximun checker speed, while guaranteeing the described behavior.

We have verified the operation of our checker by means of conventional and Monte Carlo electrical simulations, performed by HSPICE, considering possible statistical variations (with uniform distribution) of electrical parameters up to the 20%.

As an example, Fig. 3 shows the results obtained obtained in the case of a Monte Carlo simulation, considering the presence of both codeword and non-codeword inputs. In particular, the considered non-codewords are (X_1Y_1, ..., $X_{15}Y_{15}$, ..., $X_{32}Y_{32}$) = (10, ..., 11, ..., 01) and (10, ..., 00, ..., 01). The checker outputs Z_1 and Z_2 have complementary logic values in each clock semi-period, when a codeword is applied to the input, while the checker provides an output error indication in case of an input non-codeword.

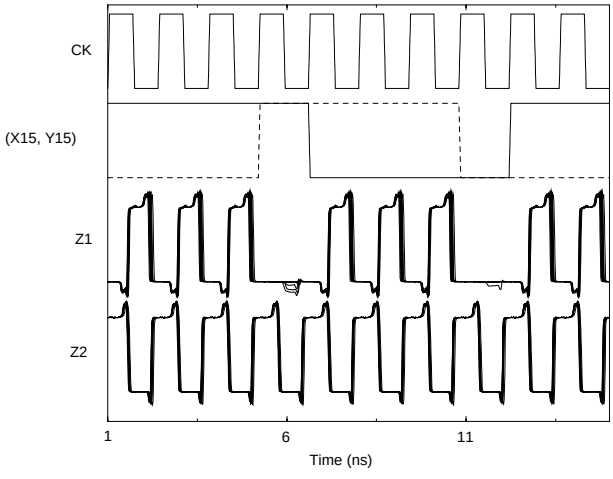

Figure 3. Results of the Monte Carlo simulation of the proposed checker, for the case of input codewords and non-codewords.

As will be described in Section 6, our checker can be easily modified to produce an output indication that remains latched until a proper reset signal is activated.

4. Self-Checking Ability

In order to evaluate the self-checking ability of the proposed checker, a set of faults ($\mathcal{F}$), representative of realistic failures, connecting resistance (R) in the interval $[0, 6k\Omega]$ [5]; 5) transient faults affecting the circuit nodes.

Moreover, the following, conventional fault hypotheses have been assumed [16]: 1) faults occur one at a time; 2) the time between the occurrence of two successive faults is long enough to allow the application of all possible input codewords.

Finally it is worth noticing that our checker can detect stuck-at faults affecting the clock signal. As for different kinds of faults affecting the clock signal and resulting in

incorrect frequency, duty cycle or skews, extra circuitry of the kind in [10, 11] could be adopted.

Let us report the performed analyses in details.

4.1. Node stuck-at faults

Node stuck-at faults (SAs) might occur on: (i) the checker input nodes (X_i, Y_i), i=1, ...,n; (ii) the checker internal nodes I_1, I_2, O_1, and O_2; (iii) the CK and $\overline{CK}$ nodes; (iv) the sampling flip-flops.

(i) This case is equivalent to the application of a non codeword at the checker input. Under this condition, an error indication is given at the checker output. Thus the checker is Totally Self-Checking (TSC) [2] with respect to SAs of kind (i).

(ii) As these nodes assume alternating (and complementary with respect to the corresponding node in the other subcircuit) logic values in each clock semi-period, both SA(0/1) result in error indications at the checker output. Thus the checker is TSC with respect to SAs of kind (ii).

(iii) If CK (or $\overline{CK}$) is SA0 (SA1), node O_1 (or O_2) will be SA0 (SA1), thus an error indication will be given to the checker output. Therefore, the checker is TSC with respect to SAs of kind (iii).

(iv) As the flip flop inputs O_1 and O_2 assume alternating complementary logic values in each clock semi-period, all possible SAs on the flip-flops' internal nodes produce an error indication on the checker outputs (Z_1 and Z_2) before the next clock period. Similarly, if Z_1 or Z_2 is SA(0/1), an error indication is produced.

Consequently, the proposed n-input two-rail checker is TSC with respect to all its possible SAs.

4.2. Transistor stuck-on faults

Transistor stuck-on faults (SONs) might affect: (i) the transistors driven by the CK and $\overline{CK}$ signals, or those composing the inverter buffers; (ii) the transistors driven by the checker input bits (X_i, Y_i), i=1, ..., n; (iii) the transistors of the sampling flip-flops.

In the presence of SONs of kind (i), the output either results in an error indication, or the fault does not affect the checker behavior. This is also true if the SONs are followed by other faults in $\mathcal{F}$. Thus, the proposed checker satisfies the SCD property with respect to this kind of faults.

In the presence of faults of kind (ii), when an input code word attempts to turn OFF the faulty device (and turn ON the connected series transistor) an electrical conflict is generated, and an error indication is produced at the checker output. Hence the proposed checker is TSC with respect to SONs of kind (ii).

SONs of kind (iii) produce an error indication on the output, or do not affect the checker behavior.

Hence, the proposed n-input two-rail code checker is TSC or SCD with respect to all its possible SONs.

4.3. Transistor stuck-open faults

Transistor stuck-open faults (SOPs) might affect: (i) the transistors driven by CK and $\overline{CK}$, or those of the inverters; (ii) the transistors of the sampling flip-flops; (iii) the transistors driven by the checker input bits (X_i, Y_i), i=1, ..., n.

We have verified that SOPs of kind (i) produce an output error message independently of the input word, since the affected transistor cannot charge (if p-channel) or discharge (if n-channel) the nodes (I_i, O_i, or Z_i) that, under fault free conditions, should assume alternating logic values in each clock semi-period. Thus, the checker is TSC with respect to SOPs of this kind.

Similarly, in presence of SOPs of kind (ii), the affected transistor cannot charge (if p-channel) or discharge (if n-channel) the internal nodes of the flip-flop, which produces an output Z_i which does not assume alternating logic values in each clock semiperiod. Therefore, our checker is TSC with respect to SOPs of this kind.

Reversely, SOPs of kind (iii) never affect the checker behavior in case of two-railed input words. Thus, they cannot be detected and the checker is not ST with respect to them. Moreover, these faults might make the checker be not able to reveal the presence of a non code-word input. As stated in [1], SOPs are becoming of major concern in future sub-micron technologies, and therefore, a method to dectect these faults is necesary. In the case of our checker, these on-line undetectable SOPs could be off-line detected by applying non-two-railed words to the checker input. In particular, the required non-two-railed words are those featuring (X_i, Y_i) = (1 1) and (0 0), and (X_j, Y_j) = (0 1) or (1 0), for $j \neq i$, j=1, ..., n.

4.4. Resistive bridgings

All resistive bridgings (BFs) possibly involving input, output and internal nodes of the proposed checker have been considered.

Electrical level simulations have been performed by means of HSPICE for each considered BF, with connecting resistances in the interval $[0, 6k\Omega]$ [5, 15]. As an example, we have analyzed the case of a 32-input two-rail code checker, designed using the previously given transistor aspect ratios (Section 3).

We have found that the proposed checker is TSC for values of R lower than a maximal value (R_{MAX}) which, depending on the considered BF, varies from a minimum of $0.9k\Omega$ up to $6k\Omega$. For values of $R \geq R_{MAX}$, we have verified that our checker satisfies the SCD property.

4.5. Transient Faults

Transient faults (TFs), for instance due to α particles or cosmic radiations, might affect input nodes as well as internal and output nodes of our proposed circuit. A TF can temporarily affect the voltage value on a considered node. This effect can be filtered out by the logic gates which follow the affected node. We have verified that, in this case, the checker satisfies the SCD property.

If the TF propagates to the output node, depending on its duration with respect to the flip flop set up and hold times, the spurious value could be sampled. If so, as the fault affects only one of the two output nodes, it will be detected. Consequently the checker is TSC with respect to the considered fault. Otherwise, the transient fault is filtered out and, also for this case, we have verified that the SCD property is satisfied.

4.6. Possible use to implement embedded checkers

The proposed checker satisfies the SCD property independently of the particular two-railed code-word applied to its input. It also satisfies the TSC property independently of its input code-word for: TFs satisfing set up and hold times; BFs for $R < R_{MAX}$ and SAs affecting nodes I_i, O_i, Z_i ($i = 1, 2$) and internal nodes of the sampling flip-flops; SONs and SOPs affecting the transistors driven by the CK and $\overline{CK}$ signals, and those transistors of the inverters and flip-flops. As regars the rest of faults analyzed in section 4 (i.e. SAs, SONs and BFs affecting either the checker input nodes (X_i, Y_i) or the transistors they drive), to satisfy the TSC property the checker requires the application of only two two-railed input words, out of all the possible 2^n. In particular, such input code words must ensure that each (X_i, Y_i) couple (i = 1,...,n) assumes both (10) and (01) logic values. This condition can be satisfied by two n-input two-rail words, for instance $(X_1Y_1, \ldots, X_iY_i, \ldots, X_nY_n)$ = (10,...,10,...,10), (01,...,01,...,01). Therefore, the proposed checker requires only 2 two-railed words (out of all the possible 2^n ones) to satisfy the TSC property with respect to all faults analyzed in section 4, while, as for the SCD property, it is satisfied independently of its inputs. Thus our checker is suitable to implement also embedded checkers, similarly to the checkers presented in [6, 9].

5. Cost Evaluation

To evaluate the costs of our proposed checker, it has been compared with both that presented in [9] and the one in [6]. All checkers have been simulated by means of HSPICE considering the same 0.18μm CMOS technology and considering for each one its worst case input conditions (from the response time point of view).

We have verified that both our checker and the one proposed in [9] produce an error indication within one clock period; however, the checker proposed here has a response time approximately 4 times lower than that in [9]. This gain in terms of speed is due to the lower parasitic capacitance to be charged/discharged during normal operation. As regard the checker proposed in [6], we have verified that it requires almost 1.5 clock period to detect a faulty condition and, therefore, it is 50% slower than our proposed checker.

Considering power consumption, we have found that the proposed checker dissipates about 2.4 times less than the checker in [6], but 50% more than that in [9]. In addition, as regard the power-delay product, it is worth noticing that our checker presents an improvement of 2.67 times with respect to the checker in [9].

As for area overhead, the checker proposed here and the one in [9] require respectively $(4n + 24)$ and $(4n + 10)$ transistors (both including the flip-flops sampling the outputs), while that presented in [6] makes use of $(14n + 2)$ or $(10n + 2)$ transistors, depending on the XOR implementation. As n increases, our checker and that in [9] have comparable area overhead. Instead, compared to the checker in [6], our checker presents a reduction in terms of area occupation for $n \geq 4$. This reduction approaches the 60% as the number of inputs increase. As an example, for $n = 32$, we allow an area overhead reduction of the 53%.

6. Error Indicator Implementation

As previously introduced, our checker can also be easily modified to allow the memorization of a generated output error message, until an external reset signal is activated, thus implementing an error indicator, according to the feedbacked two-rail checker structure introduced in [4]. The derived circuit is shown in Fig. 4.

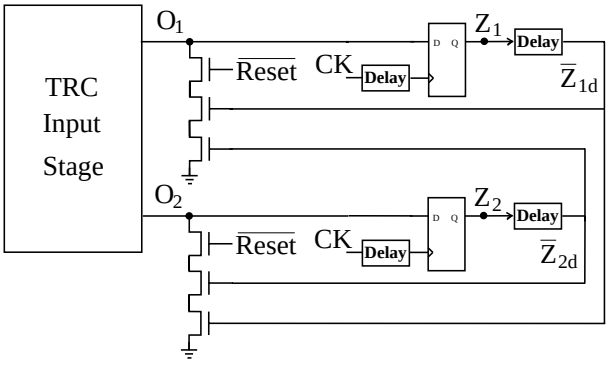

Figure 4. Error indicator derived from the proposed checker.

In the fault free case, both Z_1 and Z_2 have complementary logic values, hence at least one of the three feedback series transistors (including those driven by the $\overline{Reset}$ signal) remains turned off. Then, nodes O_1 and O_2 take the complementary logic values imposed by their respective inverter buffers. When a non codeword occurs, both outputs Z_1 and Z_2 go to a low logic value turning on both of the feedback transistors. Hence, if $\overline{Reset}$=1, nodes O_1 and O_2 hold a low logic value until the application of the reset signal. Fig. 5 shows the behavior of the error indicator in the presence of an input non code-word and subsequent activation of the reset signal.

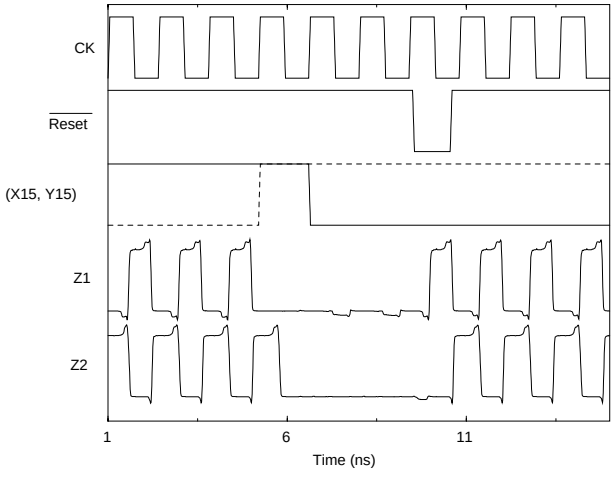

Figure 5. Example of the results of HSPICE simulation of our derived error indicator.

7. Conclusions

We have proposed a new parallel n-variable two-rail code checker that is TSC or SCD with respect to a wide set of realistic faults, including all possible node stuck-ats, transistor stuck-ons, resistive bridgings, transient faults and several transistor stuck-opens. Our checker satisfies the TSC or the SCD property with only two input codewords (out of all the 2^n possible ones), thus it is suitable to implement also embedded checkers.

Compared to other CMOS two-rail checkers recently proposed in literature, it features higher speed, a comparable (or lower) area overhead, with no drawbacks in terms of self-checking ability. Consequently, it is particularly suitable to be used for high performance self-checking systems.

Finally, we have shown how to implement an error indicator by slightly modifying the presented checker, which holds the error indication until the application of a proper reset signal.

References

[1] 2001 International Technology Roadmap for Semiconductors. *http://public.itrs.net/*.

[2] D. A. Anderson and G. Metze. Design of Totally Self-Checking Circuits for m-Out-of-n Codes. *IEEE Trans. Comput.*, c-22:263 – 269, March 1973.

[3] M. J. Ashjaee and S. M. Reddy. On totally self-checking checkers for separable codes. *IEEE Trans. Comput.*, C-26:737 – 744, August 1977.

[4] N. Gaitanis, D. Gizopoulos, A. Paschalis, and P. Kostarakis. An Asynchronous Totally Self-Checking Two-Rail Code Error Indicator. In *Proc. of IEEE VLSI Test Symp.*, pages 151 – 156, 1996.

[5] H. Hao and E. J. McCluskey. Resistive Shorts Within CMOS Gates. In *Proc. of IEEE Int. Test Conf.*, pages 292 – 301, 1991.

[6] 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.

[7] J.-C. Lo and S. Thanawastien. The Design of Fast Totally Self-Checking Berger Code Checkers Based on Berger Code Partitioning. In *Proc. of Int. Symp. Fault-Tolerant Comput.*, pages 226 – 231, 1988.

[8] C. Metra, M. Favalli, and B. Riccò. Embedded Two-Rail Checkers with On-Line Testing Ability. In *Proc. of IEEE VLSI Test Symp.*, pages 145 – 150, 1996.

[9] C. Metra, M. Favalli, and B. Riccò. Highly Testable and Compact Single Output Comparator. In *Proc. of IEEE VLSI Test Symp.*, pages 210 – 215, 1997.

[10] C. Metra, M. Favalli, and B. Riccò. On-Line Testing Scheme for Clocks' Faults. In *Proc. of IEEE Int. Test Conf.*, pages 587 – 596, 1997.

[11] C. Metra, M. Favalli, and B. Riccò. Concurrent Checking of Clock Signal Correctness. *IEEE Design & Test of Computers*, pages 42 – 48, October - December 1998.

[12] C. Metra and J.-C. Lo. Compact and High Speed Berger Code Checker. In *Proc. of 2nd IEEE Int. On-Line Testing Work.*, pages 144 – 149, 1996.

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

[14] D. Nikolos. Optimal Self-Testing Embedded Two-Rail Checkers. In *Proc. of 2nd IEEE Int. On-Line Testing Work.*, pages 154 – 161, 1996.

[15] R. Rodriguez-Montanes, E. M. J. G. Bruls, and J. Figueras. Bridging Defect Resistance Measurements in a CMOS Process. In *Proc. of IEEE Int. Test Conf.*, pages 892 – 899, 1992.

[16] J. E. Smith and G. Metze. Strongly fault-secure logic networks. *IEEE Trans. Comput.*, C-27:491 – 499, June 1978.

[17] S. Tarnick. Embedded Parity and Two-Rail TSC Checkers with Error Memorizing Capability. In *Proc. of 1st IEEE Int. On-Line Testing Work.*, pages 221 – 225, 1995.

[18] J. F. Wakerly. Detection of unidirectional multiple errors using low-cost arithmetic codes. *IEEE Trans. Comput.*, C-24:210 – 212, February 1975.

[19] P.-H. Yang, J.-S. Wang, and Y.-M. Wang. A 1GHz Low-Power Transposition Memory Using New Pulse-Clocked D Flip-Flops. *Proc. of IEEE Int. Symp. on Circuit And Systems*, 5:665 – 668, May 2000.