Ultra Fast and Low Cost Parallel Two-Rail Code Checker Targeting High Fan-In Applications *

S. Matakias[1], Y. Tsiatouhas[2], Th. Haniotakis[3] and A. Arapoyanni[1]

[1]University of Athens, Dept. of Informatics & Telecom., Panepistimioupolis, 15784 Athens, Greece
[2]University of Ioannina, Dept. of Computer Science, P.O. Box 1186, 45110 Ioannina, Greece
[3]Southern Illinois University, Dept. of Electrical & Computer Engineering, 62901 Carbondale, USA
e-mails: {s.matakias, arapoyanni}@di.uoa.gr, tsiatouhas@cs.uoi.gr, haniotak@siu.edu

Abstract

A novel current mode, periodic outputs, parallel two-rail code (TRC) checker is presented. The proposed topology is suitable for the implementation of high n-variable (high fan-in) two-rail code checkers targeting applications with a high count of monitoring lines. Compared to previous solutions the new checker is characterised by high operating frequencies and low silicon area requirements while maintaining lower power operation at higher n-variable values.

1. Introduction

Technology evolution results in an increased demand for high reliability. 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 operation. A SCC consists of a functional circuit whose outputs are monitored by a checker. The functional circuit will be called Circuit Under Monitoring (CUM). The CUM is designed to provide output codewords that belong to an error detecting code in the fault free case and non codewords in the presence of a fault. The checker produces an error indication signal whenever the CUM produces a non codeword output. This must be ensured despite the occurance of faults in the checker. Moreover, in case of internal faults, the checker must also provide an error indication. The above requirements are formalized by the *Totally Self-Checking* [2] and the S*trongly Code-Disjoint* [3] properties.

The class of two-rail code (TRC) checkers [1] is exploited to check the correctness of input words with *n* pairs of two-railed bits [4-8]. In that case they are called *n*-variable TRC (TRC_n) checkers. Usually *n*-variable TRC checkers, with $n>2$, are implemented as a tree of 2-variable TRC checkers. For large *n*-variable values the performance and the required silicon area of tree TRC checkers make them a non attractive solution for today applications. Recently, non-tree TRC checkers with periodic outputs have been proposed in [9-11]. The main disadvantage of these checkers is that in high *n*-variable TRC checker implementations they present a considerable degradation of their speed performance and increased requirements in silicon area.

In this paper we present a high speed and low cost, periodic and parallel TRC checker (or equality checker), suitable for very high *n*-variable implementations. The checker is based on current mode design techniques like these proposed in [12] for concurrent error detection and it is capable to provide very short response times with extremely low silicon area requirements. Moreover, like in [9-11], the checker requires only two input code words (out of a wide variety of equivalent pairs) to satisfy the TSC and SCD property for exactly the same, enhanced set of realistic faults as in these works.

The manuscript is organized as follows. In Section 2 the structure of the proposed in this work TRC checker is presented and its operation is analyzed by proving in parallel its code-disjoint property. Next, in Section 3 the self-checking property of the circuit is discussed for a wide set of realistic faults. Section 4 deals with design issues and simulation results are presented to verify the operation of the checker. In addition, comparisons with the checker proposed in [11] are carried out to illustrate the efficiency of the new design. Finally, in Section 5 the conclusions are given.

2. The new two-rail code checker

In general a Circuit Under Monitoring (CUM), that is monitored by a TRC checker, is designed to produce two-railed output words (X_j, Y_j, $j\in[1,..,n]$) in the fault-free ($X_j=\overline{Y_j}$) and non two-railed output words ($X_j=Y_j$) in case of internal faults. The TRC checker verifies whether the output words of a CUM are two-railed or not, providing the two-railed or not respectively output indication signals F and G.

The proposed *n*-variable pairs (X_j, Y_j, $j\in[1,..,n]$) TRC checker is presented in Fig. 1. It is divided into two sub-blocks, the FSB and GSB. The first sub-block is fed by half of the *inputs* (X_r, Y_r, $r\in[1,..,k]$, where $k=n/2$) and the second by the rest of the checker inputs (X_s, Y_s, $s\in[k+1,..,n]$). Each pair of inputs (X_j, Y_j) drives a pair of serially connected pMOS transistors and a pair of serially connected nMOS transistors in the corresponding sub-block. Thus, there are n-pairs of transistors in each block connected in parallel between the V_{DD} power supply and the input terminal of a current mirror. Additional transistors (MFC and MGC) are also connected between the V_{DD} and the input terminal of the current mirror in each sub-block respectively. The MFC transistor of the FSB is driven by the complementary of the system clock signal *CLKB* while the MGC transistor of the GSB is driven by the clock signal *CLK*. The current mirrors are formed by nMOS transistors (MFM1 - MFM2 and MGM1 - MGM2). Finally, the pMOS transistors MFL and MGL are used as loads at the output terminals of the FSB and GSB respectively. The MFL transistor is driven by the *CLKB* signal and the MGL by the *CLK* signal.

* This research was partially funded by the European Commission and the Greek Ministry of Education through EPEAK II.

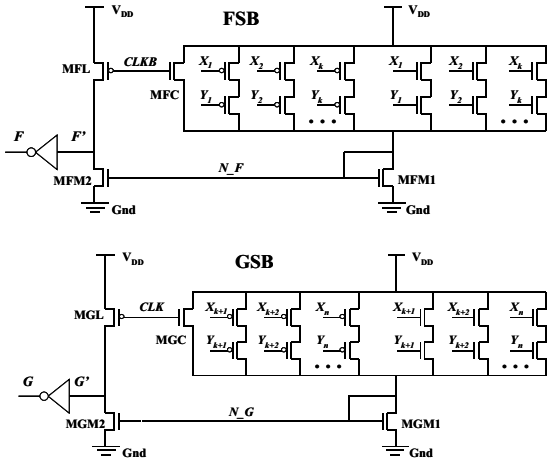

Figure 1. The proposed two-rail code checker

The checker's output nodes F and G always present complementary logic values in the fault free case and non-complementary in the opposite case providing the indication of the correct system operation or not. The checker operation is divided into two phases, transparent to the CUM, according to the clock *CLK* semi-periods. In the fault free case ($X_j = \overline{Y_j}$ $\forall$ j∈[1,..,n]) and for each phase the following stand: In the first semi-period, when *CLK*="1", the parallel connected pairs of transistors in the FSB and the transistor MFC are in the non conducting state and thus no current passes through the pertinent current mirror forcing F' to be charged to V_{DD} through MFL which is in the conducting state. However, the MGC transistor in the GSB is in the conducting state providing current to the input terminal of the corresponding current mirror. Since the MGL transistor is in the non-conducting state the mirrored current discharges node G' towards Gnd. Consequently F and G present complementary values ("low" and "high" respectively) in the first semi-period of the clock. In the second semi-period, when CLK="0", the MFC transistor in the FSB is in the conducting state providing current to the input terminal of the corresponding current mirror. Since the MFL transistor is in the non-conducting state node F' is discharged towards Gnd. On the contrary, the parallel connected pairs of transistors in the GSB and the transistor MGC are in the non conducting state and thus no current passes through the pertinent current mirror forcing G' to be charged to V_{DD} through MGL which is in the conducting state. The result is F and G to present complementary values ("high" and "low" respectively) also in the second semi-period of the clock. Thus, in the fault free case, nodes F and G are always in complementary states (two-railed) at the end of each clock semi-period.

In case that a non two-rail word is present at the inputs of the checker, at least one or more input pairs (X_j, Y_j) have equal values ($X_j = Y_j$). Three cases are observed: i) all the non two-rail input pairs feed the FSB, ii) all the non two-rail input pairs feed the GSB and iii) there are some non two-rail pairs that feed the FSB and other that feed the GSB. In case (i) there is at least one pair of serially connected transistors in FSB, driven by the non two-rail input pair that will be in the conducting state (either a pMOS pair when $X_j = Y_j$ = "0" or an nMOS pair when $X_j = Y_j$ = "1"). Thus, in the first semi-period of the clock there will be a current flow through the current mirror of the FSB which will discharge (or keep discharged) node F', since the current mirror is designed to be more conductive (dominant) over the load transistor MFL. Consequently, the F node turns "high" and since the GSB operation does not depend on the input values during the first semi-period, both F and G will be in the "high" state indicating the presence of the error. Similarly, in the second case (ii) there will be at least one pair of serially connected transistors in GSB that is driven by the non two-rail input pair and thus in the conducting state. As a result, in the second semi-period of the clock there will be a current flow through the current mirror of the GSB which will discharge (or keep discharged) node G', since also this current mirror is designed to be more conductive (dominant) over the load transistor MGL. Consequently, the G node turns "high" and since the FSB operation does not depend on the input values during the second semi-period, both F and G will be in the "high" state indicating the presence of the error. Finally, in case (iii) an erroneous pair of inputs affects both FSB and GSB. Consequently, according to (i) and (ii), F and G will be in the "high" state in both semi-periods of the clock indicating the presence of the errors.

3. The self-checking property.

In this section the self-checking property of the proposed checker is discussed with respect to a set of faults consisting of: 1) line stuck-at faults, 2) transistor stuck-on faults and 3) transistor stuck-open faults. The following two common assumptions in the checker design [2] have been taken into account: i) a single fault occurs at a time and ii) the time between two successive faults is enough to permit the application of all possible codewords.

3.1 Line stuck-at faults.

We can observe four cases of possible line stuck-at (SA) faults: a) SA faults on the input lines of the checker X_j, Y_j, j∈[1,..,n], b) SA faults on the checker outputs F and G, c) SA faults on the internal lines of the checker N_F, F', N_G and G' and d) *CLK* or *CLKB* signal lines SA faults.

a) SA faults on the input lines X_j, Y_j, j∈[1,..,n] of the checker are equivalent to non two-rail words on them. Therefore, the checker is TSC with respect to these faults.

b) Obviously, the checker is TSC considering SA faults on the output lines F or G.

c) A SA "0" or "1" fault on N_F (N_G) is equivalent to a SA "0" or "1" fault on the F (G) output of the checker. Similarly, a SA "0" or "1" fault on F' (G') will result in a SA "1" or "0" response on the output F (G). Consequently, according to (b), the checker is TSC for this kind of faults.

d) A SA "0" fault on *CLK* (*CLKB*) will result in a SA "0" ("1") value on line G (F). Moreover, a SA "1" fault on *CLK* (*CLKB*) will result in a SA "1" ("0") value on line G (F). Thus, the checker is Totally Self-Checking (TSC) with respect to these faults.

3.2 Transistor stuck-open faults.

Transistor stuck-open (TSOP) faults can be categorised into three groups: a) TSOP faults on the transistors that are driven by the checker inputs (X_j, Y_j, j∈[1,..,n]), b) TSOP faults on the

transistors that are driven by the clock signals *CLK* and *CLKB* and c) TSOP faults on the transistors of the current mirrors.

a) The TSOP faults of the first kind are not sensitised by input codewords and thus they are not detectable. Consequently, the checker is not self-testing with respect to these faults but remains fault-secure. In addition, there are also non codeword inputs that are not detectable by the checker in the presence of this kind of faults. Note, that the periodic output TRC checker presented in [11] is also non self-testing as the proposed one for the same number and type of transistors (those driven by the checker inputs). As it has been proposed in [11], it could be possible to detect these faults off-line by applying the proper non two-rail words. The necessary non two-rail words are those where $(X_i, Y_i) = (1, 1)$ and then (0, 0) while $\forall$ j ≠ i it stands that $(X_j, Y_j) = (0, 1)$ or (1, 0), with i, j∈[1,..,n]. Another approach is to use an additional BIST structure to detect these faults as it is recommended in [13].

b) In the case of a TSOP fault that affects the MFC (MGC) transistor of the FSB (GSB), it will result in a "low" response at the F (G) output of the checker during the second (first) semi-period of the clock. Thus, this fault is detectable. The same stands when the transistor MFL (MGL) is affected by a TSOP fault. The output F (G) of the checker is "high" during the first (second) semi-period and thus the fault is detectable. Consequently, the checker is TSC with respect to these faults.

c) Finally, a TSOP fault in the MFM1 (MGM1) transistor will lead node N_F (N_G) to be permanently charged to V_{DD}. Thus, transistor MFM2 (MGM2) will be always in the conducting state turning F (G) to "high" during the first (second) semi-period of the clock. Consequently, the checker is TSC for this fault. Moreover, a TSOP fault on MFM2 (MGM2) will result in a "low" value at the F (G) output of the checker during the second (first) semi-period of the clock. Hence, the checker is also TSC for this fault.

3.3 Transistor stuck-on faults.

Similarly to the TSOP faults, transistor stuck-on (TSON) faults can be categorised into three groups: a) TSON faults on the transistors that are driven by the checker inputs (X_j, Y_j, j∈[1,..,n]), b) TSON faults on the transistors that are driven by the clock signals *CLK* and *CLKB* and c) TSON faults on the transistors of the current mirrors.

a) A TSON fault on a transistor of the FSB (GSB) that is driven by the checker inputs X_j or Y_j, j∈[1,..,n] is detectable since there exists an input codeword with $(X_j, Y_j) = (0, 1)$ or (1, 0) respectively to sensitise it. After the application of this codeword at the checker inputs there will be a current flow through the current mirror of the FSB (GSB) which will set node F' (G') to "low" during the first (second) semi-period of the clock, since, as we mentioned, the current mirror is designed to be more conductive (dominant) over the load transistor MFL (MGL). Thus node F (G) will be to a "high" state during this semi-period and the fault will be detected. Consequently, the checker is TSC for this kind of faults.

a) A TSON fault that affects the MFC (MGC) transistor of the FSB (GSB), will result, as in case (a), in a "high" response at the F (G) output of the checker during the first (second) semi-period of the clock. Thus, this fault is detectable and the checker is TSC with respect to this fault. However, a TSON fault on the MFL (MGL) transistor will either lead the output F (G) to "low" at the second (first) period of the clock or does not affect the checker behaviour. Hence, concerning this fault, the checker is either TSC or SCD.

b) The presence of a TSON fault on transistor MFM1 (MGM1) will prohibit any current flow through the left branch of the current mirror, that is transistor MFM2 (MGM2), resulting in a "low" value on the F (G) output of the checker during the second (first) semi-period of the clock. Thus, this kind of fault is detectable and the checker is TSC with respect to this fault. Moreover, a TSON fault on transistor MFM2 (MGM2) will lead to a "high" value at the output F (G) of the checker during the first (second) semi-period of the clock. Therefore, the checker is TSC for a TSON fault on transistor MFM2 (MGM2).

4. Design and simulation results

The proposed parallel two-rail code checker has been designed in a 0.18μm CMOS technology for a variety of *n*-variable values ranging from 8 to 256. The used power supply was 1.8V. The operation of our checker has been verified by SPICE simulations (see Fig. 2) in all possible technology corner conditions, for the fault free and all the possible faulty conditions in the checker according to the fault models discussed in Section 3, as well as for all possible types of erroneous input code words.

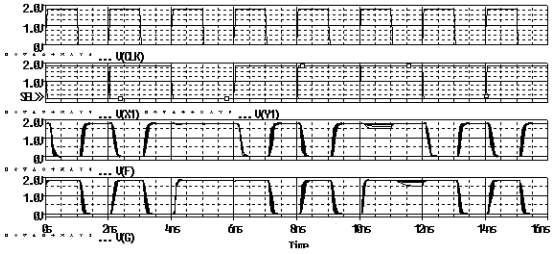

Figure 2. Simulation waveforms

As an example the special case, for today system architectures, of a 64-variable TRC checker is discussed. The sizes of the transistors used in this design were: i) W/L=0.5μm/0.18μm for the pMOS and W/L=0.28μm/0.18μm for the nMOS transistors that are driven by the inputs in the FSB and GSB sub-blocks, ii) W/L=1.6μm/0.2μm and W/L=5.0μm/0.2μm for the MFM1 (MGM1) and MFM2 (MGM2) transistors of the current mirrors respectively, iii) W/L=0.5μm/0.2μm for the load transistors MFL and MGL, and finally iv) W/L=1.24μm/0.18μm and W/L=0.7μm/0.18μm for the pMOS and nMOS transistors of the output inverters.

For comparison reasons, the checker presented in [11] has been also designed for the same range of *n*-variable alternatives. Both checkers have been optimized with respect to their response time in order to provide the minimum response delay in each case. In Table I design issues and simulation results are presented, for the proposed in this work topology and the topology introduced in [11]. Initially, in columns 2 and 3 the implementation cost in unit size transistors (UST) considering various *n*-variable values for each design, is presented. As implementation cost in UST we define the number of minimum size transistors, according to the used technology, that will cover the same area as the actual transistors in the design. Furthermore, in column 4 the cost reduction of our topology over the one in [11] is given. Next, comparisons based on simulation results between the two checkers are presented. The worst case response time (columns 5 and 6) and the power consumption in the fault free

case (columns 8 and 9) are shown. The corresponding reductions are provided in columns 7 and 10. According to Table I, the proposed in this work checker is superior over the checker in [11] with respect to the required silicon area and the response delay time, especially for high values of the *n*-variable. In addition, for high *n*-variable values, the new checker also turns to be more power efficient over the previous approach.

The performance of the proposed checker with respect to its response times is based on the current mode operation. The adopted current mirror topology is capable to provide a fast sensing of the current flow through the array of the parallel connected pairs of transistors without the need for a full voltage swing on the internal nodes (N_F and N_G) of the checker as in the case of [11]. This way fast response times can be achieved with low silicon requirements, especially in the case of high *n*-variable values, since there is no need to fully charge/discharge these internal nodes' parasitic capacitance. The only requirement is to provide a low current as input to the current mirrors of the checker. The reduced voltage swing on these nodes is also the reason of power savings in the case of large parasitic capacitances on these nodes (high *n*-variable values) despite the DC current path in each sub-block during the pertinent semi-period.

Finally, considering the noise sensitivity of the checker, note that its inputs are digital signals (full swing signals and not small analog signals) that are characterised by precisely defined noise margins. Since the checker does not need to detect small signal variations at its inputs it is easy to design it not to be sensitive to the input noise.

5. Conclusions

As the performance of digital systems increases with technology progress, the need for fast self-checking schemes becomes imperative. A checker is a critical unit in the design of a self-checking system, affecting both system's reliability and performance. Traditional TRC checkers are based on tree structures that are not capable to provide fast operating speeds. Recently, parallel TRC checkers have been proposed in the open literature but their speed performance is limited to small *n*-variable values.

In this paper we presented a high speed and low cost parallel TRC checker suitable for the implementation of high *n*-variable circuits. The new checker belongs to the periodic outputs category of TRC checkers and it is TSC or SCD for a wide set of realistic faults, without any drawbacks in the self-checking ability compared to other TRC checkers in the same category. Simulation results in a 0.18μm technology validate our checker operation, while comparisons with a recently presented high speed periodic outputs TRC checker demonstrate its efficiency.

6. References

[1] W.C. Carter and P.R. Schneider, "Design of Dynamically Checked Computers", Proc. of IFIP Congress, pp. 878-883, 1968.

[2] D.A. Anderson and G., Metze, "Design of Totally Self-Checking Circuits for m-out-of-n Codes", IEEE Trans. on Computers, vol. 22, pp. 263-269, 1973.

[3] M. Nicolaidis and B. Courtois, "Strongly Code Disjoint Checkers", IEEE Trans. on Computers, vol. 37, pp. 751-756, 1988.

[4] C. Efstathiou, "Efficient MOS Implementation of Totally Self-Checking Two-Rail Code Checkers", Int. Journal of Electronics, vol. 68, no. 2, pp. 259-264, 1990.

[5] S. Tarnick, "Embedded Parity and Two-Rail TSC Checkers with Error Memorizing Capability", 1st IEEE On-Line Testing Workshop, pp. 221-225, 1995.

[6] J. C. Lo, "A Novel Area-Time Efficient Static CMOS Totally Self-Checking Comparator", IEEE Journal of Solid-State Circuits, vol. 28, no. 2, pp. 165-168, 1993.

[7] C. Metra, M. Favali, B. Ricco, "Embedded Two-Rail Checkers with On-Line Testing Ability", IEEE VLSI Test Symposium, pp. 145-150, 1996.

[8] D. Nikolos, "Optimal Self-Testing Embedded Two-Rail Checkers", 2nd IEEE On-Line Testing Workshop, pp. 154-161, 1996.

[9] S. Kundu, E.S. Sogomonyan, M. Goessel and S. Tarnick, "Self-Checking Comparator with One Periodic Output", IEEE Trans. on Computers, vol. 45, no. 3, pp. 379-380, 1996.

[10] C. Metra, M. Favali, B. Ricco, "High Testable and Compact Single Output Comparator", IEEE VLSI Test Symposium, pp. 210-215, 1997.

[11] M. Omana, D. Rossi and C. Metra, "High Speed and Highly Testable Parallel Two-Rail Code Checker", Design Automation and Test in Europe Conference, pp. 608-613, 2003.

[12] Y. Tsiatouhas, A. Arapoyanni, D. Nikolos and Th. Haniotakis, "A Hierarchical Architecture for Concurrent Soft Error Detection Based on Current Sensing", 8th IEEE On-Line Testing Workshop, pp. 56-60, 2002.

[13] M. Nicolaidis, "Self-Exercising Checkers for Unified Built-In Self-Test (UBIST)", IEEE Trans. on Computer-Aided Design, vol. 8, pp. 203-218, 1989.

[14] D. Nikolos, "Optimal Self-Testing Embedded Parity Checkers", IEEE Trans. on Computers, vol. 47, no. 3, pp. 313-321, 1998.

Table I

Comparisons with respect to i) silicon area, ii) response delay time and iii) power consumption

Fan-In	Silicon Area Cost (UST)			Response Delay (ps)			Power Consumption (μW)		
	Proposed	[5]	Reduction	Proposed	[5]	Reduction	Proposed	[5]	Reduction
8	66	73	9.6%	193	160	-17.1%	51	7	-86.3%
16	96	140	31.4%	223	303	26.4%	82	12	-85.4%
32	154	304	49.3%	260	579	55.1%	96	26	-72.9%
64	262	1442	81.8%	303	888	65.9%	106	94	-11.3%
128	461	5757	91.9%	412	1575	73.8%	123	348	64.7%
256	859	22025	96.1%	590	2846	79.3%	161	1274	87.4%