A TOTALLY SELF-CHECKING 1-OUT-OF-3 CODE ERROR INDICATOR

A. Paschalis   N. Gaitanis   D. Gizopoulos   P. Kostarakis

Institute of Informatics & Telecommunications, NCSR “Demokritos”
Aghia Paraskevi, 15310 Athens, GREECE
E-mail: {paschali,nikos,dgizop,pkost}@iit.nrcps.ariadne-t.gr

Abstract

In this paper, an asynchronous TSC 1-out-of-3 (1/3) code error indicator is introduced that memorises erroneous 1/3 code inputs {000, 011, 101, 110, 111} with time duration greater than a discrimination time T. Such an error indicator is used to discriminate transient erroneous 1/3 code inputs from real ones as well as to detect not only faults that cause logical errors but also delay faults (short or long) that alter the circuit delay outside its specified limits (upper or lower bounds) without causing logical errors. To our knowledge, our error indicator is the first TSC 1/3 code error indicator proposed in the open literature.

I. Introduction

Due to the high complexity and density of VLSI chips the vast majority of errors are transient or intermittent. Concurrent error detection is mandatory for detecting such errors, mainly in critical applications (e.g. spatial, avionics, nuclear, industrial, military) since it provides fast detection and location of the fault preventing further corruption of the system. Thus, in designing highly reliable, available and maintainable systems the Totally Self-Checking (TSC) checkers play an important role as they are used for concurrent detection of errors caused by faults either in the functional circuit they monitor or in their circuitry.

The concept of the TSC checkers has been formally defined in [1] and [2] as the circuits which satisfy the self-testing, fault secure and code disjoint properties and usually generate a two-rail output with values 01 or 10 for normal indication and 00 or 11 for error indication. Therefore, during error-free operation, a TSC checker theoretically generates only normal indication states either 01 or 10. Unfortunately, in practice, the two-rail output (x_1,x_2) of a TSC checker may *transiently* become either 00 or 11 during one normal indication transition (either 01→10 or 10→01) due to different propagation delays of the paths associated with the outputs x_1 and x_2. Therefore, in the design of TSC systems, there is the requirement to discriminate *transient error indication states* from *real error indication* states caused by faults.

Apart from this, in the design of TSC systems, there is also the requirement to reduce the hard core in the alarm system that elaborates error indications, as well as, to maintain the brief appearance of an error indication once such an indication is reached.

The mentioned above requirements for TSC systems are satisfied by placing a TSC two-rail code error indicator appended to the outputs of a TSC checker. Such a TSC error indicator memorises error indications (00 or 11) generated by TSC checkers with time duration greater than a *discrimination time T* [3].

Apart from this, the most common failure mechanisms in digital VLSI circuits first manifest themselves as small delay faults which become progressively larger until a logical error appears [4]. Hence, during normal operation, a prompt detection of even small delay faults, sometimes before causing critical paths to fail, gain importance in critical applications. This is achieved by using TSC error indicators appended to TSC checkers [5]. Therefore, by using TSC error indicators in TSC systems we can detect concurrently either faults that cause logical errors or delay faults (short or long) that alter the circuit delay outside its specified limits (upper or lower bounds) without causing logical errors.

The paper deals with the design of an asynchronous TSC 1/3 code error indicator for first time in the open literature. The introduced TSC error indicator memorises erroneous 1/3 code inputs {000, 011, 101, 110, 111} with time duration greater than a *discrimination time* T and generates a two-rail output with values 01 or 10 for normal indication and 00 or 11 for error indication. Such an error indicator can be used to satisfy the mentioned above requirements for TSC systems, as well as, to detect faults that cause either logical errors or transitions (fast or late) that violate the discrimination time T.

Considering that a combinational TSC 1/3 code checker does not exist in gate level, (since the only three 1/3 code words can not detect all single stuck-at faults [6]), we conclude that the presented here circuit is not only a TSC 1/3 code checker in gate level, for first time, but also it is a fail-safe circuit since it maintains an error state once such a state is reached. Such a TSC error indicator can be designed with existing library cells.

ED&TC ’97 on CD-ROM

We remark that a TSC 1/3 code checker is alternatively implemented in gate level with two ways, that is, as a sequential circuit [7] or as a combinational circuit that combines the 1/3 code with a two-rail output belonging to an other TSC checker existing in the same circuit [8,9]. Apart from this, transistor level implementations with specific layout requirements have been proposed in n-MOS [10,11] and CMOS technology [12,13,14]. Such checkers can not be designed with existing library cells.

II. On asynchronous 1/3 code error indicators

Let us consider a sequential machine M=<I, S, f>, where I is the set of the 1/3 code inputs, S is the set of internal states and f is the next state function (i.e. f: I x S → S). The set of the 1/3 code inputs I is partitioned in the set of normal 1/3 code inputs I_N ={001, 010, 100} and the set of erroneous 1/3 code inputs I_E ={000, 011, 101, 110, 111}. The set of internal states S is partitioned in the set of normal states S_N ={01, 10} and the set of error states S_E = {00, 11}. A normal next state function f_N is defined for the error-free circuit while error next state functions f_E are defined for every error appearing after the occurrence of a fault in the circuit.

Definition 1: The sequential machine M=<I, S, f> is a *1/3 code error indicator* when the following three conditions are satisfied:

C1. For every normal 1/3 code input of set I_N and for both normal states of set S_N, the next state, derived from the function f_N, is also a specific normal state (01 or 10) so that the error indicator passes through both normal states {01, 10} during error-free operation after a specific number of 1/3 code input transitions.

C2. For every erroneous 1/3 code input of set I_E and for both normal states of set S_N, the next state, derived from the function f_N, is an error state of set S_E.

C3. For every either normal or erroneous 1/3 code input and for both error states of set S_E, the next state, derived from the function f_N, is also an error state of set S_E.

In what follows we concentrate our discussion in the asynchronous 1/3 code error indicator shown in Fig. 1. Such an error indicator consists of a combinational circuit C that realises the normal next state function f_N and a RESET circuit which is used to reset the error indicator in a normal state. C receives the 1/3 code input (x_1,x_2,x_3) and the current state two-rail input (y_1,y_2) which is the feedback of the next state two-rail output (y_1',y_2') through the linear delay element LD. LD delays the next state output (y_1',y_2') by D_d and determines the discrimination time T.

A fault-free asynchronous 1/3 code error indicator with discrimination time T operates as follows:

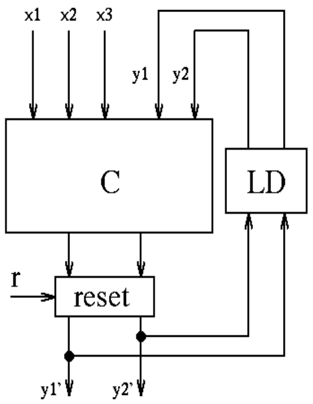

Fig. 1: The general structure of the asynchronous 1/3 code error indicator

- When either a normal 1/3 code input (of set I_N) or an erroneous 1/3 code input (of set I_E) with maximum time duration ≤T (classified as *transient*) appears at the inputs (x_1,x_2,x_3) of the error indicator being in a normal state (of set S_N), it is stabilised in a normal state (set S_N).
- When an erroneous 1/3 code input (of set I_E) with maximum time duration >T (classified as *real*) appears at the inputs of the error indicator being in a normal state (of set S_N), it is stabilised in an error state (set S_E).
- Once the error indicator is stabilised in an error state (of set S_E), it maintains an error state independently of the values at the inputs (x_1,x_2,x_3) until a reset signal is issued.

We can see easily that in case that T=0 conditions C1, C2 and C3 are satisfied.

An asynchronous 1/3 code error indicator is a TSC sequential circuit when it satisfies both the self-testing and fault-secure properties defined as follows:

Definition 2: An asynchronous 1/3 code error indicator is *self-testing* for a set of faults F if, for every fault in F, the circuit starting from any normal state in S_N is finally stabilised in an error state in S_E after at most k normal 1/3 code input transitions.

Definition 3: An asynchronous 1/3 code error indicator is *fault-secure* for a set of faults F if, for every fault in F, the circuit starting from any normal state of S_N is stabilised either in the correct normal state or in an error state of S_E, for every normal 1/3 code input transition.

We remark that the asynchronous TSC 1/3 code error indicator is a *fail-safe* sequential circuit [2] since it maintains an error state once such a state is reached.

An asynchronous TSC 1/3 code error indicator is evaluated from the following characteristics:

Sensitivity. It is derived from the minimum discrimination time that can be achieved.

Flexibility. It is derived from the capability to change easily the value of T by changing the delay element.

Testability. It is derived from the maximum number of normal 1/3 code transitions which detect every fault in F so that the self-testing property is satisfied.

Stability. It is derived from the stabilisation time D which is defined as follows.

Definition 4: In an asynchronous 1/3 code error indicator the *stabilisation time* D is the time needed for all internal signals to be stabilised.

In order to design an asynchronous 1/3 code error indicator we have to design the combinational circuit C that satisfies conditions **C1**, **C2** and **C3** and couple it with the linear delay element LD that satisfies the required discrimination time T. The design of C is based on Table 1 which shows a state table that satisfies conditions **C1**, **C2** and **C3**.

Table 1: The state table of the 1/3 code error indicator

$x_1x_2x_3$		I_N			I_E				
y_1y_2		001	010	100	000	011	101	110	111
S_N	01	01	10	01	00	11	11	11	11
	10	01	10	01	00	11	11	11	11
S_E	00	00	00	00	00	00	00	00	00
	11	11	11	11	00	11	11	11	11

From Table 1 the following irredundant normal next state function f_N is derived:

$$y_1^{'} = (x_2 + x_1x_3)(y_1 + y_2) + (x_1 + x_3)y_1y_2$$

$$y_2^{'} = (x_1 + x_3)(y_1 + y_2) + (x_2 + x_1x_3)y_1y_2$$

We can easily see that the combinational circuit C that realises straightforwardly the irredundant normal next state function is not testable for all single stuck-at faults during normal operation. Thus, we can not design a TSC asynchronous 1/3 code error indicator with respect to single stuck-at faults by using such a circuit C.

III. Design method

In order to design an asynchronous TSC 1/3 code error indicator with respect to all single stuck-at faults we have to find a new combinational circuit C* equivalent to the straightforward circuit C, which is testable for single stuck-at faults.

The introduced asynchronous TSC 1/3 code error indicator is shown in Fig. 2. The circuit C* of Fig. 2 consists of a translator from the 1/3 code into an incomplete two-variable two-rail code {(a_1,a_2), (b_1,b_2)} [9], as it is shown in Table 2, and three TSC two-rail code checkers (termed TRC1, TRC2 and TRC3, respectively).

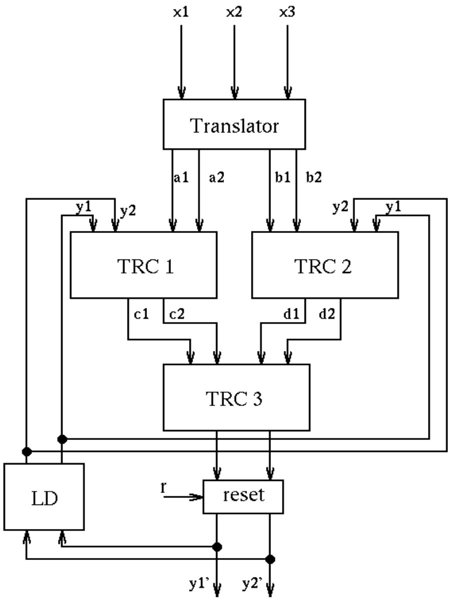

Fig. 2: The asynchronous TSC 1/3 code error indicator

TRC1 monitors the pair of two-rail outputs (a_1,a_2) and (y_1',y_2') and generates the two-rail output (c_1,c_2). TRC2 monitors the pair of two-rail outputs (b_1,b_2) and (y_2',y_1') and generates the two-rail output (d_1,d_2). TRC3 monitors the pair of two-rail outputs (c_1,c_2) and (d_1,d_2) and generates the next state output (y_1',y_2'). The two-rail outputs functions realised by circuit C* of Fig. 2 are:

- Transl.: $a_1 = x_1 + x_2$ $a_2 = x_3$ $b_1 = x_2 + x_3$ $b_2 = x_1$
- TRC1: $c_1 = y_1a_2 + y_2a_1$ $c_2 = y_1a_1 + y_2a_2$
- TRC2: $d_1 = y_1b_1 + y_2b_2$ $d_2 = y_1b_2 + y_2b_1$
- TRC3: $y_1^{'} = c_1d_2 + c_2d_1$ $y_2^{'} = c_1d_1 + c_2d_2$

First, we prove in Lemma 1 that circuit C* is equivalent to circuit C and thus circuit C* is an asynchronous 1/3 code error indicator.

Lemma 1: The proposed circuit C* of Fig. 2 is equivalent to circuit C of Fig. 1.

Proof: Function y_1' realised by the circuit of Fig. 2 is equivalent to the non-redundant form of the next state output y_1'. We have:

$$y_1^{'} = c_1d_2 + c_2d_1 =$$

$$= (y_1a_2 + y_2a_1)(y_1b_2 + y_2b_1) + (y_1a_1 + y_2a_2)(y_1b_1 + y_2b_2) =$$

$$= y_1a_2b_2 + y_1y_2a_2b_1 + y_1y_2a_1b_2 + y_2a_1b_1 +$$

$$+ y_1a_1b_1 + y_1y_2a_1b_2 + y_1y_2a_2b_1 + y_2a_2b_2 =$$

$$= (y_1 + y_2)a_2b_2 + (y_1 + y_2)a_1b_1 + y_1y_2(a_2b_1 + a_1b_2) =$$

$$= (y_1 + y_2)(a_1b_1 + a_2b_2) + y_1y_2(a_2b_1 + a_1b_2) =$$

$$= (y_1 + y_2)(x_1x_2 + x_1x_3 + x_2 + x_2x_3 + x_1x_3) +$$

$$+ y_1y_2(x_1 + x_1x_2 + x_2x_3 + x_3) \Leftrightarrow$$

$$y_1^{'} = (y_1 + y_2)(x_2 + x_1x_3) + y_1y_2(x_1 + x_3)$$

Also, function y_2' realised by the circuit of Fig. 2 is equivalent to the non-redundant form of the next state output y_2'. We have:

$$y_2' = c_1d_1 + c_2d_2 =$$
$$= (y_1a_2 + y_2a_1)(y_1b_1 + y_2b_2) + (y_1a_1 + y_2a_2)(y_1b_2 + y_2b_1) =$$
$$= y_1a_2b_1 + y_1y_2a_2b_2 + y_1y_2a_1b_1 + y_2a_1b_2 +$$
$$+ y_1a_1b_2 + y_1y_2a_1b_1 + y_1y_2a_2b_2 + y_2a_2b_1 =$$
$$= (y_1 + y_2)a_2b_1 + (y_1 + y_2)a_1b_2 + y_1y_2(a_1b_1 + a_2b_2) =$$
$$= (y_1 + y_2)(a_2b_1 + a_1b_2) + y_1y_2(a_1b_1 + a_2b_2) =$$
$$= (y_1 + y_2)(x_1 + x_1x_2 + x_2x_3 + x_3) +$$
$$+ y_1y_2(x_1x_2 + x_1x_3 + x_2 + x_2x_3 + x_1x_3) \Leftrightarrow$$
$$y_2' = (y_1 + y_2)(x_1 + x_3) + y_1y_2(x_2 + x_1x_3)$$

■

Then we prove that the proposed asynchronous 1/3 code error indicator is a TSC circuit. Let us consider the truth table of the proposed circuit for normal operation.

Table 2: The truth table of the asynchronous TSC error indicator for normal operation (s=stable, t=transient)

(x_1,x_2,x_3)	(a_1,a_2)	(y_1,y_2)	(b_1,b_2)	(y_2,y_1)	(c_1,c_2)	(d_1,d_2)	(y_1',y_2')	
001	01	01	10	10	01	01	01	s
001	01	10	10	01	10	10	01	t
010	10	10	10	01	01	10	10	s
010	10	01	10	10	10	01	10	t
100	10	01	01	10	10	10	01	s
100	10	10	01	01	01	01	01	t

Based on Table 2 the following corollaries are derived.

Corollary 1: Circuit C* has three *stable* and three *transient* input combinations of (x_1,x_2,x_3) and (y_1,y_2), as it is denoted in Table 2. Let us assume that the error indicator after receiving the normal 1/3 code input 001 is stabilised in the normal state 01. When the 1/3 code input transition 001→010 appears at the inputs, first the error indicator receives the transient input combination of (x_1,x_2,x_3)=010 and (y_1,y_2)=01 and it generates the other normal state 10, and then it receives the stable input combination of (x_1,x_2,x_3)=010 and (y_1,y_2)=10 and it is stabilised in the other normal state 10. Following this reasoning the maximum stabilisation time D is about $D = 2D_c + D_d + T$, where D_c is the propagation delay of circuit C*. This formula is also verified by experimental results (see Table 3).

Corollary 2: Circuit C* is testable for all single stuck-at faults when it receives all the six combinations of (x_1,x_2,x_3) and (y_1,y_2) shown in Table 2. When all the six combinations are applied during normal operation: a) the OR gates of the translator receive their test set (00,01,10) and b) all the three TSC two-rail code checkers receive all the four code inputs.

Corollary 3: Circuit C* is self-testing with respect to single stuck-at faults after three normal 1/3 code input transitions (k=3), for example, 010→001→010→100. During such a sequence of transitions TRC1 receives one transient code input (01,10) and three stable code inputs (01,01), (10,01) and (10,10); TRC2 receives one transient code input (01,01) and three stable code inputs (01,10), (10,01) and (10,10); TRC3 receives one transient code input (10,01) and three stable code inputs (01,01), (01,10) and (10,10). Note that the time duration of the transient code inputs increases when the value of the delay element D_d increases and is enough for fault detection, even for small values of D_d.

Also, circuit C* is fault-secure with respect to single stuck-at faults since such a fault either does not affect the correct operation of the circuit or causes the circuit to be stabilised in an error state {00, 11}.

Based on the above discussion we conclude that the circuit of Fig. 2 is an asynchronous TSC 1/3 code error indicator with respect to all single stuck-at faults.

Apart from this, according to [15,16], asynchronous TSC 1/3 code error indicators with respect to non-classical faults (bridging, transition and stuck-open faults inside TRCs) may be implemented in CMOS technology following specific rules in the design of TRCs. Moreover, all path delay faults that cause two-rail skew time greater than the discrimination time T at the next state two-rail output (y1',y2') are detected.

IV. VLSI implementation results

The error indicator was designed and simulated using the COMPASS Design Automation Framework and the 1ìm CMOS standard cells library [17]. The simulation results for the proposed error indicator are summarised in Table 3. In this table we can see for a range of delays D_d caused by the delay element LD, the corresponding discrimination time T and stabilisation time D. Circuit C* has a propagation delay of about 4 nsec.

Table 3: Simulation Results

Delay Element D_d (nsec)	Discrim. Time T (nsec)	Stabil. Time D (nsec)
3	0	13
4	1	14
5	2	15
6	2	16
7	3	17
8	4	18
9	5	19
10	5	20

From Table 3 we conclude the following:

Our TSC error indicator can be designed with discrimination time T near 0 nsec. Thus, it has the *greatest possible sensitivity*.

Our TSC error indicator is *flexible* by using a delay element at the feedback path with delay $D_d \geq 3$ nsec.

Let us consider that the asynchronous TSC 1/3 code error indicator is used to monitor the outputs of a TSC functional unit that generates a 1/3 code output in order to detect faults that cause logical errors as well as delay faults (short or long). Let us assume that all paths of the time-optimised functional unit have about the same propagation delay P, which determines the TSC system clock period. The gate delay variations within a chip are between 5% and 10% [18]. Taking into account gate delay variations of 10% due to manufacturing process variations, we classify as transient (not real) erroneous 1/3 code outputs the outputs with maximum time duration about P/10. Thus, for both manufacturing testing and concurrent delay testing of this functional unit we use our TSC error indicator with T = P/10. Thus, *the stabilisation time D of our error indicator is small enough so that it does not conflict with the TSC system clock period.* For example, if P=50 nsec we use a TSC error indicator with T=5 nsec and D=19 nsec.

V. Conclusions

We have introduced the design of an asynchronous TSC 1/3 code error indicator in gate level for first time in the literature. This error indicator is not only a TSC 1/3 code checker that can be easily implemented by using any existing library of standard gate cells, but also a fail-safe circuit that memorises erroneous 1/3 code with time duration greater than a discrimination time T. Such a design can be used additionally to discriminate transient erroneous 1/3 code inputs from real ones and to detect faults that cause either logical errors or transitions (fast or late) that violate the discrimination time T.

The existence of such a TSC error indicator opens new horizons in the design of fast TSC checkers with outputs encoded in the 1/3 code instead of the two-rail code.

Note that we can not design a TSC 1/3 code error indicator in gate level using the TSC 1/3 code checkers of [8,9] coupled with the TSC two-rail code error indicator of [3] since the 1/3 code output is combined with a two-rail output belonging to an other checker existing in the same circuit. Additionally, either the sequential TSC 1/3 code checker of [7] or the transistor level implementation dependent solutions of [10-14], coupled with the TSC error indicator of [3] require more hardware cost and have doubtful quality with regard to the characteristics of the TSC 1/3 code error indicator namely sensitivity, stability, flexibility and testability.

References

[1] D. A. Anderson, "Design of self-checking digital networks using coding techniques", Coordinated Science Lab., Rep. R-527, Univ. of Illinois, October 1971.

[2] F. Wakerly, "Error detecting codes, self-checking circuits and applications", Amsterdam, The Netherlands: Elsevier North-Holland, 1978.

[3] N. Gaitanis, D. Gizopoulos, A. Paschalis and P. Kostarakis, "An Asynchronous Totally Self-Checking Two-Rail Code Error Indicator", Proc. 14th IEEE VLSI Test Symposium, May 1996, pp. 151-156.

[4] P. Franco, E.J. McCluskey, "On-line delay testing of digital circuits", Proc. of the 12th IEEE VLSI Test Symposium, April 1994, pp. 167-173.

[5] A. Paschalis, D. Gizopoulos, N. Gaitanis, "Concurrent Delay Detection in Duplication Systems", Proc. 2nd IEEE Inter. On-Line Testing Workshop, July 1996, pp. 112-115.

[6] S. M. Reddy, "A note on self-checking checkers", IEEE Trans. Comput., vol. 23, October 1974, pp. 1100-1102.

[7] R. David, "A totally self-checking 1/3 checker", IEEE Trans. Comput., vol. 27, June 1978, pp. 570-572.

[8] P. Golan, "Design of totally self-checking checker for 1/3 code", IEEE Trans. Comput., vol. 33, March 1984, p. 285.

[9] A. M. Paschalis, C. Efstathiou, and C. Halatsis, "An efficient TSC 1/3 code checker", IEEE Trans. Comput., vol. 39, March 1990, pp. 407-411.

[10] D. L. Tao, P. K. Lala, and C. R. P. Hartmann, "A MOS implementation of totally self-checking checker for the 1/3 code", IEEE Trans. Comput., vol. 23, June 1988, pp. 875-877.

[11] J. Lo and S. Thanawastien, "On the design of combinational totally self-checking 1/3 code checkers", IEEE Trans. Comput., vol. 39, March 1990, pp. 387-393.

[12] K. Hirabayashi, "Self-checking CMOS circuits using pass-transistor logic", J. of Electronic Testing: Theory and Application, vol. 2, June 1991, pp. 205-208.

[13] C. Metra, M. Favalli, P. Olivo, and B. Ricco, "A highly testable 1/3 CMOS checker", Proc. IEEE Int. Workshop on Defect and Fault Tolerance in VLSI Systems, 1993, pp. 279-286.

[14] C. Metra, M. Favalli, and B. Ricco, "Novel 1-out-of-n CMOS checker", Electronics Letters, 18th August 1994, vol. 30, no. 17, pp. 1398-1400.

[15] S. D. Millman and E. J. McCluskey, "Bridging, transition, and stuck-open faults in self-testing CMOS checkers", Proc. 21th Fault Tolerant Computing Symposium, pp. 154-161, 1991.

[16] 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, February 1993.

[17] VLSI Technology INC., "1.0 Micron CMOS VSC370 Portable Library", Rev. 2, 1991.

[18] W. K. C. Lam and R. K. Brayton, "Timed Boolean Functions: A Unified Formalism for Exact Timing Analysis", Kluwer Academic Publishers, 1994.