VLSI DESIGN
2000, Vol. 11, No. 1, pp. 23–34
Reprints available directly from the publisher
Photocopying permitted by license only

Signal Coding and CMOS Gates for Combinational Functional Blocks of Very Deep Submicron Self-checking Circuits

CECILIA METRA[a,*], MICHELE FAVALLI[b,†] and BRUNO RICCÒ[a,‡]

[a]*DEIS – University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy;* [b] *University of Ferrara, Via Saragat 1, 44100 Ferrara, Italy*

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

In this paper we propose signal coding and CMOS gates that are suitable to self-checking circuits with combinational functional blocks implemented also by next generation, very deep submicron technology. In particular, our functional blocks satisfy the Strongly Fault-Secure property with respect to a wide set of possible, internal faults including not only conventional stuck-ats, but also transistor stuck-ons, transistor stuck-opens, resistive bridgings, delays, crosstalks and transient faults, that are very likely to affect next generation ICs. Compared to alternative, existing solutions, that proposed here does not imply any critical constraint on the circuit electrical parameters. Therefore, it is suitable to be adopted to design very deep submicron self-checking circuits which, compared to todays' circuits, will present significantly increased sensitivity to parameter variations occurring during fabrication.

Keywords: On-line testing, self-checking circuits, combinational functional blocks, very deep submicron technology, signal coding, CMOS gates

1. INTRODUCTION

Testing next generation very deep submicron ICs is currently recognized as a major problem and challenge. In fact, their reduced power supplies, capacitances, and noise margins will make them particularly prone and sensitive to faults occurring during system normal operation (in particular transients), in the past a major concern only for systems for applications with high reliability requirements (*e.g.*, space, avionic, *etc.*). On-line testing can guarantee the detection of faults occurring during circuit normal operation. Therefore, the use of on-line testing techniques is

*Corresponding author. Tel.: +39-051-2093785, Fax: +39-051-2093073, e-mail: cmetra@deis.unibo.it
†Tel.: +39-0532-293853, Fax: +39-0532-768602, e-mail: mfavalli@ing.unife.it; mfavalli@deis.unibo.it
‡e-mail: bricco@deis.unibo.it

currently emerging as a viable approach to the general testing problems of very deep submicron ICs' [27].

On-line testing is usually performed using Self-Checking Circuits (SCCs) [5]. A SCC (Fig.1) consists of a functional block (F) and a checker (C) directly connected.

To ensure that the SCC behaves correctly (*i.e.*, provides an output error message in case of incorrect data at the output of F), F must be designed to be Totally Self-Checking (TSC) [2] or Strongly Fault-Secure (SFS) [32], while C must be TSC or Strongly Code-Disjoint (SCD) [28].

Several works were dedicated to design rules and coding techniques making combinational [32, 6, 24, 25, 14, 31, 15, 3] and sequential [6, 12, 30, 16, 17] functional blocks SFS with respect to stuck-at faults. Instead, fewer papers [25, 14, 16, 17, 15] have so far accounted possible transistor stuck-ons (SONs), transistor stuck-opens (SOPs), resistive bridgings (BFs) and delay faults (DFs) which, however, are very likely to affect todays' deep submicron ICs and which, together with crosstalks (CFs) and transient faults (TFs), will be also much more likely for very deep submicron ICs.

In particular, as for combinational functional blocks, in [15] we have shown that, if SONs, SOPs, BFs and DFs are considered, the conventional SFS property may be insufficient to guarantee that F behaves correctly from the point of view of the whole SCC. This because, according to the conventional logical representation of data, the SFS property refers only to the logical correctness of the word produced at the output of F, without taking into account the analog and dynamic effects that these faults might produce on the functional block behavior that, eventually, might compromise the whole SCC correct operation. Of course, similar considerations apply also to CFs and TFs.

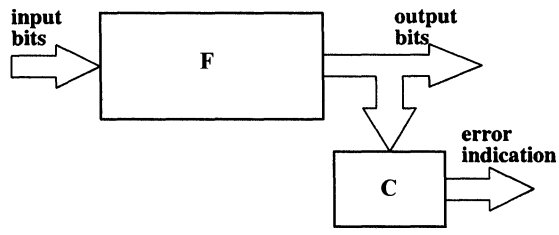


FIGURE 1 General structure of a SCC.

Based on this consideration, in [15, 17] we introduced new properties that guarantee that the SCC behaves correctly also when such faults affect its combinational or sequential functional block, as well as design rules ensuring the verification of these properties.

To cover the gap between the logical representation of data considered in the SFS definition, and the analog and dynamic effects possibly produced by the considered faults, these design rules pose specific constraints on the electrical parameters of F.

Because of the expected increased sensitivity to parameter variations occurring during manufacturing of next generation, very deep submicron technology ICs, these electrical constraints may be difficult to be satisfied by the manufactured circuit, thus possibly causing significant yield loss.

Based on these considerations, in this paper we propose a different approach to the design of SCCs with combinational functional blocks that guarantees their correct behavior, also in case of the occurrence of internal faults of the considered kind. A preliminary version of this approach was introduced in [22].

In more details, we present a signal coding technique where the conventional logical representation of data is enhanced by introducing some electrical-level-related information. This leads to the design of ICs which are inherently highly and easily testable with respect to not only conventional logical faults (*i.e.*, stuck-ats), but also faults affecting their electrical characteristics (*e.g.*, BFs, SONs, SOPs). The proposed coding technique can be either used in place of SCCs' conventional, logical error detecting codes, or jointly used with a conventional error detecting code.

It should be noted that, with the proposed coding technique, the number of bits in the encoded and unencoded information are equal to one another. Therefore, our coding is particularly

suitable to be used as the only error detecting code in all cases where the encoded information is driven along a data bus which, consequently, will not need additional bit lines.

Moreover, as shown later, our coding is also particularly suitable to be used for functional blocks of fail-safe systems [12, 29, 26].

To possibly adopt our coding technique to design combinational functional blocks of SCCs, we also present novel CMOS implementations of elementary gates that perform the basic Boolean operations (NAND, NOR, EXNOR) according to the introduced signal coding.

Differing from [15, 17], the proposed approach ensures the correct behavior of the derived CFB (from the point of view of the whole SCC) with respect to the considered realistic faults, without the need to satisfy any specific constraint on the circuits' electrical parameters.

Consequently, our approach is better suited to SCCs implemented by next generation, very deep submicron technologies, where circuits will be more prone and sensitive to parameter variations occurring during manufacturing.

This paper is organized as follows. In Section 2, we introduce the basic considerations behind the proposed approach. In Section 3, we describe the proposed signal coding technique. In Section 4, we discuss the application of such coding to the design of SCCs with combinational functional blocks, and introduce CMOS gates performing basic logical operations according to the proposed data representation. In Section 5, we discuss the impact of the proposed technique on the derived SCC's area overhead, speed and power consumption. In Section 6, we verify the correct behavior (from the point of view of the whole SCC) of the derived functional block, while some conclusive remarks are drawn in Section 7.

2. BASIC CONSIDERATIONS

The major difficulties in guaranteeing the correct behavior (from the point of view of the whole SCC) of a functional block with respect to possible internal non-logical faults (*e.g.*, BFs, SONs, SOPs) derive from the fact that the circuit behavior in the presence of these faults strongly depends on circuits' electrical characteristics which, for todays' deep submicron ICs, and especially for next generation very deep submicron ICs, can not be exactly predicted.

For instance, in case of BFs, the problems in predicting (and consequently possibly guaranteeing) their detection are due to the fact that BFs give rise to an intermediate voltage value at the output of the affected gate which may be recognized as logic "1" or "0", depending in the logic threshold of the fan-out gates. The ambiguity in the logic thresholds of the fabricated gates (due to variations of their electrical parameters occurring during manufacturing) makes BFs' eventual detection/non-detection (as logical errors) difficult, if not impossible, to be predicted or guaranteed.

Similar considerations hold true also for SONs.

As for SOPs, instead, the difficulties in ensuring their detection/non-detection are mainly due to the sequential behavior they induce on a faulty combinational circuit which, in turn, is due to the charge retaining ability of the faulty gate.

Generally, the crucial problem is to predict (and guarantee) the detection as logical error of an induced, faulty electrical behavior, which strongly depends on circuits' electrical characteristics which are prone to statistical variations during manufacturing.

3. PROPOSED SIGNAL CODING

To overcome the difficulties mentioned above, we introduce a novel data representation, where three are the possible data "values": (1) a $\mathbf{0} \rightarrow \mathbf{1}$ transition within a considered time interval T; (2) a $\mathbf{1} \rightarrow \mathbf{0}$ transition within T; (3) neither $\mathbf{0} \rightarrow \mathbf{1}$, nor $\mathbf{1} \rightarrow \mathbf{0}$ transition within T. Signals of kind 1 and 2 take place in case of correct operation, while signals of kind 3 denote the presence of a fault affecting the circuit behavior.

In addition, we can assume that signals of kind 1 and 2 are in a one-to-one correspondence with the conventional logical representation of data. For instance, signals of kind 1 correspond to the *high logic value*, while signals of kind 2 correspond to the *low logic value* (Fig. 2).

Of course, T must be chosen in order to ensure that the inputs of F are stable during such a time interval. Hence, if the inputs and outputs of F are sampled by means of flip–flops, T can generally be chosen equal to the period of the input flip–flops' clock. Alternatively, a periodic clock signal, with proper period T, should be suitably provided.

Of course, the presence of signals of kind 1 or 2 on all the functional block outputs can be regarded to as presence of a *codeword*, while that of at least one signal of kind 3, as presence of a *non-codeword.*

Let us show that our coding technique avoids the problems discussed in [14], in particular the possible occurrence of an incorrect codeword at the output of a functional block (using a conventional unordered [4], or parity code), because of a BF between the outputs of two gates with comparable driving strengths, that, in the fault-free case, should assume opposite logic values.

If our coding technique is used, a similar condition might on principle occur if, because of a BF, a $\mathbf{0} \rightarrow \mathbf{1}$ and a $\mathbf{1} \rightarrow \mathbf{0}$ signals (shorted together, and driven by gates with comparable driving strengths) were recognized by the fan-out gates as a $\mathbf{1} \rightarrow \mathbf{0}$ and a $\mathbf{0} \rightarrow \mathbf{1}$ signal, respectively. It can be easily verified that this condition can never occur, independent of the functional block electrical parameters (in particular independent of the fan-out gate logic thresholds).

In fact, as shown in Figure 3, if a $\mathbf{0} \rightarrow \mathbf{1}$ and a $\mathbf{1} \rightarrow \mathbf{0}$ signals are shorted together by a BF, intermediate voltage signals $V_{x1} \rightarrow V'_{x1}$ and $V_{x2} \rightarrow V'_{x2}$ may be produced on the shorted lines (1 and 2). Obviously it is: $V_{x1} \leq V'_{x1}$, $V_{x2} \geq V'_{x2}$, $V_{x1} \leq V_{x2}$, $V'_{x1} \geq V'_{x2}$. Instead, in order to make a $\mathbf{0} \rightarrow \mathbf{1}$ and a $\mathbf{1} \rightarrow \mathbf{0}$ be recognized as a $\mathbf{1} \rightarrow \mathbf{0}$ and a $\mathbf{0} \rightarrow \mathbf{1}$, we should have: $V_{x1} > V_{x2}$, $V'_{x1} < V'_{x2}$ (note that if $V_{x1} = V'_{x1}$, or/and $V_{x2} = V'_{x2}$ we obtain a signal of kind 3, hence a *non-codeword*). Of course, these latter conditions can never be satisfied whichever the fan-out gates' logic thresholds and electrical parameters in general.

Hence, our coding technique allows to avoid the dangerous situations discussed in [14] (compromising the SCC correct operation), without posing any critical constraint on the circuit electrical parameters. Therefore, our coding is better suited to next generation, very deep submicron technology circuits.

In addition, our coding technique inherently avoids that incorrect codewords are produced at the output of F also in the (less critical) case of internal SONs, SOPs and BFs, different from the dangerous ones considered above. This aspect, which has been proved by means of electrical level simulations, will be discussed in details later. However it is also evident that, because of these faults, it is more likely that a fault-free $\mathbf{1} \rightarrow \mathbf{0}$ (or $\mathbf{0} \rightarrow \mathbf{1}$) signal is either not affected by the fault, or transformed into a signal of kind 3, rather than into the incorrect $\mathbf{0} \rightarrow \mathbf{1}$ (or $\mathbf{1} \rightarrow \mathbf{0}$) signal.

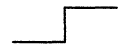
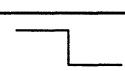
Logical repres.	Proposed repres.
"1"	
"0"	

FIGURE 2 Considered correspondence between the conventional logical representation of data and the novel representation proposed here.

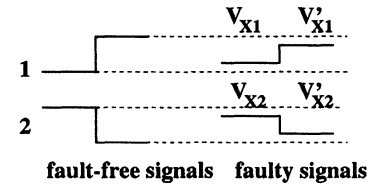


FIGURE 3 Encoded fault-free signals, and faulty signals possibly produced in the case of a BF between lines 1 and 2.

Therefore, our coding technique is inherently suited to implement functional blocks that, in case of internal SONs, SOPs and BFs, are more likely to produce an output non-codeword or the correct codeword, rather than an incorrect codeword. As for DFs, CFs and TFs possibly affecting our combinational functional block, they can be easily detected on-line by directly connecting to the functional block outputs detectors of the kind introduced in [21].

4. DESIGN OF SCCs WITH COMBINATIONAL FUNCTIONAL BLOCKS

As previously introduced, our coding technique can be used to design a CFB together with conventional SCCs' error detecting codes, or as the only error detecting code. In this last case, differing from all cases where conventional error detecting codes are used, the number of bits in the encoded and non-encoded information are equal to one another.

Similar to the cases where conventional SCCs' error detecting codes are used, an input encoding stage is generally needed.

As for the encoded information produced at the output of F, instead, it can be either decoded into the conventional logical representation of data by means of a suitable decoding stage, or it can be maintained in the obtained encoded form, depending on system's requirements. For instance, in case of fail-safe systems [12], as the proposed signal coding is very similar to that required at the system output [29, 26], no decoding stage may be needed.

It derives that, on the whole, a SCC using the proposed coding technique presents the internal organization shown in Figure 4.

It should be noted that this internal organization is the same as that of SCCs using conventional error detecting codes (*e.g.*, unordered codes).

It can be easily verified that our encoding and decoding stages can be implemented by means of simple circuits of the kind shown in Figure 5, to be directly connected to each input/output of the considered CFB.

In particular, regarding the decoding stage (Fig. 5), as in our coding technique the logic value lastly assumed by the (fault-free) encoded signal equals the logic value that should have been assumed in the conventional logical representation of data, simple flip–flops (FFs) driven by a suitable clock signal can be used. If our CFB input–output delay d is such that $0 < d \leq T/2$, CK can be directly used as clock signal of the sampling FFs. Alternatively, a suitable signal (CK_d), equal to CK, but properly delayed with respect to it (to account for the CFB input–output delay), should be used.

Therefore, compared to some frequently used error detecting codes (*e.g.*, unordered codes), our coding technique implies less complex and lower area encoding and (eventually) decoding stages.

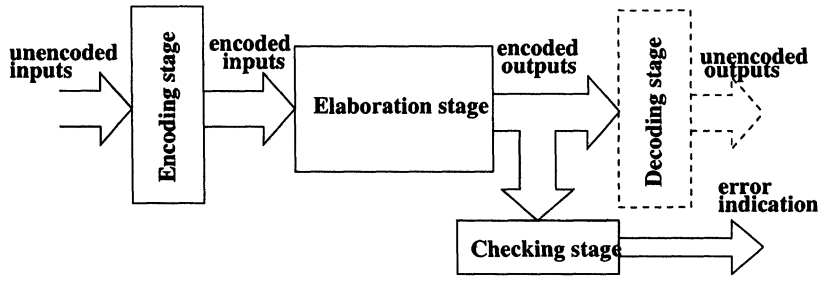


FIGURE 4 Internal organization of a self-checking circuit using our coding technique. The decoding stage may be present or not, according to system requirements.

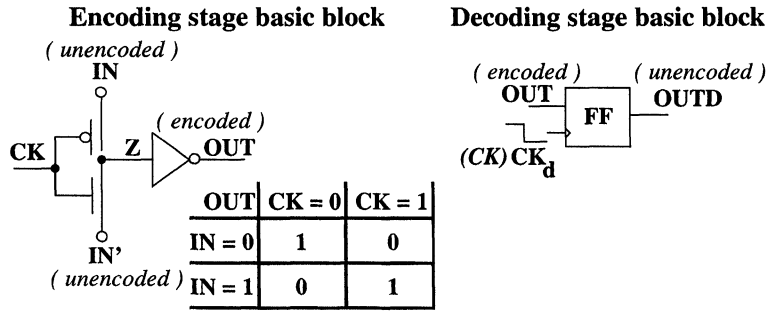


FIGURE 5 Encoding and decoding stage basic blocks (IN′ is the complement of signal IN).

As for the checking stage of the derived SCC, it is obvious that the correct operation of our CFB can be verified by simply checking that no produced output signal is a signal of kind 3 (*i.e.*, a signal not changing logic value twice within a considered time interval with duration = T). This can be accomplished by using two FFs sampling each output on both the clock falling and rising edges. These FFs' outputs can then be joined together by means of a conventional two-rail code checker (*e.g.*, one of those in [1, 13, 19]). In particular, as shown in Figure 6, if the implemented CFB has n outputs, an n-variable two-rail code checker (TRC_n) can be employed.

It should be noted that the FFs used for the checking stage can be also simultaneously used to implement the possibly required decoding stage described above, thus possibly allowing further area savings compared to SCCs using conventional error detecting codes.

Of course, similar to conventional SCCs, the outputs of our checking stage should be properly sampled by a connected error indicator [7, 23, 8, 9, 18], latching the produced error messages, and filtering out the possibly produced spurious error indications (*e.g.*, those possibly occurring during the checker input signal transitions and while the FFs' clock = 1).

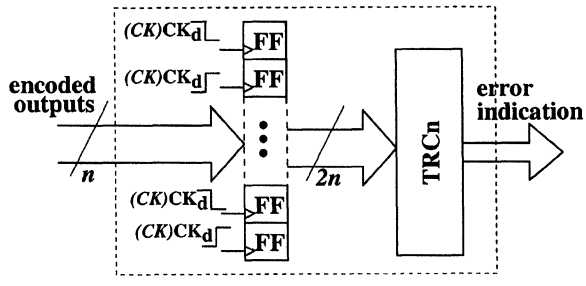
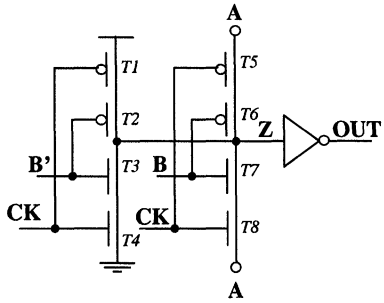


FIGURE 6 Internal structure of a possible checking stage.

Of course, in order to employ our coding technique to implement a CFB, gates performing conventional logical operations according to the proposed data representation are needed. CMOS implementations of such gates (NAND, NOR, EXNOR) are here proposed, and shown, together with their truth tables, in Figures 7, 8 and 9, respectively, where we have denoted by A′, B′ and CK′ the inverted signal A, B and CK, respectively.

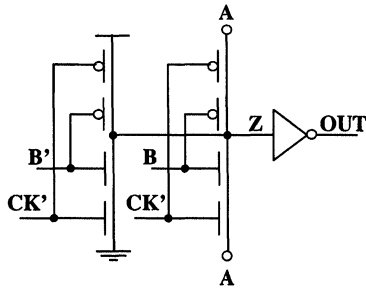
The reported truth tables show that, assuming that no reconverging path is present within our CFB, and considering (as usual [32]) that only single faults can occur at a time, these gates implement the basic NAND, NOR and EXNOR functions according to the introduced data representation. We can also notice that the NAND and NOR gates allow also the reconvergence of paths with the same inversion parity.

As previously introduced, CK denotes a suitable clock signal, whose period equals the considered time interval T of our coding technique. In this regard, the same clock signal can be used for all gates of our CFB. Alternatively, a wave pipelined structure [10] can be implemented by using, for the gates on the same propagation path, clock signals suitably delayed to one another (in order to account for the gates' input–output delay).



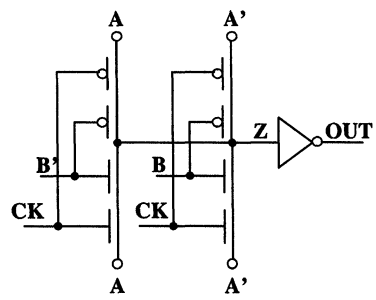
fault-free cases			*faulty cases*		
A	B	OUT	A	B	OUT
CK'	CK'	CK	CK	0	1
CK'	CK	CK	CK'	0	CK
CK	CK'	CK	CK	1	0
CK	CK	CK'	CK'	1	CK
			0	CK	1
			0	CK'	CK
			1	CK	0
			1	CK'	CK
			0	0	1
			1	1	0

FIGURE 7 Implementation and truth table of the proposed CMOS NAND.



fault-free cases			*faulty cases*		
A	B	OUT	A	B	OUT
CK'	CK'	CK	CK	0	CK'
CK'	CK	CK'	CK'	0	1
CK	CK'	CK'	CK	1	CK'
CK	CK	CK'	CK'	1	0
			0	CK	CK'
			0	CK'	1
			1	CK	CK'
			1	CK'	0
			0	0	1
			1	1	0

FIGURE 8 Implementation and truth table of the proposed CMOS NOR.



fault-free cases			*faulty cases*		
A	B	OUT	A	B	OUT
CK'	CK'	CK	CK	0	0
CK'	CK	CK'	CK'	0	1
CK	CK'	CK'	CK	1	1
CK	CK	CK	CK'	1	0
			0	CK	0
			0	CK'	1
			1	CK	1
			1	CK'	0

FIGURE 9 Implementation and truth table of the proposed CMOS EXNOR.

The behavior of these gates has been verified by means of electrical level simulations performed using HSPICE. As an example, we have considered a standard 0.7 μm CMOS technology with $V_{DD}=5\,V$, and symmetric gates with transistors' aspect ratios: $(W/L)_n=2$ and $(W/L)_p=4$.

Of course, because of the used pass-transistors, the voltage values produced on node Z (Figs. 7, 8 and 9) may be lower than V_{DD} (higher than GND), as shown as example in Figure 10 for the case of the EXNOR gate with both inputs changing logic value within T (in particular, A = CK′, B = CK (Fig. 9)).

However, we have verified that the worst case noise margins at the input of the output inverter (whose logic threshold is $\simeq 2.5\,V$) are $\simeq 1.4\,V$, hence they are high enough to ensure the successful fabrication of a reliable circuit. By means of Monte Carlo simulations we have also verified that the provided noise margins remain of the same order also considering variations of all circuit parameters (included those of the output inverter) up to the 20%, as shown, as an example, in Figure 11, which has been obtained considering the case of the EXNOR gate with both inputs changing logic value within T (in particular, A = CK′, B = CK (Fig. 9)). Of course, as for noise margin, no criticality arises when scaling to very deep submicron.

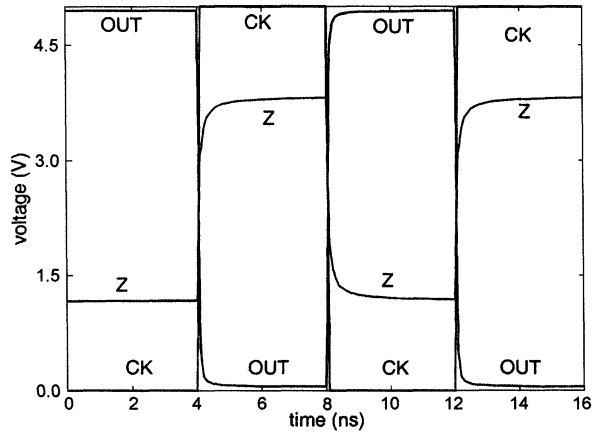


FIGURE 10 Waveforms obtained for the proposed EXNOR at the input (Z) of the output inverter, and at the gate output (OUT). The EXNOR gate input signals are: A = CK′, B = CK (Fig. 9).

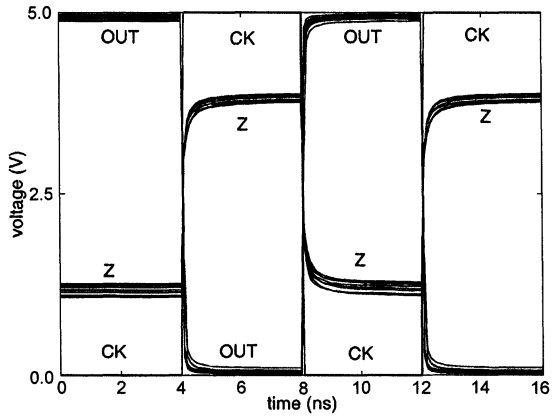


FIGURE 11 Waveforms obtained (by Monte Carlo simulations) considering variations of all circuit parameters up to the 20%.

5. COST OF THE PROPOSED TECHNIQUE

The proposed gate implementations imply an area increase compared to conventional CMOS implementations, also due to clock and complemented signals' routing.

However, as previously introduced, in our coding technique the number of bits in the encoded information equals the number of bits in the unencoded information. Moreover, we allow the use of very simple (and low area) encoding, decoding and checking stages. Hence, if adopted in place of conventional SCCs' error detecting codes (*e.g.*, two-rail codes, unordered codes), our technique offers savings, or at least does not increase global area overhead, while inherently featuring significant advantages in terms of self-checking ability.

As an example, considering the case where the two-rail code is employed, and referring to the case of a simple full-adder FCMOS design, the use of our coding allows a functional block

(included our encoding stage) relative reduction of area (estimated considering the number of required transistors) equal to the 11%. Because of clock and complemented signals' routing, such an area advantage will in practice result decreased. However area will remain smaller, or at least comparable, to that due to the use of the two-rail code. Similar considerations hold true if we consider the case of unordered codes. In addition, if we consider the system level, our method offers improved advantages, due to the unincreased number of bits in the encoded information compared to unencoded data.

Instead, our coding may imply an increase of dynamic power consumption (because of the generally increased number of signals' transitions). However, this contribution to dynamic power is partially compensated by the unincreased number of signals in the encoded information, with respect to the unencoded data, which allows a significant reduction of bus size. However, should power consumption be a primary concern, proper low power design techniques could be used together with the proposed coding, to reduce dynamic power consumption.

As for the derived SCC speed, we should consider that the proposed technique allows to simplify (with respect to the use of alternate error detecting codes) the functional block encoding, decoding and checking stages, which are usually on the SCC's critical timing path. This aspect reduces, if not fully compensates, the derived SCC speed drawback due to the need to sample twice the CFB output data. In addition, in order to improve the derived SCC speed, a wave pipelined structure could be implemented [10] for the CFB. In particular, by design, our derived CFB is particularly suitable to be implemented as a clock controlled wave pipeline which, compared to conventional pipelined structures, does not require latches' insertion at the various levels of the pipeline. In this way, a speed-up comparable to the number of levels in the implemented functional block can be achieved, with respect to the non-pipelined design.

6. SELF-CHECKING VERIFICATION

Let us verify that a CFB using our coding technique (and designed by means of the introduced elementary gates) behaves correctly from the point of view of the whole SCC, also in case of the occurrence of internal faults belonging to a set $\mathcal{F}$ composed of all possible SAs, SONs, SOPs and BFs, with values of connecting resistance $\in]0, 6k\Omega]$ [11]. As for DFs, CFs and TFs, we suppose to use a detecting scheme of the kind proposed in [21], while as for the clock signal, we assume that it is checked by the detecting scheme in [20].

In addition, we assume the fault hypotheses typically considered in case of SCCs [32], that is: (1) faults occur one at a time; (2) the time interval between the occurrence of two following faults is long enough to allow the application of all possible input codewords.

As for the possibly implemented CFB, we assume that simple gates (NAND, NOR, EXNOR), with comparable driving strengths, can be used. Moreover, we refer to the conditions on signals' reconvergency outlined in Section 4.

As our coding enhances the usual logical representation of data, differently from the conventional cases addressed in [15], the logical SFS property [32] can be still considered as a guarantee of the correct operation of the considered CFB.

Let us verify that our derived CFB satisfies the SFS property with respect to the considered set of faults $\mathcal{F}$.

To fulfill this purpose, because of the introduced assumptions and coding characteristics, it is sufficient to verify that: (a) each single fault possibly affecting one of the proposed elementary gates results in a signal not changing logic value twice within T (*i.e.*, a signal of kind 3) at the gate output, whose propagation to the CFB output is then guaranteed by design, or (b) the fault results in the presence of the correct signal (*i.e.*, a $\mathbf{0} \rightarrow \mathbf{1}$, or a $\mathbf{1} \rightarrow \mathbf{0}$ signal) at the output of the faulty gate and, in this last case, if following faults $\in \mathcal{F}$ occur, either an output signal of kind 3, or the

correct output signal is given. Condition (b) must be verified considering all possible sequences of internal faults.

As an example, in the reminder of this section we will analyze in details the case of faults $\in \mathcal{F}$ affecting the proposed NAND gate. Similar considerations hold true also for the proposed NOR and EXNOR gates, as well as for the required encoding gates.

As for SAs, it is obvious that our NAND satisfies conditions (a) or (b). Hence the derived CFB is SFS with respect to these faults.

Now let us consider all possible internal SONs.

As for the SON on transistor T1 (Fig. 7), we have verified that this fault remains undetected. If a following fault occurs, the circuit either behaves as if fault-free, or produces a signal of kind 3. In case of following faults, the undetected SON may affect the circuit correct behavior only when, because of the following fault and undetected SON, a faulty conductive path is created between V_{DD} and the output node Z. When this is the case, a high logic value is driven to node Z. Hence, in the time interval during which Z assumes, in the fault-free case, a high logic value, the behavior of the circuit is the same as that of the fault-free case. Instead, in the time intervals during which a low logic value should be given (in the fault-free case), either the correct low logic value, or a faulty high logic value is produced. Hence, either the correct signal, or a signal not changing logic value within T is given at the output of the NAND gate, as required by conditions (a) and (b).

Therefore, our CFB is SFS with respect to the SON on transistor T1.

Similar considerations hold true for SONs affecting T2, T3 and T4 (Fig. 7).

By means of similar considerations we have verified that the SON on transistor T5 (Fig. 7) does not affect the circuit behavior. As for the occurrence of following faults, the dangerous case is when the following fault is the SON on transistor T6. In this case, if B = CK, then OUT = A′ (as in the fault-free case). Hence the correct output codeword is produced. Instead, if B = CK′, the branch with inputs CK and B′ drives signal CK′ on node Z, while the branch with inputs CK, B and A drives signal A on such a node. Hence, if A = CK′, then OUT = CK (as in the fault-free case). Instead, if A = CK, an electrical conflict arises between the branch driving CK′ on node Z, and the branch driving CK on the same node. Hence an intermediate voltage value results on such a node that, depending on the logic threshold of the output inverter, results in a high, or a low logic value at the NAND output.

Therefore, either the correct signal, or a signal of kind 3 is produced at the output of the NAND gate, and our CBF is SFS with respect to the considered SON.

By means of similar analyses we have verified that our CFB is SFS also with respect to SONs on transistors T6, T7 and T8 (Fig. 7).

Similarly, we have also verified that our CFB is SFS with respect to the SONs possibly affecting the output inverter. Hence our CFB is SFS with respect to all possible SONs affecting its NAND gates.

By means of similar analyses, we have verified that our CFB is SFS with respect to all possible SOPs affecting its NAND gates.

Finally, let us consider the BFs possibly affecting our NAND gate.

We have performed electrical level simulations by means of HSPICE considering, for each BF, values of connecting resistance (R) the interval $]0, 6\Omega]$ [11]. Moreover, a standard 0.7 μm CMOS technology, and the transistors' aspect ratios reported in Section 4 have been assumed as reference.

The results achieved have shown that, depending on the considered BF and R values, either a signal of kind 3, (as shown, as an example, in Fig. 12 for the case of a BF with $R = 6k\Omega$ between nodes S and Z (Fig. 7)), or the correct signal is given. In this latter case, if following faults occur, either a signal of kind 3, or the correct output, is produced.

Hence our CFB is SFS with respect to the BFs possibly affecting its NAND gates.

Similarly, it can be easily verified that our CFB is SFS with respect to faults $\in \mathcal{F}$ affecting

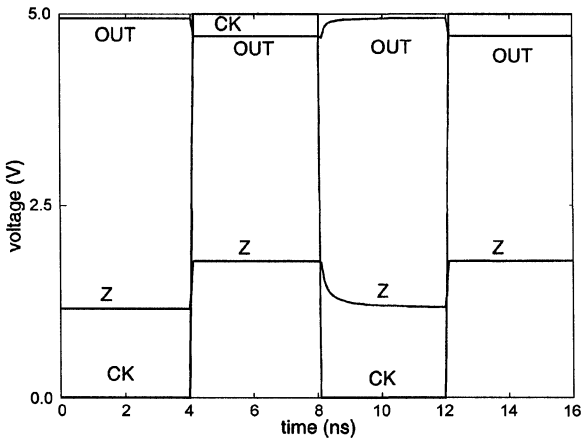


FIGURE 12 Waveforms obtained for the proposed NAND at the input (Z) of the output inverter and at the gate output (OUT), in case of the presence of a BF with $R = 6k\Omega$ between nodes S and Z (Fig. 7). The considered NAND inputs are A = B = CK.

its NOR and EXNOR gates, and with respect to BFs involving nodes of different gates.

7. CONCLUSIONS

In this paper we have presented signal coding and CMOS gates that are suitable to self-checking circuits with combinational functional blocks implemented also by very deep submicron technology.

In particular, the derived functional blocks satisfy the Strongly Fault-Secure property with respect to a wide set of possible, internal faults including not only conventional stuck-ats, but also transistor stuck-ons, transistor stuck-opens and resistive bridgings, whose likelihood, together with that of delays, crosstalks and transients (easily detected using a detecting scheme of the kind in [21]), can not be any longer neglected in the perspective of next generation, very deep submicron ICs.

Compared to the alternative solution presented in [15] (where not only stuck-ats, but also transistor stuck-ons, stuck-opens, resistive bridgings and delays were considered), the technique presented here does not imply any critical constraint on the circuit electrical parameters. Therefore, it is suitable to be adopted to design very deep submicron self-checking circuits which, compared to todays' circuits, will present significantly increased sensitivity to parameter variations occurring during fabrication.

References

[1] Anderson, D. A., "Design of self-checking digital network using coding techniques", *Tech. Report R-527, CSL*, Univ. of Illinois, IL, 1971.

[2] Anderson, D. A. and Metze, G., "Design of Totally Self-Checking Circuits for *m*-Out-of-*n* Codes", *IEEE Trans. Comput.*, **c22**, 263–269, March, 1973.

[3] Bolchini, C., Salice, F. and Sciuto, D. (1997). "Designing Networks with Error Detection Properties Through the Fault-Error Relation", *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 290–297.

[4] Bose, B., "On Unordered Codes", *IEEE Trans. Comput.*, **40**, 125–131, February, 1991.

[5] Carter, W. C. and Schneider, P. R. (1968). "Design of dynamically checked computers", *Proc. IFIP' 68*, Edinburgh, Scotland, pp. 878–883.

[6] Diaz, M., Azema, P. and Ayache, J. M., "Unified Design of Self-Checking and Fail-Safe Combinational Circuits and Sequential Machines", *IEEE Trans. Comput.*, **C28**, 276–281, March, 1979.

[7] Gaitanis, N., "A Totally Self-Checking Error Indicator", *IEEE Trans. Comput.*, **C34**, 758–761, August, 1985.

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

[9] Gaitanis, N., Kostarakis, P. and Paschalis, A. (1996). "A Three Rail Totally Self-Checking Error Indicator", *Proc. of 2nd IEEE Int. On-Line Testing Work*, pp. 50–52.

[10] Ghosh, D. and Nandy, S. K., "Design and Realization of High-Performance Wave-Pipelined 8×8 b Multiplier in CMOS", *IEEE Trans. on VLSI Systems*, **3**(1), March, 1995.

[11] Hao, H. and McCluskey, E. J. (1991). "Resistive Shorts Within CMOS Gates", *Proc. of IEEE Int. Test Conf.*, pp. 292–301.

[12] Lala, P. K., *Fault Tolerant and Fault Testable Hardware Design*, Prentice-Hall International, Englewood Cliffs, NJ, 1985.

[13] Lo, J. C., "A Novel Area-Time Efficient Static CMOS Totally Self-Checking Comparator", *IEEE J. of Solid State Circuit*, **28**, 165–168, February, 1993.

[14] Metra, C., Favalli, M., Olivo, P. and Riccò, B. (1993). "Design Rules for CMOS Self-Checking Circuits with Parametric Faults in the Functional Block", *Proc. of IEEE Int. Work. on Defect and Fault Tolerance in VLSI Systems*, pp. 271–278.

[15] Metra, C., Favalli, M., Olivo, P. and Riccò,, B., "On-Line Detection of Bridging and Delay Faults in Functional Blocks of CMOS Self-Checking Circuits", *IEEE Trans. on CAD*, **16**(7), 770–776, July, 1997.

[16] Metra, C., Favalli, M. and Riccò, B. (1994). "CMOS Self-Checking Circuits with Faulty Sequential Functional Blocks", *Proc. of IEEE Int. Work. on Defect and Fault Tolerance in VLSI Systems*, pp. 133–141.

[17] Metra, C., Favalli, M. and Riccò, B., "Design of CMOS self-checking sequential circuits with improved detectability of bridging faults", *IEE Electronics Letters*, **30**(23), 1934–1936, November, 1994.
[18] Metra, C., Favalli, M. and Riccò, B. (1996). "Compact and Highly Testable Error Indicator for Self-Checking Circuits", *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 204–212.
[19] Metra, C., Favalli, M. and Riccò, B. (1996). "Embedded Two-Rail Checkers with On-Line Testing Ability", *Proc. of IEEE VLSI Test Symp.*, pp. 145–150.
[20] Metra, C., Favalli, M. and Riccò, B. (1997). "On-Line Testing Scheme for Clocks' Faults", *Proc. of IEEE Int. Test Conf.*, pp. 587–596.
[21] Metra, C., Favalli, M. and Riccò, B. (1998). "On-Line Detection of Logic Errors due to Crosstalk, Delay, and Transient Faults", *Proc. of IEEE Int. Test Conf.*, pp. 524–533.
[22] Metra, C., Favalli, M. and Riccò, B. (1998). "Signal Coding Technique and CMOS Gates for Strongly Fault-Secure Combinational Functional Blocks", *Proc. of IEEE Int. Symp. on Defect and Fault Tolerance in VLSI Systems*, pp. 174–182.
[23] Nanya, T. and Kawamura, T., "On Error Indication for Totally Self-Checking Systems", *IEEE Trans. Comput.*, **c36**, 1389–1391, November, 1987.
[24] Nanya, T. and Uchida, M. (1989). "The Design of Strongly Fault-Secure and Strongly Code-Disjoint Combinational Circuits", *Proc. of Joint Fault-Tolerant Computing Symposium*, pp. 245–250.
[25] Nicolaidis, M. (1991). "Shorts in Self-Checking Circuits", *J. of Electronic Testing: Theory and Application*, **1**, 257–273.
[26] Nicolaidis, M., "Fail-Safe Interfaces for VLSI: Theoretical Foundations and Implementations", *IEEE Trans. Comput.*, pp. 62–77, January, 1998.
[27] Nicolaidis, M. (1998). "Scaling Deeper to Submicron: On-Line Testing to the Rescue", *Proc. of IEEE Int. Test Conf.*, p. 1139.
[28] Nicolaidis, M. and Courtois, B., "Strongly Code Disjoint Checkers", *IEEE Trans. Comput.*, **37**, 751–756, June, 1988.
[29] Nicolaidis, M., Noraz, S. and Courtois, B. (1989). "A Generalized Theory of Fail-Safe Systems", *Proc. of Int. Symp. Fault-Tolerant Comput.*, pp. 398–406.
[30] Parekhji, R. A., Venkatesh, G. and Sherlekar, S. D. (1991). "A Methodology for Designing Optimal Self-Checking Sequential Circuits", *Proc. of IEEE Int. Test Conf.*, pp. 283–291.
[31] Saposhnikov, V., Dmitriev, A., Goessel, M. and Saposhnikov, V. (1996). "Self-dual parity checking – a new method for on-line testing", *Proc. of IEEE VLSI Test Symp.*, pp. 162–168.
[32] Smith, J. E. and Metze, G., "Strongly fault-secure logic networks", *IEEE Trans. Comput.*, **C27**, 491–499, June, 1978.

Authors' Biographies

Cecilia Metra received the Eng. degree in Electronics Engineering and the Ph.D. degree in Electronics Engineering and Computer Science from the University of Bologna (Italy). Currently she is Contract Professor in electronics at the University of Bologna (Italy).

Her main research interests are defect analysis, fault modeling, on-line testing, self-checking circuit design, fault-tolerance and FPGA testing.

She has been General Co-Chair of "The IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems" 1999, Program Co-Chair of "The IEEE Int. Symposium on Defect and Fault Tolerance in VLSI Systems" 1998, and Vice-Program Co-Chair of the "IEEE Int. On-Line Testing Workshop" 1998, 1999. She is a member of the IEEE.

Michele Favalli received the Dr. Eng. degree in Electronics Engineering from the University of Bologna in 1987 and the Ph.D. in Electronics Engineering and Computer Science in 1993. From 1993 he has worked at the Department of Electronics at the University of Bologna as a Researcher. From 1998 he is Associate Professor of Computer Science at the University of Ferrara. His research interest are in the area of digital IC's design, simulation and testing, including design for testability, self-checking and fault tolerant circuits, fault modeling and simulation.

Bruno Riccò born in 1947, graduated in electrical engineering in 1971; Ph.D. at the University of Cambridge (UK) in 1976; Full Professor of Electronics at the University of Bologna since 1980; visiting scientist at Stanford and the IBM Research Center (Yorktown Heights) in 1983–1986; European Editor of IEEE Transactions ED from 1988 to 1996; since 1991 President of the Scientific Board of the Consortium ULISSE; since 1994 Director of the Journal Alta Frequenza; since 1995 President of the Group of Electronics of the Italian Association of Electrical and Electronics Engineers (AEI); since 1999 President of the Group of Electronics Engineers in Italy; Marconi award of AEI in 1995. Prof. Riccò works on semiconductor devices and integrated circuits and has published more than 260 papers on major international journals and conferences.