INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2015, VOL. 61, NO. 1, PP. 67–75
Manuscript received December 24, 2014; revised March 2015. DOI: 10.1515/eletel-2015-0009

New Structure of Test Pattern Generator Stimulating Crosstalks in Bus-type Connections

Tomasz Garbolino

***Abstract*—The paper discloses the idea of a new structure for a Test Pattern Generator (TPG) for detection of crosstalk faults that may happen to bus-type interconnections between built-in blocks within a System-on-Chip structure. The new idea is an improvement of the TPG design proposed by the author in one of the previous studies. The TPG circuit is meant to generate test sequences that guarantee detection of all crosstalk faults with the capacitive nature that may occur between individual lines within an interconnecting bus. The study comprises a synthesizable and parameterized model developed for the presented TPG in the VLSI Hardware Description Language (VHDL) with further investigation of properties and features of the offered module. The significant advantages of the proposed TPG structure include less area occupied on a chip and higher operation frequency as compared to other solutions. In addition, the design demonstrates good scalability in terms of both the hardware overhead and the length of the generated test sequence.**

***Keywords*— integrated circuit interconnections, crosstalk, test pattern generator, built in self-test, system-on-a chip**

I. Introduction

RAPID development of technologies associated with manufacturing of VLSI circuits has led to the situation when an entire digital system can fit on a single silicon chip. Such a one-chip integrated circuit, referred to as System-on-Chip (SoC) is made up of a set of embedded cores that communicate with each other via a network of long dedicated connections implemented, for instance, as shared bus lines [23], [24], [28]. In parallel, the idea of data exchange between modules of the integrated circuit via a Network-on-Chip (NoC) has become pretty popular over the recent years [13], [33], [25], [10], [5], [20], [17], [8], [12]. But anyway, interconnections between nodes of such a network are usually implemented as an internal bus.

Widespread application of nanometer technologies to manufacturing of one-chip integrated circuits entails increasing of parasite capacitances and inductances between interconnections that are geometrically close one to another [23], [24]. This, in turn, results in substantial crosstalks that adversely affect data transmission reliability and timing performances of the system [7], [3], [18]. Bus connections that enable transmission between nodes of a NoC usually comprise from several to several dozens or even several hundreds of long lines routed in parallel and very close each other. It is why they are particularly exposed to the hazard of adverse interferences that results from crosstalks [21], [7], [37]. By its nature such buses represent communication means that are really sensitive to interferences. It results from the need to assure cohesion and full synchronization of information that is transmitted via individual lines of the bus [31].

The available literature report a number of design and manufacturing solutions aimed at reduction of crosstalks between interconnections in VSLI circuits, for instance [38], [36]. However, there is no single method that guarantees successful and complete resolving of the problem [23]. Attempts to entirely eliminate crosstalks may prove cost inefficient since it is extremely difficult to keep full control of parasitic capacitances and inductances between interconnections when integrated circuits are manufactured with use of nanometer technologies. In addition it is infeasible to take account of all possible phenomena associated with dispersion of manufacturing parameters and of possible defects that may occur during the manufacturing process and lead to crosstalks. It was observed that dispersion of technological parameters during the manufacturing process may eventually lead to twofold growth of the maximum amplitude of the crosstalk pulse [30], [23].

Thus, it proves indispensable to consider crosstalk faults in manufacturing tests [1], [19], [28], [23]. To detect, classify and localize dynamic faults it is necessary to supply test vectors to the input of the interconnection network at the rates that correspond to the nominal frequency of the circuit operation, i.e. in the “test per-clock” mode. Unfortunately, most of external test pattern generators are capable of working merely with frequencies that are by several orders lower than the frequency rates typical for clocking of internal structures within SoCs. It entails the need to apply various techniques for built-in self-testing of interconnections (IBIST) [4], [26], [32], [27], [23], [29]. Conventional IBIST architectures for bus-type interconnections comprise a Test Patter Generator (TPG) for stimulating input vectors and an analyzer of test responses [37], [14], [11], [15], [28], [23], [29].

This study is dedicated to an innovative structure of a test pattern generator aimed at stimulation of crosstalk faults that may happen to bus-type connections. The fully scalable and synthesizable model of such a TPG unit has been developed in the VHDL language. The model was subjected to thorough investigations with regard to the length of the output test sequence and the associated hardware overhead.

The TPG disclosed in this study is an improved option of the solution already described in [9]. The new design of the TPG benefits from reduction in length of the shift register and alteration of the test vector counter. These measures made it possible to achieve a TPG structure that highlights with less hardware overhead and much higher operation frequency than the solution presented in [9].

The Author is with Faculty of Automatic Control, Electronics and Computer Science, Silesian University of Technology, Gliwice, Poland (e-mail: Tomasz.Garbolino@polsl.pl).

The further part of the study is structured in the following way: Section II comprises classification of crosstalk faults, basic types of test sequences dedicated to stimulate such faults are outlined in Section III, whereas Section IV is dedicated to the concept of the new TPG and the analysis of its key properties is provided in Section V. The proposed solution is compared against the existing ones in Section VI and finally, the recapitulation of the current achievements and plans for future developments are covered by Section VII.

II. Classification of Crosstalk Faults

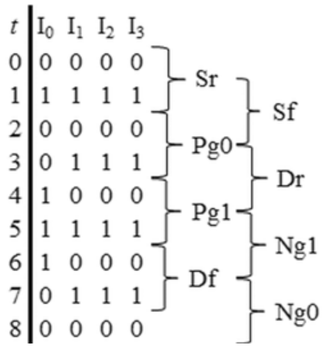
Basically, crosstalk faults lead to either undesired temporary changes in the logic state of a victim line or to delayed or premature occurrence of a desired change in the line status. Each of the foregoing phenomena that shall be further referred to as a crosstalk fault is caused by occurrence of a rising or a falling edge, i.e. transition 0→1 or 1→0, on one or several aggressor lines. Hereinafter the victim and aggressor lines shall be denoted correspondingly with V and A symbols.

One of undesired effect entailed by crosstalk faults consists in a positive glitch (Pg) or a negative glitch (Ng) pulses in the victim line [7]. Depending on the initial status of the victim line, i.e. whether the steady state is low (0) or high (1), the mentioned symptoms of crosstalks are classified as one of four types of crosstalk faults [22], [34]. The Pg0 /Pg1/ shall stand for a positive pulse in the victim line that is normally in the low (0) /high (1)/ logic state, whilst Ng1 /Ng0/ corresponds to a negative pulse in the victim line that is normally in the high (1) /low (0)/ logic state. Another adverse effect of crosstalks between adjacent lines is a delay of a rising edge (Dr) or a falling edge (Df) in the victim line [7], [22]. Crosstalk between interconnections may also lead to premature occurrence of a rising or a falling edge, which is referred to as the rising edge speed-up (Sr) or the falling edge speed-up (Sf) in the interfered interconnection [2], [31].

III. Test Sequences Dedicated to Stimulate Crosstalk Faults

In practice it proves infeasible to test the entire network of interconnections to find out all possible physical defects and deviations of the technological process that may entail crosstalks between data transmission lines and, eventually, lead to errors in conveyed digital signals [7]. It is why abstractive models of crosstalk faults are used for investigations. Such a model should define, among other things, the requirements that must be fulfilled by vectors of the test sequence to guarantee that all types of crosstalk fault that are considered by the model are stimulated.

Reference [7] discloses the model of crosstalk faults that is referred to as the Maximum Aggressor Fault Model (MAFM) and is commonly used in many scientific studies including [37], [6], [11] that deal with testing of crosstalk faults. The model assumes that at each specific moment of time only one interconnection within the network is the victim line and all other ones can act as aggressors. Thus, to stimulate a crosstalk fault in the victim line one has to simultaneously produce the interfering edge with the same transition directions (i.e. either 0→1 or 1→0) in all aggressor interconnections. In consequence, an error can appear in the victim line as a result of a crosstalk when an ordered pair of parallel test vectors is supplied to inputs of the interconnecting network.



t	I_0	I_1	I_2	I_3
0	0	0	0	0
1	1	1	1	1
2	0	0	0	0
3	0	1	1	1
4	1	0	0	0
5	1	1	1	1
6	1	0	0	0
7	0	1	1	1
8	0	0	0	0

Fig. 1. A fragment of the XMAFM sequence for four interconnections

In [37] a test sequence is described that is based on the MAFM model and is suitable for testing of crosstalk faults. It enables stimulation of four, the most typical crosstalk faults, i.e. Pg0, Ng1, Dr, Df. That sequence is commonly referred to in various literature sources as the MA, MAF or MAFM sequence but only that last name shall be used in this study. The total length of such a sequence is 6*v*, where *v* stands for the number of interconnections within the network.

The primary MAFM sequence is incapable of stimulating the crosstalk fault of the Pg1, Ng0, Sr and Sf types. Thus, this study considers the extended version of that sequence that is denoted as the XMAFM (the eXtended MAFM sequence) since it guarantees that all aforementioned types of crosstalk faults are reliably stimulated.

For the network that comprises *v* interconnections the minimum length of the XMAFM sequence is 6*v*+3. A fragment of the XMAFM sequence for a network that is made up of four interconnections I_0, I_1, I_2 and I_3 is shown in Fig. 1.

The *t* numbers in the left-hand side of the figure stand for subsequent moments of time when the specific PTV_t (Parallel Test Vector) is supplied to inputs of the network of interconnections under test. The crosstalk faults in the I_0 victim line, which are stimulated by the individual pairs of the PTV_t and PTV_{t+1} vectors, are marked in the right side of Fig. 1. At the moments of transitions between the PTV_0 and PTV_1 vectors as well as between PTV_1 and PTV_2 ones each of the interconnections is at the same time both an aggressor and a victim line.

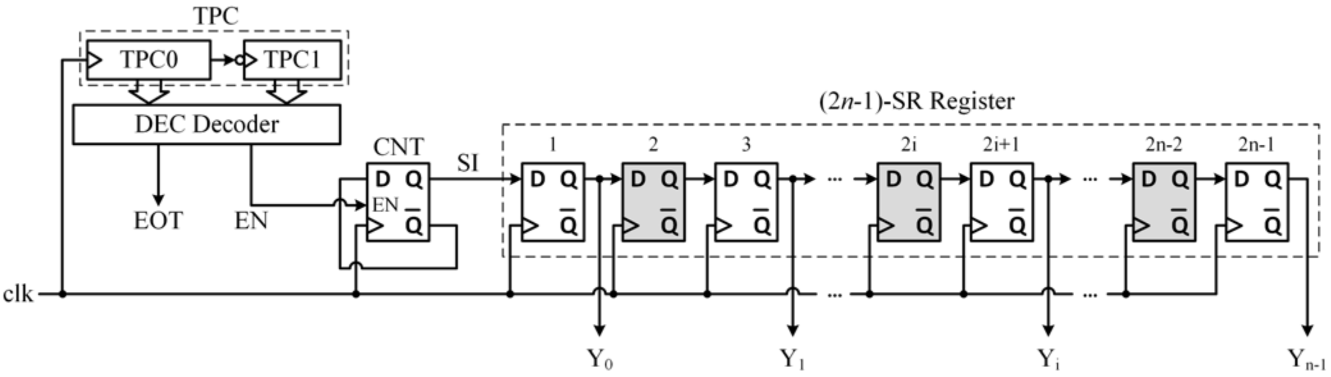


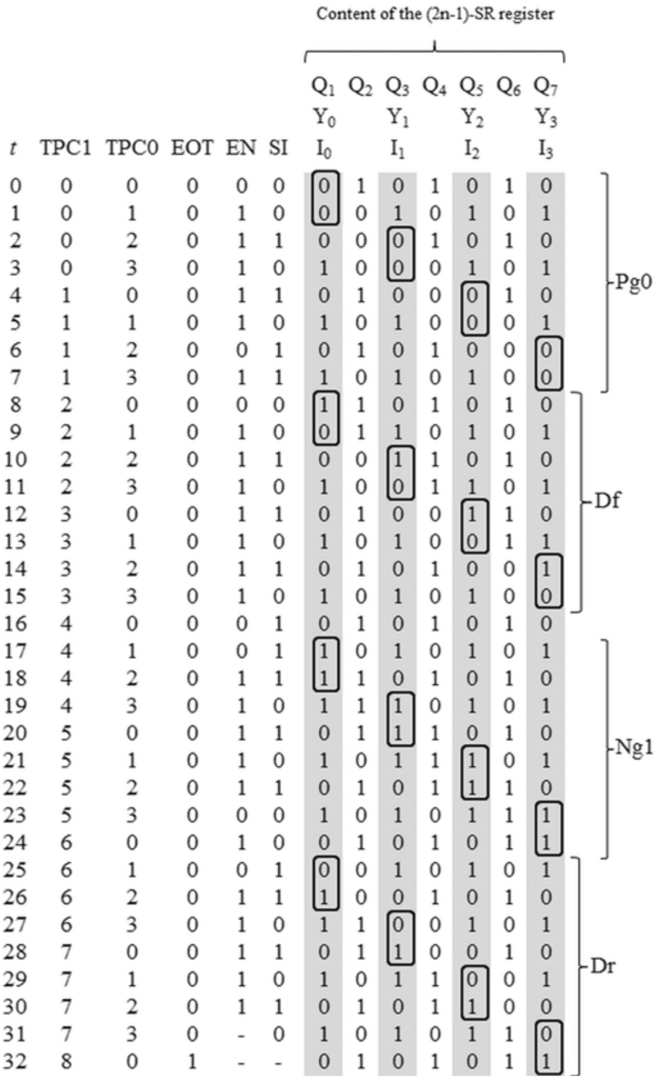
Fig. 2. The structure of the MAFM and XMAFM Test Pattern Generators that use the (2*n*-1)-SR register

IV. Innovative Concept of the Test Pattern Generator for Crosstalk Faults

For needs of this study the TPG of the MAFM and XMAFM types is made up of a (2*n*-1)-SR shift register that comprises 2*n*-1 synchronous flip-flops of *D* type, where *n*=*v* stands for the number of data transmission lines within the network of interconnections under test. The TPG structure, denoted hereinafter as (2*n*-1)-SR-TPG is shown in Fig. 2. The *n* parameter shall be also referred to in the subsequent part of this study as the size of the (2*n*-1)-SR-TPG module. Each I_i line of the network under test, where $i \in \{0,1,\ldots,n-1\}$ is connected to the Y_i output of the TPG, which is also the output with the 2*n*-1 number of the aforementioned (2*n*-1)-SR register. The *SI* input of that register is supplied with the sequence from the output of a one-bit binary counter (a binary toggle) that is made up of a D-type flip flop with the enabling (EN) input. In Fig. 2 this counter is denoted as CNT. The EN signal is provided by a decoder that is connected to outputs of the Test Pattern Counter (TPC). When EN=1 (high) the output of the CNT counter toggles to the opposite state upon each rising edge of the clock signal but for EN=0 (low) the CNT counter remains unaltered regardless the changes at the clock input. The decoder also produces the End of Test (EOT) signal that issues the EOT=1 (high) signal at its output when generation of the test sequence is finished. The Test Pattern Counter is made up of two counters, TPC0 and TPC1, connected in series, as it is shown in Fig. 2. The most significant output of the TPC0 counter is connected to the clock input of the TPC1 counter. In case of a circuit designed to generate the test sequence according to the Maximum Aggressor Fault Model (MAFM) the TPC0 counter is meant to count up modulo *n* whilst the TPC1 is a 4-bit counter that counts in the natural binary code. When the TPG is required to produce the XMAFM test sequence the TPC0 counter counts up modulo 2*n* whilst the TPC1 is operated as a 3-bit binary counter. The TPC0 and TPC1 circuits are synchronized by the corresponding rising and falling edges of the clock waveform. The new design of the circuit for counting of test vectors makes it possible to achieve higher rates of the TPG operating frequency as compared to the solution disclosed in [9].

Example 1

The sequence of events associated with generation of the MAFM test sequence for an example network of interconnections that comprises *v* = *n* = 4 lines is illustrated in Fig. 3. The figure presents content of the TPC1 and TPC0 counters as well as the content of the (2*n*-1)-SR register, including statuses of the Y_0, Y_1, Y_2 and Y_3 outputs of that register for subsequent test vectors. Also the levels of EOT, EN and SI signals are shown for subsequent clocks. The '-' denotes that the signal level is irrelevant at the specific moment of time since it has no impact onto correct operation of the TPG circuit. Sequences of bits supplied to I_0, I_1, I_2 and



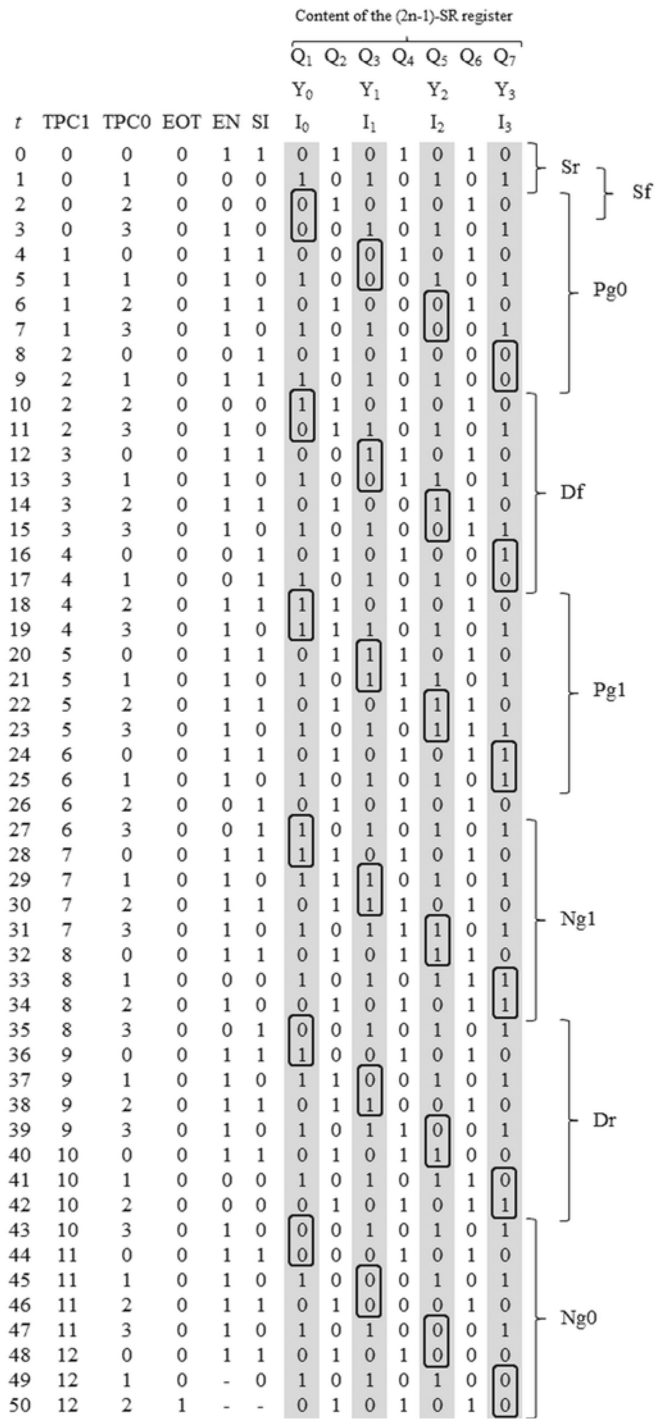
						Content of the (2n-1)-SR register							
						Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	
						Y_0		Y_1		Y_2		Y_3	
t	TPC1	TPC0	EOT	EN	SI	I_0		I_1		I_2		I_3	
0	0	0	0	0	0	0	1	0	1	0	1	0	Pg0
1	0	1	0	1	0	0	0	1	0	1	0	1	
2	0	2	0	1	1	0	0	0	1	0	1	0	
3	0	3	0	1	0	1	0	0	0	1	0	1	
4	1	0	0	1	1	0	1	0	0	0	1	0	
5	1	1	0	1	0	1	0	1	0	0	0	1	
6	1	2	0	0	1	0	1	0	1	0	0	0	
7	1	3	0	1	1	1	0	1	0	1	0	0	
8	2	0	0	0	0	1	1	0	1	0	1	0	Df
9	2	1	0	1	0	0	1	1	0	1	0	1	
10	2	2	0	1	1	0	0	1	1	0	1	0	
11	2	3	0	1	0	1	0	0	1	1	0	1	
12	3	0	0	1	1	0	1	0	0	1	1	0	
13	3	1	0	1	0	1	0	1	0	0	1	1	
14	3	2	0	1	1	0	1	0	1	0	0	1	
15	3	3	0	1	0	1	0	1	0	1	0	0	
16	4	0	0	0	1	0	1	0	1	0	1	0	
17	4	1	0	0	1	1	0	1	0	1	0	1	Ng1
18	4	2	0	1	1	1	1	0	1	0	1	0	
19	4	3	0	1	0	1	1	1	0	1	0	1	
20	5	0	0	1	1	0	1	1	1	0	1	0	
21	5	1	0	1	0	1	0	1	1	1	0	1	
22	5	2	0	1	1	0	1	0	1	1	1	0	
23	5	3	0	0	0	1	0	1	0	1	1	1	
24	6	0	0	1	0	0	1	0	1	0	1	1	
25	6	1	0	0	1	0	0	1	0	1	0	1	Dr
26	6	2	0	1	1	1	0	0	1	0	1	0	
27	6	3	0	1	0	1	1	0	0	1	0	1	
28	7	0	0	1	1	0	1	1	0	0	1	0	
29	7	1	0	1	0	1	0	1	1	0	0	1	
30	7	2	0	1	1	0	1	0	1	1	0	0	
31	7	3	0	-	0	1	0	1	0	1	1	0	
32	8	0	1	-	-	0	1	0	1	0	1	1	

Fig. 3. Explanation how the MAFM sequence is generated for an example network with four (4) interconnections

I_3 interconnections under test are distinguished in Fig. 3 with grey background. The fragments of the test sequence while each specific data line acts as a victim are enveloped with rectangles individually for each interconnection. The curly brackets placed at the right-hand side of the figure distinguish these fragments of the test sequence that are dedicated to stimulate specific types of crosstalks. The test sequence length is m = 33. The EN signal adopts the zero level (EN = 0) for the following combinations between contents of the TPC1 and the TPC0 counters: ⟨*TPC*1, *TPC*0⟩ ∈ {⟨0,0⟩, ⟨1,2⟩, ⟨2,0⟩, ⟨4,0⟩, ⟨4,1⟩, ⟨5,3⟩, ⟨6,1⟩}. However, when ⟨*TPC*1, *TPC*0⟩ = ⟨7,3⟩ and the content of the TPC1 counter is not less than 8 (*TPC*1 ≥ 8), the logic level of the EN signal is irrelevant (EN = -) that is beneficial to reduction of the decoder complexity. For all remaining combinations between contents of the TPC0 and TPC1 counters the EN signal adopts the high level (EN = 1). The EOT signal is high (EOT = 1) only for the last vector of the sequence, which means, for the case in question, for the combination between contents of the TPC1 and TPC0 counters equal to ⟨*TPC*1, *TPC*0⟩ = ⟨8,0⟩. For all further combinations between contents of the both counters, i.e. when *TPC*1 = 8 and *TPC*0 > 0 as well as for *TPC*1 > 8, the level of the EOT signal is irrelevant (EOT = -). ♦

Example 2

The XMAFM test sequence is produced for the network that comprises $v = n = 4$ interconnections by the (2*n*-1)-SR-TPG circuit in quite similar way as is exhibited in Example 1 for the MAFM sequence (see Fig. 4). Hence the considerations here shall be limited merely to differences between the generation procedures for the both said sequences. The length of the XMAFM sequence in question is $m = 51$. The set of combinations between the contents of the TPC1 and TPC0 counters when the EN signal adopts the low level (EN=0) comprises the following components: ⟨*TPC*1, *TPC*0⟩ ∈ {⟨0,1⟩, ⟨0,2⟩, ⟨2,0⟩, ⟨2,2⟩, ⟨4,0⟩, ⟨4,1⟩, ⟨6,2⟩, ⟨6,3⟩, ⟨8,1⟩, ⟨8,3⟩, ⟨10,1⟩, ⟨10,2⟩}. When *TPC*1 = 12 and simultaneously *TPC*0 > 0 as well as for *TPC*1 > 12, the level of the EN signal can be irrelevant (EN=-) with no adverse impact to correct operation of the Test Pattern Generator. For all remaining combinations between contents of the TPC1 and TPC0 counters the EN signal remains high (EN=1). The EOT signal adopts the high logic level (EOT=1) only for a single combination between contents of the TPC1 and TPC0 counters, i.e. for: ⟨12,2⟩. Otherwise, when *TPC*1 = 12 and *TPC*0 > 2 as well as for *TPC*1 > 12, the EOT level is irrelevant (EOT = -). ♦



						Content of the (2n-1)-SR register						
						Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
						Y_0		Y_1		Y_2		Y_3
t	TPC1	TPC0	EOT	EN	SI	I_0		I_1		I_2		I_3
0	0	0	0	1	1	0	1	0	1	0	1	0
1	0	1	0	0	0	1	0	1	0	1	0	1
2	0	2	0	0	0	0	1	0	1	0	1	0
3	0	3	0	1	0	0	0	1	0	1	0	1
4	1	0	0	1	1	0	0	0	1	0	1	0
5	1	1	0	1	0	1	0	0	0	1	0	1
6	1	2	0	1	1	0	1	0	0	0	1	0
7	1	3	0	1	0	1	0	1	0	0	0	1
8	2	0	0	0	1	0	1	0	1	0	0	0
9	2	1	0	1	1	1	0	1	0	1	0	0
10	2	2	0	0	0	1	1	0	1	0	1	0
11	2	3	0	1	0	0	1	1	0	1	0	1
12	3	0	0	1	1	0	0	1	1	0	1	0
13	3	1	0	1	0	1	0	0	1	1	0	1
14	3	2	0	1	1	0	1	0	0	1	1	0
15	3	3	0	1	0	1	0	1	0	0	1	1
16	4	0	0	0	1	0	1	0	1	0	0	1
17	4	1	0	0	1	1	0	1	0	1	0	0
18	4	2	0	1	1	1	1	0	1	0	1	0
19	4	3	0	1	0	1	1	1	0	1	0	1
20	5	0	0	1	1	0	1	1	1	0	1	0
21	5	1	0	1	0	1	0	1	1	1	0	1
22	5	2	0	1	1	0	1	0	1	1	1	0
23	5	3	0	1	0	1	0	1	0	1	1	1
24	6	0	0	1	1	0	1	0	1	0	1	1
25	6	1	0	1	0	1	0	1	0	1	0	1
26	6	2	0	0	1	0	1	0	1	0	1	0
27	6	3	0	0	1	1	0	1	0	1	0	1
28	7	0	0	1	1	1	1	0	1	0	1	0
29	7	1	0	1	0	1	1	1	0	1	0	1
30	7	2	0	1	1	0	1	1	1	0	1	0
31	7	3	0	1	0	1	0	1	1	1	0	1
32	8	0	0	1	1	0	1	0	1	1	1	0
33	8	1	0	0	0	1	0	1	0	1	1	1
34	8	2	0	1	0	0	1	0	1	0	1	1
35	8	3	0	0	1	0	0	1	0	1	0	1
36	9	0	0	1	1	1	0	0	1	0	1	0
37	9	1	0	1	0	1	1	0	0	1	0	1
38	9	2	0	1	1	0	1	1	0	0	1	0
39	9	3	0	1	0	1	0	1	1	0	0	1
40	10	0	0	1	1	0	1	0	1	1	0	0
41	10	1	0	0	0	1	0	1	0	1	1	0
42	10	2	0	0	0	0	1	0	1	0	1	1
43	10	3	0	1	0	0	0	1	0	1	0	1
44	11	0	0	1	1	0	0	0	1	0	1	0
45	11	1	0	1	0	1	0	0	0	1	0	1
46	11	2	0	1	1	0	1	0	0	0	1	0
47	11	3	0	1	0	1	0	1	0	0	0	1
48	12	0	0	1	1	0	1	0	1	0	0	0
49	12	1	0	-	0	1	0	1	0	1	0	0
50	12	2	1	-	-	0	1	0	1	0	1	0

Fig. 4. Explanation how the XMAFM sequence is generated for an example network with four (4) interconnections

V. Properties of the (2*n*-1)-SR-TPG generator

Selected parameters of the (2*n*-1)-SR-TPG circuit and the MAFM and XMAFM test sequences generated with use of it are summarized in Table I. All figures in that table are expressed as functions of the *n* parameter. The 2nd and 3rd columns of the table specify respectively the actual length *m* of the test vector sequence generated by the specific circuit and the m/m_{min} ratio of the actual length to its minimum theoretical limit. The columns 4 and 5 contain sizes l_1 and l_0 for respective TPC1 and TPC0 counters that are used for the (2*n*-1)-SR-TPG circuit that is employed to produce the test sequence with the length of *m*. The combinations between the contents of the TPC1 and TPC0 counters when EN=0, EN=-, EOT=1 and EOT=- are provided in columns from 6 to 9. These combinations are written either as ordered pairs (Cartesians), i.e ⟨*TPC*1, *TPC*0⟩ or as a set of logic conditions that are imposed to the content of one or both counters. For instance, the notation ⟨5,*n*-1⟩ in column 6 of the table stands for the expression that for the TPC1 content of 5 (*TPC*1=5) and, simultaneously, the content of the *TPC*0 counter equal to *n*-1, the EN signal is low (EN = 0). Similarly, the logic expression "*TPC*1=5 & *TPC*0>2, *TPC*1>5" that can be found

TABLE I
SELECTED PARAMETERS OF THE (2n-1)-SR-TPG CIRCUIT AS WELL AS THE MAFM AND XMAFM TEST SEQUENCES GENERATED BY THAT CIRCUIT EXPRESSED AS FUNCTION OF THE n PARAMETER

1	2	3	4	5	6	7	8	9	10
Test sequence	m	m / m_{min}	l_1	l_0	The combinations between contents of the TPC1 and TPC0 counters, for which				CNT initial state
					EN = 0	EN = -	EOT = 1	EOT = -	
MAFM	8n+1	(8n+1) / 6n	4	$\lceil \log_2(n) \rceil$	⟨0,0⟩, ⟨1,n-2⟩, ⟨2,0⟩, ⟨4,0⟩, ⟨4,1⟩, ⟨5,n-1⟩, ⟨6,1⟩	*TPC*1=7 & *TPC*0>n-1, *TPC*1>7	⟨0,0⟩	*TPC*1=8 & *TPC*0>0, *TPC*1>8	0
XMAFM	12n+3	(12n+3) / (6n+3)	3	$\lceil \log_2(n)+1 \rceil$	⟨0,1⟩, ⟨0,2⟩, ⟨1,0⟩, ⟨1,2⟩, ⟨2,0⟩, ⟨2,1⟩, ⟨3,2⟩, ⟨3,3⟩, ⟨4,1⟩, ⟨4,3⟩, ⟨5,1⟩, ⟨5,2⟩	*TPC*1=6 & *TPC*0=0, *TPC*1>6	⟨5,2⟩	*TPC*1=5 & *TPC*0>2, *TPC*1>5	1

in the bottom line of the column 9 informs that the EOT signal can adopt the “don’t care” (irrelevant) level when the content of the TPC1 counter is five (5) and, at the same time, the content of the TPC0 counter is more than two (2) and, in any case, when the content of the TPC1 counter is more than five (5), regardless the content of the TPC0 counter. The rightmost column (#10) of Table I contains information about the initial content for the CNT counter that is necessary to start its operation to enable the (2n-1)-SR-TPG circuit to correctly produce the desired sequence of either MAFM or XMAFM vectors. Finally, the numerical values of the m, m/m_{min} and l parameters for n = 8, 16, 32, 64, 128, 256, 512 and 1024 are disclosed respectively in columns 3, 4 and 5 of Table II.

The analysis of contents from Tables I and II enables formulation of important findings. The length for the both test sequences in question, i.e. MAFM and XMAFM is linearly dependent on the size n of the Test Pattern Generator. Thus, the proposed TPG structure is easily scalable. On the other hand, for the values of n adopted for the considered example the MAFM test sequence generated by the (2n-1)-SR-TPG circuits is from 1.33 to 1.35 times longer than the minimum possible limit for that tests sequence. Similarly, the XMAFM test sequence is from 1.94 to 2 times longer than the minimum one.

In order to estimate the hardware overhead associated with implementation of the proposed Test Pattern Generator the next step consisted in development of a parameterized and synthesizable model of the (2n-1)-SR-TPG circuit in the VHDL language, where the type of the test sequence (MAFM or XMAFM) and the TPG size (n) were the parameters. Then the synthesis of the model was carried out with use of the TSMC 180 nm standard cell library for the both types of test sequences and for the figures of n as specified in Table II.

The K_{TPG} cost for Test Pattern Generators obtained as the result of the synthesis, expressed as a number of equivalent 2-input NAND gates is provided in column 6 of Table II. The K_{TPG} parameter comprises chiefly the cost of the (2n-1)-SR register, the TPC and CNT counters and the DEC decoder.

It also includes the cost of multiplexers that facilitate integration of the (2n-1)-SR-TPG circuit with components of the scan path as well as the transmission bus that is to be tested with use of the TPG in question.

The column 7 of the mentioned table specifies the cost of the (2n-1)-SR-TPG circuit brought down to each single line of the interconnection network under test. It must be noted that the cost per a single line decreases in pace with growth of the number of n. Therefore the Test Pattern Generator proposed in this paper is an easily scalable solution in terms of implementation cost.

It is also worth to mention that a great deal of digital IP cores is typically provided with a scan path (SP) and Built-in Self-Test (BIST) hardware. Moreover, the process of matching such cores with a SoC frequently assumes that cells of the

TABLE II
NUMERICAL VALUES FOR SELECTED PARAMETERS OF THE (2n-1)-SR-TPG STRUCTURE AS WELL AS THE MAFM AND XMAFM TEST SEQUENCES GENERATED BY SUCH A STRUCTURE

1	2	3	4	5	6	7	8	9	10	11
Test sequence	n	m	m / m_{min}	l_0	K_{TPG}	K_{TPG} / n	K_{CNT}	K_{DEC}	K_{CNT} + K_{DEC}	f_{max}
MAFM	8	65	1,35	3	142	17,8	9	15	24	854
	16	129	1,34	4	223	13,9	9	17	26	905
	32	257	1,34	5	376	11,8	9	19	28	858
	64	513	1,34	6	669	10,5	9	20	29	788
	128	1025	1,33	7	1245	9,7	9	22	31	788
	256	2049	1,33	8	2389	9,3	9	24	33	767
	512	4097	1,33	9	4666	9,1	9	28	37	743
	1024	8193	1,33	10	9207	9,0	9	30	39	638
XMAFM	8	99	1,94	4	147	18,4	9	19	28	790
	16	195	1,97	5	228	14,3	9	21	30	818
	32	387	1,98	6	378	11,8	9	21	30	773
	64	771	1,99	7	671	10,5	9	22	31	770
	128	1539	2,00	8	1246	9,7	9	23	32	747
	256	3075	2,00	9	2389	9,3	9	23	32	647
	512	6147	2,00	10	4665	9,1	9	26	35	643
	1024	12291	2,00	11	9202	9,0	9	24	33	629

Wrapper Boundary Register (WBR) are added to such a core that enables conformity with the IEEE 1500 standard. In such a case the (2n-1)-SR register that is the key part of the Test Pattern Generator can be implemented as a fragment of the scan path or the Wrapper Boundary Register. The only thing that is necessary consists in appropriate settings for the software tool that is used to implement the scan path within the IP core or the Wrapper Boundary Register at its outputs to accordingly connect the SP or WBR cells that shall be incorporated into the (2n-1)-SR register to arrange them in the appropriate order. In turn, the Test Pattern Counter can be simply implemented on the basis of the test vector counter that is usually already available within the BIST structure of any digital IP core. If so, the only surplus components that have to be added to implement the Test Pattern Generator of the (2n-1)-SR-TPG type are the DEC decoder and a one-bit CNT counter. The individual costs for such components, denoted respectively as K_{CNT} and K_{DEC}, are provided in columns 8 and 9 of Table II. The column 10 of the table presents the totalized cost of the both components that never exceeds several dozens of equivalent NAND gates. In the case under consideration that surplus is actually the only additional cost associated with implementation of the Test Pattern Generator of the (2n-1)-SR-TPG type. Being aware how sophisticated the IP cores incorporated into modern SoCs can be the hardware overhead amounting to several dozens of gates should be considered as negligibly insignificant.

The column 11 of Table II specifies the estimated maximum frequency rate of the (2n-1)-SR-TPG module for various values of the n parameter.

VI. Comparison between the (2n-1)-SR-TPG Circuit and Other Solutions

Table III provides comparison between the costs of the Test Pattern Generator of the (2n-1)-SR-TPG type and capable of generating the MAFM sequence against the solutions that are disclosed in studies [37] and [11]. The authors of both of these works have developed test patterns generators with a similar structure. Each of the TPGs contains a finite state machine that is responsible for providing the appropriate logic values for the currently selected victim line and other connections, acting as the aggressor lines. Selection of the victim line and aggressor lines is done via a network of multiplexers, which are controlled by a counter and decoder.

The column 1 of Table III provides the size n of the Test Pattern Generator that is equal to the width of the data transmission bus under test, i.e. $n = v$. The column 2 comprises estimation of cost for the Test Pattern Generator of the (2n-1)-SR-TPG type expressed as the number of equivalent 2-input NAND gates. The cost of TPGs disclosed in [37] and [11] is specified respectively in the 3rd and the 5th column of the table. In turn, columns 4 and 6 present the benefit that is understood as the percentage of the hardware overhead reduction due to application of the Test Pattern Generator of the (2n-1)-SR-TPG type instead of the solutions disclosed respectively in reference studies. All in all, the hardware overhead achieved from implementation of the proposed (2n-1)-SR-TPG structure is from 17,6% up to as much as 35,2% less than in case of TPGs revealed in referenced studies [37], [11].

TABLE III
COMPARISON BETWEEN IMPLEMENTATION COSTS FOR THE TEST PATTERN GENERATOR OF THE (2n -1)-SR-TPG TYPE AND OTHER SOLUTIONS

1	2	3	4	5	6
n	(2n-1)-SR-TPG	[37]		[11]	
8	142	219	35,2%	–	–
16	223	334	33,2%	287	22,3%
32	376	525	28,4%	471	20,2%
64	669	–	–	812	17,6%

Nevertheless, the MAFM test sequences supplied by the (2n-1)-SR-TPG circuit are from 33% to 35% longer that the ones achieved in [37]. However, they are longer by only 3 clock periods than the ones described in [11] under the assumption of the same set of stimulated faults.

Further considerations are provided for comparison between operation of the (2n-1)-SR-TPG circuit and the solution already disclosed in [28]. To generate test vectors required to stimulate crosstalk faults in an n-bit interconnection bus study [28] assumes application of a Linear Feedback Shift Register (LFSR) with the length of 2n bits (2n-LFSR). Actually the 2n-LFSR is made up of a shift register with its length of 2n supplemented with an external linear feedback with only XOR gates. The study [28] considers the buses with the width (in bits) equal to $n = 8$. The considerations take account of crosstalk faults of the Pg0, Ng1, Dr and Df types that may occur in the victim line due to the impact of k aggressor lines, where $k = 1, 2, 3, 4$. To stimulate each of the aforementioned crosstalk faults the paper assumes to use pairs of subsequent (adjacent) vectors from the MAFM sequence with the vector size of k+1 bits. Every vector of such a pair contains k+1 bits out of n bits of the vector included into a pseudo-random sequence produced at odd outputs of the 2n-LFSR register. The study provides the experimental proof that simulation of the 2n-LFSR register operation starting from a randomly selected initial content of that register makes it possible to find out the test sequence with the acceptable length in less than 100 trials and such a sequence shall provide complete coverage for crosstalk faults in question. Nevertheless, for the solution disclosed in this study, the number of aggressor lines is always $k = n$–1.

TABLE IV
COMPARISON OF THE LENGTH OF TEST SEQUENCES PRODUCED BY THE 2n -LFSR REGISTER AND THE TPG OF THE (2n -1)-SR-TPG TYPE

1	2	3	4	5	6
$n = v$	2n-LFSR				(2n-1)-SR-TPG
	$k = 2$	$k = 3$	$k = 4$	$k = 5$	k = n-1
8	34	205	986	4478	65
12	42	233	1406	6454	97
16	50	305	1410	8177	129
20	56	351	1840	9191	161
24	60	366	2099	10038	193
28	70	403	2192	11378	225
32	72	449	2338	12466	257

The comparison for lengths of test sequences generated by the 2n-LFSR and the (2n-1)-SR-TPG circuits is provided in Table IV. The first (leftmost) column of that table contains information about the size n of the bus under test. The

TABLE V
COMPARISON OF THE AREA OVERHEAD AND MAXIMUM OPERATING FREQUENCY
FOR THE TEST PATTERN GENERATORS OF THE 2*n*-SR-TPG AND (2*n* -1)-SR-TPG TYPE

1	2	3	4	5	6	7	8
Test sequence	*n*	Area overhead		Area reduction	$f_{\max}$ [MHz]		Frequency increase
		2*n*-SR-TPG [9]	(2*n*-1)-SR-TPG		2*n*-SR-TPG [9]	(2*n*-1)-SR-TPG	
MAFM	8	152	142	6,58%	743	854	14,9%
	16	229	223	2,62%	794	905	14,0%
	32	382	376	1,57%	728	858	17,9%
	64	679	669	1,47%	635	788	24,1%
	128	1259	1245	1,11%	551	788	43,0%
	256	2392	2389	0,13%	544	767	41,0%
	512	4669	4666	0,06%	539	743	37,8%
	1024	9214	9207	0,08%	470	638	35,7%
XMAFM	8	157	147	6,37%	759	790	4,1%
	16	241	228	5,39%	798	818	2,5%
	32	392	378	3,57%	748	773	3,3%
	64	698	671	3,87%	580	770	32,8%
	128	1274	1246	2,20%	554	747	34,8%
	256	2402	2389	0,54%	539	647	20,0%
	512	4684	4665	0,41%	545	643	18,0%
	1024	9223	9202	0,23%	474	629	32,7%

columns from 2 to 5 provide lengths of sequences generated by the 2*n*-LFSR circuit respectively for k = 1, 2, 3 and 4. The last (rightmost) column 6 presents information related to lengths of test sequences produced by the Test Pattern Generator of the (2*n*-1)-SR-TPG type that is suggested in this study. The data in Table IV demonstrate that exclusively for $k = 1$ the linear register of the 2*n*-LFSR type is capable of generating shorter test sequences to completely cover all crosstalk faults in question than it is in the case when the (2*n*-1)-SR-TPG circuit is employed. For k = 2, 3 and 4 the test sequences produced by the 2*n*-LFSR register are respectively several, more than dozen or several dozens times longer than the ones obtained at outputs of the (2*n*-1)-SR-TPG circuits. In addition, use of the TPG of the (2*n*-1)-SR-TPG type for detection of crosstalk faults in interconnection lines has one more substantial advantage as compared to application of the 2*n*-LFSR register for the same purpose. Supplying of the MAFM test sequence produced by the (2*n*-1)-SR-TPG circuit leads to dissipation of much less power in the interconnections under test than it takes places in case of a pseudo-random test sequence produced by the 2*n*-LFSR register [29].

Unfortunately, the study [28] provides neither information about hardware overhead for the solution disclosed therein nor its maximum operation frequency. Therefore it is impossible to directly compare these parameters for the 2*n*-LFSR register and for the (2*n*-1)-SR-TPG circuit. But even at a glance onto design schematics of the both TPGs circuits it is possible to assume that the hardware overhead added by the both structures is comparable. Nevertheless it is necessary to note that the 2*n*-LFSR register has longer lines for the global feedback that imposes constrains for the maximum frequency of its operation. On the contrary, the TPG of the (2*n*-1)-SR-TPG type can be designed in such a manner that all its connections would be short. It is why the maximum operation frequency for the (2*n*-1)-SR-TPG circuit should be higher than the one for the 2*n*-LFSR, particularly for high values of *n*.

The Test Pattern Generator of the (2*n*-1)-SR-TPG type was compared against the 2*n*-SR-TPG circuit disclosed in [9]. Information related to the hardware overhead added by the both structures is summarized in Table V. The first (leftmost) column of the table specifies the type of the test sequence produced by each TPG type. The second column provided information about the size of the interconnecting bus under test. Then the columns 3 and 4 comprise figures related to the overall chip area occupied respectively by the 2*n*-SR-TPG and (2*n*-1)-SR-TPG circuits, where that area is expressed as the number of equivalent two-input NAND gates. The gain in the TPG area achieved by application of the (2*n*-1)-SR-TPG circuits instead of the 2*n*-SR-TPG one and expressed in percents is listed in column 5. In turn, information about the maximum operation frequency for the both TPGs can be found in columns 6 and 7. Finally, the last (rightmost) column (#8) of the table informs how much faster (in percents) the (2*n*-1)-SR-TPG circuit operates as compared to the 2*n*-SR-TPG one.

The improvements that make it possible to convert the 2*n*-SR-TPG circuit into the one of the (2*n*-1)-SR-TPG type include reduction of the register length by one bit and the structure rearrangement for the counter of test vectors. Such amendments have led to slight reduction of the chip area occupied by the test pattern generator. The maximum gain for that reduction at the level of 6.4 to 6.8% was achieved for the TPGs with the minimum size of n = 8 bits. However, for TPGs that are used to generate test sequences for buses with the size more or equal to 256 bits the savings on the TPG area is below 1%, i.e. it is negligible.

On the other hand one has to note that the new structure for Test Pattern Generators disclosed in this paper and referred to as the (2*n*-1)-SR-TPG type demonstrates much higher clock frequency for the circuit operation as compared to the 2n-SR-TPG circuits. It is really important since the gain on the operation frequency is sometimes as high as 43%.

VII. CONCLUSIONS

The study proposes an innovative and original structure of a Test Pattern Generator dedicated for detection of crosstalk faults that may happen to bus-type interconnections between embedded IP cores within a digital SoC. The structure, referred to the (2*n*-1)-SR-TPG circuit is designed to supply the MAFM or XMAFM test sequences that guarantee detection of all possible crosstalks of capacitive nature between individual data transmission lines.

The TPG structure discussed in this study is easily scalable both in terms of additional hardware overhead and the length of the test sequence provided by the generator. The hardware overhead associated with application of the Test Pattern Generator of the (2*n*-1)-SR-TPG type is substantially less, i.e. from 17,6% up to as much as 35,2% than in case of solutions disclosed in other studies [37], [11]. In addition, the maximum frequency for the circuit operation is higher by 2.5% up to 43% than for the circuit disclosed in [9]. Although the MAFM test sequences provided by the TPG of the (2*n*-1)-SR-TPG type are by as much as 35% longer than the ones reported in [37], for more than one aggressor line they are much shorter than in case of the solution revealed in [28].

The design of the (2*n*-1)-SR-TPG circuit enables also easy integration with a scan path or with cells of a Wrapper Boundary Register compliant with the IEEE 1500 standard and/or BIST solutions already implemented within certain IP cores of a SoC. In such a case the hardware overhead due to implementation of the proposed TPG structure never exceeds several dozens of simple gates, which is the negligibly insignificant number.

The (2*n*-1)-SR-TPG circuit is perfectly suitable for incorporation into the BIST architecture that is disclosed in [15]. In such a case the techniques proposed in [16] can be applied to localization and identification of crosstalk faults.

The MAFM and XMAFM test sequences guarantee stimulation of all possible crosstalks but only of the capacitive nature. Some studies, including [35], [31], [23] demonstrate that in case of inductive crosstalks application of test sequences different from the ones as presented above may lead to even longer delays of signal edges in the victim line or undesired pulses with higher amplitudes. Therefore, further efforts of the author shall be focused on IBIST structures that should benefit from new types of test sequences to enable stimulation of crosstalk faults of both capacitive and inductive nature.

References

[1] N. Ahmed, M. H. Tehranipour, M. Nourani, “Extending JTAG for testing signal integrity in SoCs”, in *Proc. Design, Automation and Test in Europe – DATE’03*, Munich, Germany, 2003, pp. 218-223.

[2] Aniket, R. Arunachalam, “A novel algorithm for testing crosstalk induced delay faults in VLSI circuits”, in *Proc. 18th Int. Conf. on VLSI Design*, Kolkata, India, 2005, pp. 479-484.

[3] X. Aragones, J. L. González, F. Moll, and A. Rubio, “Noise Generation and Coupling Mechanisms in Deep-Submicron ICs”, *IEEE Design & Test Computers*, vol.19, no.5, pp. 27-35, September-October 2002.

[4] A. Attarha, M. Nourani, “Testing interconnects for noise and skew in gigahertz SoCs”, in *Procs. of IEEE Int. Test Conf. – ITC’01*, Baltimore, MD, USA, 2001, pp. 305-314.

[5] L. Benini, G. De Micheli, “Networks on Chips: A New SoC Paradigm”, *Computer*, vol. 35 no. 1, pp. 70-78, January 2002.

[6] M. Botelho, F. L. Kastensmidt, M. Lubaszewski, E. Cota, L. Carro, „A Broad Strategy to Detect Crosstalk Faults in Network-on-Chip Interconnects”, in *Proc. 2010 18th IEEE/IFIP Int. Conf. on VLSI and System-on-Chip*, Madrid, Spain, 2010, pp. 298-303.

[7] M. Cuviello, S. Dey, Xiaoliang Bai, Yi Zhao, “Fault modeling and simulation for crosstalk in system-on-chip interconnects”, in *Proc. 1999 IEEE/ACM Int. Conf. on Computer-Aided Design*, San Jose, CA, USA, 1999, pp. 297-303.

[8] W. J. Dally, B. Towles, “Route Packets, Not Wires: On-Chip Interconnection Networks”, in *Procs. 38th IEEE/ACM Design Automation Conference – DAC’01*, Las Vegas, NV, USA, 2001, pp. 683-689.

[9] T. Garbolino, “Designing of test pattern generators for stimulation of crosstalk faults in bus-type connections.”, *Proc. 2014 IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits and Systems – DDECS’2014*, Warsaw, Poland, April 23-25, 2014, pp. 270-273.

[10] C. Grecu, P. P. Pande, A. Ivanov, R. Saleh, "Structured Interconnect Architecture: A Solution for the Non-Scalability of Bus-Based SoCs”, in *Proc. Great Lakes Symposium on VLSI – GLSVLSI’04*, New York, NY, USA, 2004, pp 192-195.

[11] C. Grecu, P. Pande, A. Ivanov, R. Saleh, “BIST for Network-on-Chip Interconnect Infrastructures”, in *Proc. 24th IEEE VLSI Test Symposium – VTS'06*, Berkeley, CA, USA, 2006, pp. 30-35.

[12] P. Guerrier, A. Greiner, ”A Generic Architecture for on chip Packet-switched Interconnections”, *Proc. Design, Automation & Test in Europe – DATE’00*, Paris, France, 2000, pp. 250-256.

[13] G. Haiyun, “Survey of Dynamically Reconfigurable Network-on-Chip”, in *Proc. 2011 Int. Conf. on Future Computer Sciences and Application - ICFCSA 2011*, Hong Kong, China, 2011, pp. 200-203.

[14] A. Jutman, “At-Speed On-Chip Diagnosis of Board-Level Interconnect Faults”, in *Procs. 9th IEEE European Test Symposium – ETS’04*, Ajaccio, France, 2004, pp. 2-7.

[15] M. Kopeć, T. Garbolino, K. Gucwa, A. Hławiczka, “Test-per-clock detection, localization and identification of interconnect faults”, in *Procs. 11th IEEE European Test Symposium – ETS’06*, Southampton, United Kingdom, 2006, pp. 233-238

[16] M. Kopeć, T. Garbolino, K. Gucwa, A. Hławiczka, “On application of polynomial algebra for *identification* of dynamic faults in interconnects”, *Electronics and Telecommunications Quarterly*, vol. 54, no. 1, pp. 29-41, 2008.

[17] S. Kumar, et al., “A Network on Chip Architecture and Design Methodology,” *in Procs. IEEE Computer Society Annual Symposium on VLSI*, Pittsburgh, PA, USA, 2002, pp. 117-124.

[18] K. Shu-Min Li; Chung Len Lee; Chauchin Su; J.E. Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI”, in Procs. *13th IEEE Asian Test Symposium*, Kenting, Taiwan, 2004, pp. 145-150

[19] K.S.-M. Li, Yi-Yu Liao, Yuo-Wen Liu, Jr-Yang Huang, “IEEE 1500 Compatible Interconnect Test with Maximal Test Concurrency”, in *Procs. 18th IEEE Asian Test Symposium – ATS 2009*, 2009, Taichung, Taiwan, pp. 269-274.

[20] P. Magarshack, P.G. Paulin, “System-on-Chip beyond the Nanometer Wall”, in *Procs. 40th IEEE/ACM Design Automation Conference – DAC’03*, Anaheim, USA, 2003, pp. 419-424.

[21] P. Nordholz, D. Treytnar, J. Otterstedt, H. Grabinski, D. Niggemeyer, T.W. Williams, “Signal Integrity Problems in Deep Submicron arising from Interconnects between Cores”, in *Procs. IEEE VU1 Test Symposium*, 1998, pp. 28-33.

[22] M. Nourani, A. Attarha, “Built-In Self-Test for Signal Integrity”, in *Proc. 38th IEEE Design Automation Conference – DAC 2001*, Las Vegas, NV, USA, 2001, pp. 792-797.

[23] R. Nourmandi-Pour, A. Khadem-Zadeh, A. M. Rahmani, “An IEEE 1149.1-based BIST method for at-speed testing of inter-switch links in network on chip”, *Elsevier Journal of Microelectronics*, vol. 41, no. 7, pp. 417-429, July 2010.

[24] R. Nourmandi-Pour, N. Mousavian, “A fully parallel BIST-based method to test the crosstalk defects on the inter-switch links in NOC”, *Microelectronics Journal*, vol. 44, no. 3, pp. 248-257, March 2013.

[25] P. Pande, C. Grecu, A. Ivanov, R. Saleh, G. De Micheli, “Design, synthesis, and test of networks on chips”, *IEEE Design & Test of Computers*, vol. 22, no. 5, pp. 404-413, 2005.

[26] R. Pendurkar, A. Chatterjee, Y. Zorian, “Switching Activity Generation with Automated BIST Synthesis for Performance Testing of Interconnects”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 9, pp. 1143-1158, 2001.

[27] Y.-L. Peng, C.-W. Wu, J.-J. Liou, C.-T. Huang, “BIST-based diagnosis scheme for field programmable gate array interconnect delay faults”, *IET Computers & Digital Techniques*, vol. 1, no. 6, pp. 716-723, November 2007.

[28] T. Rudnicki, T. Garbolino, K. Gucwa, A. Hławiczka, “Effective BIST for crosstalk faults in interconnects”, in *Proc. 2009 IEEE Symposium on*

Design and Diagnostics of Electronic Circuits and Systems. DDECS'2009, Liberec, Czech Republic, 2009, pp. 164-169.

[29] K. Sekar, S. Dey, "LI-BIST: A Low-Cost Self-Test Scheme for SoC Logic Cores and Interconnects", in *Journal of Electronic Testing: Theory and Applications*, vol. 19, no. 6, pp. 113-123, 2003.

[30] A. Sinha, S. K. Gupta, M. A. Breuer, "Validation and test issues related to noise induced by parasitic inductances of VLSI interconnects", *IEEE Transactions on Advanced Packaging*, vol. 25, no. 3, pp. 329–339, August 2002.

[31] Sunghoon Chun, Yongjoon Kim and Sungho Kang, "MDSI: Signal Integrity Interconnect Fault Modeling and Testing for SoCs", *Journal of Electronic Testing: Theory and Applications*, vol. 23, no. 4, pp. 357-362, August 2007.

[32] C. Su, W. Tseng, "Configuration free SoC interconnect BIST methodology", in Procs. *IEEE International Test Conference, "Tackling Test Tradeoffs"*, Baltimore, MD, USA, 2001, pp. 1033-1038.

[33] O. Tayan, "Networks-on-Chip: Challenges, trends and mechanisms for enhancements", in *Proc. The 3rd International Conference on Information and Communication Technologies – ICICT'09*, Karachi, Pakistan, 2009, pp. 57-62.

[34] M. H. Tehranipour, N. Ahmed, M. Nourani, "Multiple transition model and enhanced boundary scan architecture to test interconnects for signal integrity", in *Proc. 21st Int. Conf. on Computer Design*, San Jose, CA, USA, 2003, pp. 554-559.

[35] M. H. Tehranipour, N. Ahmed, M. Nourani, "Testing SoC interconnects for signal integrity using extended JTAG architecture", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 5, pp. 800-811, May 2004.

[36] C. Tsai, J. Wu, C. Kao, T. Lee, R. Hsiao, "Coupling-aware RLC-based clock routing for crosstalk minimization", in *Proc. 2006 IEEE International Symposium on Circuits and Systems – ISCAS'06*, Island of Kos, Greece, 2006, pp. 497-500.

[37] Xiaoliang Bai, S. Dey, J. Rajski, "Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects, in *Proc. 2000 37th IEEE Design Automation Conference – DAC'00*, Los Angeles, CA, USA, 2000, pp. 619-624.

[38] H. Zhou, D. F. Wang, "Global routing with crosstalk constraints", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 11, pp. 1683-1688, November 1999.