Hybrid BIST for System-on-a-Chip Using an Embedded FPGA Core

Gang Zeng† and Hideo Ito††
† Graduate School of Science and Technology, Chiba University, Japan
†† Faculty of Engineering, Chiba University, Japan
E-mail: sogo@graduate.chiba-u.jp h.ito@faculty.chiba-u.jp

Abstract

In this paper, a novel hybrid built-in self-test (BIST) approach for system-on-a-chip (SOC) test using an embedded FPGA core is presented. The hybrid BIST combining pseudorandom test with deterministic test can achieve not only complete fault coverage but also minimal test cost by selecting the appropriate number of pseudorandom patterns. Most importantly, the FPGA-based hybrid BIST has minimal hardware overhead, since after testing, the FPGA core can be reconfigured as normal mission logic. Experimental results for ISCAS 89 benchmarks and a platform FPGA chip have proven the efficiency of the proposed approach.

1. Introduction

As the complexity of system-on-a-chip (SOC) increases, it results in larger volume of test data, higher operation speed, more difficulty for accessing various embedded intellectual property (IP) through limited SOC I/O pins [1]. However, the I/O channel capacity, speed, and data memory of traditional automatic test equipment (ATE) are limited such that it leads to high test cost. Therefore, new design-for-test (DFT) techniques are required to reduce test cost.

Recently, a hybrid ASIC and FPGA trend is emerging [3-5]. The idea is to embed a programmable logic core (also known as FPGA) into a general ASIC SOC. The use of embedded FPGA will expand SOC design flexibility, reduce design risk and lengthen SOC product lifetimes significantly by allowing devices to adapt to changing standards and extra features to be added over time [3]. Thus, this approach is suitable for the platform-based SOC design and the embedded FPGA core is an ideal vehicle for infrastructure IP [2]. Examples of commercial product include Triscend's E5, A7, IBM's Cu-08 family etc.

In this paper, we propose a novel test approach for SOC using the embedded FPGA core. First, it is assumed that the embedded FPGA core has passed all structural tests (i.e., it is fault free), and then the FPGA core is configured as hybrid BIST. Finally, it is used for testing other cores in the SOC. The hybrid BIST can achieve minimal test cost by selecting the appropriate number of pseudorandom patterns that can reduce the total test length. Moreover, the proposed fixing-flipping method for deterministic test pattern generator (DTPG) need not change the order of test sequence, thus it can support delay test. At the same time, fixing the don't care bits to 1 or 0 will lead to lower scan test power. Most importantly, since the approach is based on the reconfigurable FPGA core, it can achieve minimal hardware overhead. After testing, the FPGA core can be reconfigured as normal mission logic.

The rest of the paper is organized as follows. Section 2 describes previous work. Section 3 presents the architecture of the embedded tester using an FPGA core. Section 4 discusses how to select the appropriate number of pseudorandom patterns for hybrid BIST. Section 5 presents the proposed method to generate the deterministic patterns. In section 6 the experimental results are provided. Finally, section 7 concludes the paper.

2. Previous work

Logic BIST is generally based on pseudorandom pattern (PRP). Most circuits, however, have inherent random pattern resistance (r.p.r), which results in incomplete fault coverage (FC) within reasonable test time [11]. For this reason, the classical approaches such as test point insertion and weighted random pattern method are applied to improve FC. However, these methods still cannot obtain complete FC. In recent years, several techniques have been proposed to achieve complete FC. One method is known as "reseeding technique" [12-13] in which the conventional LFSR is used to generate deterministic test pattern by loading it with a precomputed seed. Another method is known as "deterministic BIST" [17-20]. The method is to identify patterns in the PRP that do not detect any new faults and map them by dedicated hardware into deterministic patterns. The third method is an encoding-decoding method by combining ATE with decoder or BIST [14-16]. The deterministic test cubes are encoded and stored in ATE. During test, the test data is

Proceedings of the 22nd IEEE VLSI Test Symposium (VTS 2004)
1093-0167/04 $20.00 © 2004 **IEEE**

decoded by an on-chip decoder or BIST. In addition, techniques for using embedded processor to generate deterministic test pattern have also been explored [13-14]. The approach has the advantage of low hardware overhead. Although all these methods can achieve complete FC, some implementations can be improved in terms of test cost and implementing performance as follows.

(1) In order to reduce the hardware overhead or improve the compression, the method that embeds deterministic cube into pseudorandom pattern [14-20], generally needs to apply a large pseudorandom test set (e.g. over 10000 patterns). Nevertheless, most of PRPs cannot detect any new faults and result in long test time as well as increased power consumption. At the same time, the method cannot guarantee the order of test pattern after they have been embedded in PRP. Therefore, it is not applicable for two-pattern test, which is typically used for delay test.

(2) The method that generates deterministic test pattern with on-chip logic [12] [17-20] suffers from the low design flexibility. Since the on-chip test logic is test set-dependent, once the test set is changed, the on-chip logic needs to be redesigned. Nevertheless, when the chip has been fabricated, the redesign is impossible for conventional ASIC chip.

(3) From literature [20], it is demonstrated that when applying deterministic BIST to industry circuit, the area overhead ranges from 10% to 30% for both BIST and scan chain implementation. In some cases the overhead is too large to be accepted, furthermore, the paper did not present test method for BIST logic itself that occupies 5% to 15% volume of hardware.

(4) As discussed in paper [7], the method that employs an embedded processor [13-14] on test pattern generation has the drawbacks of the low efficiency and high-energy consumption in comparison with configurable logic-based implementation.

3. Embedded tester using an FPGA core

First, we show the characteristics of state-of-the-art FPGA cores, and then give a conceptual architecture for embedded tester; finally, the proposed hybrid BIST architecture is presented.

3.1. Characteristics of embedded FPGA core

The embedded FPGA core generally contains sufficient configurable logic cells, flip-flops as well as configurable I/O resources. Unlike conventional FPGA chip, the embedded RAM blocks are excluded from FPGA core to reduce the required area when embedding the FPGA core into an ASIC chip. We demonstrate the characteristics of FPGA core with the products of Triscend and Xilinx.

Triscend's Embedded Programmable Logic Core is similar to mainstream SRAM-based FPGA, with IBM 0.18um process technology node. The features of standard products are shown in Table 1. The configurable system logic (CSL) is the basic unit. Each CSL consists of a 4-input lookup table coupled with a flip-flop. Each bank composed of an 8 x 8 array of CSL pairs. As the Table shows, Tricend's FPGA core provides system gate ranging from 25k to 150k. In Table 1, the column of max I/O and max general signal port indicate the maximal I/O pad connection with external circuit and the maximal signal port connection with internal circuit respectively.

Table 1. Standard configurable system logic IP core products [10]

Size in Banks	CSL Cells	System Gates	Flip-Flops	Max. I/O	Max. General Signal Ports
2 X 2	512	25000	512	128	512
3 X 3	1152	60000	1152	192	768
4 X 4	2048	100000	2048	256	1024
5 X 5	3200	150000	3200	320	1280

From literatures [3] [9], Xilinx's FPGA core which is similar to its Virtex FPGA will be embedded in IBM's Cu-08 family ASIC products using IBM's 90nm ASIC technology. Three core sizes are planned, from 10k to 40k gates, with up to 640 I/Os. Multiple cores can be implemented within a single ASIC device. The available FPGA gates that can be used for design will range from 20k to 100k.

3.2. Hybrid BIST based on an FPGA core

Based on the FPGA core, we can build an embedded tester for SOC testing. Prior to applying this approach, the fault free FPGA core should be guaranteed. In [2] several BIST schemes are proposed for FPGA core test. Taking into account the structural testability of FPGA, if an FPGA core has passed all structural tests, the functional implementation configured after structural test can be considered as fault free. As mentioned above, an FPGA core with sufficient configurable logic resources and flip-flops is feasible to implement logic BIST. Furthermore, with the abundant configurable I/O resources an FPGA core has the flexibility in designing the test access mechanism (TAM) with other core or external ATE. As shown in Fig.1, the possible TAM includes bus-based TAM1 and dedicated TAM2. In [22], an FPGA core is configured as decoder and works with an external ATE in order to reduce test cost. In this paper, we use the FPGA core for implementing the hybrid BIST. As shown in Fig.1, the test strategy eliminates the need for expensive external tester, a low-cost ATE or PC-based ATE can be used to configure the FPGA core and collect test results. Thus, it can reduce further the test cost. At the same time without the limitation of I/O pin number and speed, the strategy can achieve higher test speed and test quality. During test, the FPGA core is configured as hybrid BIST, and then it delivers test patterns and receives test responses though the TAM, and finally, it sends the test results to the outside of

chip. If the SOC has passed all tests, then the FPGA core can be reconfigured as normal mission logic.

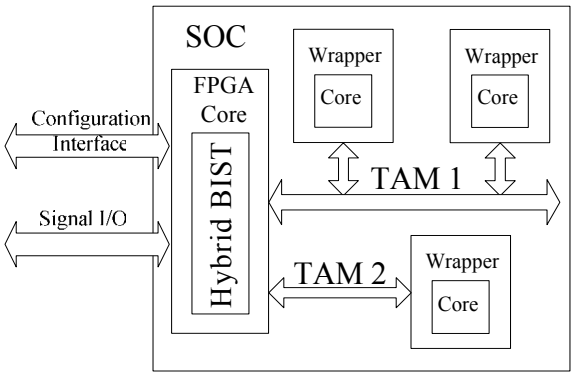

Fig.1 A Conceptual architecture for hybrid BIST

3.3. The proposed architecture of hybrid BIST

The proposed hybrid BIST mainly comprises of three parts as shown in Fig.2. One part is the conventional BIST that consists of LFSR, MISR (multiple input signature register) and BIST controller. Since conventional MISR cannot tolerate the X-states captured in scan cells [21], masking technique can be applied that is similar to the techniques used in [15]. The second part is deterministic test pattern (DTP) generator, denoted as bit fixing-flipping logic in Fig.2. The third part is a MUX used for selecting pseudorandom or deterministic test pattern for scan chain. As shown in Fig.2, the pattern counter (PC) and sequence counter (SC), which are utilized by conventional BIST controller [11], can be shared by BIST and DTPG to minimize the hardware. Although just one scan chain is shown in Fig 2, in practice, the method can also be applied to multiple scan chains.

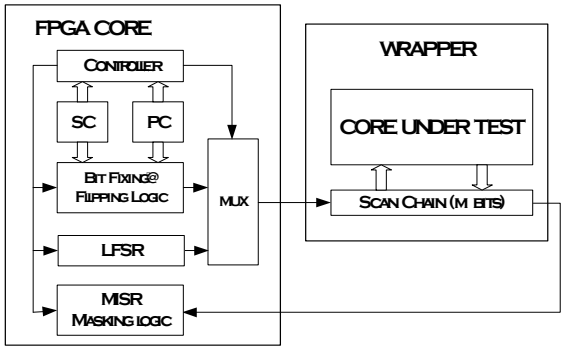

Fig. 2. Hybrid BIST architecture for core test

4. Selecting the appropriate number of pseudorandom patterns for hybrid BIST

As far as conventional hybrid BIST is concerned, the test cost mainly consists of the time cost and hardware cost. Considering the following cost function:

$$C_{TOTAL}=C_{TIME}+C_{HARDWARE}$$

Where C_{TIME} is the cost related to the time for generating the PRP and DTP that is mainly determined by the total length of PRPs plus DTPs. $C_{HARDWARE}$ is related to the hardware cost for implementing normal BIST and DTPG. During the test process of hybrid BIST, PRPs are firstly applied to detect "easy to detect" faults, then switch to DTPs to detect the remaining r.p.r faults. Generally, with the increase in the number of pseudorandom patterns, the hardware cost of DTPG will be reduced. As for the proposed FPGA-based hybrid BIST, so long as the required programmable resources do not exceed the available resources in targeted FPGA core, the hybrid BIST can achieve minimal hardware overhead, which includes the necessary routing overhead, compared with ASIC-based approach. Therefore, the main consideration with the proposed hybrid BIST is how to select the appropriate number of PRPs to achieve minimal test cost. There are two possibilities. In the first case, if there are sufficient programmable resources in targeted FPGA core for implementing the entire hybrid BIST, the selected PRP should help to obtain minimal test set, i.e., minimal test cost in this case. An example for s5378 is shown in Table 2. It is found from the simulation results that if we switch to DTP after applying 16 PRPs, we will obtain the minimal test set (i.e. total 64 patterns). We refer to this moment as the best switching point. The possible reason is that at the beginning of test, the full-specified random pattern is more efficient than the part-specified test cube in detecting the "easy to detect" faults. In other words, the 16 PRPs have the same function as the corresponding 25 test cubes in detecting faults. As a result, substituting the 25 test cubes with 16 PRPs will result in a reduction of 9 in the total number of test patterns. The main method for selecting the best switching point is fault simulation. Due to the efficient pseudorandom test set usually is small; the simulation time is acceptable. For example, the simulation results for ISCAS 89 benchmark show the PRP set is not larger than 32. While using a small pseudorandom test set can achieve minimal test cost, however, in the case of the required programmable resources exceed the available resources in targeted FPGA core, more pseudorandom patterns have to be applied to reduce the required hardware cost. Since the long test application time is acceptable in comparison with hardware overhead, particularly, for the proposed approach. Therefore, in this case the appropriate number of PRPs should meet the requirement that makes the implementation of entire hybrid BIST fit within the targeted FPGA core. As far as test cost is concerned, although the test time is longer than the first case, the total test cost still maintains a small value due to the minimal hardware cost and using a low-cost ATE or PC-base ATE. However, it should be noted that with the further increase

in the number of PRPs, the added PRP is less efficient than deterministic pattern in detecting remaining faults. For example, as shown in Table 2, among 10000 PRPs, just 217 of them are useful; most of them cannot detect any new faults, however, result in longer test time and more power consumption. Therefore, under the constraint of targeted FPGA core resources, the applied number of PRPs should be as small as possible.

Table 2. Hybrid test pattern for s5378

# PRP Pat.	# Pat. useful	FC (%)	#Deter. Pat.	#Total Pat.
0	0	0	73	73
1	1	23.42	71	72
4	4	50.17	63	67
8	8	63.82	58	66
16	**16**	**75.96**	**48**	**64**
32	31	81.69	45	77
64	53	87.58	42	108
100	80	91.02	35	135
1000	171	98.17	15	1015
10000	217	99.92	3	10003

5. Fixing-Flipping method for DTPG

The proposed fixing-flipping method for generating deterministic pattern, is based on the fact that an actual test set generated by automatic test pattern generation (ATPG) tool generally includes many of don't care bits. These don't care bits need not be constructed when recovering these patterns by hardware logic, since they can simply be fixed to 0 or 1. Thus, just care bits need to be generated by hardware logic. To minimize further the hardware logic for DTPG, we can make efficient use of the skew probability of occurring 1 or 0 in the same bit site for different test patterns. An observation can be made from a specific test set is that in the same bit site, most test patterns have the same logic value. The reason is that when generating test patterns to detect faults, ATPG just need to change some of the inputs, at the same time, remain most of the inputs to a constant value. Therefore, based on the number of 1 and 0 in each bit site for all test patterns, if we fix the bit to corresponding bits with more number, then just the remaining fewer bits need to be flipped. For example, in Fig.3 (a), the bit site for 30 test patterns include 20 care bits, i.e., 15 bits 0 and 5 bits 1, so if this bit site is fixed to 0, then just 5 of 30 test patterns need to flip the bit to 1 while the other test patterns remain the fixed 0. Similarly, in Fig.3 (b), if the bit site is fixed to 1, the number of flipping bits will be reduced to 3 bits. The fixing-flipping method reduces not only care bits but also the required hardware logic to generate the flipping bit. Besides this, the method also has some other benefits. Instead of filling the don't care bits to random value, the fixed 0 or 1 will significantly decrease the switch activity during shifting the test pattern into scan chains, which will lead to lower scan test power [6]. Furthermore, the fixing-flipping method need not change the order of test patterns when applying them to circuits under test; hence, it can also be used for two-pattern test.

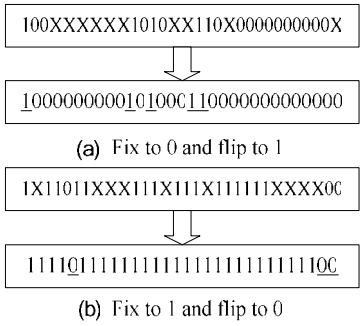

Fig. 3 An example for reduction in care bits

6. Experimental results

Experiments are performed to evaluate the reduction in test length and care bits as well as the hardware implementation of the proposed hybrid BIST approach.

6.1. Evaluation of reduction in test length and care bits

We perform experiments on the six largest ISCAS 89 full-scan benchmark circuits that include r.p.r faults. Since the length of pseudorandom test applied by the hybrid BIST is very short, the degree of the LFSR used in this approach can be very small (e.g., 16). A commercial ATPG tool, Synopsys's Tetramax, is used to generate test cube and perform fault simulation. The simulation results for the benchmarks with the minimal test sets and complete FC are listed in Table 3. The column of "deterministic test" gives the test results of applying deterministic patterns without PRP. The remaining columns in Table 3 show the test results of applying hybrid BIST with the minimal test set. In this part, the first four columns describe the length information of the minimal test set. The remaining columns denote the reduction in care bits. The percentage reduction in care bits is computed as:

(OriginalCareBits - FlippingBits)×100 / OriginalCareBits

The column of Dec^1 denotes the reduction in care bits obtained by fixing-flipping method in comparison with the deterministic Care Bit^1. The column of Dec^2 denotes the reduction in care bits obtained by hybrid BIST in contrast to the deterministic Care Bit^2. As can be seen, the application of a small pseudorandom test set (4 to 24 PRPs) can achieve average 18% reduction in the total test length by selecting the best switching point. In addition, the proposed fixing-flipping method for DTPG achieves average 75% reduction in care bits and the proposed hybrid BIST achieves average 81% reduction in care bit in contrast to the deterministic test without applying PRPs.

Table 3. Statistical results for the reduction in test length and care bits

Circuit	# Scan Cells	Deterministic Test		Hybrid BIST Test with Minimal Test Set							
		# Cube	# Deter. Care Bit[2]	# PRP Pat.	# Deter. Pat.	# Total Pat.	Dec.(%) Pat.	# Deter. Care Bit[1]	# Flip Bit	Dec[1].(%) Care Bit	Dec[2].(%) Care Bit
S5378	214	73	3961	16	48	64	12	2422	658	73	83
S9234	247	106	6055	8	88	96	9	5233	1299	75	79
S13207	700	185	9708	4	174	178	4	8491	1980	77	80
S15850	611	140	15601	8	106	114	19	13263	2470	81	84
S38417	1664	482	34578	24	191	215	55	28475	6413	77	81
S38584	1464	199	31735	16	155	171	14	22249	6728	70	79

In case of the programmable resources of targeted FPGA core cannot implement the entire DTPG, we can make a trade-off between the required resource and test length. As shown in Table 4, if we increase the PRP to 100 or 1000, the flipping bits, which need to be generated by FPGA core, will be decreased greatly. An observation can be made is that with the increase in the number of PRPs, the proposed fixing-flipping method can achieve more reduction in care bits. Therefore, not only the increased PRPs but also the fixing-flipping method contributes to the total reduction in care bits. Take s38584 for example, when PRPs are increased from 16 to 1000, the flipping bits will be decreased from 6728 to 1621 where the increased PRPs result in the reduction in care bit from 22249 to 7831, and fixing-flipping method results in further reduction in care bits from 7831 to 1621. The meanings of Dec^1 and Dec^2 in Table 4 are the same as in Table 3.

Table 4. Trade-off between flipping bits and test length for hybrid BIST test

Circuit	#PRP Pat.	# Deter. Pat.	# Deter. Care Bit[1]	# Flip Bit	Dec[1] (%)	Dec[2] (%)
S15850	8	106	13263	2470	81	84
	100	100	11671	1749	85	89
	1000	89	9566	1158	88	92
S38584	16	155	22249	6728	70	79
	100	133	15458	3972	74	87
	1000	96	7831	1621	79	95

6.2. Evaluation of the required hardware resources

It is difficult to perform experiments on an actual hybrid SOC chip since the hybrid ASIC and FPGA chips are still not be shipped to market in plenty until recently. Therefore, we select Xilinx's Virtex II pro platform family FPGA products [8] to evaluate the proposed fixing-flipping DTPG. Since, as described in section 3.1, we can assume that an FPGA core, which is similar to Virtex family FPGA, is embedded in an ASIC SOC chip. The platform family FPGA integrates RISC processor with Virtex II FPGA architecture. It contains system-level features that offer sufficient design flexibility, performance, and system integration that is fit for platform design. The feature parameters of platform family are depicted in Table 5 where XC2VP2 and XC2VP125 represent the available minimum and maximum resources provided by chips respectively. For platform family FPGA, each logic cell consists of a 4-input lookup table coupled with a flip-flop and carry logic; each CLB (configurable logic block) includes four slices and two 3-state buffers. We use Xilinx's ISE-WebPACK integration development software to implement the design. The DTPG with SC, PC counters and control logic is programmed in verilog. Prior to synthesis, we prohibit the function of extraction RAM and ROM, since, as described in 3.1, we assume that only logic and flip-flop resources are available in the targeted FPGA core. The DTPG is constructed based on the minimal hybrid BIST test set as shown in Table 3. The synthesis results are presented in Table 6. It can be seen that with the increase in flipping bits, the required FPGA resources also increase. Note that, since the selective FPGA chip contains minimum programmable resources, for large benchmark circuits, such as S38417 and S38584, the required resources have exceeded the available resources in XC2VP2 chip. Although the implementation of DTPG does not include the normal BIST parts, it is obvious that the remaining flip-flops in FPGA are sufficient to implement the LFSR and MISR as shown in Table 6.

Table 5. Feature parameters of platform family FPGA chip [8]

Platform Family Features	XC2VP2	XC2VP125
Logic Cells	3168	125,136
CLB: Slices	1408	55,616
CLB: Max Distr RAM (kb)	44	1,738
Block RAM: 18kb Blocks	12	556
Block RAM: Max Block RAM (kb)	216	10,008
Maximum User I/O Pads	204	1200

To validate the effectiveness of the trade-off between the required resources and test length, we increase the number of pseudorandom pattern to 1000, and then synthesize the corresponding DTPG again. The synthesis results are reported in Table 7. The results clearly express that with the proper increase in pseudorandom pattern, the corresponding flipping bits, i.e., the required FPGA

resources will be decreased significantly. Take s13207 for example, the required LUT will be decreased from 55% to 18%, the required slice will be decreased from 62% to 21%. At the same time the total test length, i.e., the test cost, still maintains a small value.

Table 6. Synthesis results for DTPG with the minimal hybrid test set

Circuit	# Det. Pat.	# Flip Bit	Virtex II Pro Family Platform FPGA: XC2VP2					
			Used Slice	Per. (%)	Used Flip	Per. (%)	Used LUT	Per. (%)
S5378	48	658	247	17	14	0.5	436	15
S9234	88	1299	506	35	17	0.6	900	31
S13207	174	1980	884	62	20	0.7	1567	55
S15850	106	2470	951	67	22	0.7	1692	60
S38417	191	6413	2731	193	24	0.8	4786	169
S38584	155	6728	3182	225	30	1	5573	197

Table 7. Synthesis results for DTPG after applying 1000 PRPs

Circuit	# Det. Pat.	# Care Bit	# Flip Bit	Virtex II Pro Family Platform FPGA: XC2VP2					
				Used Slice	per %	Used Flip	per %	Used LUT	per %
S5378	12	400	91	69	4	14	0.5	119	4
S9234	66	2994	644	310	22	17	0.6	546	19
S13207	110	3602	639	301	21	19	0.7	533	18
S15850	89	9566	1158	567	40	19	0.7	1004	35
S38417	171	24302	4744	2067	146	21	0.7	3616	128
S38584	96	7831	1621	791	56	20	0.7	1410	50

7. Conclusions

This paper has proposed an SOC test approach using an embedded FPGA core. It employs hybrid BIST and achieves the minimal test cost by taking advantage of the reconfigurable ability of FPGA and selecting the appropriate number of pseudorandom patterns. More importantly, the approach achieves minimal hardware overhead and high design flexibility. Experimental results for ISCAS 89 benchmark and a platform FPGA chip indicate the proposed approach is efficient. Future work includes improving the DTPG to reduce more care bits and designing a reasonable TAM between the FPGA core and other core.

References

[1] Y. Zorian, E. J. Marinissen and S. Dey, "Testing Embedded Core Based System Chips", *Proc. of IEEE Int. Test Conf.,* pp.130-143, 1998.

[2] M. Abramovici, et al. "Using Embedded FPGAs for SOC Yield Improvement", *Proc of ACM/IEEE Design Automation Conference (DAC)*, June 2002.

[3] P. Zuchowski et al. "A Hybrid ASIC and FPGA Architecture",*Proc. of IEEE/ACM International Conference on Computer Aided Design (ICCAD*), pp. 187-194, Nov. 2002.

[4] S. Wilton and R. Saleh, "Programmable Logic IP Cores in SoC Design: Opportunities and Challenges", *IEEE Custom Integrated Circuit Conference,* May 2001.

[5] DATE 2001 Roundtable: "Adding Reconfigurable Logic to SOC Design", *IEEE Design&Test of Computers,* Vol.18 No.4, pp. 65-71. July-August 2001.

[6] R. Sankaralingam, R. R. Oruganti, and N. A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation", *Proc. of IEEE VLSI Test Symposium,* pp. 35-40, 2000.

[7] S. Greg and V. Frank, "Energy Advantages of Microprocessor Platforms with On-Chip Configurable Logic" *IEEE Design&Test,* Vol.19 No.6, pp.36-43, Nov./Dec. 2002.

[8] Xilinx, Inc, http:// www.xilinx.com

[9] Xilinx Inc White Paper, "IBM Licenses Embedded FPGA Cores from Xilinx for Use in SoC ASICs" June 24, 2002

[10] Triscend Inc, Advance Product Brief, "Triscend Embedded Programmable Logic Core for SoC Applications", March 2003.

[11] C. Stroud, A Designer's Guide to Built-In Self-Test, Kluwer Academic Publishers, 2002.

[12] E. Kalligeros, X. Kavousianos and D. Nikolos, "A ROMless LFSR Reseeding Scheme for Scan-based BIST", *Proc. of Asia Test Symposium,* pp.206-211, 2002.

[13] S. Hellebrand, H. J. Wunderlich and A. Hertwig, "Mixed-Mode BIST Using Embedded Processors", *Proc. of IEEE Int. Test Conf. (ITC*), pp.195-204, 1996.

[14] S. Hwang and J. A. Abraham, "Optimal BIST Using an Embedded Microprocessor", *Proc. of IEEE Int. Test Conf. (ITC*), pp.736-745, 2002.

[15] T. Hiraide, K.O. Boateng, et al, "BIST-Aided Scan Test A New Method for Test Cost Reduction", *Proc. of IEEE VLSI Test Symposium,* pp. 359-364, 2003.

[16] E. H. Volkerink, A. Khoche and S. Mitra, "Packet-based Input Test Data Compression Techniques", *Proc. of IEEE Int. Test Conf. (ITC*), pp.154-163, 2002.

[17] N. A. Touba and E. J. Mccluskey, "Bit-Fixing in Pseudorandom Sequences for Scan BIST", *IEEE Trans. Computer-Aided Design,* Vol.20, No. 4, pp. 545-555, April 2001.

[18] H. J. Wunderlich and G. Kiefer, "Bit-Flipping BIST", *Proc. of Inc. Conf. On Computer-Aided Design (ICCAD),* pp.337-343, 1996.

[19] G. Kiefer, H. J. Wunderlich, "Deterministic BIST with Multiple Scan Chains", *Proc. of IEEE Int. Test Conf. (ITC*), pp. 1057–1064, 1998.

[20] G. Kiefer, H. Vranken, E. J. Marinissen, H. J. Wunderlich, "Application of Deterministic Logic BIST on Industrial Circuits", *Proc. of IEEE Int. Test Conf. (ITC*), pp. 105–114, 2000.

[21] G. Hetherington et al., "Logic BIST for Large Industrial Designs: Real Issues and Case Studies", *Proc. of IEEE Int. Test Conf.,* pp.358-367, 1999.

[22] G. Zeng and H. Ito "Efficient Test Data Decompression for System-on-a-chip Using an Embedded FPGA Core" *Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 503-510, 2003.