

Built-In Self-Test of Configurable Logic Blocks in Virtex-5 FPGAs



Bradley F. Dutton and Charles E. Stroud
Dept. of Electrical and Computer Engineering
Auburn University
Auburn, Alabama 36849
duttobf@auburn.edu strouce@auburn.edu

Abstract— A Built-In Self-Test (BIST) approach is presented for the configurable logic blocks (CLBs) in Xilinx Virtex-5 Field Programmable Gate Arrays (FPGAs). A total of 17 configurations were developed to completely test the full functionality of the CLBs, including distributed RAM modes of operation. These configurations cumulatively detect 100% of stuck-at faults in every CLB. There is no area overhead or performance penalty and the approach is applicable to all levels of FPGA testing (wafer, package, and in-system). A novel output response analyzer (ORA) design, which is efficiently implemented in FPGAs, provides both an overall single-bit pass/fail result and optimal diagnostic resolution when faults are detected. The implementation of the BIST approach in all Virtex-5 FPGAs and experimental results are discussed.

Keywords - Built-In Self-Test; Field Programmable Gate Array; configurable logic block; Virtex-5

I. INTRODUCTION AND BACKGROUND

Built-In Self-Test (BIST) for Field Programmable Gate Arrays (FPGAs) is typically targeted at manufacturing defects and operational faults that can appear at any point in the product life-cycle. As a result, BIST for FPGAs employs a defect-oriented test strategy [1]. Ideally, a BIST approach would be applicable to all levels of testing, from manufacturing test to in-system test, and would be entirely independent of the end user function. Additionally, the BIST would achieve maximal stuck-at fault coverage and would be executed at-speed to provide high fault coverage for a variety of fault models. When possible, high diagnostic resolution of detected faults is desired for fault-tolerant applications. This paper presents a BIST approach for the configurable logic blocks (CLBs) in Virtex-5 FPGAs that represents the culmination of over 15 years of work in FPGA BIST to address these concerns.

The first BIST for the configurable logic in FPGAs was proposed in [2]. The approach exploits the re-programmability of FPGAs to create BIST circuitry in the FPGA fabric during off-line testing. The only overhead is the external memory required to store the BIST and system function configurations along with the time required to download and execute the BIST. No area overhead or performance penalties are incurred

since the BIST logic “disappears” after the test session. Furthermore, the tests are applicable at all levels of testing since they are independent of the system function and require no external test fixture or equipment. The basic idea for the BIST is to configure some of the CLBs as Test Pattern Generators (TPGs) and Output Response Analyzers (ORAs) while configuring other CLBs as blocks under test (BUTs). The BUTs are repeatedly configured until they have been tested in every mode of operation [1]. These tests achieve maximal fault coverage by applying pseudo-exhaustive test patterns such that each sub-circuit of the BUT is exhaustively tested [2].

Several examples of BIST for the CLBs in FPGAs have been published, with each offering some improvement over the previous approach. Reference [3] introduced Boundary Scan as a means of controlling the BIST sequence. Xilinx engineers, in [4], introduced a set of iterative array logic tests with similarities to the approach presented in [2] and [3]. The general BIST approach, which is independent of the CLB array size, can also be adapted for on-line BIST techniques, as discussed in [5]. Previous examples of the implementation of this BIST approach on Xilinx 4000, Spartan, Virtex-I, Spartan-II and Atmel FPGAs are contained in [6], [7], and [8]. Partial reconfiguration was used in [9] to reduce the overall download and test times as well as system down time.

The BIST approach for Virtex-5 FPGAs builds primary on the previous work in [2], [3], [8], and [10]. However, our approach offers an improved ORA architecture and fewer total test configurations. We also improve the accuracy of the fault simulation models and add verification of the configurations on the target device via configuration memory bit fault injection. The remainder of this paper is organized as follows. Section II gives an overview of the CLB architecture in Virtex-5 FPGAs. Section III describes the BIST approach and implementation specific to Virtex-5 FPGAs. Section IV describes the experimental result and verification of the BIST. Section V summarizes and concludes the paper.

TABLE I. LIST OF ACRONYMS

Acronym	Definition	Acronym	Definition
CLB	Configurable Logic Block	BUT	Block Under Test
BIST	Built-in Self-test	LUT	Look-Up Table
ORA	Output Response Analyzer	SliceL	Logic Slice
TPG	Test Pattern Generator	SliceM	Memory Slice

This work was sponsored by the National Security Agency under contract H98230-04-C-1177 and supported in part by the National Science Foundation Grant CNS-0708962.

II. OVERVIEW OF VIRTEX-5 CLBS

The basic Virtex-5 logic element, illustrated in Fig. 1, is composed of a 6-input look-up table (LUT), a configurable flip-flop/latch, and multiplexers to control the combinational logic output and the registered output (flip-flop/latch input). Additional dedicated fast carry logic is included to perform special logic and arithmetic functions. In some slices, the LUT can be configured as a small RAM, called a distributed RAM or LUT RAM, or as a shift register [11]. Four such basic logic elements are grouped to form a slice, and two slices are grouped to form a complete CLB, as shown in Fig. 2 [11]. Each CLB is connected by a switch matrix to local and global programmable routing resources. Identical CLBs are tiled in columns and rows with larger devices including more columns and/or rows of CLBs. Additionally, the structure of the CLB is identical across all devices in the Virtex-5 family. The 6-input LUTs are designed with two outputs each. The primary output, O6, can utilize the full 64-bit LUT to implement any 6-variable Boolean function. The secondary output, O5, can be used to initialize the carry chain, or both the O5 and O6 output can implement an independent 5-variable Boolean function for five shared inputs. Either LUT output can be selected by the configuration multiplexers for the registered or combinatorial CLB output paths [11].

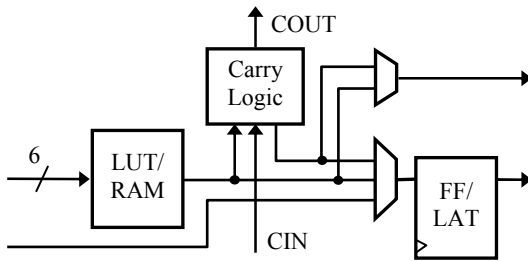


Figure 1. Simplified basic logic element

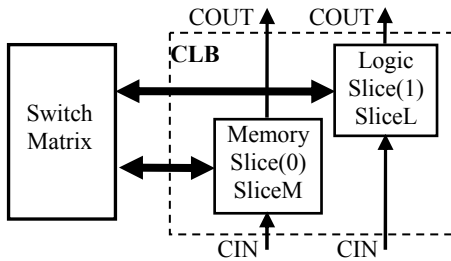


Figure 2. Virtex-5 configurable logic block [11]

Some slices (specifically the lower slice in every other column of CLBs and both columns to the left of a digital signal processor column) also support RAM and shift register modes of operation. The LUT RAMs in each slice have independent read address inputs and share a set of write address inputs. The independent read inputs facilitate the construction of dual-port RAMs within a slice. Each LUT can be configured as a simple 64×1 -bit or 32×2 -bit RAM. Dynamically controlled multiplexers in each slice allow the four LUTs to form a 256×1 -bit RAM. Additionally, the four LUTs can share five read address inputs and utilize eight independent data inputs to form a 32×8 -bit RAM. Each LUT can also form a single 32-bit

or two 16-bit shift registers. The four LUTs can be cascaded to form a 128-bit shift register or can operate in parallel form a 16×8 -bit shift register bank [11].

III. BIST APPROACH AND ARCHITECTURE

The BIST approach takes advantage of the regular structure of FPGAs by using comparison-based ORAs to compare the outputs of multiple identical BUTs. This detects all faults affecting any combination of BUTs (since all fault-free BUTs must produce the same pattern) so long as all of the BUTs compared by a set of ORAs do not fail identically and at the same time [3]. Since a faulty TPG could cause a faulty BUT to escape detection, multiple identical TPGs are used to drive alternating BUTs. This eliminates the assumption that the TPGs are fault-free because, with multiple identical TPGs, a faulty TPG will cause the outputs of some of the BUTs to disagree, resulting in ORAs reporting failures.

The CLB BIST architectures can be divided into two categories based on the slice mode being tested. The first set of configurations tests every CLB in the FPGA in SliceL (logic) mode of operation. The second set of configurations tests every SliceM. Only those slices which support SliceM (memory) mode are tested during the second set of configurations.

In SliceL BIST architecture, alternating columns of CLBs are configured as ORAs and BUTs, as illustrated in Fig. 3. The set of BIST configurations is repeated twice with the roles of the CLBs reversed such that every CLB serves both as ORA and as BUT. Two outputs of each BUT are compared by an ORA with the outputs of two adjacent identically configured BUTs in the same row, as shown in Fig. 4. A mismatch of two identically configured BUT outputs latches a logic 0 in the ORA flip-flop. Otherwise, a logic 1 is retained in the ORA and is interpreted as a passing result at the end of the test sequence. Traditionally, the results of the BIST are recovered via partial configuration memory readback where the contents of every ORA are retrieved from the configuration memory. However, we use a new ORA design that utilizes the dedicated carry logic in the CLB to form an iterative-OR of the ORA outputs. In each ORA, a passing result of logic 1 selects the Carry-in input, which is the Pass/Fail result of the previous ORA.

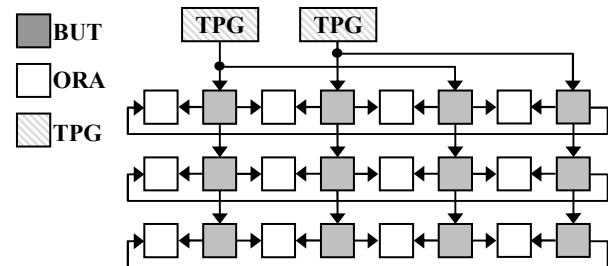


Figure 3. Circular comparison architecture

The Carry-in input of the first ORA in the iterative-OR chain is connected to Boundary Scan Test Data In (TDI), with the output of the last ORA connected to Test Data Out (TDO). If any ORA in the chain registers a failure, a logic 0 on the output of that ORA will select the logic 1 input of the carry

chain multiplexer which translates to a logic 1 on TDO. Otherwise, TDO passes the state of TDI such that by toggling TDI and observing TDO, the integrity of the iterative-OR chain can be verified at the end of the BIST sequence. If the output of the OR chain indicates a failure (TDO is a logic 1 regardless of the state of TDI), the contents of the ORAs can be retrieved via partial configuration memory readback to determine the location(s) of the failing BUT(s). This facilitates the single-bit pass/fail indication for faster test time without sacrificing diagnostic resolution for fault-tolerant applications.

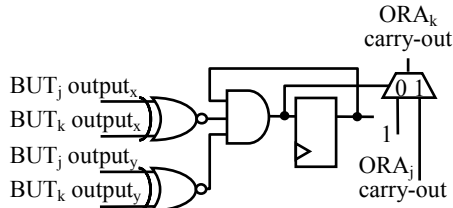


Figure 4. Equivalent ORA architecture

In Virtex-5 FPGAs, the carry-in of the bottom CLB and the carry-out of the top CLB in each column are not connected. To continue the carry chain, the carry-out of the top ORA in one column is connected to the D output and is routed to the AX input of the bottom ORA in an adjacent column. The AX input is selected as the carry-chain input in the bottom ORA in each column. In the ORA, each LUT is programmed with the hexadecimal value 0x90090000FFFFFFFF. By tying the A6 LUT input to logic 1, the O6 LUT output reads only the upper 32-bits of the LUT which implements the comparison ORA equation shown in (1), while the O5 output reads only the lower 32-bits of the LUT (which controls the carry chain multiplexer for the iterative-OR chain).

$$O6 = (\overline{A1 \oplus A2}) \cdot (\overline{A3 \oplus A4}) \cdot A5 \quad (1)$$

The architecture of the Virtex-5 CLBs requires a minimum of six configurations to test each of the 6 inputs to the flip-flop input multiplexers, (A-C)FFMUX. The first five of these configurations can also test the 5 inputs to the combinational logic output multiplexers (A-D)OUTMUX. Alternating XOR and XNOR functions in the LUTs detects every LUT stuck-at fault in two BIST configurations. Multiple identical TPGs are implemented in a column of embedded digital signal processors (DSPs) and drive alternating columns of BUTs. This reduces loading on the TPGs in large devices and eliminates the assumption that the TPG is fault-free. The DSPs are configured to accumulate a large prime number placed on the DSP inputs. This number, 0xCA6691, was shown in [12] to produce an exhaustive sequence of 12-bit test patterns in 2^{12} clock cycles with a relatively high number of transitions in the most significant bits of the accumulator output. Virtex-5 CLBs require at least 12 TPG lines for pseudo-exhaustive testing, and, therefore, 4,096 clock cycles for the exhaustive set of test patterns to be produced by the accumulator. Six of the TPG outputs fan out to the inputs of each of the four LUTs. Adjacent LUTs are alternately programmed with XOR and XNOR functions such that adjacent LUTs will produce opposite logic values. Another six TPG lines exercise the AX,

BX, CX, DX, CE, and SR slice inputs with pseudo-exhaustive test patterns. A total of 12 SliceL BIST configurations are generated, such that every CLB is a BUT for six configurations and an ORA for another six configurations. A summary of the SliceL BIST configurations is given in Table II.

TABLE II. SLICEL LOGIC BIST CONFIGURATIONS

Config.#	A-D LUTs	FF/Latch	CYINIT	CLKINV
#1	XOR/XNOR	FF INIT1	#OFF	CLK
#2	XNOR/XOR	FF INIT0	AX	CLK
#3	XOR/XNOR	FF INIT0	0	CLK
#4	XNOR/XOR	LAT INIT1	1	CLK
#5	XOR/XNOR	FF INIT0	0	CLK
#6	XNOR/XOR	FF INIT1	AX	CLK B
Config.#	A-D FFMUX	A-D MUX		
#1	O6, O6, O6, O6	CY, CY, CY, CY		
#2	O5, O5, O5, O5	XOR, XOR, XOR, XOR		
#3	AX, BX, CX, DX	O5, O5, O5, O5		
#4	XOR, XOR, XOR, XOR	O6, O6, O6, O6		
#5	CY, CY, CY, CY	F7, F8, F7, CY		
#6	F7, F8, F7, DX	F7, F8, F7, CY		

Every other CLB column contains a SliceM. In addition, the CLB column to the left of a DSP column contains a SliceM and, in SX devices, the second CLB column to the right of a DSP column contains a SliceM. In columns containing SliceMs, only the bottom slice in each CLB is a SliceM. Therefore, every SliceM can be tested simultaneously since there is at least one SliceL for every SliceM (located in the same CLB) that can serve as an ORA. The ORAs for the SliceM BIST architecture are the same as those used in the SliceL BIST architecture, including the iterative-OR chain. However, the circular comparison chain is formed along each column containing SliceMs by comparing the outputs of each BUT with the identically configured BUT in an adjacent row. A 2048x18-bit block RAM, effectively configured as a ROM, is used to store deterministic test patterns and, in conjunction with a DSP configured as an address counter, forms a TPG. Multiple identical TPGs are configured to drive alternating rows of BUTs. The SliceM BIST configurations are summarized in Table III. To test the LUT RAMs in single-port modes (configurations #1 and #2), the block RAMs are initialized with the test patterns for a March Y test algorithm. A March Y RAM test requires 8N test patterns, where N is the number of address locations [10][13]. For the remaining configurations, the block RAMs are initialized with test patterns for a dual-port RAM test algorithm [1][6].

TABLE III. SLICEM BIST CONFIGURATIONS

Config.#	RAM mode	DI1MUX	WEMUX	FFMUX
#1	SPRAM64	DX	CE	O6
#2	SPRAM32	A-DX	CE	O6
#3	DPRAM32	DX	WE	O5
#4	SRL32	MC31	WE	MC31
#5	SRL16	A-DX	WE	O6
Config.#	OUTMUX	WA8used	WA7used	BIST CCs
#1	O6	0	0	2,048
#2	O6	#OFF	#OFF	2,048
#3	O6	#OFF	#OFF	2,048
#4	O6	#OFF	#OFF	2,048
#5	MC31	#OFF	#OFF	2,048

IV. EXPERIMENTAL RESULTS

The BIST configurations were developed using accurate gate-level models of the Virtex-5 CLB. The SliceL and SliceM were modeled separately for fault simulation. For both SliceL and SliceM, the BIST configurations and their associated fault coverage were first optimized using these gate-level models. The single stuck-at gate-level fault coverage for SliceL and SliceM BIST configurations obtained from fault simulations of these models are summarized in Fig. 5 and Fig. 7, respectively.

The BIST configurations were then verified on Virtex-5 LX30T and SX35T devices via configuration memory bit fault injection. Using the fault injection approach, configuration memory bits can be manipulated to emulate physical faults in the FPGA core including shorts and opens in programmable interconnect as well as almost any fault in logic resources controlled by a configuration memory bit. Configuration bits controlling the SliceLs and SliceMs were injected with faults and the BIST configurations were executed with the faulty configuration on the device. The BIST results of the faulty configuration are retrieved via partial configuration memory readback. The fault injection results show that the 17 BIST configurations cumulatively detect every configuration memory bit fault in every CLB. The results of the fault injection for SliceL BIST are shown in Fig. 6. The similarity of the fault injection results and fault simulation results serve as a good indicator of the accuracy of the gate-level fault models, which include every stuck-at fault in the CLB (including configuration memory bits). Fig. 7 and Fig. 8 summarize the fault simulation results and the results of configuration memory bit fault injection, respectively, for the SliceM BIST configurations. It should be noted that three of the SliceM faults are detected by SliceL configurations.

There are two methods by which the results of the BIST sequence can be obtained. First, the single bit pass/fail result can be determined via the TDO output of the ORA iterative-OR chain. However, the location of failing BUTs can not be determined using this method. Another option is to perform a partial configuration memory readback to determine the contents of each ORA at the end of the BIST. By this method, the location of the failing BUT(s) can be easily determined with diagnostic resolution of LUT or flip-flop. To minimize test time and achieve maximum fault resolution, a combination of the two methods is used. First, the pass/fail status of the BIST is determined by observing TDO. If TDO presents a logic 1 regardless of the state of TDI, at least one ORA has observed a failure. Partial configuration memory readback can then be used to obtain the locations of the failing ORA(s) and, thereby, determine the location(s) of the faulty BUT(s).

We have developed two C programs that automatically generate the 17 BIST configurations for all Virtex-5 LX, LXT, SXT, and FXT devices. Table IV summarizes the total download file size for the 17 BIST configurations, the maximum BIST clock frequency, and the total number of BIST clock cycles for full chip tests on several Virtex-5 devices. The total full chip test time for serial and parallel configuration interfaces is summarized in Fig. 9 and Fig. 10. The calculated test time assumes a 40 MHz BIST clock for all configurations and devices. However, on most devices, the BIST configurations can operate at higher clock frequencies.

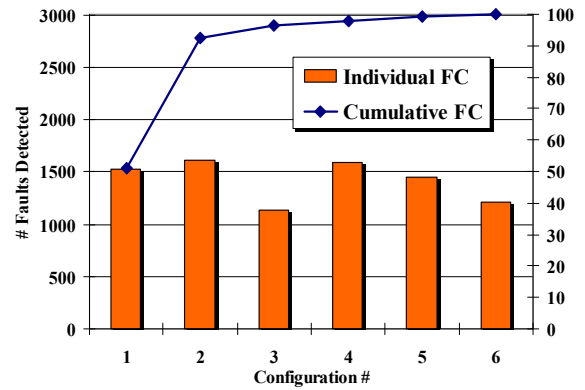


Figure 5. SliceL fault coverage (simulation)

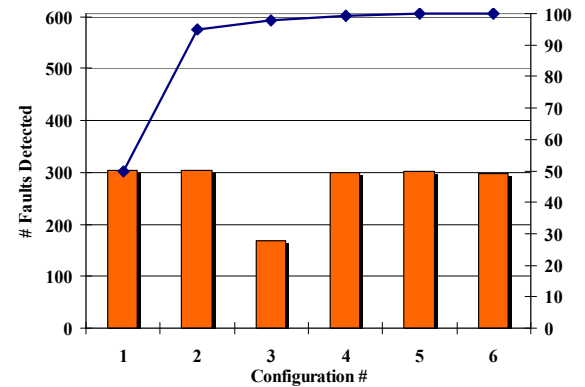


Figure 6. SliceL fault coverage (fault injection)

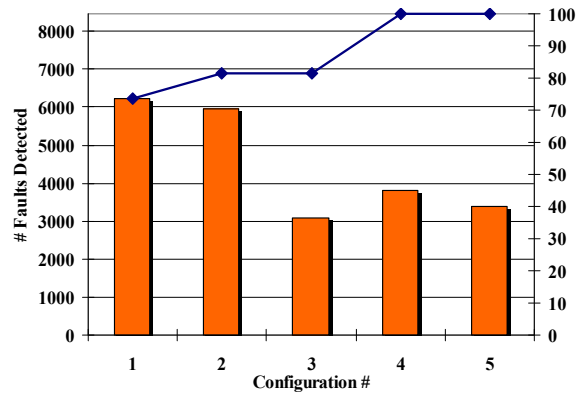


Figure 7. SliceM fault coverage (simulation)

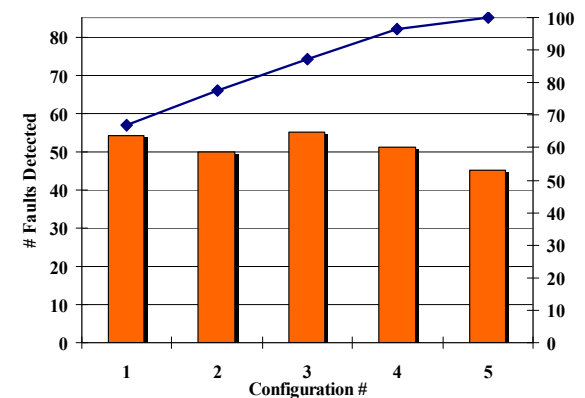


Figure 8. SliceM fault coverage (fault injection)

TABLE IV. CLB BIST TOTALS (17 CONFIGURATIONS)

Device	Total Config. Size (kB)	Max. BIST Clock Freq.	BIST CCs
LX20T	1,762	90.7 MHz	59,392
LX30T	2,630	74.0 MHz	59,392
LX50T	3,930	74.4 MHz	59,392
LX85T	6,265	58.2 MHz	59,392
LX110T	8,837	58.0 MHz	59,392
SX35T	3,378	59.2 MHz	59,392
SX50T	5,041	61.1 MHz	59,392
SX95T	8,818	44.7 MHz	59,392

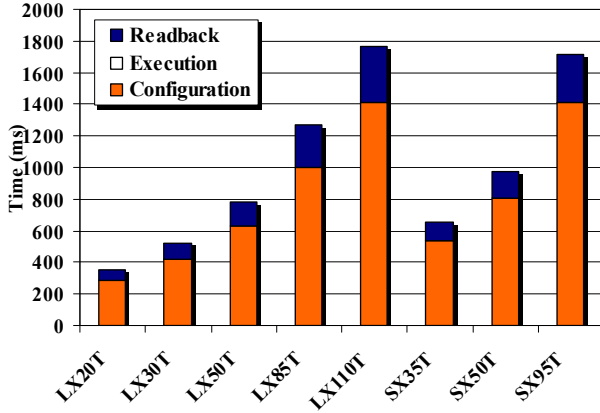


Figure 9. Boundary scan interface test time

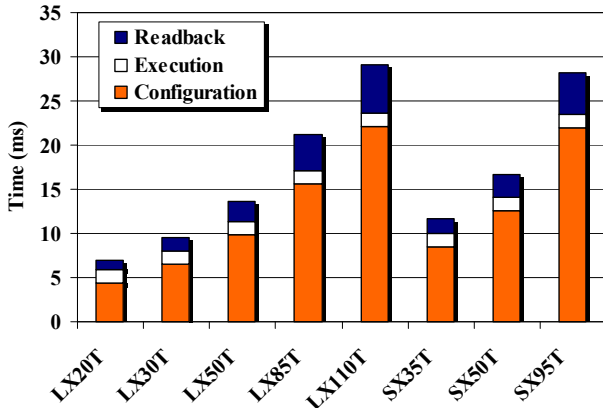


Figure 10. 32-bit parallel interface test time

In early FPGAs, all LUTs were able to function as small RAMs such that the first BIST configuration applied typically tested the LUTs in the RAM mode of operation. Using this approach, the first BIST configuration was able to detect most faults that could affect the LUT [2]. When combined with a simultaneous test of the flip-flop, the first BIST configuration was able to achieve around 80% fault coverage. A similar characteristic can be observed in the first SliceM BIST configuration in Fig. 7, which achieves greater than 70% fault coverage. However, current FPGAs, such as Virtex-4 and Virtex-5, limit the number of LUTs that can function as small RAMs. Therefore, two BIST configurations are required (with alternate XOR and XNOR programming) to detect most of the faults in all LUTs. This can be observed in Fig. 5, where the cumulative fault coverage after the first configuration reaches 51% and after two configurations exceeds 92%.

V. SUMMARY AND CONCLUSIONS

A BIST approach for testing the CLBs in Virtex-5 FPGAs was presented. A total of 17 test configurations were developed to achieve 100% stuck-at fault coverage in every CLB. Twelve of these configurations pseudo-exhaustively test every SliceL and every SliceM in the SliceL mode. Another five configurations test every SliceM in their RAM and shift register modes of operation. The BIST configurations were developed using accurate gate-level fault models of the CLB and verified using configuration memory bit fault injection. A novel ORA design provides a single bit pass/fail result for each BIST sequence and is independent of the configuration interface. Optional partial configuration memory readback provides optimal diagnostic resolution for fault-tolerant applications when the pass/fail output indicates failures. As a result, the BIST approach is applicable to all levels of FPGA testing including manufacturing testing and in-system testing for fault-tolerant applications. We modified SliceL BIST to support FXT devices by creating two circular comparison chains across rows directly above the PowerPC core because CLBs above the PowerPC have no carry-in routing. We have also applied this approach to Virtex-4 devices resulting in 20 and 5 BIST configurations for SliceL and SliceM tests, respectively, compared to 31 total configurations for Virtex-4 CLBs reported in [8]. Our Virtex-4 CLB BIST also includes the new ORA design for single bit pass/fail indication.

REFERENCES

- [1] L-T Wang, C. Stroud, and N. Touba, *System-on-Chip Test Architectures*, Morgan Kaufmann, 2007.
- [2] C. Stroud, S. Konala, P. Chen, and M. Abramovici, "Built-in self-test of logic blocks in FPGAs," *Proc. IEEE VLSI Test Symp.*, pp.387-392, 1996.
- [3] M. Abramovici and C. Stroud, "BIST-based test and diagnosis of FPGA logic blocks," *IEEE Trans. on VLSI Syst.*, vol. 9, no. 1, pp. 159-172, 2001.
- [4] S. Toutouchi and A. Lai, "FPGA test and coverage," *Proc. IEEE Int. Test Conf.*, pp. 599-607, 2002.
- [5] M. Abramovici, C. Stroud, and J. Emmert, "Online BIST and BIST-based diagnosis of FPGA logic blocks," *IEEE Trans. on Very Large Scale Integr. (VLSI) Syst.*, vol.12, no.12, pp. 1284-1294, 2004.
- [6] C. Stroud, K. Leach, and T. Slaughter, "BIST for Xilinx 4000 and Spartan series FPGAs: a case study," *Proc. IEEE Int. Test Conf.*, pp. 1258-1267, 2003.
- [7] C. Stroud, J. Harris, S. Garimella, and J. Sunwoo, "Built-in self-test for system-on-chip: a case study," *Proc. IEEE Int. Test Conf.*, pp. 837-846, 2004.
- [8] S. Dhingra, D. Milton, and C. Stroud, "BIST for logic and memory resources in Virtex-4 FPGAs," *Proc. IEEE North Atlantic Test Workshop*, pp. 19-27, 2006.
- [9] S. Dhingra, S. Garimella, A. Newalker, and C. Stroud, "Built-in self-test of Virtex and Spartan II FPGAs using partial reconfiguration," *Proc. IEEE North Atlantic Test Workshop*, pp. 7-14, 2005.
- [10] C. Stroud and S. Garimella, "BIST and diagnosis of multiple embedded cores in SoCs," *Proc. Int. Conf. on Embedded Systems and Applications*, pp. 130-136, 2005.
- [11] *Virtex-5 FPGA User Guide*, UG190 (v 4.2), Xilinx Inc., San Jose, CA, May 2008. Available: www.xilinx.com
- [12] S. Gupta, J. Rajski, and J. Tyszer, "Test pattern generation based on arithmetic operations," *Proc. IEEE Int. Conf. on Computer-Aided Design*, pp. 117-124, 1994.
- [13] A. van de Goor, *Testing Semiconductor Memories Theory and Practice*, John Wiley and Sons, 1991.