from Proc. IEEE Southeast Regional Conf., pp. 506-509, 2010

On System-Level Use of BIST for Programmable Input/Output Buffers in FPGAs

Bradley F. Dutton, Lee W. Lerner, Sudheer Vemula, and Charles E. Stroud
Dept. of Electrical and Computer Engineering
Auburn University
Auburn, AL
duttobf@auburn.edu

***Abstract*—We describe a Built-In Self-Test (BIST) approach that was developed for the programmable Input/Output (I/O) buffers in Field Programmable Gate Arrays (FPGAs). The approach is unique when compared with previous work because the I/O buffers are tested separately from the other programmable logic in the I/O cells. The capabilities and limitations of system-level use of this I/O buffer BIST are discussed in conjunction with experimental results from the implementation and actual use of the approach in systems.**

I. INTRODUCTION

Input/Output (I/O) buffers are typically tested at the system-level using the IEEE 1149.1 Boundary Scan EXTEST feature [1]. However, Field Programmable Gate Arrays (FPGAs) have a significant amount of programmable logic resources associated with the I/O buffers that cannot be tested in this manner [3][4][5]. The programmable logic resources include multiplexers and flip-flops/latches for improving system timing specifications such as set-up and hold times as well as clock-to-output delay, as illustrated in Fig. 1. Additional logic resources are included to support single data rate (SDR) and double data rate (DDR) transmission and reception as well as for serialization/de-serialization (SERDES) modes of operation [4] [5]. This additional logic is summarized in Table I for three different FPGA families which are representative of the range of resources typically associated with the programmable I/O cells in FPGAs (excluding SERDES operation in Virtex-4 and Virtex-5).

The Boundary Scan INTEST feature can be used to test the configurable logic resources in an I/O cell [1]. However, the INTEST feature is supported by few FPGA manufacturers. With any approach, the programmable I/O cells must be reconfigured a number of times in order to completely test all of the programmable functionality. As indicated in Table I, for example, we have developed 25, 76 and 77 BIST configurations to completely test the I/O cells (including the I/O buffers) in Atmel AT94K, Xilinx Virtex-4, and Xilinx Virtex-5 devices, respectively. In this paper, we describe the BIST architecture that was developed for the programmable I/O buffers in Virtex-4 and Virtex-5 FPGAs. Some of the unique capabilities and limitations observed during the implementation and actual use of the BIST for system-level testing of the I/O buffers in FPGAs are also discussed.

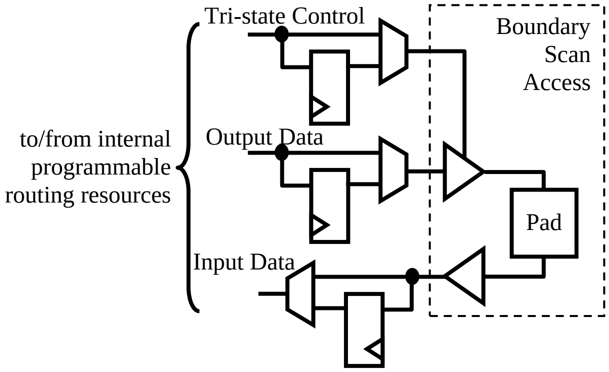

Figure 1. Simplified programmable I/O cell.

TABLE I. COMPARISON OF FPGA I/O CELL FEATURES

Feature	AT94K [3]	Virtex-4 [4]	Virtex-5 [5]
Multiplexers	4	32	32
Flip-flops/latches	2	10	10
Output drive levels	3	7	7
I/O standards	3	64	75
BIST Configurations	25	76	77

II. PRIOR WORK

There has been limited prior work in the area of testing I/O cells in, or applicable to, FPGAs [2][6]-[10]. In [2], a system-level BIST architecture is presented for the I/O cells of Atmel FPGAs. The overall BIST approach was similar to that used for configurable logic resources in the FPGA core [11]. The BIST architecture in [2] consists of a single TPG implemented in configurable logic blocks (CLBs) sourcing test vectors to the I/O cells under test. A single TPG was implemented under the assumption that internal FPGA resources had already been tested and found to be fault-free. The I/O cells under test are identically configured with bidirectional I/O buffers such that the output responses are sent back into the FPGA internal resources. However, for in-system testing, this requires that all external devices be tri-stated during testing. The output responses of the I/O cells are monitored by CLBs configured as comparison-based output response analyzers (ORAs). While presenting a general architecture applicable to any FPGA or configurable SoC with an FPGA core and bidirectional I/O buffers, [2] implemented 25 BIST configurations applicable to the Atmel AT94K SoC and AT40K FPGA families only.

In [6], we presented a BIST approach for the programmable logic in the I/O cells of Xilinx Virtex-4 and Virtex-5 FPGAs. An important difference between the I/O tile BIST architecture in [6] and other prior work is in the configuration of the I/O tiles under test. Prior approaches relied on bidirectional I/O buffers to provide the return path for test patterns exiting the output logic and returning to the ORAs via input logic [2][9][10]. But the reliance on bidirectionally configured I/O buffers severely limits the applicability of this type of BIST for in-system testing. With every I/O buffer configured in the path of the logic under test, the previous approaches required that all connecting devices be tri-stated during in-system testing. Connecting passive devices, such as termination resistors or light emitting diodes (LEDs), introduce another problem since these devices cannot be disconnected or tristated during in-system tests. For example, we have observed that, at certain BIST clock frequencies, LEDs connected to I/O buffers under test caused the comparison ORAs to erroneously report failures for otherwise fault-free I/O tiles. These failures were observed at frequencies as low as 325 kHz, which is unacceptable for an at-speed test of the logic resources. As a result, the generality of the BIST is compromised. Fortunately, the I/O tiles in Virtex-4 and Virtex-5 FPGAs include dedicated routing from the OLOGIC to the ILOGIC that bypasses the I/O buffer [4][5]. Using this feedback routing instead of the I/O buffer means that no signals from the FPGA under test can reach, and therefore be influenced by, external devices during testing of programmable logic in the I/O tile. Furthermore, bypassing the I/O buffer does not sacrifice fault coverage in the I/O tile logic resources. With the I/O buffers removed from all tests for logic resources, these tests may be applied without concern for the external test environment, thus making our approach applicable to all levels of FPGA testing. The remainder of this paper discusses the implementation and system application of stand-alone BIST configurations developed for the I/O buffers that are independent of the tests for the additional programmable logic resources in the I/O tiles.

III. I/O BUFFER BIST

The basic concept of the BIST architecture, illustrated in Fig. 2, is to configure I/O buffers as bidirectional buffers to allow test patterns produced by multiple identically configured test pattern generators (TPGs) to be applied to output buffers while providing a return path through input buffers leading to the core of the FPGA. The output response of each I/O buffer under test is then compared to the output responses of other identically configured I/O buffers under test by circular comparison-based ORAs, illustrated in Fig. 3, to detect mismatches in output responses of I/O buffers due to faults. The ORA contents can be retrieved via partial configuration memory readback at the end of the BIST sequence such that the faulty I/O buffers can be determined using a diagnostic algorithm developed for the circular comparison-based BIST approach [7]. When diagnosis of faults is not required, a single-bit pass result is available, as described in [6]. The TPGs and ORAs are constructed from the programmable logic resources in the core of the FPGA during off-line testing and are replaced by the intended system function during normal on-line operation. Hence, there is no area or performance penalty to the normal system function.

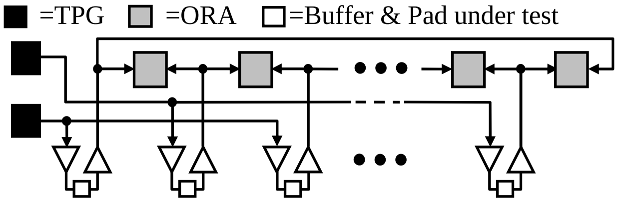

Figure 2. General I/O BIST architecture.

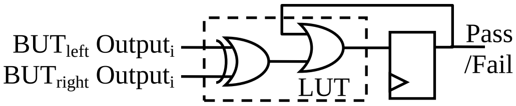

Figure 3. Comparison-based output response analyzer.

One TPG is configured for each of the I/O buffers under test. The TPGs, which are 2-bit counters, apply exhaustive test patterns to all of the I/O buffers under test, which each have two inputs (T and O), as illustrated in Fig. 4. By applying the 2-bit count pattern for an additional clock cycle, both inputs are tested for 0-to-1 and 1-to-0 transitions.

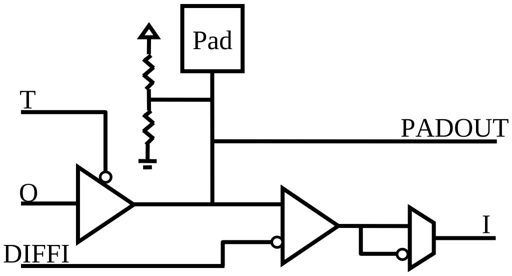

Figure 4. Virtex-5 programmable I/O buffer.

In Virtex-4 and Virtex-5 devices, each I/O tile includes two programmable bidirectional I/O buffers. Each buffer supports a wide variety of I/O standards and has programmable drive strength, slew rate, and on-chip termination with digitally controlled impedance (DCI) matching. A programmable pull-up, pull-down, keeper circuit is also included. The supported single-ended I/O standards include LVCMOS, LVTTL, HSTL, SSTL, GTL, and PCI [4][5]. Differential I/O standards are supported by grouping the two I/O buffers in an I/O tile and utilizing the DIFFI input (Fig 4.) of the input buffer and PADOUT connection to the DIFFI input of the adjacent input buffer. Supported differential I/O standards include LVDS, HT, LVPECL, BLVDS, differential HSTL, and SSTL [4][5]. A simplified model of the bidirectional buffer is shown in Fig. 4. In Virtex-5, 40 I/O buffers are grouped in an I/O bank (32 I/O buffers in Virtex-4). The output drive voltage, V_{CCO}, is supplied via a dedicated pin for each I/O bank. Therefore, all I/O buffers in an I/O bank must share the same output drive source voltage [4][5]. For single-ended I/O standards with a differential input buffer that requires a reference voltage, two pins in each I/O bank are configured as V_{ref} inputs.

Printed circuit board (PCB) traces are typically terminated with resistors to match the impedance of the receiver or driver to that of the trace. The I/O buffers in Xilinx Virtex-4 and Virtex-5 FPGAs allow for on chip termination of traces using a DCI circuit (included with each I/O bank). DCI adjusts the input termination of the receiver or output impedance of the driver to equal an external reference resistance [4][5]. Two reference resistors are required per I/O bank, and are connected to two reference pins in each I/O bank. Note that theses two pins may not operate as regular I/O when any I/O buffer in the bank is configured with DCI. The architectures and operation of the controlled impedance driver and controlled impedance input termination circuits are described in detail for Virtex-4 in [4] and Virtex-5 in [5].

Multiple configurations of the I/O buffers are required to test all of the modes of operation of the I/O buffers. Our BIST configurations are divided into four categories: single-ended I/O standards without a voltage reference, single-ended I/O standards with a voltage reference, DCI enabled I/O standards without a voltage reference, and DCI enabled I/O standards with a voltage reference. For testing purposes, all of the I/O buffers in an I/O bank are identically configured. When external references voltages or reference resistors are required, the reference pins in each I/O bank are not configured for test, and the appropriate voltage or resistance is supplied at the reference pin. Additionally, the appropriate output drive voltage, V_{CCO}, is supplied on a per-bank basis. For manufacture testing, all I/O buffers in the device are identically configured and the appropriate reference voltages and/or reference resistances for the configured I/O standard are supplied via the testing apparatus. For in-system testing of the I/O buffers, the reference voltages and/or reference termination resistors are supplied by the system, and I/O buffers are tested on a per-bank basis in the system mode of operation only. All configurations of the I/O buffers in Virtex-4 devices that are testable under the current scheme are shown in Table II at the end of the paper. Note that I/O standards not supporting a bidirectional mode of operation cannot be tested because, without a bidirectional buffer, there is no return path for signals driven by an output buffer to the ORAs in the FPGA fabric.

IV. SYSTEM-LEVEL USE

The bidirectional buffers configured during BIST can be expected to have different load characteristics in the system depending on the way they are terminated and whether they are normally an input, output, or bidirectional port during system operation. Therefore, we can expect the I/O buffers to produce different results at difference frequencies, depending on the loading present at the pin in the system. All of the I/O buffers can be tested at a single low frequency (determined for a particular system) that is guaranteed to be sufficiently slow to allow fault-free I/O buffers to generate passing results. However, this may result in faulty I/O buffers escaping detection in the event of delay faults and other parametric variations in the I/O buffers. Alternatively, the I/O buffers can be grouped together by loading characteristics to be tested at different frequencies. The following examples illustrate the sensitivity of the BIST approach to external loading factors.

We designed a PCB with an Atmel AT94K series FPGA with adjacent I/O cells connected to LEDs that are in turn connected to ground via adjacent resistors in a common resistor network, as illustrated in Fig. 5. The routing on the printed circuit board is identical for all of these nets in terms of length and width of the traces. As a result, the loading characteristics are similar for all output buffers. The only difference is the colors of the LEDs and their associated voltage and current characteristics, as noted in Fig. 5. The BIST was performed on the I/O cells with indicated pass and failures as noted in Fig. 5. When the BIST clock frequency is gradually increased, I/O buffers connected to the green LEDs fail the BIST while the I/O buffers connected to the red and yellow LEDs continue to pass. As the BIST clock frequency is further increased, all of these I/O buffers begin to fail the BIST. The BIST clock period and frequency is indicated in Fig. 5 at the point where the I/O buffers fail the BIST and at the point where the I/O buffers pass the BIST.

The failing test configuration in the above example configured pull-down resistors on the pads. We observed a similar situation with other configurable options, such as Schmitt trigger on input buffers. But in this latter case the I/O buffers connected to red LEDs failed first as we increased the BIST clock frequency, while the I/O buffers connected to the yellow and green LEDs continued to pass. By continuing to increase the BIST clock frequency, failing results could be obtained for all of the I/O buffers.

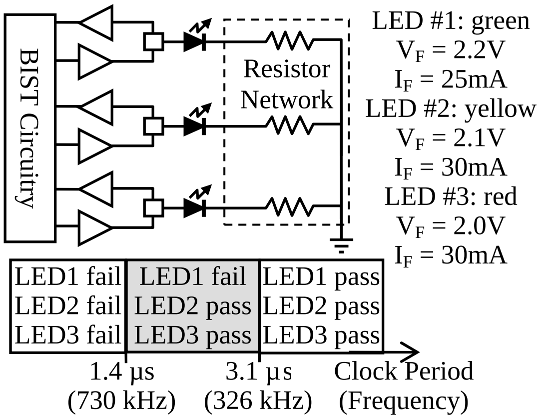

Figure 5. LED failures vs. BIST clock frequency.

We observed similar results on PCBs with Virtex-4 and Virtex-5 FPGAs. For example, I/O buffers connected to a bank of green LEDs were observed to fail at BIST clock frequencies above 1.9 MHz. Other terminated pins consistently failed at frequencies above 2.8 kHz. Differential pairs with termination resistors failed when testing programmable pull-up or pull-down resistors on bidirectional buffers regardless of operating frequency. Therefore, external connections must be taken into account when considering which I/O buffers to test during system-level use of BIST.

V. CONCLUSIONS

A BIST approach for the programmable I/O buffers in FPGAs was presented that is applicable at all levels of testing. Results of the actual implementation of the approach in systems with FPGAs were discussed, and the provided examples give an indication of the sensitivity of the I/O buffer BIST approach to external system loading effects. As a result of this sensitivity, we currently test I/O cell logic resources associated with SDR, DDR, and SERDES modes of operation using internal loopback capabilities in the Xilinx Virtex-4 and Virtex-5 I/O cells to avoid the affects of external system loading such that all I/O cell logic can be tested at a fixed clock frequency. Only the logic and configuration options (including I/O standards) associated with the buffers are tested using the bidirectional buffer modes of operation. It should be noted that Boundary Scan EXTEST cannot be used to test the LED connections illustrated in Fig. 5 without the assistance of an external mechanism to monitor the illumination state of the LEDs. Yet, this BIST approach could possibly be used to determine if the LEDs are illuminated based on the pass/fail characteristics of the test configuration without visual monitoring. In addition, other attributes of the system application could potentially be tested such as capacitive and resistive loading by grouping I/O buffers for testing at different BIST clock frequencies.

REFERENCES

[1] *IEEE Standard Test Access Port and Boundary-Scan Architecture*, IEEE Std 1149.1-2001, 2001.

[2] S. Vemula and C. Stroud, "Built-In Self-Test for Programmable I/O Buffers in FPGAs and SoCs", Proc. IEEE Southeastern Symp. on System Theory, pp. 534-538, 2006.

[3] Atmel, "AT94K Series Field Programmable System Level Integrated Circuit," Data Sheet, Atmel Corp., 2001 (available at www.atmel.com).

[4] Xilinx, "Virtex-4 User Guide," UG070 (v 2.0), Xilinx, Inc., 2007 (available at www.xilinx.com).

[5] Xilinx, "Virtex-5 User Guide," UG190 (v 4.2), Xilinx, Inc., 2008 (available at www.xilinx.com).

[6] B. Dutton and C. Stroud, "Built-In Self-Test of Programmable Input/Output Tiles in Virtex-5 FPGAs," Proc. Southeastern Symp. On System Theory, 2009.

[7] L-T Wang, C. Stroud, and N. Touba, System-on-Chip Test Architectures, Morgan Kaufmann, 2007.

[8] C. Jia and L. Milor, "A BIST Solution for the Test of I/O Speed," *Proc. IEEE Int. Test Conf.*, pp. 1023-1030, 2003.

[9] L. Zhao, D. Walker and F. Lombardi, "IDDQ Testing of Input/Output Resources of SRAM-Based FPGAs," *Proc. Asian Test Symp.*, pp. 375-380, 1999.

[10] L. Lerner, "Built-In Self-Test for Input/Output Tiles in Field Programmable Gate Arrays," M.S. thesis, Dept. of Elect. and Comput. Eng., Auburn Univ., Auburn, AL, Dec. 2007.

[11] D. Milton, S. Dhingra, and C. Stroud, "Embedded Processor Based Built-In Self-Test and Diagnosis of Logic and Memory Resources in FPGAs," *Proc. Int. Conf. on Embedded Systems and Applications*, pp. 87-93, 2006.

TABLE II. VIRTEX-4 I/O STANDARDS AND BIST CONFIGURATIONS

I/O Standard	**Type**	**V_{REF}**	**Cfg**	**DCI I/O Standards**	**Type**	**V_{REF}**	**Cfg**
LVTTL	SE	N/R	1	LVDCI_33	SE	N/R	34
LVCMOS33	SE	N/R	2	LVDCI_25	SE	N/R	35
LVCMOS25	SE	N/R	3	LVDCI_18	SE	N/R	36
LVCMOS18	SE	N/R	4	LVDCI_15	SE	N/R	37
LVCMOS15	SE	N/R	5	LVDCI_DV2_25	SE	N/R	38
PCI_33_3	SE	N/R	6	LVDCI_DV2_18	SE	N/R	39
PCI_66_3	SE	N/R	7	LVDCI_DV2_15	SE	N/R	40
PCIX	SE	N/R	8	HSTL_II_T_DCI	SE	0.75	41
HSTL_I	SE	0.75	9	HSTL_II_T_DCI_18	SE	0.9	42
HSTL_II	SE	0.75	10	SSTL2_II_T_DCI	SE	1.25	43
HSTL_III	SE	0.9	11	SSTL18_II_T_DCI	SE	0.9	44
HSTL_IV	SE	0.9	12	DIFF_SSTL2_II_DCI (M)	D	N/R	45
HSTL_I_18	SE	0.9	13	DIFF_SSTL2_II_DCI (S)	D	N/R	46
HSTL_II_18	SE	0.9	14	DIFF_SSTL18_II_DCI (M)	D	N/R	47
HSTL_III_18	SE	1.1	15	DIFF_SSTL18_II_DCI (S)	D	N/R	48
HSTL_IV_18	SE	1.1	16	DIFF_HSTL_II_DCI (M)	D	N/R	49
HSTL_I_12	SE	0.6	17	DIFF_HSTL_II_DCI (S)	D	N/R	50
GTL	SE	0.8	18	DIFF_HSTL_II_DCI_18 (M)	D	N/R	51
GTLP	SE	1	19	DIFF_HSTL_II_DCI_18 (S)	D	N/R	52
SSTL2_I	SE	1.25	20	GTL_DCI	SE	0.8	53
SSTL2_II	SE	1.25	21	GTLP_DCI	SE	1	54
SSTL18_I	SE	0.9	22	HSTL_I_DCI**	SE	0.75	NA
SSTL18_II	SE	0.9	23	HSTL_II_DCI	SE	0.75	55
LVPECL_25 (M)**	D	N/R	NA	HSTL_III_DCI**	SE	0.9	NA
LVPECL_25 (S)**	D	N/R	NA	HSTL_IV_DCI	SE	0.9	56
LVDS_25**	D	N/R	NA	HSTL_I_DCI_18**	SE	0.9	NA
LVDSEXT_25**	D	N/R	NA	HSTL_II_DCI_18	SE	0.9	57
BLVDS_25 (M)	D	N/R	24	HSTL_III_DCI_18**	SE	1.1	NA
BLVDS_25 (S)	D	N/R	25	HSTL_IV_DCI_18	SE	1.1	58
ULVDS_25**	D	N/R	NA	SSTL2_I_DCI**	SE	1.25	NA
LDT_25**	D	N/R	NA	SSTL2_II_DCI	SE	1.25	59
RSDS_25**	D	N/R	NA	SSTL18_I_DCI**	SE	0.9	NA
DIFF_SSTL2_II (M)	D	N/R	26	SSTL18_II_DCI	SE	0.9	60
DIFF_SSTL2_II (S)	D	N/R	27	LVDS_25_DCI**	D	N/R	NA
DIFF_SSTL18_II (M)	D	N/R	28	LVDSEXT_25_DCI**	D	N/R	NA
DIFF_SSTL18_II (S)	D	N/R	29	HSLVDCI_33	SE	Vcco/2	61
DIFF_HSTL_II (M)	D	N/R	30	HSLVDCI_25	SE	Vcco/2	62
DIFF_HSTL_II (S)	D	N/R	31	HSLVDCI_18	SE	Vcco/2	63
DIFF_HSTL_II_18 (M)	D	N/R	32	HSLVDCI_15	SE	Vcco/2	64
DIFF_HSTL_II_18 (S)	D	N/R	33	Total Configurations			64
SE = Single-Ended; D = Differential; **Unsupported Non-Bidirectional Standard							