

BUILT-IN SELF-TEST FOR I/O BUFFERS IN FPGAs



Sudheer Vemula and Charles Stroud
Dept. of Electrical and Computer Engineering
200 Broun Hall, Auburn University, AL 36849-5201
email: vemulsu/strouce@auburn.edu

ABSTRACT: A Built-In Self-Test (BIST) approach for the programmable Input/Output (I/O) buffers in Field Programmable Gate Arrays (FPGAs) is presented. The I/O buffers are tested for their various modes of operation along with their associated routing sources. A general BIST architecture, applicable to any FPGA, is presented along with the features and limitations of the approach. Experimental results are given for BIST configurations developed to test the I/O buffers in Atmel AT40K series FPGAs and associated with the FPGA core of Atmel AT94K series System-on-Chip (SoC).¹

1. INTRODUCTION

All Field Programmable Gate Arrays (FPGAs) include programmable Input/Output (I/O) buffers to communicate with the other components present in the system. Typical FPGA I/O buffers have several programmable features as well as associated routing resources to/from of the core of the FPGA. Since defects are possible in any resource of an FPGA, they may also occur in the I/O buffers. If the I/O buffers are faulty, then the information exchange with other components in the system may not be possible or reliable. This implies that the I/O buffers should be tested to ensure the fault-free operation of the FPGA in the same manner as the programmable logic and interconnect resources in the core of the array should be tested. While a number of Built-In Self-Test (BIST) approaches have been developed for the programmable logic and routing resources in the FPGA core [1]-[7], they have neglected testing the I/O buffers and their associated routing resources.

In this paper, we present a BIST approach that has been developed to test the I/O buffers of Atmel AT40K series FPGAs, but we emphasize that this general approach can be applied to I/O buffers of any other FPGA. We begin with an overview of relevant prior work in BIST for FPGAs and in FPGA I/O buffer testing in Section 2 followed in Section 3 by an overview of architectural issues of the AT40K series FPGA and the FPGA core in the AT94K series SoC that impact the I/O buffer BIST approach. A detailed description of the BIST approach is presented in Section 4. Experimental results are given in Section 5 for fault simulations and actual implementations in two different Atmel devices, before concluding the paper with a discussion of limitations of the BIST approach in Section 6.

¹ This work was sponsored by the National Security Agency under contract H98230-04-C-1177.

2. BACKGROUND AND MOTIVATION

A number of BIST approaches have been developed to test the programmable logic and routing resources in FPGAs [1]-[7]. The basic technique is to configure some of the programmable logic blocks (PLBs) as Test Pattern Generators (TPGs) and as Output Response Analyzers (ORAs). These are then used to detect faults in PLBs under test (BUTs) in logic BIST or wires under test (WUTs) in routing BIST.

In logic BIST, the BUTs and ORAs are arranged in alternating columns (or rows) and two or more identical TPGs are used to drive the alternating columns (or rows) of BUTs [2][3][7], as illustrated in Figure 1. The output responses of identically programmed BUTs are compared by ORAs in neighboring columns (or rows). The basic design of the comparison-based ORA is illustrated in Figure 2. During a given test session (Figure 1a), the BUTs are repeatedly reconfigured in their various modes of operation until they are completely tested. During the next test session (Figure 1b), this architecture is flipped and the roles of the PLBs are reversed such that those previously configured as TPGs and ORAs become BUTs and vice versa. The PLBs can be tested in only two test sessions when at least half the PLBs are configured as BUTs during a given test session. While the logic resources of the PLBs are completely tested by most logic BIST approaches, there is logic in the I/O buffers, such as flip-flops and latches, that should also be tested. In fact, logic in unbonded I/O buffers is sometimes used by FPGA synthesis tools to implement the system function.

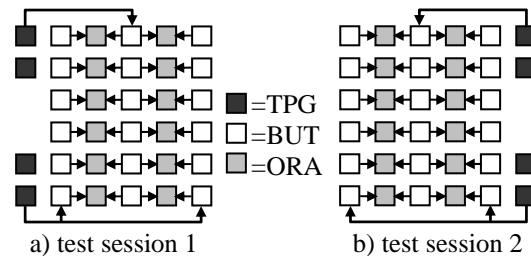


Figure 1. FPGA logic BIST architecture

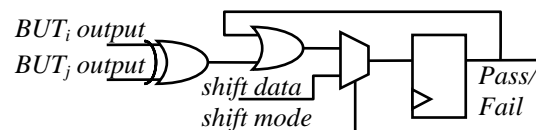


Figure 2. ORA design

Two types of routing BIST approaches have proven to be effective in testing the programmable interconnect resources in FPGAs. The first is a comparison-based approach in which the TPG drives exhaustive test patterns over two sets of WUTs that are monitored at the other end by a comparison-based ORA [4][5]. The second approach is parity-based where the TPG sources exhaustive test patterns over a set of WUTs and produces a parity-bit that is sent to the ORA [6][7]. The ORA generates parity over the data observed on the WUTs and compares the generated parity-bit with that sent by the TPG. While both approaches have been shown to be effective in detecting faults, they have been used almost exclusively to test the routing resources in the core of the FPGA, leaving the routing resources associated with I/O buffers untested.

BIST has been implemented to test the speed of the I/O in [8]. Additional circuitry such as a delay locked loop, test registers and the compare circuits has been included for each register under test. A BIST circuit which can be implemented in an Application Specific Integrated Circuit (ASIC) has been developed. With the BIST circuitry included, the setup and hold time of the registers in the I/O buffer can be tested. But other resources present in an I/O buffer are not tested by this method [8]. A testing approach was proposed in [9] for I_{DDQ} testing of I/O buffers in an FPGA where the test signals were generated both internally and externally but the results were analyzed only by monitoring the signal values at the outputs. It should be noted that this testing technique is not a BIST approach to test the I/O buffers of an FPGA. Instead, it should be observed that internal logic resources were used in [9] for generation of test patterns to the I/O buffers due to the limited external access of I/O buffers, particularly unbonded I/O buffers.

The Boundary Scan external test (EXTEST) mode [10] can be used to test the input and output buffers, tri-state control, and pads. However, the EXTEST capability cannot be used to test the flip-flops used for registered inputs and outputs or the programmable routing resources connecting the I/O buffer to the FPGA core. While the Boundary Scan internal test (INTEST) could be used to test these flip-flops and routing resources, the INTEST feature is not supported in many FPGAs; this includes Atmel AT40K and AT94K series devices [11][12]. Therefore, a BIST approach for I/O buffers would provide the ability to test the flip-flops used for registered inputs and outputs, dedicated programmable routing resources, and other programmable resources (such as pull-up and pull-down circuitry) typically associated with the programmable I/O buffers in FPGAs regardless of what Boundary Scan features are supported or unsupported for a given family or manufacturer of FPGAs.

3. ARCHITECTURAL OVERVIEW OF ATMEL FPGAs

The primary architectural features of the Atmel AT40K series FPGA [11] and FPGA core in the AT94K series SoC [12] that impact the I/O buffer BIST architecture include the PLBs, the I/O buffers, and the routing resources interconnecting the two. The PLB consists of two 3-input Look Up Tables (LUTs), one D flip-flop (with programmable asynchronous set/reset) and several multiplexers (MUXs) to control routing and logic resources. The PLB has a maximum of four data inputs and two outputs. Two of the four inputs can be connected directly from the outputs of directly adjacent and diagonally adjacent PLBs. The remaining two inputs can be connected only by using the global routing resources present around the PLB. The direct adjacent PLB connections are referred to as Y inputs and outputs while the diagonally adjacent connections are referred to as X inputs and outputs. As illustrated in Figure 3, the PLBs located along the edge of the array of PLBs have X and Y connections to/from the I/O buffers. There are two I/O buffers associated with each periphery PLB, referred to as primary and secondary I/O buffers. Primary I/O buffers connect to one PLB using the Y connections while secondary I/O buffers connect to two PLBs using the X connections. Both primary and secondary I/O buffers also have connections (not shown in Figure 3) to the global routing resources associated with their nearest neighbor PLBs [11][12].

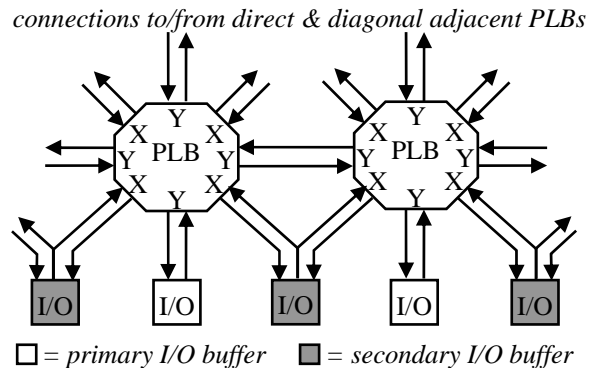


Figure 3. I/O Buffer Connections to PLB

The primary difference between the I/O buffers of the AT40K series FPGA and the I/O buffers associated with the FPGA core of the AT94K series SoC is that the AT94K I/O buffers have two D flip-flops with a reset signal to provide registered input and output signals for critical system timing parameters such as set-up and hold times or clock-to-output delay [12]. Other than these flip-flops, the I/O buffers in both AT40K and AT94K series devices are identical [11][12]. Each I/O buffer contains four multiplexers (MUXs): one used to select the output signal to be sent to the pad, another used to select the tri-state control signal for the output buffer, and the final two multiplexers are used to select

registered or non-registered signals at both the input and output portions of the I/O buffer.

The I/O buffers contain pull-up and pull-down transistors to maintain known voltage levels at the input portion of the buffer when the output portion is in tri-state mode [11][12]. The input buffer has four programmable delays as well as an optional Schmitt trigger circuit to filter out input noise. In addition, the input threshold level can be programmed to be compatible with either TTL or CMOS voltage levels. The output buffer has three programmable drive capabilities for sourcing and sinking different current values. The complete primary I/O buffer is illustrated in Figure 4 with the various components described above. The input and output portions of the I/O buffer share access to programmable routing resources via the transmission gates shown in Figure 4.

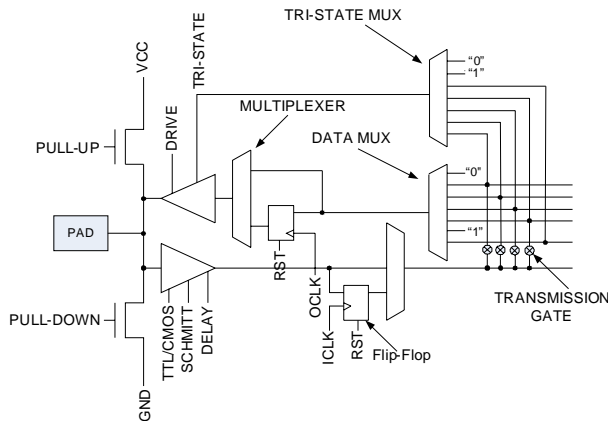


Figure 4. Primary I/O Buffer of Atmel FPGA

The secondary I/O buffer has the same features as the primary I/O buffer. The only difference (besides the connections to the PLBs shown in Figure 3) is in the size of the multiplexers that select the output and tri-state control signals as well as the number of transmission gates. In the case of the secondary I/O buffers, the output and tri-state control multiplexers have six inputs instead of the seven shown in Figure 4. Instead of the four transmission gates shown in Figure 4 for the primary I/O buffer, the secondary I/O buffer has only two transmission gates. Therefore, the secondary I/O buffer represents a subset of the primary I/O buffer.

There are two features associated with both primary and secondary I/O buffers that are not shown in Figure 4. One is the ability for any I/O buffer to drive the global set/reset signal within the FPGA that can be selected as the set/reset signal to the flip-flops of the PLBs as well as the reset signal to the flip-flops in the I/O buffers. The other feature is an additional input to the tri-state control multiplexer that provides for banked control of a set of eight adjacent I/O buffers, including four primary and four secondary I/O buffers.

4. I/O BUFFER BIST ARCHITECTURE

The basic idea of the I/O buffer BIST architecture is to configure some of the PLBs as TPGs and ORAs while the I/O buffers under test are configured as bi-directional buffers. In this way, test patterns produced by the TPG can be applied to the output portion of the I/O buffer while comparison-based ORAs monitor the input portion of two or more identically configured I/O buffers to detect mismatches that result from faults in one or more I/O buffers. As a result, the pad provides a loop-back path of the TPG test patterns to the ORAs such that testing of the complete I/O buffer is accomplished. This BIST architecture is similar in some respects to the logic BIST architecture of Figure 1. Here the I/O buffers replace the BUTs and the PLBs acts as the TPGs and ORAs. The output response of each buffer is compared with the output responses of two other I/O buffers adjacent to it; this greatly reduces the chances of equivalent faults in adjacent I/O buffers escaping detection as is the case in logic BIST [1]. The comparison-based ORAs use the same construction as in logic BIST, as shown in Figure 2. One difference in I/O buffer BIST is that only one TPG is required since the PLBs and routing resources in the core of the FPGA would have previously been tested with logic and routing BIST approaches.

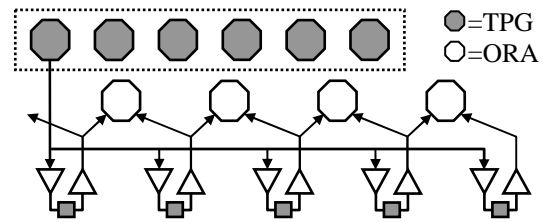


Figure 5. General I/O Buffer BIST Architecture

The architecture shown in Figure 5 can be used to test the I/O buffers of any FPGA. In addition, the comparison-based BIST architecture facilitates the use of diagnostic procedures originally developed for identifying faulty PLBs in logic BIST architectures [2] and later extended to any set of multiple identical structures or cores (like I/O buffers) in FPGAs and SoCs [13]. While there is loss in diagnostic resolution along the edge of the array in logic BIST [2][13], the lower usage of programmable logic and routing resources in I/O buffer BIST allows complete pair-wise comparison of all I/O buffers. In an FPGA, where the programmable I/O buffers extend around the entire perimeter of the FPGA, this circular pair-wise comparison is easily accomplished. In SoCs, however, there are dedicated I/O buffers along one or more edges of the PLB array for interfacing to other internal cores such as processors and/or memories. Therefore, global routing resources are needed to route the output response of an I/O buffer on one side of the

device to be compared with the output response of another I/O buffer on the other side of the device in order to complete the circular comparison. As a result, there is no loss of diagnostic resolution with this circular comparison and new diagnostic procedures [14].

The TPG for the I/O buffer BIST for Atmel FPGAs consists of a 6-bit counter where the Most Significant Bit (MSB) of the counter is used to drive the reset signal to the flip-flops of the I/O buffer while the remaining five bits of the counter are applied as inputs to the output and tri-state control multiplexer inputs. The I/O buffers are repeatedly reconfigured in their various modes of operation such as activated pull-up or pull-down, selection of different multiplexer inputs, registered versus non-registered inputs and outputs, etc. Similarly, the routing resources associated with the I/O buffers are also tested via the connections made from the TPGs and to the ORAs. The minimum number of BIST configurations needed to test the I/O buffers and their associated routing resources is primarily determined by the number of inputs to the largest multiplexer in the I/O buffer. That would be the tri-state control multiplexer in this case. This implies (eight) configurations for the primary I/O buffer (for the seven inputs shown in Figure 4 plus the bank tri-state control input) and (seven) configurations for the secondary I/O buffer. The developed BIST configurations will be discussed in more detail in the next section. This BIST architecture facilitates the testing of all bonded or un-bonded I/O buffers and is, therefore, package independent.

Due to the dedicated X and Y connections associated with each type of I/O buffer (primary or secondary) to/from the PLB illustrated in Figure 3, the PLBs along the periphery of the array are used for routing test patterns from the TPG and output responses to the ORAs. As a result, the TPGs and ORAs are located in the center of the array. Another ramification of the connections between the periphery PLBs and the I/O buffers is that primary and secondary I/O buffers are tested in two different test sessions where each test session consists of the set of BIST configurations associated with the set of I/O buffers under test in that session. For example, when testing the secondary I/O buffers, each I/O buffer has direct connections to the X inputs and outputs of two PLBs, as shown in Figure 3, and both sets of connections must be tested. By using the PLBs along the periphery of the array for routing purposes, these connections can be tested with minor routing modifications to the BIST configurations while maintaining the locations of the PLBs configured as TPGs and ORAs. However, not all features and resources of the I/O buffers are completely tested with this BIST architecture.

For example, different BIST configurations can be used to test the transmission gates, shown in Figure 4,

for stuck-off faults (stuck-on faults are detected by the BIST configurations using the architecture shown in Figure 5). To test stuck-off faults in the transmission gates in the I/O buffers associated with the FPGA core of the AT94K series SoC, opposite logic values are stored in the two flip-flops of the I/O buffer which can be done while the test is performed in the previous BIST configuration. Next, one of the transmission gates is activated by dynamic partial reconfiguration such that there is a feed-back loop from the output of the input buffer to the input of the output buffer, as illustrated in Figure 6. As the flip-flops are clocked, the toggling output is monitored by the ORAs for a few clock cycles to observe whether the transmission gates are able to pass both logic values. This same procedure is performed in turn for each transmission gate in the I/O buffer.

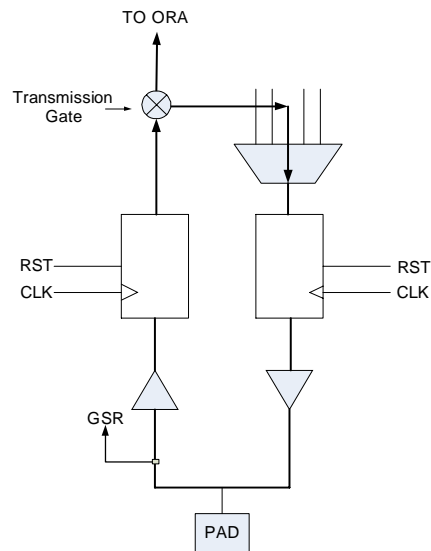


Figure 6. Transmission Gate Test Configurations

The lack of flip-flops in the I/O buffers of the AT40K series FPGA requires a different approach to testing the transmission gates. The transmission gates can be tested with the same BIST architecture shown in Figure 5 but the direction of signal flow in the routing resources associated with the transmission gates must be changed from one configuration to the next. As a result, the TPG to I/O buffer routing cannot be maintained from one BIST configuration to the next since the output response from the input portion of the I/O buffer must be routed through a different transmission gate to the ORA in each subsequent BIST configuration. This increases the BIST configuration development effort for the AT40K series device.

Another example of a feature that requires a different type of BIST configuration is the ability of each I/O buffer to drive the global reset. However, at any given time, only one I/O buffer can drive the global reset. As a result, a separate BIST configuration is

needed for each I/O buffer in order to test its ability to drive the global reset. All previously discussed BIST approaches are independent of the array size [1]-[7] or, in this case, the number of I/O buffers. As a result, the BIST test time (excluding download time) is constant and independent of the array size. However, the number of configurations required to test the global reset input on all of the I/O buffers is equal to the number of programmable I/O buffers associated with the FPGA core (this characteristic appears to be unique to Atmel FPGAs). Fortunately, by developing a regular BIST architecture and using partial reconfiguration, this feature can be tested without a significant impact on total test time. The BIST architecture used to test the global reset connections routes a TPG signal in a daisy-chain through all of the I/O buffers configured as non-registered, bi-directional buffers. Each I/O buffer is configured, in turn, to drive the global reset with the test pattern from the TPG in one of the partial reconfigurations. Meanwhile, a logic 1 is clocked into the input of a flip-flop whose output is monitored by a comparison-based ORA and compared with the expected output response of the flip-flop generated by the TPG. As a result, stuck-off faults in the programmable switch connecting each I/O buffer to the global reset are detected if the test pattern fails to reset the flip-flop. Stuck-on faults in the programmable switch connection from each I/O buffer to the global reset are detected in the BIST configurations illustrated in Figure 5 since a stuck-on fault would cause an inadvertent reset during the BIST sequence.

5. IMPLEMENTATION RESULTS

Due to the restrictions of the dedicated routing resources between the I/O buffers and the periphery PLBs in the array, the primary and secondary I/O buffers are tested in separate sets of BIST configurations. A set of nine BIST configurations were developed to test the resources in the primary I/O buffers and a set of eight BIST configurations were developed for the secondary I/O buffers. These BIST configurations exclude the transmission gates and global reset input in each I/O buffer. Four additional BIST configurations were developed to test the four transmission gates in the primary I/O buffers and two BIST configurations were developed to test the two transmission gates in the secondary I/O buffers. Therefore, a total of 23 BIST configurations were developed to test the resources in all of the I/O buffers with the exception of the global reset input connection. This set of 23 BIST configurations is independent of the array size or the number of I/O buffers associated with the FPGA core in any Atmel AT94K series SoC. A subset of the 23 BIST configurations can be used to test the I/O buffers of the AT40K series FPGA since those I/O buffers do not contain the D flip-flops for registered input and outputs.

Fault simulations were performed to verify stuck-at gate-level fault coverage for the nine primary I/O buffer BIST configurations and eight secondary I/O buffer BIST configurations. The fault simulation results are shown in Figure 7 in terms of the individual and cumulative fault coverage obtained with each BIST configuration. When the BIST configurations for the transmission gates are included, fault coverage of 98.1% is obtained for the primary I/O buffers and fault coverage of 97.8% is obtained for the secondary I/O buffers with the only undetected faults remaining being stuck-off faults associated with the global reset input which will be detected with the global reset BIST configurations.

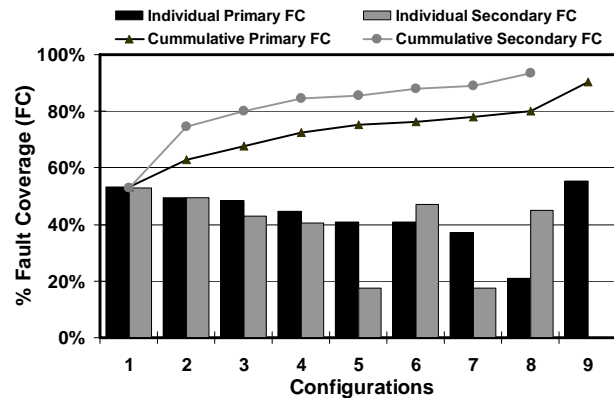


Figure 7. I/O Buffer Fault Coverage

To generate the complete set of BIST configurations, three master BIST configurations were developed (one for primary I/O buffers, one for secondary I/O buffers, and one for global reset tests for both primary and secondary) using Atmel's Macro Generation Language. The remaining BIST configurations were then automatically generated from the three master configurations by manipulating the bits in the configuration files to be downloaded into the FPGA to run BIST sequence.

6. SUMMARY AND CONCLUSIONS

A general BIST approach was presented to test the programmable I/O buffers and their associated routing resources in an FPGA. This BIST approach is applicable to any FPGA but we have specifically developed the BIST configurations to test the I/O buffers of Atmel AT40K series FPGAs and the FPGA core in Atmel AT94K series SoCs. For these devices, 100% stuck-at gate-level fault coverage was obtained indicating that the I/O buffer BIST approach can detect all the faults present in the logic and routing resources associated with each I/O buffer. While the BIST can also detect major defects that affect the analog programmable features (such as pull-up, pull-down, tri-state, etc.), it cannot, however, detect all parametric faults such as V_{OL} , V_{OH} , V_{IL} , V_{IH} , current sink and source capabilities, delay, etc. It should be noted that the BIST will detect

faults in the configuration bits that control the programmable analog parametric features such as drive capability, delay, and voltage levels. Therefore, the BIST approach provides a good sanity test of the I/O buffers.

A total of 23 BIST configurations were developed to test the primary and secondary I/O buffers associated with the FPGA core of the Atmel AT94K series SoC and a subset of the same BIST configurations can be used for the I/O buffers in the AT40K series FPGA. While this number of BIST configurations is independent of array size and the number of I/O buffers, it is a relatively large number of BIST configurations compared to the 16 configurations for logic BIST and 48 configurations for routing BIST in these FPGAs [7]. This is primarily due to routing restrictions in the connections between I/O buffers and periphery PLBs in the array which requires that the primary and secondary I/O buffers be tested in separate testing sessions. This characteristic appears to be unique to the Atmel FPGAs along with the global reset input connection in each I/O buffer. Testing this global reset connection leads to a number of BIST configurations that is a function of the number of I/O buffers, unlike the other 23 BIST configurations as well as logic and routing BIST configurations that have been previously developed for the core resources of FPGAs [1]-[7]. Fortunately, a single BIST configuration can be used in conjunction with partial reconfiguration to cycle through the global reset input connections and test all I/O buffers without having to download a new BIST configuration for each I/O buffer. This is particularly efficient in the AT94K series SoC where the embedded processor core can be used to algorithmically reconfigure the I/O buffers to test each global reset input in turn. As a result, only one BIST configuration download is required followed by the execution of a program in the embedded processor to reconfigure and execute the BIST sequence, where the execution time is proportional to the number of I/O buffers.

The BIST approach described in this paper can be used in manufacturing testing at wafer-level and device-level testing. Since it can test all unbonded as well as all bonded I/O buffers, it can be used to test packaged devices without additional test development or increase in testing complexity due to limited access to unbonded pads. Testing unbonded I/O buffers is important since FPGA synthesis tools sometimes use the logic and routing resources associated with unbonded I/O buffers for implementing the intended system function. The BIST approach can also be used for system-level testing only if all other connecting devices can be tri-stated since all I/O buffers are configured as bi-directional buffers during the I/O buffer BIST sequence.

REFERENCES

- [1] C. Stroud, *A Designer's Guide to Built-In Self-Test*. Kluwer Academic Publishers, Boston MA, 2002.
- [2] M. Abramovici and C. Stroud, "BIST-Based Test and Diagnosis of FPGA Logic Blocks," *IEEE Trans. on VLSI Systems*, Vol. 9, No. 1, pp. 159-172, 2001.
- [3] C. Stroud, K. Leach, and T. Slaughter, "BIST for Xilinx 4000 and Spartan Series FPGAs: A Case Study", *Proc. IEEE International Test Conf.*, pp. 1258-1267, 2003.
- [4] D. Fernandes and I. Harris, "Application of Built-In Self-Test for Interconnect Testing of FPGAs", *Proc. IEEE International Test Conf.*, pp. 1248-1257, 2003.
- [5] C. Stroud, J. Nall, M. Lashinsky and M. Abramovici, "BIST-Based Diagnosis of FPGA Interconnect," *Proc. IEEE International Test Conf.*, pp. 618-627, 2002.
- [6] X. Sun, J. Xu, B. Chan and P. Trouborst, "Novel Technique for BIST of FPGA Interconnects," *Proc. IEEE Int'l Test Conf.*, pp. 795-803, 2000.
- [7] C. Stroud, J. Sunwoo, S. Garimella and J. Harris, "Built-In Self-Test for System-on-Chip: A Case Study", *Proc. IEEE International Test Conf.*, pp. 837-846, 2004.
- [8] Cheng Jia and L. Milor, "A BIST Solution for the Test of I/O Speed," *Proc. IEEE International Test conf.*, pp. 1023-1030, 2003
- [9] L. Zhao, D. Walker and F. Lombardi, "I_{DDQ} Testing of Input/Output Resources of SRAM-Based FPGAs," *Proc. Asian Test Symp.*, pp. 375-380, 1999
- [10] C. Maunder and R. Tulloss, *The Test Access Port and Boundary Scan Architecture*, IEEE Computer Society Press, 1990
- [11] ___, "AT40K Series Field Programmable Gate Array," Data Sheet, Atmel Corporation, 2002, available at www.atmel.com
- [12] ___, "AT94K Series Field Programmable System Level Integrated Circuit," Data Sheet, Atmel Corp., 2001, available at www.atmel.com
- [13] C. Stroud, S. Garimella, and J. Sunwoo, "On-Chip BIST-Based Diagnosis of Embedded Programmable Logic Cores in System-on-Chip Devices," *Proc. ISCA International Conf. on Computers and Their Applications*, pp. 308-313, 2005
- [14] C. Stroud and S. Garimella, "Built-In Self-Test and Diagnosis of Multiple Embedded Cores in SoCs," to be published *Proc. International Conf. on Embedded Systems and Applications*, 2005