from Proc. International Conf. on Embedded Systems & Applications, pp. 86-92, 2010

The First Clock Cycle Is A Real BIST

Charles E. Stroud and Bradley F. Dutton
Dept. of Electrical and Computer Engineering
Auburn University, Alabama, USA

Abstract—*The primary goal of Built-In Self-Test (BIST) for Field Programmable Gate Arrays (FPGAs) is to completely test all programmable logic and routing resources in the device such that the user can be assured that their system function is downloaded to a fault-free device. In this paper, we present case studies of developing BIST configurations for some of the less thought of, though equally important, logic resources in current FPGAs including programmable clock buffers and cyclic redundancy code (CRC) modules. While seemingly trivial, these modules present interesting testing challenges in FPGAs. Among the most interesting challenges are the problems, and importance, associated with the very first clock cycle of the BIST sequence when testing FPGA initialization capabilities, or lack thereof, via the configuration download. While the conditions may last for only one clock cycle, testing these features is essential to insure the intended system function is properly initialized during operation.* [1]

1. INTRODUCTION

Built-In Self-Test (BIST) for Field Programmable Gate Arrays (FPGAs) has been a research and development topic of interest for the past fifteen years [1]-[5]. The basic idea of BIST for FPGAs is to configure the FPGA to test itself, such that the test logic disappears after the off-line test is complete and there is no area or performance penalty during normal operation of the intended system function. Due to their programmable nature, resources under test must be reconfigured multiple times during the course of testing in order to test all modes of operation and ensure high fault coverage. BIST approaches have been developed for primary programmable resources including configurable logic blocks (CLBs) [3], random access memories (RAMs) [4], digital signal processors (DSPs) [5], and input/output (I/O) logic [6]. Yet, there are a number of resources that have not been addressed including programmable resources such as global clock buffers and Cyclic Redundancy Check (CRC) modules. These particular resources are of interest since they present some of the most interesting testing challenges associated with FPGAs, and more specifically, many of these challenges are associated with the first clock cycle of the BIST sequence when testing FPGA initialization capabilities, or lack thereof, via the configuration download.

This paper presents complete BIST approaches for the programmable global clock buffers in Xilinx Virtex-4 [7] and Virtex-5 [8] FPGAs as well as the CRC modules in Virtex-5 FPGAs [9][10]. We begin with a general overview of the BIST architecture and operation in Section 2. BIST configurations for CRC modules and global clock buffers are presented in Sections 3 and 4, respectively, along with the specific challenges they pose as well as experimental results from actual implementation and execution in FPGAs. The paper is summarized and concludes in Section 5.

1 This work was supported in part by the National Science Foundation Grant CNS-0708962.

2. BIST ARCHITECTURE

In the general BIST architecture, programmable logic resources, such as CLBs or DSPs, are configured as Test Pattern Generators (TPGs) and CLBs are configured as Output Response Analyzers (ORAs). Global clock buffers and CRC modules are configured as Blocks Under Test (BUTs). As illustrated in Figure 1, each output of every BUT is monitored by two ORAs and compared with the outputs of two other identically configured BUTs, making it highly improbable that any set of multiple faults can escape detection. Multiple TPGs are used to supply test patterns to alternating BUTs which ensures that any faults in the TPGs will be detected.

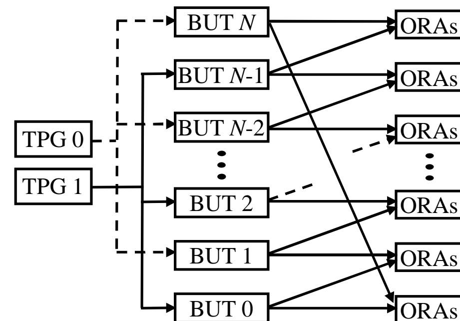

Fig. 1. Circular comparison BIST architecture

Comparison-based ORAs, illustrated in Figure 2, latch a logic 0 in the ORA flip-flop anytime a mismatch between two identically configured BUT outputs is observed. 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 BIST results were obtained via partial configuration memory read back where the contents of every ORA flip-flop are retrieved from the configuration memory. However, the ORA design utilizes the dedicated carry logic in the CLB to form an iterative-OR of all ORA outputs [3]. In each ORA, a passing result of logic 1 selects the Carry-in input, which is the Pass/Fail result of the previous ORA. This comparison-based ORA design provides a single-bit

pass/fail result for all resources under test. However, partial configuration memory read back can still be used when faults are detected so that faulty resources can be diagnosed.

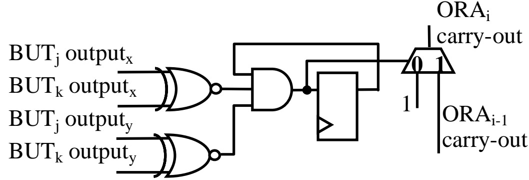

Fig. 2. ORA architecture [3]

Due to the programmable nature of the resources in FPGAs, the target resources under test must be reconfigured and tested multiple times in order to test their various modes of operation. A test session consists of a set of multiple BIST configurations needed to completely test the target resource in all of its modes of operation. For each BIST configuration in a given test session, the BIST configuration must be downloaded to the FPGA, the BIST sequence must be executed, and the BIST results must be retrieved to determine the pass/fail status for that BIST configuration. Therefore the total test time, T_{test}, for a given test session is given by:

$$T_{test} = \sum_{i=1}^{N} (D_i + E_i + R_i) \quad (1)$$

where D_i, E_i, and R_i represent time required to *Download* the BIST configuration, *Execute* the BIST sequence, and *Retrieve* the BIST results, respectively, for BIST configuration i. In addition, N represents the number of BIST configurations for the test session. For a large number of test sessions and BIST configurations per test session, the total test time can be considerable. Therefore, a primary goal in BIST for FPGAs is to minimize the number of test sessions as well as the number of BIST configurations for each test session.

The download time tends to be the dominant factor in Equation 1, particularly for the larger devices, while the BIST results retrieval time typically has the second largest impact. Therefore, another important goal is to develop and use techniques that minimize the download time as well as the results retrieval time. This includes the use of compressed configuration bit files where the regularity of the BIST architecture and multi-frame write features of recent FPGAs are exploited to write a single frame of configuration data to multiple frame addresses [11][12]. Similarly, partial reconfiguration files are used to change BUT configurations between each BIST sequence. The BIST results retrieval time is made negligible by incorporating the comparison-based ORA with iterative-OR chain illustrated in Figure 2 and described above.

3. CRC MODULE BIST

There are two CRC modules associated with each gigabit transceiver module in the right hand column of the Virtex-5 LXT, SXT, and FXT devices as well as both the left and right hand columns of the TXT devices [9][10]. Each CRC module (Figure 3b) can function as two independent 32-bit CRC (CRC32) modules or as one 64-bit CRC (CRC64) module. All input and output pin connections of the CRC64 module are shared with one of the CRC32 modules including the 32 outputs CRCOUT as well as the CRCCLK, CRCRESET, and CRCDATAVALID inputs with the CRCIN[63:32] inputs shared with the CRCIN[31:0] inputs of its associated CRC32 module while the CRCIN[31:0] inputs of the CRC64 module are shared with the CRCIN[31:0] inputs of the other CRC32 module. The CRCDATAWIDTH[2:0] inputs indicate how many and which input data bytes are valid during any given clock cycle.

The only programmable options associated with the CRC module are the active edge of the CRCCLK and an initialization value specified by CRC_INIT[31:0] that is synchronously loaded into the CRC register when the CRCRESET input is active. It should be noted, however, that the CRC register cannot be initialized via a download to the FPGA before system operation begins.

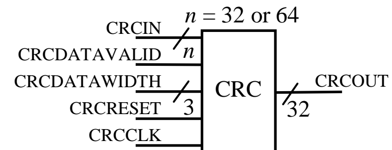

Fig. 3. CRC modules

Using information collected from datasheets [9][10], an accurate gate level model was developed for the CRC32 module as illustrated in Figure 4 for a single unit cell, where the XOR gate shown in gray is only used in those cells that include a feedback connection for characteristic polynomial:
$P(x)=x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

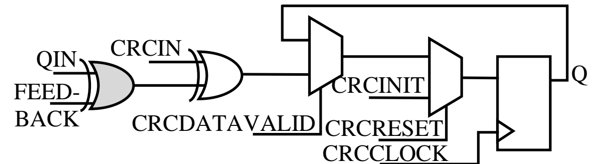

Fig. 4. Unit cell for CRC32 module fault simulations

Fault simulation was used to determine the appropriate test pattern sequences and configuration options to achieve 100% single stuck-at gate level fault coverage (for 921 collapsed stuck-at gate-level faults for a single CRC32 module). From this analysis we determined that the CRC32 modules could be completely tested with the two BIST configurations summarized in Table I using a 10-bit counter as the TPG. The resultant fault coverage is given in Table II for both gate-level fault simulation and configuration memory bit fault injection emulation analysis that was later performed after the final BIST configurations were developed and implemented in Virtex-5 FPGAs. The TPG counter bits are assigned to CRC module inputs as summarized in Table III.

TABLE I. CRC BIST Configurations

Config Element	Config #1	Config #2
CRCCLKINV	Non-Inverted	Inverted
CRC_INIT	0xAAAAAAAA	0x55555555
Functional Mode	CRC64 mode	CRC32 mode

TABLE II. CRC BIST Fault Detection

Fault Coverage		Config #1	Config #2
Fault Simulation (921 faults)	Individual	90.05%	89.73%
	Cumulative	90.05%	100%
Fault Injection (68 faults)	Individual	50%	50%
	Cumulative	50%	100%

TABLE III. TPG Bit Assignements for CRC Inputs

Counter Bit	CRC Module Input
0	CRCIN0, CRCIN4, CRCIN8, … CRCIN60
1	CRCIN1, CRCIN5, CRCIN9, … CRCIN61
2	CRCIN2, CRCIN6, CRCIN10, … CRCIN62
3	CRCIN3, CRCIN7, CRCIN11, … CRCIN63
4	CRCDATAWIDTH0
5	CRCDATAWIDTH1
6	CRCDATAWIDTH2
7	CRCDATAVALID
8	CRCRESET
9	ORA Clock Enable

Since the CRC64 module shares all pin connections (input and output pins) with one of the CRC32 modules (and the CRCIN inputs to the other CRC32 module), the first BIST configuration tests the CRC64 module along with the CRC32 module that does not share output pin connections with the CRC64 module. As a result, all 64 outputs of both CRC32 modules (or the CRC64 module and one CRC32 module) are connected to the ORAs during both BIST configurations such that the two BIST configurations maintain constant routing, including TPG-to-BUT connections and BUT-to-ORA connections, and only the configuration options of the CRC modules change. This allows testing all modes of operation in only two BIST configurations while minimizing the download time for the second BIST configuration via partial reconfiguration and achieving 100% fault coverage. This was verified by configuration memory fault injection analysis of the actual BIST configurations developed for the CRC BIST and downloaded into an actual FPGA as indicated in Table II for 68 stuck-at faults controlled by 34 configuration memory bits for a CRC32 module including the configuration memory bit that controls CRC64 versus CRC32 operation.

When we downloaded and verified our initial CRC BIST configurations, we encountered a number of output response analyzer (ORA) failure indications. As we investigated these in an attempt to debug the BIST configurations, we found that the failure indications occurred during the first clock cycle of the BIST sequence and the failure indications were not consistent with repeated downloads of the same BIST configuration. In fact, we encountered a few cases where the BIST sequence ran without any failures. Upon further investigation, we determined that the CRC_INIT is actually a set/reset value in that the specified value is only loaded into the CRC register when the CRCRESET is active such that the register is not initialized during the download and subsequent start-up process (as opposed to the typical case for INIT values of other registers, flip-flops, and memory elements in the array which are initialized during the download and start-up process). To overcome this problem, we included a Clock Enable on the ORAs which is controlled by the most significant bit of the TPG counter (as indicated in Table III). As a result, the ORAs do not monitor the CRC outputs during the first half of the 1,024 clock cycle BIST sequence, allowing the CRC modules to be initialized by the CRCRESET during the first 512 clock cycles of the BIST sequence.

We developed a C program, *V5crcBIST.c*, which generates a BIST template file in Xilinx Design Language (XDL) format, and another C program, *V5crcMOD.c*, which generates the two BIST configurations (summarized in Table I) from the routed BIST template XDL file. The two programs support all Virtex-5 FPGAs including LXT, SXT, FXT, and TXT devices. The XDL files are converted into an NCD file, as illustrated in Figure 5, using the Xilinx program, *XDL.exe*. The NCD file can be viewed and/or routed in FPGA Editor. The Xilinx tool, *BitGen.exe*, converts the NCD file to a configuration bitstream which can then be downloaded into FPGAs for the actual testing process.

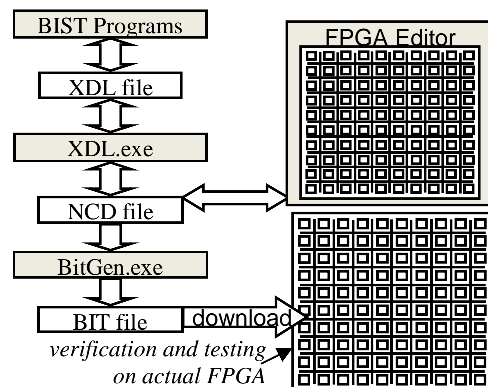

Fig. 5. Automatic BIST configuration generation

The unrouted and routed BIST template file is illustrated in Figure 6 for a Virtex-5 TX150T. The ORAs for the circular comparison of the outputs of an identically configured column of CRC modules are located in the CLB columns nearest to the columns containing the CRC modules. In the TX150T in Figure 6, the CRC modules are located in the left most and right most columns of the array with the ORAs located in the adjacent CLB columns on the left side and two columns to the left on the right side (skipping the PCIE/TEMAC column on the right). In the LXT, SXT, and FXT devices, there is only one column of CRC modules on the right side of the array with the ORAs located in the CLBs two columns to the left (again skipping the PCIE/TEMAC column). The TPGs are DSPs configured as counters. In the TXT150T device in Figure 6, the DSP

column is located about half way between the left most column and the middle of the array.

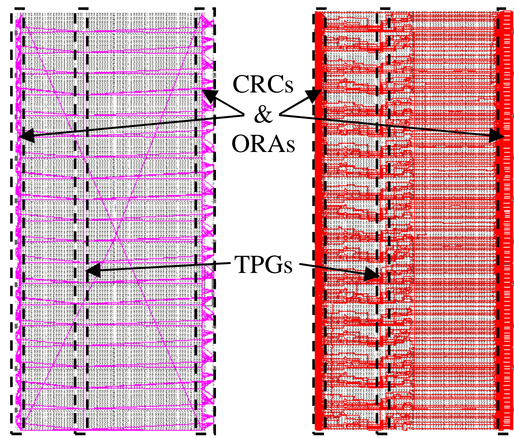

Fig. 6. Unrouted and routed CRC BIST in TX150T

We generated the CRC BIST configurations for all of the Virtex-5 devices to ensure that the BIST programs were working properly. We ran Xilinx *Trace.exe* timing analysis on all of the configurations to determine the maximum BIST clock frequency in each case. The results are summarized in Figure 7 where it can be seen that first BIST configuration runs in excess of 60 MHz for all Virtex-5 FPGAs while the second BIST configuration runs at roughly half that frequency due to opposite edge clocking when testing the CRCCLK invert configuration option as indicated in Table I.

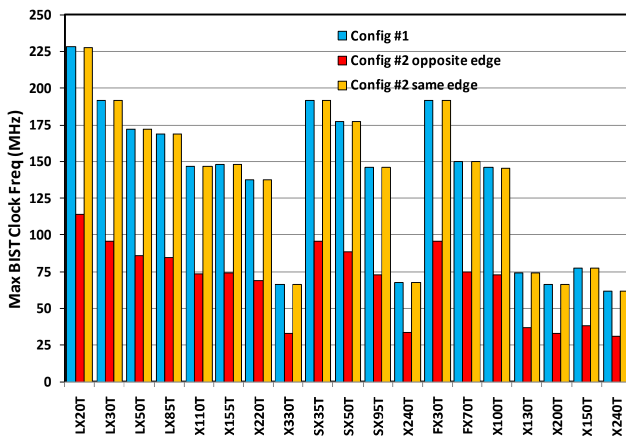

Fig. 7. Timing analysis for CRC BIST configurations

By inverting the active clock edge in the TPGs and ORAs during the second BIST configuration, the maximum BIST clock frequency is roughly the same as that for the first BIST configuration as can be seen in Figure 7. This is done as part of the *V5crcMOD.exe* program so that the routing remains constant for both BIST configurations to help minimize the size of the partial reconfiguration files, minimize download time, and thereby minimize testing time. However, the partial reconfiguration files are larger since the frames containing the configuration memory bits that control the active edge of the clock for the TPGs and ORAs must be rewritten as part of the second BIST configuration. Therefore, the increase in maximum BIST clock frequency would also increase the size of the partial reconfiguration files and associated download and testing time for BIST configurations to test falling edge clocking options.

We generated the partial reconfiguration files for all of the Virtex-5 devices to see the number of frames that change for the two clocking approaches. For each column of CRC modules there are four frames that are written per configuration row. In order to change the clock edge of the TPGs and ORAs there is one frame per configuration row for the DSPs and one frame to change per configuration row per column of ORAs. Therefore, there is an increase from four frames to six frames per configuration row for LXT, SXT, and FXT devices. For TXT devices there is an increase from eight frames to eleven frames per configuration row. This corresponds to a 50% increase in download time for LXT, SXT, and FXT and a 37.5% increase for TXT devices (there is also some overhead in terms of instructions to the configuration interface and registers for the additional frames). It should be noted, however, that this increased download time applies to only those partial reconfiguration files in which we inverted the clocks in the TPGs and ORAs and not the entire testing time. As a result, the increase in total test time is small; in all cases there is less than 1% increase in total test time (the LX20T device being the worst case at 0.873%). Note that the total test time includes an additional 1200 clock cycles for each BIST configuration needed for application of the BIST clock (1050 clock cycles) and for the overhead of Boundary Scan instructions for download and execution of each BIST configuration (estimated at 150 clock cycles), but this additional 2400 clock cycles is insignificant compared to the size of the compressed download file for the first BIST configuration which range from 1,858,688 for an LX20T to 17,183,794 for a TX240T.

We downloaded and verified the CRC BIST configurations in Virtex-5 LX30T, LX50T, SX35T, SX50T, FX30T and FX70T devices. Furthermore, we performed fault injection emulation analysis on all four CRC32 modules and both CRC64 modules associated with a single gigabit transceiver. The two BIST configurations detected all 268 configuration memory bit stuck-at faults which includes the two configuration memory bits that control CRC64 operation. Finally, every configuration memory bit fault injected was detected by a total of 32 ORAs over the course of the 1,024 clock cycle BIST sequence for the BIST configuration detecting that fault.

4. Clock Buffer Built-In Self-Test

There are 32 clock buffer (BUFGCTRL) modules located along the “center” column of Virtex-4 [7] and Virtex-5 [8] FPGAs. These buffers are used to drive dedicated clock routing resources in the device as well as the general routing resources for other high fan-out signals in a system design.

These modules operate in seven different modes including BUFGCTRL which represents a superset of the other six modes of operation [7][8]. Therefore, the BUFGCTRL mode is used for BIST in which there are eight inputs to each buffer and one output illustrated in Figure 8a. Six of the inputs (S0, CE0, IGNORE0, IGNORE1, CE1, and S1) facilitate and control glitch-free switching between two asynchronous clock inputs (I0 and I1) with the ability to program switching to occur either on the rising or falling edge as well as the ability to ignore those edges. The six control inputs have programmable active levels, either high or low. The output can be programmed to initially be logic 0 or 1 (using the INIT_OUT configuration option) as well as to preselect one of the two inputs upon configuration of the device and just prior to normal system operation (using the PRESELECT_I0 and PRESELECT_I1 configuration options). This ability to initialize or preselect the clock buffer output value prior to system operation can be an extremely important feature given that these buffers can be driving critical system signals such as clocks and global control signals.

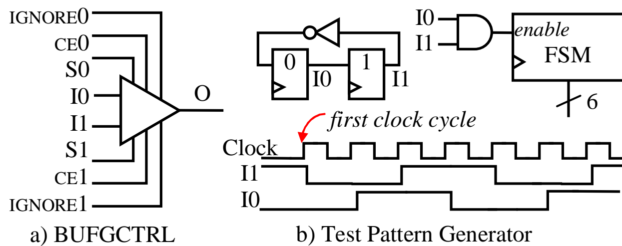

Fig. 8. Clock buffer BUFGCTRL and TPG

With only one output, each ORA for clock buffer BIST monitors only two outputs (as opposed to the four outputs monitored in Figure 2) to achieve diagnostic resolution to the faulty buffer. The clock buffers can be completely tested with two BIST configurations summarized in Table IV for the nine configuration options associated with each clock bufffer. This was verified via configuration memory bit fault injection analysis in an actual FPGA [13], as summarized in Table V. Six faults were detected by the first clock cycle.

Two TPGs are incorporated to drive alternating clock buffers. The challenge in this case was to be able to test all 32 buffers in a single test session in order to minimize the number of downloads and the resulting testing time. This was challenging due to the limited routing associated with the clock buffers as well as the limited access from the dedicated clock trees driven by the buffers to the general routing network for connection to the data inputs of the ORAs. We were able to accomplish this in the end by including look-up table (LUT) buffers to allow a single load on the clock buffers before fanning out to the two ORAs monitoring each buffer output signal.

Each TPG consists of a 2-bit twisted ring counter initialized to 01 and a finite state machine (FSM) as shown in Figure 8b. The FSM is enabled once each cycle of the twisted ring counter and generates the test patterns to the six control inputs (S0, S1, CE0, CE1, IGNORE0, and IGNORE1) summarized in Table VI. Note that the I1 and I0 inputs are supplied by the outputs of the twisted ring counter flip-flops. As a result, the I0 and I1 input signals have the same frequency but 90 degrees out of phase. This is important to set up the conditions for switching between the inputs with control signals to test for glitch-free operation when used for the selection of clocks. The one clock cycle delay provides all four combinations of input values as well as providing opposite logic values at the inputs upon configuration by initializing the I0 flip-flop to logic 0 and the I1 flip-flop to logic 1, as indicated in Figure 8b. This is critical to the detection of the six configuration memory bit faults can only be detected on the first clock cycle. These faults are associated with the PRESELECT_I0, PRESELECT_I0, and INIT_OUT configuration options.

TABLE IV. Clock Buffer BIST Configurations

Config Element	Config #1	Config #2
S0	Non-Inverted	Inverted
CE0	Non-Inverted	Inverted
IGNORE0	Non-Inverted	Inverted
S1	Non-Inverted	Inverted
CE1	Non-Inverted	Inverted
IGNORE1	Non-Inverted	Inverted
PRESELECT_I0	True	False
PRESELECT _I1	False	True
INIT_OUT	1	0

TABLE V. Clock Buffer BIST Fault Detection

Fault Coverage		Config #1	Config #2
Fault Injection (18 faults)	Individual	50%	50%
	Cumulative	50%	100%

TABLE VI. Test Patterns for Clock Buffer Controls

BUFGCTRL Inputs					
IG1	IG0	CE1	S1	CE0	S0
0	0	1	1	0	1
0	0	1	1	1	0
0	0	1	1	1	1
0	0	0	0	0	0
0	1	1	1	1	1
1	0	1	1	1	1
0	0	0	1	1	1
0	0	1	0	1	1

Implementations of clock buffer BIST configurations are illustrated in Figure 9 in Virtex-4 FX12 and Virtex-5 FX30T devices. While the BIST circuitry takes very little room as can be seen in the FX12 implementation in Figure 9a, the placement of the components is critical in order to achieve complete routing. A close-up of the clock buffer BIST is shown in Figure 9b where nets highlighted in dark blue are the routes from clock buffers under test to the LUT buffers and nets highlighted in green are the routes from the LUT buffers to the ORAs. The remaining nets are associated with the TPGs including the TPG-to-BUT routing. A critical issue in the clock buffer BIST is that LUT buffer locations must straddle both horizontal and vertical center lines of the

device in order to properly distribute the output signals of the BUFGs over the dedicated clock resources and connect to the LUT buffers. As a result, the left hand LUT buffers must be moved to the left hand side of the PowerPC in Virtex-5 FX30T and FX70T FPGAs, as illustrated in Figure 9c.

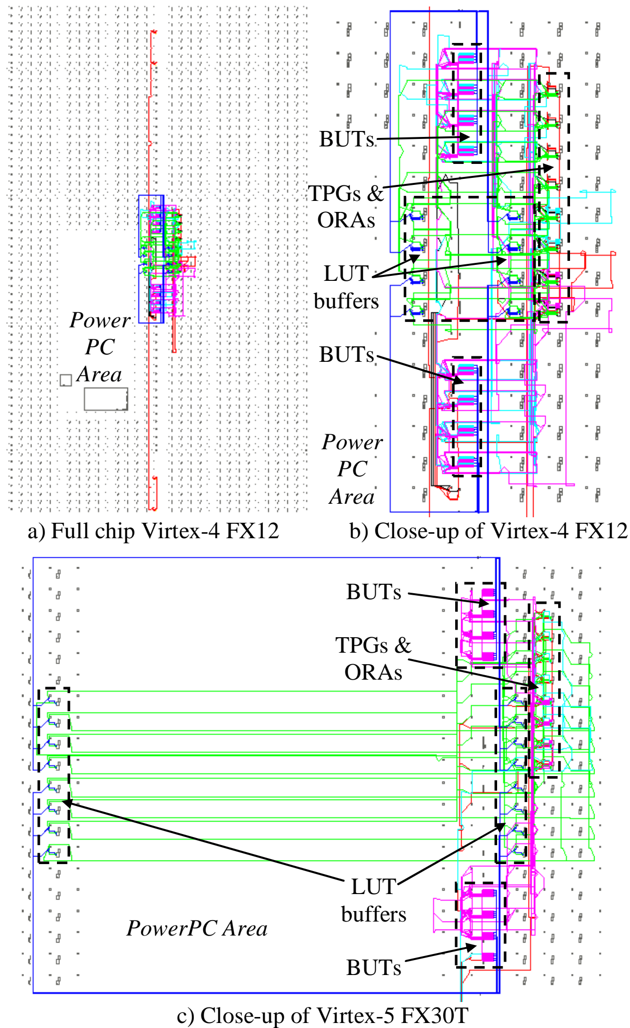

Fig. 9. Clock buffer BIST in Virtex-4 and Virtex-5

The clock routing to the TPGs and ORAs cannot use the clock buffers or the dedicated clock routing since these are occupied by the clock buffers under test and their output signals. Therefore, the TPGs and ORAs are placed close to each other and to the clock source to limit the routing of the clock through normal general signal routing resources. Also the ORAs are located such that the contents can be read in a single frame during configuration memory read back for BIST results retrieval when fault diagnosis is desired. However, the ORAs also have the single pass/fail output as was the case in CRC BIST configurations.

Our C program for generating the BIST configurations for all Virtex-5 devices is *V5bufgBIST.c*. Our C program for generating the BIST configurations for all Virtex-4 devices is *V4bufgBIST.c*. Since the 32 clock buffers in Virtex-4 have the same configurable elements with the same options as Virtex-5, a single C program, *V45bufgMOD.c*, modifies the routed BIST template for either Virtex-4 or Virtex-5 using the same design flow as illustrated in Figure 5. Since there is one column of CLBs between the PowerPC and the center column of Virtex-4 FX devices (illustrated in Fig. 9b), there is no need to move the LUT buffer locations as was the case in Virtex-5 FX30T and FX70T (illustrated in Fig. 9c).

The BIST approach for clock buffers has been successfully implemented in Xilinx Virtex-4 and Virtex-5 FPGAs. The approach is applicable to any device in the Virtex-4 and Virtex-5 families. It has been downloaded and verified in Virtex-4 LX60, SX35 and FX12 devices as well as Virtex-5 LX30T, LX50T, SX35T, SX50T, FX30T and FX70T devices. Due to the similarities in clock buffers, the approach should also be applicable to Virtex-6 FPGAs [14] with very little modification.

Timing analysis of the two clock buffer BIST configurations for all Virtex-5 FPGAs indicates that the maximum BIST clock frequency is more or less device independent, as shown in Figure 10. The exceptions are the FX30T and FX70T devices where the routing from the clock buffers to the LUT buffers, then the LUT buffers to the ORAs must bypass the PowerPC module (illustrated in Fig. 9c). This additional delay in the routing resources reduces the maximum BIST clock frequency by about 14%.

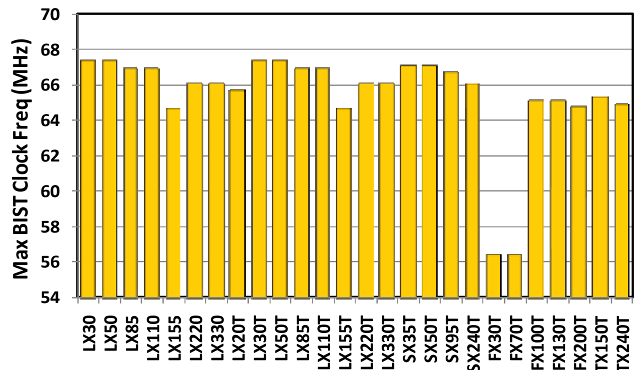

Fig. 10. Timing analysis for clock fuffer BIST configurations in Virtex-5

Since download time is the dominant factor in total test time, taking advantage of multiple frame write features in recent FPGAs can significantly reduce the download time and overall test time. This is illustrated in Figure 11 where the download time obtained with clock buffer BIST compressed configuration files is given for all Virtex-5 FPGAs in terms of the percent of that required for full download. As can be seen, the compressed configuration is at most only 19% of that of a full configuration since the majority of the array is unused during clock buffer BIST. The compressed configuration would apply to the first clock buffer BIST configuration downloaded to the array. During the second BIST configuration, the overall BIST circuitry

remains constant in that the TPGs, ORAs, and all routing remains unchanged such that only the configuration options in the clock buffers themselves need to be modified (as seen in Table IV). Therefore, partial reconfiguration can be used to modify the buffers under test. The partial reconfiguration time for the second BIST configuration is summarized in Figure 12 in terms of the percentage of full and compressed configuration time. As can be see, the partial reconfiguration file is at most only 1.6% that of the compressed configuration file for the first BIST configuration.

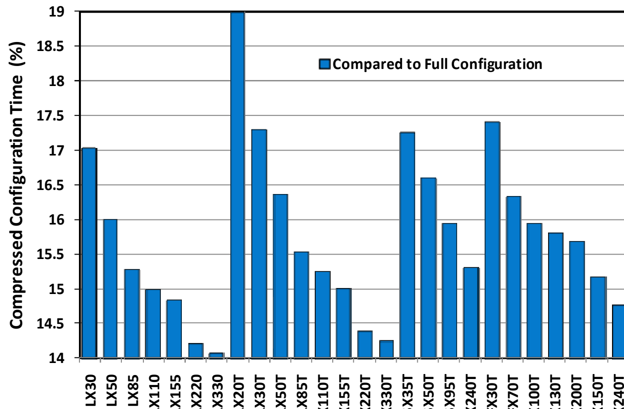

Fig. 11. Download time with first BIST configuration as compressed configuration file

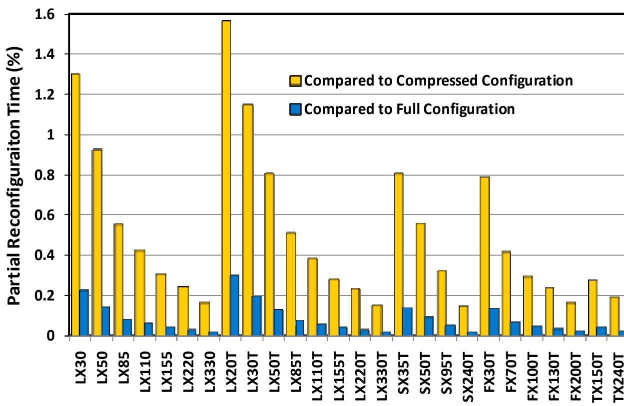

Fig. 12. Download time with second BIST configuration a partial reconfiguration files

5. Summary and Conclusions

We have presented BIST configurations for lesser know but important logic resources in FPGAs including clock buffers in Virtex-4 and Virtex-5 FPGAs and CRC modules in Virtex-5 FGPAs. The BIST approaches we have presented achieve 100% fault coverage in only two configurations. The BIST approaches are applicable to all level of testing, and there is no area or performance penalty during normal system operation. Of particular interest with these logic resources are the new testing challenges they pose compared to more traditional FPGA resources such as CLBs, DSPs, and RAMs. Both of these resources have (in the case of clock buffers), or do not have (in the case of the CRC modules), initialization features for setting values at the outputs of the resources after configuration of the FPGA but before normal system operation. This requires special testing conditions and requires being able to detect faults in the features (in the case of the clock buffer BIST), or being able to avoid false failure indications (in the case of CRC module BIST) during the first clock cycle of the BIST sequence. These issues and requirements help to illustrate the complex nature of test development for modern FPGAs.

Acknowledgements: We acknowledge the assistance of undergraduate students Jarred Beck and Eric Ingram of Auburn Univ. Dept. of Electrical & Computer Engineering.

References

[1] L-T Wang, C. Stroud, and N. Touba, *System-on-Chip Test Architectures*, Morgan Kaufmann, 2008.

[2] S. Toutounchi and A. Lai, "FPGA Test Coverage," *Proc. IEEE International Test Conf.*, pp. 1248-1257, 2003.

[3] B. Dutton and C. Stroud, "Built-In Self-Test of Configurable Logic Blocks in Virtex-5 FPGAs," *Proc. IEEE Southeastern Symp. on System Theory*, pp. 235-240, 2009.

[4] B. Garrison, et. al., "Built-In Self-Test of Embedded Programmable Memory Resources in Virtex-4 FPGAs," *Proc. ISCA International Conf. on Computers and Their Applications*, pp. 63-68, 2009.

[5] M. Pulukuri and C. Stroud, "Built-In Self-Test of Digital Signal Processors in Virtex-4 FPGAs," *Proc. IEEE Southeastern Symp, on System Theory*, pp. 34-38, 2009.

[6] B. Dutton and C. Stroud, "Built-In Self-Test of Programmable Input/Output Tiles in Virtex-5 FPGAs," *Proc. IEEE Southeastern Symp, on System Theory*, pp. 235-239, 2009.

[7] "Virtex-4 FPGA User Guide," UG070 (v2.5), Xilinx Inc., 2008. [2]

[8] "Virtex-5 FPGA User Guide," UG190 (v5.2), Xilinx Inc., 2009. [2]

[9] "Virtex-5 FPGA RocketIO GTP Transceiver User Guide," UG196 (v2.1), Xilinx Inc., 2009. [2]

[10] "Virtex-5 FPGA RocketIO GTX Transceiver User Guide," UG198 (v3.0), Xilinx Inc., 2009. [2]

[11] "Virtex-4 FPGA Configuration User Guide," UG071 (v1.1), Xilinx Inc., 2008. [2]

[12] "Virtex-5 FPGA Configuration User Guide," UG191 (v2.7), Xilinx Inc., 2008. [2]

[13] B. Dutton, et. al., "Embedded Fault and SEU Injection in Virtex-4 and Virtex-5 FPGAs," *Proc. International Conf. on Embedded Systems and Applications*, pp. 183-189, 2009.

[14] "Virtex-6 Clocking Resources User Guide," UG362 (v1.2), Xilinx Inc., 2010. [2]

[2] available at www.xilinx.com