A Non-Intrusive BIST Capability for UART

D. Mahesh Kumar and A.M. Sreeram

***Abstract*---**As the device size is shrinking, device density is increasing. This increases the functional complexity on the chip and also accessing of internal sub-circuits of chip for testing purposes is becoming very difficult, as they are not directly accessible through primary inputs. So, the testing of chip which has become very compulsory is a very time consuming and costly process with increasing cost. The term "functional BIST" describes a test method to control functional modules so that they generate a deterministic test set, which targets structural faults within other parts of the system. Built-In Self-Test (BIST) techniques which are non-intrusive to the circuitry under test are investigated for incorporation in UART. In this paper, the analysis of area overhead and increase in delay for implementing non intrusive BIST technique in UART is carried out. The technique can provide shorter test time compared to an externally applied test and allows the use of low-cost test equipment during all stages of production. Hence a UART with BIST has the objectives of firstly to satisfy specified testability requirements, and secondly to generate the lowest-cost with the highest performance implementation. We have implemented Universal asynchronous receiver transmitter (UART) with Non-intrusive BIST capability using LFSR techniques and compared these techniques for the logic utilization in SPARTAN2E XC2S300-PQ208 FPGA device.

***Keywords*---**BIST, UART, LFSR, MISR, BILBO, FPGA

I. INTRODUCTION

BUILT-IN SELF-TEST (BIST) strategies embed the functions needed for testing a given Unit Under Test (UUT) into the chip itself. These functions consist of a Test Pattern Generator (TPG) and a Test Response Compactor (TRC) at least, and until now, they have been performed by specialized dedicated hardware mainly based on Linear Feedback Shift Register (LFSRs) or cellular automata. A BIST Universal Asynchronous Receive/Transmit (UART) has the objectives of firstly to satisfy specified testability requirements, and secondly to generate the lowest-cost with the highest performance implementation.

This typically requires that additional circuitry and functionality be incorporated into the design of the circuit to facilitate the self-testing feature. This additional functionality must be capable of generating test patterns as well as providing a mechanism to determine if the output responses of the circuit under test (CUT) to the test patterns correspond to that of a fault-free circuit. An alternative to test point insertion is not modifying the UUT but the pattern generators. For this purpose, test methods based on weighted random patterns [1][2][3] and, more recently, based on deterministic test patterns have been developed [4][5][6][7]. The reseeding technique presented in [8] [9] computes initial values of an LFSR so that the output sequence includes pre-computed deterministic test patterns. Recently an innovative BIST technique has been proposed which exploits the system functionalities for test generation and is less intrusive than using test registers.

The present paper intends to exploit the flexibility of the method [10] to investigate the usability of the functional BIST approach for testing UARTs. Actual UARTs include a variety of functional units, library modules (e.g., RX, TX, LFSR, MFSR, etc.), as well as custom blocks. Moreover, these modules usually form a strongly connected network, in which each unit is functionally linked to many other system modules either by bus-oriented or by multiplexer-oriented interconnections.

In the implementation phase, the BIST technique will be incorporated into the UART design before the overall design is synthesized by means of reconfiguring the existing design to match testability requirements. The UART is targeted at broadband modem, base station, cell phone, and PDA designs.

The acceptance of the design for test techniques has been largely due to the possibility of VHDL support to this design style. It is desirable to eventually have available a BIST approach with similarly VHDL support. The high degree of standardization makes it possible to have most testability feature previously added to a design using VHDL [11].

II. BASIC BIST ARCHITECTURE

A representative architecture of the BIST[17] circuitry as it might be incorporated into the CUT is illustrated in the Fig 1. This BIST architecture includes two essential functions as well as two additional functions that are necessary to facilitate execution of the self-testing feature while in the system. The two essential functions include the test pattern generator (TPG) and output response analyzer (ORA). While the TPG produces a sequence of patterns for testing the CUT, the ORA compacts the output responses of the CUT into some type of *Pass/Fail* indication. The other two functions needed for system-level use of the BIST include the test controller (or BIST controller) and the input isolation circuitry. Aside from the normal system I/O pins, the incorporation of BIST may also require additional I/O pins for activating the BIST sequence (the *BIST Start* control signal), reporting the results of the BIST (the *Pass/Fail* indication), and an optional indication (*BIST Done*) that the BIST sequence is complete

D. Mahesh Kumar, Assistant Professor, Department of Electronics, PSG College of Arts and Science, Coimbatore. Email: dmaheshkumar@rediffmail.com

A.M. Sreeram, Student, Applied Electronics, PSG College of Arts and Science, Coimbatore

and that the BIST results are valid and can be read to determine the fault-free/faulty status of the CUT.

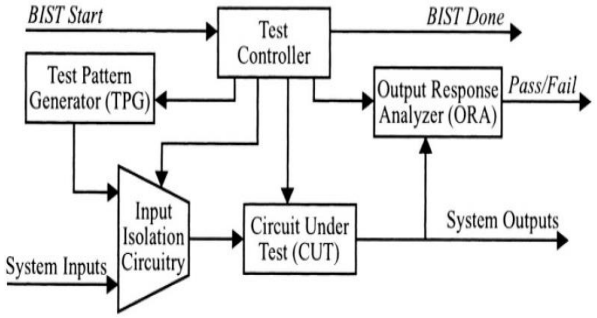

Fig-1: Basic BIST Architecture

III. NON-INTRUSIVE BIST ARCHITECTURES

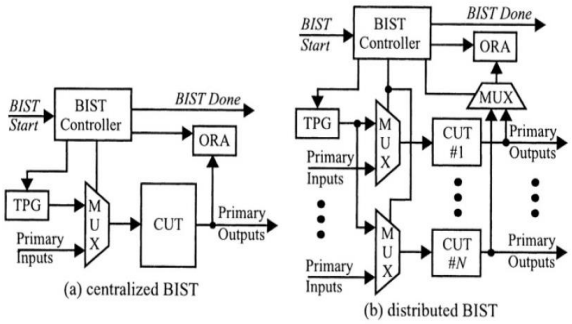

Fig-2: Non-Intrusive BIST Architecture

The basic BIST architecture is a separate, test-per-clock, BIST architecture. Since the TPG and ORA functions are external from the CUT, they can be used to test multiple CUTs. This can significantly reduce the area overhead. Non-intrusive BIST architectures (also referred to as non-invasive BIST) do not impose performance penalties on the CUT like embedded BIST approaches. The only performance penalty is additional set-up time on the primary inputs due to the input isolation multiplexers.

There is additional clock-to-output delay due to the fan-out of the primary outputs to the ORA. The CUT in embedded BIST approaches incurs performance penalties not only from the gate delays of the embedded BIST circuitry, but also from the fact that in VLSI devices, the CUT will be spread out to make room for the embedded BIST circuitry. This means that wire lengths for routing will be longer with more resistance and capacitance to increase the routing delays. In VLSI applications of the non-intrusive BIST approaches, it is possible that the layout of the CUT will not be affected by the separate BIST circuitry and, as a result, will not incur additional performance penalty. Obviously, the chip area will be larger due to the BIST circuitry, which will in turn increase the routing lengths at the primary inputs and outputs, but the maximum operating frequency of the CUT may not be affected. As a result, non-intrusive BIST approaches are good candidates for high speed applications.

IV. BIST PATTERN GENERATION

There are various methods and approaches have been used to generate test patterns during BIST. This can be described in brief below:

(a) **LFSR** - Linear Feedback Shift Register is used to generate pseudorandom test patterns. This normally requires a sequence of one million or more tests pattern in order to achieve high fault coverage. One of the advantages of LFSR is it uses very little hardware and thus is currently the preferred BIST pattern generation method. In this project, LFSR is being chosen as the test pattern generation method.

(b) **Binary Counters** - A binary counter can generate an exhaustive but not randomized test sequences. Drawback of binary counters as the pattern generator is, it requires more hardware than typical LFSR pattern generator.

(c) **Modified Counters** - Modified counters also have been successfully as test-pattern generators. However, they also require long test sequences.

(d) **ROM** - This method stores a good test-pattern set from an ATPG program in a ROM on the chip. However, drawback of this approach is relatively expensive in chip area.

(e) **Cellular Automaton** - In this method, each pattern generator cell has a few logic gates, a flip-flop, and connections only to neighboring gates. The cell is replicated to produce the cellular automaton.

V. UNIVERSAL ASYNCHRONOUS RECEIVE/TRANSMIT (UART)

Serial data is transmitted via its serial port. A serial port is one of the most universal parts of a computer. It is a connector where serial line is attached and connected to peripheral devices such as mouse, modem, printer and even to another computer. In contrast to parallel communication, these peripheral devices communicate using a serial bit stream protocol (where data is sent one bit at a time). The serial port is usually connected to UART, an integrated circuit which handles the conversion between serial and parallel data [12][13].

Fig. 3 shows how the UART receives a byte of parallel data and converts it to a sequence of voltage to represent 0s and 1s on a single wire (serial). To transfer data on a telephone line, the data must be converted from 0s and 1s to audio tones or sounds (the audio tones are sinusoidal shaped signals). This conversion is performed by a peripheral device called a modem (modulator/demodulator). The modem takes the signal on the single wire and converts it to sounds. At the other end, the modem converts the sound back to voltages, and another UART converts the stream of 0s and 1s back to bytes of parallel data.

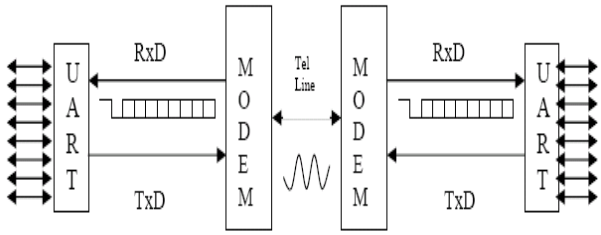

Fig-3: Serial Data Transmission and Receive

VI. LFSR DESIGN

It is easy to figure out that the radix-*N* polynomial requires N D-FFs. The coefficient of each exponent denotes the insertion points of exclusive-OR gates in the shifting path. When the LFSR starts, the LFSR [14] is reset to zero first. Then the seed value is applied sequentially from I0. As the LFSR is operating in test pattern generation mode, the I0 is set to 0.

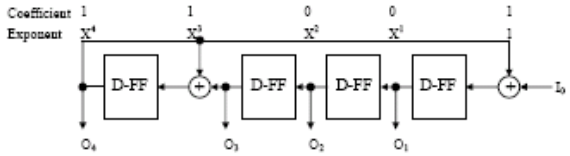

Fig-4: The Division type LFSR with Polynomial X4 + X3 + 1

The bit positions selected for use in the feedback function are called "taps". The list of the taps is known as the "tap sequence". By convention, the output bit of an LFSR that is n bits long is the nth bit; the input bit of an LFSR is bit 1. The state of an LFSR that is n bits long can be any one of 2n different values. The largest state space possible for such an LFSR will be 2n - 1, all possible values except the zero state. All zero is not allow in LFSR as it will always produce 0 in spite of how many clock iteration. Because each state can have only once succeeding state, an LFSR with a maximal length tap sequence will pass through every non-zero state once and only once before repeating a state.

During BIST [16], it is important that the circuit be excited once and only once with a particular pattern. This is due to a given pattern causes an error vector to appear at the faulty circuit outputs, which are read by the BIST response compactor, and repeating the pattern later cause the same error vector to be appear again. Since the response compactor is an XOR-ing system as well, the two erroneous responses from that error vector will cancel and leave the BIST system with only the good machine response. As a result, this causes the testing hardware to accept a faulty circuit as a good circuit. Thus, it is critical to avoid repeating any of the LFSR patterns more than once. Besides, as discussed, initialize the LFSR to all zeros is strictly prohibited as this will hang the LFSR indefinitely in all zero state.

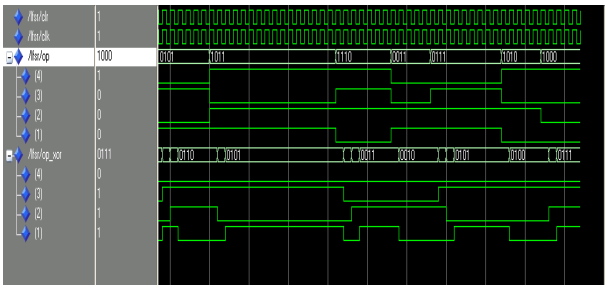

Fig-5: Simulation of LFSR

Different structures of LFSR will generate different sequence of test pattern. It means that if the BIST time is limited, the structure of LFSR will affect the BIST time and Fault Coverage (FC) of Circuit Under Test (CUT).

VII. BUILT – IN LOGIC BLOCK OBSERVERS

Built-In Logic Block Observers (BILBO) is a circuitry that combines the functionality of the D flip-flop, a standard LFSR testing hardware pattern generator (for the circuit portion driven by the BILBO Q outputs), a testing response compacter (for the circuit portion driven by the BILBO D inputs) and a scan chain function. By shifting in an all-zero pattern into the BILBO in serial scan modem the scan chain can be reset to zero.

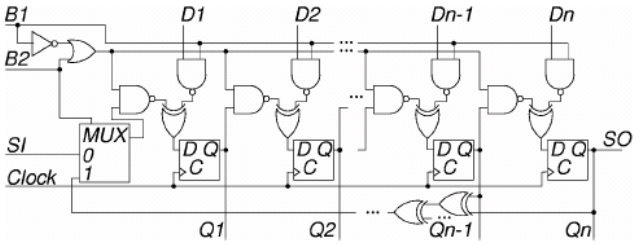

Fig-6: BILBO Circuit

Figure 6 shows the circuit for the BIBLO while Table 2.1 shows the control mode for the BILBO of Figure 6. The BILBO in Figure 6 uses the NAND gate to accelerate the speed over the implementation with AND and OR gates.

TABLE 1:CONTROL MODES FOR BILBO

B1 B2	Operation Mode
00	BILBO Serial SCAN Mode
01	BILBO LFSR Pattern Generator Mode
10	D- flip-flop mode
11	MISR mode

Figure 7 illustrates the effective BILBO hardware in serial scan mode, when B1 and B1 equal to “00”, Figure 8 shows hardware in LFSR mode with B1 and B2 equal to “01” and Figure 9 shows the hardware in D flip-flop mode with B1 and B2 is “10” and Figure 10 shows the hardware in MISR mode

when B1 and B2 is 11. The bold lines show the enabled data path.

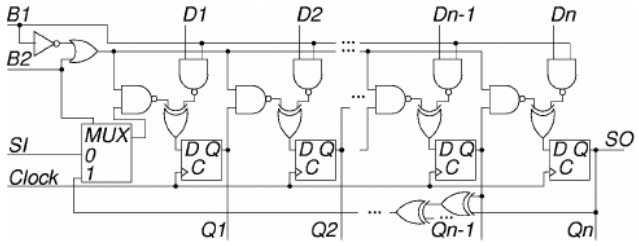

Fig-7: BILBO in Serial scan mode

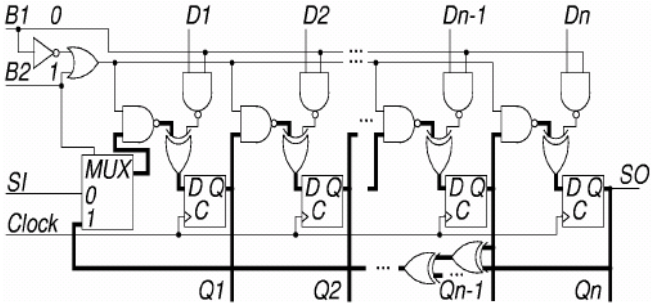

Fig-8: BILBO in LFSR mode

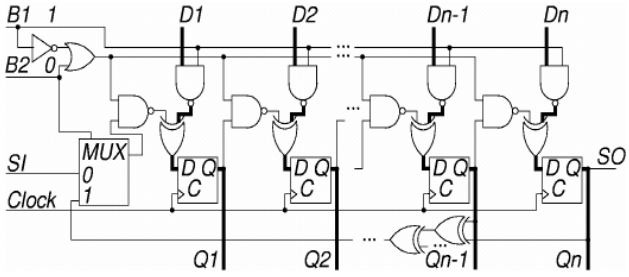

Fig-9: BILBO in normal D flip-flop mode

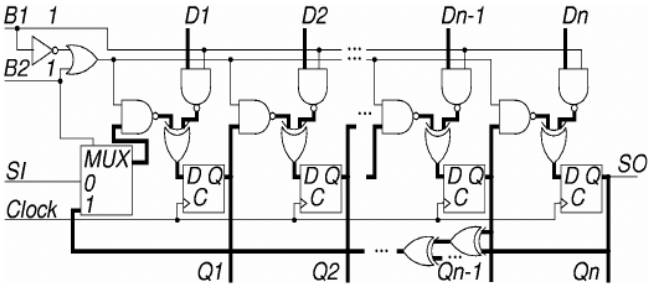

Fig-10: BILBO in MISR mode

VIII. BIST RESPONSE COMPACTION

During BIST, for every test pattern that being generated, the CUT produces a set of output values. In order to ensure the chip is fault free, every output values from the CUT for each test pattern will need to compare with the correct output values obtained from the simulations. This is a tedious and time consuming process. Thus, it is necessary to reduce the enormous of circuit responses to a manageable size that can be either store in the chip or can easily compared with the golden response values. For example, a BIST pattern generator in a chip can produce 1 million test patterns. If the chip has a total of 100 primary output, at the end of the BIST process, it will generate a total of 1 million output values or 1000000 x100 = 100 million bits of output values. With such a huge amount of data, it is very costly and almost impossible to store in the storage or ROM inside a chip. Thus, the circuit response must be compacted.

There are several approaches and method can be used for response compaction, such as transition count response compaction, LFSR for response compaction, Modular LFSR response compaction and multiple input signature register. In this project, multiple input signature register will be used as response compactor.

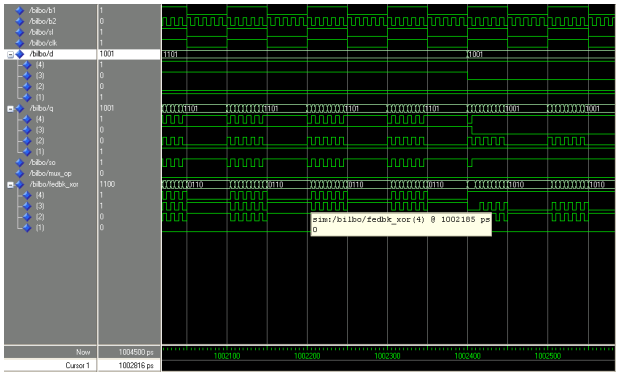

Fig-11: Simulation of CUT

IX. UART WITH BILBO REGISTER AND TESTING

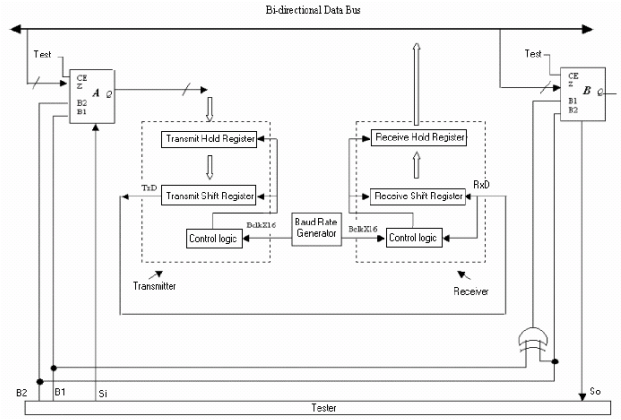

Fig-12: UART with BILBO Register and Testing

BILBO is a scan register that can be modified to serve as a state register, a pattern generator, a signature register, or a shift register. Fig. 4 shows how to apply BILBO registers to test the UART design [15]. In this structure, "Register A" and "Register B" may be configured by mode control ("bilbo_mode") signal to act as either a shift register, a test pattern generator (PRPG), normal application mode function (normal) or a data compressor (MISR). The test starts with the initialization of the BILBO by applying a "seed" to its serial-in (si) pin. The initialization can be obtained by configuring BILBO's operating mode ("bilbo_mode") to "00" (shift register mode). Following the initialization, the bilbo_mode is set to "01" so that "Register A" is configured as LFSR ("bilbo_mode" = "01") and Register B as MISR ("bilbo_mode" = "11").

"Register A" (LFSR) produces an 8-bits pseudo random pattern data in parallel. The parallel data is then fed to the UART's transmitter. The UART converts the pseudo random parallel data to serial data which is then looped back to its receiver to create an internal diagnostic capability. The UART's receiver converts the serial data back to parallel and will be accepted by "Register B" (MISR). A signature will be produced after 255 clock iterations (8 data bits produce 28 = 256 PRPG) and this completes the test. The signature is scanned out from serial output (so) pin by configuring bilbo_mode to "00". Following the scan, it is compared with the correct signature achieved from the simulation of the entire self-test sequence approach in a tester. If the signature produced by MISR is similar to the correct signature, it can be concluded that the UART is working properly.

To start the test, the BILBO_MODE (before 3us) is set to "10B" to operate in normal mode. From 3us to 13us, the BILBO_MODE is set to "00B" and acts as a shift register. BILBO shift register then shifts the value of S_DATA_IN from right to left (LSB to MSB) by using a serial in (si) pin (Fig. 8). The shifting is conducted to initialize the LFSR and MISR with a 'seed' data. S_DATA_IN then pushes the data to S_QOUT after 3-clock delay. Table 1 shows S_QOUT and S_DATA_IN in binary. As can be observed at the end of Table 1, the LFSR is initialized to "00001111" and MISR is initialized to "00101110".

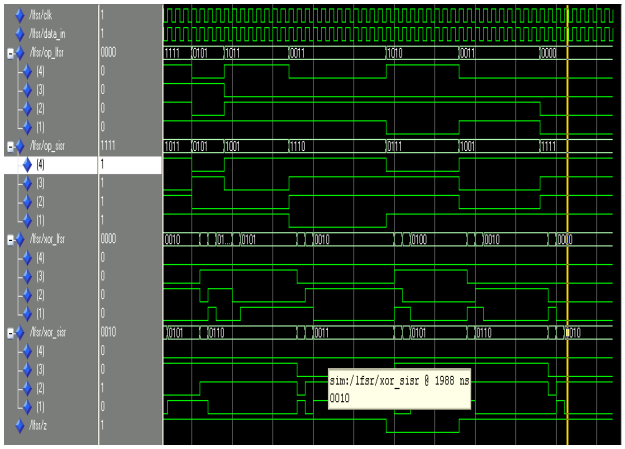

Fig-13: Simulation of BILBO

TABLE 2: S_QOUT AND S_DATA_IN IN BINARY

S_QOUT (BINARY)	S_DATA_IN (BINARY)
00000001	00101110
00000010	01011100
00000100	10111000
00001000	01110000
00010001	11100000
00100010	11000000
01000101	10000001
10001011	00000011
00010111	00000111
00101110	00001111

Fig. 13 shows that the BILBO_MODE is set to "01B". In this mode, the BILBO at the transmitter is configured as an LFSR and BILBO at the receiver is configured as an MISR. During BILBO_MODE = "01B" DATA[7:0] can be ignored.

S_DATA_IN acts as an LFSR that produces parallel pseudo random pattern (PRPG) signals to the UART's transmitter. These parallel signals are then converted to serial data in a communication line and will be looped back to the receiver. The receiver converts the data back to parallel and forwards it to S_RXDATA. S_QOUT (MISR) compresses all the received pseudo random parallel data (S_RXDATA) into one signature. The produced signature is then compared with the correct signature.

S_DATA_IN (PRPG) is achieved by XOR-ing bit 1, 2, 3 and 7 (MSB…LSB, b7…b0) and the XORed result is placed to bit 0 (LSB). The other remaining bits (b6…b0) are shifted to the left. The S_QOUT (MISR) is achieved as S_DATA_IN except that the produced data (PRPG) is then XORed with S_RXDATA.

The process of feeding the transmitter with pseudo random data and compressing it with MISR is complete after 255 clocks iteration. The final result of MISR (S_QOUT) in Fig. 8 can be observed as "01010001B". However, S_QOUT (MISR) is the internal data of the designed UART. Therefore, there should be a method to send out the signature without sacrificing extra observance output pins. The signature is shifted out at serial data out (so) output's pin. To shift out the signature, BILBO_MODE is set back to "00B" and it acts as a shift register. In Fig. 13, 'bilboen' signal enables the signature to be transmitted as a serial out data (so). As can be observed, 'so' is transmitted as the following sequence: 1 low bit, 1 high, 1 low, 1 high, 3 low and 1 high. The result is the serial data of "01010001B", which is equal to the value of MISR at S_QOUT.

The test setup for MISR is almost the same as the PRPG except that the operation mode needs to be set to MISR ("PC_D2 & PC_D1" = "11") after initialized with "seed" value (using "si" and shift register). For this test, the test data (z<7, 6, 5,…..1, 0>) are set to "00000111" using an external circuit and switch (or use XSTEND board provided by Xess

Corp). These data act as the data output of the circuit under test. The MISR outputs are then observed at outputs q<7, 6, 5, ….. 1, 0> using mixed signal oscilloscope.

Since the functional Non-intrusive BIST approach is an at-speed test, reducing the number of dummy patterns and reducing the overall test length compared to an external scan based test was not a target. Allowing a larger number of dummy patterns and increasing the test length may even improve the defect coverage, and reduce the memory requirements for the seeds while the test application time will still be shorter than the time for external testing.

X. CONCLUSIONS

The present paper works inside the context of the functional BIST strategy and investigates different functional units as possible deterministic test generators. The functional BIST approach is less intrusive than traditional BIST techniques, since no test points and no additional registers are introduced into the modules to be tested. The efficiency of all the functional units investigated so far is as least as high as efficiency of the classical LFSR reseeding technique, and in many cases it is higher. The simulated waveforms presented in this paper have proven the reliability of the VHDL implementation to describe the characteristics and the architecture of the designed UART with embedded Non-intrusive BIST. The simulated waveforms also have shown the observer how long the test result can be achieved by using the Non-intrusive BIST technique.

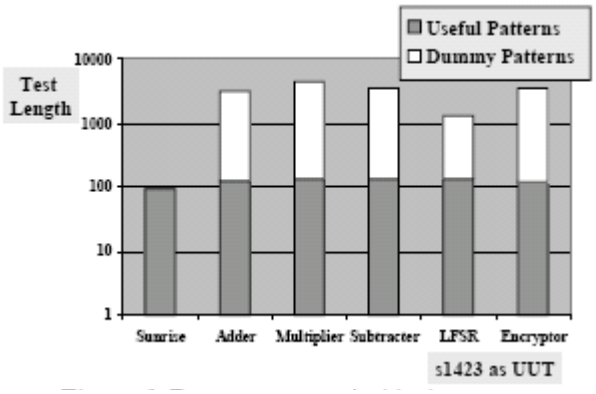

Fig-14: Dummy Patterns inside the Test Set

From the design summary, it can be observed that the number of Configurable Logic Block (CLB) used after the implementation of Non-intrusive BIST technique is increased from 81% to 96%. The difference of 15% of the total CLBs area overhead results in 1.803 ns (i.e. 13.626 ns - 11.823 ns) increment of the maximum net delay. This shows that the UART with embedded Non-intrusive BIST design may not work well if its host clock frequency exceeded the maximum frequency of 31.98 MHz. The reason why the design may not cater for high-speed clock is due to the possibility of a real time delay, which may be caused by temperature or the delay within the FPGA design itself. The faulty data captured may lead to errors at the output pins.

With the implementation of Non-intrusive BIST, expensive tester requirements and testing procedures starting from circuit or logic level to field level testing are minimized. The LFSR replaces the function of the external tester features such as a test pattern generator by automatically generating pseudo random patterns to give 100% fault coverage to the UART module. The MISR acts as a compression tool, compressing the output result when automatic pseudo random pattern is fed to the UART. The shift register minimized the input/output overhead by shifting the parallel signature produced by MISR into serial signature. The reduction of the test cost will lead to the reduction of overall production cost.

REFERENCES

[1] H.-J. Wunderlich, Protest: A Tool for Probabilistic Testability Analysis, IEEE/ACM 22nd Design Automation Conference, Las Vegas, 1985, pp. 204-211.

[2] J. A. Waicukauski, E. Lindbloom et al., WRP: A Method for Generating Weighted Random Patterns, IBM Journal of Research and Development, Vol. 33, No. 2, March 1989, pp. 149-161.

[3] F. Brglez et al., Hardware-Based Weighted Random Pattern Generation for Boundary-Scan, IEEE International Test Conference, Washington, DC, 1989, pp. 264-274.

[4] S. Hellebrand, S. Tarnick, J. Rajski, B. Courtois, Generation of Vector Patterns Through Reseeding of Multiple-Polynomial Linear Feedback Shift Registers, IEEE International Test Conference, Baltimore, MD, September 1992, pp. 120-129.

[5] N. A. Touba and E. J. McCluskey, Altering a Pseudo-Random Bit Sequence for Scan-Based BIST, IEEE International Test Conference, Washington, DC, 1996, pp. 167-175.

[6] H.J Wunderlich, G. Kiefer, Bit-flipping BIST, IEEE International Conference on Computer-Aided Design, 1996, pp. 337-343.

[7] S. Chiusano, F. Corno, P. Prinetto, M. Sonza Reorda, Cellular Automata for Deterministic Sequential Test Pattern Generation, IEEE VLSI Test Symposium, 1997, pp.60-65.
[8] S. Hellebrand, B. Reeb, S. Tarnick, H.-J. Wunderlich, Pattern Generation for a Deterministic BIST Scheme, IEEE/ACM International Conference on CAD-95, San Jose, CA, November 1995, pp. 88-94.
[9] S. Cataldo, S. Chiusano, P. Prinetto, H-J. Wunderlich, Optimal Hardware Pattern Generation for Functional BIST, IEEE Design Automation and Test in Europe (DATE), 2000, pp.292-297.
[10] A. P. Stroele, and H. J. Wunderlich, "Hardware-Optimal Test Register Insertion", in IEEE Transactions On Computer-Aided Design of Integrated Circuits and Systems, June 1998, Vol. 17, No. 6, pp. 531-539.
[11] Z. Navabi, "VHDL Analysis and Modeling of Digital Systems", McGraw-Hill Inc., 1991.
[12] M. S. Michael, "A Comparison of the INS8250, NS16450 and NS16550AF Series of UARTs", National Semiconductor Application Note 493, April 1989.
[13] "PC16550D Universal Asynchronous Receiver/Transmitter with FIFOs", National Semiconductor Application Note, June 1995.
[14] Shikha Kakar, Balwinder Singh, and Arun Khosla, Implementation of BIST Capability using LFSR Techniques in UART, International Journal of Recent Trends in Engineering, Vol 1, No. 3, 2009, pp.301-304.
[15] Mohd Yamani Idna Idris, Mashkuri Yaacob, Zaidi Razak . A VHDL IMPLEMENTATION OF UART DESIGN WITH BIST CAPABILITY Faculty of Computer Science and Information Technology. University of Malaya.
[16] Chakrabarty, et. al, "Deterministic Built-in Test Pattern Generation for High-Performance Circuits Using Twisted-Ring Counters," IEEE Trans. of VLSI Systems, Vol. 8, No.5, pp. 633-636, Oct. 2000.
[17] Charles E. Stroud "A Designer's Guide to Built-in Self-Test" Kluwer Academic Publishers 2002.
[18] M. Ibrahim Abubakar, "A Built in Self Testable Bit-Slice Processor", Faculty of Computer Science & Information Technology, University of Malaya, May 1995.

BIOGRAPHIES

D. Mahesh Kumar obtained his B.Sc., Electronics and M.Sc., Applied Electronics from PSG College of Arts and Science, Coimbatore in 1996 and 1998 and also M.Phil., in Electronics from PSG College of Arts and Science, Coimbatore in 2006. He has been working in the teaching field for about 11 years. He has published one book on the topic "Textbook of Operational Amplifier and Linear Integrated Circuits" by Macmillan India Ltd., New Delhi.

A. M. Sreeram obtained his B. Sc., Electronics from VLB Janaki Ammal College of Arts and Science, Coimbatore in 2010. At present he is doing M.Sc., Applied Electronics in PSG College of Arts and Science, Coimbatore.