FOREX Publication
Open Access | Rapid and quality publishing

International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 139-145 | e-ISSN: 2347-470X

Enhancing FPGA Testing Efficiency: A PRBS-Based Approach for DSP Slices and Multipliers

Jugal Kishore Bhandari[1], **Yogesh Kumar Verma**[2*] **and S.K Hima Bindhu**[3]

[1,2] School of Electronics and Electrical Engineering, Lovely Professional University,
Jalandhar, Punjab, 144411, India; yogesh.25263@lpu.co.in[1]*, bhandari.jugal@gmail.com*[2]
[3]Department of Electronics and Communication Engineering, Marri Laxman Reddy Institute of Technology and Management,
Hyderabad, 500043, India; bindukrupal@mlritm.ac.in

*****Correspondence:** Yogesh Kumar Verma ; yogesh.25263@lpu.co.in

ABSTRACT- The multiplication operations are pivotal in (Application-Specific Integrated Circuits) ASICs and Digital Signal Processors (DSPs). The integration of Field-Programmable Gate Arrays (FPGAs) into modern embedded systems, efficient Built-in Self-Tests (BISTs), particularly for complex components like DSP slices, is essential. This paper evaluates Pseudo Random Binary Sequence (PRBS) generators and checkers as BIST tools for high-speed data transfers in FPGAs. The design achieves minimal errors and remarkable efficiency with less than 4% logic utilization within available Look-Up Tables (LUTs). The testing of embedded multipliers in modern FPGAs is analyzed, shedding light on their performance. The analysis includes Built-in Self-Test (BIST), PRBS generator, PRBS checker, and Bit Error Rate (BER), providing insights into FPGA-based testing. This analysis assesses PRBS tools for high-speed FPGA data transfers. A hybrid multiplier design, featuring BIST and PRBS capabilities, notably reduces DSP slice utilization from 16% to 5%. This liberated FPGA resource enhances operational capabilities. The runtime PRBS data control at the block level design exemplifies adaptability in FPGA testing. The findings underscore PRBS-based BIST potential in FPGA testing. The hybrid multiplier not only optimizes FPGA resources but also aligns with dynamic digital system requirements. This research aids FPGA designers and engineers in advanced testing strategies for evolving embedded systems.

Keywords: BIST, PRBS Generator Checker, FPGA, DSP Slices.

ARTICLE INFORMATION

Author(s): Yogesh Kumar Verma, Jugal Kishore Bhandari and S.K Hima Bindhu;

Received: 06/11/2023; **Accepted**: 30/01/2024; **Published**: 26/02/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 0611-05;
Citation: 10.37391/IJEER.120120
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120120.html

crossref member
CROSSREF.ORG
THE CITATION LINKING BACKBONE

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

The Built-In Self-Test (BIST) techniques are essential in the field of digital circuit design to ensure the reliable and efficient operation of complex components like multipliers. Multipliers play a pivotal role in numerous applications, including signal processing, arithmetic calculations, and data compression. It's crucial to verify their correctness and robustness under various conditions. One effective approach for testing multipliers is to employ a Pseudo-Random Binary Sequence (PRBS) generator and checker-based BIST methodology. Multipliers are fundamental components used for performing arithmetic operations in digital systems. They calculate products of binary numbers, which are essential in various computational tasks. Ensuring the correctness of multipliers is vital to maintain the integrity of digital systems. BIST is a methodology that enables digital circuits to test and verify their functionality autonomously without external test equipment. BIST techniques are designed to enhance the reliability and ease of testing complex digital circuits. PRBS is a deterministic sequence of binary values that exhibits properties similar to random sequences. PRBS generators produce sequences of 1s and 0s that appear random but are generated based on a specific algorithm. These sequences are commonly used in testing and data communication to ensure robustness and thorough coverage. A PRBS generator is used to produce a pseudo-random binary sequence. This sequence is fed as one of the inputs to the multiplier under the test. The other input(s) may consist of fixed values or additional PRBS sequences to cover various input patterns. Multiplier under Test is the digital multiplier that requires testing. It could be a standalone multiplier or part of a more extensive digital system. The PRBS checker receives the output from the multiplier under test. It compares this output with the expected PRBS sequence that should result from the given inputs. If the output matches the expected sequence, it indicates correct operation. Any discrepancies suggest a fault or error in the multiplier. PRBS sequences offer comprehensive coverage of input patterns, making it effective in detecting various faults. PRBS sequences exhibit random characteristics, enabling the detection of faults that might be missed by deterministic test vectors. The BIST technique can be automated, reducing the need for external test equipment and simplifying the testing process. Developing robust PRBS generators and checkers is crucial to the success of this BIST approach. Ensuring proper synchronization between the PRBS generator and checker can be challenging. The PRBS generator and checker-based BIST approach provides an efficient and reliable method for testing multipliers in digital circuits. It leverages the power of PRBS sequences to thoroughly test the multiplier's functionality and detect potential faults, ultimately contributing to the overall reliability and performance of digital systems [1]-[15].

2. BLOCK DIAGRAM REPRESENTATION

Figure.1 consists of PRBS generator, DUT, and PRBS checker which are the major components of BIST. A PRBS (Pseudo-Random Binary Sequence) generator is a crucial component in Built-In Self-Test (BIST) systems used for testing digital circuits. PRBS sequences are deterministic binary sequences that appear random, making them valuable for testing a circuit's functionality and fault detection. In BIST, a PRBS generator generates these sequences to stimulate the circuit under test. The DUT is the target of the testing process, and the BIST system is designed to verify the functionality. The PRBS checker is responsible for assessing the quality and correctness of the received data stream. It compares the incoming data against a reference PRBS sequence to detect errors, assess signal integrity, and determine the link's performance.

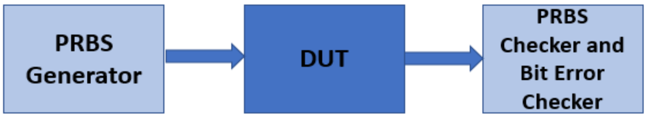


Figure. 1. Major components of BIST

Figure 2 comprises of the "judge" component as a critical part of the BIST process as it determines the effectiveness of the self-test and provides information about the condition of the tested component or system. It plays a key role in identifying and diagnosing faults, which is essential for maintaining the reliability and functionality of electronic systems. *Figure 2* also comprises a test vector generator as a critical component used in digital circuit design and semiconductor manufacturing. Its primary function is to generate a set of test patterns or vectors that are applied to the circuit under test to check for faults, verify functionality, and ensure the circuit's correctness. Another major component is the "Design under Test" (DUT) refers to the specific digital circuit or component that is being tested. The Data compression block in Built-In Self-Test (BIST) systems is a technique used to reduce the volume of test data that needs to be generated, stored, and transmitted during the testing of digital circuits. Compressing test data can help improve testing efficiency, reduce memory requirements, and enable faster testing, especially in high-complexity circuits. The "compare" and "analyze" are two distinct steps in the testing process that help identify and diagnose faults or defects within the tested electronic component or system. In the context of Built-In Self-Test (BIST) and digital circuits, a "ROM block" typically refers to a Read-Only Memory block. A ROM is a digital circuit component that stores data in a non-volatile manner, meaning the data remains intact even when the power is turned off. ROMs are used for various purposes in digital systems, including data storage, code storage, and as lookup tables for digital functions. The control block" is a critical component responsible for managing and coordinating various aspects of the BIST process. The control block plays a pivotal role in the initiation, execution, and control of tests on the circuit under evaluation.

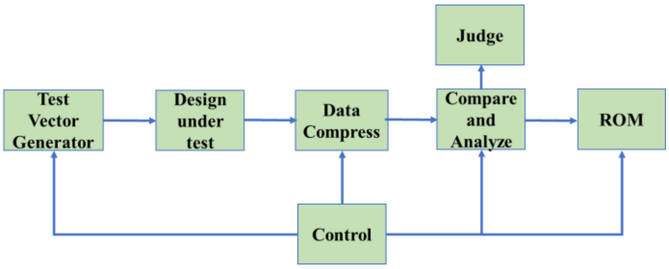


Figure. 2. Conventional Structure of BIST

Figure 3 comprises of a PRBS (Pseudorandom Binary Sequence) generator-checker pair is commonly used in SERDES (Serializer/Deserializer) transceivers for testing and validating the functionality and performance of high-speed serial data communication links.

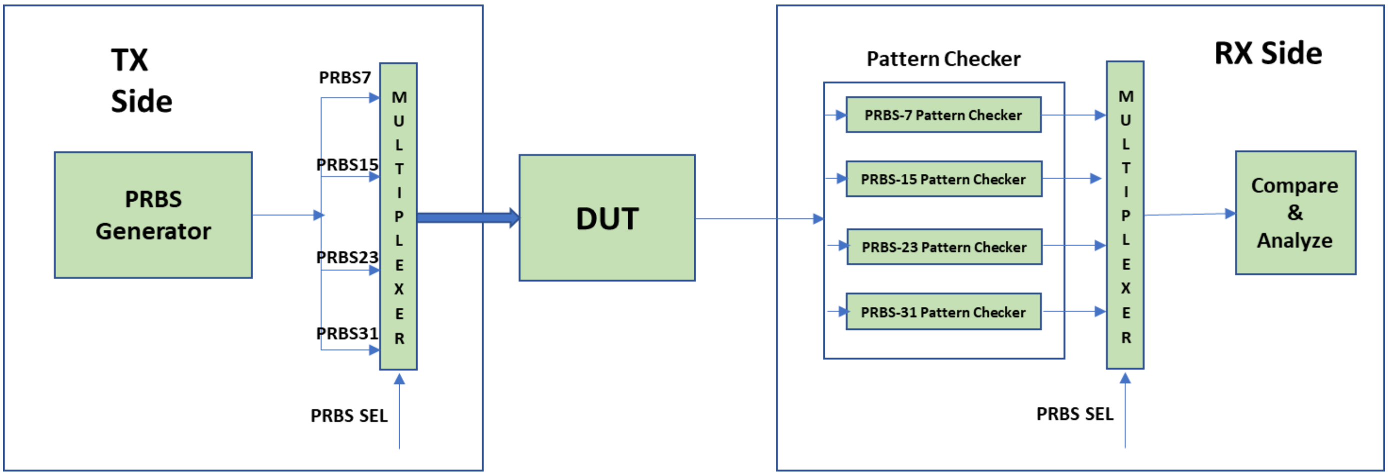


Figure. 3. PRBS (Generator -Checker) for SERDES transceivers

Figure 4 comprises of GTH transceivers, developed by Xilinx, are part of their Ultrascale FPGA and Zynq Ultrascale+ MPSoC families. These transceivers are designed to provide high-speed serial communication capabilities for applications such as high-speed data transfer and interface protocols like PCIe and Ethernet. GTH transceivers offer various modes of operation, including Near-end Loopback (NEL) and Far-end Loopback (FEL) modes, which are used for testing and debugging purposes.

The integration of GTH transceivers, a product of Xilinx embedded within the Ultrascale FPGA and Zynq Ultrascale+ MPSoC families, heralds a new era in high-speed serial communication capabilities. Designed to cater to the demands of modern applications such as high-speed data transfer and interfacing with protocols like PCIe and Ethernet, these transceivers offer a versatile set of functionalities. Among them, the inclusion of runtime PRBS data control proves to be a pivotal feature. This dynamic control mechanism allows for continuous monitoring during operation, facilitating error detection and correction to ensure the robustness of data transfer. In Near-end Loopback (NEL) mode, the transceiver's ability to send PRBS patterns to its own receiver enables localized testing and self-diagnosis, while Far-end Loopback (FEL) mode extends testing capabilities to the entire communication link, including external components. The runtime PRBS data control emerges as a crucial tool for performance verification, remote testing, and dynamic debugging, empowering engineers to optimize the functionality of GTH transceivers in diverse scenarios.

In practical terms, runtime PRBS data control for GTH transceivers transforms the testing landscape for high-speed communication systems. In Near-end Loopback (NEL) mode, the transceiver's self-contained testing capabilities streamline the process of identifying and diagnosing issues within its own components. This localized testing proves invaluable for swift debugging without external dependencies. Conversely, in Far-end Loopback (FEL) mode, the capability to send PRBS patterns to a remote transceiver facilitates comprehensive end-to-end testing, offering insights into the performance of the entire communication link, including external elements like cables. The dynamic nature of the PRBS data control empowers engineers to adapt testing scenarios in real-time, providing a flexible approach to debugging and troubleshooting. Consequently, the runtime PRBS data control emerges as a linchpin for ensuring the reliability, performance, and adaptability of GTH transceivers in the ever-evolving landscape of high-speed communication applications.

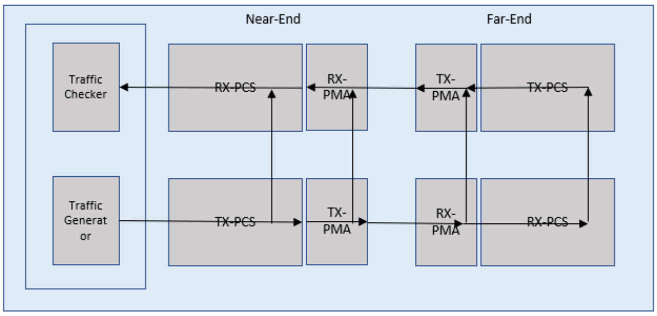
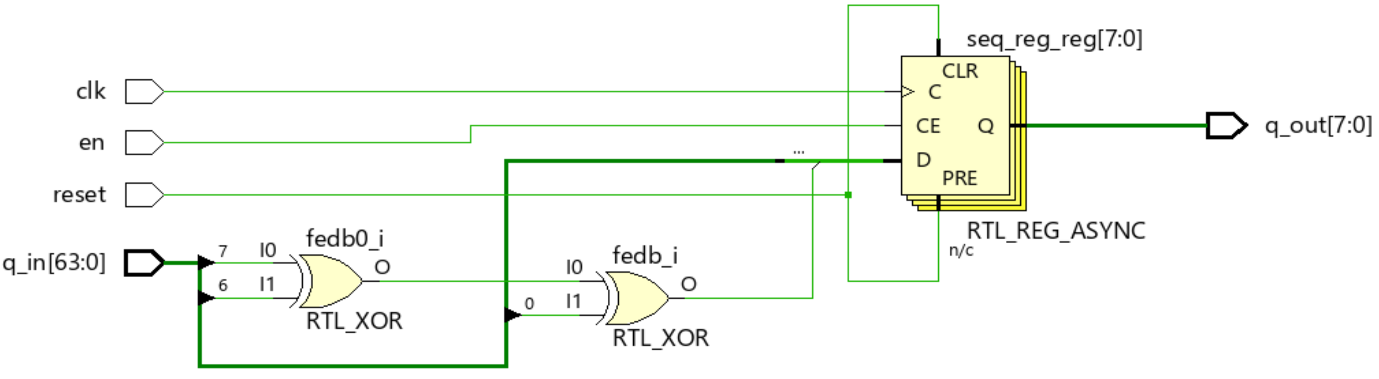


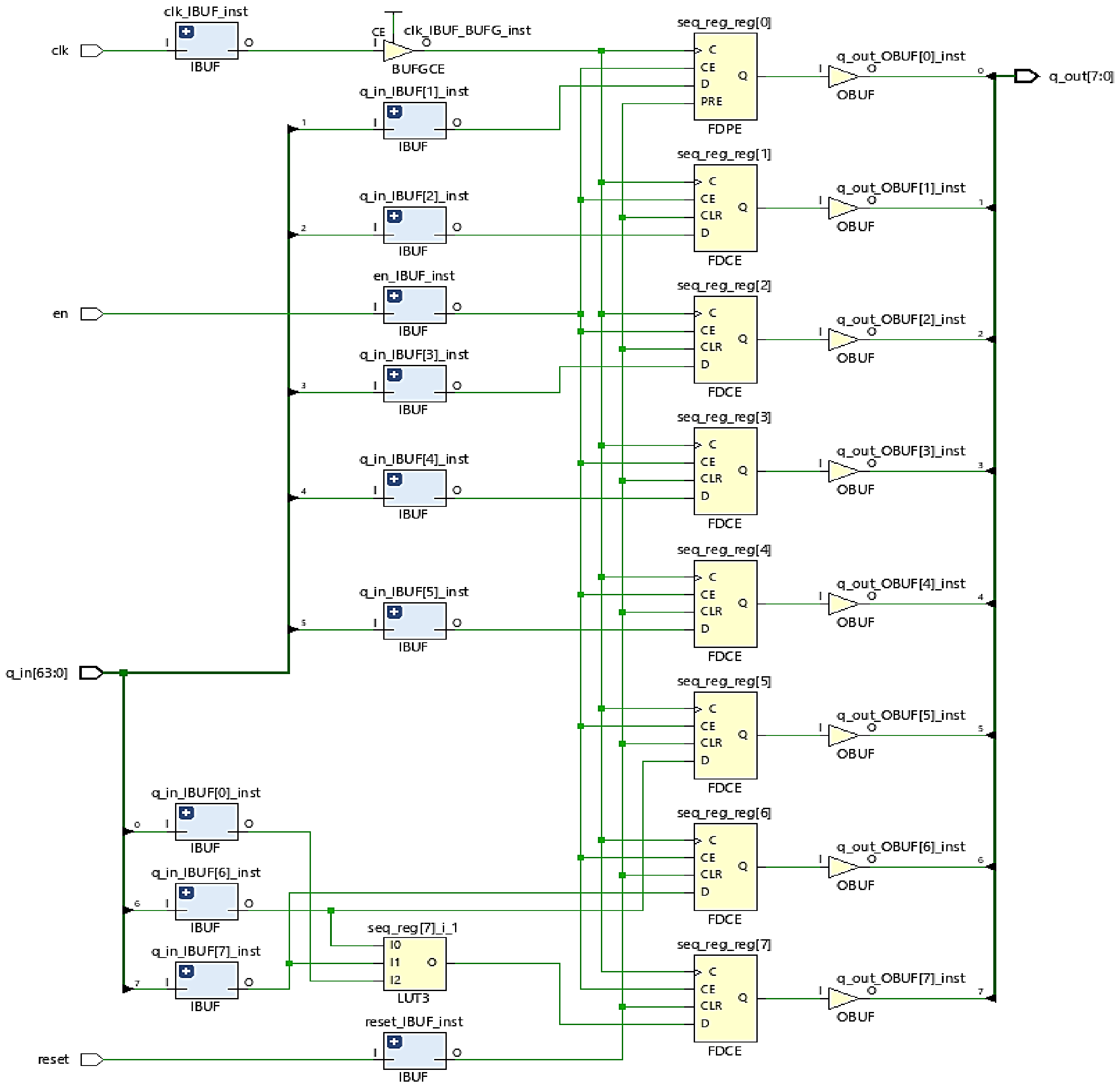
Figure. 4. GTH transceivers

Figure 5 represents a series-parallel PRBS-7 (Pseudo-Random Binary Sequence) generator is a combination of two types of shift registers: a series shift registers and a parallel shift register. *Fig. 6* represents the proposed PRBS checker consisting of a PRBS generator, DUT, clock data recovery, error detection unit, comparator, BER, synchronization detection block, reference PRB sequence, and control logic unit.

Error detection unit includes the PRBS generator, which generates the next seed and compares with the in-coming seed. Depending on the comparison result it counts the error in data. It also counts the errors in Bits. A hybrid multiplier is used in the design to do multiplication for high speed received data. The proposed hybrid multiplier introduces a novel design concept by combining two distinct types of multipliers—Wallace and Vedic multipliers. This innovative approach aims to address the evolving requirements of modern computing systems, emphasizing the need for low-power consumption, reduced area, and minimal delay in multiplier operations [17]. To further enhance the efficiency of the hybrid multiplier, a carry look ahead adder is employed. This addition aims to reduce carry propagation delay, a critical factor in multiplier performance. This PRBS provides flexibility to select different pattern length using a multiplexer in runtime. Along with the PRBS the multiplexer is also connected to user data, whereas the inbuilt PRBS option doesn't support this feature.



(a)



(b)

Figure. 5. (a), (b) PRBS-7 generator

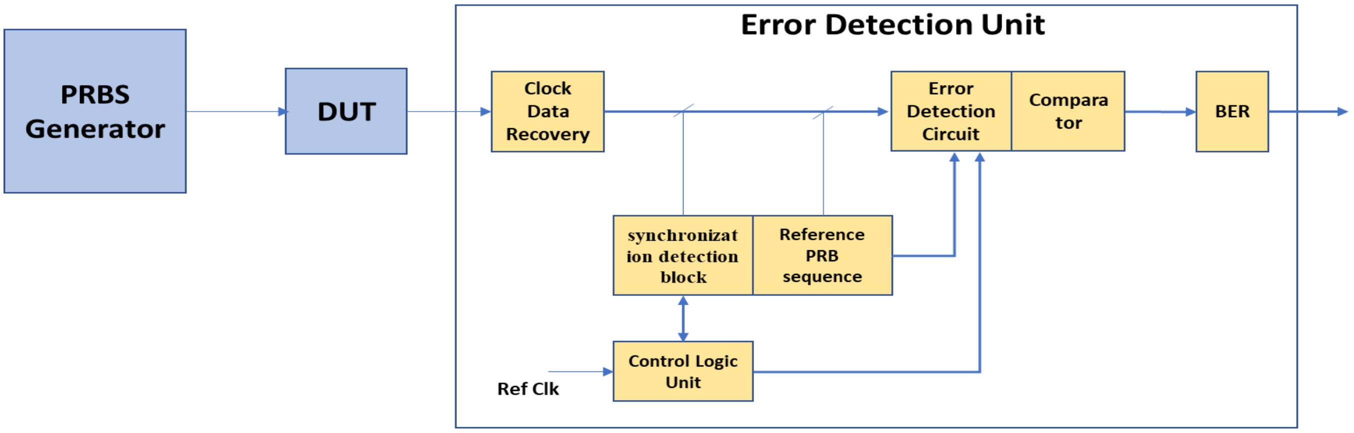


Figure. 6. Proposed PRBS checker

3. RESULTS AND DISCUSSIONS

Table 1 represents the resource utilization, revealing intriguing insights. The Proposed Design utilized more lookup tables (LUTs) and flip-flops, indicating potentially more complex logic. However, the Reference Design made more extensive use of Block RAMs, while both designs equally utilized Gigabit Transceivers (GT). The Proposed Design boasted slightly lower latency and integrated customized PRBS, indicating a versatile approach. The figures referenced in the information likely provide visual representations of the experimental setup and results, adding valuable context to this comparison.

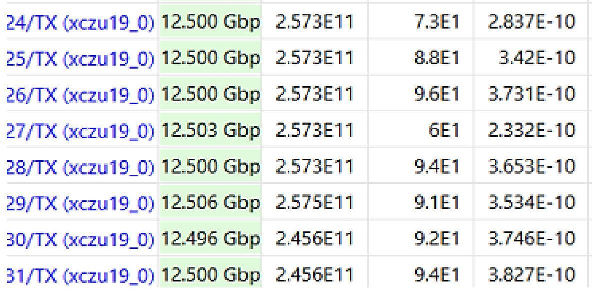
Comparison table of Standard Multiplier used in FPGA with proposed Hybrid multiplier.

- The multiplier also has BIST and PRBS inbuilt.
- The results show advantage in terms of DSP slice utilization from 16% to 5%, providing the DSP blocks available for other application/operation.

Figs. 7(a) and *(b)* represent the bits applied in order of 1011, errors to the order of 10-10 for proposed PRBS with hybrid multiplier and pf the order 10-2 for standard multiplier with inbuilt PRBS. The variation clearly shows the advantage of proposed PRBS-Multiplier in terms of Bit Error Rate.

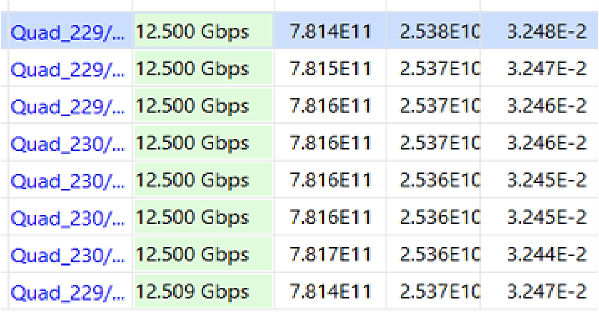
Table 1: Comparison of resource utilization

Resource	Availability	Standard Multiplier [7]	Hybrid Multiplier
		Utilization	
LUTs	522720	17555	18555
FFs	1045440	44007	54000
Block RAMs	984	84.5	78
GT	100%	25%	25%
Latency	-	40 ns (Min)	35 ns (Min)
BER applied	10E-1 to 10E-9	10E-1 to 10E-9	10E-1 to 10E-9
PRBS	Hard PRBS	Hard PRBS	Customized PRBS added in Design
F7 Mux	261360	389	754
F8 Mux	130680	128	250
DSP Slices	1968	324	100
LUT RAM	161280	2216	2100



24/TX (xczu19_0)	12.500 Gbp	2.573E11	7.3E1	2.837E-10
25/TX (xczu19_0)	12.500 Gbp	2.573E11	8.8E1	3.42E-10
26/TX (xczu19_0)	12.500 Gbp	2.573E11	9.6E1	3.731E-10
27/TX (xczu19_0)	12.503 Gbp	2.573E11	6E1	2.332E-10
28/TX (xczu19_0)	12.500 Gbp	2.573E11	9.4E1	3.653E-10
29/TX (xczu19_0)	12.506 Gbp	2.575E11	9.1E1	3.534E-10
30/TX (xczu19_0)	12.496 Gbp	2.456E11	9.2E1	3.746E-10
31/TX (xczu19_0)	12.500 Gbp	2.456E11	9.4E1	3.827E-10

Figure. 7(a) Serial I/O Link demonstrating Data rate, number of bits applied, Errors and BER for Proposed PRBS



Quad_229/...	12.500 Gbps	7.814E11	2.538E1C	3.248E-2
Quad_229/...	12.500 Gbps	7.815E11	2.537E1C	3.247E-2
Quad_229/...	12.500 Gbps	7.816E11	2.537E1C	3.246E-2
Quad_230/...	12.500 Gbps	7.816E11	2.537E1C	3.246E-2
Quad_230/...	12.500 Gbps	7.816E11	2.536E1C	3.245E-2
Quad_230/...	12.500 Gbps	7.816E11	2.536E1C	3.245E-2
Quad_230/...	12.500 Gbps	7.817E11	2.536E1C	3.244E-2
Quad_229/...	12.509 Gbps	7.814E11	2.537E1C	3.247E-2

Figure. 7(b) Serial I/O Link demonstrating Data rate, number of bits applied, Errors and BER for Inbuilt PRBS []

Figure. 8 represents the design for testing and verifying the integrity of data. This technique is widely used in various applications, including telecommunications, networking, and high-speed data transfer. *Figure. 9* represents the major components of PRBS based BIST. *Figures. 5, 6, 8 and 9* provides the Block level design and RTL view of proposed logic. The design is well optimized in block level as well as in RTL level using Verilog and VHDL scripts (top_0 and prbs_generator_0 block in *figure 8*) which helps in better results.

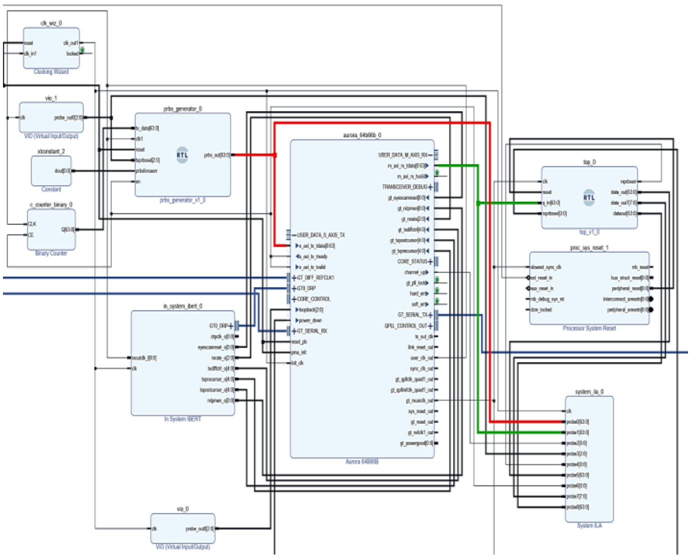


Figure. 8. Hybrid multiplier the PRBS is added as IP in block level design, which gives run time feature to control the PRBS type data to be applied.

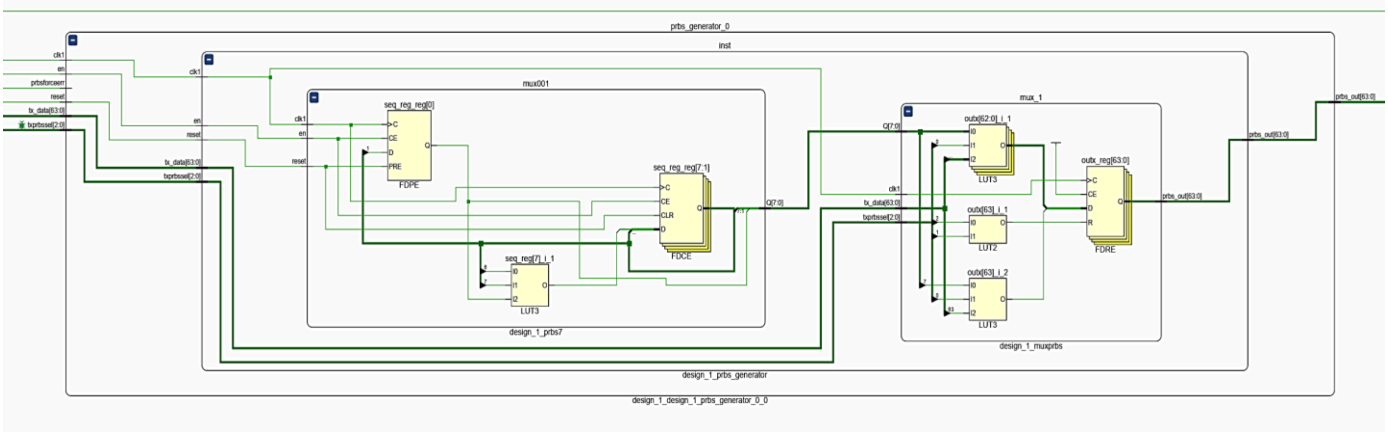
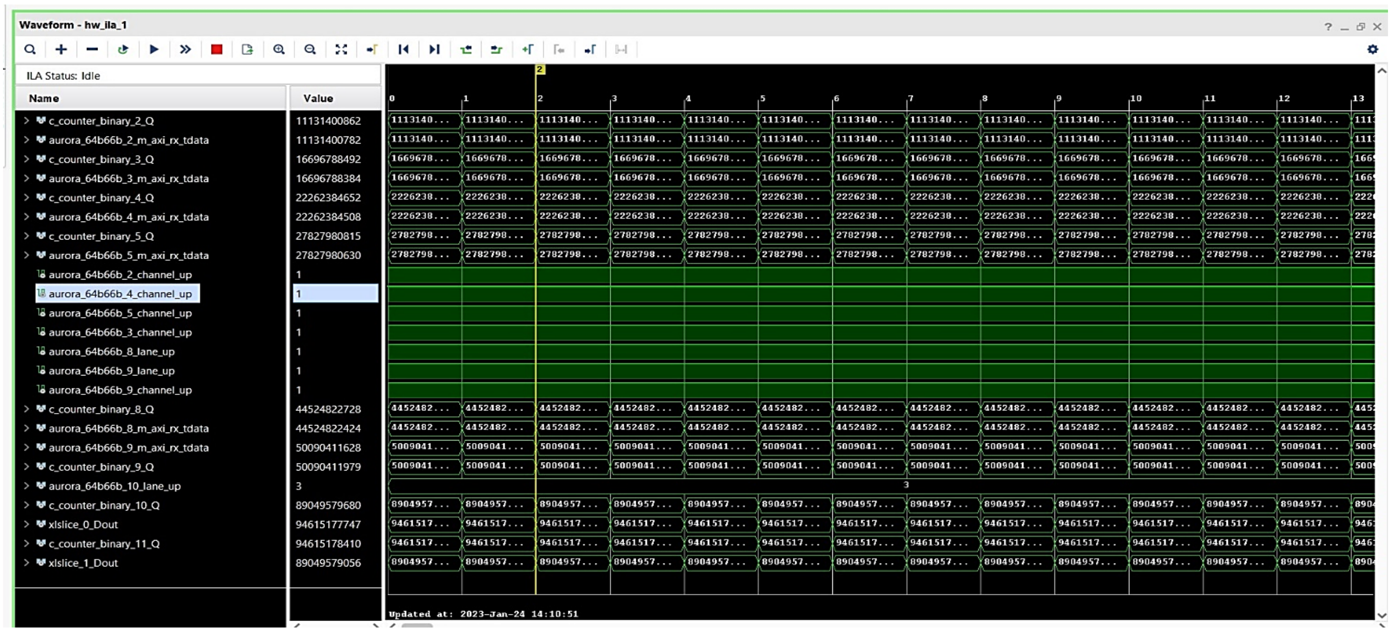


Figure. 9. Major components of PRBS-BIST

Figure. 10 represents the "ILA View of all RX-TX signals, showing data type and latency" refers to utilizing an Integrated Logic Analyzer (ILA) to monitor data transmission and reception signals. The ILA captures these signals, while also identifying their data type, such as binary or ASCII, and measuring latency, which is the time it takes for data to travel from the transmitter to the receiver. This analysis is crucial for assessing timing performance, ensuring data is transmitted and received within expected timeframes, and gaining insights into the behavior and efficiency of the communication system in a concise and detailed manner. *Figure. 11* represents the hardware manager displays BER injection and loopback mode options. "Inject" allows controlled bit error introduction, while "0-external loopback mode" signifies data sent externally for testing. *Figure. 8* represents the hybrid multiplier design with PRBS as block-level IP, offering runtime PRBS data control for testing flexibility.



Waveform - hw_ila_1

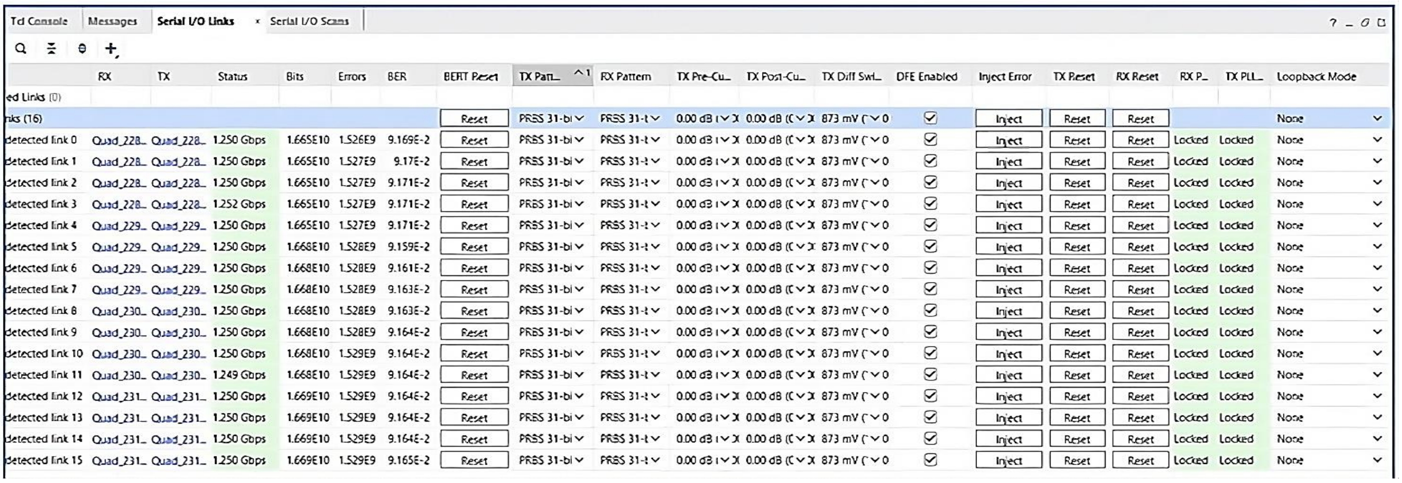
ILA Status: Idle

Name	Value
c_counter_binary_2_Q	11131400862
aurora_64b66b_2_m_axi_rx_tdata	11131400782
c_counter_binary_3_Q	16696788492
aurora_64b66b_3_m_axi_rx_tdata	16696788384
c_counter_binary_4_Q	22262384652
aurora_64b66b_4_m_axi_rx_tdata	22262384508
c_counter_binary_5_Q	27827980815
aurora_64b66b_5_m_axi_rx_tdata	27827980630
aurora_64b66b_2_channel_up	1
aurora_64b66b_4_channel_up	1
aurora_64b66b_5_channel_up	1
aurora_64b66b_3_channel_up	1
aurora_64b66b_8_lane_up	1
aurora_64b66b_9_lane_up	1
aurora_64b66b_9_channel_up	1
c_counter_binary_8_Q	44524822728
aurora_64b66b_8_m_axi_rx_tdata	44524822424
aurora_64b66b_9_m_axi_rx_tdata	50090411628
c_counter_binary_9_Q	50090411979
aurora_64b66b_10_lane_up	3
c_counter_binary_10_Q	89049579680
xlslice_0_Dout	94615177747
c_counter_binary_11_Q	94615178410
xlslice_1_Dout	89049579056

Updated at: 2023-Jan-24 14:10:51

Figure. 10: ILA View of all RX-TX signals, showing data type and latency

The provided information presents a comprehensive comparison between a "Reference Design" and a "Proposed Design" for an IP-based transmission loopback system employing PRBS (Pseudorandom Binary Sequence) generator and checker. In *Table I*, various key parameters were evaluated. Notably, the Proposed Design, which utilizes more advanced 16nm technology, displayed promising improvements. It operated at a slightly lower supply voltage, indicating enhanced power efficiency. Despite a significantly lower clock frequency of 156.25 MHz, the Proposed Design achieved a higher data rate (12.3125 Gbps) and exhibited lower worst-case jitter (0.576 ps). Additionally, it consumed less power under nominal conditions. Conversely, the Reference Design excelled in supporting a higher maximum data rate of 13 Gbps in its PRBS checker.



	RX	TX	Status	Bits	Errors	BER	BERT Reset	TX Patt...	RX Pattern	TX Pre-Cu...	TX Post-Cu...	TX Diff Swi...	DFE Enabled	Inject Error	TX Reset	RX Reset	RX P...	TX PLL...	Loopback Mode
ed Links (0)																			
nks (16)							Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset			None
detected link 0	Quad_228_	Quad_228_	1.250 Gbps	1.665E10	1.526E9	9.169E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 1	Quad_228_	Quad_228_	1.250 Gbps	1.665E10	1.527E9	9.17E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 2	Quad_228_	Quad_228_	1.250 Gbps	1.665E10	1.527E9	9.171E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 3	Quad_228_	Quad_228_	1.252 Gbps	1.665E10	1.527E9	9.171E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 4	Quad_229_	Quad_229_	1.250 Gbps	1.665E10	1.527E9	9.171E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 5	Quad_229_	Quad_229_	1.250 Gbps	1.668E10	1.528E9	9.159E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 6	Quad_229_	Quad_229_	1.250 Gbps	1.668E10	1.528E9	9.161E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 7	Quad_229_	Quad_229_	1.250 Gbps	1.668E10	1.528E9	9.163E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 8	Quad_230_	Quad_230_	1.250 Gbps	1.668E10	1.528E9	9.163E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 9	Quad_230_	Quad_230_	1.250 Gbps	1.668E10	1.528E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 10	Quad_230_	Quad_230_	1.250 Gbps	1.668E10	1.529E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 11	Quad_230_	Quad_230_	1.249 Gbps	1.668E10	1.529E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 12	Quad_231_	Quad_231_	1.250 Gbps	1.669E10	1.529E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 13	Quad_231_	Quad_231_	1.250 Gbps	1.669E10	1.529E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 14	Quad_231_	Quad_231_	1.250 Gbps	1.669E10	1.529E9	9.164E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None
detected link 15	Quad_231_	Quad_231_	1.250 Gbps	1.669E10	1.529E9	9.165E-2	Reset	PRBS 31-bi	PRBS 31-t	0.00 dB (	0.00 dB (	873 mV (	☑	Inject	Reset	Reset	Locked	Locked	None

Figure. 11: Hardware manager showing BER (Inject option) and loopback mode(0-external loopback mode) for Hard PRBS available in Xilinx IP

4. CONCLUSION

The research findings, coupled with the supplementary results, underscore the substantial advancements offered by the proposed hybrid multiplier in the FPGA landscape. This innovative approach, enriched with integrated BIST and PRBS capabilities, emerges as a formidable solution. The data reveals a noteworthy reduction in DSP slice utilization from 16% to a mere 5%, a substantial achievement. This liberated computational resource, within the FPGA, can be redirected to address other complex tasks, significantly enhancing the device's overall efficiency. The runtime control feature for PRBS data types at the block level design not only ensures adaptability but also aligns with the dynamic needs of modern digital systems development. Additionally, future work could delve into optimizing the hybrid multiplier's performance in scenarios involving real-time constraints or exploring its potential applicability in different FPGA architectures. Further refinement of the runtime control feature for PRBS data types and its adaptability to diverse digital system requirements could enhance the versatility of the proposed solution.

REFERENCES

[1] Wei-Zen Chen, Guan-Sheng Huang, “A Low Power Programmable PRBS Generator and A Clock Multiplier Unit for 10 Gbps Serdes Applications,” IEEE ISCAS, pp. 3273-3276, 2006.

[2] Wang Ying, Li Lei, Liu Huihua, Zhou Wanting, Xu Feng, "BIST for 2.5-Gb/s SerDes based on dynamic detection", 2011 International Conference on Computational Problem-Solving (ICCP), pp.132-135, 2011.

[3] Leon Pavlovič, Matjaž Vidmar and Sašo Tomažič, " 10 Gb/s 215-1 pseudo-random binary sequence generator", Journal of Microelectronics, Electronic Components and Materials Vol. 42, No. 2 (2012), 104 – 108.

[4] R. R. R. Bodha, S. Sarafi, A. Kale, M. Koberle and J. Sturm, "A half-rate built-in self-test for high-speed serial interface using a PRBS generator and checker", Proc. Austrochip Workshop Microelectron. (Austrochip), pp. 43-46, 2019.

[5] Zhiqian Zhang and Jinmei Lai, "An efficient method for testing multipliers of embedded DSPs in FPGA," 2016 13th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT), 2016, pp. 695-697, doi: 10.1109/ICSICT.2016.7999014.

[6] A.S. Oyeniran, S.P. Azad and R. Ubar, "Combined Pseudo-Exhaustive and Deterministic Testing of Array Multipliers", IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 2018.

[7] P. Bhaskar Reddy, B. Adi Narayana, " Design and Testing of High-Speed Multipliers By Using Liner Feedback Shift Register", IJERA, Vol. 3, Issue 4, Jul-Aug 2013, pp.503-507.

[8] N. G. Padharpurkar and V. Ravi, "Implementation of BIST using self-checking circuits for multipliers," 2015 International Conference on Computer, Communication and Control (IC4), 2015, pp. 1-5, doi: 10.1109/IC4.2015.7375565.

[9] T.Srinivasa Rao, V Bharathi Devarakonda, " Implementation of Low Power BIST for 32 bit Vedic Multiplier", IJLTET, Vol. 6, Issue3, Jan2016, pp. 6-15, http://www.ijltet.org/journal//2.pdf.

[10] B. Mishra, R. Jain and R. Saraswat, "Low power BIST based multiplier design and simulation using FPGA," 2016 IEEE Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2016, pp. 1-6, doi: 10.1109/SCEECS.2016.7509284.

[11] M. D. Pulukuri, G. J. Starr and C. E. Stroud, "On Built-In Self-Test for multipliers," Proceedings of the IEEE SoutheastCon 2010 (SoutheastCon), 2010, pp. 25-28, doi: 10.1109/SECON.2010.5453929.

[12] Dimitris Bakalis, Emmanouil Kalligeros, Dimitris Nikolos, Haridimos T. Vergos, George Alexiou, "On the design of low power BIST for multipliers

[13] with Booth encoding and Wallace tree summation ", Journal of System Architecture: the EUROMICRO Journal, Volume 48 Issue 4-5, December 2002, pp 125–135 https://doi.org/10.1016/S1383-7621(02)00121-2.

[14] A. S. Oyeniran, S. P. Azad and R. Ubar, "Parallel Pseudo-Exhaustive Testing of Array Multipliers with Data-Controlled Segmentation," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8350936.

[15] Pushpraj Singh Tanwar, Priyanka Shrivastava, "VHDL Implementation of Logic BIST (Built in Self-Test) Architecture for Multiplier Circuit for High Test Coverage in VLSI Chips", 2014, Vol 03 (11), pp. 12864-12870, 10.15662/ijareeie.2014.0311011.

[16] Dileep Kumar, Ghanshyam, " VLSI Implementation for BIST Controller using Signed and Unsigned Multiplier", 2015, Vol 04 (1), pp. 1226-1230, Paper ID: SUB15447

[17] Bhandari, Jugal & Verma, Yogesh & Singh, Laxman & Gupta, Santosh. (2022). A Novel Design of High-Performance Hybrid Multiplier. Journal of Circuits, Systems and Computers. 31. 10.1142/S0218126622502681.