EasyChair Preprint

№ 2916

Implementation of X-tolerant LBIST with Adaptive Intelligence Testing

Leela Krishna Thota, Ravi Teja Battala and Arun Madhavan

March 11, 2020

Implementation of X-tolerant LBIST with Adaptive Intelligence Testing

Leela Krishna Thota
Samsung Semiconductor R&D
Benguluru, INDIA
leela.thota@samsung.com

Ravi Teja Battala
Samsung Semicondctor R&D
Benguluru, INDIA
r.tejabatta@samsung.com

Arun Madhavan
Synospsys India Pvt. Ltd.
Benguluru, INDIA
amadhav@synopsys.com

***Abstract*— Complexed silicon chips are the foundation for everything from smartphones and wearables to self-driving cars and machines that learn. Requirements to meet ISO 26262 with Automotive Safety Integrity Levels up to ASIL D is quite challenging. Deployment of an automotive chip to provide dynamic in-system testing for critical failures and in-order to meet stringent Automotive Functional Safety requirements Advanced Intelligent Test (AIT) integration with POST and XLBIST was effective. RTL DFT integration flow allows us to verify the physically aware and power optimized placement early in design flow. XLBIST leverages its unique data reseeding approach with which observations on significant test coverage and test time improvements over regular logic BIST solution looks tremendous. This provides us an exceptional high level of in-system test fault coverage with minimal test time. AIT provides the functionality to execute autonomous and independent BIST operation on resources via IEEE 1500 network compatible.**

Keywords— ASIL, FuSa, AIT, IEEE 1500, XLBIST, Power management unit, ISO 26262

I. INTRODUCTION

Our design is targeted for automotive application, where safety is of utmost importance. Inspite of high standard manufacturing test, due to Complex nature of SOC's used in automotive, reliability defects manifest during Insystem operation due to operational and environmental conditions. Hence, Insystem architecture integrating X-Tolerant Logic BIST methodology with AIT is used. Toughest challenge lies in translating the structural ATPG vectors into a complete set of self-contained functional feature. This paper intends to provide details on implementation of Insystem architecture, challenges faced and solutions adopted.

II. GOALS

First, Primary goals of implemented architecture are as below.

- To achieve high permanent test coverage with minimal latency and data volume.
- To provide OFF field diagnostic capability.
- To run Power ON/Insystem self test utilizing functional resources for test storage, application and access

III. INTRODUCTION TO XLBIST

Even though our design is not targeted for ASIL certification, XLBIST is fully compliant with ISO26262 safety standards required in automotive industry. Unknown X's can be of static/dynamic in nature. Static X's are due to tristate structures, uninitialized state elements, analog blocks that are not completely wrapped for test, cross-domain clock signals etc. Dynamic X's that arise from timing exceptions are even more challenging as it may be more difficult to fix them without affecting timing/power constraints of design. This is the main reason behind choosing XLBIST over LBIST.

A. *Advantages of XLBIST:*

- Low pin count.
- High Compression.
- Low area Impact.
- Not necessary to do 100% X bounding.

Diagnosis flow helps to regenerate XLBIST patterns for masking X's arising post silicon that are dynamic in nature.

B. *Implementation:*

Frequent on/off application of automotive require low latency than ate test. To minimize embedded memory cost, data volume needs to be minimal. One possible solution to minimize test time could be running shift clock at higher frequency, however this creates difficulty in timing closure and increased test power. Hence, 24 MHZ is chosen as shift frequency. Analysis shows that latency and data volume can be reduced to a greater extent by eliminating Xs. One possible way to eliminate Xs is by making design robust, this comes at the disadvantage of increased hardware complexity. Alternative could be to have test patterns that are X free using ATPG tool specific enhancements.

This SOC implementation used combination of above-mentioned strategies at all levels to achieve significant test coverage and low data volume. Testmax solution is used to insert scan & xlbist logic during RTL stage. This provided a flexibility to verify the design during RTL stage itself. Care is taken to isolate functional logic from logic under test using isolation methodology. Irrespective of the sequential hardware compression or LBIST, X's are always a challenge in DFT. Wrappers included during RTL stage are enabled with isolation logic, thereby preventing X propagation into inputs, and leakage of X from outputs due to random scan toggle.

During Scan insertion, X sources are identified using Design compiler and test points are inserted to make design x robust. Also RRFA analysis is preformed to identify hard to detect faults. These optimizations allowed to bring test time and test data volume within automotive platform budget requirements.

C. *Overview of XLBIST at Core Level:*

- XLBIST IP access is done through IEEE1500 interface
- Re-Uses existing Scan Structure and OCC (On-Chip Clocking) circuits
- Various Modes of Operation Supported is listed in below table.

TABLE I. MODES OF OPERATION

Mode	Mode Access
Function(Mission)	Mission mode setting by test mode pin
XLBIST	Mission + IEEE1500 (Custom Data Register setting)
Compression(DFTMAX,DFTMAX-Ultra)	Compression mode setting by test mode or IEEE1500
Internal Scan / Wrapper INTEST	Internal Scan/Wrapper INTEST mode setting by test mode pin or IEEE1500
Wrapper EXTEST	Wrapper EXTEST mode setting by test mode pin or IEEE1500

Below schematic shows the co-existence of both DFTMAXULTRA and XLBIST structures.

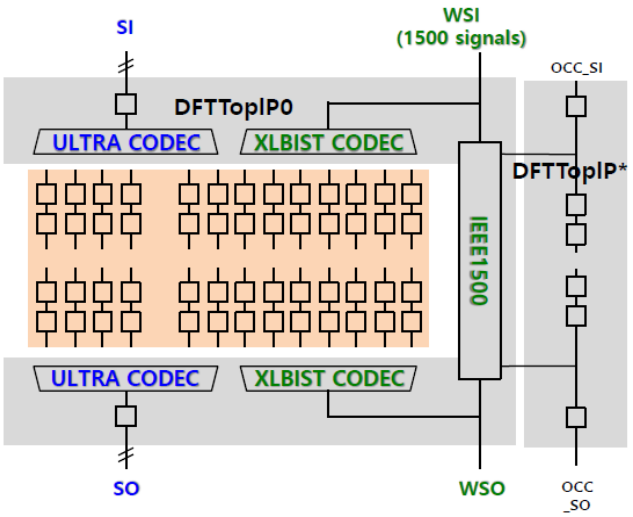


Fig. 1. Core level Implementation of ULTRA & XLBIST

From the above Fig. 1. It is clear that the OCC chain is separately stitched which is common for both Ultra & Xlbist.

Identify applicable funding agency here. If none, delete this text box.

IV. ADAPTIVE INTELLIGENCE TESTING

AIT provides functionality to execute structural patterns on components through IEEE 1500 interface ,thereby bridging the gap between ate and platform specific operating conditions. IEEE 1500 compliant structural patterns are generated using ICL and PDL. A Network is built to port these patterns from sub block to AIT Controller level using Embedit. IEEE 1500 compliant structural patterns are converted to AIT packets using Testmax and stored on chip memory. AIT controller fetches, decodes and executes instructions stored on chip memory, drives and observes components on system through IEEE 1500 interface. There is a provision in design to Bypass AIT & access components on CHIP through General Purpose I/O's, for ATE testing of XLBIST ring and diagnosis.

A. *DFT Architecture for POST / In-System Testing*

Design specification requires to enable POST (key-in/key-off) and in-system requirements below additional DFT IPs are architected into the SoC.

1. Synopsys based XLBIST.
2. Built in self-test circuitry for logic at Block level.
3. Synopsys based Server AIT controller.
4. SoC level module to enable functioning of XLBIST access through various interfaces viz JTAG, APB & custom interface.

B. *Overview of AIT_SERVER Controller*

The AIT controller integrated at system level enables access to all the XLBIST IP inserted at blocks. The other features of AIT_SERVER controller include:

- Enable ATE (Automatic Testing Equipment) access through JTAG 1149.1 interface.
- Provides APB and custom interface that could be used through PCIe or PMU during mission mode.

Two modes of access provided to XLBIST module are as below

- Instructions to XLBIST are stored in RAM/ROM accessible through AIT.
- JTAG or APB direct access to XLBIST IP through bypass mechanism.

In current design, there is a provision to operate XLBIST in various modes as mentioned below Fig. 2.

1.) **POST MODE**
2.) **Insystem Test MODE**
3.) **ATE MODE**

1) **POST MODE**: Stuck at test for all cores and transition mode for sub block with PCIe interface (which interacts with host) is supported in this mode. In this mode, AIT controller is triggered through controller interface by PMU as part of boot sequence. AIT controller starts executing the instruction from ROM address given as a part of boot sequence. AIT packets fetched from ROM are decoded and fed to XLBIST 1500 interface from AIT P1500 interface. After the completion of XLBIST operation, AIT provides test status to PMU via controller interface.

2) **InSystem Test MODE:** In this mode, any partition on demand from the host can be run. Insystem test could be performed in various modes. Few among them are mentioned below.
    i) **Through AIT Control Interface**: In this mode, HOST programs PMU registers which in turn trigger AIT controller through its controller interface. Rest of the operation is similar to POST mode. Partition in test is isolated from rest of the logic which is controlled by HOST programming PMU(Power Management Unit) registers. AIT upon the completion of XLBIST operation provides status output to PMU via controller interface.
    ii) **Through APB Interface of Server:** In this mode HOST programs the XLBIST logic of cores directly through server APB interface. AIT controller is bypassed. APB transactions are converted to P1500 transactions and fed to 1500 interface of XLBIST i/p. After waiting for test to complete, MISR is read through APB interface to compare with golden signature.

3) **ATE MODE**: This mode is used for diagnosis as well as for manufacturing test.
    i) **Through JTAG interface of sever:** AIT is bypassed in this mode. Server drives XLBIST 1500 interface by converting 1149.1 transactions fed from JTAG interface to P1500 transactions. After waiting for test to finish, MISR signature is readout through TDO and compared with golden signature. Using XLBIST pattern data during ATE test provides additional flexibility and test opportunity from the generally limited self-test environment. The test sequence through an IEEE 1149.1 interface is directly mappable to the equivalent deterministic-ATPG test sequence; this correspondence supports direct diagnosis of failures. Also there is a flexibility to execute various cores parallelly. To accommodate boot time requirements, few cores are grouped(considering switching activity) and tested.
    ii) **Through JTAG interface of server to program AIT controller**: In this mode AIT is controlled from IEEE 1149.1 interface. Server programs AIT to operate in POST mode by translating IEEE 1149.1 signals to P1500 signals. Upon trigger AIT starts fetching data from ROM, start address is provided from JTAG interface by Programming TDR registers in server. P1500 interface of AIT drives XLBIST inputs. After completion of test, MISR signature is compared and status bits are updated which can be read through GPIOs.

Below is schematic to explain system level interaction of AIT, SERVER and XLBIST :

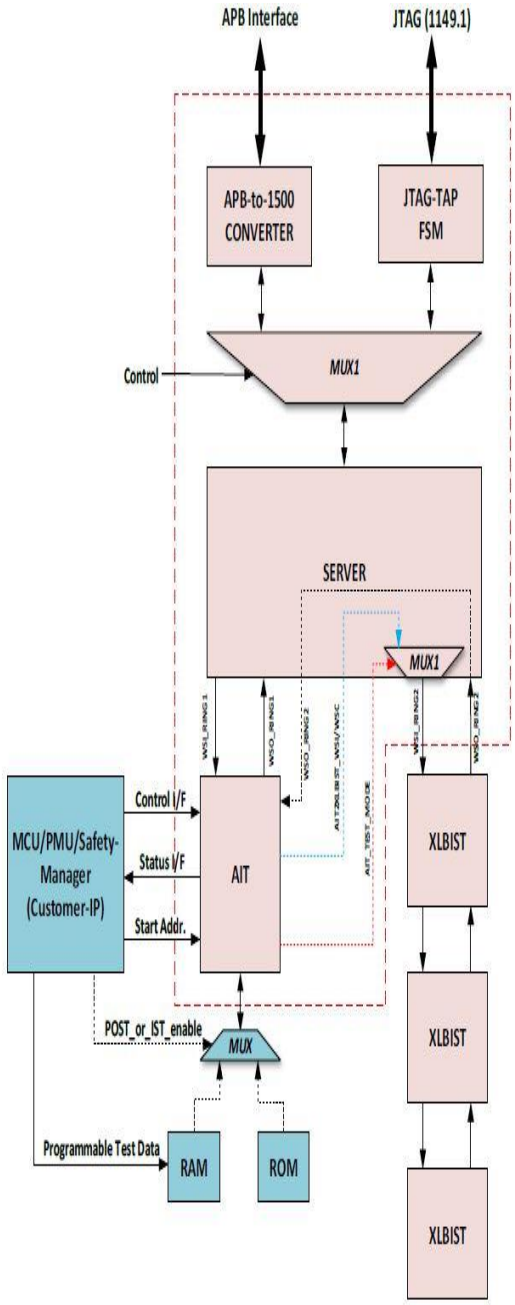


Fig. 2. Interaction of AIT controller - Server - XLBIST

V. SOC ARCHITECTURE

SOC is built by integration of various third party IP's & embedded memories to achieve desired functionality. Testing challenges faced due to unpredictable X sources from third party IP vendors made this safety critical automotive chip more complex. Proposed X-Tolerant Logic BIST methodology is implemented in bottom-up approach to attain optimal test data volume & time.

At chip level we need to access for JTAG-based test controller, and choose available test mode by setting instruction & data registers. Once test mode and other setting values are assigned, we can access each hierarchy through top level PAD's. DFT Pin muxing is implemented in 2 levels.

First level happens inside blocks to reduce the signal crossings between the blocks & TOP. Final level of Pin muxing is implemented at Top.

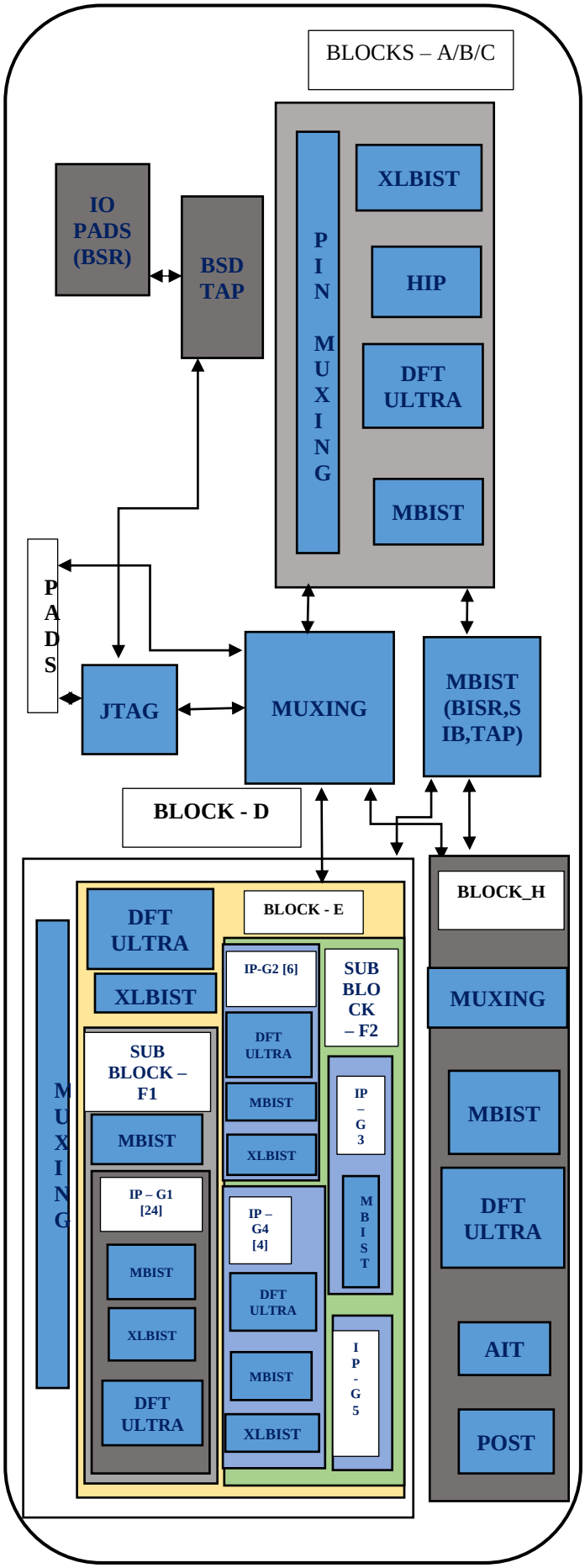


Fig. 3. SOC DFT Architecture

The XLBIST implementation within different blocks is integrated at SOC to fulfil the requirements. There are a total of 6 blocks within which few sub blocks are embedded. Blocks A,B,C are of similar type where XLBIST is integrated along with DFT Ultra, MBIST & atleast one HIP. I/O pads interact with BSD Tap to decide the sequence of operations, which are controlled from JTAG_Top. MBIST at top level provides BISR to support self-repair feature. Block D is combination of Block E & pin muxing. Due to high complexity involved in Block D, hierarchical implementation is adapted in DFTMAX Seq compression technique. Complexity of Block E is huge due to the presence of Sub blocks – F1 & F2 along with their codecs for both ULTRA & XLBIST. Sub block - F1 consists of IP-G1 which is of 24 instances. Bottom up strategy is adopted in order to build the circuitry of inherited IP's. Sub block F2 consists of IP-G2, IP-G3,IP-G4 & IP-G5 which are of 6,2,4 & 1 instances respectively as shown in Fig. 3.

VI. Challenges & Resolutions

1. As hardmacro chains are tested in Internal loop back test and also to minimize X leakage, hardmacro chains are excluded from XLBIST. Additional Logic is added to get control over test specific signals viz Scan Reset, clock gating, memory marginal ports in Insystem mode. Certain dynamic X sources from MBIST logic due to multi cycle paths are handled by adding separate clock blocking circuitry in transition mode.
2. Data Volume depends on number of seeds used and test time depends on number of patterns. Non-interacting OCC's covering maximum scan flops are effectively grouped (manual clock weighting) for optimal utilization of seeds, thereby reducing test data volume.
3. Latches inferred in RTL are mapped to pre-existing Integrated clock Gating (ICG) library models easing timing closure. Also ATPG tools require circuitry to be designed precisely for testing to operate effectively. This is done in the RTL stage itself by testing the design for scan readiness using SPYGLASS, thereby fixing issues at the early stages.
4. Shadow wrapper test points are inserted around untestable blocks to allow logic around it to be tested. Force_01 test points inserted at output pins of blocks to drive known value. Observe points are inserted at input capturing known values to improve coverage. Test points are inserted for random resistant faults and to block x sources. Inserted registers are shared among various test points to mitigate area overhead. Please refer the below Table II inorder to understand the effect of test point insertion over test coverage.

TABLE II. Coverage Improvement with Test Point Insertion

BLOCK NAME	Original TEST COVERAGE(%)	Improved TEST COVERAGE(%)	# Test points	X Blocking
IP-G1	93.77	94.14	10716	32
IP-G2	95.3	96.4	9774	16
IP-G4	92.26	92.46	9669	605
BLK_A	87.62	90.53	13939	1700
BLK_B	88.98	92.11	29500	334
BLK_C	93.52	93.14	2590	231
SUB_BLK_F1	91.6	95.11	41843	364
BLK_E	90.01	93.11	37316	560

VII. Results

The below Fig. 4. captures the test coverage analysis obtained from best seed search evaluation.

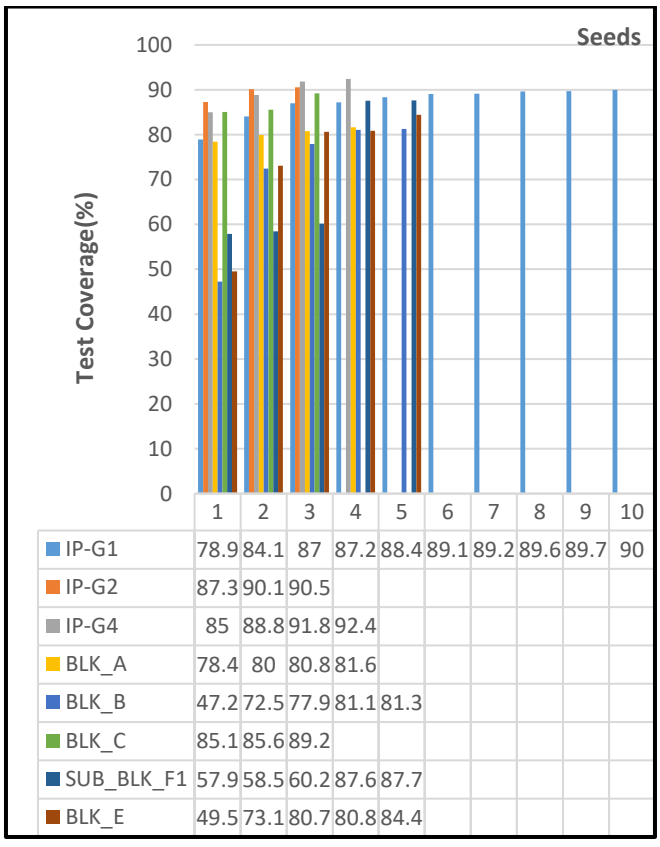


Fig. 4. SOC DFT Architecture

Various experiments are tried out with TetramaxII to find the best seeds for each block. Seeds thus obtained has maximal utilization thereby meeting the budgetary requirements of Boot ROM.

The coverage obtained for various blocks are captured for both the fault models – Stuck at & At speed in Fig.5 & Fig. 6

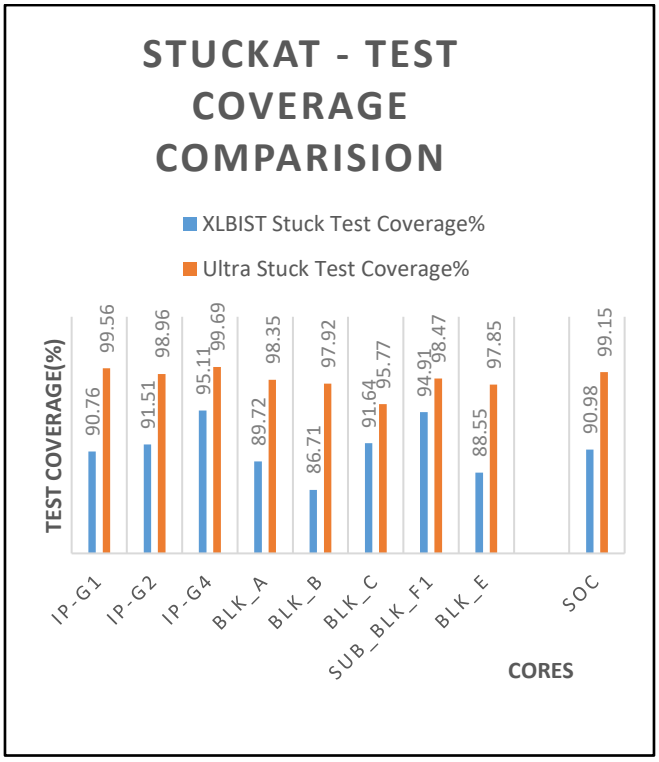
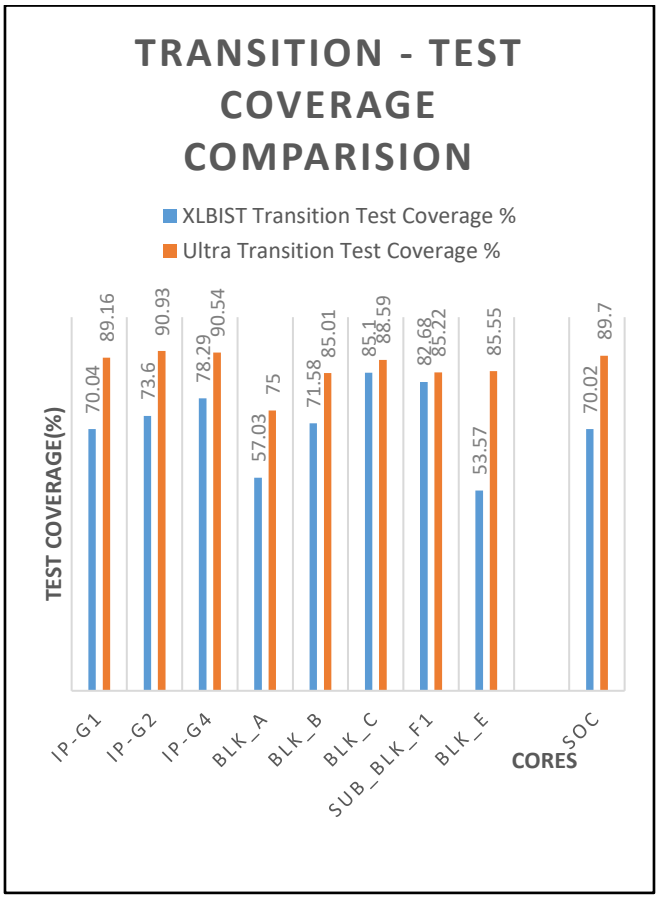


Fig. 5. Coverage Comparision between DFTMAX Seq Compressor Architecture Vs XLBIST.



Due to randomness & memory constraints, coverage obtained is comparatively less for Insystem XLBIST patterns over deterministic manufacturing patterns.

TABLE III. BOOT TIME CALCULATION – STUCK AT TEST

Block	No of AIT packets	No of instances	Stuck-AT Test Coverage	Test Time (ms)
IP-G1	7548	3	88.37	1426
IP-G2	1981	6	90.11	1826
IP-G4	2456	4	92.39	1302
BLK_C	1929	2	89.23	792
BLK_E	7930	1	86.42	1670
SUB_BLK_F1	4943	1	94.91	564
BLK_A	2687	2	81.95	886
BLK_B	13003	1	83.11	1052

Above Table III explains the number of AIT packets used by each partition and corresponding boot time.

VIII. CONCLUSION

Considering number of challenged to be dealt in the design & verification space productizing the complex Self testing architecture was not an easy task. The implemented architecture has capability for detecting both manufacturing as well as permanent defects occurring over a period of time.

REFERENCES

[1] Pavan Kumar Datla Jagannadha, Mahmut Yilmaz, Milind Sonawane, Sailendra Chadalavada, Shantanu Sarangi, Bonita Bhaskaran, Shashank Bajpai, Venkat Abilash Reddy, Jayesh Pandey, Sam Jiang, Special Session: In-System-Test (IST) Architecture for NVIDIA Drive-AGX Platforms, 2019 IEEE 37th VLSI Test Symposium

[2] P. Wohl , J.A. Waicukauski ,G.A. Maston ,J.E. Colburn,XLBIST-X-Tolerant Logic Bist,2018 IEEE International Test Conference(ITC)

[3] Ghazanfar Ali ,Fawnizu Azmadi Hussin ,Noohul Basheer Zain Ali , Nor Hisham Hamid,On using IEEE 1500 standard for functional testing,IEEE conference,Penang,Malaysia

[4] .Mahmut Yilmaz, Animesh Khare, Rahul Garg, Mayank Parasrampuria,Nitin Yogi, Jaison Kurien, Rahul P R, Shantanu Sarangi, Krishna Rajan,"X elimination to improve the quality of scan compression",International Test Conference (ITC)-India, 2017