

A Novel Fault Zone Tiling Approach Based Error Correcting and Detecting Method for Network on Chip Design

P. Ponsudha^a, G. Akshaya^b, P. K. Anusuya^c, S. Elizabeth Tabitha^d, J. Pavithra^e

^{a,b,c,d,e} Dept of ECE, Velammal Engineering College, Chennai, Tamilnadu, India.

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 23 May 2021

Abstract: The behavior of System-on-Chip (SoC) is complicated because they have multiple processors that communicate with each other through concurrent interconnects, such as Network-on-Chip (NoC). It is difficult to debug such SoCs based on a classification of debugging scope and granularity. The lack of observability of internal operations during emulation and post-silicon validation of networks-on-chip (NoCs) makes it difficult to detect and debug functional bugs. Tests that exercise the control-flow portion of the NoC's functionality while abstracting the data content of traffic are required to verify its correctness. The limited trace port bandwidth and buffer size limit the effectiveness of at-speed silicon debug, necessitating very efficient data compression solutions. Because unknown 'X' values are almost always present in silicon debugging, trace compressors must include an X-tolerance features to avoid significantly reducing error detection capability. To address this issue, this paper introduces X-Tracer, a novel reconfigurable X-tolerant trace compressor that can tolerate as many X-bits as possible while maintaining a huge compression ratio with low cost of additional design-for-debug hardware. This work also included fault zone protection based on Tiling so as to prevent incorrect rechecking. The proposed project was coded in HDL and simulated with Xilinx

Keywords: Trace Data Compression, Network on chip, Silicon Debug, X-Tolerance, Tiling

1. Introduction

Network-on-Chip (NoC) is an approach to design the communication subsystem between intellectual property cores in an SoC design. SoC uses dedicated buses between communicating resources which have no flexibility in terms of communication. As the number of resources grows, the common buses do not scale very well. NoC aims to address these issues by implementing a communication network of micro routers and resources [1-2]. A different router ECDR2 has been proposed by reducing the pipeline stages [3]. The NoC design paradigm has been proposed as the future of ASIC design [4]. The major driving force behind the transition to NoC based solutions is the inadequacy of current-day VLSI inter-chip communication design methodology for the deep sub-micron chip manufacturing technology [5]. A generic two stage router that supports high reliability and less cost for NoC is introduced [6].

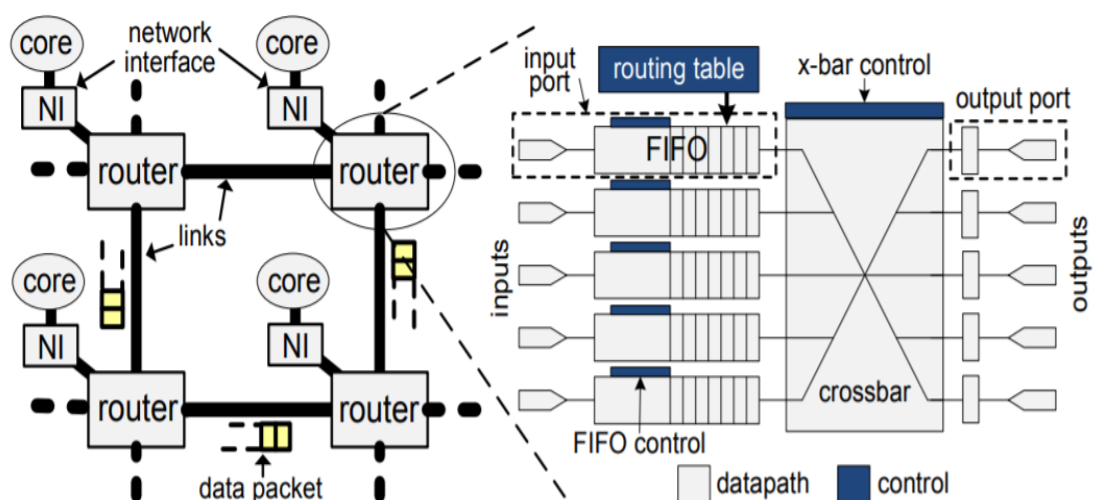


Figure 1. Basic Architecture of Network-on-chip.

Various design issues are imposed by NoC based System-on-Chips in the fabrication of such ICs. Initially, the reasonable topology for target NoCs such that the design and performance constraints are met. Secondly, to transport data among processing cores, the physical interconnection mechanisms are provided by the design of network interfaces. Further, suitable communication protocols are selected for on-chip interconnection networks. Finally, since technology scales and switching speed increases, networks on chips in the future will be very

sensitive, prone to faults and errors. Critical parameter for on-chip communications is fault tolerance [7]. For current SoCs, to minimize wire routing congestion, to ease timing closure, to change IP easily and for high operating frequencies, a network on chip IP interconnect fabric is required. Compared to a bus-based solution, NoC design space is larger, as different routing organizations of the communication infrastructure and arbitration strategies can be. Design productivity in NoC platform can grow as fast as technology capabilities and can eventually vanish the design productivity gap [8]. To deal with communication bottlenecks and to tolerate faults and NoCs have an inherent redundancy [9]. For detecting short faults in communication data links in NoC, a new BIST based testing approach is introduced [10].

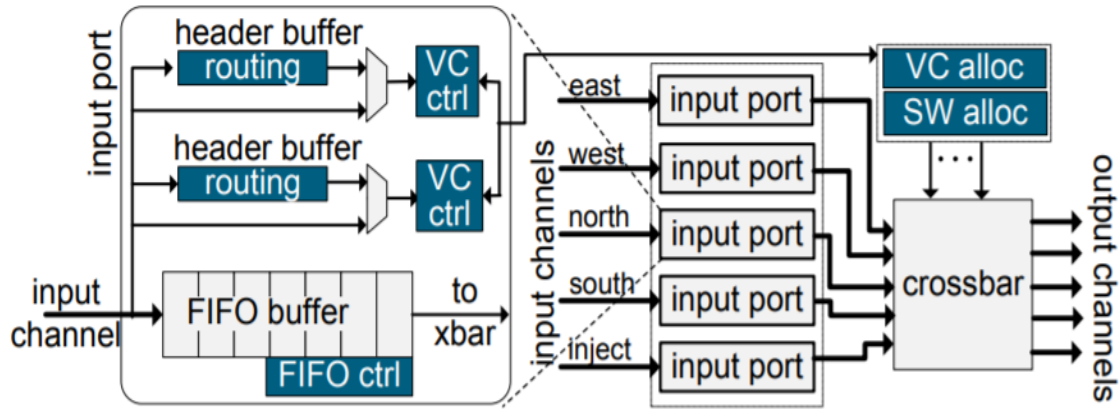


Figure 2. A virtual channel router.

An NoCs are a group of *routers* connected together through links and organized in a certain topology. *Links* are a series of wires that interconnects the network's routers. Two full-duplex NoC links bind the routers on a network. The channel bit width is the number of wires per channel which is uniform throughout the network. To transmit packets between routers, links are used. These packets are divided into small blocks known as flits.

Flits are Flow control bits. Each block in Figure. 3 represents flits. Packets are injected into the network through a dedicated network interface which creates the interface between a network router and a core.

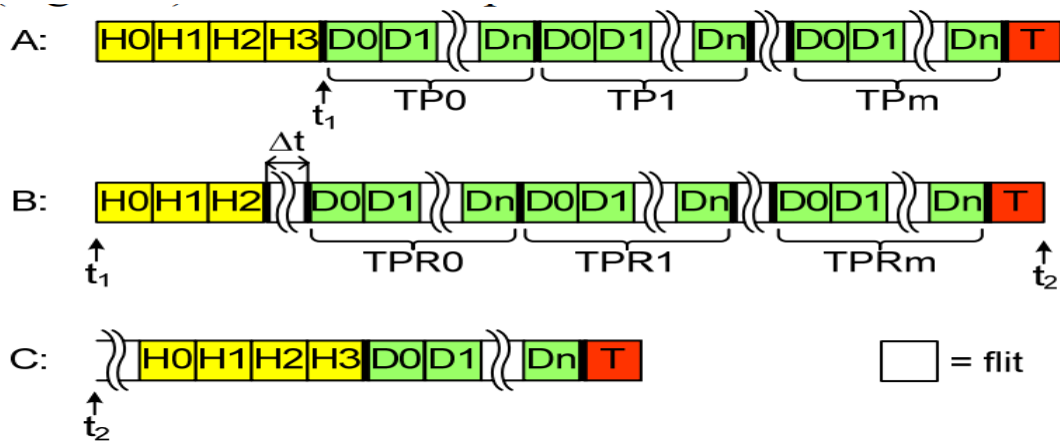


Figure 3. NoC test-data packets Structure[23]

As packets are injected into the network, packets are provided with temporary buffering by its routers as they are routed to respective destinations according to defined routing protocol.

In section II related works such as X-tolerant Trace compressor, NoC Fault detection and correction and tiling approach are explained, in section III proposed system is designed and analyzed, in section IV simulation results are discussed in section V performance of the system is analyzed and in section VI the paper is concluded.

2. Related Work

a. X-Tolerant trace compressor

In silicon debugging, the main challenge is the restricted visibility of internal signals because the circuit under debug is a piece of silicon that has been manufactured already. To enhance its observability to fix this issue, design-for-debug circuitries are typically added to the design. One of the effective silicon debug techniques is trace-based debug, which enables designers to track a series of signals in consecutive cycles in real time while remaining non-intrusive to the circuit's regular operation. The aim of trace-based debugging is to find and correct errors in as less debug runs as possible. On the one hand, it is not possible to trace all the circuit's internal signals, the efficacy of the trace-based debugging relies on the accuracy of the selected trace signals, which includes both manually selected signals by experienced designers and signals selected by automated-solutions driven by certain visibility enhancement metrics [11-14]. On the other hand, even with pre-determined trace signals, a bug can only manifest itself at a particular moment, so it's important to make sure the signals on "right" time can be traced. Since trace-based debug features a important overhead and it solely features a little trace buffer size and some external pins to use as trace ports, the number of trace in every debug run is restricted. The on top of trace compression solutions have to this point targeted on debugging microprocessors and signal tracing techniques normally logic circuits to enhance their error detection capability, and that they are often typically classified into 3 types:

- Lossless trace compressors have faith in the locality of trace information to realize lossless compression. Anis and Nicolici provided variety of dictionary-based compressors for following repeatable information in [15]. In [16] differential information was compressed to realize higher compression quality supported the observation that toggling rate of state values is usually low.
- A spatial lossy trace compressor compresses a series of N signals into M parity signals until signal tracing ($N > M$) using associate degree XOR network [17]. Such spatial compressors area unit commonly organized as a tree-like structure as a part of the trace interconnection fabric to minimize routing overhead.
- Using the multiple-input signature register, that was originally used for test response compaction within the VLSI testing domain, temporal lossy trace compressors compress variety of cycles of information into a signature throughout signal tracing [18]. In [18] they in turn zooms-in the failure signatures by reconfiguring the compaction ratio in their MISR-based compressor for every debug run to localize the error. With the idea that the CUD acts repeatedly in numerous debug iterations, [18] zooms-in the failure signatures by reconfiguring the compaction ratio in their MISR-based compressor for every debug run to localise the error.

b. NoC Fault detection and correction

The key technical issues surrounding Network - on - Chip are investigated, focusing on the analysis of common NoC mapping issues and conducting additional research on fault-aware NoC task mapping. NoC and Field Programmable Gate Array issues were verified on the verification platform [19]. For on-chip communication a faulty aware routing algorithm is given that detects the precise locations of faulty nodes and faulty links on the network and routes information around them. In a dynamic NoCs setting where the position and variety of faulty nodes/links change throughout runtime. This routing approach uses algorithmic error detection mechanisms to convey an adjustive routing path from source to destination tile in a very dynamic NoCs environment where the location and variety of faulty nodes/links change throughout runtime [20]. A cost-effective error correction code technique called the 2D fault coding methodology is given, to overcome the multi-bit transient fault issue of NOC links, The wires of a link are modelled when a matrix, and light-weight parity check coding is performed on the two dimensions of the matrix [20]. For each NOC and PEs layered stack of recovery mechanisms is proposed, making certain correct application execution within the presence of transient or permanent faults [21]. A novel online fault tolerance approach is introduced for NOC interconnects that addresses each permanent and transient faults. As a technique of characteristic between permanent and temporary faults, the principle of retransmission credit is introduced [22].

c. Tiling approach

The main principle of tiling is to avoid the fault zone areas. These tiled architectures show promise for high performance, larger integration, good scalability and probably high energy efficiency.

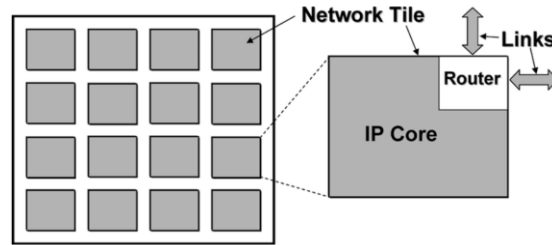
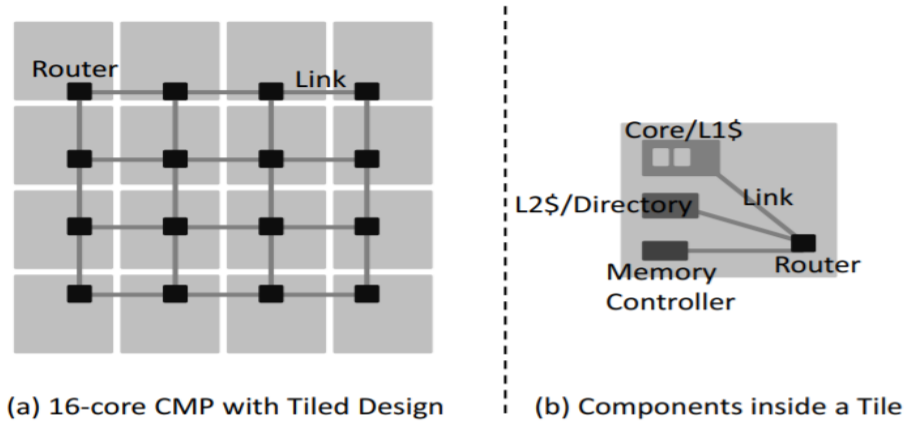


Figure. 4 Tiled Network Architecture



(a) 16-core CMP with Tiled Design

(b) Components inside a Tile

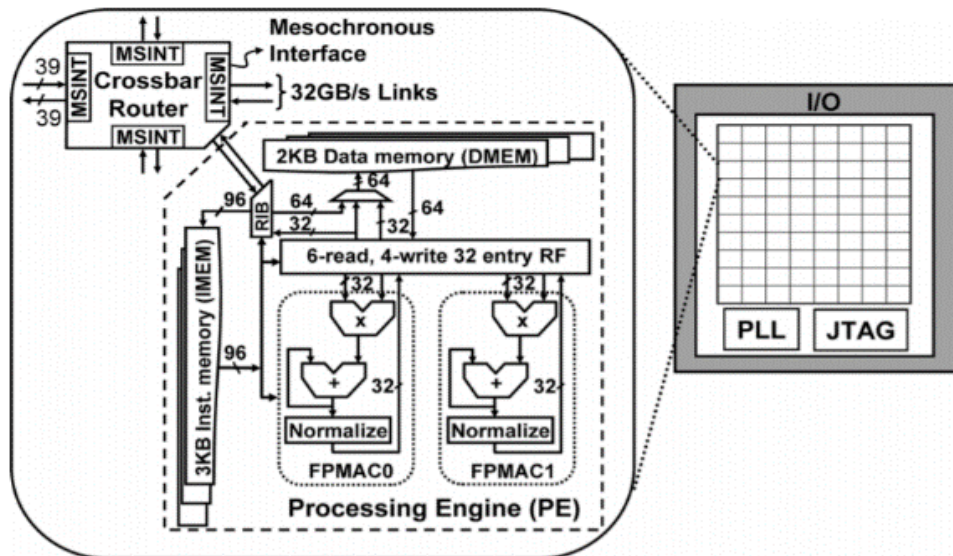


Figure. 5 NoC and Tile Architecture

3. Proposed System

a. Fault Zone Tiling Approach For Network-On-Chip :

The approach to a reliable SOC depends on the dependable data streaming techniques used. NOCs with interconnected identical processing tiles (PT) produce reliable SOC designs. An internal fully reconfigurable test pattern generator (TPG) and test pattern evaluator (TPE) is proposed.

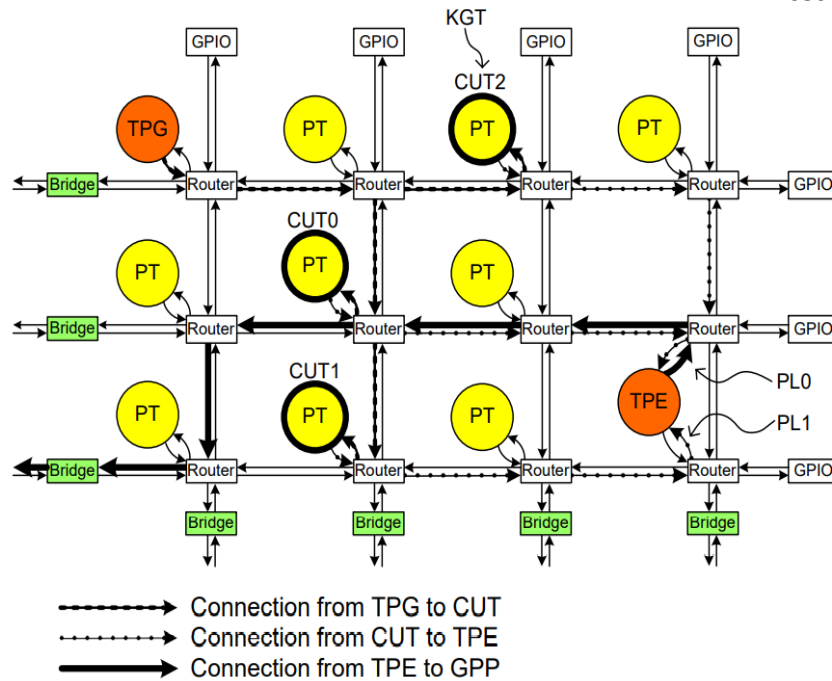


Figure. 6 SOC with Tiled identical PT [23]

In Figure 6, Test Pattern Generator (TPG) and Evaluator (TPE) in NoC. CUT Circuit under test, PT processor tiles, PL Physical links connecting NOC to the TPE are shown.

The NOC is used as test accessing device for the processor cores. The data packets from TPG through the NOC, and to the core are of the pattern is shown. The header flit starts the connection setup and the tail flit ends the connection. A direct effect of the ever-increasing IC transistor density is the rapid increase of volume of the test data. To reduce the volume of test stimulus patterns and a multiple-input signature register for the test response compaction by combining a linear-feedback shift register, test data compression methods like Deterministic Built In Self-Test (DBIST) is proposed.

The LFSR seeds are stored on test generator IP, and the test response signatures are compared to those of a proven fault-free design. The test responses are the same, when the same test stimuli are applied to many similar fault-free tiles on a many-core processor with several identical processing cores (tiles). If only one tile is defective at a time, the faulty tile can be detected by comparing the test responses from all tiles-under-test to the test responses from a Known-Good-Tile.

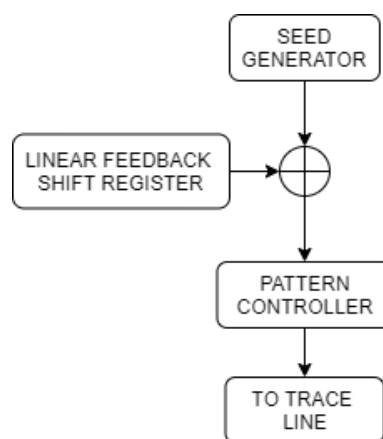


Figure 7. Test pattern generation and control

Internal test pattern generation and test-pattern validation infrastructure IPs are included in the implementation. Deterministic test patterns are produced by the LFSR combined by the seeding technique.

The diagnostic information is used to remap routing paths and tile resources. In the case of software for a specific application, erroneous paths and/or tiles The principle has been established. Simulations were used to verify the results.

b. TILE Test Pattern Evaluator

The test pattern evaluator (TPE) is physically connected to the NOC through the Physical Links. Connections are made between the NOC and the Processor IP cores. The TPG generates the test-pattern to be routed to the selected Circuit under test (CUT) and the trace generated is compared with the response data of the other fault free data. The comparator is made self checking.

c. Fault Zone Tiling Process

The steps in the system are as follows:

- Partitioning
- Testing
- Detection

Additional control circuitry is connected to each subcomponent during the failure isolation and recovery phases. The first stage of analysis is used to identify a system failures, and the second stage is then used to locate the defect in its individual subcomponent(s) [24].

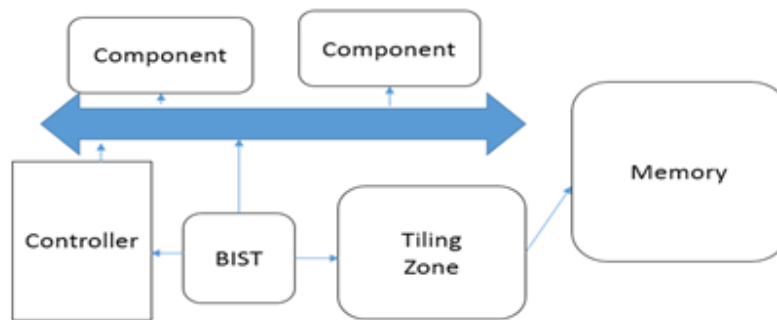


Figure 8: Implementation of Tiling Zone process

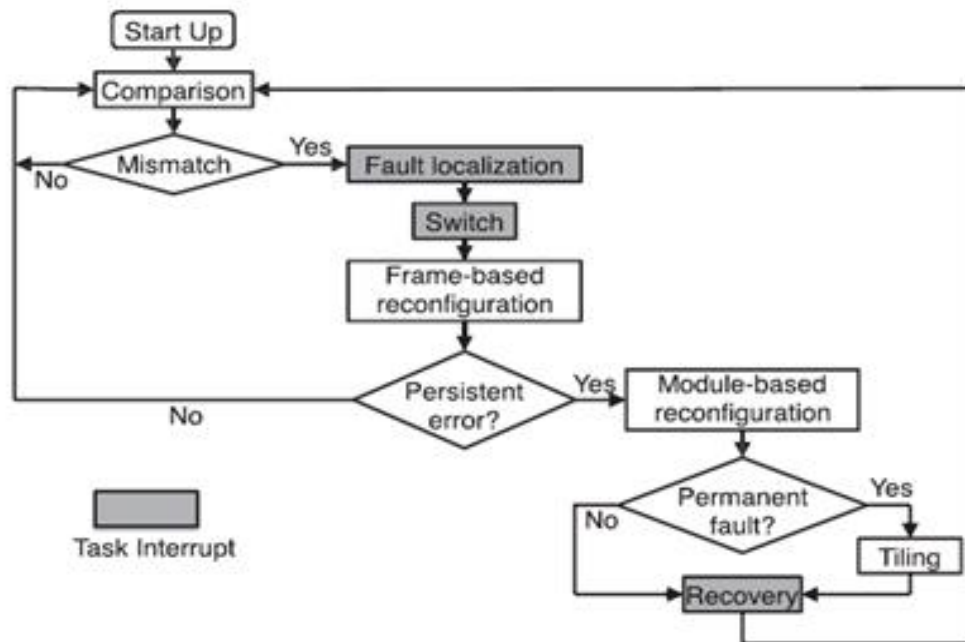


Figure 9: Flow chart of tiling process

The faulty part is substituted with reconfigurable logic just after detection point. We bypassed the failing portion while the segmentation and completed the process of recovery. The Comparison block from Figure 9 endlessly supervises the module throughout normal operation, whereas the all re-configurable blocks within the background are checked by Configuration Engine signal to be triggered by the Comparison block. A fault in one amongst 2 softcore processors or the Comparison block itself might cause a mismatch signal to be triggered by the Comparison block. To request scanning of the component, the Comparison block sends a mismatch signal to the FT Configuration Engine. The FT Configuration Engine instantly begins scanning the Comparison block and, if a

fault is found, the FT Configuration Engine can reconfigure it. Since the error should result to a fault in one in all the two cores if the Comparison block MUX is fault-free, the FT Configuration Engine solely has to search one core to search out the supply of the error. The scan doesn't have an effect on the core's mission, however the improved Lockstep scheme should be halted throughout this special scan to avoid any disastrous effects.

If the error remains during a frame despite recent configuration upsets being resolved in it, a permanent configuration fault occurs. Different reconfiguration techniques are chosen looking on the fault duration: standard partial reconfiguration for a brief fault or tiling technique for a permanent fault.

By precompiling an equivalent design with completely different implementations, every of that has the subsequent two properties, the application principle avoids using the FPGA's faulty field. First, it has a prohibited zone, which may be used to mask a permanent fault by charging the desired implementation configuration during which the prohibited zone overlaps the defective region. Second, it has its own bitstream, sanctioning the fault masking method to be administered by downloading the mandatory bitstream via ICAP then performing the partial reconfiguration procedure. The bitstream generation for numerous implementations within the system is completed using a simple placement constraint prohibit, that is assigned to the synthesis tool through the user placement constraint register. The X and Y coordinates of 2 points describe the taboo parallelogram region. The prohibited zone will be shifted inside the PRR by changing these coordinates.

d. X-Tolerant Trace Compressor Design

This mainly focuses on temporal trace compressor, that encompasses a considerably higher compression ratio than different compressor varieties. The mentioned three trace compression techniques are not mutually exclusive; in point of fact, they'll be combined to boost the CUD's real-time observability [25].

4. Simulation Results

The Figures displayed shows the Verilog simulation results for a 2D mesh 3X3 router NOC and TPE inputs and outputs. The input signals are the ini, reset, clock, down, up, right, left, trace (7:0). The data coming as input to NOC from the PT is the trace. TPE data is shown in q(7:0). Distance (3:0) shows the distance covered through the routers by the control signal inputs.

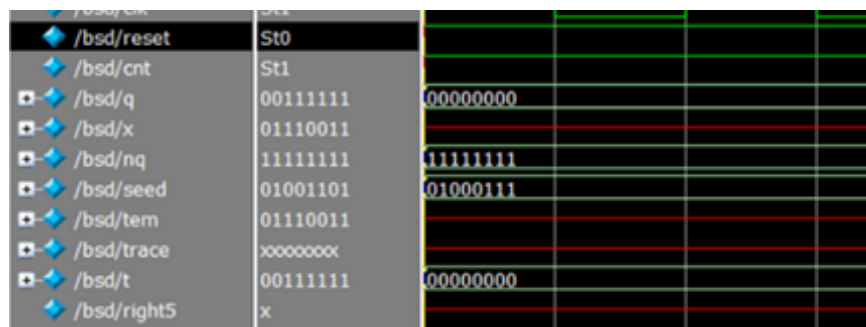


Figure 10. Router Initialization

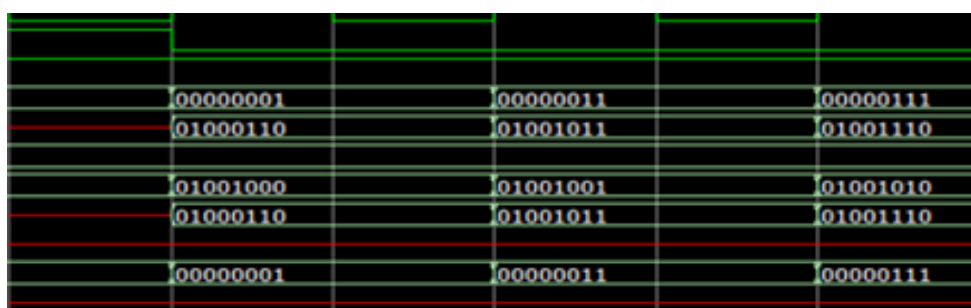


Figure 11. Pattern Generation

Signals are disabled after resetting, the Router configuration and generation of test patterns for trace buffers are shown in the figure 10. The created pattern with the fewest switching activities is shown in the figure 11.

The selected pattern for trace buffers and corresponding router to router paths is shown in the figure 11. The applied pattern traces after rejection are shown in the figure 11. The test line in the figure indicates whether the pattern was implemented or not.

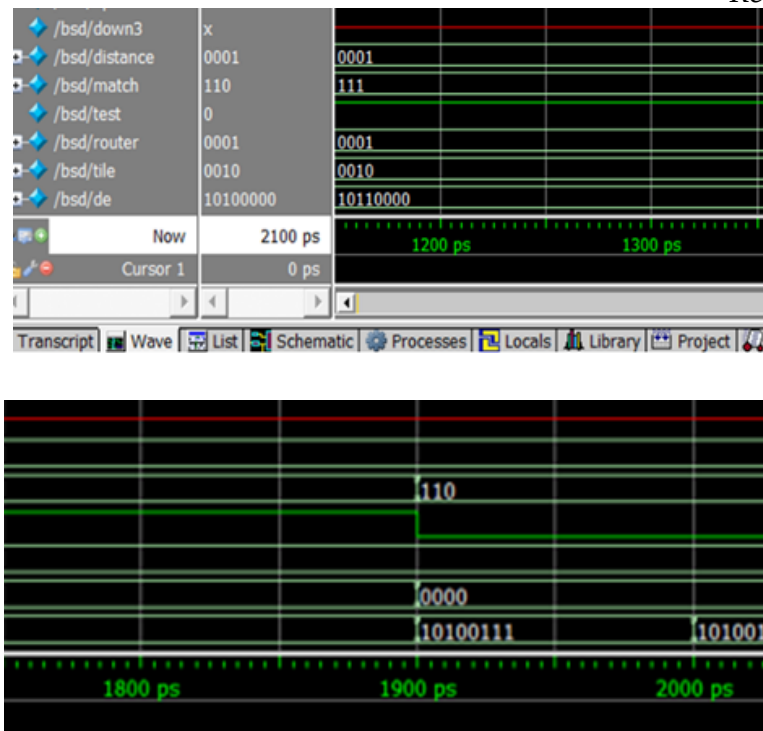


Figure. 12 Trace input monitoring

Using Xilinx, the proposed has been simulated, and the synthesis report can be obtained.

```
# pattern moniterd and rejected
VSIM 21> run
# pattern moniterd and rejected
VSIM 22> run
# pattern moniterd and rejected
VSIM 23> run
# pattern moniterd and rejected
VSIM 24> run
# pattern moniterd and rejected
VSIM 25> run
# pattern moniterd and rejected
VSIM 26> run
# pattern moniterd and rejected
VSIM 27> run
VSIM 28> run
VSIM 29> run
# faulty zone
```

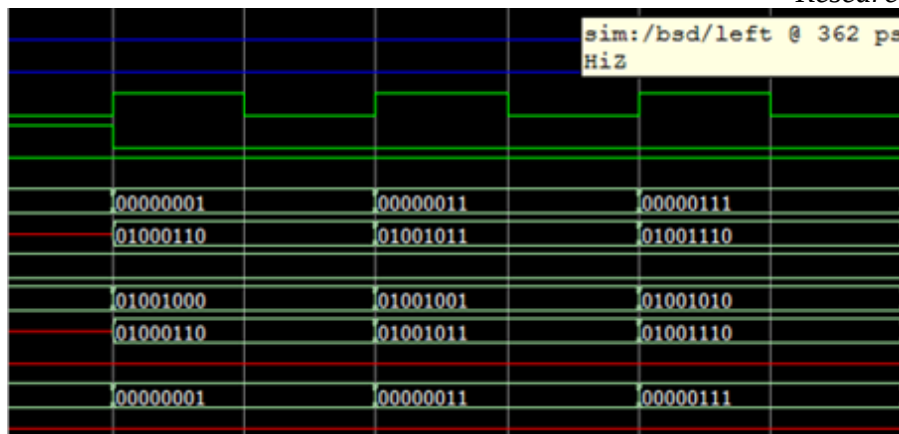



Figure 13. Fault Zone Isolation

Table 1 lists the different parameters used to compute current and proposed systems.

S.No	Parameter	Existing (Router Trace lines Without tiling)	Proposed (Router Trace Lines With tiling)
1	Slice	10	7
2	Slice FF	7	3
3	LUT	19	14
4	Power	40mW	32mW
5	Speed	28ns	19ns

Table 1.

5. Performance Analysis

Based on the implementation results achieved using the Spartan-3 processor, the figure below shows that there is a significant reduction in time and area. Thus making the power consumption and area consumed less with significant increase in speed.

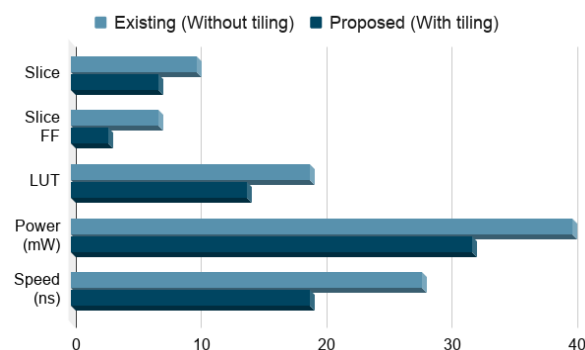


Figure 14. Performance Analysis

When compared to the current method, the proposed algorithm consumes significantly less space.

6. Conclusion

The buffer design is crucial with inexpensive, superior, and energy-efficient on-chip networks since network-on-chip performance is directly associated with router buffer configuration. Proposed a reconfigurable trace data compressor architecture for silicon debug combined with defective zone isolation with several X-bits throughout

signal tracing in re-used buffers in this paper. As demonstrated in the experimental results, the projected X-tracer design, collaborated with novel algorithms to get helpful trace information from polluted signatures of trace, makes it possible to gather the maximum trace data whereas holding a high amount of compression ratio. The proposed tiling eliminates excessive testing and reduces overall slice and LUT debugging requirements. The Comparison between power and speed of existing and proposed system gives useful results. The power is reduced by 20 percentage and the speed is increased by 32 percentage in the proposed error correction method using tiling.

References

1. Rickard Holsmark and Magnus Hgberg, "Modelling and Prototyping of a Network on Chip," Master of Science Thesis, 2002 Electronics.
2. Ali, Muhammad, Michael Welzl, and Martin Zwicknagl . "Networks on chips: scalable interconnects for future systems on chips." 2008 4th European Conference on Circuits and Systems for Communications. IEEE, 2008.
3. Huang, Letian, et al. "ECDR2: Error Corrector and Detector Relocation Router for Network-on-Chip." IEEE Transactions on Computers (2020).
4. Pirretti, Matthew, et al. "Fault tolerant algorithms for network-on-chip interconnect." IEEE computer society annual symposium on VLSI. IEEE, 2004..
5. Cidon, Israel, and Idit Keidar. "Zooming in on Network-on-Chip Architectures." SIROCCO. 2009.
6. Wang, Lu, et al. "A high performance reliable NoC router." Integration 58 (2017): 583-592.
7. Zhou, Xinan. "Performance evaluation of network-on-chip interconnect architectures." (2009).
A. Jantsch and H. Tenhunen. Networks on Chip, Kluwer Academic Publishers, 2003.
8. Dally, William J., and Brian Towles. "Route packets, not wires: on-chip interconnection networks." Proceedings of the 38th annual design automation conference. 2001.
9. Aghaei, Babak, et al. "A new bist-based test approach with the fault location capability for communication channels in network-on-chip." Journal of Electronic Testing 33.4 (2017): 501-513.
10. Ko, Ho Fai, and Nicola Nicolici. "Automated trace signals identification and state restoration for improving observability in post-silicon validation." 2008 Design, Automation and Test in Europe. IEEE, 2008.
11. Liu, Xiao, and Qiang Xu. "Trace signal selection for visibility enhancement in post-silicon validation." 2009 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2009.
12. Yang, Joon-Sung, and Nur A. Touba. "Automated selection of signals to observe for efficient silicon debug." 2009 27th IEEE VLSI Test Symposium. IEEE, 2009.
13. Liu, Xiao, and Qiang Xu. "On multiplexed signal tracing for post-silicon debug." 2011 Design, Automation & Test in Europe. IEEE, 2011.
14. Anis, Ehab, and Nicola Nicolici. "On using lossless compression of debug data in embedded logic analysis." 2007 IEEE International Test Conference. IEEE, 2007.
15. Prabhakar, Sandesh, Rajamani Sethuram, and Michael S. Hsiao. "Trace buffer-based silicon debug with lossless compression." 2011 24th International Conference on VLSI Design. IEEE, 2011.
16. Yang, Joon-Sung, and Nur A. Touba. "Enhancing silicon debug via periodic monitoring." 2008 IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems. IEEE, 2008.
17. Anis, Ehab, and Nicola Nicolici. "Low cost debug architecture using lossy compression for silicon debug." 2007 Design, Automation & Test in Europe Conference & Exhibition. IEEE, 2007.
18. Lu, Zhi, et al. "The fault-tolerant NoC techniques with FPGA." 2015 IEEE International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD). IEEE, 2015.
19. Khichar, Jyoti, and Sudhanshu Choudhary. "Fault aware adaptive routing algorithm for mesh based NoCs." 2017 International Conference on Inventive Computing and Informatics (ICICI). IEEE, 2017.
20. Wachter, Eduardo, et al. "A layered approach for fault tolerant NoC-based MPSoCs—Special session: Dependable MPSoCs." 2016 17th Latin-American Test Symposium (LATS). IEEE, 2016.
21. Št'áva, Martin. "Efficient Error Recovery Scheme in Fault-tolerant NoC Architectures." 2019 IEEE 22nd International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS). IEEE, 2019.
22. Kerkhoff, Hans G., O. Kuiken, and Xiao Zhang. "Increasing SoC Dependability via Known Good Tile NoC Testing." IEEE Intern. Conf. on Dependable Systems and Networks (DSN08). 2008.
23. Venishetti, Sandeep K., Ali Akoglu, and Rahul Kalra. "Hierarchical built-in self-testing and fpga based healing methodology for system-on-a-chip." Second NASA/ESA Conference on Adaptive Hardware and Systems (AHS 2007). IEEE, 2007.
24. Yuan, Feng, Xiao Liu, and Qiang Xu. "X-tracer: a reconfigurable X-tolerant trace compressor for silicon debug." Proceedings of the 49th Annual Design Automation Conference. 2012.