A Method to Detect and Avoid Hardware Trojan for Network-on-Chip Architecture based on Error Correction Code and Junction Router (ECCJR)

Hafiz Ali Hamza Gondal[1]*, Sajida Fayyaz[2], Arooj Aftab[3], Saira Nokhaiz[4], Muhammad Bilal Arshad[5], Waqas Saleem[6]
The University of Lahore, Sargodha, Department of Computer Sciences, Sargodha, Pakistan[1, 2, 3, 4, 5]
University of Engineering and Technology Taxila, Department of Computer Engineering Taxila, Pakistan[6]

***Abstract*—Modern technologies has changed our life, such as everywhere computing communication and internet. Number of transistors increasing in a system day by day and this trend will continue further. The wire connection is easily breakable and not a reliable technology in field of networks. In conventional network dedicated wired path is used among the intellectual property (IP) core for the purpose of communication and due to this wired connection network is not reliable and not scalable. Network-on-Chip Architecture was introduced to solve these problems and gave notable improvements over conventional bus and crossbar communication architectures. Many companies prefer third party vendors for the development of their design in order to reduce the cost. It gives advantage but due to the access of design anyone can do changes at any stage of development cycle. This type of malicious modification of hardware during design or fabrication process is known as Hardware Trojan (HT). It can change the functional behavior of a system or may leak the secret information of critical application which results in degradation of system performance. The proposed research is based on combination of Error Correcting Code and Junction router to detect and avoid HT which can be used for reliable communication in NoC architecture. Simulation results showed good performance of proposed algorithm in term of Packet Latency and Reliability.**

Keywords*—Hardware trojan; network-on-chip; intellectual property; error correcting code; junction router

I. Introduction

In a conventional network of System on Chip, dedicated wired path is used among the Intellectual property (IP) core for communication. This type of communication medium is not reliable not scalable and has a lot limitations. To solve this problem a new architecture named Network on Chip is introduced which is network based on communication subsystem on an integrated circuit. When we put multiple processors, peripherals, memories on to a chip to build a networked multiprocessor system on Chip (MPSoC) then this is known as NoC [1]. A traditional NoC consists of Processing Element (PE), Network Interface (NI), Link and a router as shown in Fig. 1. NI is a communication interface between IP core because it helps to packetize and de-packetize the data while links are used for travelling of packets in network. Packets are moved in a given network on the basis of routing algorithms.

NoC has three major types of routing algorithm one is knows as deterministic second is adaptive and third is oblivious. In deterministic routing path between two IP cores are already defined and during communication no one can use this path [2]. In adaptive routing different path can be selected by considering the state of network. While in case of oblivious different path are available but packet don’t consider the state of network which can be result in deadlock state. The router architecture depends upon the design of NoC and a Topology which is very important feature. NoC supports many network topologies like mesh, torus, butterfly, polygon, mesh, tree, star and ring. Moreover NoC also helps to overcome the packet delays, reliability and cost which are primary problems in MPSoC. Many organizations prefer third party vendors for fabrication of their designs in order to reduce the cost but it can be harmful. During the development cycle of hardware malicious modification is possible at any stage. This activity is known as Hardware Trojan (HT) [4]. The addition of HT in hardware can change the behavior of system, may affect the performance or even can leak the secret information [18]. For example a 32bit vector data can be inverted by adding a sample NOT gate due to which whole output can be changed. Similarly if the connection is established maliciously between output and a Wi-Fi module during any phase of development cycle then confidential data will be compromised [3].

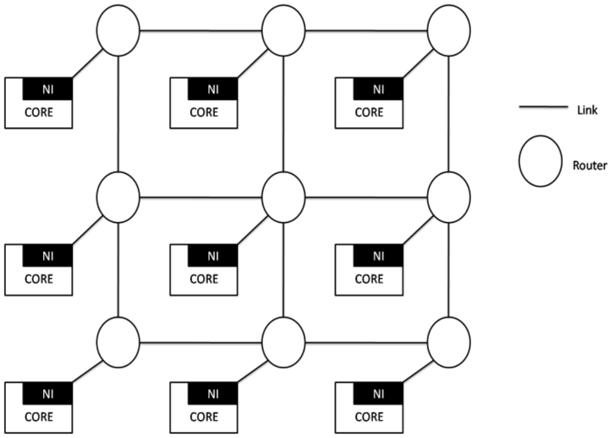


Fig. 1. Network on Chip Architecture.

*Corresponding Author

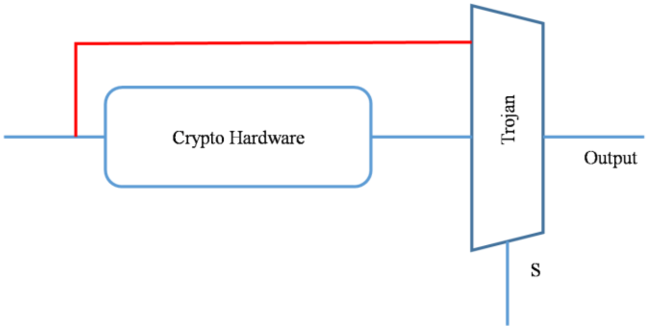


Fig. 2. Hardware Trojan Model [4].

The entire activity of the Trojan is determined when it is triggered and it is termed as its payload [16]. A simple Trojan is given in Fig. 2 in which the Trojan comprises of a single multiplexer. An encrypted data is sent as an output during its normal functioning. When Trojan is active the plain text bypasses the crypto-hardware and is sent on output.

HT is very harmful as it reduces the overall performance of a system. To solve this problem this research proposed a method to detect and avoid HT efficiently for reliable communication in NoC architecture.

II. Literature Review

In this section detail study of Trojan, comparison with faults and different detection methods is presented. After this section proposed methodology and experimental results are discussed in later section.

A. *Trojan vs Fault*

Trojan is intentionally addition of hardware in a given design at any stage of development cycle. While in case of faults it can be possible due to any reason [4]. Table I describes more details about the comparison of Trojan and Faults.

1) Activation mechanisam of trojan: Activation mechanism of a Trojan can be categorized into two ways internally triggered and externally triggered [5].

a) Internally Triggered: In this case Trojan is activated without the disturbance of external environment and based on an internal event. This event may be physical or timer or based on the value of different parameters like temperature, power etc.

b) Externally Triggered: There is need of external input for externally triggered Trojan to start its malicious activity. Any of the component that interact with the target device may be an external component trigger.

2) Trojan detection method: Many Researchers proposed different methods to detect the HT. The classification of detection methods are given in Fig. 3.

In destructive approach reverse engineering of IC is done and functionality of given design is compared with golden IC [6]. In non destructive logic is tested by using different input combinations. While different parameters like power, temperature can be used in side channel analysis for Trojan detection.

TABLE I. Comparison of Trojan and Fault [3]

	Fault	**Hardware Trojan**
Activation	Usually at known functional state	Arbitrary combination/ sequence of internal circuit states(digital/analog)
Insertion Agent	Accidental	Intentionally inserted by an adversary during design or fabrication
Manifestation	Functional	Functional/ Parametric failure or information leakage

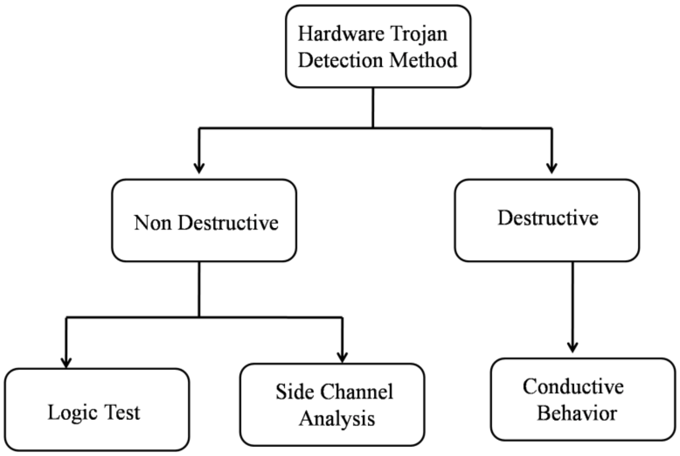


Fig. 3. Trojan Detection Classification [3].

Idea of encryption scheme in NI was introduced by Sajesh and Kappor for secure communication in NoC architecture. In this method when a secure core wanted to communicate with other core data was encrypted first then it was sent in a given network [7]. To increase the security of NoC [8] Gebotys introduced the method based on security wrapper in which data was encrypted, decrypted and message authentication code (MAC) was generated by security wrapper [8]. A lot of work is done for avoiding the DoS Trojan model like a method proposed in [9] consists of two block one is DoS security block and second is ED security block. In this method researcher used Linear Feedback shift register and burs error control unit to detect the Trojan.

In [13] Basim Shanyour Proposed standard cell placement method for the detection of Low power Trojan. This method proves low area overhead. A data packet authentication scheme was also presented in [14] which is based on bandwidth awareness. In this method packets is assigned tags at source routers but aggressively authenticated thorough out the packet transmission. The researcher’s results showed 36% saving of bandwidth and 56% low area over head. In [15] researchers gave a method for the detection and avoidance of snooping data. In this method they used snooping invalidator module at NI to discard the duplication of packets. Their experimental results showed the 48.4% increase of performance. For the avoidance of Denial of Services attack by misrouting of packet is given in [17]. In this method secure routing is used in such a way that if router is directed in wrong direction the neighbor router will automatically detect it. This method is quite efficient with only 0.4% area overhead. A secure Wireless NoC architecture is introduced in [19].

Researchers worked on three Trojan related to snooping, denial of services and eaves dropping. This algorithm is based on message authentication code (MAC) and observation of channel capacity.

3) *Error Correcting Codes (ECC)*

a) Cross Talk Avoidance Double Error Correction CADEC: This is combination of Hamming Code and Dual Add parity (DAP) technique. First of all hamming is implemented on 32 bit input data and 38 bit output data is generated with 6 parity bits. Then DAP is implemented which gives 77 bits including parity bit. In decoding two copies of data are separated and parity of each copy is calculated and different decisions are made by comparison as shown in Fig. 4 [10].

b) Joint Cross Talk Avoidance Triple Error Correction (JTEC): This algorithm is similar to CADEC but it gives triple error correction. The steps of encoding are similar to CADEC but decoding steps are little bit different. In decoding after the separation of copies Syndrome is calculated and all decisions are made on the basis of comparison of syndrome. The detail of decoding steps is given in Fig. 5 [10].

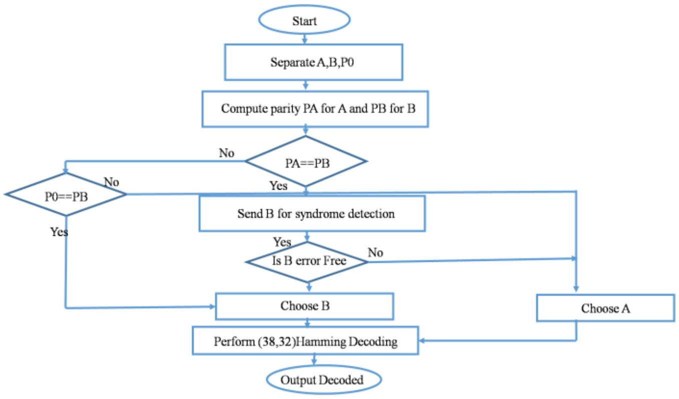

Start
Separate A,B,P0
Compute parity PA for A and PB for B
No
PA==PB
No
P0==PB
Yes
Send B for syndrome detection
Yes
Yes
Is B error Free
No
Choose B
Choose A
Perform (38,32)Hamming Decoding
Output Decoded

Fig. 4. CADEC Decoding Algorithm [10].

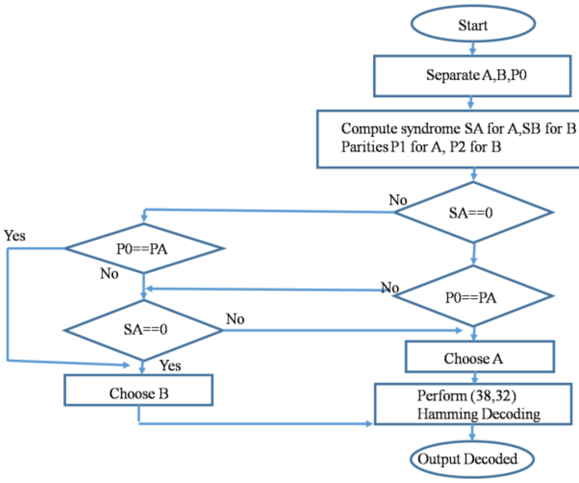

Start
Separate A,B,P0
Compute syndrome SA for A,SB for B and Parities P1 for A, P2 for B
No
SA==0
Yes
P0==PA
No
No
P0==PA
No
SA==0
Yes
Choose A
Choose B
Perform (38,32) Hamming Decoding
Output Decoded

Fig. 5. JTEC Decoding Algorithm [10].

III. Research Methodology

A. *Threat Model*

For simulation purpose mesh topology of 6x6 was selected on which different applications were running. There is no restriction of application type it can be simple or complex. For distinction of packets, an application ID field was introduced in flit. The mesh network is shown in Fig. 6.

B. *Activation Mechanism of Trojan*

A combinational circuit's output was used as a medium for activation of Trojan model as shown in Fig. 7. In this four inputs are given which are randomly selected data bits form packets. The motivation for adopting this Trojan model was taken from Yu, Q. and Frey [11].

C. *Payload of HT*

Payload is an activity performed by HT after it is activated. As discussed in literature review payload can be denial of resource or latency in packets or corrupted data. The corrupted data packets were taken as a payload of thread model.

D. *Proposed Algorithm*

Everything was going perfectly as data packets of application was reaching at their actual destination but after some time the data packets of application start corrupting. For the solution of this problem error correction code ECC named as Joint cross talk avoidance triple error correction JTEC is used. A threshold vale of 20% for corrupted packets was selected. In other words we can say if 20% of data packets are corrupted then JTEC will be enabled. JTEC has capability of 3bit error correction. Some routers were selected as junction routers on which ECC decoder and encoder will be implemented. Transmitter (TX) has ECC encoder which will add redundancy bits with data while at receiver (RX) side syndrome is calculated to detect which links are controlled by HT. JTEC decoder is implemented on inputs ports of junction router while JTEC encoder implemented on output ports. Selection of junction routers for 6x6 mesh topology is given in Fig. 8. Black boxes are those routers which are selected as Junction Router.

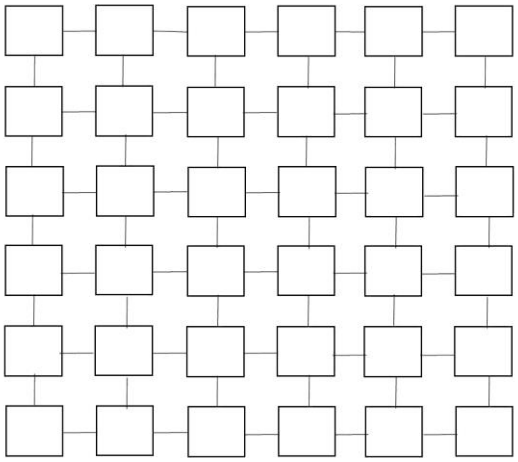


Fig. 6. Mesh Network of 6x6.

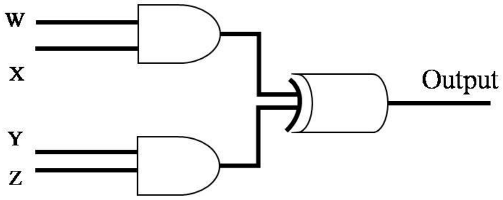


Fig. 7. Combinational Circuit for Activation of HT [11].

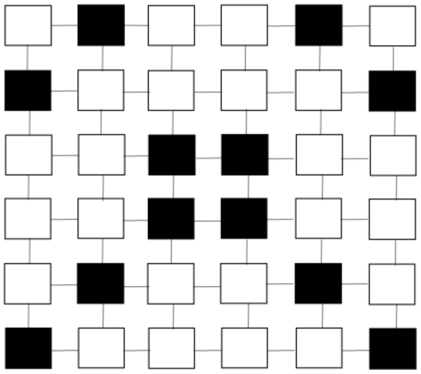


Fig. 8. Selection of Junction Router for 6x6 Mesh Network.

The selection of junction router can be find by equation 1.

JC = n / 4 for n > 2 (1)

Where

JC = Junction Router

n = Size of mesh

For packet movements modified XY dimension order routing was used in such a way that after 2 hops packet will pass through junction router. Whenever corrupted packets entered in a Junction router, JTEC corrects errors and sets out in network from output ports. This algorithm will help to detect and avoid 3 bit error. The flow chart of proposed method is shown in Fig. 9.

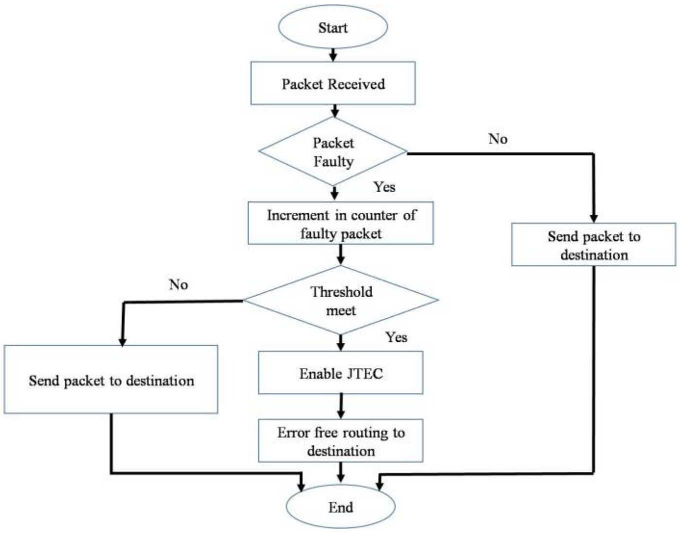

Start
Packet Received
Packet Faulty
No
Yes
Increment in counter of faulty packet
Send packet to destination
No
Threshold meet
Yes
Send packet to destination
Enable JTEC
Error free routing to destination
End

Fig. 9. Flow Chart of Proposed Method.

E. Routing Algorithm

Oblivious routing algorithm is used in proposed method for the movements of packets in a given network. In oblivious routing algorithm the condition of network is considered before the sending of packets from the output ports to avoid the problem of traffic congestion. Error correcting code JTEC is used in Junction router which has capability of 3 bit error correction. Example of proposed algorithm is shown in Fig. 10 in which packet path from router 7 to 29 is highlighted.

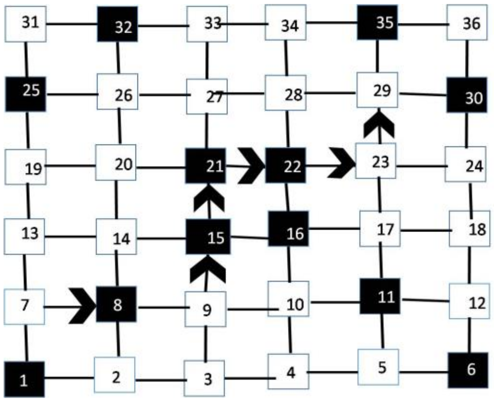


Fig. 10. Packet Path from Router 7 to 29 Proposed Method.

IV. Experimental Results

For the simulation results of proposed algorithm Booksim 2.0 was used. This simulator allows us to use different traffic pattern for analysis. Like Uniform, Tornado and neighbor. In Uniform traffic source S sends an equal amount of packet to each destination d. For traffic pattern tornado and neighbor, equations 2 and 3 are given, respectively [12].

$d_x = S_x + [k/2] - 1 \bmod k$ (2)

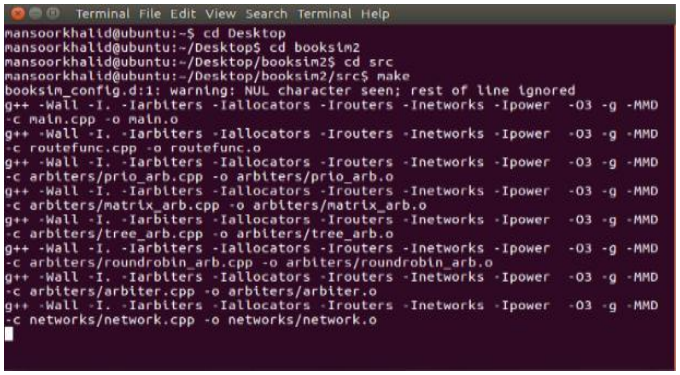
$d_x = S_x + 1 \bmod k$ (3)

Where dx is destination, Sx is source and k is Total numbers of routers. Running of simulation is shown in Fig. 11.

Latency is defined as time taken by the packet to reach from source to destination. It is very important parameter to be considered for performance evaluation.

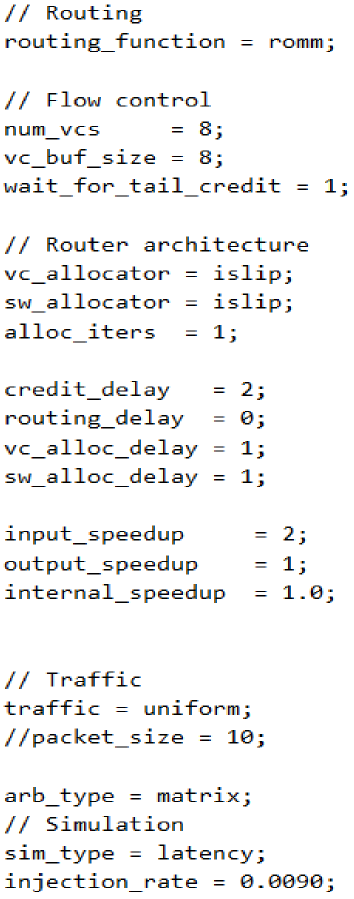
In booksim a file "mesh_lat" is available in which configuration parameters can be set as shown in Fig. 12. In proposed algorithm default values of parameters were used only changes were made to "routing_function" for routing packets, "traffic" for different types of traffic to analyze the behavior of proposed algorithm. The "injection_rate" varied from 0.01 to 0.1 per cycles and calculated different latencies for performance evaluation. Fig. 13 shows the packet latency of packet for uniform traffic pattern when HT is active and JTEC is enabled. Fig. 14 shows the Network Latency when HT is active and JTECT is enabled for tornado traffic. Fig. 15 Show Average hop count for uniform traffic pattern when Trojan is active.

From Fig. 13 and 14 it is clearly shown when HT is active latency of packets is increased as packets are unable to reach their destination. When JTEC is enabled no data is corrupted and destination fields will remain same as it was on the time of packet generation. Since after implementation of proposed algorithm packets will reach at their destination on time so latency will be reduced.



```
mansoorkhalid@ubuntu:~$ cd Desktop
mansoorkhalid@ubuntu:~/Desktop$ cd booksim2
mansoorkhalid@ubuntu:~/Desktop/booksim2$ cd src
mansoorkhalid@ubuntu:~/Desktop/booksim2/src$ make
booksim_config.d:1: warning: NUL character seen; rest of line ignored
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c main.cpp -o main.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c routefunc.cpp -o routefunc.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c arbiters/prio_arb.cpp -o arbiters/prio_arb.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c arbiters/matrix_arb.cpp -o arbiters/matrix_arb.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c arbiters/tree_arb.cpp -o arbiters/tree_arb.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c arbiters/roundrobin_arb.cpp -o arbiters/roundrobin_arb.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c arbiters/arbiter.cpp -o arbiters/arbiter.o
g++ -Wall -I. -Iarbiters -Iallocators -Irouters -Inetworks -Ipower  -O3 -g -MMD
-c networks/network.cpp -o networks/network.o
```

Fig. 11. BookSim 2.0 Simulation Running.



```
// Routing
routing_function = romm;

// Flow control
num_vcs     = 8;
vc_buf_size = 8;
wait_for_tail_credit = 1;

// Router architecture
vc_allocator = islip;
sw_allocator = islip;
alloc_iters  = 1;

credit_delay   = 2;
routing_delay  = 0;
vc_alloc_delay = 1;
sw_alloc_delay = 1;

input_speedup     = 2;
output_speedup    = 1;
internal_speedup  = 1.0;


// Traffic
traffic = uniform;
//packet_size = 10;

arb_type = matrix;
// Simulation
sim_type = latency;
injection_rate = 0.0090;
```

Fig. 12. Configuration Parameters of Simulation.

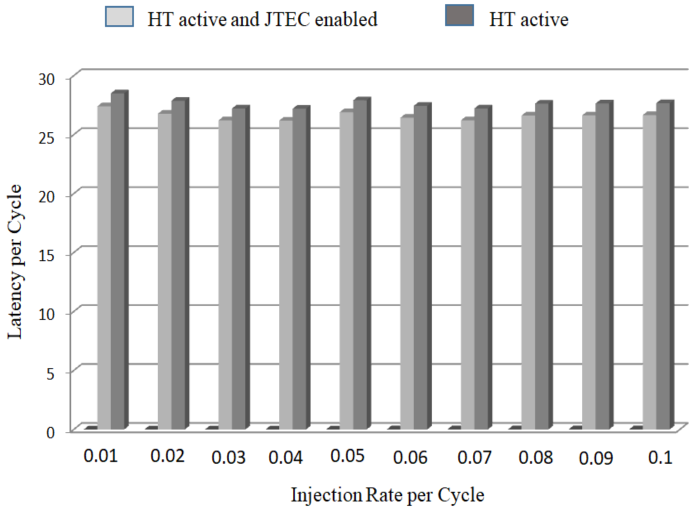


Fig. 13. Packet Latency Comparison for Uniform Traffic when HT Active and JTECT is Enabled.

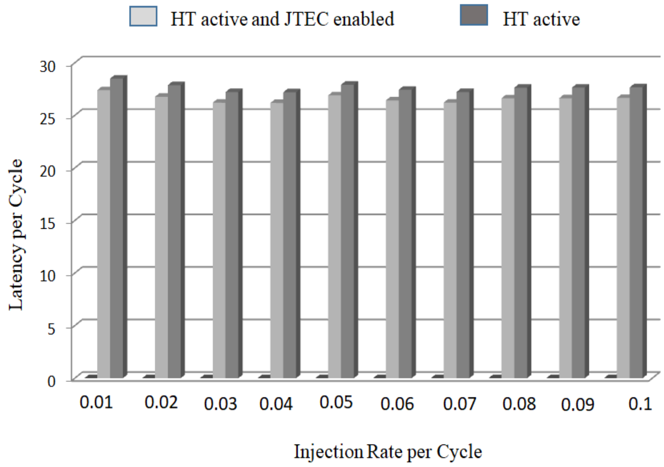


Fig. 14. Network Latency Comparison for Tornado Traffic when HT Active and JTECT is Enabled.

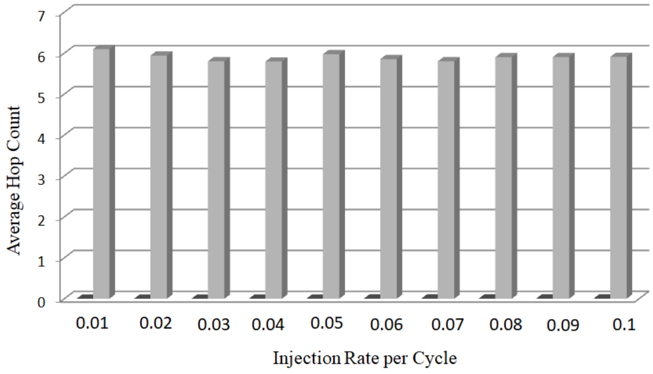


Fig. 15. Average Hop Count for uniform Traffic Pattern when Trojan is Active.

V. Conclusion

This research includes the detection and avoidance of Hardware Trojan in Network-on-Chip Architecture. Proposed algorithm is basically the combination of error correcting code which is Joint cross talk avoidance triple error correction JTEC and junction router (ECCJR). JTEC having the capability of 3 bit error correction and it is implemented only on junction routers. The Trojan model is based on the output of combinational circuit and payload of hardware Trojan is packets corruption. The selected value for the threshold of faulty packets was 20%. Whenever it meets the threshold value proposed algorithm will be started. After the implementation of proposed algorithm ECCJR no packet was corrupted and reliability reaches to 100%. In future different routing algorithms can be used to reduce the latency of packets per cycle which will increase the performance of whole system. Different error correction code can also be used in router other than Junction routers for error free communication of a network.

References

[1] Engel, M., & Spinczyk, O. (2009, January). A radical approach to network-on-chip operating Fsystems. In System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on(pp. 1-10). IEEE.

[2] Achballah, A.B. and Saoud, S.B., 2013. A survey of network-on-chip tools. arXiv preprint arXiv:1312.2976.

[3] Bhunia, S., Hsiao, M.S., Banga, M. and Narasimhan, S., 2014. Hardware Trojan attacks: threat analysis and countermeasures. Proceedings of the IEEE, 102(8), pp.1229-1247.

[4] Rajendran, J., Gavas, E., Jimenez, J., Padman, V. and Karri, R., 2010, May. Towards a comprehensive and systematic classification of hardware trojans. In Proceedings of 2010 IEEE International Symposium on Circuits and Systems (pp. 1871-1874). IEEE.

[5] Wolff, F., Papachristou, C., Bhunia, S. and Chakraborty, R.S., 2008, March. Towards Trojan-free trusted ICs: Problem analysis and detection scheme. In Proceedings of the conference on Design, automation and test in Europe (pp. 1362-1365). ACM.

[6] Agrawal, D., Baktir, S., Karakoyunlu, D., Rohatgi, P. and Sunar, B., 2007, May. Trojan detection using IC fingerprinting. In 2007 IEEE Symposium on Security and Privacy (SP'07) (pp. 296-310). IEEE.

[7] Sajeesh, K. and Kapoor, H.K., 2011, December. An authenticated encryption based security framework for noc architectures. In 2011 International Symposium on Electronic System Design (pp. 134-139). IEEE.

[8] Gebotys, C.H. and Zhang, Y., 2003, October. Security wrappers and power analysis for SoC technology. In First IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and Systems Synthesis (IEEE Cat. No. 03TH8721) (pp. 162-167). IEEE.

[9] Vashist, A., Keats, A., Dinakarrao, S.M.P. and Ganguly, A., 2019, July. Securing a wireless network-on-chip against jamming based denial-of-service attacks. In 2019 IEEE Computer Society Annual Symposium on VLSI (ISVLSI) (pp. 320-325). IEEE.

[10] Patel, K.N. and Markov, I.L., 2003, April. Error-correction and crosstalk avoidance in DSM busses. In Proceedings of the 2003 international workshop on System-level interconnect prediction (pp. 9-14). ACM.

[11] Yu, Q. and Frey, J., 2013, October. Exploiting error control approaches for hardware trojans on network-on-chip links. In 2013 IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFTS) (pp. 266-271). IEEE.

[12] Jiang, N., Michelogiannakis, G., Becker, D., Towles, B. and Dally, W.J., 2010. Booksim 2.0 user’s guide. Standford University.

[13] Shanyour, B. and Tragoudas, S., 2018. Detection of Low Power Trojans in Standard Cell Designs using Built-in Current Sensors. In 2018 IEEE International Test Conference (ITC) (pp. 1-10). IEEE.

[14] Hussain, M. and Guo, H., 2018, October. A Bandwidth-Aware Authentication Scheme for Packet-Integrity Attack Detection on Trojan Infected NoC. In 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC) (pp. 201-206). IEEE.

[15] Raparti, V.Y. and Pasricha, S., 2019, June. Lightweight Mitigation of Hardware Trojan Attacks in NoC-based Manycore Computing. In 2019 56th ACM/IEEE Design Automation Conference (DAC) (pp. 1-6). IEEE.

[16] Daoud, L., 2018, August. Secure network-on-chip architectures for mpsoc: Overview and challenges. In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 542-543). IEEE.

[17] Daoud, L. and Rafla, N., 2018, August. Routing aware and runtime detection for infected network-on-chip routers. In 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS) (pp. 775-778). IEEE.

[18] Wang, J., Guo, S., Chen, Z. and Zhang, T., 2019. A Benchmark Suite of Hardware Trojans for On-Chip Networks. IEEE Access, 7, pp.102002-102009.

[19] Lebiednik, B., Abadal, S., Kwon, H. and Krishna, T., 2018, October. Architecting a secure wireless network-on-chip. In 2018 Twelfth IEEE/ACM International Symposium on Networks-on-Chip (NOCS) (pp. 1-8). IEEE.