# Dead Flit Attack on NoC by Hardware Trojan and its Impact Analysis

Mohammad Humam Khan[†], Ruchika Gupta [‡], John Jose[‡], and Sukumar Nandi[‡]
[†]Department of Mathematics, [‡]Department of Computer Science & Engineering
Indian Institute of Technology Guwahati, India

## ABSTRACT

With the advancement in VLSI technology, Tiled Chip Multicore Processors (TCMPs) with packet switched Network-on-Chip (NoC) have emerged as the most popular design choice for compute and data intensive embedded and parallel systems. Tight time-to-market constraints and budget limitations have forced the designers to explore the possibilities of using several third party Intellectual Property (IP) cores. Use of such unsecured inexpensive third party IPs may pose severe security challenges that are not detected at manufacturing and testing phases. Recent research shows that manipulation of the NoC packet content by Hardware Trojan (HT) has the potential to disrupt the on-chip communication resulting in application level stalling. We model a novel HT that alters the common prefix field of NoC packets leading to the creation of dead flits in router buffers. We introduce two variants of this proposed HT: one that modifies head flit to body flit and another one that modifies the body flit to head flit. We analyze the HT impact at core level, cache level, and NoC level. The experimental analysis on a 16-core TCMP demonstrates that the proposed HT significantly reduces IPC, increases the average cache miss penalty, and increases the average buffer occupancy of selected packets in NoC.

## CCS CONCEPTS

• **Computer systems organization** → Interconnection architectures; • **Security and privacy** → Hardware security.

## KEYWORDS

Network-On-Chips, Hardware Trojans, Secured NoC Design, Control Field Manipulation, Dead Flit

**ACM Reference Format:**
Mohammad Humam Khan[†], Ruchika Gupta [‡], John Jose[‡], and Sukumar Nandi[‡], [†]Department of Mathematics, [‡]Department of Computer Science & Engineering, Indian Institute of Technology Guwahati, India . 2021. Dead Flit Attack on NoC by Hardware Trojan and its Impact Analysis. In *14th edition of International Workshop on Network on Chip Architectures (NoCArc'21), October 18–22, 2021, Virtual Event, Greece.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3477231.3490425


*NoCArc'21, October 18–22, 2021, Virtual Event, Greece*

ACM ISBN 978-1-4503-8711-8/21/10…$15.00
https://doi.org/10.1145/3477231.3490425

## 1 INTRODUCTION

Embedded System devices and Internet-of-Things gadgets employing third-party IP to match time-to-market constraints has become a common practice for more than a decade now. Functional security of these devices can be compromised when potentially untrusted IPs are involved in the design [8] [16]. Malicious circuits like HTs implanted into the design are one such major security threat in modern SoCs [3] [12]. The dormant and untraceable nature of HTs during verification and testing phases make them the most popular choice for attackers [16] [13]. They can remain dormant until a particular event activates them. Some HTs are triggered for a short span of time that makes it even harder to trace their presence [17] [20]. Information leakage, unauthorized access, delay-of-service, and denial-of-service are some of the extensive security attacks impacted by HTs.

Network on Chip (NoC) provides on-chip communication infrastructure for TCMP [2]. Figure 1 shows a typical TCMP with an underlying 4x4 mesh NoC connecting 16 homogeneous tiles. In modern TCMPs like Intel Xeon Phi and AMD Ryzen Threadripper, each tile has a private L1 cache and a slice of distributed shared L2 cache [7]. Due to bandwidth limitations in NoC, packets are further divided into smaller flow control units called flits. A packet consists of a head flit carrying control information, a number of body flits carrying the data/payload and a tail flit marking the end of the packet. Routing decisions are done on the head flit and other flits of the packet follow the same route using wormhole switching. Apart from the flit channel that carries the data between adjacent pair of routers, we define few additional control lines called as common prefixes that are attached parallel to the flit channel. The common prefixes represent Flit Type (FT) and VC id number (VCID) to ensure a hassle free wormhole switching.

Being the communication backbone of TCMP, attackers exploit the positional advantage of NoC for mounting HTs. One of the most common type of attacks deployed by HT is Denial of Service (DoS) preventing legitimate cores from accessing required data and services [18]. HTs can be localized by monitoring unusual traffic behaviour against DoS attack [6]. Security zones managed by a centralized security manager can protect sensitive information stealing by malicious agents [19]. Data scrambling, packet authentication using sophisticated ciphers, and node obfuscation methods are proposed to protect data from compromised NoC [1]. Bit shuffling and error correcting codes are helpful in handling HTs that corrupt data [5] [11]. Trojan aware routing can bypass an HT infected router using cost effective shielding [14]. Machine Learning algorithms can also be used to detect DoS attack in NoC [21].

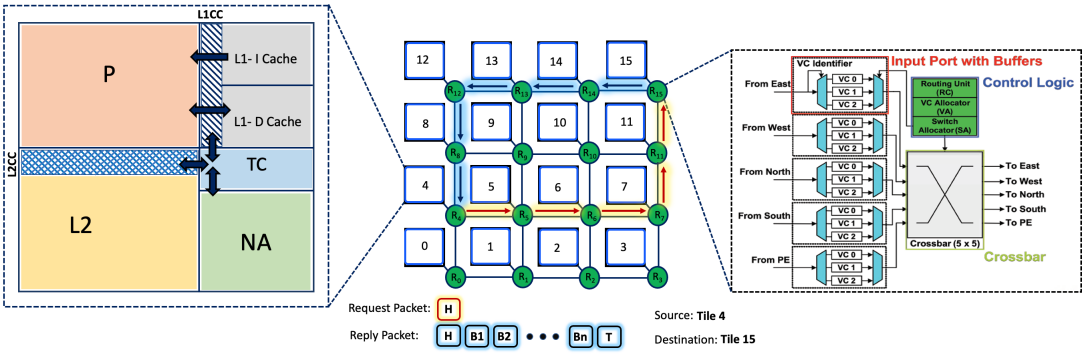



**Figure 1: Internal architecture of tile and router in a 4×4 TCMP. Tile consists of Processor (P), L1 and L2 Caches, Cache Controllers (L1CC, L2CC), Tile Controller (TC), and Network Adapter (NA). For communication between tiles 4 and 15 using XY routing, red and blue arrows indicate the path of the request packet and corresponding reply packet, respectively.**

The possibility of an attack on head flit is explored recently where HT manipulates destination id of cache miss request packets to create DoS attacks and application level stalling [10]. The control fields present in the head flits are used by intermediate routers for taking buffering and routing decisions and hence they are stored in an unencrypted format despite the fact that data encryption can be done in the body and tail flits that carry payload. We model an HT that maliciously modify common prefixes leading to alteration of flit type field. To the best of our knowledge no previous work has explored such type of HTs and conducted an in-depth impact analysis at processor, cache, and network level. We make following contributions in this paper:

(1) We propose a novel HT that can modify the common prefix of flits leading to formation of dead flits in NoC router buffers.
(2) We introduce two variants of this proposed HT: one that modifies head flit to body flit and another one that modifies the body flit to head flit.
(3) We conduct impact analysis of the proposed HT at processor, cache, and NoC levels in a 4x4 TCMP.

## 2 BASELINE TCMP ARCHITECTURE

We consider a 16-tile TCMP organized in a 4x4 mesh topology as the baseline architecture as shown in Figure 1. Each tile consists of Processor, L1 and L2 caches, Tile Controller and a Network Adapter. L2 cache is shared and distributed among all the 16 tiles in a sequential fashion using SNUCA mapping [7]. NoC router consists of Input Port buffers, Routing Unit, Virtual Channel (VC) Allocator, Switch Allocator, and a Crossbar. NoC traffic consists of cache miss requests, cache miss responses, write updates, and coherence packets. Whenever there is an L1 cache miss, the L1CC forwards the request to TC which computes the L2 tile number and forwards it to NA in order to contact the remote tile. NA creates and forwards the packet to the router attached to the source tile. A miss request packet consists of a single head flit (H), whereas a response packet consists of a head flit followed by multiple body flits and a single tail flit ($H, B_1, B_2, ..., B_n, T$). NoC follows wormhole switching where body and tail flits follow the head flit path. Router performs route computation based on routing technique and forwards the flits to appropriate output ports.

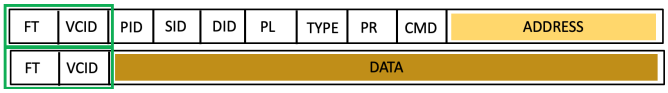



**Figure 2: Head and Body Flit Format with Common Prefix**

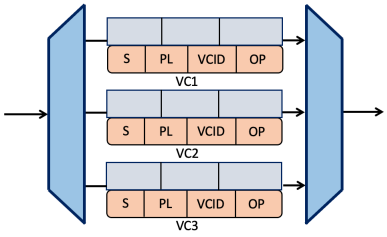



**Figure 3: Internal Architecture of Input Port Buffers**

In this paper we consider few additional control fields called as common prefixes that are attached parallel to the flit channel to facilitate wormhole switching. Figure 2 shows the format of head and body flits along with the common prefixes. All the flits of a packet have a common prefix consisting of two fields, namely, FT and VCID. FT distinguishes the type of the flit viz. head, body, or tail. VCID is assigned after head flit wins the VC allocation and specifies the VC number the head flit must occupy upon reaching the downstream router. The same VCID is then inherited by all the subsequent flits of the same packet. PID uniquely identifies a packet in the network. The address of the source tile and the destination tile is given by SID and DID, respectively while PL indicates how many non-head flits are there in a particular packet. Since NoC carries all category of packets exchanged between tiles, the TYPE field specifies the type of the message a packet carries. The PR field contains the priority of a packet, used for arbitration during port conflicts. CMD carries additional meta data about a packet and the ADDRESS field carries the physical address communicated between various memory levels.

Figure 3 shows the internal architecture of the input port of a router consisting of 3 VCs. Each VC forms a FIFO queue and can accommodate a maximum of 3 flits at any given point in time. Every VC has a control buffer attached to it that consists of Status (S),

Packet Length (PL), Virtual Channel Identifier (VCID), and Output Port (OP). When a flit reaches a router, the input port de-multiplexer extracts the VCID from the common prefix of the incoming flit and buffers the flit into the assigned VC. If it is a head flit, the PL field of the flit is copied to the PL of the control buffer, and S is set to Active or Busy after asserting that it is Idle. The route computation unit extracts DID from the head flit to compute the next outgoing port and OP is updated accordingly. Hence, even if the head flit advances to the next router, other flits still inherit the OP and move forward thereby facilitating wormhole routing. When the tail flit leaves the router, all the fields of control buffer are reset. For every head flit, the VC Allocator allocates a new VC based on the VC availability at the next downstream router. VC availability of downstream routers is updated every cycle by credit exchange between the neighboring routers. PL gets decremented for every non-head flit arriving in the VC and PL counter reaching zero indicates end of a packet and S is again set to Idle.

## 3 THREAT MODEL

In this work, we define an HT that can manipulate the common prefix of the flits. The proposed HT is a small circuit that has access to flits stored in VCs of a given router. In our threat model, HT is always active, however an attack is triggered randomly with a probability *p*. When a flit enters the infected router, our proposed HT modifies the FT field of the common prefix before routing and VC allocation operations are carried out. We define two possible HT variants, one that modifies a head flit to body flit (HT-HB) and the another one that modifies body flit to head flit (HT-BH). Since every packet has a head flit, HT-HB can act on any packet passing through the infected router, however, HT-BH can impact only packets with body flits, mostly cache miss reply and write back packets.

After the flit gets buffered in its assigned VC, the FT field is checked to perform further actions accordingly. If it is a head flit, the State (S) of control buffer is changed from idle to active. Afterwards the routing unit computes the output port for the packet using the DID field and subsequently the OP field in control buffer is updated. Based on the availability of VC in the downstream router, VCID field is updated. This VCID assigned to the head flit is used by subsequent body and tail flits to perform wormhole switching. If the incoming flit is a body or tail flit, neither route computation nor VCID allocation takes place, instead it follows the OP and VCID already assigned for the preceding head flit.

We first consider the case when HT modifies a head flit to body flit. Since the FT field indicates that it is a body flit no route computation is done leaving OP, VCID, and PL fields in the control buffer unset. As a direct consequence of this, the control buffer lookup for OP field in the Switch Arbitration phase returns false due to which this particular VC never competes for the crossbar. The flit never moves out of the buffer of the HT router and remains there forever. If the victim packet is a cache miss response packet, the subsequent body and tail flits queue up in the input-port buffer behind the infected head flit and never proceed further. We call them *Dead Flits*. The dead flits consume the VC forever and consequently the mentioned VC can never be assigned to any other future packets. As HT continues its malicious activity, it creates more dead flits in other VCs of the router.

The second variant is the the case when HT modifies a body flit to head flit. Such modification initiates actions that are supposed to be performed on a head flit arrival. S field of the control buffer is checked for its status and changed to active if idle. However, the preceding original head flit has made it active already. In due course any further movement of the infected and other flits following the same packet gets halted. This eventually leads to a situation similar to the dead flits, consuming network resources and propagating back-pressure in the upstream routers. Since body flits are present in cache miss reply packets only, this HT does not affect inter-tile packets having only head flits.

The proposed HT never infects all the VCs of an input port. It impacts at most *N-1* VCs out of the *N* VCs per VNet available in each input port. This guarantees movement of flits through the infected router and makes it extremely difficult to detect and localize the HT. Since out of all the VCs, now only 1 VC is available for the flit movement, it creates congestion in the network around the infected router. The dead flits that occupy other VCs are part of either cache miss request packet or cache miss reply packet bringing the core to an application level stall due to the ever pending cache misses. Since the FT field in the common prefix is just 2-bit encoding (00-head flit, 01-body flit, 10-tail flit and 11-undefined), the HT attack is realised by flipping just one bit of the FT field.

## 4 EXPERIMENTAL SETUP

To study the impact of our proposed threat model, we use gem5 [4], an event driven cycle accurate simulator to model a 16-tile TCMP with a 4x4 mesh NoC. Each tile consists of an Out-of-Order super-scalar processor with ALPHA architecture and dynamic instruction scheduling. Each tile also houses a two level inclusive cache hierarchy consisting of a 16 KB, 4-way set associative, private L1 Instruction as well as Data caches and a shared 2 MB, 8-way set associative L2 cache with SNUCA technique mapping. We use a main memory of size 8 GB. Ruby module of gem5 is used for modeling the memory module and two level MESI protocol models cache coherence operations. NoC operations are modelled in Garnet 2.0 integrated with gem5. NoC routers use XY routing algorithm and have 4 VCs per VNet per input port. We use single flit request packet, 5 flit reply packets, and 64-bit flit channel. In our simulations, L1-Miss Status Holding Register (MSHR) per core can accommodate 256 entries and Reorder Buffer per core has 192 entries.

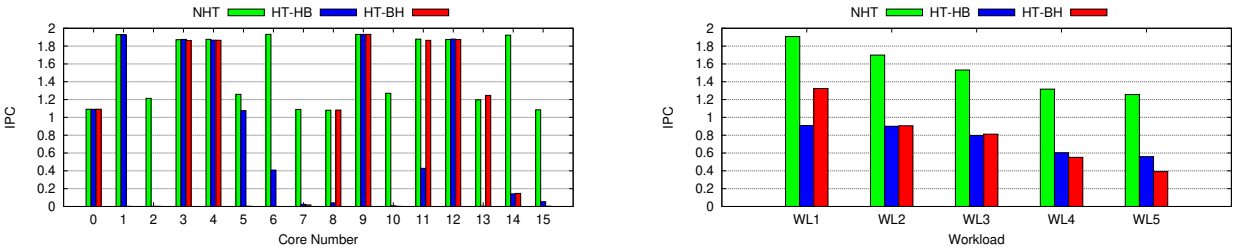
We consider the following architectures in our study.

- **NHT**: Baseline architecture without any HT.
- **HT-HB**: Baseline architecture with an active HT on Router 6 that modifies FT field of common prefix from head flit to body flit with an attack probability *p=0.05*.
- **HT-BH**: Baseline architecture with an active HT on Router 6 that modifies FT field of common prefix from body flit to head flit with an attack probability *p=0.05*.

We analyze HT behaviour with the real application workloads consisting of SPEC CPU 2006 benchmarks [9]. Based on the nature of benchmark and its miss rate, we categorize the benchmarks into High MPKI (greater than 20 misses per 1000 instructions) and Low MPKI (less than 10 misses per 1000 instructions). Based on the experimental values of MPKI obtained we pick *soplex* and *cactusADM*

**Table 1: Workload Details of SPEC CPU 2006 Benchmark Mixes**

| Workload | Benchmark Name (Number of Instances) | | | | Miss Characteristics of Benchmarks |
|---|---|---|---|---|---|
| WL1 | *gromacs* (8) | | *hmmer* (8) | | 100% Low MPKI |
| WL2 | *soplex* (2) | *cactusADM* (2) | *gromacs* (8) | *hmmer* (4) | 75% Low MPKI, 25% High MPKI |
| WL3 | *soplex* (4) | *cactusADM* (4) | gromacs (4) | *hmmer* (4) | 50% Low MPKI, 50% High MPKI |
| WL4 | *soplex* (8) | *cactusADM* (4) | *gromacs* (2) | *hmmer* (2) | 25% Low MPKI, 75% High MPKI |
| WL5 | *soplex* (8) | | *cactusADM* (8) | | 100% High MPKI |




**(a) Core-wise analysis for WL3**

**(b) Workload wise analysis**

**Figure 4: Core-wise and Workload-wise comparison of IPC**

under High MPKI group along with *gromacs* and *hmmer* under Low MPKI group. To run a simulation on a 16-tile TCMP, an application has to be scheduled in each core for the execution. Accordingly, we create five workloads, each consisting of 16 benchmark instances. The details and composition of these workloads (WL1, WL2, WL3, WL4, and WL5) are shown in Table 1. For example WL1 consists of eight instances of each of the benchmarks *gromacs* and *hmmer* and WL3 consists of four instances of each of the benchmarks *soplex, cactusADM, gromacs* and *hmmer*. This classification and subsequent analysis across various workloads help us to understand the impact of HT under applications of varying NoC injection rate and cache miss behaviors. During simulation, the execution is first fast forwarded for 1 million instructions followed by detailed execution of 0.5 million instructions to collect the relevant statistics for each of the above mentioned three architectures. After the fast forwarding phase, HT in router 6 is activated and is triggered with a probability *p=0.05* per packet and statistics are collected.

## 5 EXPERIMENTAL RESULTS

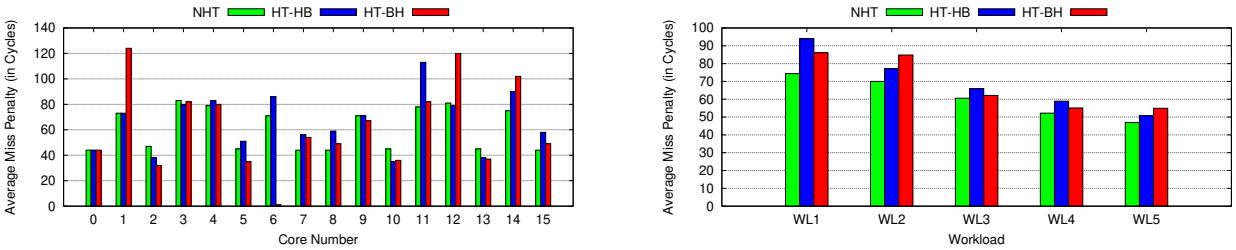
We study the impact of both variants of HT at processor, cache as well as NoC level in terms of various performance metrics like Instructions Per Cycle (IPC), Average miss penalty of L1 cache, Average packet latency, Average buffer occupancy etc.

### 5.1 Impact at Processor Level

In a dynamic scheduling based speculative out-of-order processor, instruction fetch and issue happen in-order, instruction execution happens out-of-order and instruction commit takes place in-order. To ensure an in-order commit, a Re-Order Buffer (ROB) is used which gives an illusion that instruction execution happens in-order. At any point in time, ROB holds all those instructions that are issued but not yet committed. A completed instruction gets committed only when it reaches the head of ROB. Further, an instruction is issued if and only if the following two conditions are met: (i) a free entry is allotted in ROB (ii) free reservation station is allotted in the functional unit where the operation is to be carried out.

LOAD instructions accessing L1-D cache may encounter cache misses that can lead to an NoC packet creation. As HT is attacking a fraction of packets only due to its random triggering, we observe that some of these miss request packets are getting impacted. HT changes the type field in the common prefix from head to body. This leads to a situation where the modified cache miss request packet gets blocked in the infected router, as routing cannot be done on the body flits. As this request packet never reaches its destination, miss reply packet is not created. Similarly, if the HT is affecting a cache miss reply packet, it gets blocked in the infected router and never reaches the source core from which the request is initiated. In both cases, the outstanding cache miss requests in the source core MSHR remains there forever as the reply packets never come back. Since this is a data cache LOAD miss on an already issued instruction, there exists an entry for the corresponding instruction in the ROB of the processor. After few clock cycles, instructions issued prior to this get completed in ROB and they eventually get committed, thereby keeping the HT impacted instruction at the head of ROB. Instructions that are issued after this instruction also get completed but are waiting to get committed. They are not allowed to commit as the head of ROB is our impacted instruction which in an infinite wait remains uncommitted. Over a period of time, ROB gets full with completed instructions that are not able to commit which stalls further instruction issue.

We plot the corewise IPC comparison for WL3 in Figure 4a. We could see that there are some cores where reduction in IPC for




**(a) Core-wise analysis for WL3**

**(b) Workload wise analysis**

**Figure 5: Core-wise and Workload-wise comparison of Average Miss Penalty**

HT infected NoC (HT-HB and HT-BH) with respect to baseline architecture (NHT) is observed. Our analysis shows that these are the cores which suffer from ROB getting full because of pending instruction commit happening due to cache miss request or reply packet getting impacted by HT. This instruction upon reaching the head of ROB prevents further instructions from committing and subsequently blocking issue of new instructions. Cores with very low IPC are the ones whose cache miss request or reply packets are impacted very early in the execution timeline, resulting in stalling of instruction execution at a very early stage giving very low IPC.

We also study the core wise IPC impact in other workloads to understand HT impact. The average IPC across all cores in the respective workloads is plotted in Figure 4b. As expected, workloads consisting of high MPKI applications (WL4 and WL5) have low IPC as they offer higher network load leading to a higher probability of getting impacted by HT earlier in the execution window. Depending on the variation in proportion of cache miss request and reply packets in the total NoC packets, the variation in IPC for HT-HB and HT-BH varies.

## 5.2 Impact at Cache Level

We analyze the impact of HT at cache level using L1 cache miss penalty. L1 cache miss penalty is the number of extra cycles required to fetch the missing data block into the L1 cache from L2 cache or main memory. The different types of NoC packets involved in meeting L1 cache demand miss are L1 request and response packets as well as L2 request and response packets. Since HT-HB can affect any of these packets, this may lead to either request packets being stuck in VC of infected router due to which response packet never gets generated or response packets being stuck in the infected router. In either case, response never reaches the source core where L1 cache miss is initiated which can lead to stalling of instructions as discussed earlier. Similarly HT-BH impacts only response packets and changes it type control field from body to head leading to blocking of response packets which has same impact at processor level as discussed above. Further, the L1 cache misses whose completion involves any of the above packets passing through the infected router suffer an additional delay due bottleneck being created at the infected router because of reduction in available VCs per direction for the traffic to pass through. This in turn leads to increase in L1 cache Miss Penalty compared to baseline architecture which further impacts the instruction execution decreasing efficiency.

**Table 2: Percentage increase in Average Buffer Occupancy of flits traveling from the neighbors to HT infected router.**

| Neighbor | HT-HB | | HT-BH | |
|---|---|---|---|---|
| | Average | Maximum | Average | Maximum |
| North | 11.54% | 850% | 0% | 0% |
| South | 0 | 0% | 6.32% | 250% |
| East | 13.53% | 550% | 12.62% | 750% |
| West | 7.06% | 1200% | 9.28% | 550% |

We plot core-wise Average Miss Penalty (AMP) of WL3 in Figure 5a. The AMP values are calculated for only those requests that are completed. It can be observed that for some cores like 1, 11, 12, etc. there has been a significant increase in the value of Average Miss Penalty for HT-HB and HT-BH as compared to baseline architecture (NHT). We observe that the cores that suffer maximum in terms of increase in AMP are the ones whose either L1 or L2 miss request or response packets pass through the infected router and suffer an additional delay due to reduced number of available VCs. Also these cores are the ones that are running low MPKI benchmarks like *gromacs* and *hmmer*. As discussed earlier, due to less MPKI, there is less chance that cache miss or response packets generated by these cores get attacked by HT in earlier part of execution window. However in later part of execution timeline when Trojan action is complete, packets passing through infected router that are involved in completion of cache miss requests initiated by these low MPKI cores suffer in terms of increase in AMP value. We also analyze the core wise AMP in other workloads to understand HT impact and the average AMP across all cores in the respective workloads is represented in Figure 5b.

## 5.3 Impact at NoC Level

The proposed HT never infects all the VCs of an input port. It impacts at most 3 VCs out of the 4 VCs per VNet in each input port thereby not completely blocking free movement of flits through the infected router. Upon being impacted by HT, each VC creates dead flits leading to bottleneck and congestion for further traffic movement. We analyze the packets passing through the infected

router and its neighbors and find that they indeed suffer additional delays compared to the baseline. We calculate the Average Buffer Occupancy (ABO) of these packets in the neighboring routers of the infected router and results for the same are compiled in Table 2. We calculate the average increase in ABO and maximum increase in ABO for flits coming from a neighboring router of HT and passing through the HT. Some of the flits incur as high as 12 times increase in ABO in the west neighbor of HT. This situation arises when several packets arrive from a particular direction at the infected router simultaneously, leading to congestion since only 1 VC is available for movement and all are competing for it. This causes an abrupt increase in the Buffer Occupancy Period of these packets ultimately delaying further movement of these packets. Accordingly, the average amount of time for which the flit is buffered in neighboring routers of HT infected router has increased up to 13% which in turn degrades the performance of running applications. Zero entry indicates that flits from those respective neighbors did not incur any additional delay in the simulation window due to the presence of dead flits.

### 5.4 Impact of Varying HT Attack Probability

All the results discussed above have been compiled with an HT attack probability of p=0.05. To further study the impact of HT in terms of attack probability, we model HT with three varying attack probabilities: *p=0.05*, *p=0.1* and *p=0.15*. We observe that as the attack probability is increased, the impact of HT activation happens early in the simulation window. As a consequence of this, the number of packets that suffer from congestion increases leading to higher average packet latency and average cache miss penalty.

### 5.5 Hardware Overhead of Trojan Circuit

We use ProNoC [15] that facilitates prototyping of NoC based systems to implement the proposed HT in Verilog HDL and append it along with baseline NoC router architecture in ProNoC to get the design blueprint of the HT infected router. The design is synthesised in Synopsys Design Compiler using 90nm technology for functional verification and analysis of the timing constraints. Synthesis results show that the HT circuitry is not operating in the critical path and makes sure that the HT infected router also can operate at the same frequency as that of the baseline router at 1 GHz. We observe that the HT circuit incurs an area overhead of 1.2% and a minimal static power overhead of 0.3% with respect to baseline router. The dynamic power dissipation is negligibly minimal as the HT remains dormant for most of time.

## 6 CONCLUSION AND FUTURE WORK

This paper presented an HT attack on common prefixes of flits in NoC and analyzed the impact created by it. The proposed HT resides in the input buffer of the infected router and manipulates the common prefix fields. By modelling the HT behaviour, we demonstrated that the HT can create dead flits in input buffers of HT infected router. We experimentally proved that these dead flits consume available resources in NoC and create congestion bottleneck in neighbors of HT router. We studied the impact of the proposed HT at core level, cache level, and at NoC level. We also experimentally proved that the proposed HT mounted on an NoC router had enough potential to degrade overall system performance by stalling applications running on cores. As future works we plan to propose suitable cost-effective mitigation mechanisms.

## 7 ACKNOWLEDGEMENT

This work is supported in part by the funding from Information Security Education and Awareness Project Phase-II, MeitY, Govt. of India. Mohammad Humam Khan thanks Samsung India Electronics Limited for funding through Samsung Fellowship.

## REFERENCES

[1] Dean Michael Ancajas, Koushik Chakraborty, and Sanghamitra Roy. 2014. Fortnocs: Mitigating the threat of a compromised noc. In *Proceedings of the 51st Annual Design Automation Conference*. 1–6.
[2] Luca Benini and Giovanni De Micheli. 2002. Networks on chips: A new SoC paradigm. *computer* 35, 1 (2002), 70–78.
[3] Swarup Bhunia, Michael S Hsiao, Mainak Banga, and Seetharam Narasimhan. 2014. Hardware Trojan attacks: Threat analysis and countermeasures. *Proc. IEEE* 102, 8 (2014), 1229–1247.
[4] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R Hower, Tushar Krishna, Somayeh Sardashti, et al. 2011. The gem5 simulator. *ACM SIGARCH computer architecture news* 39, 2 (2011), 1–7.
[5] Travis Boraten and Avinash Kodi. 2018. Mitigation of Hardware Trojan based Denial-of-Service attack for secure NoCs. *J. Parallel and Distrib. Comput.* 111 (2018), 24–38.
[6] Subodha Charles, Yangdi Lyu, and Prabhat Mishra. 2019. Real-time detection and localization of DoS attacks in NoC based SoCs. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 1160–1165.
[7] George Chrysos. 2014. Intel® Xeon Phi™ coprocessor-the architecture. *Intel Whitepaper* 176 (2014), 43.
[8] Farimah Farahmandi, Yuanwen Huang, and Prabhat Mishra. 2020. *System-on-Chip Security*. Springer.
[9] John L Henning. 2006. SPEC CPU2006 benchmark descriptions. *ACM SIGARCH Computer Architecture News* 34, 4 (2006), 1–17.
[10] Vedika JK, Manju R, Ruchika Gupta, John Jose, and S. Nandi. 2021. Packet Header Attack by Hardware Trojan in NoC based TCMP and its Impact Analysis. In *2021 15th IEEE ACM International Symposium on Networks on Chip (NOCS)*. IEEE/ACM.
[11] Manoj Kumar JYV, Ayas Kanta Swain, Sudeendra Kumar, Sauvagya Ranjan Sahoo, and Kamalakanta Mahapatra. 2018. Run time mitigation of performance degradation hardware trojan attacks in network on chip. In *2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 738–743.
[12] He Li, Qiang Liu, and Jiliang Zhang. 2016. A survey of hardware Trojan threat and defense. *Integration* 55 (2016), 426–437.
[13] Sharad Malik and Pramod Subramanyan. 2016. Specification and modeling for Systems-on-Chip security verification. In *2016 53nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. IEEE, 1–6.
[14] R Manju, Abhijit Das, John Jose, and Prabhat Mishra. 2020. SECTAR: Secure NoC using Trojan Aware Routing. In *2020 14th IEEE/ACM International Symposium on Networks-on-Chip (NOCS)*. IEEE, 1–8.
[15] Alireza Monemi, Jia Wei Tang, Maurizio Palesi, and Muhammad N Marsono. 2017. ProNoC: A low latency network-on-chip based many-core system-on-chip prototyping platform. *Microprocessors and Microsystems* 54 (2017), 60–74.
[16] Nachiketh Potlapally. 2011. Hardware security in practice: Challenges and opportunities. In *2011 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, 93–98.
[17] Manju Rajan, Abhijit Das, John Jose, and Prabhat Mishra. 2021. Trojan Aware Network-on-Chip Routing. *Network-on-Chip Security and Privacy* (2021), 277.
[18] Anderson Camargo Sant'Ana, Henrique Medina, and Fernando Gehm Moraes. 2021. Security Vulnerabilities and Countermeasures in MPSoCs. *IEEE Design & Test* (2021).
[19] Johanna Sepúlveda, Daniel Flórez, and Guy Gogniat. 2015. Reconfigurable security architecture for disrupted protection zones in NoC-based MPSoCs. In *2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*. IEEE, 1–8.
[20] Kan Xiao, Domenic Forte, Yier Jin, Ramesh Karri, Swarup Bhunia, and Mohammad Tehranipoor. 2016. Hardware trojans: Lessons learned after one decade of research. *ACM Transactions on Design Automation of Electronic Systems (TODAES)* 22, 1 (2016), 1–23.
[21] Jiaqi Yao, Ying Zhang, Zhiming Mao, Sen Li, Minghui Ge, and Xin Chen. 2020. On-line Detection and Localization of DoS Attacks in NoC. In *2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, Vol. 9. IEEE, 173–178.