Mehran University Research Journal of Engineering and Technology
Vol. 39, No. 4, 686 - 704, October 2020
p-ISSN: 0254-7821, e-ISSN: 2413-7219
DOI: 10.22581/muet1982.2004.02

Protector: A Permanent Fault Resilient Router Architecture for Network on Chip

Naveed Khan Baloch[1a], Ayaz Hussain[1b], and Muhammad Iram Baig[2]

RECEIVED ON 05.07.2018, ACCEPTED ON 26.07.2019

ABSTRACT

The decreasing size of the transistor has increased the vulnerability towards faults. Increasing number of cores on a single chip has made the concept of Network on Chip (NoC) a standard communication backbone among cores. This facility comes with vulnerability of faults in the system due to decreasing size of transistors. A permanent fault in the network leads to undesirable consequence such as permanent blocking of flits or failure of the whole router. Preserving the router in the operational state has a significant impact on the reliability of the system. Permanent fault in buffers and pipeline stages of the router has a high impact on performance. The proposed router architecture Protector provides faults protection to both buffers and pipelines stages by exploiting the concepts of borrowing from other resources, using bypass paths and by creating multiple paths to reach output. The proposed router incurred an area overhead of 30% as compared to the baseline design. Reliability analysis using Silicon Protection Factor indicates that the proposed router has better fault tolerance efficiency as compared to state of the art. Latency analysis using PARSEC and SPLASH-2 benchmarks indicates proposed router incurs 13% and 16% latency overhead in the presence of faults.

Keywords: Network-on-Chip, Fault Tolerance, Silicon Protection Factor, Router Architecture.

1. INTRODUCTION

Miniscule technology feature sizes into the deep nanometer regime have enabled microprocessors with billions of transistors on a single chip [1-2]. This extraordinarily abundant number of resources has directed the designers to another computational architecture type Chip-Multiprocessor (CMP) [3]. The large quantity of components on a single silicon chip has shifted the design paradigm from computational-centric to communication-centric architectures. The communication between various computational cores on a single chip has a high influence on the performance of the chip. The need to handle this severe communication necessity has led to the initiation of NoC architectures [4-5]. In NoC computational cores are separated from communication infrastructure. Data is transmitted from source to destination in the form of packets. Routers and links are used to construct the necessary NoC structure. NoC is becoming the promising solution to interconnect the on-chip cores due to its scalability and modular nature [6].

The complexity of circuits, rapid advancement in technology scaling and decreasing feature size of the transistors may affect the reliability of the chip and cause many fault mechanisms [7]. Faults can be classified into three types, i.e. Permanent faults, Intermittent faults and Transient faults [8]. Permanent faults occur at the fabrication time or during the operation of the circuit are logic faults in which transistors are open or short, i.e. Stuck at 0 or Stuck at 1, delay faults in which transistors are very slow and cause timing violations. Transient faults are temporary

[1] Department of Computer Engineering, University of Engineering and Technology, Taxila, Pakistan.
Email: [a]Naveed.khan@uettaxila.edu.pk (Corresponding Author), [b]academia4ayaz@gmail.com
[2] Department of Electrical Engineering, University of Engineering and Technology, Taxila, Pakistan.
Email: iram.baig@uettaxila.edu.pk

faults which occur for one or few clock cycles at random location of the chip. These faults may occur continuously during the lifetime of the chip and affect the packets traveling into the network. Intermittent faults are like transient faults but occur in a burst at the same location. Crosstalk and electromagnetic interference are the leading causes of intermittent faults, and with time, these faults may lead to the permanent faults.

The semiconductor industry has categorized the faults for in-field failure form manufactures opinion. Premature failure happens due to manufactures deficiencies and rate declines over time. Radiations cause random failure and rate constant over time. Wear-out failure is due to the aging process, and rate of failure increases over time. Radiation is one of the failure mechanism due to alpha particles initiating from the device impurities [9]. Radiation may cause flipping of bits which is called Single Event Upset (SEU). Radiation-based SEU can also cause errors in logic circuits presented by [10-12]. Crosstalk between two wires is another type of fault mechanism which is the main reason of electromagnetic interference in the chip. A signal on one wire can disturb the other wires and can cause increased signal delay and glitches [13-14]. Electrostatic discharge is also one of the reasons for faults occurring in the chip which may cause PN junction breakdown or wiring breakdown [15]. The Electromigration is another fundamental fault mechanism. It first creates increased delay and then permanently damage the wires [16]. Negative Bias Temperature Instability (NBTI) raises the threshold voltage of the transistors with the passage of time which may cause faults in the circuits [17-18]. Hot carriers (electron-hole pairs) penetration into the dielectric material resulted in increased switching time of transistors and degrades the performance of the circuit [19]. Work is presented about wear-out and aging problem for example [20] which causes faults in the chip.

A single fault in a NoC creates errors in the chip which results in undesirable conditions, i.e. increased latencies, packet loss and degraded performance. So, its utmost desire to include fault tolerant techniques in initial design stages. In this work, we have a focus on tolerating permanent faults in NoC router. Fault tolerant designs for links have been presented previously by many researchers [21-25] and is out of the scope of this paper. A faulty router may be handled by fault-tolerant deflection routing [26]. If the faulty router is treated as a node or link failure, then task remapping occurs, that can degrade the performance of the network. A generic router consists of buffers, virtual channel allocators, switch allocator, crossbar, muxes, and de-muxes. In this paper, a fault tolerant router architecture is presented to tolerate permanent faults occurring at various locations in the router and affecting the reliability of the chip. This architecture not only provides fault tolerance but also enhances the performance of the network by a grouping of adjacent ports, sharing of resources, temporal parallelism, rectification circuitry and multiple routes to avoid faults.

The rest of the paper is arranged as follows. In Section 2 previous related work is presented, Section 3 describes the generic NoC router architecture, proposed fault-tolerant router design is given in Section 4, Results and analysis are presented in Section 5, and finally, the conclusion is drawn in Section 6.

2. RELATED WORK

In this section, we present the previous fault tolerant router architectures, which tackle the permanent faults in router buffers and pipeline stages. Authors in [27] presented Bullet Proof router architecture which employs Triple Modular Redundancy (TMR) and Error Control Coding (ECC) to provide fault tolerance in the design. The spatial redundancy-based techniques are very expansive because it requires multiple copies of hardware and thus more silicon area on the chip. Vicis is another router architecture to provide fault tolerance both at the network level and at the router level provided by [28]. It uses an adaptive routing algorithm and input port swapping to tolerate faults occurring at nodes. A bypass bus is used in the router to tolerate the crossbar faults, ECC is used to tolerate the link faults. In [29] the authors have presented fault tolerant router architecture RoCo. This architecture decouples the row and column resources with separate arbiters and smaller crossbars. If one of the components fails other continue to work. This way

the router remains in working condition by fault-free components in the presence of a permanent fault in another component but results in degraded performance. [30] Proposes a technique of Default Backup Paths (DBP) to bypass the router's faulty internal components. In this way, all the routers in the network form a ring topology, and cores continue to communicate with each other even when all the routers become faulty. [31] have presented a fault tolerant technique to avoid both permanent and transient faults. A fault-tolerant deflection routing algorithm is proposed to avoid the permanent faults occurring in the router. REPAIR is another router architecture that utilizes some partial redundancies combine with ECC technique to protect the faults in buffers and crossbar switch presented by [32].

A transient and permanent fault tolerant based Enhanced Reliability Aware Virtual Channel Architecture (ERAVC), presented by [33] and more comprehensive version [34] utilizes the virtual channels in such a way that input channel being idle because faulty neighboring routers utilize it efficiently to improve the performance. Authors in [35] proposed a Partial Virtual Channel (PVC) sharing router. The idea to pair two adjacent ports via a common DeMux to provide better resource sharing and fault tolerance. However, if a fault occurs in a common DeMux, all corresponding resources cannot be utilized anymore. [36] presented a low cost, high-performance router architecture by grouping the ports via Dynamic Resource Sharing (DRS) block to provide fault tolerance to the input buffers only. They provide the details analysis for SPF (Silicon Protection Factor) but ignore the pipeline stages of the router. [37] presented a permanent fault tolerant router architecture to tackle the faults occurring at the router pipeline stages. They assumed that ECC techniques are used to protect the buffers thoroughly. [38] presented a fault tolerant router architecture Shield. They enhanced the previous work and provided a critical gate identification-based algorithm to tackle transient fault. They provide analysis for SPF for only the pipeline stages and ignore the input ports.

Bahrebar *et. al.* [39] proposed a dynamically reconfigurable routing technique for tolerating the faults. Previous fault tolerant deflection routing techniques were creating hotspots around the faulty router and thus creating more delays. The proposed technique not only bypasses the faulty router but also avoids the frequently communicating nodes and allow packets to travel on shorter paths for minimal latency. Yuan *et. al.* [40] designed low overhead micro-architecture for the NoC router. Commonly (Error Correction Codes (ECC) are utilized to correct the error in packets. The encoders and decoders used in ECC techniques introduced area overhead in the designs. This work is based on reusing the decoders present in the Network Interfaces (NI) and named as Send-Back ECC. The proposed design gives a much better performance and low hardware overhead compared to previous work, utilizing ECC for fault tolerance.

Moriam and Fettweis [41] highlighted the need for precise, flexible and fast analytical models to evaluate the fault-tolerant routing algorithms. They utilized the algebraic manipulation of channel dependency matrix to design the analytic approach for the evaluation of adaptive fault resilience routing techniques. The presented model can assess the number of substitute paths between the communicating cores in the existence of faults. This model can be adapted to assess the reliability of network topology and with any fault resilience routing algorithm.

Kasem *et. al.* [42] presented a solution for area overhead problem for self-healing reliable NoC router architectures. These architectures are based on spatial redundancy and use additional components to provide fault tolerance. In such systems, routers are isolated from the Processing Equipment (PE) which reduces the performance of the system. This work keeps the record of faulty routers and successfully delivers the packet to PE by sharing the other ports to the local ports.

Our proposed router architecture named Protector is different from previous work in a way that it protects each portion of the router separately. It exploits idle time of existing resources and employs minimum correction circuitry to achieve the fault tolerance from multiple permanent faults. We have a focus to tolerate stuck at 0, 1 and other permanent faults occurring in the router.

The motivation behind the proposed architecture Protector is to provide a better fault-tolerant router that can be used in future NoC chips. Existing architectures does not provide fault tolerance for all the components of the router and incurs more considerable overhead or results in degraded performance. Protector provides fault tolerance at the cost of smaller overhead and without compromising the performance of the system.

The significant contributions of this paper can be summarized as:

1- The protector can provide fault tolerance in input ports, buffers, arbiters and cover all the pipeline stages.
2- Performance analysis of the Protector involves area and latency comparison concerning baseline router architecture design.
3- Reliability of Protector is compared with state of the art techniques.

3. BASELINE NOC ROUTER ARCHITECTURE

This section describes the generic NoC router architecture which is modified in the next section to provide fault tolerance in router. Fig. 1 shows the interconnection of routers in a 4x4 mesh topology. Each router is connected to some PE at the local port. Fig. 2 [16] shows the overview of the baseline router architecture used in the NoC. The primary router architecture used in the mesh topology consists of five inputs and five output ports for communication among different cores. It consists of four directional ports East, West, North, South and one local port for interfacing with PE. The PE is attached to the local port via NI. The essential components in the router consist of four pipeline stages along with VC buffers, mux, and de-mux. First, three stages in the router pipeline are Routing Computation (RC), Virtual Channel Allocation (VA), Switch Allocation (SA) are responsible for the generation of the control signals for smooth flow of the packet in the network. The fourth stage, XB (Crossbar) connects all input ports to the output ports.

For efficient utilization of the NoC bandwidth, the wormhole switching is used in the network. In wormhole switching a packet is segmented into multiple flits. There are three types of flits named as head, payload and tail flit. The head flit in the network is used for the allocation of necessary resources required by the packet to traverse through the network. The payload contains the actual information to be communicated. The tail flit is used for de-allocating the resources reserved by the head flit for a specific packet. Every incoming packet proceeds through router pipeline stages to improve the performance of the system [44]. Packets entering the router are stored in the input port VC buffers. Each input port consists of the mux, de-mux and VC buffers as shown in Fig. 3 [43]. De-mux is used for guiding the flit to be placed in the assigned VC buffer, and the Mux is used to transfer the winning flit to the crossbar.

The input port architecture for baseline router is shown in Fig. 3. Each flit arrives at the input port is placed in these VC buffers until they get crossbar time. For each VC buffer 5 states are maintained in the status register. States are named as a Global state (G), Route (R), Output VC (O), Pointer (P) and Credit count (C).

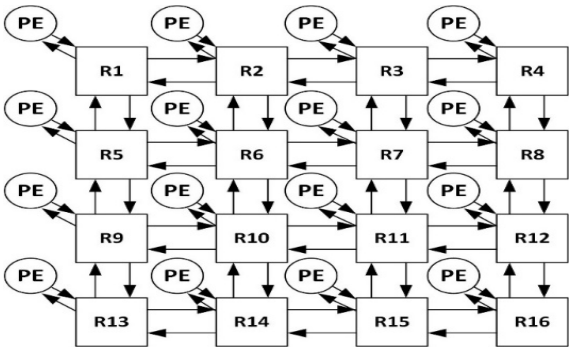

Fig. 1: Routers Connected in 4X4 Mesh NoC

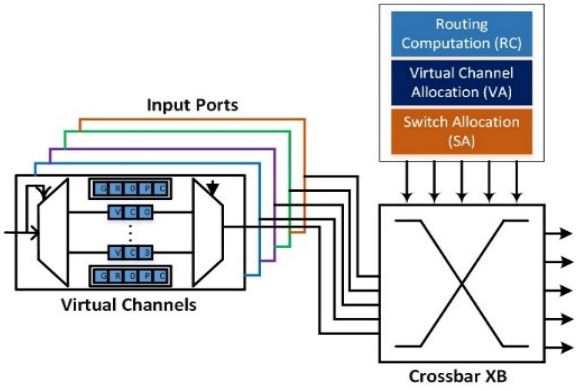

Fig. 2: Generic NoC Router Details

For efficient utilization of the NoC bandwidth, the wormhole switching is used in the network. In wormhole switching a packet is segmented into multiple flits. There are three types of flits named as head, payload and tail flit. The head flit in the network is used for the allocation of necessary resources required by the packet to traverse through the network. The payload contains the actual information to be communicated. The tail flit is used for de-allocating the resources reserved by the head flit for a specific packet. Every incoming packet proceeds through router pipeline stages to improve the performance of the system [44]. Packets entering the router are stored in the input port VC buffers. Each input port consists of the mux, de-mux and VC buffers as shown in Fig. 3 [43]. De-mux is used for guiding the flit to be placed in the assigned VC buffer, and the Mux is used to transfer the winning flit to the crossbar.

The input port architecture for baseline router is shown in Fig. 3. Each flit arrives at the input port is placed in these VC buffers until they get crossbar time. For each VC buffer 5 states are maintained in the status register. States are named as a Global state (G), Route (R), Output VC (O), Pointer (P) and Credit count (C).

The G indicates the pipeline stage of the packet. The result of routing computation stage is placed in the R. The result of virtual channel allocation is placed in O. The P indicates the number of flits in the virtual channel. The C indicates the number of free slots available at the downstream router. These status registers ensure smooth flow of the packet through the network.

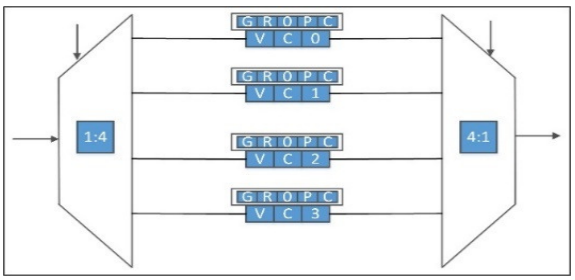

Fig. 3: Input Port Architecture

The baseline router contains four pipeline stages as shown in Fig. 4 [39]. The first stage is RC. This stage extracts the destination information in the header part of the flit. The result of RC gives the output port which is determined by the routing protocol used in the network.

Fig. 4: Router Pipeline Stages

The second stage is VA which is responsible for allocating free VC buffer at the downstream router. The virtual channel allocation stages are designed to remove conflicts among multiple VC buffer requests. This process is performed in two stages as shown in Fig. 5 [45]. In the first stage, local arbitration is done to reduce the number of requests. Since one input virtual-channel is reserved for one VC buffer, the second stage of the virtual channel allocation is performed to remove the conflicts among input ports to access the same VC buffer at the downstream router.

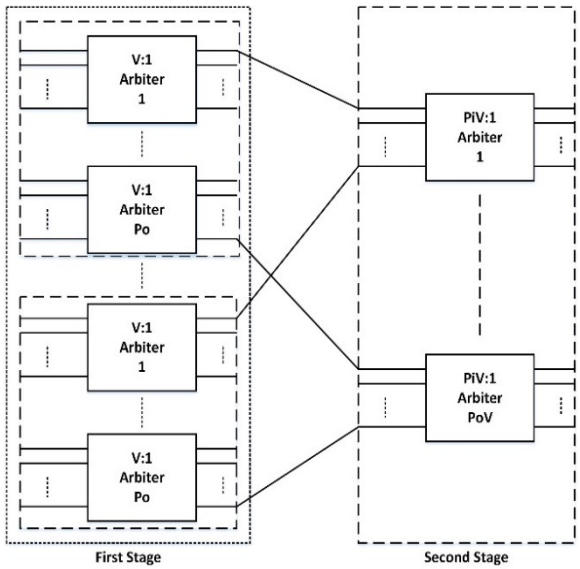

Fig. 5: Virtual Channel Arbiter

Next stage in the pipeline after VA is SA, which is responsible for granting permission of VC buffer to access the crossbar. The switch allocation is performed in two stages as shown in Fig. 6 [45]. The first stage chose a winning VC buffer from each input port and the second stage is responsible for removing conflicts among winning VC buffers of different input ports trying to transmit a flit through the crossbar. The last stage is XB which is used to create a connection between the input and output ports of the router. In this stage, winning flits from each input port is transmitted

to the selected output of the router. Fig. 7 shows the design of XB for the baseline router. It consists of five multiplexers of 5X1. One multiplexer for each output port.

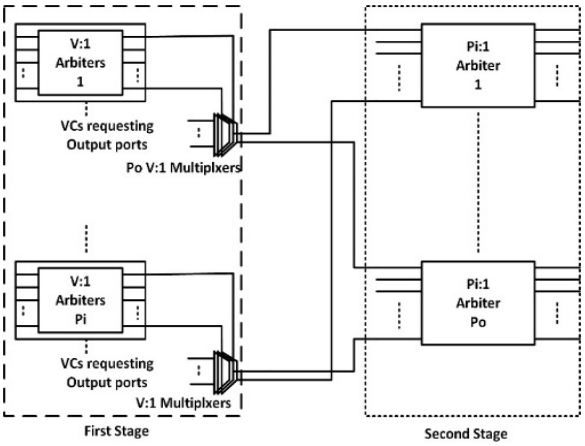

Fig. 6: Switch Arbiter

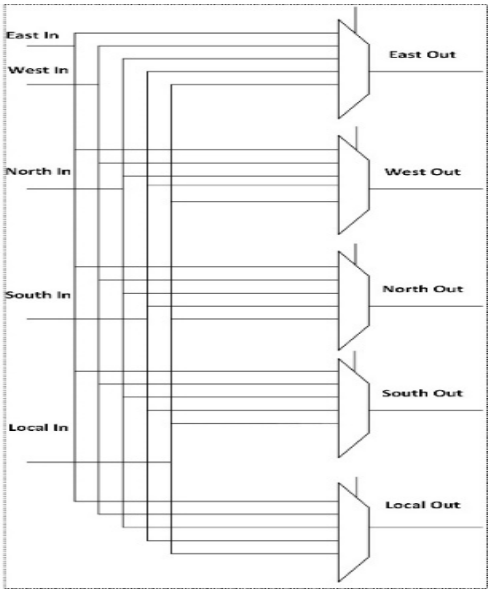

Fig. 7: Crossbar

4. PROTECTOR: PROPOSED PERMANENT FAULT TOLERANT ROUTER

The proposed router architecture Protector provides fault tolerance to both buffers and pipeline stages of the router in the presence of the permanent faults. We design this router to modify the baseline architecture stages. The design to individual stages of the proposed router is explained below in separate sub-sections.

4.1 Protector: Input Port

The First In First Out (FIFO) buffers in the router is the first place where each incoming flit resides. The significant portion of the router consists of buffers. They consume the most substantial fraction of the dynamic and static power [34] than the packet transmission [35]. It is evident that the probability of a permanent faults occurring in the input port is high because it occupies a larger area. Thus, it is necessary to provide fault protection for this portion of the router. Fault tolerance at this stage is provided by sharing the neighboring port resources without adding extra resources. The proposed design in this paper has achieved the fault protection for the input port architecture by using DRS approach by [36]. We utilize the sharing approach for the input ports in the form of (2, 2, 1) pairing. We paired East with North, West with South while the local port remains alone to achieve the best tradeoffs between NoC critical performance parameters. The DRS module in each group operates independently in such a way that occurrence of a permanent fault in router does not fail the whole group. The decoupled structure enables the router to tolerate multiple faults in the input port architecture. In this way fault in one input port is not disturbing the other port and paired group perform their functions in the presence of a fault. The pairing architecture adopted for the proposed router is shown in Fig. 8.

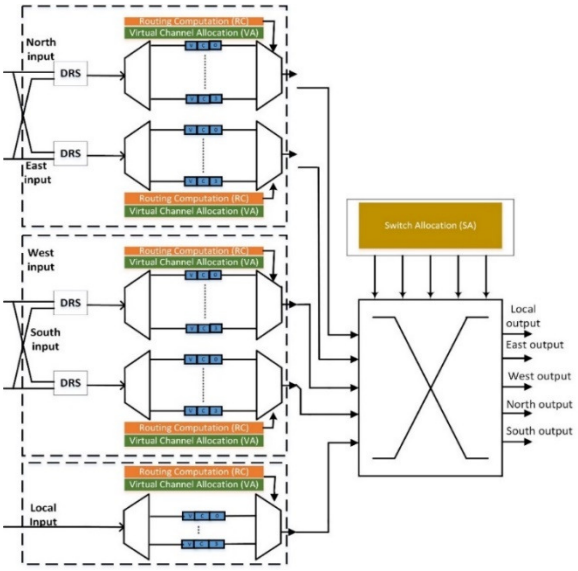

Fig. 8: Input Port Pairing in a Router

If no fault occurs in the input port, then each of the port uses its default way to transmit the packet. Otherwise, the bypass paths are used to complete the communication. We utilize the fault detection mechanism by using the checkers designed by NoCAlert [46]. The fault control unit of the router can detect the faults on the input, demux, and mux of the input port. The pseudo code for the working of FCU (Fault Control Unit) used is paired port is described in Algorithm-1. One paired group can tolerate one RC fault, seven VC buffers faults, one DRS fault, one Mux and one demux fault. So, total faults tolerated by two groups are ((1+7+1+1+1) x2) = 22. The local port remains alone. Thus it can tolerate only 3 VC buffers faults.

4.2 Protector: RC Unit

A separate RC unit is connected to each input port to extract the destination information from the packet. The complexity of the RC unit depends upon the routing protocol. We utilized dimension order (XY) routing algorithm. The XY routing does not require tables to stores the path of the packet thus it causes less area overhead [47]. If the RC unit suffers from a permanent fault, then it is not able to compute the output port. The traversal of flit through the router depends upon the output of the RC unit. Thus, it is necessary to protect the RC stage. Fig. 9 shows the checkers based of NoC Alert [46] for detection of RC faults in the router. Checker Fig. 9(a) detects the calculation of wrong output port, which can transmit the packet in the wrong direction away from the destination which results in increased latency and deadlock. Checker in Fig. 9(b) can detect the invalid output port direction. As shown in the figure, each output port direction is assigned a 3-bit code from 0-4. Rest of the numbers in 3-bit representation from 5 onwards are invalid. These are very lightweight checkers for detection of faults in the router and give minimum area overhead.

Each input port has its RC unit thus without adding extra component the RC protection is achieved by sharing the RC for the nearby port. The grouping of the ports is in the form of (2, 2, 1) pairs results in protection of one 1 RC unit in each group. In this way total, 2 RC faults can be tolerated in a group.

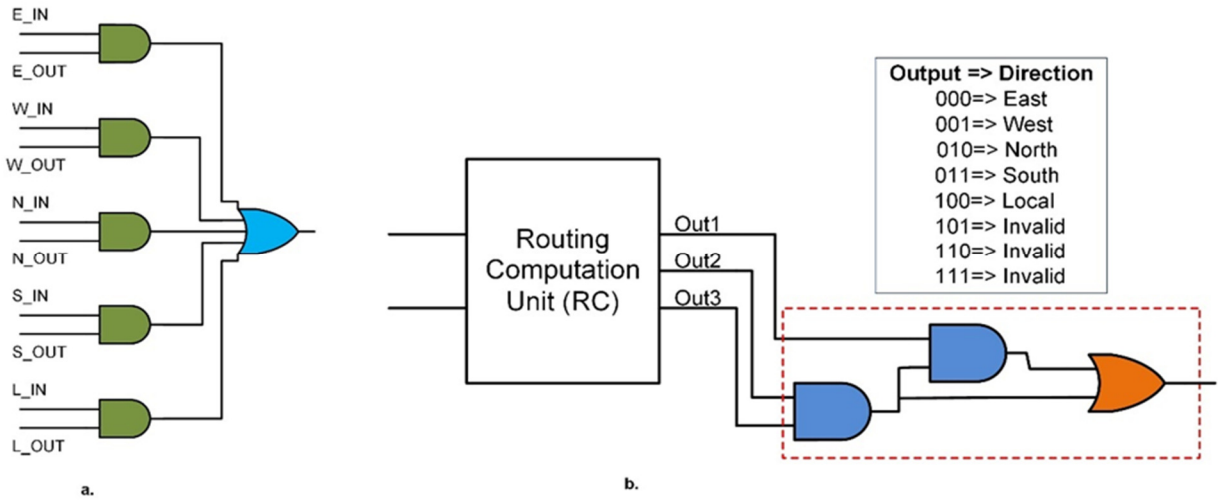

Fig. 9: RC Fault Detection Checkers (a) Wrong Output Port (b) Invalid Output Port

ALGORITHM-1: FCU IS WORKING IN PAIRED PORT

if(no fault exists in group)
then

output channel=Select_Channel_2;//default channel

else if(default channel or Demux or Mux of input is faulty)
then

output channel=Select_Channel_1;//Other paired group input channel

else if(channel_1 is faulty or Demux or Mux of input is faulty)
then

output channel is assigned to default and paired group input channel

else //default channel is faulty
selected channel is not important

end if;

ALGORITHM-2: PSEUDO CODE FOR THE SELECTION OF BYPASS PATH IN SA
if(Directional port arbiters are faulty) **then** **if(paired channel_1 arbiter faulty)** **then** Mux_out=Default VC buffer is selected **else if(paired channel_1 arbiter is faulty and VC buffer register is faulty)** **then** Mux_out=Use paired channel_2 VC buffer ID to select VC buffer from faulty port **else if(paired channel_2 arbiter faulty)** **then** Mux_out=Default VC buffer is selected **else if(paired channel_2 arbiter is faulty and VC buffer register is faulty)** **then** Mux_out=Use paired channel_1 VC buffer ID to select VC buffer from faulty port **else(Local port arbiter is faulty)** **then** Mux_out=Default VC buffer is selected **else** Mux_out=arbiters selected VC buffer **end if;**

4.3 Protector VA Unit

Two sub-stages of VA are shown in Fig. 5. Each VC buffer is associated with Po V:1 arbiter. The term V represents the number of VC buffers presents in the downstream router where Po is the number of output ports. The DRS module does not provide fault protection for the pipelines stages. The pipeline stages in the router are responsible for optimal utilization of the resources and ensure smooth flow of traffic. We propose fault protection for each pipeline stage separately to increase the reliability of the architecture towards permanent faults. If a permanent fault manifests in one of the arbiter associated with VC buffers, then all arbiters associated with that VC buffers are considered to be faulty. In the presence of the permanent fault, the flits in the VC buffer is not able to arbitrate for an empty VC buffer at the downstream router which may lead to starvation and blocking of the packet that resides in that VC buffers. Each VC buffer is associated with Po V:1 arbiter thus we can utilize other VC buffers arbiters to participate in the VC allocation process. As we paired input ports in the form of (2, 2, 1) grouping, we can use arbiter of the other VC buffers resides in the same port and also from the paired group input port. To achieve this, we modified the input port architecture to share the arbiters within a group. Thus, by using another VC buffer arbiter, VA stage can be performed in the presence of a fault. The possible fault scenarios and delay involved in sharing arbiter among the group are as follow:

The modified VA architecture is shown in Fig. 10. Each input port has 4 VC buffers, and each VC buffer has a set of arbiters associated with it. If 1 permanent fault occurs in one VC buffer of the port, still 3 fault-free VC buffers are present in that port. The Faulty VC buffer request to use another VC buffer by analyzing the G status register of the other VC buffers. If it finds out that other VC buffer is in Ideal state or Routing state, it means that arbiters associated with that VC buffers are free. The delay in finding out the independent VC buffer arbiters in same port lies on the critical path. Thus, it does not result in overhead. If all the VC buffers within port are busy, then it looks for independent VC arbiters in the paired group. It results in a delay of 1 cycle. If it is not able to find a free VC buffer, then it results in unsuccessful virtual channel allocation.

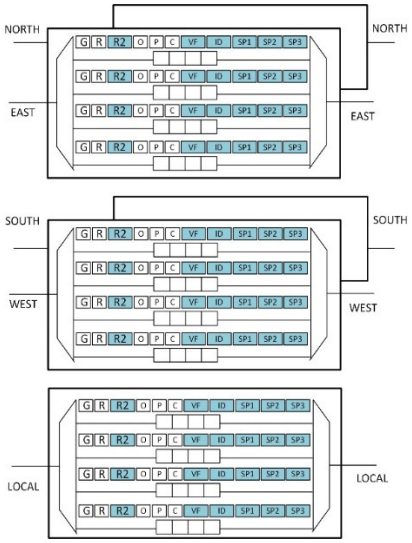

Fig. 10. Protector: Modified VA Stage

If 4 permanent faults occurred in same input port which is paired, then it uses arbiter of the other paired group. If arbiter of the other port is busy in doing arbitration, then it results in a delay of two cycles, 1 cycle for finding arbiter in the same port and another

cycle for finding free arbiter in paired group. The two-cycle delay occurs if all arbiters of the paired group are busy which results in unsuccessful virtual channel allocation.

The proposed VA stage can tolerate a maximum of 7 arbiter faults in each group and 3 arbiters fault in the local port. Total faults tolerated by the VA stage is 17.

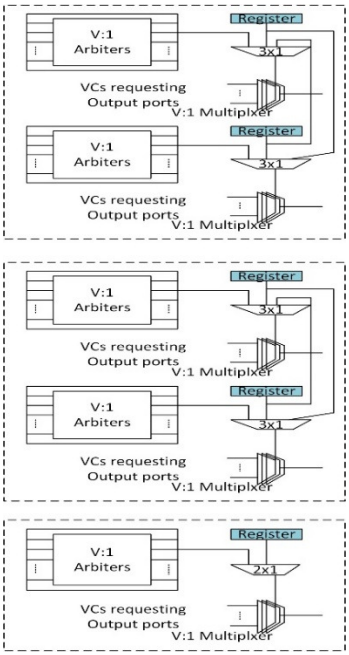

Fig.11: Protector: Modified SA Stage

4.4 Protector Modified Input Port Architecture

The input port architecture of the baseline router is modified for the proposed router Protector. New fields are added to support the arbiter sharing among VC buffers and to support protection for the crossbar design. Fig. 12 shows the modified input port architecture with extra state fields R2, VF, ID, SP1, SP2, and SP3. The arbiter sharing is started by placing the RC results of faulty VC buffer in other VC buffer status register R2 in which arbiters are free. The identification of the faulty VC buffer in shown in ID field and raise VF flag high. The VF is used for the identification purpose to check the free VC buffers. When virtual channel allocation is completed for the faulty arbiters, then its entries are flushed out and appropriate virtual channel states are updated. Other fields are used for providing fault protection in XB stage. The tolerance for permanent faults is achieved by using redundant resources.

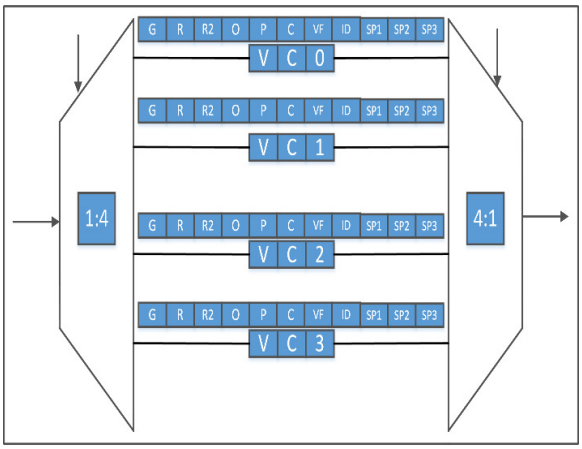

FIG. 12. PROTECTOR: MODIFIED INPUT PORT

4.5 Protector: SA Unit

The switch allocation is performed on two sub-stages as shown in Fig 6. In the first stage, each input port is associated with V:1 arbiter. The first stage of the switch allocation is used to select a VC buffer from each input port. The winning flit from each input port then participate in the switch allocation process. The second stage of switch allocation is responsible for removing conflicts among these winning VC buffers in the first stage and grant access to traverse through the crossbar.

When input port suffered from arbiter fault, then arbiter cannot choose a VC buffer from that port and cannot participate in the switch allocation process. Packets reside in that port are permanently blocked because they never win arbitration. To overcome this situation, we proposed to create a multiple bypass paths for each V:1 arbiter that can be used to select the default VC buffer. The proposed SA first stage is shown in Fig. 11. This is achieved with the help of 3:1 mux. One input to this mux is from the arbiter, one from a register which contains default ID of the VC buffer, one from the other paired group input port register containing default Id of the VC buffers. In case of faulty arbiter, the bypass paths are used to select the default id of the VC buffer as a winner. The pseudo code for the selection of Bypass path is shown in Algorithm-2. There are total 4 VC buffers in each input port having an id of VC1, VC2, VC3, and VC4. Any of the VC buffers ID can be chosen as a winning virtual channel. In the proposed router architecture, there exist multiple paths for providing fault protection against permanent faults. We picked VC1 as default virtual channel.

If a permanent fault occurs in the North port which is paired with East, the North port is not able to participate in the virtual channel allocation process. Modified SA for Protector is shown in Fig. 11. This fault is tolerated by selecting default VC buffer ID which came from a register present in the North Port. Here two possible scenarios can occur. If the selected VC buffer has flits to transmit, then it results in zero overhead on latency. If the selected VC buffer is empty and other VC buffers have flits to transmit the contents of other VC buffer is shifted in the default VC buffers. This content shifting results in an overhead of one cycle. The proposed SA can also tolerate the outcome of a permanent fault in the register storing default VC buffer id. This fault is tolerated by using a second bypass path which came from the paired group East register. As each port has an equal number of VC buffers thus using default virtual channel ID of the other paired port does not result in inconsistency of the data and still default VC buffer is selected using this ID. The register contains only VC buffer ID and the second path gives the only ID which is decoded by the Fault control unit to select VC buffer from the faulty port via that VC buffer ID. Each group tolerates 4 faults while local port tolerates 1 fault.

The second stage of SA contains Pi:1 arbiter which belong to each output port. The winning VC buffers in the first stage get access to the selected output port. If a number of the arbiters associated with that port is faulty, then it is not able to access the output port. To solve this problem, we modified the XB design. The proposed XB design has maximum two paths to access each port. The working XB stage is explained in the subsection where we discussed the proposed XB stage.

4.6 Protector XB Protection

The baseline design of XB stage is shown in Fig. 7. In the baseline design of XB, each output port is associated with a multiplexer. Total 5 multiplexers are presents in the crossbar. Each input port can reach output port using multiplexers. If a permanent fault occurs in that multiplexer, then the path to reach that specific port is blocked because there exists the only path to reach each output port. This permanent fault results in blocking of the flits trying to reach that port. To tolerate a permanent fault in the crossbar, we modified the baseline crossbar in such a way that results in better fault protection by creating two paths for each input port to reach an output port. The modified crossbar is shown in Fig. 13.

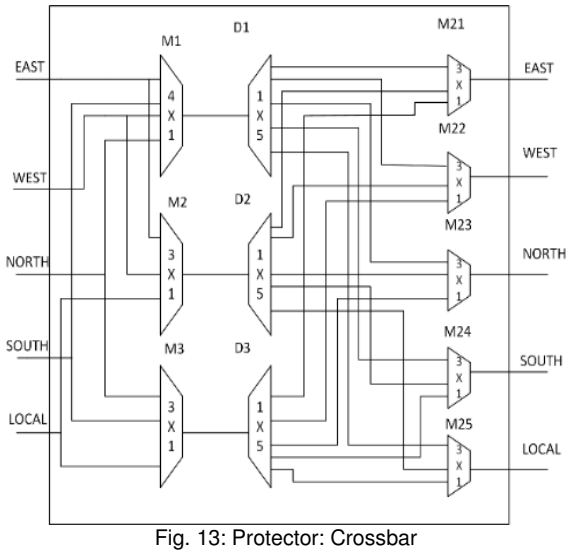

Fig. 13: Protector: Crossbar

There are total three levels of multiplexers in the proposed crossbar design. Level 1 contains one 4:1 mux and two 3:1 mux. Level 2 contains three 1:5 de-mux. Level 3 contains a total of five 3:1 mux. It is evident that each input port can reach each output port using two paths. Three extra fields are included in the modified input port architecture named as SP1, SP2, and SP3. These three extra fields are used to select which of the path is used by the input port to each output port. Consider Local input port wants to transmit a flit to the East output port. There exist two paths using M2, D2, M21 and another path is from M3, D3, and M21. Each of the input port has a default path to reach output port.

The default path values for Local input port for SP1 is to select local port using M2, SP2 is to send output through D2 which goes into M21 and SP3 is to select input which came from D2 to East output port. If a fault exists either in M2 or D2, then East output port becomes inaccessible for default local input port. To tolerate a fault in default path alternative path is chosen which is to transmit a flit through M3, D3, and M21. Transmitting a flit through the alternative path requires SP1, SP2 and SP3 fields to be updated. The SP1 field is now modified to select local input from M3, SP2 is modified to send D3 output to the line which goes in M21 and SP3 is updated to select input which came from D3. These fields are updated as the fault control unit finds out that output port becomes

inaccessible. Thus, in this way the alternative path is selected to access the output port.

5. EXPERIMENTAL RESULTS

In this section, we present the performance analysis of the Protector with state-of-the-art permanent fault tolerant router architecture concerning area overhead, latency overhead, and reliability.

5.1 Synthesis Results of Protector

For analysis, both baseline and the proposed router is implemented in Verilog HDL and synthesized using Cadence Encounter RTL compiler at 45 nm technology. In this work, we utilized XY routing. The XY routing algorithm is chosen because of its simplicity and low-cost implementation. Fault detection mechanism of NoC alert [46] is incorporated in the network to detect faults in the Protector. The results after implementing fault detection mechanism in the network reveal that Protector incurs an area overhead of 30%. Fig 13 presents an overhead area comparison of Protector with other state of the art permanent fault tolerant router architectures.

5.2 Protector Reliability Analysis

Different metrics may be utilized to determine the reliability of the proposed design. The area plays a vital role in fault tolerance efficiency. Spatial redundancy can be used to increase the reliability of the architecture towards faults at the cost of more considerable area overhead. Thus, such a metric which consider both areas and fault tolerance capability can be very useful. For comparing the reliability of the proposed architecture with the existing fault tolerant routers, SPF [27] is considered.

Silicon protection factor can be obtained using Equation (1):

$$\text{SPF} = \frac{\text{Mean No of faults to cause router failure}}{\text{Area Overhead}} \quad (1)$$

where Mean No faults cause to router failure is obtained from Equation (2):

$$\text{Mean No of faults} = \text{Minimum fault to failure} + \text{Maximum fault to failure} \quad (2)$$

where area overhead is obtained by Equation (3):

$$\text{Area Overhead} = \frac{\text{Fault tolerant design area}}{\text{Baseline Area}} \quad (3)$$

In Equation (1), normalization with area overhead is performed because as the area overhead increases, the number of gates in the circuit also increases. More gates in the circuits imply that design faces a high number of faults. Higher the value of SPF means that design has high fault tolerance towards permanent faults. In this work, we calculated SPF with each input port consists of 4 VC buffers. The overall SPF is calculated by considering faults tolerated by each stage separately.

Input Port: The fault tolerance for the VC buffers, mux and de-mux and RC unit is achieved by grouping the adjacent ports. For the proposed (2,2,1) pairing for the input port, in the worst-case scenario, if a fault occurs insides the local port de-mux or mux it causes in the failure of the proposed router architecture. Faults can happen in all units of the ports which are paired together. One paired group includes 4 VC buffers, RC unit, DRS module, Mux, De-mux and 3 VC buffers faults in an adjacent port. In this way, one paired group can tolerate total 11 faults. The local input port can tolerate maximum 3 VC buffers faults. The router can tolerate maximum 25 ((11x2) +3) faults. The minimum number of faults to cause failure is 1. Permeant fault in local input port mux or de-mux or RC unit fails the router.

RC Stage: The fault protection for the RC unit is achieved with the help of grouping ports in the form (2,2,1). The proposed grouping schemes shared RC unit within the group. If a permanent fault manifests in one of RC unit of paired port, then fault free RC unit is shared in paired ports. In this way, the router can tolerate a maximum 1 fault in each paired group. In the best case, the router can tolerate maximum 2 RC faults, 1 in each of the paired group. The local port is not paired with any of directional port and remains alone thus RC protection is not provided for the local port. Thus, a minimum number of faults to cause failure of the router is 1.

VA Stage: Fault protection for the VA stage is achieved with the help of borrowing arbiters within the paired port. If the arbiters associated with a VC buffer are considered faulty, then it can use arbiters of other

VC buffers available in that port or arbiters of the paired input port. There are 8 VC buffers in a paired group and 4 VC buffers in the local input port. So, a packet in one VC buffer can borrow arbiter from other seven arbiters in case of the paired group. In case of local port, it can borrow arbiters from other 3 VC buffers because the local port is not paired with any of the directional ports. Maximum 7 VA faults can be tolerated in a single paired group and 3 VA faults in the local input port. Thus, the proposed router can tolerate a maximum 17 ((7x2) +3) faults in the VA stage. In the worst-case scenario, if consecutive 4 arbiters faults manifest in the local input port, then the router cannot tolerate these faults. Thus, a minimum number of faults to cause router failure in VA stage is 4.

SA Stage: The SA stage is performed in two sub-stages. The protection strategies for tolerating a permanent fault in SA stages is achieved by creating multiple bypass paths in the first stage of the SA. Modified crossbar architecture tolerates faults in the second stage of switch allocator. Faults in the second stage of the switch allocator result in blocking the path to reach the output port. Our proposed crossbar design tolerates this fault by creating multiple paths to reach the output port. There are total 5 arbiters in the first stage of switch allocator which are paired in the form of (2,2,1) groping. In each paired group, the router can tolerate maximum 4 faults. In local port, the router can tolerate maximum 1 fault. In the best case, the router can tolerate maximum 9 ((4X2)+1) fault. The minimum number of faults to cause failure of the router is 2, as local port remains alone.

XB Stage: The proposed crossbar design creates two paths for each input port to reach the output port. The VC buffer use default path for transmitting a flit if the fault is not present. If a permanent fault manifests in the default path, then the alternative route is used to access output port. For example, Local input port can access the East output port through, M2, D2, M21 and through M3, D3, and M21. The default path is through M2 and D2. Faults in M2, D2 can be tolerated by updating the status register fields to choose a path through M3 and D3. The minimum number of fault to cause failure is also 2.

5.3 SPF of the Protector

The minimum number of fault to cause the router failure is selected by taking a minimum number of faults to cause failure among all the input port unit and pipeline stages. The minimum number of faults to cause failure of the router is 1 in our proposed protector architecture. The maximum number of faults to cause failure of the router is calculated by taking the sum of all the faults tolerated by each protection strategy separately. The sum of all faults becomes 25(Input port and RC) + 17 (VA) + 9 (SA) + XB (2) =53 faults. 53 is the maximum number of faults tolerated by router architecture. One more fault results in router failure. So, the maximum number of faults to cause failure of the router is 53+1=54. Thus, the mean number of faults to cause failure of the router is (54+1)/2=27.5 faults. The area overhead incurred is 30 percent. Thus, using Eq. (1), the SPF of the Protector can be calculated as 27.5/1.30 = 21.15.

5.4 Results and Discussion

We compare our proposed router design with other fault tolerant router architectures Bullet Proof [27], DRS [36], Vicis [28], shield [38], PVS router [35] and RoCo [29] by SPF, area overhead and mean no. of faults to cause failure. The comparison of the proposed router architecture by area overhead, the mean number of faults, and SPF with existing methodology is shown in Figs. 14-16.

The Proposed router architecture protect both buffers and pipeline stages. Thus, we compared our design with all architectures which worked on buffers and pipeline stages of the router. The bulletproof has chosen different design configuration for the router. The DRS and PVS architectures protect only buffers. Vicis, shield, and RoCo provide permanent fault protection for the router architecture. RoCo is Row-Column decoupled router architecture. It uses 2X2 smaller crossbars instead of 5X5 crossbar. These 2X2 crossbars decoupled the Row and Columns of the network and the router continues to work if one of the crossbar is faulty. RoCo only gives protection to crossbars and utilized the concept of look-ahead routing for RC faults. That's why the area overhead is very low as compared to our technique that provide full protection to all the components in the router. SPF

value of RoCo is less then 5.5 because it can only handle 5.5 mean no. of faults. Protector can handle 27.5 mean no. of faults as shown in Fig 14. The SPF value of protector is 21.15 which is much more than RoCo architecture. As shown in Fig. 14, our proposed router architecture Protector incurs the fourth lowest area overhead as compared to other existing methodologies. The DRS achieve lowest area overhead, but it protects the buffers only. The permanent faults in the pipeline stages are not tackled which will results in failure of the router if permanent fault manifests in the router pipelines. Thus, DRS is not a reliable architecture. Our proposed router Protector incurs 30 percent area overhead but results in reliable architecture towards permanents faults in any portion of the router architecture. The protector provides fault protection for both buffers and pipeline stages thus it is more reliable than state of the art architectures available.

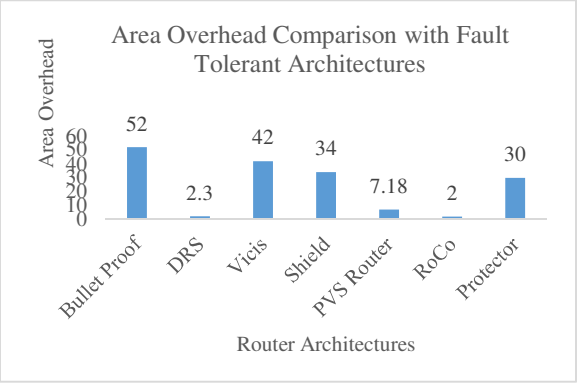

Fig. 14: Comparison of Area Overhead

Fig. 15 represents the comparison of Protector with other state of the art by the mean number of faults to cause failure. Bulletproof used different design configuration to achieve a most substantial mean number of fault to cause router failure. Thus, we consider two design configurations of the bulletproof router. Our proposed router architecture Protector achieve a second highest mean number of faults to cause failure as compared to methodologies available. The protector achieves 27.5 mean number of faults to cause failure at the cost of 30% area overhead. The bulletproof achieves 38 mean number of faults to failure at the cost of 242% area overhead. Protector provides better trades of the condition among a mean number of faults and area overhead.

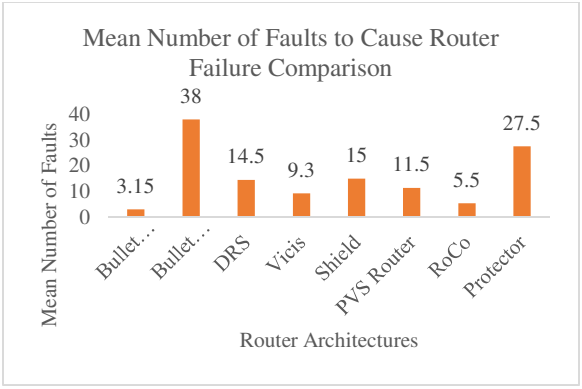

FIG. 15: COMPARISON OF MEAN NUMBER OF FAULTS

Fig. 16 provides a comparison with state of the art by the SPF. The highest value of the SPF indicates that design is more reliable towards permanents faults at the cost of less area overhead. The Protector achieves the highest value of the SPF 21.5, which is highest among all state of the art. The highest value of the SPF of proposed design indicates that Protector provides better trades of among area and fault protection. Thus, we conclude that Protector achieves better reliability than existing methodologies.

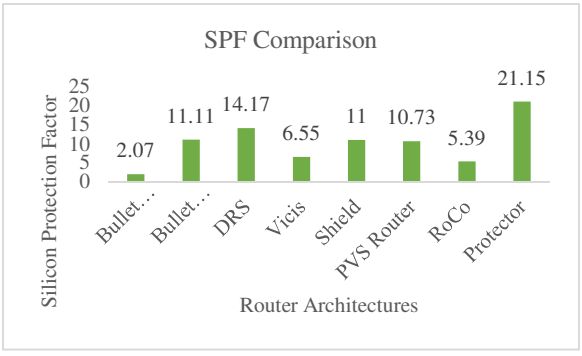

Fig. 16: Comparison of Silicon Protection Factor (SPF)

5.5 Lifetime improvement estimation using MTTF

The Mean Time to Failure (MTTF) is the estimated time a device lasts in operation. MTTF is an important metric to measure the reliability of the hardware. Equation 4 can be used to calculate the MTTF of a given piece of hardware.

$$\frac{10^9}{\text{Failure-in-Time(FIT)}} \quad (4)$$

where FIT is the failure of operation of a given component per billions of hours. Failure in Time

estimation model proposed by Paluri *et al.* [48] can be used to find out the FIT of the router. Shin *et al.* [49] proposed a lifetime modelling framework which is also utilized for the calculation of FIT. We utilized the time dependent dielectric breakdown model for measuring the FIT. To find out the value of FIT for single Field Effect Transistor (FET) Equation (5) [50] can be used as given.

$$\mathrm{FIT}_{\text{per FET}} = \text{dutycycle} \times \frac{10^9}{A_{TDDB}} \times \mathrm{Vdd}^{a-bT} \times e^{\frac{-x+\frac{Y}{T}+ZT}{kT}} \quad (5)$$

where A_{TDDB}, a, b, X, Y, Z are the fitting parameters and k is the Boltzmann's constant. T is the temperature of 300 kelvin and Vdd is the operating voltage of 1V. Dutycycle in Equation 5 is selected to be 100% for the calculation of FIT. So, the FIT value of a basic logic gate can easily be calculated by multiplying the transistor count with the $\mathrm{FIT}_{\text{per FET}}$. The Sum of Failure (SOFR) model presented in [51] can be used to find out the FIT of a component and then the entire router. FIT estimation of the baseline router is given in the Table 1.

Table 1: Fit Estimation of Baseline NoC Router

Unit	FC	FIT of FC	No. of FC	FIT of the Unit
Input Buffer	32-bit DFF	0.25	40960	10240
RC	6-bit Comparator	11.7	10	117
VA	4:1 Arbiter 20:1 Arbiter	4.7 36.7	100 20	1478
SA	4:1 Mux 4:1 Arbiter 5:1 Arbiter	4.8 7.4 9.3	25 5 5	203
XB	32-bit 5:1 Mux	204.8	5	1024

5.5.1 FIT Estimation of the Reliable Router: Protector

Input Port: In the proposed router protector the input port is protected from faults by grouping the neighboring ports. Flits for the faulty port can be sent to the neighboring port to tolerate the faults. This fault protection mechanism does not need any extra circuitry.

RC Unit: RC is responsible for calculating the output port for the incoming header flits. In our protection mechanism extra circuitry is not needed for the fault tolerance of RC unit. Flits can utilize the RC of the neighboring port in case of faults.

VA Unit: VA is protected by virtual channel sharing among the input port and the neighboring port. This fault protection strategy is achieved by adding the 20 3-bit DFF to store the R2 field, 20 1-bit DFF for VF filed and 20 2-bit DFF for the ID field.

SA Unit: Default winner strategy is utilized to provide the fault tolerance at this stage of the router. Extra circuitry of 60 3-bit DFF for storing the SP1, SP2 and SP3 fields are required for this technique. For the selection and default winner 5 2-bit DFF registers and 5 3:1 muxes are also required.

XB Unit: Crossbar is used in the router to connect the input port to the output port. Fault tolerance at this stage is provided by adding the redundant paths to reach the output port. For this protection strategy the 5 3:1 muxes are required.

Table 2 shows the FIT value and the extra components utilized in the reliable router protector.

Table 2: FIT Estimation of the Reliable Router: Protector

Unit	Component	FIT value of the Unit
VA	20 3-bit DFF('R2'), 20 1-bit DFF('VF')	60
	20 2-bit DFF('ID')	
SA	60 3-bit DFF('SP1','SP2','SP3')	94
	5 2-bit DFF(Register),5 3:1 muxes	
XB	5 32 bit 3:1 muxes	1652.8

5.5.2 MTTF of Proposed NoC Router: Protector

The MTTF value of the baseline NoC router can be calculated by the SOFR model using Equation (6).

$$\mathrm{MTTF}_{\text{Baseline}} = \frac{10^9}{10240+117+1478+203+1024} \approx 76{,}557.85 \text{ hours} \quad (6)$$

FIT of the reliable router protector calculated by the SOFR model is 60 + 94 + 1652.8 = 1806.8 Equation (7) can be used to find out the MTTF value of the reliable router utilizing the above data.

$$\mathrm{MTTF}_{\text{Reliable router}} = \frac{10^9}{\mathrm{FIT}_1} + \frac{10^9}{\mathrm{FIT}_2} + \frac{10^9}{\mathrm{FIT}_1 + \mathrm{FIT}_2} \quad (7)$$

Here the FIT_1 is the FIT value of the baseline unprotected router (13,062) and FIT_2 is the FIT value of the reliable router (1806.8) protector. Hence, the MTTF value of the Protector is 697,275 which is 9.1 times to the baseline router. So, our reliable router Protector is 9.1 times more reliable than the baseline router. Shield [38] is state of the art reliable router which is 6 times more reliable to its baseline unprotected router.

5.6 Latency Analysis

In this section, we discussed the performance of the Protector from the load vs. latency point of view. The fault model affects the design policy of the fault tolerant router architecture. For the evaluation of the Protector fault tolerant router, we assume the occurrence of the single event upset faults in the router architecture. Specifically, Single bit permanents faults are injected in the router architecture at different possible locations and during different pipeline stages.

For latency analysis, we simulate the architecture for both synthetic traffic and benchmark application. We simulate the network consist of Protector router using GEM5 [52] simulator. The generic primary router is simulated using GARNET [53]. We modified the baseline architecture according to our proposed router requirements. The input ports are paired in the form of (2,2,1) grouping along with modification for the pipeline stages of the router.

Simulating the network for synthetic traffic pattern, we chose 8x8 mesh-based NoC with uniform random synthetic traffic patterns and tornado traffic pattern. For analysis of load vs. latency, we inject uniform random synthetic traffic at various injection rates, range from (0.01 to 0.1 packets/node/cycle). Each packet consists of 5 flits where the size of each flit is 16 bytes. The link latency is assumed to be one. Each simulation runs for the 500,000 cycles, and each injection rate simulation is repeated 10 times and an average value is taken. The average latency is calculated using Equation (8):

$$\text{Average latency} = \frac{\text{Total network latecny}}{\text{Total number of flits received}} \qquad (8)$$

After the calculation of the baseline results, we simulate Protector for the same configuration. To simulate the faults, we inject faults based on the uniform random number of variable. A fault is injected in buffers and pipeline stages of the router during runtime system operation. Due to faults in the pipeline and input port architecture, the Protector completes its execution using proposed protection strategy. The Figs. 17-18 show the results of latency overhead for uniform random and tornado traffic pattern. For the uniform and tornado traffic pattern, latency is increased by 7% and 5% respectively. For benchmark traffic, we simulated 8x8 mesh NoC using GEM5. For each core, separate cache and directory are used, and for coherence purpose, MOESI CMP directory is used. Figs. 19-20 show the latency comparison for the SPLASH-2 [54] and PARSEC [55] benchmarks. For SPLASH-2 and PARSEC, protector incurs a latency overhead of 16% and 13% respectively.

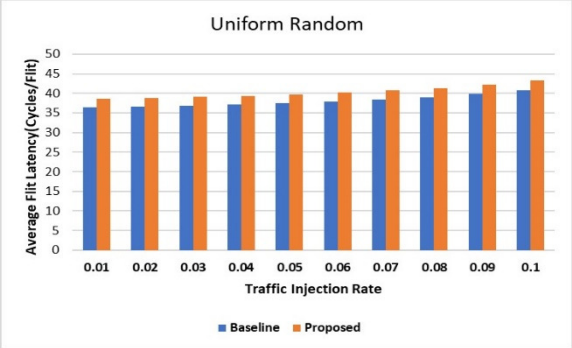

Fig. 17: Load Vs. Latency for Uniform Traffic

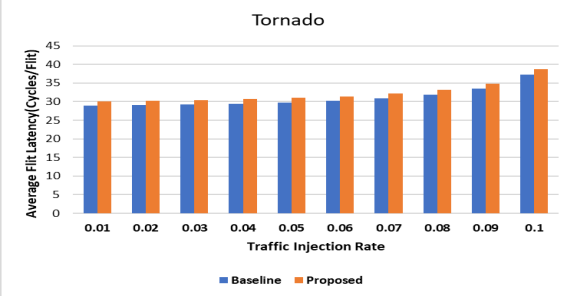

Fig. 18: Load Vs. Latency for Tornado Traffic

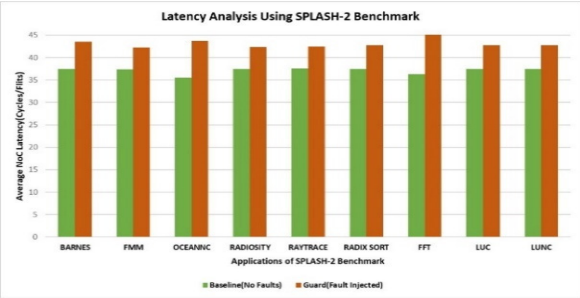

Fig. 19: Latency Analysis for Splash

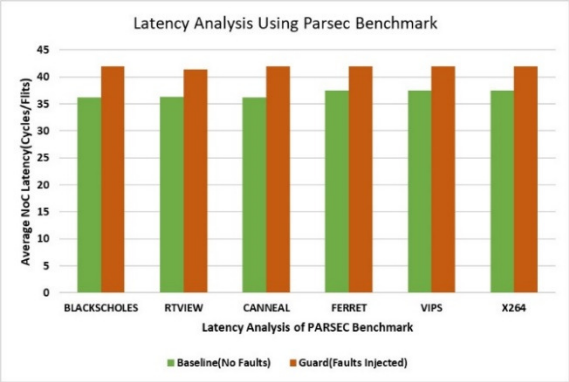

FIG. 20: Latency Analysis for PARSEC

The proposed methodologies involve better reliability with minimum overhead. The synthesis of the proposed design discloses that enhancement in the router architecture resulted in area overhead of 30%. From the perspective of reliability using SPF, we showed that Protector achieves highest SPF among all other existing fault-tolerant architecture. The evaluation results show that Protector achieves second lowest area and highest mean number of faults to failure and maximum fault coverage as compared to state-of-the-art methods available.

6. CONCLUSIONS

We propose a permanent fault tolerant router architecture for NoC. It uses diverse fault resilience strategies for input buffers and pipelined stages (RC, VA, SA and Xbar). Reliability analysis using SPF metric reveals that the proposed design achieves SPF of 21.5 which is highest as compared to the state of art architectures available. The higher value of SPF suggests that the proposed design provides better reliability with less overhead. In the presence of faults, the proposed design incurs 13 and 16% latency overhead for PARSEC and SPLASH-2 benchmarks. Synthesis results reveal that the proposed router incur area overhead of 30% as compared to the baseline router.

7. FUTURE WORK

In future we are planning to tolerate the transient faults on links and network interfaces which are used to connect the routers by using the ECC techniques. This would allow us to design more efficient network with better fault protection strategies for all kind of faults occurring in NoC.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their valuable feedback. This work is supported by Digital Systems Laboratory in University of Engineering and Technology Taxila, Pakistan.

REFERENCES

[1] Borker S., "Thousand core chips: a technology perspective", *Proceedings of the 44th Annual Design Automation Conference*, pp. 746-749, USA, 2007.

[2] Damaraju S., George, V., Jahagirdar, S., Khondker, T., Milstrey, R., Sarkar, S., Siers, S., Stolero, I., Subbiah, A., "A 22nm IA multi-CPU and GPU system-on-chip", *Proceeding of the IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC)*, pp. 56-57, USA, 2012.

[3] Krishnan V., Torrellas J., "A chip-multiprocessor architecture with speculative multithreading", *IEEE Transactions on Computers*, Vol. 48, No. 9, pp.866-880, USA, 1999.

[4] Benini L., De Micheli G., "Networks on chips: A new SoC paradigm", *Computer*, Vol. 35, pp.70-78, USA, 2002.

[5] Kumar S., Jantsch A., Soininen J.P., Forsell M., Millberg M., Oberg J., Tiensyrja K., Hemani, A., "A network on chip architecture and design methodology", *Proceedings of IEEE Computer Society Annual Symposium on VLSI, New Paradigms for VLSI Systems Design*, pp. 117-124, USA, 2002.

[6] Dally W.J., Towles B., "Route packets, not wires: on-chip interconnection networks", *Proceedings of the 38th annual Design Automation Conference*, pp. 684-689, USA, 2001.

[7] Borkar S., "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation", *IEEE Micro*, Vol. 25, pp. 10-16, USA, 2005.

[8] Constantinescu C., "Trends and challenges in VLSI circuit reliability", *IEEE Micro*, pp.14-19,

USA, 2003.

[9] Dodd P.E., and Massengill L.W.," Basic mechanisms and modeling of a single-event upset in digital microelectronics", *IEEE Transactions on Nuclear Science*, Vol. 50, pp. 583-602, USA, 2003.

[10] Baumann R., "Soft errors in advanced computer systems", *IEEE Design and Test of Computers*, Vol. 22, pp. 258-266, USA, 2005.

[11] Shivakumar P., Kistler M., Keckler S.W., Burger D., Alvisi L., "Modeling the effect of technology trends on the soft error rate of combinational logic", *Proceedings of the International Conference on Dependable Systems and Networks*, pp. 389-398, USA, 2002.

[12] Zhang M., Shanbhag N.R., "Soft-error-rate-analysis (SERA) methodology", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, pp.2140-2155, USA, 2006.

[13] Cuviello M., Dey S., Bai X., Zhao, Y., "Fault modeling and simulation for crosstalk in system-on-chip interconnects", *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, pp. 297-303, USA, 1999.

[14] Walker M.G., "Modeling the wiring of deep submicron ICs", *IEEE Spectrum*, Vol. 37, pp.65-71, Canada, 2000.

[15] Yang Y., *Issues of ESD Protection in Nano-Scale CMOS*, PhD Dissertation, George Mason University, Fairfox, VA, USA, 2010.

[16] Keane J., Kim C.H., "An Odometer For CPUs: Microprocessors don't normally show wear and tear, but wear they do", *IEEE Spectrum*, Vol. 48, pp. 26-31, Canada, 2011.

[17] Wittmann R., Puchner H., Hinh L., Ceric H., Gehring A., Selberherr S., "Simulation of dynamic NBTI degradation for a 90 nm CMOS Technology", *Materials Science*, 2005.

[18] Zhang, B. and Orshansky, M., "Modeling of NBTI-induced PMOS degradation under arbitrary dynamic temperature variation", *Proceedings of the 9th International Symposium Quality Electronic Design (ISQED)*, pp. 774-779, 2008.

[19] Takeda, E., Yang, C.Y.W. and Miura-Hamada, A., "*Hot-Carrier Effects in MOS Devices*," Academic Press, 1995.

[20] Agarwal, M., Paul, B.C., Zhang, M. and Mitra, S., "Circuit failure prediction and its application to transistor aging," *Proceeding of 25th IEEE in VLSI Test Symposium*, pp. 277-286, USA, 2007.

[21] Aisopos, K., DeOrio, A., Peh, L.S. and Bertacco, V., "Ariadne: Agnostic reconfiguration in a disconnected network environment," *Proceedings of International Conference in Parallel Architectures and Compilation Techniques (PACT)*, pp. 298-309, USA, 2011.

[22] Bertozzi, D., Benini, L. and De Micheli, G., "Error control schemes for on-chip communication links: the energy-reliability tradeoff," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 24, No. 6, pp.818-831, USA, 2005.

[23] Fick, D., DeOrio, A., Chen, G., Bertacco, V., Sylvester, D. and Blaauw, D., "A highly resilient routing algorithm for fault-tolerant NoCs," *Proceedings of Design, Automation, and Test in Europe*, pp. 21-26, France, 2009

[24] Kodi, A.K., Sarathy, A. and Louri, A., "Adaptive channel buffers in on-chip interconnection networks-A power and performance analysis," *IEEE Transactions on Computers*, Vol. 57, pp.1169-1181, USA, 2008

[25] Lin, S., Shi, J. and Chen, H., "Designing cost-effective network-on-chip by a dual-channel access mechanism," *Journal of Systems Engineering and Electronics*, Vol. 2, pp.557-564, 2011.

[26] Valinataj, M., Mohammadi, S., Plosila, J., Liljeberg, P. and Tenhunen, H., "A reconfigurable and adaptive routing method for fault-tolerant mesh-based networks-on-chip," *AEU-International Journal of Electronics and Communications*, Vol. 65, pp.630-640, 2011.

[27] Constantinides, K., Plaza, S., Blome, J., Zhang, B., Bertacco, V., Mahlke, S., Austin, T. and Orshansky, M., "Bullet Proof: A defect-tolerant CMP switch architecture," *Proceeding of The Twelfth International Symposium on High-Performance Computer Architecture*, pp. 5-16, USA, 2006.

[28] Fick, D., DeOrio, A., Hu, J., Bertacco, V., Blaauw, D. and Sylvester, D., "Vicis: a reliable network for unreliable silicon," *Proceedings of 46th Annual Design Automation Conference*, pp. 812-817, California, 2009.

[29] Kim, J., Nicopoulos, C., Park, D., Narayanan, V., Yousif, M.S. and Das, C.R., "A gracefully degrading and energy-efficient modular router architecture for on-chip networks," *ACM SIGARCH Computer Architecture News*, Vol. 34, No. 2, pp.4-15, USA, 2006.

[30] Koibuchi, M., Matsutani, H., Amano, H. and Pinkston, T.M., "A lightweight fault-tolerant mechanism for network-on-chip," *Proceeding of Second ACM/IEEE International Symposium on Networks-on-Chip*, pp. 13-22, USA, 2008.

[31] Feng, C., Lu, Z., Jantsch, A., Zhang, M. and Xing, Z., "Addressing transient and permanent faults in NoC with efficient fault-tolerant deflection router," *IEEE Transactions on Very Large-Scale Integration (VLSI) Systems*, Vol. 21, pp.1053-1066, USA, 2013.

[32] Xie, L., Mei, K. and Li, Y., " Repair: A reliable partial-redundancy-based router in NoC.", *Proceeding of IEEE Eighth International Conference in Networking Architecture and Storage (NAS)*, pp. 173-177, China, 2013

[33] Neishaburi, M.H., and Zilic, Z., "ERAVC: Enhanced reliability-aware NoC router," *Proceedings of 12th International Symposium in Quality Electronic Design (ISQED)*, pp. 1-6, 2011

[34] Neishaburi, M.H., and Zilic, Z., "NISHA: A fault-tolerant NoC router enabling deadlock-free Interconnection of Subnets in Hierarchical Architectures," *Journal of Systems Architecture*, Vol. 59, pp. 551-569, USA, 2013

[35] Latif, K., Rahmani, A.M., Nigussie, E., Seceleanu, T., Radetzki, M. and Tenhunen, H., "Partial virtual channel sharing: a generic methodology to enhance resource management and fault tolerance in networks-on-chip," *Journal of Electronic Testing*, Vol. 29, pp. 431-452, 2013

[36] Valinataj, M. and Shahiri, M., "A low-cost, fault-tolerant and high-performance router architecture for on-chip networks," *Microprocessors and Microsystems*, Vol. 45, pp.151-163, 2016.

[37] Poluri, P. and Louri, A., 2013," Tackling permanent faults in the network-on-chip router pipeline," *Proceedings of 25th International Symposium on Computer Architecture and High-Performance Computing (SBAC-PAD)*, pp. 49-56, Brazil, 2013.

[38] Poluri, P. and Louri, A.," Shield: A reliable network-on-chip router architecture for chip multiprocessors," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 27, pp.3058-3070, USA, 2016.

[39] Bahrebar, P., Jalalvand, A., and Stroobandt, D., "Dynamically reconfigurable architecture for fault-tolerant 2D Networks-on-Chip", *Proceedings of 26th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1-7, Canada, 2017.

[40] Yuan, C., Huang, L., Wang, J. and Li, Q., "Micro-Architecture Design for Low Overhead Fault Tolerant Network-on-Chip," *Proceeding of IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1-5, Italy, 2018.

[41] Moriam S, Fettweis GP. "Reliability assessment of fault-tolerant routing algorithms in networks-on-chip: an analytic approach", *Proceedings of the Conference on Design, Automation and Test in Europe*, pp. 61-66, Switzerland, 2017

[42] Khalil K, Eldash O, Bayoumi M. "Self-healing router architecture for reliable network-on-chips", *Proceedings of 24th*

IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 330-333, Georgia, 2017

[43] Dally, W.J. and Towles, B.P., "Principles and practices of interconnection networks," Elsevier, 2004.

[44] Collet, J.H., Louri, A., Bhat, V.T. and Poluri, P., "ROBUST: a new self-healing fault-tolerant NoC router," *Proceedings of 4th International Workshop on Network on Chip Architectures*, pp. 11-16, USA, 2011.

[45] Peh, L.S. and Dally, W.J., "A delay model and speculative architecture for pipelined routers," *Proceedings of The Seventh International Symposium on High-Performance Computer Architecture (HPCA)*, pp. 255-266, Mexico, 2001.

[46] Prodromou, A., Panteli, A., Nicopoulos, C. and Sazeides, Y., "Noc alert: An online and real-time fault detection mechanism for network-on-chip architectures," *Proceedings of 45th Annual IEEE/ACM International Symposium on Microarchitecture*, pp. 60-71, Canada, 2012.

[47] Fu, B. and Ampadu, P., "*Error Control for Network-on-Chip Links*," Springer Science and Business Media, 2011.

[48] Poluri P., A. Louri, An improved router design for reliable on-chip networks, *Proceedings of 28th IEEE International Symposium on Parallel and Distributed Processing*, pp. 283–292, USA, 2014

[49] J. Shin, V. Zyuban, Z. Hu, J.A. Rivers, P. Bose, A framework for architecture-level lifetime reliability modeling, *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, Edinburgh, pp. 534–543, UK, 2007

[50] Oussalah S., F. Nebel, On the oxide thickness dependence of the time-dependent-dielectric breakdown, *IEEE Proceedings of Electron Devices Meeting*, pp. 42–45, Hong Kong, 1999

[51] Williams T., Probability and statistics with reliability, queueing and computer science applications, Journal of Operations Research, Vol. 34, pp. 916–917, 1983.

[52] Binkert, N., Beckmann, B., Black, G., Reinhardt, S.K., Saidi, A., Basu, A., Hestness, J., Hower, D.R., Krishna, T., Sardashti, S. and Sen, R., "The gem5 simulator", ACM SIGARCH Computer Architecture News, Vol. 39, pp.1-7, USA, 2011

[53] Agarwal, N., Krishna, T., Peh, L.S. and Jha, N.K., "GARNET: A detailed on-chip network model inside a full-system simulator.", *Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 33-42, USA, 2009.

[54] Woo, S.C., Ohara, M., Torrie, E., Singh, J.P. and Gupta, A., "The SPLASH-2 programs: Characterization and methodological considerations", *ACM SIGARCH Computer Architecture News*, Vol. 23, No. 2, pp. 24-36, 1995.

[55] Bienia, C., Kumar, S., Singh, J.P. and Li, K., "The PARSEC benchmark suite: Characterization and architectural implications", *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, pp. 72-81. ACM, Canada, 2008.