electronics

MDPI

Article

Savior: A Reliable Fault Resilient Router Architecture for Network-on-Chip

Ayaz Hussain [1], Muhammad Irfan [2], Naveed Khan Baloch [3], Umar Draz [4,*], Tariq Ali [5,*], Adam Glowacz [6,*], Larisa Dunai [7] and Jose Antonino-Daviu [8]

1 Department of Computer Science, University of Management and Technology Sialkot, Sialkot 51310, Punjab, Pakistan; ayaz.hussain@skt.umt.edu.pk
2 College of Engineering, Electrical Engineering Department, Najran University, Najran 61441, Saudi Arabia; irfan16.uetian@gmail.com
3 Department of Computer Engineering, University of Engineering and Technology, Taxila 47050, Pakistan; naveed.khan@uettaxila.edu.pk
4 Department of Computer Science, University of Sahiwal, Sahiwal, Punjab 57000, Pakistan
5 Department of Computer Science, COMSATS University Islamabad, Sahiwal Campus, Sahiwal 57000, Pakistan
6 Department of Automatic Control and Robotics, Faculty of Electrical Engineering, AGH University of Science and Technology, Automatics, Computer Science and Biomedical Engineering, al. A. Mickiewicza 30, 30-059 Kraków, Poland
7 Centro de Investigación en Tecnologías Gráficas, Universitat Politècnica de València, 46022 Valencia, Spain; ladu@upv.es
8 Instituto Tecnologico de la Energía Camino de Vera s/n, Universitat Politecnica de Valencia, 46022 Valencia, Spain; joanda@die.upv.es
* Correspondence: sheikhumar520@gmail.com (U.D.); tariqhsp@gmail.com (T.A.); adglow@agh.edu.pl (A.G.)

Received: 31 August 2020; Accepted: 30 September 2020; Published: 27 October 2020

Abstract: The router plays an important role in communication among different processing cores in on-chip networks. Technology scaling on one hand has enabled the designers to integrate multiple processing components on a single chip; on the other hand, it becomes the reason for faults. A generic router consists of the buffers and pipeline stages. A single fault may result in an undesirable situation of degraded performance or a whole chip may stop working. Therefore, it is necessary to provide permanent fault tolerance to all the components of the router. In this paper, we propose a mechanism that can tolerate permanent faults that occur in the router. We exploit the fault-tolerant techniques of resource sharing and paring between components for the input port unit and routing computation (RC) unit, the resource borrowing for virtual channel allocator (VA) and multiple paths for switch allocator (SA) and crossbar (XB). The experimental results and analysis show that the proposed mechanism enhances the reliability of the router architecture towards permanent faults at the cost of 29% area overhead. The proposed router architecture achieves the highest Silicon Protection Factor (SPF) metric, which is 24.8 as compared to the state-of-the-art fault-tolerant architectures. It incurs an increase in latency for SPLASH2 and PARSEC benchmark traffics, which is minimal as compared to the baseline router.

Keywords: reliability; reconfigurable architecture; fault tolerance; network-on-chip; permanent faults

1. Introduction

The abundant availability of on-chip transistors, coupled with the desire to design low-power chips, that either maintain the same level of performance or improved performance as compared to their predecessors has led to the inception and rise of chip multiprocessors (CMPs). It results in a

paradigm shift from the design of computation-oriented architectures to communication-oriented architectures. Rapid technology scaling into deep submicron has facilitated the designer to fabricate the billions of transistors on a single chip [1]. The efficient handling of the communication in CMP has led to the inception of the Network on Chip (NoC) paradigm [2]. NoC constitutes an interconnection architecture of future and massively parallel multiprocessors that assemble hundreds of processing cores on a single chip [3]. Since 2000, the NoC has emerged due to the advent of multicore CPUs and the foreseeable trend towards massively integrated many-core architectures. Due to technology scaling, transistor size shrinks, resulting in vulnerabilities of the transistors and wires towards various faults [4]. Faults can be classified as permanents faults, transient faults, and intermittent faults [5]. A fault in the router causes a deadlock in the network, an increase in packet latency, or packet loss, which results in reduced performance of the system or may lead to system failure. Transient faults occur due to alpha particle strikes from packaging material, thermal radiations from cosmic rays, and process variations. The traditional causes of permanents faults are time-dependent dielectric breakdown (TDDB) [6], hot carrier injection, and electron migrations.

The transient faults remain in the circuit for a short duration of time, while the impact of the permanent faults exists for a large duration of time. The permanent faults may occur due to fabrications defects or the operation time of the circuit and continue to affect the functionality of the system. In this work, we have focused on providing a solution for permanent faults. The transient errors can be tackled by adopting retransmission [7] and multiple ways [8]. Previously proposed fault-tolerant methods are either based on the architectural modifications or simply a fault-tolerant based deflection routing algorithm [9]. The existing methods provide partial fault protection to pipeline stages or input port architectures. A detailed discussion of the state-of-the-art approaches is provided in the literature review section.

The main contributions of this paper can be summarized as follows:

1. To provide fault tolerance in all the stages of the router architecture.
2. Performance analysis of the proposed router architecture with state-of-the-art architectures based on finding the area overhead, and average latency.
3. Reliability analysis using the SPF metric.

The remaining sections of the paper are presented as follows. In Section 2, we describe the effects of faults on router architecture. In Section 3, the related work and problem statement are presented. In Section 4, the proposed permanent fault-tolerant router architecture is described in detail with fault tolerance capability to all the pipeline stages. In Section 5, results and discussions are provided. Section 6 concludes the whole work.

2. Impact of Faults on Router

In this section, the impact of faults on different components of the router is discussed. Figure 1 illustrates the baseline design of an NoC router architecture.

As shown in Figure 1, the input port consists of buffers, multiplexers and demultiplexers. In the case of demultiplexer failure, the flits cannot enter the router and all the resources are wasted. In the case of multiplexer failure, the flits cannot leave the specific port and remain in the buffer, which results in starvation, increased average latency and in some cases the deadlock may occur. So, it is very necessary to provide fault tolerance to this unit of the router.

The first pipeline stage is the RC unit which is used to find the output port for the packet according to some routing algorithm. If the RC unit is faulty, it may calculate the invalid or wrong output port. Valid output ports such as east, west, north, south and local are numbered as 0 to 4, respectively. The output of RC is invalid if the value is ≥5. In such cases, the packets remain in the same router and result in deadlock. For wrong output port selection, the packet moves away from the destination and results in increased average latency.

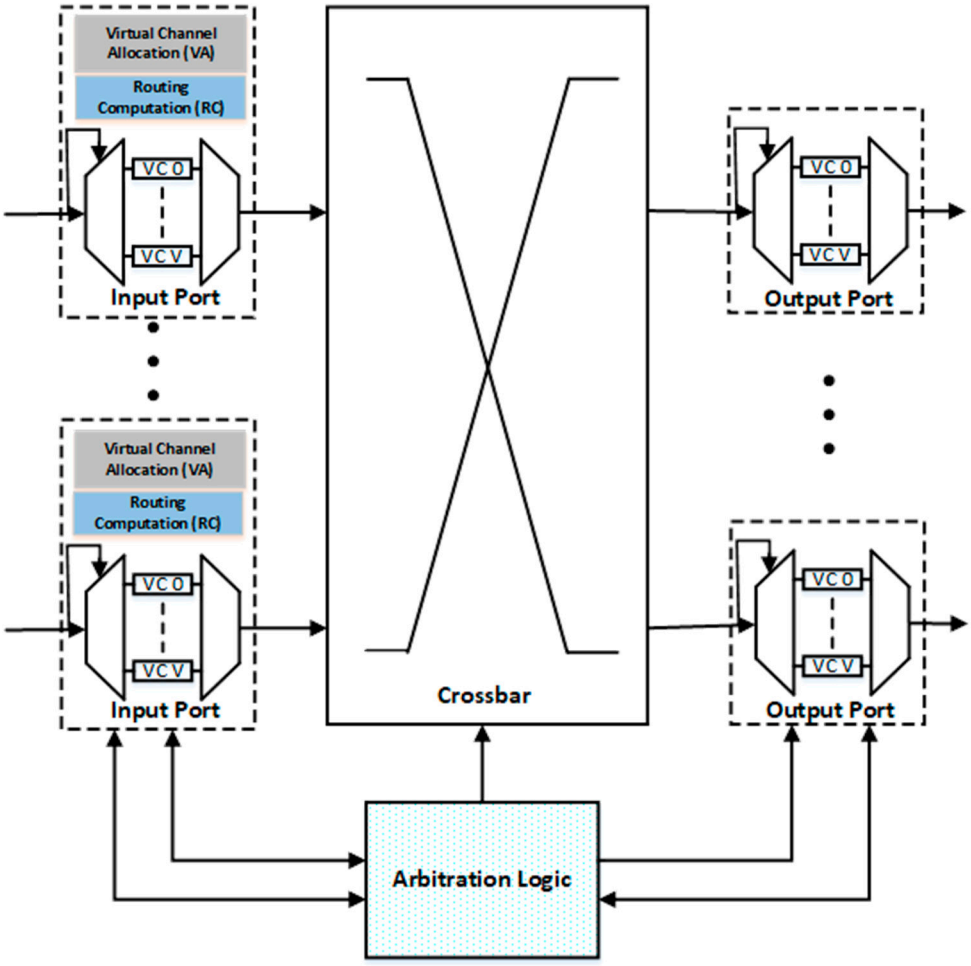

Figure 1. Baseline router architecture for Network on Chip.

The second pipeline stage is the VA unit which is used to assign the empty virtual channel (VC) buffer in the downstream router. A faulty VA unit may assign the occupied VC in the next router or never assign a VC. In case of the assignment of occupied VC, the data available in that VC are corrupted, and, in case of the permanent unsuccessful allocation of VC, the packet remains in the same buffer and cannot proceed to the next pipeline stages.

The third pipeline stage is the SA unit, which is used to assign the XB time to the selected VCs. In case of failure, VC cannot get access the XB to reach the output port and block in the same input port, which results in a deadlock. The last pipeline stage is XB unit which is used to connect the input ports to the output port. Fault in this unit prevents the packets to reach the output port and results in the deadlock.

3. Related Work

Permanent faults constantly impact the working of the router throughout the lifecycle. Researchers have proposed different solutions to tackle these faults. Various fault-tolerant deflection routing algorithms are utilized for the optimal selection of paths [10]. Poluri et al. [11] proposed the solution of tackling both transient and permanent faults in the pipeline of the router. The fault-tolerant techniques employed in the pipeline stages of the router are spatial redundancy, exploitation of idle cycles, bypassing faulty resources, and selective hardening. In [12], the authors have proposed a scheme based on a dynamic resource sharing approach that can tolerate soft errors in multiplexers, demultiplexers, and VCs of the input port unit. In [13], the author has proposed a router architecture named Vicis, which can tolerate permanent faults at both the network level as well as the router level.

Vicis has employed inherent redundancy in the router and in its network to maintain the correct operations of the router. For tolerating permanent faults in the router architecture, a port swapping algorithm is utilized as well as a bypass path for the crossbar to tolerate the defects in it. The interesting achievement of Vicis is that it has employed a distributed routing algorithm to avoid faults in the network. The author used an input port swapping algorithm and network rerouting techniques to improve the performance of the router. In [14], the author proposed a decoupled router architecture named RoCo, which disintegrates the router into the individual row and column. This architecture results in decoupled rows and columns having smaller crossbar and parallel arbiters. This modular approach results in the degraded performance of the network in case of permanent faults.

In [15], the author proposes a defect-tolerant CMP switch architecture named BulletProof. They employed a generic model of the bathtub curve for permeant fault models. This model described the permanent fault model's behavior up to 65 nm technology. For tolerating the permanent faults, a simplified approach used is the triple modular redundancy (TMR) approach [16]. In this approach, each component of the router architecture is duplicated or tripled depending upon the N-modular redundancy approach used. To improve the fault tolerance of the circuit, a selective hardening of the gate approach is utilized [17]. In a selective hardening of the gates, first, the critical gates are identified and resized, which lies on the critical path. One study focuses on tolerating permanent faults in the router's input port, particularly the virtual channel state fields [18], while some have worked on protecting all the stages of the router separately [19].

The HPR [19] has designed permanent fault protection techniques for an NoC router. Error-correcting codes are utilized for the protection of single-bit faults in the flits. They presented a concept of a double routing technique for the protection of RC faults. The VA in case of faults uses the default winning technique. The SA used the runtime arbiter selection approach for tolerating the faults. The XB design used a bypass path technique for tolerating the faults in this stage. The overall design achieved higher reliability but still, the design was not able to tolerate the multiple faults occurring in the multiplexers and demultiplexers of the input ports. The router fails if permanents faults occur in these components.

NocGuard [20] utilized the double routing strategy for RC, run time arbiter selection for VA, default winning strategies for the SA, and bypass path approaches for the XB stage. This work is designed to provide the protection techniques to only the pipeline stages to achieve higher reliability as compared to existing architectures. The baseline router architecture consists of 80% VCs, and providing fault protection for these components is inevitable.

If a permanent fault occurs inside the VC buffers, write signal or on the links of the router, the whole protection strategies will become useless because no packets can enter to this port due to faults. This results in a permanent deadlock situation. This is the major drawback of NoCGuard because it cannot provide protection to the input ports. A comparison of different router architectures proposed by researchers, techniques employed, and error tolerance capability is described in Table 1.

Different fault tolerance techniques were proposed in the state-of-the-art fault-tolerant router architecture design to handle the permanent faults occurring in the router. Bulletproof architecture, as discussed in the literature review, is based on spatial redundancy technique which results in larger area overhead. This architecture does not provide fault tolerance to the crossbar stage. RoCo architecture provides fault tolerance but results in performance degradation. This architecture cannot tolerate the VA and SA faults. The router fails if a fault occurs in these pipeline stages. Shield [11] results in better performance but results in larger area overhead. As the area increases, the fault probability of the circuit also increases. If two faults occur in the RC stage, then the router fails its operation. The HPR [19] and NoCGuard [20] provide the faults protection to pipeline stages but left the input port architecture. The study of the previous design revels that they protect permanent faults at the cost of larger area overhead and power consumption. There was a need to provide such an architecture that can tolerate maximum faults and results in smaller area overhead so that correction circuity does not result in increased fault probability. The proposed architecture solves these issues and provides

enhanced reliability as compared to the state-of-the-art architecture discussed in the literature review. The proposed router architecture is able to handle the faulty links, deadlock situation due to permanent fault occurring in the buffers and load management by providing resource sharing in the input port architecture. For each pipeline stage, the correction circuity is developed to tolerate the permanent faults. The details of each stage are discussed in Section 4.

Table 1. Comparison of the existing router architecture for fault protection capability.

Method	Techniques Employed	Errors Tolerance
STNR [21]	Spatial redundancy Temporal redundancy	Soft errors tolerance in RC, SA, and VA stages of the pipeline
Shield [11]	spatial redundancy, exploitation of idle cycles, bypassing faulty resources, and selective Harding.	Can handle both soft and permanents faults in RC, SA, VA, and XB stages of the pipeline.
BulletProof [15]	Resource sparing, automatic cluster decompositions	Fault-tolerant router architecture.
RoCo [14]	Decoupled router architecture	Provides degraded performance in case of permeant faults.
TMR [22]	Spatial redundancy	Depends upon the N-modular approach used for router components.
Vicis [23]	Adaptive routing, input port swapping	Network-level and router level
DRS [12]	Buffer sharing approach	Buffer errors, Multiplexer error, Demultiplexer
HPR [19]	Spatial redundancy, default winning strategies, default design for XB	RC, VA, XB, SA faults are tolerated
NoCGuard [20]	Double routing, runtime arbiter selection, default winning, bypass path	RC, VA, XB, SA faults are tolerated

4. Savior: Proposed Permanent Fault Tolerant Router

The router plays a crucial role in NoC design. The baseline router architecture consists of VC buffers, multiplexers, demultiplexer, RC unit, VA, SA and XB. The major portion of the router is composed of input ports which results in larger power dissipation as compared to packet transmission [24,25]. For the better performance of NoC architecture, the optimal utilization of these components is necessary. If a permanent fault occurred in this portion of the router, then it will result in degraded performance of the system or permanent deadlock. From these arguments, the need for such an architecture which can tolerate the permanent faults in these portions of the router is highly desired. Different types of fault scenarios occurred in different stages. For handling different faults scenarios, each stage is handled separately and different strategies are utilized for handling permanent faults. The proposed architecture handles the permanent faults in the RC stage by utilizing the RC sharing approaches. For the VA stage, we proposed an input pairing architecture along with arbiter borrowing within the paired group. For the SA stage, multiple backup paths are provided. The crossbar faults are handled by adding extra circuitry that creates multiple paths to reach the output port. The further details of faults scenarios occurring in these stages along with the working of the proposed methodology are discussed in sub-sections.

4.1. Input Port Fault Tolerance

The proposed design in this paper makes use of a pairing approach with sharing resources within the paired group along with providing runtime configuration among the paired group. All modules in a paired group such as buffers, multiplexer, demultiplexer, RC, and VA faults can be tolerated and can be shared in case of a fault. These resources are shared in two adjacent ports and the local port remains alone due to its

different nature. The proposed router architecture by using ports sharing approach for the 5 × 5 router is shown in Figure 2. In the proposed router architecture, the four directional ports are paired together in the form of two paired groups and both these paired can be configured together to increase the fault tolerance. The proposed design provides run-time configuration between the paired group to overcome the drawback in the dynamic resource sharing approach. The proposed router increases the VC utilization because of sharing it with other ports. The maximum VC utilization can be achieved with the help of sharing buffers among all input ports. However, this sharing approach will result in increased power consumption and complexity. The pairing is done between south and west, north and east, while the local port is left alone due to its different nature. The dynamic sharing among the input ports is achieved with the help of a modified dynamic resource sharing (MDRS) module. The MDRS module proposed approach worked in such a way that the occurrence of a fault in one unit does not affect the other. Fault in both paired ports is tolerated by providing dynamic configuration at runtime.

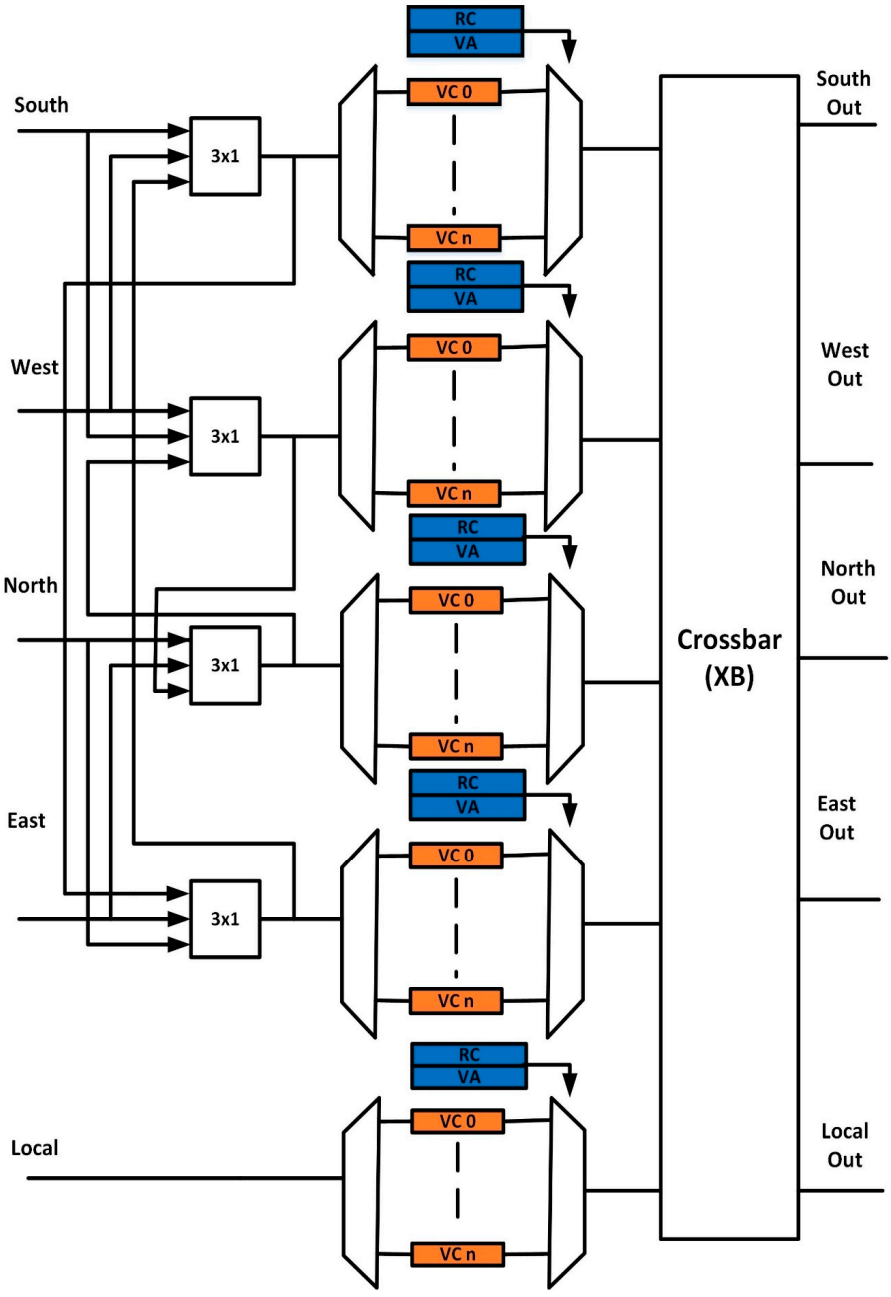

Figure 2. Port grouping in 5 × 5 router architecture.

This architecture ensures that no fault can cause all ports to fail. The paired group tolerates the fault of any unit in the input port. If both paired ports become faulty, then the dynamic pairing adopted in this work can tolerate this fault by configuring these ports with other healthy ports.

The proposed router architecture can tolerate different types of faults. It can retain the performance of the system up to a certain level after the occurrence of faults. In NoC architecture, a fault can manifest in various components such as physical link, buffers, controller, and pipeline stages of the router. The major portion of the router is occupied by buffers; therefore, occurrence of the fault probability in the buffer is high. The proposed architecture protects the physical link, buffers, and pipeline stages.

Case 1: Faulty links solution:

The occurrence of faults on links can be tackled without modifying the routing computation unit. For handing the permanent fault on links, we mostly used a permanent fault-tolerant routing algorithm that finds alternative paths in the network or uses a deflection routing algorithm [26–28]. These methods have fault tolerance capability, but they result in increased average latency. In the proposed model, each input is connected to two MDRS modules to use the alternative route if the main path is faulty. The ports are connected to a decoupled structure which can recover the router from port failure. In this way, faults on links are tolerated without the modification of the routing algorithm.

Case 2: Deadlock due to fault tolerance:

The purpose of having multiple VC buffers in the input port is to avoid the deadlock situation in the network. When the number of faults in the VC buffer exceeds a limit, the input link uses the alternative path to access the paired port VC. A deadlock situation is avoided with the use of other input port fault-free VC buffers.

Case 3: Load management:

If multiple VC buffers in a port becomes faulty, then this port becomes overloaded because there are fewer resources available, which result in a larger delay. For load balancing, the resources are shared between paired ports, and dynamic resource sharing among the paired ports results in better resource utilization. Thus, fault impact in one link is distributed equally among the paired port, thus resulting in better load management.

Case 4: Routing logic fault tolerance:

If a permanent fault manifested in the routing computation unit of a port, all resources of that input port cannot be used. If a permanent fault manifested in the RC of one input port, then it will use the paired input port RC unit to complete its execution.

4.2. Savior: RC Stage

In the baseline router architecture, each port has its sperate RC unit to perform the routing computation. The occurrence of an RC fault in the port fails that specific port. The flits present in that input port are not able to reach the destination, thus resulting in a permanent deadlock. As shown in Figure 1, each port is working independently. If the RC unit of the east input port is affected with permanent faults, then flits residing in that port will never be able to traverse resulting in a deadlock situation. To avoid this deadlock situation and to increase the fault tolerance in the first stage, we have utilized the resource sharing approach among RC units. In the absence of faults, each input port performs routing computation using its RC. In the case of a faulty RC unit, the input port can share the RC of any other port. The local port is not paired with any other port so we have used spatial redundancy for local port.

This sharing approach is controlled with the help of a fault control unit. Sharing results in a delay of one cycle. The unavailability of RC also incurs more latency. The increase in latency depends upon

the number of faults occurring in the routing computation stage. In the worst case only, one RC is working which is shared among all the flits. The latency incurs due to the unavailability of the RC unit depends upon the traffic load and number of faults occurring at this stage.

4.3. Savior: VA Stage

The VA unit has two sub-stages for performing the virtual channel allocation. In the first sub-stage, each input virtual channel is associated with a specific number of arbiters. The virtual channel allocation process is started as soon as there is a head flit in the VC. The VA makes use of RC output for the head flit. The v:1 arbiter associated with that output port is used to select an empty VC at the downstream router. The occurrence of a permanent fault in one of the arbiters will fail to arbitrate an empty VC, which in turn results in the blocking of that flit. Each input port has the same number of arbiters. After observing this behavior, we have utilized arbiter borrowing along with pairing architecture for the input ports to increase the tolerance of the circuit towards faults.

Each input port has the same number of arbiters, and the main functionality of these arbiters is to serve each input VC to find a free virtual channel at the downstream router. The sharing of all the VA arbiters in the router can create a more significant overhead. To avoid larger overhead grouping, the directional ports are done for facilitating the arbiter borrowing within the paired group. Directional ports can be paired in different groups. We have paired input port east with north, south with west, and the local port remains independent. Working on the proposed technique and possible fault scenarios are described as follow:

In the absence of fault, VA performs in a standard way, resulting in no impact on latency. For example, if less than 4 permanent faults manifest in the east port it borrows the arbiter from within the port to handle this fault. The occurrence of the fourth fault within the east port fails all arbiters. Now we cannot take advantage of borrowing arbiter because all arbiters of the east port are faulty.

As input ports are paired up, this situation is handled with the help of arbiter borrowing from the adjacent port. In this way, north and east ports work together by utilizing arbiters within the paired group. In this way, each group can tolerate a maximum of 7 faults within a paired group. Thus, the total faults tolerated by the VA stage is 17 (14 faults in the paired group, and 3 faults in the local port). The increase in latency depends upon the unsuccessful sharing of arbiters, number of faults, and a load of traffic. The arbiter sharing within a port increases the latency of 1 cycle. Overhead of 1 cycle will be caused due to arbiter borrowing within a port with a maximum of 2 cycles, when the arbiter is borrowed from the paired group. The modified input port architecture for VA is shown in Figure 3.

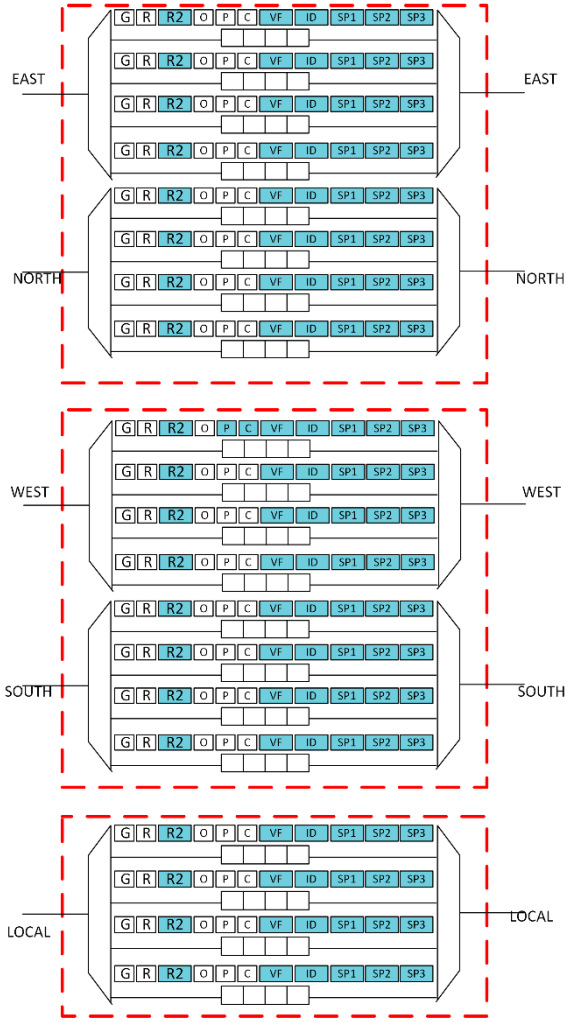

Figure 3. Modified port architecture for the proposed VA stage.

4.4. Savior: SA Stage

In the first stage of switch allocator, the v:1 arbiter is associated with each input port. The first stage is responsible for selecting a virtual channel from each of the input port which competes in the second stage. If the selected virtual channel wins, the arbitration in the second stage of it traverses a flit through the crossbar in the next cycle. When a permanent fault occurs in one of the arbiters associated with the specific port, a virtual channel from that port is not selected to participate in the second's stage of the. Thus, flits associated with the faulty port gets blocked because they never get a turn to participate in the virtual allocation process. To tolerate this fault, we have chosen a bypass path and pairing architecture for SA to tackle this situation as shown in Figure 4. The pairing approach consists of a pair of two ports along with a local port that remains alone. A default register is used to store the ID of the default virtual channel. The default path is activated when a fault is detected by the fault control unit. Within a single pair, multiple paths are created by adding a mux of 3×1, which provides the multiple paths for a virtual channel to participate in the arbitrations. Two pairs of ports are also connected with the local port with the help of 4×1 multiplexer, where one input is from the virtual channel, one from the default register, and one input from each pair of the group. The proposed SA designed is shown in Figure 4. When an arbiter of the input port gets faculty, the default register is used for arbitrations. Within a pair, if a fault occurs in an arbiter, register input line to multiplexer, 3×1 multiplexer output line, input line from the shared port register, and the arbiter of the shared port, the fault control unit bypasses the data to the local port using 4×1 mux present in the local port.

In this way, a single pair can tolerate 5 faults in a pair. A local port can tolerate one fault. The total fault tolerated by the SA stage is 11($(5 \times 2) + 1$).

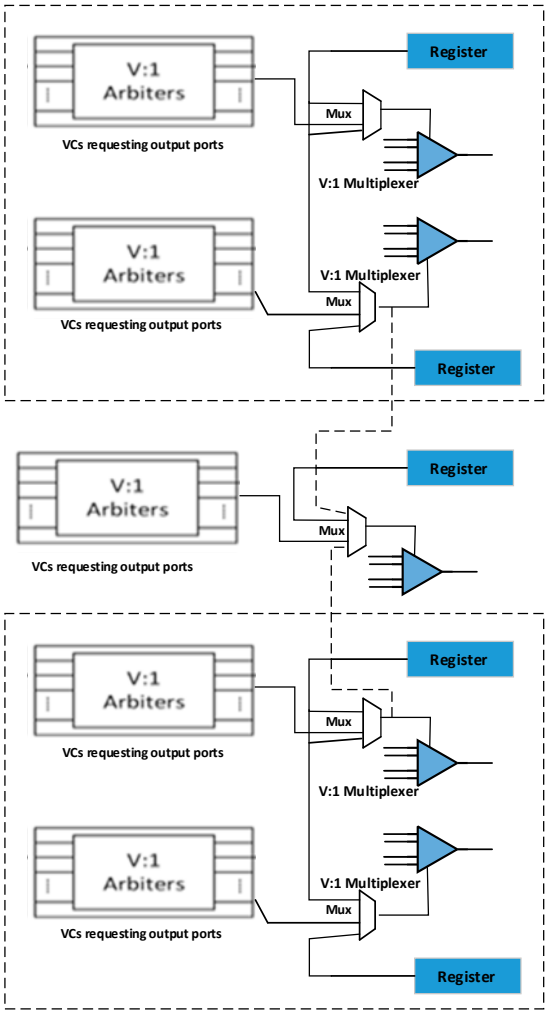

Figure 4. Modified first stage of the proposed SA.

By using a bypass path each time, a default virtual channel is selected as a winner from each input port. If the default virtual channel has a flit to transmit, it traverses a flit in the next cycle. If it is empty and other virtual channels have flits to transmit, then flits from another virtual channel are transmitted into the default virtual channel. This flit transfer between virtual channels causes a latency of 1 cycle. This flit transfer mechanism and with the help of a bypass path, we can tolerate arbiter fault.

The second stage of switch allocation is used to give access to a specific port. The occurrence of the arbiter's fault in the second stage makes the output port unreachable to the winning flit. To resolve this issue, we modified the crossbar architecture having a duplicate path to reach the output port. Flits can access the output port using the secondary path. To use this path, we have modified the input port architecture to specify the secondary path available.

4.5. Savior: XB Stage

In the baseline design, each input is associated with a multiplexer. A fault in one of the multiplexers fails to reach that specific port. There is only one path to reach the output port. The occurrence of the fault results in blocking the flits to reach that specific port. For tolerating such a fault, we modified the architecture of the crossbar design. Our proposed crossbar design has created two paths for each of the outputs, as shown in Figure 5. The demultiplexers D1, D2, and D3 create two paths to reach an output

port with the help of 3 × 1 multiplexers named M21, M22, M23, M24, and M25. Three extra flags are added to the input port to handles the operation of the crossbar design. The working of crossbar design for normal operation is also changed. In normal mode, the SP3 flag is equal to zero, which means M21, M22, M23, M24, and M25 value needs to be zero to select the usual path. The flag SP2 determines the value of D1, D2, and D3 selection. The flag SP1 determines the value selection of M1, M2, and M3.

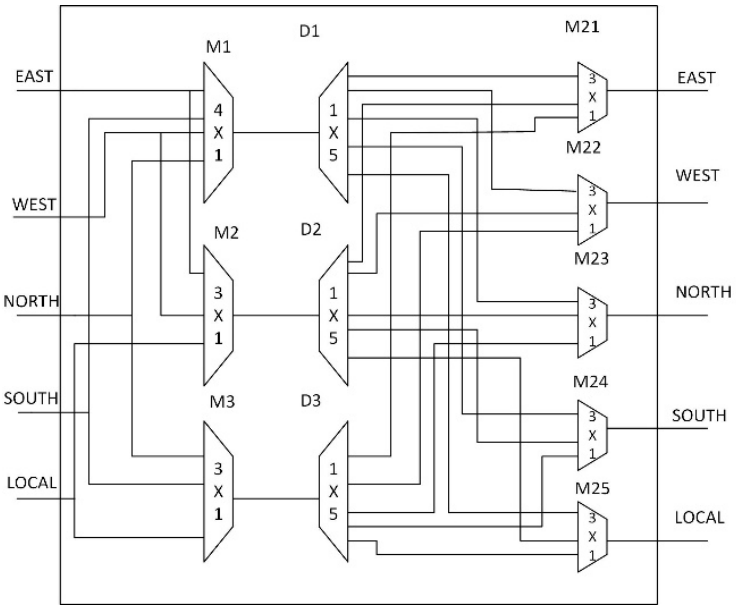

Figure 5. Modified XB architecture.

For example, a virtual channel from the east input port needs to access the west output port as the value computed by the RC unit of the east input port. After winning the arbitration in the switch allocation stage, the value of the SP1 flag indicates to activate the M1 and select east input port, SP2 indicates to activate D1 and send output to the respective output line which goes to M22. The SP3 flag selects this value by the M22 zero selection value. In this way, the east output port can access the west output port. In case of a fault in the D1 and M1, we are not able to access the west port. As the fault is detected, the switch allocator changes the values of fields SP1, SP2, and SP3 to activate the secondary path for an east input port to reach the output port. The new values of fields for the east input port to access west output are changed. The SP1 field is used to activate M2 and select east input, SP2 activates the D2 and sends output to the respective line which goes into M22, and SP3 changes the selection of M22 to 1 to select the second path of the east input port to access the west port.

4.6. Proposed Input Port Architecture

Figure 3 also shows the modified input port architecture for all the ports in VA design. The field R2, VF, and ID are added to facilitate the arbiter borrowing [29] within a port. The R2 is used for storing the result of borrowing virtual channels, VF flags show the status that arbiter is free or borrowed and the ID field shows the identification of borrowing VA. Three extra fields SP1, SP2, and SP3, are added to facilitate the traversing of the flit through our proposed crossbar design. The working of SP1, SP2, and SP3 is used to select the different paths for each input port to reach an output port. These fields provide fault protection to the crossbar design.

5. Results and Discussion

We have evaluated our proposed router architecture design in terms of area, latency, and SPF. We chose a generic 5 × 5 router architecture with five input ports and five output ports, with each input port having four virtual channels.

5.1. Synthesis Results

For calculating power and area overhead of the proposed router architecture, we developed both baseline and proposed router design in Verilog. Cadence Encounter RTL Compiler was used for synthesizing the design of 45 nm technology. A comparison of area overhead with state-of-the-art fault-tolerant router architecture is shown in Figure 6. It is shown by comparing the baseline and proposed Savior design that the area and power overhead of the proposed design is 27% and 26%, respectively. For fault detection, we utilized checkers designed by the NoCAlert [30]. Incorporation of fault detection mechanism resulted in area and power overhead of 29% and 28%. The results show that the proposed architecture achieves higher reliability by incurring smaller overhead as compared to an existing architecture.

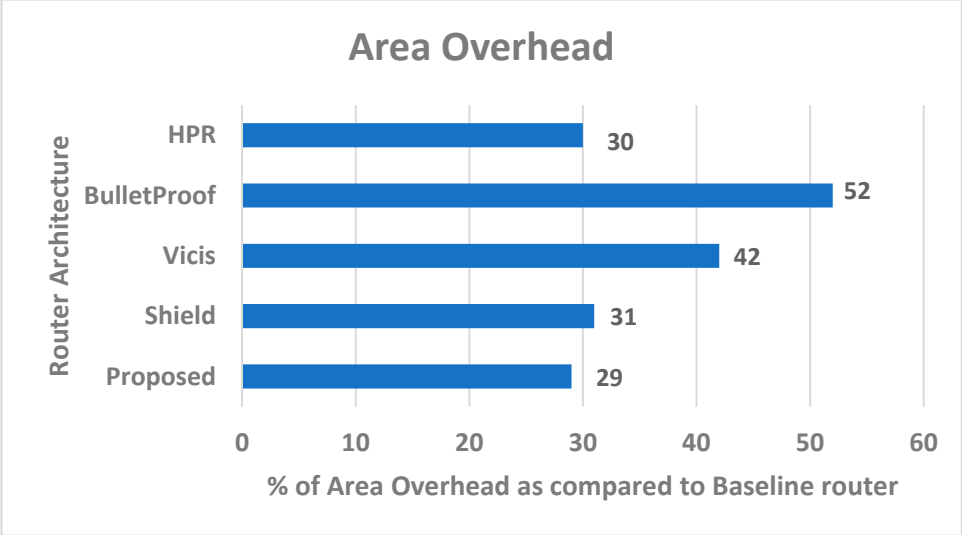

Figure 6. Comparison of area overhead.

5.2. Fault Tolerance

The area plays an important role to determine the fault tolerance capacity of router architecture design. Numerous factors can be used to determine the fault tolerance capacity of router architecture design, but one of the most critical factors that determine the fault tolerance capacity of the router in terms of the area is the SPF. The silicon protection factor can be defined as a ratio of mean numbers of faults that cause failure to a router area overhead in comparison to a baseline router. Thus, the higher the value of the silicon protection factor means the higher reliability of design towards permanents errors. The overall SPF is calculated by considering each stage separately.

(1) Input port: In the worst-case scenario for the proposed router input port, the minimum number of faults to cause router failure is one due to an unprotected local port. For the best-case scenario, it can tolerate a maximum of 27 faults. The proposed router provides a runtime sharing approach between the paired groups. The maximum number of faults to cause failure can be obtained by considering one whole paired group to be faulty except one MDRS module in working condition. In this way, one paired group tolerates eight VC buffers' faults, two multiplexers, two demultiplexers, and one MDRS module fault. Thus, one pair tolerates a total of 13 faults. Another pair can tolerate seven VC buffers' faults, one MDRS, one multiplexer, and one demultiplexer fault. Thus, it can tolerate a total of 10 faults in other pairs of the input port. The local port tolerates three VC buffers' faults. The total fault tolerated by the proposed router is 26 (13 + 10 + 3). The router will fail if another fault occurs in the paired group. Thus, a maximum of 27 faults are tolerated by the router.

(2) RC Stage: The fault tolerance for the RC stage is already provided by the pairing architecture designed for the input port. The extra protection is added to improve fault tolerance. In the absence of fault, each port performs its routing computation using its RC unit. In case of a fault in one input port, it can share the RC of the other paired input ports. The control unit configures the circuitry to share the RC of other ports with the faulty input port. Therefore, it can tolerate a maximum of four RC faults. In the worst case, if all RC of the input ports is faulty, then the router stops working. The minimum number of faults to cause failure for this stage is 2 as for the local port.

(3) VA Stage: Fault tolerance is achieved at this stage by arbiter borrowing and pairing up two ports together. Each input port pairs can also share its arbiters. The input port east is paired up with the north, west with south, and the local port remains independent. Each input port has four VCs. Each pair has a total of eight arbiters. In case of a fault in one VC, it borrows an arbiter from other VCs within the same pair. The VA stage can tolerate seven faults in each pair. The local port is not paired up with any of the input port; it can tolerate three faults. The maximum number of faults in this stage can tolerate is 17 ((7×2) +3). Due to a local port, which is working alone, the minimum number of faults to cause failure for this stage is 4.

(4) SA Stage: The switch allocation process consists of two sub-stages. The first stage makes use of arbiters associated with the input port to select a virtual channel for participating in the second stage. If the arbiter of the input port gets affected by the fault, then it cannot participate in the switch allocation process. Fault tolerance is provided by pairing two ports and providing a bypass path. The local port is paired with the other two pairs of the router input ports. After pairing up and providing a bypass path, one pair can tolerate five faults. The total faults tolerated by two pairs is 10. One port is working independently along with the bypass path. It can tolerate one fault. SA can tolerate a maximum of 11((5×2) +1) faults. The minimum number of faults to cause failure for this stage is two due to the local port, which is working independently.

(5) XB stage: A crossbar connects the input ports with output ports by providing multiple inputs and multiple output connections. In the baseline router, a single fault in the multiplexer associated with any of the input ports can cause the failure of the router. Fault tolerance is provided by creating multiple paths to reach the output port.

Each output port can be reached through two paths. The occurrence of a fault on one path can be tolerated by configuring D1, D2 demultiplexers, and Mux21. The maximum fault tolerance capability of the crossbar design is 2.

5.3. SPF of the Proposed Router Architecture

According to the SPF definition, it is the mean number of faults to cause a router failure divided by the area overhead due to extra circuitry added to provide fault tolerance shown in Equation (1).

$$\text{SPF} = \frac{\text{Mean number of faults}}{\text{Area overhead}} \tag{1}$$

The term Mean number of faults to cause the failure can be defined in Equation (2).

$$\text{Mean number of faults } = \frac{(\text{Min faults to failure } + \text{ max faults to failure})}{2} \tag{2}$$

The minimum fault for input port to cause failure is 2, the RC stage to cause failure is 5, VA stage results into failure at four faults, SA causes failure if two faults occur and for the crossbar stage, the minimum fault to failure is 2. Considering minimum faults from all the pipeline stages is equal to 2. The maximum number of faults tolerated by the router can be taken by considering the sum of faults tolerated by the individual stage, which results in 26(Input) + 4(RC) + 17(VA) + 12(SA) + 2(XB) = 61 faults. Router microarchitecture can tolerate a maximum of 61. One more fault, the entire router becomes faulty and stops working. The maximum faults to cause failure are 61 + 1 = 62. Thus, the mean

number of faults by using Equation (2) to cause failure is (62 + 2)/2 = 32 faults. The proposed architecture resulted in an area overhead of up to 29%. According to the definition of SPF, the proposed design SPF can be calculated as 32/1.29 = 24.8.

The proposed router architecture is compared with state-of-the-art fault-tolerant router architecture in terms of area overhead, SPF, and mean numbers of faults to cause failure. The proposed design has the lowest area overhead as compared to the state-of-the-art design, as shown in Figure 6. The router achieves the highest mean number of faults, resulting in a higher SPF value, as shown in Table 2. According to the definition of SPF, a higher SPF value means better reliability towards permanents errors. The router achieves the highest means number of faults and SPF as compared to existing architectures. The results are shown in Table 2.

Table 2. Comparison of the existing router architecture for fault protection capability.

Papers	Mean Faults	SPF
HPR [19]	28.5	21.9
BulletProof [15]	3.15	2.07
Vicis [13]	9.3	6.55
Shield [11]	15	11.4
Proposed Savior	32	24.8

5.4. Latency Analysis

To evaluate the performance of the proposed router architecture, we modify the baseline router architecture present in Gem5 [31] and the Garnet [32] tool integrated with it. All simulation is evaluated on simulating 64 nodes in a 2D mesh network. The topology designed consists of an 8 × 8 network, while all nodes are connected in the mesh topology. Each input port contains four virtual channels and a capacity of residing 16 flits per VC. The mesh network is evaluated on synthetic traffic patterns. The garnet synthetic traffic injector works with Garnet standalone coherence protocols. The uniform random and tornado traffic patterns are injected in the networks on a varying injection rate of 0.01 to 0.1. The proposed router architecture can tolerate the faulty links and also pipeline stages. The baseline architecture is simulated using a fault-tolerant routing strategy [33] to tolerate the links' faults. For a fair comparison, both routers are injected with the same number of faults. In the first configuration, the router architecture is simulated on a uniform and tornado synthetic traffic for patterns at varying injection rates (0.01, 0.03, 0.05, 0.07, 0.09, 0.1 packets/node/cycle), and 10 million cycles are simulted. Each simulation was performed multiple times and the mean value considered. The fault is inserted during the runtime simulation and added randomly in the network. The faults are added on the links, buffers write signal, RC, VA, SA, and XB stages. Initially, one fault is injected, and the total number of faults is injected until the router reaches a stage where it can tolerate the maximum number of faults. It is observed that, as the number of faults increases, the latency of the router also increases, but it continues working. For the detection of the fault, the NoCAlert [30] checkers are utilized to activate the correction circuitry to provides fault tolerance. The Graph depicts the proposed router achieves slightly higher latency than the baseline router. The average increase in latency is observed to be minimal as compared to the baseline router. The proposed architecture is evaluated on both synthetic and benchmark traffic. The router resulted in an overhead of 8% and 6% for uniform and tornado traffic patterns, as shown in Figures 7 and 8. The second configuration of the router consists of an 8 × 8 mesh network simulated for SPLASH2 and PARSEC benchmark traffic. The proposed router results in an increase of 12% and 10% as compared to baseline router architecture, as shown in Figures 9 and 10.

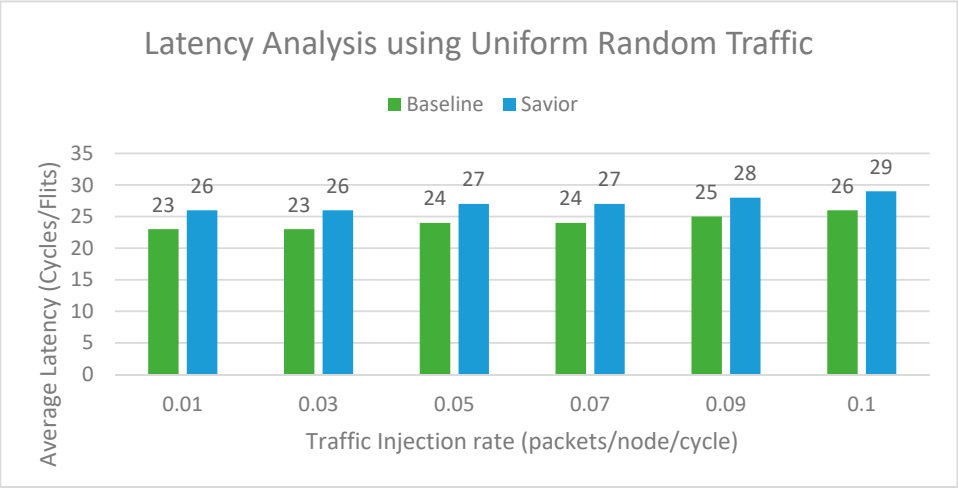

Figure 7. 8 × 8 NoC with uniform synthetic traffic.

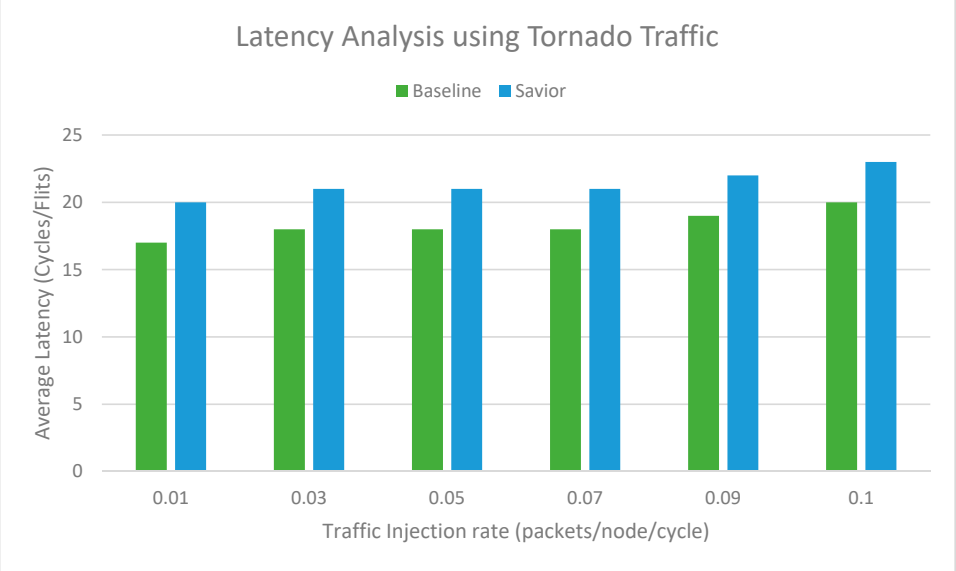

Figure 8. 8 × 8 NoC with tornado synthetic traffic.

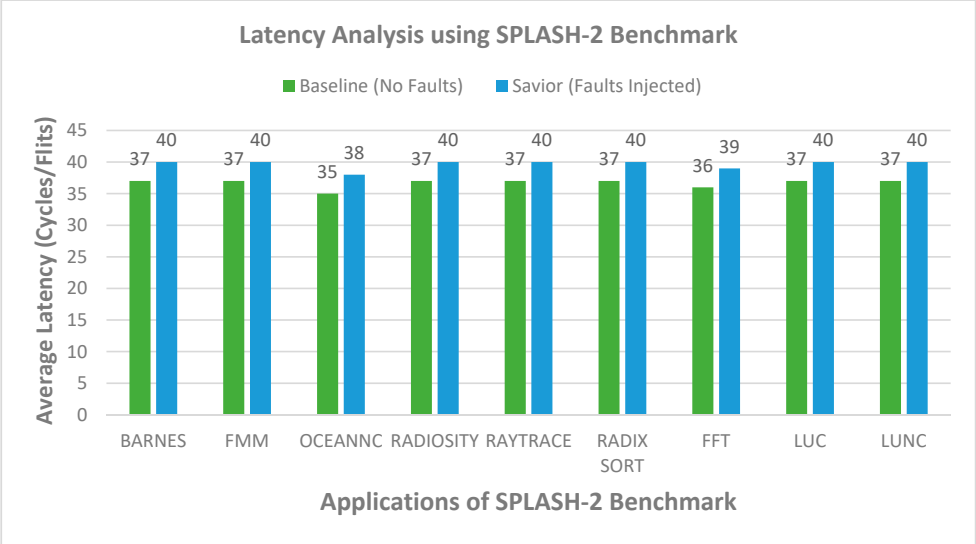

Figure 9. 8 × 8 NoC with Savior routers using SPLASH-2.

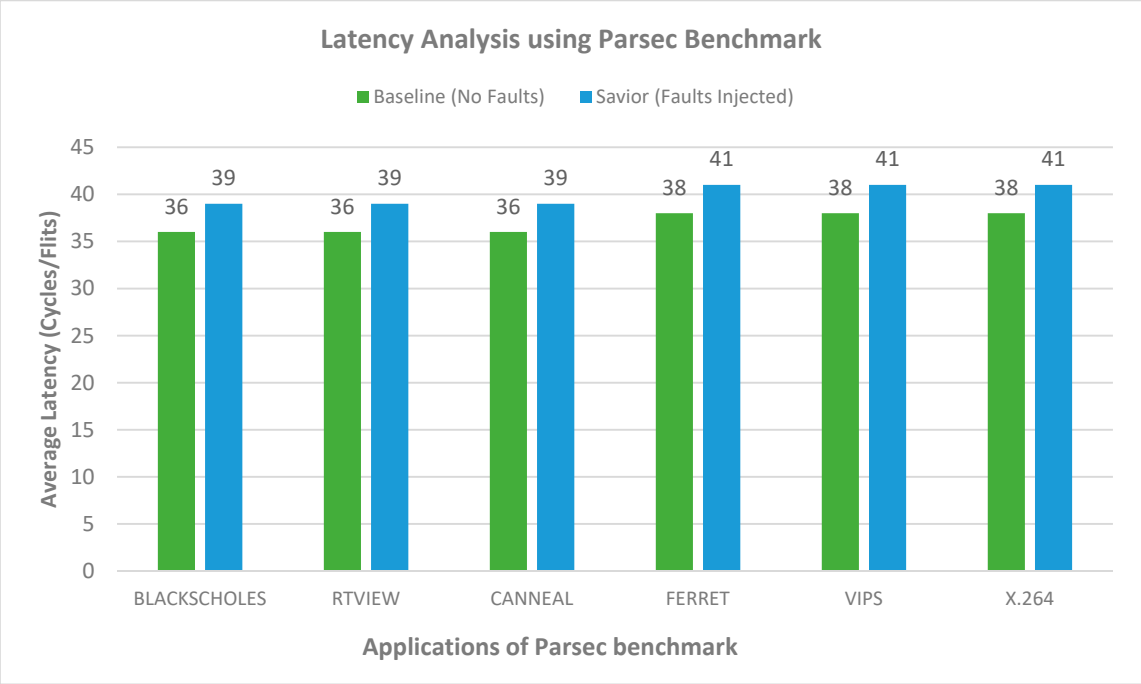

Figure 10. 8 × 8 NoC with Savior routers using PARSEC.

6. Conclusions

Router plays a critical role in an NoC architecture. Providing fault tolerance for permanent faults is necessary for smooth communication between multiple cores. The proposed architecture achieves fault tolerance in buffers and router stages. A pairing architecture is done for an input port, RC sharing in routing computation, dynamic sharing for VA, multiple and bypass path for SA, and creating multiple paths for the crossbar to connect input with output. 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 and power overhead of 29 and 28 percent. From the perspective of reliability using SPF, we showed that the proposed architecture achieves the highest SPF 24.8 among all other existing fault-tolerant architecture. The evaluation results show that our technique achieves the lowest area and highest mean number of faults to failure as compared to other state-of-the-art methods available. Overall, the results show that the proposed router achieves the right balance between reliability improvements achieved and the overhead incurred.

Author Contributions: A.H. and N.K.B. proposed the research conceptualization and methodology. The technical and theoretical framework was prepared by T.A. and M.I. The technical review and improvement were performed by A.G. and U.D. The overall technical support, guidance, and project administration were done by N.K.B., L.D. and J.A.-D. The editing and, finally, proofreading was done by A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish 'Ministerio de Ciencia Innovación y Universidades' and EDER program in the framework of the 'Proyectos de I+D d Generación de Conocimiento del Programa Estatal de Generación de Conocimiento y Fortalecimiento Científico y Tecnológico del Sistema de I+D+i, Subprograma Estatal de Generación de Conocimiento' (ref: PGC2018-095747-B-I00).

Acknowledgments: The authors acknowledge the Ministry of Education and the Deanship of Scientific Research, Najran University, Kingdom of Saudi Arabia, under code number NU/ESCI/19/001.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Borkar, S. Design challenges of technology scaling. *IEEE Micro* **1999**, *19*, 23–29. [CrossRef]
2. Benini, L.; De Micheli, G. Networks on chip: A new paradigm for systems on chip design. In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 4–8 March 2002; pp. 418–419.
3. Latif, K.; Rahmani, A.-M.; Nigussie, E.; Seceleanu, T.; Radetzki, M.; Tenhunen, H. Partial virtual channel sharing: A generic methodology to enhance resource management and fault tolerance in networks-on-chip. *J. Electron. Test.* **2013**, *29*, 431–452. [CrossRef]
4. Borkar, S. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro* **2005**, *25*, 10–16. [CrossRef]
5. Agarwal, A.; Iskander, C.; Shankar, R. Survey of network on chip (noc) architectures & contributions. *J. Eng. Comput. Archit.* **2009**, *3*, 21–27.
6. Oussalah, S.; Nebel, F. On the oxide thickness dependence of the time-dependent-dielectric-breakdown. In Proceedings of the IEEE Hong Kong Electron Devices Meeting (Cat. No. 99TH8458), Shatin, Hong Kong, China, 26 June 1999; pp. 42–45.
7. Dutta, A.; Touba, N.A. Reliable network-on-chip using a low cost unequal error protection code. In Proceedings of the 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007), Rome, Italy, 26–28 September 2007; pp. 3–11.
8. Ali, T.; Noureen, J.; Draz, U.; Shaf, A.; Yasin, S.; Ayaz, M. Participants Ranking Algorithm for Crowdsensing in Mobile Communication. *ICST Trans. Scalable Inf. Syst.* **2018**, *5*. [CrossRef]
9. Ebrahimi, M.; Daneshtalab, M.; Liljeberg, P.; Plosila, J.; Tenhunen, H. Agent-based on-chip network using efficient selection method. In Proceedings of the IEEE/IFIP 19th International Conference on VLSI and System-on-Chip, Kowloon, Hong Kong, China, 3–5 October 2001; pp. 284–289.
10. Ali, T.; Draz, U.; Yasin, S.; Noureen, J.; Shaf, A.; Zardari, M. An Efficient Participant's Selection Algorithm for Crowdsensing. *Int. J. Adv. Comput. Sci. Appl.* **2018**, *9*, 399–404. [CrossRef]
11. Poluri, P.; Louri, A. Shield: A reliable network-on-chip router architecture for chip multiprocessors. *IEEE Trans. Parallel Distrib. Syst.* **2016**, *27*, 3058–3070. [CrossRef]
12. Valinataj, M.; Shahiri, M. A low-cost, fault-tolerant and high-performance router architecture for on-chip networks. *Microprocess. Microsyst.* **2016**, *45*, 151–163. [CrossRef]
13. Fick, D.; De Orio, A.; Hu, J.; Bertacco, V.; Blaauw, D.; Sylvester, D. Vicis: A reliable network for unreliable silicon. In Proceedings of the 46th Annual Design Automation Conference, San Francisco, CA, USA, 26–31 July 2009; pp. 812–817.
14. Kim, J.; Nicopoulos, C.; Park, D.; Narayanan, V.; Yousif, M.S.; Das, C.R. A gracefully degrading and energy-efficient modular router architecture for on-chip networks. *ACM SIGARCH Comput. Arch. News* **2006**, *34*, 4–15. [CrossRef]
15. Constantinides, K.; Plaza, S.; Blome, J.; Zhang, B.; Bertacco, V.; Mahlke, S.; Austin, T.; Orshansky, M. BulletProof: A defect-tolerant CMP switch architecture. In Proceedings of the 12th International Symposium on High-Performance Computer Architecture, Austin, TX, USA, 11–15 February 2006; pp. 5–16.
16. Kastensmidt, F.L.; Sterpone, L.; Carro, L.; Reorda, M.S. On the optimal design of triple modular redundancy logic for SRAM-based FPGAs. In Proceedings of the Conference on Design, Automation and Test in Europe, Munich, Germany, 7–11 March 2005; pp. 1290–1295.
17. Polian, I.; Hayes, J.P. Selective Hardening: Toward Cost-Effective Error Tolerance. *IEEE Des. Test Comput.* **2010**, *28*, 54–63. [CrossRef]
18. Mohammed, H.J.; Flayyih, W.N.; Rokhani, F.Z.B. Tolerating Permanent Faults in the Input Port of the Network on Chip Router. *J. Low Power Electron. Appl.* **2019**, *9*, 11. [CrossRef]
19. Wang, L.; Ma, S.; Li, C.; Chen, W.; Wang, Z. A high performance reliable NoC router. *Integration* **2017**, *58*, 583–592. [CrossRef]
20. Shafique, M.A.; Baloch, N.K.; Baig, M.I.; Hussain, F.; Zikria, Y.B.; Kim, S.W. NoCGuard: A Reliable Network-on-Chip Router Architecture. *Electronics* **2020**, *9*, 342. [CrossRef]
21. Poluri, P.; Louri, A. A soft error tolerant network-on-chip router pipeline for multi-core systems. *IEEE Comput. Arch. Lett.* **2015**, *14*, 107–110. [CrossRef]

22. Yu, Q.; Zhang, M.; Ampadu, P. Exploiting inherent information redundancy to manage transient errors in NoC routing arbitration. In Proceedings of the 2011 Fifth IEEE/ACM International Symposium on Networks on Chip (NoCS), Pittsburgh, PA, USA, 1–4 May 2011; pp. 105–112.
23. Fick, D.; De Orio, A.; Chen, G.; Bertacco, V.; Sylvester, D.; Blaauw, D. A highly resilient routing algorithm for fault-tolerant NoCs. In Proceedings of the Conference on Design, Automation and Test in Europe Conference and Exhibition, Nice, France, 20–24 April 2009; pp. 21–26.
24. Banerjee, N.; Vellanki, P.; Chatha, K.S. A power and performance model for network-on-chip architectures. In Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, Paris, France, 16–20 February 2004; pp. 1250–1255.
25. Ye, T.T.; Micheli, G.D.; Benini, L. Analysis of power consumption on switch fabrics in network routers. In Proceedings of the 39th annual Design Automation Conference, New Orleans, LA, USA, 10–14 June 2002; pp. 524–529.
26. Feng, C.; Lu, Z.; Jantsch, A.; Zhang, M.; Xing, Z. Addressing transient and permanent faults in NoC with efficient fault-tolerant deflection router. *IEEE Trans. Very Large Scale Integr. Syst.* **2012**, *21*, 1053–1066. [CrossRef]
27. Liu, J.; Harkin, J.; Li, Y.; Maguire, L.P. Fault-tolerant networks-on-chip routing with coarse and fine-grained look-ahead. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2015**, *35*, 260–273. [CrossRef]
28. Runge, A. FaFNoC: A Fault-tolerant and Bufferless Network-on-chip. *Procedia Comput. Sci.* **2015**, *56*, 397–402. [CrossRef]
29. Poluri, P.; Louri, A. Tackling permanent faults in the network-on-chip router pipeline. In Proceedings of the 25th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD), Pernambuco, Brazil, 23–26 October 2013; pp. 49–56.
30. Prodromou, A.; Panteli, A.; Nicopoulos, C.; Sazeides, Y. Nocalert: An on-line and real-time fault detection mechanism for network-on-chip architectures. In Proceedings of the 2012 45th Annual IEEE/ACM International Symposium on Microarchitecture, Vancouver, BC, Canada, 1–5 December 2012; pp. 60–71.
31. Binkert, N.; Beckmann, B.; Black, G.; Reinhardt, S.K.; Saidi, A.; Basu, A.; Hestness, J.; Hower, D.R.; Krishna, T.; Sardashti, S. The gem5 simulator. *ACM SIGARCH Comput. Archit. News* **2011**, *39*, 1–7. [CrossRef]
32. Agarwal, N.; Krishna, T.; Peh, L.-S.; Jha, N.K. GARNET: A detailed on-chip network model inside a full-system simulator. In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software, Boston, MA, USA, 26–28 April 2009; pp. 33–42.
33. Valinataj, M.; Liljeberg, P.; Plosila, J. A fault-tolerant and hierarchical routing algorithm for NoC architectures. In Proceedings of the NORCHIP Conference, Lund, Sweden, 14–15 November 2011; pp. 1–6.