Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.Doi Number

Defender: A Low Overhead and Efficient Fault-Tolerant Mechanism for Reliable on-Chip Router

Naveed Khan Baloch¹, Muhammad Iram Baig², Masoud Daneshtalab³
¹Computer Engineering Department, University of Engineering and Technology, Taxila, 47040, Pakistan
²Electrical Engineering Department, University of Engineering and Technology, Taxila, 47040, Pakistan
³School of Innovation, Design and Engineering, Malardalen University, Vasteras, Sweden

Corresponding author: Naveed Khan Baloch (e-mail: naveed.khan@uettaxila.edu.pk).

This work was carried out at the Digital Systems Laboratory in Computer Engineering Department, University of Engineering and Technology Taxila, Pakistan and was fully supported by the Swedish Knowledge Foundation (KKS) through the DeepMaker and DPAC projects.

ABSTRACT The ever-shrinking size of a transistor has made Network on Chip (NoC) susceptible to faults. A single error in the NoC can disrupt the entire communication. In this paper, we introduce Defender, a fault-tolerant router architecture, that is capable of tolerating permanent faults in all the parts of the router. We intend to employ structural modifications in baseline router design to achieve fault tolerance. In Defender we provide the fault tolerance to the input ports and routing computation unit by grouping the neighboring ports together. Default winner strategy is used to provide fault resilience to the virtual channel arbiters and switch allocators. Multiple routes are provided to the crossbar to tolerate the faults. Defender provides improved fault tolerance to all stages of routers as compared to the currently prevailing fault tolerant router architectures. Reliability analysis using silicon protection factor (SPF) and Mean Time to Failure (MTTF) metrics confirms that our proposed design Defender is 10.78 times more reliable than baseline unprotected router and then the current state of the art architectures.

INDEX TERMS Network-on-Chip; Router Architecture; Permanent Fault Tolerance; Silicon Protection Factor; Mean Time to Failure

I. INTRODUCTION Advancements in semiconductor technology [1], [2] allow us to fabricate silicon dies with billions of transistors. This facilitates the development of chip multiprocessors (CMPs). To provide communication among multiple cores on the chip, a distinctive architecture, called Network on Chip (NoC) [3], [4], is used. NoC has two types of components (also called blocks): (i) computational blocks and (ii) communication blocks. Computation blocks are processing cores, and communication blocks consist of routers and links, that transfers data between cores in the form of packets from source to destination routers. As the technology scales down to nanometers and operating frequencies grow higher, NoC components incur various faults [5] [6]. The two classes of faults are permanent and transient faults. Faults that arise either at the production time or due to physical damage during operation of the circuit and continually affect the function of the circuit are called permanent faults. Reasons of permanent faults are time-dependent dielectric breakdown (TDDB) [7], negatively biased temperature instability [8], hot carrier injection [9], and electro-migration [10]. The fault that lasts only for a few clock cycles and temporally affects the operation of the circuit is called transient fault. These faults are usually arising because of alpha particles striking the logic from packaging material [11], thermal radiations from cosmic rays [12], and manufacturing process variations [13]. A fault produces errors in the operation of the chip, for example causing increased latency, packet loss, packet error, and can create a deadlock. A single fault may paralyze the whole chip if proper precautions are not considered in the design. Hence it is an utmost desire at design stage to create such a circuit that can avoid the faults at later stages. Our focus in this paper is to provide fault-tolerance in each component of the router.
The router in NoC architecture consists of many components like buffers, Routing Computation unit (RC), Virtual channel Allocators (VA), Switch Allocators (SA) and Crossbar (XB). Traversing of the packet through the pipeline of the router is shown in Figure 1.

**FIGURE 1.** Router pipeline stages

When a packet entered in a router, it arrives in the assigned buffer and then traverses through these pipeline stages, starting from RC to the downstream components. The detailed working of the input port and pipeline components of the baseline router is explained in section 2. The proper operation of these stages is essential for a router to work correctly and to forward the packet to the correct destination. Many researchers have already worked to tolerate the faults occurring in NoC router and links [14-18]. However, these approaches do not provide fault tolerance to each component of router. We have proposed the techniques to tolerate the faults occurring on all the components of the router by grouping the input ports for buffers and RC protection, temporal parallelism for VA, rectification circuitry at SA, and the bypass path for the crossbar.

The remaining sections of the paper are presented as follows.

In section 2, we provide an overview of baseline router architecture with all the internal components. In section 3, we describe the other permanent fault tolerant router architectures related to our work. In section 4 the problem statement is described in detail. In section 5 the proposed fault-resilient router: Defender architecture is described with fault tolerance capability for all internal components. In section 6, experimental setup is given and a detailed discussion on results for area overhead, reliability and performance analysis is provided. Section, 7 concludes the paper.

II. OVERVIEW OF BASELINE NETWORK ON CHIP ROUTER

There are many router architectures available in the literature such as [19][20]. We have selected a generic 2-stage router [21] as shown in Figure 2. The baseline router consists of the multiplexers, de-multiplexers, buffers, virtual channels, routing computation (RC) unit, virtual channel allocator (VA), switch allocator (SA) and crossbar (XB).

A. INPUT UNIT

This is the first part of the router which receives the incoming packet. It consists of MUX, DeMUX, and fixed length buffers (as shown in Figure 2). These buffers are named a virtual channel. A single physical channel is shared among all the virtual channels. Each arriving flit kept in a specific VC buffer selected by its VC identifier.

B. ROUTING COMPUTATION UNIT

A packet consists of multiple flits, i.e., head flit, one or more body flits and a tail flit. In routing computation stage, the output port for downstream router is calculated for a head flit of packet as soon as it arrives in the VC. The working procedure of the routing computation stage depends on the routing algorithm. There are many routing algorithms available for the NoC, some are simple dimension order and more complex, i.e., adaptive routing algorithms.
In the first stage a single request is selected from each requesting virtual channel and the second stage removes the conflicts between different input virtual channels that have assigned the identical virtual channel at the downstream router.

**D. SWITCH ALLOCATION UNIT**

After the virtual channel is allocated to a flit, the next stage is switch allocation. In this stage, a virtual channel competes with other input virtual channels to gain access to the crossbar for a specific output port (as shown in Figure 4). Unlike virtual channel allocation, which is performed only for head flit, this stage is performed for each flit. In a generic 2-stage NoC router architecture the two switch allocators are implemented. One is used to handle the speculative requests and other for non-speculative requests. The priority of non-speculative requests is higher.

**FIGURE 4.** Baseline switch allocation

**E. CROSSBAR UNIT**

The crossbar is used to connect the input ports to the output ports. The winning virtual channel in the switch allocation stage can now use the crossbar to transmit its flit to the downstream router (as shown in Figure 5). The crossbar connections configure each cycle, which is determined by the winning flit in the switch allocation stage.

**FIGURE 5.** Baseline crossbar unit

Several approaches are proposed as discussed in the next section to improve the performance and reliability of NoC router.

**III. RELATED WORK**

The preceding methodologies for fault tolerant NoC router design are reviewed in this section. Permanent fault in any component of the router may outcome in complete failure of that router and affects the communication in the network. Researchers have proposed various approaches to tackle these faults. Pavan et al. [22] [23] proposed the methodology to tolerate permanent faults in the pipeline stages of the router. The proposed fault resilience technique is achieved by using spatial redundancy, resource sharing, and adding some correction circuitry. Each port in the router contains an additional RC unit so that if one RC fails the other can be used to calculate the output port for a packet. VA stage of the router is protected by arbiter sharing within the port. The faults in the SA stage are tolerated by using a default path. Crossbar is protected by providing the additional paths to reach the output port.

Kim, et al. [24] proposed a router architecture named as RoCo, which disintegrate the components of a router and structure them into rows and columns resulted into smaller crossbars and parallel arbiters. This architecture can have performance degradation in case of faults. When a hard fault occurs in the crossbar connecting north and south output port, the flits trying to reach that output port will be blocked results in performance loss in the network. Constantinides, et al. [25] suggested a defect tolerant chip multiple processor (CMP) router architecture named as Bulletproof. It is based on spatial redundancy techniques which required multiple copies of the same component to tolerate the hard faults resulting in more considerable area overhead. Bulletproof uses a generic model of a bathtub curve to represent permanent faults. Moreover, bulletproof suggests automatic cluster decomposition model for achieving modularity in router architecture design. It takes a netlist and creates equal size partitions. It divides the fault tolerance process into sub-processes of detection, diagnosis, repair, and recovery.

Fick et al. [26] proposed a router architecture named Vicis, which can tolerate permanent faults on links and routers. Vicis employed inherent redundancy to maintain correct operations of the router. Port swapping and bypassing a path is used to tolerate permanent faults in router microarchitecture. Moreover, Vicis used distributed routing algorithm to avoid faults in the network. The faulty link information is collected with the help of the BIST controller and is used to reroute the traffic around faulty links and routers. Vicis used input port swapping algorithm and network rerouting techniques to improve the reliability of the
router. To support the input port swapping, the links of Vicis router are functionally bidirectional. Each link consists of two input ports and two output ports. If any of the port is failed, the remaining three ports are available to make new connections. The Vicis router also has a bypass bus to protect the router against crossbar failure.

Mohit et al. in [27] presented a routing algorithm that is designed for the mesh of tree based connected network. This routing algorithm can avoid faulty routers in case of failure. They modified the simple deterministic algorithm for NoC to provide fault tolerance and proved that the proposed algorithm is live-lock free and ensures the delivery of packets to the destination nodes. This can only be used for the transient faults and cannot tolerate the faults occurring inside the router components. In [28] another fault resilient routing algorithm is specified which can provide better performance and scalability to the mesh network in case of faults. They have considered the router and link failure in the network and show that their algorithm can provide better performance and reduces the hop count compared to the other recent algorithms. These fault tolerant techniques disconnect the healthy cores in the network which is undesired. Khalil et al. in [29] proposed a self-healing router architecture which can tolerate the transient and permanent faults. The fault detection unit available in the router can inform the neighboring routers about the faulty state. In case of faults the router reconfigures itself to be used just as a path for packets. Routing control unit in faulty state is used to calculate the output ports and all the packets coming from different input ports are routed in a round robin fashion. The proposed technique can avoid the internal faults but the router can stop working with single fault in routing control unit or the MUX used for selection in reconfigured state.

Junchi et al. in [30] describe the need for continuous testing in NoC. Built in self-test (BIST) is necessary for detection of faults. They Proposed a framework EsyTest which can detect the faults in NoC very efficiently and utilizes the idle cycles for testing to reduce the performance loss. It can test the data path and control unit separately and tolerate the detected faults by reconfiguration of router or by fault resilient routing algorithm. Hala et al. in [31] presented another fault resilient router architecture and focus on the faults occurring in the input ports. They use the Built-in self-test technique for the detection of faults and by hardware redundancy approach to avoid the faults occurring particularly in the state fields of the virtual channels.

The recovery process can be further classified in term of disabling, ignoring, and replacing faulty components. For tolerating the permanent faults, a simplified approach is the Triple Modular Redundancy (TMR) [32]. In this approach, each component of the router architecture is duplicated or triplicated depending upon the N-modular redundancy approach, where N is the number of additional components available. For example, in a triple modular redundancy approach, each unit has two extra copies. All three units perform the computation, and the final output is determined by a voting system that compares all three results, and the majority output is selected as a final output. In this way, it can tolerate a single fault at the cost of more substantial area overhead, which is undesired. Wang et al. [33] have proposed a high-performance reliable (HPR) fault resilient router architecture. They have utilized virtual channel closing for an input port, lookahead routing for routing computation stage, default winning strategy for arbiters and bypass bus to tolerate the crossbar faults.

### IV. PROBLEM STATEMENT

There is a need to provide such an architecture which can avoid a greater number of faults in the input ports and pipeline stages. Input ports in the routers consist of VC buffers, DeMUXes and MUXes and gets the most significant share of area and power inside the router. The optimal utilization of these components is essential for reliable packet communication. Shield [23] and HPR [33] are state of the art and higher fault tolerating architectures available. Shield [23] can only tolerate the faults in the pipeline components of the router and assumed that input port buffers are well protected by ECC techniques. They ignored the failure of other components present in the input port, i.e., MUX and DeMUX. HPR [33] uses the virtual channel closing technique for fault tolerance in the input ports. They utilize the double bit detection and single bit correction ECC code. When the flit entered in the input port, the ECC code is stored in the particular buffer and compared with the generated code. In the case of a single bit error, the fault recovery unit can correct it otherwise that virtual channel is closed for future use. They are tolerating a total 16 faults in all the input ports of the router using this technique, but as shown in figure 6 with a single MUX and DeMUX failure in HPR [33] input port, all the resources become inoperative, and all the detection and correction circuitry is useless.

So, there is a need to provide more reliability in this unit. We have grouped the adjacent ports to solve this problem. Our architecture provides improved reliability in the input port units and provides enhanced reliability to the RC and VA stages as compared to the HPR [33] with low area overhead. We also propose better fault tolerant architecture to the SA unit by utilizing the default winning strategy. Our proposed router architecture Defender is superior from these techniques in that it provides tolerance to a higher number of faults to the input port and for each pipeline stage.

It utilizes the inherent redundancy in the router microarchitecture to tolerate the permanent faults by
utilizing temporal parallelism, rectification circuitry, and multiple routes and offers noticeable improvements over baseline and state-of-the-art router designs.

![Fault detection and protection circuitry](image)

**FIGURE 6.** MUX and DeMUX failure in input port

V. DEFENDER THE PROPOSED FAULT TOLERANT ROUTER ARCHITECTURE

We named our proposed fault-tolerant router architecture as Defender. It provides better reliability compared to the state-of-the-art reliable router architectures. In this section first, we present the effects of faults at each stage, and then we present the techniques used for the detection and tolerance of these faults.

A. DEFENDER: INPUT PORT FAULT TOLERANCE

Input port consists of MUX, DeMUX, and virtual channels and consumes the most significant area of the router. Therefore, it is evident that protection to all the parts of the input port is necessary otherwise the router may fail its operation, i.e., in case of faulty DeMUX at the start of the input port, flits cannot enter into the router, and all the resources become inoperable. Similarly, permanent faults in the MUX and VC buffers also affects the operation of the input port and block the flits. Network on Chip is a highly redundant architecture. There are five ports in a typical NoC router. To protect the input ports from faults, we grouped the adjacent ports. As shown in Figure 7, we grouped the north with the east port, and another group contains the remaining three south, west, and local ports. The internal resources of these groups are shared among the members in fault situations. Each module that is used for grouping of ports consists of MUXes and an Error Control Unit (ECU) which utilize the NoCAAlert checkers [34] for the detection of faults in the input port. In the proposed shared input port architecture, each input channel has two paths to reach the downstream router, therefore any fault resilient deflection routing is not required. The dimension order XY routing algorithm is enough. So, there is no chance for a deadlock or live lock to occur in this architecture. Shield [23] and HPR [33] provides the separate fault tolerant mechanism for RC and VA. The Defender also provides the separate fault tolerant techniques to these units as well as due to the sharing of ports the reliability of these stages is improved. In Figure 7, we have also shown the fault-tolerant architectures for the SA and XB for reference, but it is discussed in the separate sections.

![Proposed fault tolerant router architecture (Defender)](image)

**FIGURE 7.** Proposed fault tolerant router architecture (Defender)

B. DEFENDER: RC STAGE FAULT TOLERANCE

In the RC stage, the next hop is calculated for the packet, which takes it one step closer to the destination router. A faulty RC may calculate the wrong output port which takes the packet away from its destination. In case of adaptive routing, this misrouted packet may reach its destination with increased latency, and in case of deterministic routing, a deadlock may occur in the network. A faulty RC may also calculate the invalid output port. For example, if the router has five ports (numbered 0 to 4), value 5 and onwards are invalid [35].
Figure 8 shows the checkers based of NoCAlert [34] for detection of RC faults in the router. Checker in (Figure 8a) detects the calculation of wrong output port, which can transmit the packet in the wrong direction away from the destination. Checker in (Figure 8b) can detect the invalid output port direction. As shown in the figures, each output port direction is assigned a 3-bit code from 0 to 4. Rest of the numbers in 3-bit representation from 5 onwards are invalid.

Lookahead routing is utilized in 2-stage NoC router, in which a current router ‘Q’ can calculate the address of downstream router ‘Q+1’and calculates the address of next downstream ‘Q+2’ for a specific head flit. So, the faulty RC computation in ‘Q+1’ does not affect this stage because the next RC in ‘Q+2’ router uses the lookahead routing in one additional clock cycle to calculate the output port direction. We exploit this redundant RC computation strategy available in baseline router to avoid faults at this stage, without any additional hardware. Our proposed design is more fault resilient to this stage; if two RC modules in a row become faulty, our architecture allows packets to go to the other member port of the group.

C. DEFENDER: VA STAGE FAULT TOLERANCE

Virtual Channel Allocation (VA) stage is responsible to allocate a free VC buffer to the incoming packet. Its calculation is based on the destination information available in the head flit. Faults on this stage can allocate a used VC buffer to a new packet, allocation of VC buffer without request and allocation to nobody when there is at least one request. When an inaccurate VC allocation occurs to the occupied VC buffer in the downstream router, then the new flit overwrites the existing flit, results in data corruption. If VA allocates the downstream VC buffer without request, the allocated buffer resource is wasted and cannot be allocated to the needy flit at later times.

A fault when the VA allocates to nobody results in the flit to stay in the current router and results in the deadlock. Fault resilience at this stage can prevent data corruption, prevention against resource wastage and to avoid the deadlock in the network.

Detection and localization of faults is necessary to provide tolerance. We have utilized the detection mechanism given by NoCAlert [34] for this purpose. Checker given in Figure 9a can detect the fault if the downstream VC buffer is allocated without a request for prevention of resource wastage. It consists of simple logic gates for fault detection and gives very less area overhead. Checkers in Figure 9b detect the faults of multiple grants to a simultaneous VC buffer to avoid data corruption and grant to nobody to avoid deadlock in the network.

Utilize the VA fault tolerance mechanism provided by the HPR [33] in which a register is used as a default path if the arbiter is faulty. The output VC’s are compared and the one with most free buffers is allocated to the register as default winner, shown in Figure 10.

A 2x1 MUX is used to select the output of the arbiter in the absence of fault and default winner result from the register in the presence of a fault. This technique can provide protection even when all the VA’s in the router are faulty, but this fault prevention mechanism is only for the first stage of VA. The second stage of the VA is not tolerated. Our proposed architecture uses this mechanism for tolerating the first stage faults and provide even more resilience against faults because of input port grouping. In the case of faulty VA, the flits can be transmitted to the other port in the group having healthy VA. In this way, our proposed design outperforms the state-of-the-art mechanisms in term of reliability.
D. DEFENDER: SA STAGE FAULT TOLERANCE

Switch allocation stage is responsible for arbitrating between different input VC of the input ports to assign a crossbar time in the next cycle. A faulty SA stops the flits from reaching downstream router by traversing the crossbar. In this situation, the flits remain in the VC buffer and unable to release it, which is the cause of performance degradation and can also create the deadlock in the network. For detection of these faults, we utilized the light weight checkers provided by [34]. The detection circuitry in the form of checkers is given in the previous section.

FIGURE 11. SA fault tolerance: Runtime arbiter selection

Dual switch allocators are available in 2-stage NoC router. One is used to handle the speculative requests, and the other one is used for non-speculative requests. This hardware redundancy can be utilized to provide fault tolerance at this stage. In case of a fault, one of the two arbiters can be used for allocation, which is originally proposed by [33]. As shown in the figure in Figure 11, arbiters are selected at runtime by a faulty control unit and 2X1 MUXes. The basic idea is to ensure that the non-speculative requests are handled properly and the output port is reachable. Speculative requests are handled by SA_1_IN_i and SA_1_OUT_j, while the non-speculative requests are handled by SA_2_IN_i and SA_2_OUT_j. If the SA_2_IN_i is faulty, then the fault control unit selects the SA_1_IN_i for non-speculative requests and put the speculative requests to the other faulty arbiter. In this way, the speculative requests are also handled in the next cycle when the healthy arbiter is allocated. This idea works fine, but what if both the arbiters become faulty? We have proposed a solution to this problem by little modification in the internal architecture of the arbiter circuit, as shown in Figure 12 below.

FIGURE 12. SA fault tolerance: Bypass path to tolerate the internal faults

E. DEFENDER: XB STAGE FAULT TOLERANCE

The crossbar in the router is used to connect ports. A faulty crossbar restricts the flits to reach the downstream router. Crossbar consists of multiple multiplexers for establishing the connection. A faulty crossbar is detected by confirming the defects in the multiplexers. The light weight checkers provided by [34] are used for the detection of permanent faults. The multiplexer is considered as faulty if the selected signal for the multiplexer provides no output on the port.

Fault tolerant crossbar design is shown in figure 13, as initially proposed by [33]. There are two bypass busses in this design one for the X dimension and other for Y dimension. The first bus is used to connect all the input ports of X dimension and the local port to all the output ports. The second bypass bus connects the Y dimension input ports to the output ports and the local port. There is no conflict between these two bypass busses.
VI. RESULTS AND ANALYSIS

We evaluated the defender our proposed router architecture in terms of area overhead, reliability analysis as Silicon Protection Factor (SPF), and Mean Time to Failure (MTTF). Performance analysis by calculating the latencies using synthetic and real traffic patterns. These results are compared with the baseline and another state-of-the-art architecture.

A. AREA OVERHEAD

We implement the baseline router and Defender: our proposed fault resilient router in Verilog. Cadence Encounter RTL compiler is used for synthesis of the designs in NangateOpenCell 45 nm library. Area overhead is an important factor to confirm the fault tolerance efficiency. The following equation 1 obtains area Overhead.

\[
\text{Area Overhead} = \frac{\text{Fault tolerant design area}}{\text{Baseline Area}} \tag{1}
\]

Figure 14 shows the area overhead of the baseline design with state-of-the-art architectures. It can be seen that defender is more area efficient in providing the fault tolerance compared to previous designs.

B. RELIABILITY IMPROVEMENT COMPARISON USING SPF

There are many types of metrics available to access the reliability of the circuit. Considering the area overhead for this purpose is very beneficial to confirm the fault tolerance efficiency. Increase in area overhead is obvious if we use the spatial redundancy to provide the fault resilience to a circuit. Therefore, it is necessary to consider a metric that utilizes area overhead with fault tolerance ability. We are using Silicon Protection Factor (SPF) given by Constantinides et. al in [25] for the comparison of reliability of proposed router with the state-of-the-art architectures.

\[
\text{SPF} = \frac{\text{Mean No of faults to cause failure}}{\text{Area Overhead}} \tag{2}
\]

Here, the mean no of faults to cause failure can be calculated by Equation (3):

\[
\text{Mean no of faults} = \frac{\text{Min faults} + \text{Max faults}}{2} \tag{3}
\]

Here Min faults and Max faults are the minimum and maximum no of faults to failure. A fault at the input port is defined as a faulty MUX, DeMUX and faulty virtual channels. A faulty MUX or DeMUX can block the entire input port, and all the resources become inoperable. A fault in the virtual channel causes an incorrect flit. In defender, we have grouped the adjacent ports to provide fault tolerance. The internal resources of these groups are shared among the members in fault situations. One paired group provides fault tolerance to 4 VC buffers, MUX, DeMUX, One Error Control Unit (ECU) module used for grouping the ports and 3 VC buffers faults in an adjacent port. So, a total 10 number of faults are tolerated by one paired group. The other group which connects three input ports can also tolerate 10 faults. The router can tolerate a maximum of 20 (10x2) faults at the input ports. The minimum number of faults to cause failure at the input port is 2.

RC is the next stage in the router in which the occurrence of fault results in the calculation of the wrong output port. Our protection strategy to this unit can avoid 7 faults. A generic 2-stage router provides inherent redundancy which can
tolerate 5 RC faults in the router. Two more faults can be tolerated by grouping the ports. A failure occurs at minimum two faults if the RC at the downstream router is not working.

After the routing computation, the next stages are VA and SA. In our protection strategy, the SA can work properly in the presence of a maximum 15 faults. Due to the inherent redundancy in 2-stage NoC router our proposed technique can provide guard to non-speculative requests if all the arbiters are failed. To tolerate the internal faults, we have utilized the bypass path inside the SA, as explained in section 5, which can tolerate the 5 more faults. A minimum of 3 faults can cause failure in SA. A maximum of 20 faults are tolerated in the VA unit in our proposed strategy. It works fine even if all the arbiters are faulty. On the other side only a presence of 2 faults can cause failure if the default winner and input VC are faulty simultaneously. The crossbar is used to connect the input ports with the output. A fault in this unit is defined as the faulty MUX. In the proposed technique with the inclusion of two bypass buses in this unit can result in tolerating a total 5 faults in it. Only 2 faults in the crossbar result in failure if both the MUXes and bypass buses are faulty. The minimum number of faults to cause failure can be calculated as min (2(Input port), 2(RC), 2(VA), 3(SA), 2(XB)), which is 2 faults. The maximum number of faults tolerated by the router can be calculated to sum all the faults tolerated by various components which are 20(Input port) + 7(RC) + 20(VA) + 15(SA) + 5(XB) = 64. One more fault in the router cause failure in operation. Mean a number of faults to cause failure is 33. Area overhead of the proposed design is 28%. SPF of our proposed router defender is 33 / 1.28 = 25.8

The SPF value of our proposed router is compared with the other designs, as shown in Table I. Greater the SPF value shows that the architecture is highly reliable, it can be seen that Defender is much more reliable than the state-of-the-art architectures.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Faults to cause failure</th>
<th>SPF</th>
</tr>
</thead>
<tbody>
<tr>
<td>BulletProof</td>
<td>3.15</td>
<td>2.07</td>
</tr>
<tr>
<td>Vicis</td>
<td>9.3</td>
<td>6.55</td>
</tr>
<tr>
<td>RoCo</td>
<td>5.5</td>
<td>&lt;5.5</td>
</tr>
<tr>
<td>Shield</td>
<td>15</td>
<td>11.4</td>
</tr>
<tr>
<td>HPR</td>
<td>28.5</td>
<td>21.9</td>
</tr>
<tr>
<td>Defender</td>
<td>33</td>
<td>25.8</td>
</tr>
</tbody>
</table>

C. LIFETIME IMPROVEMENT ESTIMATION USING MTTF: Lifetime improvement comparison of defender with the baseline can be calculated by the Mean Time to Failure (MTTF) [36],[37]. Where, MTTF of a circuit can be calculated by the following equation 4.

\[
\text{FIT} = 10^9 \frac{\text{Failures} \times \text{in} \times \text{Time} (FIT)}{}
\]

Here FIT is the failure of operations per billion hours in a given component. To find out the FIT, we use the Failure in Time Estimation Model proposed in paluri et al. [37] Lifetime modeling framework proposed by Shin et al. [38] is utilized in this work. The value of FIT for a single field effect transistor (FET) which is produced by time dependent dielectric breakdown (TDDB) can be calculated by Equation 5 [39]

\[
\text{FIT}_{\text{FET}} = \frac{\text{dutycycle} 	imes 10^9}{A_{\text{TDDB}}} \times Vdd^{a-kT} \times e^{-\frac{X+Y+Z}{kT}}
\]

Here \(A_{\text{TDDB}}\), a, b, X, Y, Z are fitting parameters, k is the Boltzmann’s constant, T is operating temperature which is 300 Kelvin and Vdd is operating voltage which is 1V. In equation 5, the dutycycle is set to 100%. The FIT value of basic logic gate then easily can be calculated by multiplying the transistor count with the \(\text{FIT}_{\text{FET}}\). The Sum of Failure (SOFR) [40] model is used to calculate the FIT value of the component and then the entire router. FIT estimation of baseline 2-stage NoC router is calculated by HPR [33] and is given in Table II.

<table>
<thead>
<tr>
<th>Unit</th>
<th>FC</th>
<th>FIT of the FC</th>
<th># of FC's</th>
<th>FIT of the Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Buffer</td>
<td>128-bit DFF</td>
<td>0.5</td>
<td>40960</td>
<td>20480</td>
</tr>
<tr>
<td>RC</td>
<td>6-bit comparator</td>
<td>11.7</td>
<td>10</td>
<td>117</td>
</tr>
<tr>
<td>VA</td>
<td>20:1 Arbiter</td>
<td>36.7</td>
<td>20</td>
<td>1468</td>
</tr>
<tr>
<td>SA</td>
<td>4:1 Arbiter</td>
<td>7.4</td>
<td>10</td>
<td>215</td>
</tr>
<tr>
<td></td>
<td>5:1 Arbiter</td>
<td>9.3</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4:1 mux</td>
<td>4.8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>XB</td>
<td>128-bit 5:1 mux</td>
<td>819.2</td>
<td>5</td>
<td>4096</td>
</tr>
</tbody>
</table>

D. FIT ESTIMATION OF RELIABLE ROUTER: DEFENDER

• Input Port: To protect the input port from faults, we have grouped the adjacent ports. If one port is not working, the flits of that port can be transferred to the neighboring port. This technique doesn't need any extra circuitry.

• RC unit: No extra circuitry is needed for the protection of the RC unit in our double routing strategy. Moreover, if RC of a port is not working, it can utilize the neighboring port RC for the calculation of the output port.

• VA unit: the default winner strategy provides fault tolerance at this stage. 20 2bit registers are used to save the
result of default winner and 20 2:1 multiplexer is needed to select the output result from the registers are arbiters.

• **SA unit:** 20 2:1 multiplexer is needed in SA to select the nonfaulty arbiters to ensure the non-speculative switch allocation requests. To provide the inside protection to the SA, we have utilized the 5 2-bit DFF registers for default winner and 5 3:1 multiplexer is utilized to select between the registers and arbiters.

• **XB unit:** Crossbar unit in a reliable router is protected from faults by additional circuitry of four 3:1 multiplexer and three 2:1 multiplexer.

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>FIT ESTIMATION OF RELIABLE ROUTER DEFENDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unit</td>
<td>Component</td>
</tr>
<tr>
<td>VA</td>
<td>20 2-bit DFF</td>
</tr>
<tr>
<td></td>
<td>20 20-bit 2:1 Muxes</td>
</tr>
<tr>
<td>SA</td>
<td>20 2:1 Muxes</td>
</tr>
<tr>
<td></td>
<td>5 2-bit DFF (registers), 3:1 muxes</td>
</tr>
<tr>
<td>XB</td>
<td>3 128 bit 2:1 muxes</td>
</tr>
<tr>
<td></td>
<td>4 128 bit 3:1 muxes</td>
</tr>
</tbody>
</table>

Table III shows the FIT estimation of the reliable router for individual components. It also shows the extra component used to avoid the faults.

**E. MTTF OF PROPOSED ROUTER: DEFENDER**

Using the SOFR model the MTTF value of the baseline 2-stage router is calculated as given in equation 6

$$MTTF_{Baseline} = \frac{10^9}{20480 + 117 + 1468 + 215 + 4096} = \frac{10^9}{37913 \text{ hours}}$$

(6)

The FIT of the reliable router using the SOFR model is calculated as 660 + 57 + 2252.8 = 2969.8. Equation 7 can is utilized to calculate the MTTF value of the reliable router defender and given below.

$$MTTF_{Reliable \text{router}} = \frac{10^9}{FIT_1} + \frac{10^9}{FIT_2} + \frac{10^9}{FIT_1 + FIT_2}$$

(7)

Here $FIT_1$ is the FIT value of baseline router (26376), and $FIT_2$ is the FIT value of the reliable router (2969.8). Hence the MTTF value of Defender is 408713 hours, which is 10.78 times to that of baseline router. Defender is 10.78 times more reliable then baseline. Figure 15 shows the comparison of MTTF of other states of the art reliable NoC routers with the Defender, which shows that Defender is more reliable than other fault tolerant routers.

**F. PERFORMANCE ANALYSIS**

For performance analysis, we have calculated the latencies for synthetic and real traffic in the simulator. We use GEM5 [41] integrated with Garnet2.0 [42] for simulating 8x8 mesh on different synthetic traffic patterns. We make use of uniform random synthetic traffic pattern at injection rates ranges from 0.01 to 0.1. We adopted XY routing protocol for transmitting the packets within the router. The Link latency is assumed to be 1. For each injection rate, simulation is repeated 10 times and taking average value to calculate the average flit latency. The simulation cycles of each injection rate are placed 500000 to calculate the flit latency. We have injected faults randomly in randomly chosen 8x8 router. It is observed that as the number of faults increases the latency of the proposed router architecture also increase. The latency of the router increases up to 6% for Uniform Random traffic pattern as compared to the baseline router architecture, shown in Figure 16.

**FIGURE 15. Mean Time to Failure (MTTF) Comparison**

**FIGURE 16. Latency Analysis using Uniform Random Traffic pattern**

**FIGURE 17. Latency Analysis using Tornado Traffic pattern**
VII. CONCLUSION

In this work, we presented Defender, a fault tolerant router architecture capable of tolerating permanent faults in all the parts of the router. We considered each component of the router separately and proposed a fault-tolerant mechanism for that part. The proposed architecture gives minimum latency overhead as shown in the simulation results. The Mean Time to Failure metric use for analysis show that Defender is 10.78 more reliable than the baseline router. The Silicon Protection Factor confirms that the defender can tolerate a greater number of faults with less area overhead compared to the state-of-the-art reliable router architectures. The techniques used in this work for fault tolerance can also be used to the other routers. Overall evaluation of the defender shows that the it can tolerate a greater number of faults with less area overhead compared to the other state of the art reliable router architectures.

REFERENCES


[36] P. Ramachandran et al., “Metrics for architecture-level lifetime reliability analysis”, in Proc. ISPASS, Austin, TX, USA, April, 2008, pp. 202-212


NAVEED KHAN BALOCH received his BSc degree in Computer Engineering from University of Engineering and Technology, Taxila, Pakistan in 2007. He has worked in multinational companies as embedded system designer from 2007 to 2010. He joined the university as lecturer after completing his MS degree from UET Taxila currently he is pursuing his PhD from the same university. He has published many research papers in his field and have experience in Embedded system designing, Fault tolerant systems, reconfigurable computing and currently he is working on Self-healing digital systems. Currently he is working as Assistant Professor in Computer Engineering Department UET Taxila. During his tenure in the academia he did many collaborations with industry and foreign universities in the field of on-chip networks, embedded vision and reconfigurable computing.

MUHAMMAD IRAM BAIG Dr Iram baig is working as Professor in Electrical Engineering Department, University of Engineering and Technology Taxila, Pakistan. He is the director of Centre of ASIC Design and DSP in the department, and creates the facilities for experimentation and research. He has published many research papers in Fault tolerant systems, Reconfigurable Computing and Embedded systems. He did many collaborations in the industry and academia and delivered many successful projects on ASIC design and FPGA. His current research interests include FPGA based accelerators for deep learning algorithms, high performance computing, and fault tolerant designs.

MASoud Daneshtalab is currently a tenured associate professor at Mälardalen University (MDH) and a researcher visitor at Royal Institute of Technology (KTH) in Sweden. He joined KTH as European Marie Curie Fellow in 2014. Before that, he was a university lecturer and group leader at University of Turku in Finland from 2012-2014. He has been appointed as Associate Editors of Elsevier Journals of Computers & Electrical Engineering (CAEE) and Microprocessors and Microsystems (MICPRO) along with World Research Journal of Computer Architecture (JCA); and in the Editorial Board of The Scientific World Journal, International Journal of Distributed Systems and Technologies (JDST), International Journal of Adaptive, Resilient and Autonomic Systems (IARAS), International Journal of Embedded and Real-Time Communication Systems (IJERTCS), and International Journal of Design, Analysis and Tools for Integrated Circuits and Systems (IJDATICS). He has also served as a Guest Editor for several prestige Journals, like Springer Computing, IET CDT, ACM TECS, ACM JETC, Elsevier’s JSA, MICPRO, Integration, and CEE. He is also the key organizer of SIGMARC ACM/IEEE International workshop Network-on-Chip Architectures (NoCArc / 8 editions) and ACM International workshop on Many-core Embedded Systems (MES / 4 editions). He has represented Sweden in the management committee of the EU COST Actions IC1202: Timing Analysis on Code-Level (TACLe). Since 2016 he is in Eurimicro board of Director and a member of the HiPEAC network. His research interests include on/off-chip interconnection networks, hardware/software co-design, reconfigurable computing, neuromorphic architectures, and evolutionary multi-objective optimization. He has published 1 book, 4 book chapters, and over 180 refereed international journals and conference papers within H-index 24. He is currently in a Technical Program Committee member of different IEEE and ACM conferences, including DAC, DATE, ASPDAC, ICCAD, NOCs, ESTIMedia, VLSI Design, ICA3PP, SOCC, VDAT, DSD, PDP, ICESS, Norchip, MCSoC, CADS, EUC, DTIS, NESEA, CASEMANS, NoCaRc, MES, HPIN, PACBB, MobileHealth, and JEC-ECC.