ROBUST: A new Self-healing Fault-Tolerant NoC Router

Jacques Henri Collet
LAAS-CNRS, Université de Toulouse
7 av. du colonel Roche
F31077 Toulouse CEDEX 04, France
Jacques.collet@laas.fr

Ahmed Louri, Vivek Tulsidas Bhat, Pavan Poluri
University of Arizona
1230 E. Speedway Boulevard,
Tucson, AZ-85721, USA.
{Louri, pavanp}@email.arizona.edu

ABSTRACT

This work addresses the general problem of making Network-on-Chips (NoCs) routers totally self-healing in massively defective technologies. There are three main contributions. First, we propose a new hardware approach based on Built-In Self-Test techniques and multi-functional blocks (called Universal Logic Blocks, ULBs) to autonomously diagnose permanent faults and repair faulty units. ULBs have the capability to assume the functionality of various functional units within the router through simple reconfiguration and thus enable the repair of multiple permanent faults within the NoC router. Second, we propose a new reliability metric and introduce a probabilistic model to estimate the router reliability improvement achieved by the protection circuitry. Third, we compare our architecture to two router architectures (Vicis and Bulletproof) and we show that our design provides superior reliability improvement especially in extremely defective nanoscale technologies (i.e., typically above 30% of faulty routers). The most striking result is that the self-healing of the routers enables maintaining the communications at fault levels, where it is normally impossible to preserve communications.

Categories and Subject Descriptors

B.8 [**Performance and Reliability**]: Reliability, Testing & Fault-tolerance. C.1.2 [**Multiprocessors**]: Interconnection architectures.

General Terms

Performance, Reliability.

Keywords

Network-on-Chip, Fault-tolerance, Multi-core architectures, Self-Healing.

1. INTRODUCTION

As feature size continues to decrease to reach today's nanometer scales, reliability has become a serious impediment to the design of efficient NoC architectures [i,ii]. The major sources of failures can be classified as transient and permanent faults. Transient faults mostly occur due to cosmic radiations, alpha particle strikes and electromagnetic interference. On the other hand, electromigration, gate-oxide breakdown and other manufacturing imperfections have been cited as the major sources of permanent faults. In this paper, we address the general problem of making NoCs totally self-healing in massively defective technologies.

In tackling the problem of tolerating permanent faults in NoCs, one may consider implementing fault tolerance in the communication layer, in the physical architecture, or ultimately, directly at the router or link level. In the first strategy, the communication routes are discovered by some fault tolerant routing algorithm (FTRA) [iii,iv]. However, this approach leads to serious limitations in massively defective NOCs. The main reason is that it does not repair any faulty element (link or router). It "only" takes advantage of the natural redundancy existing in the interconnection network (i.e., the 2D-mesh) to find detour routes, which avoid the defective links and/or routers. Thus, a FTRA cannot find more interconnection routes than there are in a defective NOC and it is easy to show that the number of fault-free routes between any two points drops dramatically when about 30-35% of routers or links are defective in a 2D mesh [v]. This result is nothing more than the rediscovery, in the framework of communication networks, of the well-known percolation theorem [vi]. The disappearance of routes becomes all the more serious with the reduction of dimensions and the generalization of nanotechnologies. Consequently, it becomes necessary to heal components directly in the physical layer simply to maintain the ratio of defective routers and links less than 30-35%. Our work focuses on this question.

When addressing fault tolerance directly in the physical layer, one must consider faults for interconnects (links) and routers. Since there has been extensive works on tolerating faulty links with very efficient solutions [vii,viii], we will not address link failures here, rather we focus on tolerating permanent faults in routers. Several approaches are possible [ix,x,xi,xii]. Here, we shall only consider self-healing routers, i.e., routers able to self-test and repair (STAR) the diagnosed faults [x,xii]. There are three main contributions of the proposed work:

1) We introduce a new self-healing Fault-tolerant Router (FTR) architecture called **ROBUST** (**RO**uters with **BU**ilt-in **S**elf-healing **T**echniques). Permanent fault diagnosis is performed at chip start-up. BIST (Built-In Seft Test) circuit diagnosis errors trigger the self-repair mechanism. During self-repair, the ROBUST design utilizes multifunctional blocks called *Universal Logic Blocks* (ULBs) to replace the defective functional units. ULBs have the capability to assume the functionality of various functional units within the router by a process of simple reconfiguration. By utilizing an efficient resource sparing technique involving ULBs and by exploiting the existing structural redundancy in the NoC router, the ROBUST design tolerates multiple permanent faults at the cost of minimal and reasonable hardware overhead.

2) We introduce a router dependability model and a new metric to calculate the reliability improvement gained with the addition of the STAR circuitry in a FTR. The new metric provides an indication whether the reduction of fault occurrence is larger in the baseline router (in a probabilistic approach) than the occurrence of new irreparable faults in the protection overhead.

3) We compare permanent fault tolerance for ROBUST, Vicis [xii] and Bulletproof [x] FTRs. We show that the proposed design provides superior reliability improvement especially in very defective NOCs, i.e., when the fraction of faulty routers is in the range previously described, above 30%.

NoCArc'11, December 4, 2011 , Porto Alegre, Brazil.

2. RELATED WORK

Kim *et al.* [ix] proposed the RoCo router, which exploits a series of architectural techniques at different error-prone stages along the NoC router pipeline in order to tolerate permanent faults. Due to the decoupled nature of the router design, a permanent fault can be tolerated by blocking the faulty module while keeping the remaining healthy modules in operation. The authors evaluate their router architecture using a composite metric called Performance, Energy and Fault-tolerance (PEF) to study the impact on network latency, power consumption and fault-tolerance.

Koibuchi *et al.* [xi] have proposed the DBP router that makes use of default backup paths within the NoC router. These default backup paths serve as alternative datapaths within the router to circumvent functional units that have encountered hard faults. The authors evaluate their router design by studying its impact on network latency, throughput, hardware cost, and energy consumption. Note that the DBP approach includes no BIST mechanism.

Constantinides *et al.* [x] proposed the BulletProof router. The authors have proposed multiple configurations depending on the granularity at which the protection technique is employed, the fault-diagnosis and the fault-repair strategy used for the purpose of protection. In what follows, we only consider the C_2SP_BIST configuration based on BIST diagnosis.

Fick *et al.* [xii] proposed a hybrid router architecture called Vicis that uses a port-swapping algorithm to tolerate permanent faults at the network level and a crossbar bypass along with ECC units to tolerate permanent faults at the router level. Both BulletProof and Vicis router architectures are evaluated using a reliability metric called Silicon Protection Factor (SPF).

The ROBUST design is fundamentally different from the above proposed solutions. The major difference lies in the fact that ROBUST design provides protection to all the functional units of the NoC router that fall on the critical path i.e., the buffer units, the crossbar switch, the multiplexers (MUXes) and the demultiplexers (DEMUXes). ROBUST utilizes a unique resource sparing technique involving ULBs to perform the repair of faulty functional units unlike the remaining solutions discussed in this section.

3. ROBUST IMPLEMENTATION

In this section, we first describe the baseline router considered in this work. Then we describe the hard-fault diagnosis sub-system. Since the self-repair operation is performed using the ULBs, we also discuss the internal circuit details of the ULB. Lastly, we provide details regarding the ROBUST router architecture.

3.1 Baseline Router Design

The baseline design considered in this work consists of five-port NoC router architecture as shown for instance in [ix]. The five ports correspond to the four directions and a connection to the local Processing Element (PE). The router is composed of five functional modules: the Routing Computation (RC) unit, the Virtual Channel Allocator (VA), the Switch Allocator (SA), the crossbar switch and the buffer units within the five input ports. The design employs pipelining at the RC, VA, SA and crossbar stages in order to improve performance [xiii]. Every packet that arrives at the input port proceeds through the four pipeline stages before it is delivered to the appropriate output port. The input port is composed of Port DEMUXes which help in guiding the flit to the appropriate input virtual channel (VC) depending on the VC Identifier (VCID). Port MUXes help in directing the winning flit to the crossbar input. Similarly, each input VC is composed of a VC DEMUX and a VC MUX that help in storing and retrieving a flit from the flit buffers. The baseline NoC router consists of P=5 input ports, with each input port having four VCs (v = 4) and each VC having four flit buffers (r = 4). This gives a total of 80 flit buffers in the NoC router (5 input ports * 4 VCs/port * 4 flit buffers/VC). Each packet is composed of four flits and each flit is 32-bits long.

Prior research work in this area has shown that the RC and VA functional units are only responsible for processing information within the head flit [xii]. In contrast, the VC buffers, MUXes, DEMUXes, crossbar and SA unit are responsible for processing all the flits that arrive in the router. Secondly, the VC buffers, MUXes, DEMUXes and the crossbar account for larger portion of the hardware overhead in the NoC router [xiv]. In comparison, the RC, VA and SA stages are composed of only a few gates. Considering the above two arguments, it is evident that the VC buffers, MUXes, DEMUXes and the crossbar constitute the critical path of the NoC router. Therefore, we propose to protect these components using self-healing techniques. On the other hand, the RC, VA and SA functional units can be protected using traditional fault-tolerance techniques such as N-modular redundancy (NMR).

3.2 Self-Diagnosis of Permanent Faults

In ROBUST, permanent faults within the NoC router are diagnosed with the help of BIST strategy [xv,xvi]. We have adopted a BIST strategy similar to what has already been presented by Lin *et al.* [xvi] for the following reasons: 1) this approach detects 98% of stuck-at faults in the NoC router. 2) It accounts for a delay overhead of roughly 117 cycles (testing performed at 200MHz) which is considerably small as compared to other BIST solutions, and 3) The area overhead is small, approximately 13% of the total area of the baseline NoC router.

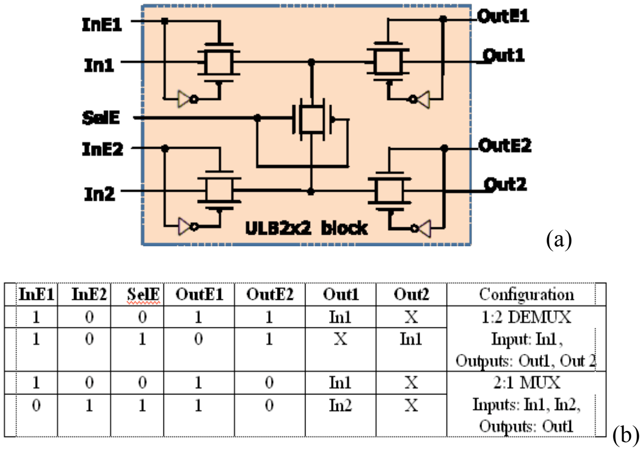
The BIST module consists of the following components: Test Pattern Generator (TPG), Test Response Analyzer (TRA), BIST controller, and Fault Isolation (FI) block. The TPG block generates multiple test patterns to test the various components within the NoC router datapath. The BIST controller controls the test procedures. The TRA block compares the results obtained from the test procedures with the desired outputs and identifies the faulty blocks within the router datapath. These results are forwarded to the FI block to isolate the faulty block and activate the corresponding correction circuitry. Since most of the system-level faults for a router translate to stuck-at faults at the router micro-architectural level, they will be caught by using pseudo-exhaustive test patterns generated by the BIST block.

3.3 Self-Repair using Universal Logic Blocks

In this section, we describe the circuit details of the two ULBs which serve as the basic building blocks in constructing ULBs of larger configuration that will be used for providing protection.

3.3.1 *ULB$_{nxn}$ block*

A ULB$_{nxn}$ block has n * 32-bit input/output lines and is built up using multiple transmission gates which act like switches. This block can be configured as a 1: n DEMUX or an n: 1 MUX. In order to understand the functionality of an ULB$_{nxn}$ block, we consider a ULB$_{2x2}$ block as shown in Figure 1a (1-bit). A ULB$_{2x2}$ block has two 32-bit input lines, *In1* and *In2*, and two 32-bit output lines, *Out1* and *Out2*. It consists of five transmission gates which can be used to direct data between the different input and output lines. In Figure 1a, *InE1*, *InE2*, *OutE1*, *OutE2* and *SelE* represent the control signals to the transmission gates. Figure 1b explains the functionality of a ULB_{2x2} block. It can be configured as a 1:2 DEMUX, with *In1* as the sole input line and *Out1* and *Out2* as output lines. Similarly, it can also be configured as a 2:1 MUX with *In1* and *In2* as input lines and *Out1* as the sole output line. We use the same idea with a ULB_{nxn} block and choose an appropriate n depending on the type of functional units that need to be protected.



InE1	InE2	SelE	OutE1	OutE2	Out1	Out2	Configuration
1	0	0	1	1	In1	X	1:2 DEMUX Input: In1, Outputs: Out1, Out 2
1	0	1	0	1	X	In1	
1	0	0	1	0	In1	X	2:1 MUX Inputs: In1, In2, Outputs: Out1
0	1	1	1	0	In2	X	

Figure 1. a) ULB_{2x2} circuit (1-bit), and (b) Truth table indicating the working of a ULB_{2x2} circuit. 'X' indicates logic don't care.

3.3.2 *ULB_{vc} block*

A ULB_{VC} block is shown in Figure 2 (1-bit). It has a single 32-bit input/output line and consists of several one-bit memory cells. Read operations from the memory cells are controlled using transmission gates that have *REn1-REn4* as enable signals. Similarly, write operations into the memory cells are controlled using transmission gates that have *WEn1-WEn4* as enable signals. The ULB_{VC} block can be used as a replacement for a faulty VC within the input port of a NoC router. Since the baseline NoC router design consists of four flits buffers within every VC, and each flit is 32-bits wide, the ULB_{VC} block should have at least 128 (32 * 4) one-bit memory cells to store flits.

3.4 ROBUST Router Design

In ROBUST, we consider splitting the NoC router into six sub-blocks for the purpose of protection. The first five sub-blocks correspond to the five input ports of the NoC router and the sixth sub-block corresponds to the crossbar switch. ROBUST provides protection to all the functional units of the router that fall on the critical path using multiple ULB_{hybrid} blocks. A ULB_{hybrid} block is composed of a ULB_{5x5} block and a ULB_{VC} block. The ULB_{5x5} block is similar in design to the ULB_{2x2} block described earlier except that it has five input/output lines. Since the ULB_{5x5} block can be used to mimic a 1:4 DEMUX or a 4:1 MUX, it can be used to protect the Port DEMUX/MUX. The ULB_{vc} block will be used to protect the VCs within the input port. In ROBUST, every input port has a dedicated ULB_{hybrid} block for the purpose of protection. Figure 3 shows how the ULB_{hybrid} block can be integrated along with an input port of a NoC router to protect the VCs, Port MUX and the Port DEMUX.

In the event of a permanent fault occurring in the Port MUX/DEMUX, the corresponding ULB_{hybrid} block will be activated and configured to act as a MUX/DEMUX and its outputs will be forwarded to the next stage in the router pipeline.

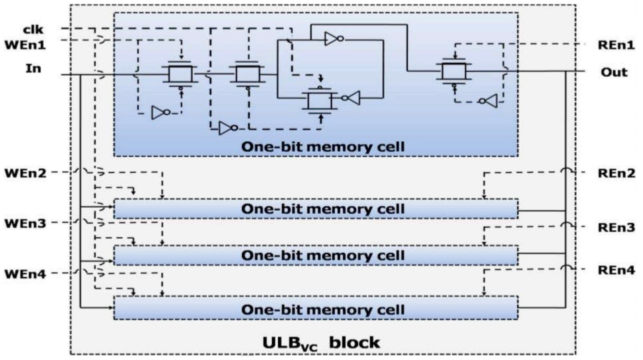


Figure 2. ULB_{VC} circuit (1-bit).

It should be also noted that in the event of a permanent fault occurring in both the Port MUX and the Port DEMUX, only one among the two can be protected using the ULB_{hybrid} block. This is because the ULB_{5x5} block can act as either a MUX or a DEMUX and not both at the same time. A similar protection strategy is used for ensuring the protection of VCs.

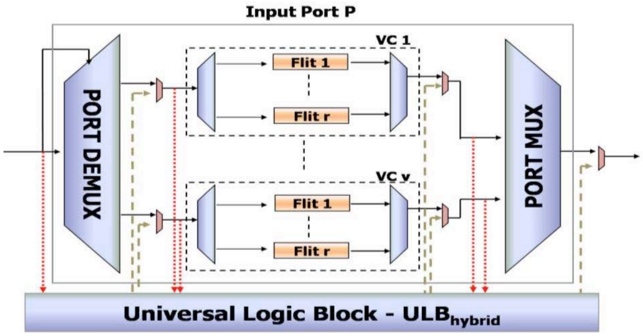


Figure 3. Protection circuitry for the Port MUX/DEMUX and the VCs of an input port P using a ULB_{hybrid}

The inputs of the VCs are also forwarded to the ULB_{hybrid} block, then within this block to the ULB_{VC} block. Among the four VCs in an input port, any one faulty VC can be replaced using a ULB_{hybrid} block. This is because the ULB_{vc} block mimics the functionality of a single VC. The repair operation requires the use of additional 2:1 MUXes to choose between the outputs of the ULB_{hybrid} block and the faulty functional unit. These MUXes are added after the Port DEMUX, Port MUX and all the input VCs in order to aid the replacement of any of these units if they encounter permanent faults.

The 5x5 crossbar switch is protected using five ULB_{hybrid} blocks, one for each output port of the switch as shown in Figure 4. The five inputs *XIn1-XIn5* are also forwarded to five ULB_{hybrid} blocks. Within the ULB_{hybrid} block, these inputs are directed towards the ULB_{5x5} block. Since a ULB_{5x5} block can also serve as a 5:1 MUX, it can be used to assume the functionality of a single crossbar output port. The advantage of adopting such a strategy is that, if the crossbar switch has encountered permanent fault(s) at certain output ports due to which it is not able to route flits to certain directions, those output ports can be displaced and replaced with a corresponding ULB_{hybrid} block while keeping the remaining output ports of the crossbar switch in operation. It should be noted that for the ULB_{hybrid} blocks used to protect the crossbar switch, the ULB_{vc} blocks are internally disabled since they are not required to protect the crossbar switch. However, in the event of multiple VC failures at the input port(s), these ULB_{vc} blocks can be activated and used as a second level of protection.

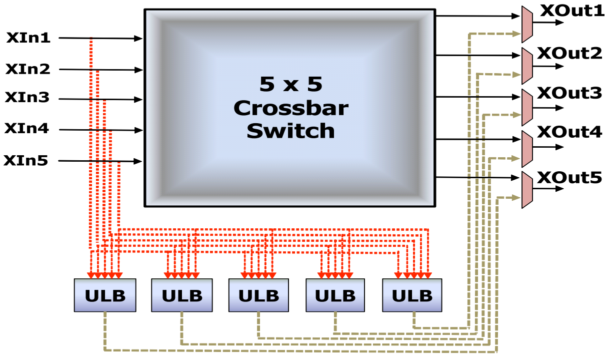


Figure 4. Crossbar protection using multiple ULB_{hybrid} blocks

4. RELIABILITY MODEL

In this section, we describe a probabilistic model to estimate the benefits of adding the STAR circuitry to tolerate permanent faults in the baseline router. Note that the STAR circuitry repairs a fraction of the faults appearing in the baseline router. Contrarily, faults occurring in the STAR circuitry itself are irreparable. The challenge consists in comparing the probability F_P that irreparable permanent fault(s) occur in the FTR (i.e., in the baseline router plus the STAR circuitry) to the probability F that faults occur in the baseline router alone. The router reliability is improved if $F>F_P$.

As described in section 3, the STAR circuitry is made up of a permanent fault detector (PFD) and some correction circuits (CC), which include one or more spares and some additional logic to aid at replacing the faulty functional unit. The protection of a module is achieved in two steps: 1) The PFD tracks permanent fault(s) in the functional units of the baseline router. 2) When it diagnoses a functional unit as faulty, it activates the CC to replace it. In order to analyze the occurrence of permanent faults, we have defined five states for thFTR as described in Table 1. *D* indicates that the baseline router contains exclusively detectable faults and *UD* indicates that it contains undetectable faults. X denotes a don't care, '1' indicates that the entity is faulty and '0' indicates that the entity is not faulty. Table 1 is a truth table, which shows the States 1-5 which account for all the possible occurrences of permanent faults within the FTR.

Table 1: FTR States

FTR States	State #	D	UD	PFD	CC
Reliable States	State 1	0	0	0	X
	State 2	1	0	0	0
Unpredictable States	State 3	1	0	0	1
	State 4	X	1	0	X
	State 5	X	X	1	X

We distinguish two set of states:

1.*Reliable states*: The FTR will operate in a reliable and deterministic manner when it is in State 1 because it is fault-free, or in State 2 because detectable faults occur exclusively in the baseline router and they are reparable using the CC. It should be noted that for State 1, a permanent fault in the CC will not disturb the regular operation of the router when the CC is not activated. Therefore, the detection of permanent faults should include MUXes and DEMUXes to isolate the CC when it is not active.

2. *Unpredictable states*: The forthcoming behavior of the FTR is unpredictable in States 3, 4 and 5 as it contains faults that are irreparable (states 3 and 5) or that cannot be detected in the baseline router (state 4).

Let us define the following variables necessary to calculate the occurrence probabilities of the different states:

- R: probability that the baseline router contains no permanent fault. F is the complement of R.
- k: permanent-fault coverage of the PFD. kF is the probability that the baseline router contains faults which are all detectable by the PFD and $(1-k)F$ that it contains some undetectable fault(s).
- R_D: probability that the PFD contains no permanent faults. F_D is the complement of R_D.
- R_C: probability that the CC contains no permanent faults. F_C is the complement of R_C.

Using these variables, the occurrence probabilities P_1-P_5 of the States 1-5 read:

$P_1 = RR_D$; $P_2 = kFR_DR_C$; $P_3 = kFR_DF_C$; $P_4 = (1-k)\ FR_D$; $P_5 = F_D$

Since the five considered states account for all the possible fault states of the FTR, the above probabilities verify the normalization identity: $\sum_{i=1}^{5} P_i = 1$. The probability F_P that the FTR is in an unpredictable (i.e., in states 3, 4, or 5) is therefore:

$$F_P = P_3 + P_4 + P_5 = 1 - P_1 - P_2 = 1 - R_D(R + kFR_C) \qquad (1)$$

In what follows, we extend the reliability model by considering splitting up the NoC router into M sub-blocks. The following parameters are defined for a sub-block i among M sub-blocks:

R_i: probability that the sub-block i contains no permanent faults. F_i is the complement of R_i.
R_{Ci}: probability that the CC for sub-block i contains no permanent faults. F_{Ci} is the complement of R_{Ci}.
The reliability of the protected router can be rewritten as follows:

$$R_P = R_D \prod_{i=1}^{M} (R_i + kF_iR_{Ci}) \qquad (2)$$

As we already stressed, the FTR is more reliable than the baseline router if $F_P<F$. To enable the comparison, let us assume the sub-block i is made up of N_i transistors with each transistor having failure probability as p. Based on this assumption, R_i can be approximated as follows:

$$R_i = e^{(-pN_i)} \qquad (3)$$

Using (3) we obtain: $R_i = (1-F)^{N_i/N}$ (4)

where N is the total number of transistors in the baseline router. Substituting (3) and (4) in (2), we can calculate F_P directly as a function of F.

5. PERFORMANCE EVALUATION

In this section, we evaluate the ROBUST design in terms of reliability improvement.

5.1 Reliability Improvement Analysis

Since protection in ROBUST is achieved using BIST for permanent-fault diagnosis and ULBs as spares, we chose to compare it with BulletProof C_2SP_BIST and Vicis router architectures which also achieve protection using BIST for diagnosis and some additional logic for the purpose of repair (see section 2). Standard metrics such as MTBF (Mean Time Between Failures), MTTR (Mean Time to Repair) and Availability are not suitable to estimate the benefits of adding STAR circuits to cure permanent faults. The dependability for BulletProof and Vicis designs was previously analyzed using a metric called *Silicon Protection Factor* (SPF) proposed by Constantinides *et al.*[x]. SPF is defined as the ratio of the mean number of defects required to cause a router failure to the area overhead of the protection technique used. Thus, it measures the efficiency of the transistors in the STAR overhead to tolerate faults in the baseline router, and consequently, it does not tell whether the full FTR (that is to say the baseline router plus the STAR circuitry) is less prone to irreparable faults than just the baseline router. Indeed, it is obvious that if one adds some STAR circuitry as complicated as the baseline router, the improvement in reliability of the full FTR is expected to be small (to say the least) because the irreparable faults in the STAR circuitry will replace the irreparable faults in the baseline router. We propose a new reliability metric called *Reliability Improvement Factor* (RIF). It is defined as *the ratio of the probability F that permanent faults occur in the baseline router to the probability F_P that irreparable permanent faults occur in the full FTR* (i.e., in the baseline router plus all the STAR overhead). RIF can be computed using (2) as:

$$RIF = \frac{F}{F_P} \qquad (5)$$

F_P is calculated using the reliability model described in section 4. Therefore, we would be requiring the transistor counts for each of the sub-blocks that are protected by the CC, the transistor count of the CC for each sub-block, and that of the PFD (BIST). For this purpose an RTL model of the baseline router design was synthesized using Synopsys Design Compiler considering TSMC 90nm technology library at a clock frequency of 500 MHz and an operating voltage of 1V. The area occupied by each of the sub-blocks was obtained and divided by the area of the 2-input NAND gate in TSMC 90nm to obtain an approximate gate count. The transistor count of the sub-blocks was calculated from the gate count. The transistor counts for the BulletProof and Vicis designs were obtained from the area of each portion of the router. This methodology to evaluate transistor count is based on the assumption that the density of transistors per unit area is approximately homogenous across the circuit. Since RIF evaluation is based on the ratio of the transistor counts of the various portions of the router (see Eq. 4) and not the absolute transistor counts, we believe that this is a fair assumption.

Table 2: Transistor counts of routers (units: kTransistor)

Design	ROBUST	BulletProof	Vicis
Baseline router (N)	200.8	200.8	200.8
Correction Circuitry (N_{CC})	84.9	657.2	92.8
BIST block (N_B)	29.6	29.6	29.6

Table 2 shows the transistor counts for the ROBUST, BulletProof and the Vicis architectures. For ROBUST, it is the direct result from our implementation of the router models as described in Section 3.4. It should be noted that the correction circuitry transistor count for ROBUST router includes the transistor counts of the correction circuitry for each of the six sub-blocks. Here, the correction circuitry for the first five sub-blocks (each of the five input ports) account for a transistor count of 60,525 transistors and the correction circuitry for the sixth sub-block (crossbar switch) accounts for 24,335 transistors. The correction circuitry transistor count includes the transistor counts of the ULB_{hybrid} blocks and the additional MUXes added to aid the replacement of the faulty functional unit(s) from the router. In our analysis, we considered the C_2SP_BIST configuration of the BulletProof design for the purpose of comparison. In the C_2SP_BIST configuration, the router is partitioned such that each component is protected using two additional spares. Additionally, the design uses BIST in order to diagnose permanent faults. Therefore, in Table 2, the correction circuitry for the BulletProof design corresponds to the hardware overhead of the two additional spares and some additional logic used to aid the replacement of faulty components. Similarly, correction circuitry for the Vicis design corresponds to the extra logic involved in enabling the port-swapping, the crossbar bypass bus and the ECC units added for protection. In order to make the comparison fair, we assume that the fault-coverage (k) and the transistor count of the BIST block to be the same for all the designs.

The RIF for the three design configurations is displayed in Figure 5.

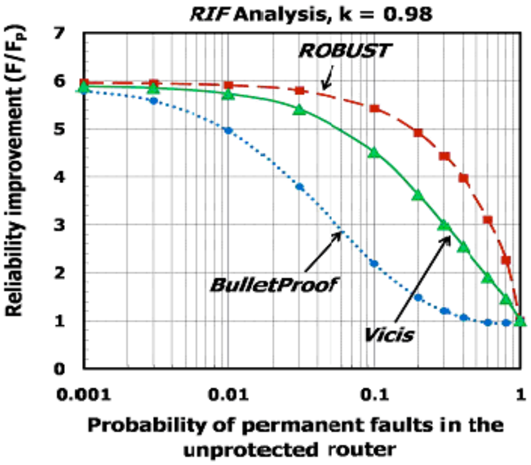


Figure 5. RIF analysis when k = 0.98

In the first step, we calculate the RIF when the fault coverage k is set as 98% [xvi]. We consider other values of k later in this section. Figure 5 shows that for all the designs, RIF gradually diminishes as the probability of permanent fault occurrence in the baseline router (F) increases. This indicates that all the protection mechanisms become inefficient when fault probability is high. When F is very low (in the order of 0.1%), all the protection mechanisms reduce the probability of incurable fault-occurrence (IFO) in the protected router by a factor of six, which corresponds actually to the ratio N/N_B. However, when F increases above 10%, the ROBUST design provides considerably higher reliability improvement than the BulletProof and Vicis, which drop considerably. This difference is a crucial advantage of the ROBUST router as it enables to maintain communications in massively defective NoC architectures when the fraction of defective routers exceeds R_T≈40% [v]. As we already stressed in the introduction, this threshold is intrinsic to 2D meshes, purely topological, and independent of communication protocols. Figure 5 shows that when the baseline router is 50% likely to encounter permanent faults, ROBUST reduces the IFO by a factor slightly larger than three, and consequently, the average fraction of defective router in NoCs should drop around 10%., preserving whereas BulletProof and Vicis will fail.

We also recalculated the RIF of the different designs assuming k=1 as shown in 6(a). Figure 6 (a) is identical to Figure 5 except that we see an improvement in obtained RIF of nearly 10% in this case.

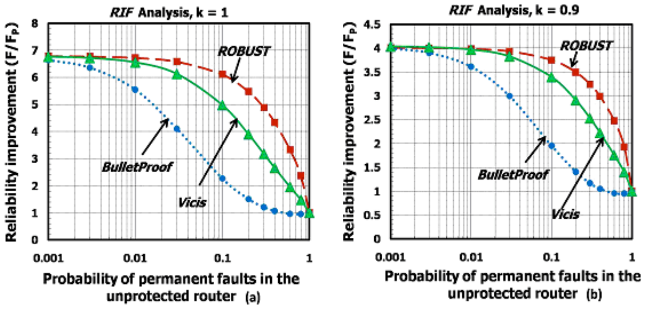


Figure 6. RIF analysis when (a) *k* = 1 and (b) *k* = 0.9.

On the other hand, one may wonder what would be the consequence of decreasing *k*. We recalculated the RIF of the ROBUST designs assuming *k* = 0.9 as shown in Figure 6(b). Figure 6(b) is identical to Figure 5 except that we see a decrease in RIF of nearly 35%. In this case, the test duration would be reduced to approximately 80 test patterns. Once again, while the time for testing is greatly reduced, the drop in reliability improvement achieved does not make this a good choice for *k*. Therefore, we conclude that setting *k* = 0.98 is a good compromise between the reliability improvement achieved and the time required for testing.

It is important to stress that, while our results are based on 90nm technology libraries, changing the technology should not have a major impact on the reliability benefits obtained by ROBUST. This is because, while the device dimensions will shrink across technology generations, the approximate transistor density in a unit area will be relatively similar. Since our analysis is based on the transistor counts, the RIF results will not change with change in technology libraries.

6. Conclusion

In this work, we proposed ROBUST, a new self-healing NoC router to autonomously diagnose and repair multiple permanent faults. ROBUST routers utilize reconfigurable multi-functional blocks called ULBs for the purpose of repair. Our results show that the ROBUST design provides better reliability improvement as compared to the Vicis and BulletProof solutions in the framework of a new reliability metric called *Reliability Improvement Factor* (RIF). The most appealing property of the ROBUST design is that it is efficient in massively defective technologies, when the average number of faulty baseline routers exceeds 30-40%. In this case, it is able to cure most routers, reducing the number of faulty routers in the NoC typically by a factor 4.

7. Acknowledgements

This research was partially supported by NSF grants ECCS-0725765 and CCF-0915537.

8. References

[i] S. Borkar, “Designing Reliable Systems from Unreliable Components: The Challenges of Transistor Variability and Degradation,” IEEE Micro, vol. 25, no. 6, pp. 10-16, November 2005.

[ii] R. Marculescu, U. Ogras, L.-S.Peh, N. Jerger, and Y. Hoskote, “Outstanding Research Problems in NoC Design: System, Microarchitecture and Circuit Perspectives,” IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems, vol. 28, pp. 3-21, January 2009.

[iii] M. Pirretti, G. M. Link, R. R. Brooks, N.Vijaykrishnan, M. Kandemir, and M.J. Irwin, “Fault tolerant algorithms for network-on-chip interconnect,” in Proceedings IEEE Computer Society Ann. Symp. on VLSI, pp. 46-51, 2004

[iv] T. Schonwald, J. Zimmermann, O. Bringmann, and W. Rosenstiel, “Fully Adaptive Fault-Tolerant Routing Algorithm for Network-on-Chip Architectures,” in 10th Euromicro Conf. on Digital System Design Architectures, Methods and Tools (DSD), pp. 527-534, 2007

[v] P. Zajac, J. Collet, and A. Napieralski, “Self-Configuration and Reachability Metrics in Massively Defective Multiport chips,” in Proceedings of 14th IEEE Int’l On-Line Testing Symp., pp. 219-224, 2008

[vi] http://en.wikipedia.org/wiki/Percolation_threshold

[vii] C. Grecu, P. Pande, A. Ivanov, R. Saleh, "BIST for network-on-chip interconnect infrastructures," in Proc. of 24th IEEE VLSI Test Symposium, pp.35, 2006.

[viii] T. Lehtonen, P. Liljeberg, and J. Plosila, “Online reconfigurable self-timed links for Fault-tolerant NoC.” VLSI Design, pp. 1-13, 2007

[ix] J. Kim, C. Nicopoulos, D. Park, V. Narayanan, M. Yousif , and C. Das, “A Gracefully Degrading and energy-efficient modular router architecture for on-chip networks,” in Proceedings of 33rd Ann. Int’l Symp. On Computer Architecture, pp. 4-15, 2006

[x] K. Constantinides, S. Plaza, J. Blome, B. Zhang, V. Bertacco, S. Mahlke, T. Austin and M. Orshansky, “BulletProof: a Defect-Tolerant CMP switch architecture,” in Proceedings of 12th Int’l Symp. On High Performance Computer Architecture, pp. 5-16, 2006.

[xi] M.Koibuchi, H. Matsutani, H. Amano, and T. M. Pinkston, “A Lightweight Fault-Tolerant Mechanism for Network-on-Chip,” in Proceedings of 2nd ACM/IEEE Int’l Symp. On Networks-on-Chip, pp. 13-22, 2008

[xii] D. Fick, A. DeOrio, J. Hu, V. Bertacco, D. Blaauw, and D. Sylvester, “Vicis: a reliable network for unreliable silicon,” in Proceedings of the Design Automation Conf. pp. 812-817, San Francisco, CA, USA, 2009

[xiii] L.-S. Peh and W. J. Dally, “A Delay Model and Speculative Architecture for Pipelined Routers,” in Proceedings of the 7th Int’l Symp. On High-Performance Computer Architecture, pp. 255-266, 2001.

[xiv] P. Kundu, “On-die interconnects for the next-generation CMPs,” in Proceedings of Workshop on On- and Off-Chip Interconnection Networks for Multi-core Systems, 2006.

[xv] M. Amory, E. Brio, E. Cota, M. Lubaszewksi, and O.G.Moraes. “A Scalable Test Strategy for Network-on-Chip Routers,” in Proceedings of IEEE International Test Conference, 2005. pp. 591-599, 2005.

[xvi] S.-Y. Lin, W.-C. Shen, C.-C. Hsu, C. –H. Chao, and A. –Y. Wu, “Fault-Tolerant Router with Built-In-Self-Test/Self-Diagnosis and Fault Isolation circuits for 2D-Mesh Based Chip Multiprocessor Systems,” Int’l Journal of Electrical Engineering, vol. 16, no.3, pp. 213-222, 2009.