A New Low Cost Fault Tolerant Solution for Mesh based NoCs

Mehrdad Seyrafi, Arghavan Asad, Amir Ehsani Zonouz, Reza Berangi, Mahmood Fathy, Mohsen Soryani
Department of Computer Engineering, Iran University of Science & Technology
Tehran, Iran
Email: {Mehseyrafi, ArghavanAsad, Amir.Ehsani.Z}@gmail.com
{RBerangi, MahFathy, Soryani}@iust.ac.ir

Abstract— In this paper a new fault tolerant routing algorithm with minimum hardware requirements and extremely high fault tolerance for 2D-mesh based NoCs is proposed. The LCFT (Low Cost Fault Tolerant) algorithm, removes the main limitations (forbidden turns) of the famous XY. So not only many new routes will be added to the list of selectable paths as well as deadlock freedom, but also it creates high level of fault tolerance. All these things are yielded only by the cost of adding one more virtual channel (for a total of two).

Results show that LCFT algorithm can work well under almost bad conditions of faults in comparison with the already published methods.

Keywords- LCFT; Fault Tolerant; Routing Algorithm; Network-on-Chip

I. INTRODUCTION

Following the reduction in feature size of semiconductor technology and the Moore’s Law, the number of transistors increases enormously and enables us to integrate complicated System-on-Chip (SoC) designs. Because of communication challenges in SoC architectures, NoC solution is becoming the main solution for more performance improvement than the traditional communication structures.

The high density in VLSI circuits makes them more susceptible to faults. VLSI faults can be categorized into two categories: permanent and transient faults.

Permanent faults are faults that may paralyze whole functionality of a router or only a number of its ports or maybe the links between the nodes. This type of fault is mostly because of accelerated aging effects like electromigration [1].

Transient faults occur mainly due to Gaussian noise on the channels, Alpha-particle strikes on logics, crosstalk, EMI and so on [2].

Occurrence of faults is one of the most challenging problems that researchers are facing with. There are a couple of methods to make on-chip-networks fault tolerant: Hardware redundancy, broadcast (flood), non deterministic routing algorithms, multipath (software redundancy). Each one has its own pros and cons. There is a tradeoff between them in cost, latency, fault tolerance, complexity, processing overhead and so on. Moreover the degree of fault tolerance in each algorithm can be calculated and compared with the others with respect to its effect on the other performance parameters as well as the cost.

In [3] a turn based routing algorithm is proposed for a single fault, but it cannot handle a faulty region containing more than one faulty router [4]. In [5] a turn based routing algorithm is presented. The algorithm creates a region containing all the faulty nodes, wrapping the faults, and bypasses this area by neighboring nodes. It is clear that this algorithm works badly in the cases that the region becomes greater and greater as a high traffic must pass through the neighbors to the region.

In [1] an adaptive fault tolerant algorithm with a good degree of fault tolerance but at the cost of a large area has been proposed. This large area increases the probability of fault occurrence in its routers. This large area corresponds to the rich set of tables and also the three priority queues used in the algorithm. In addition, there is no deadlock avoidance mechanism, nor deadlock recovery mechanisms.

A. Patooghy et al. in [6] proposed a XY and YX based algorithm, namely XYX that uses software redundancy to transmit packets to their destinations. The algorithm claims that both permanent and transient faults can be detected.

A fault tolerant deadlock free algorithm that defined a default path in each router from input ports to an output port is proposed in [7]. When a router becomes faulty, the default path only forwards the incoming packets to the default output port. The set of default paths in the network forms a Hamiltonian cycle. It dedicates a good level of fault tolerance. The only disadvantage is that it needs some hardware redundancy in the router logic and also it may cause the path to become too long with a high likelihood.

W. Song et al. in [8] select the next direction randomly in their proposed fault tolerant algorithm. The algorithm handles deadlock cases with a timer, to release one of the streams involving the deadlock.

A. Ehsani et al. in [9] presented a new solution, namely DCS, for 2D-mesh structures in which each core is connected to two neighboring routers. This foundation saves cores with a faulty router from being disconnected from the network infrastructure. They have shown that the latency decreases in the absence of faults because of the hardware redundancy used in it. The algorithm doesn't solve the deadlock problem.

This paper proposes a new routing algorithm, namely LCFT (Low Cost Fault Tolerant), based on turn model. The new algorithm sends two copies of each packet (flit) to their...
destination by two sub-algorithms. Note that these two algorithms which are appeared firstly in this paper can be used for routing independently.

By using both of these two sub-algorithms simultaneously, the result will be an extremely high fault tolerant routing algorithm.

The proposed routing algorithm is evaluated analytically. Also a simulator based on SystemC was developed and used for evaluation and verification. The algorithm is compared with a number of already existing algorithms.

We consider that each packet in the NoC consists of several consecutive 64-bit flits starting with a header flit, following by data flits, and ending with a tail flit. The header flits reserve channels on the path and tail flits release them.

The remaining of the paper is organized as follows: In section 2 the proposed routing algorithm is deeply described. In section 3 some important metrics is discussed. Analytical and simulation based results appear in section 4. At the end section 5 concludes the paper.

II. PROPOSED LCFT ALGORITHM

The proposed algorithm is defined for 2D-mesh based NoCs. The LCFT routing algorithm routes packets based on semi-local information of faulty nodes and links. Each router acquires faulty status of all nodes and links in its column and row by a special mechanism using configuration flits. So each router knows the reachable distance (without encountering any fault) in each direction (north, south, east and west). For example in Figure 1, node 19 can only send data packets up to two hops away by its west port.

Each router has four registers for its four ports, namely Domain-Register (DR). Each DR entry bit width is \( \log_{2^{N_i-1}} \), where \( N_i \) is the size of \( i \)-th dimension for \( i=1,2 \). Each of these registers stores the available hop count passing the correlated port. These registers are initialized by all 1 value (maximum possible value) in the manufacturing process and just after each chip reset.

The detection of fault on routers and links can be done by the popular methods such as BIST [10] and also methods introduced in [11-13]. After each fault occurrence, all four neighbors of the faulty node (or the two nodes in the case of fault on a link), detect the event, generate a single flit packet, namely configuration flit, and transmit it to the opposite direction. All receiving nodes forward the flit to their opposite direction. This can be seen in Figure 1 in which node 39 got faulty and its neighbors, nodes 32, 38, 40 and 46, each create configuration fits and forward it to their next nodes until each flits reach borders.

The configuration flit carries a TTL (time to live) value, initialized at 1 in the source node. Each node stores TTL in its DR entry for the direction that packet came from, increments it and forwards the flit to the next router. The configuration flit has the highest priority, so all of the nodes in the fault column and row will be aware of fault so quickly.

An example DR value for router no 19 is shown in table I. The DR of east port contains value 111, which means there is no faulty router, nor faulty link in the east direction of router no 19.

TABLE I. DOMAIN-REGISTER VALUES OF ROUTER NO 19 OF FIG. 1

<table>
<thead>
<tr>
<th>Direction</th>
<th>Domain-Register values</th>
</tr>
</thead>
<tbody>
<tr>
<td>North</td>
<td>111</td>
</tr>
<tr>
<td>West</td>
<td>010 2</td>
</tr>
<tr>
<td>East</td>
<td>111 ( \infty )</td>
</tr>
<tr>
<td>South</td>
<td>000 0</td>
</tr>
</tbody>
</table>

The data transmission in LCFT is done by transmitting the data through two paths simultaneously and strictly separated. One of these paths is based on Extended-XY and the other based on Extended-YX routing algorithms. Both of these two algorithms need 2 virtual channels: vc0 and vc1. Each of these algorithms achieves some degree of fault tolerance. LCFT uses both of these algorithms and so gains extra high degree of fault tolerance. These algorithms, which are proposed for the first time in this paper, are based on simple XY and YX algorithms. These are based on turn model just as their ancestors, preserving the deadlock freedom property.

A. Extended-XY

The algorithm is based on simple XY routing algorithm, a dimension order based algorithm, in which packets are routed firstly along the X dimension and then along the Y dimension until they reach their destination. The Extended-XY algorithm functionality is depicted in Figures 2.a-c. The blue lines show paths defined by the Extended-XY algorithm. The algorithm includes 6 phases.

According to the fault conditions in each path, some of these phases may be absent. For example after performing first phase, it may be needed to perform third phase, and there may be no need to perform phases 2, 4, 5 and 6.

The LCFT routing algorithm works as follows. Firstly packets are transmitted in X dimension in vc0 approaching to destination (phase 1). There must be no fault in the path until packets reach the destination column (otherwise the
algorithm fails and discards the packets). Now the header flit must be in the destination column. In the destination column, if DR shows that there is no fault in Y dimension till destination, the packets are transmitted directly to destination in vc0 (just such as simple XY) (phase 3). Else in the case of one or more fault in the Y dimension, packets are forwarded one more node in X dimension (phase 2). If DR allows transmission in Y dimension, they will be sent to the direction approaching to the destination in Y dimension an in vc0 (phase 3). The data packets are routed in Y dimension until they reach the destination row. In phase 4 of LCFT algorithm, LCFT finds the first row that the data packets can reach the destination itself or destination column and then transmit data in X dimension and in vc1 (channel switch from vc0 to vc1) (phase 5).

![Figure 2. All possible paths from node S to node D in Extended-XY and Extended-YX routing algorithm](image)

This is the only virtual channel switching in Extended-XY. The single lines in Figure 2 show that vc0 is used and dashed lines are used for vc1. The last phase (phase 6) includes routing data in Y dimension using vc1, if header flit is not now in the destination node.

If source and destination nodes are in a same column, phase 1 (of 6) is dismissed and in the first step in phase 2, packets can be forwarded to left or right selectively, which one is better (i.e. the direction with more distance to the borders) . If source and destination nodes are both in same row, only phase 1 is needed. So in this case there will be no fault tolerance.

### B. Extended-YX

This routing algorithm is similar to Extended XY except that phase 1 and 2 are not included in the process. The functionality of the algorithm is depicted in Figure 2. The red lines mark all possible paths in Extended-YX routing algorithm.

![Figure 3. Examples of data transmission in presence of faults with (a) Extended-XY and (b) Extended-YX](image)

#### C. LCFT Routing Algorithm

The LCFT routing algorithm uses the Extended-XY and Extended-YX algorithms simultaneously. So the fault tolerance property of the combined algorithm improves inordinately. There are two special cases that seem they need to be handled differently. In Figures 2b and 2c, the two cases are depicted in which source and destination nodes are placed in same row or same column.

### III. IMPORTANT METRICS

#### A. Fault Tolerance

- **Permanent Faults:** The LCFT routing algorithm tolerates permanent faults, as it is explained previously. This type of fault can arise only in limited areas, known as single faults, or arise in a contiguous block of more than one router and link. All these types of faults can be effectively covered by the proposed algorithm.

- **Transient Faults (errors):** Because LCFT sends two copies of all packets to each destination, it is possible for destination nodes to select the non erroneous one. This can be done using error detection mechanisms, such as parity bits or CRC.

#### B. Throughput

In LCFT algorithm the traffic seems to be two times higher than regular algorithms that send only one copy of packets. But there are some facts that must be considered too. The most important fact is that in the presence of transient errors, these algorithms do not guarantee error freedom. So retransmission is common in these algorithms. Also adding virtual channels to a network reduces the dependence of throughput on the number of stages in the network, so existence of the additional virtual channel used in LCFT, improves throughput. Nonetheless, in the presence of permanent faults, some of paths lengthen, so raising traffic in the newly added nodes to source-destination paths and degrading throughput. However due to these faults, some of paths are blocked and so the traffic will decrease in the rest of these paths.
C. Contention Free Latency

The contention free latency of LCFT and pure XY as a minimal routing algorithm, are almost same. The reason is that at least (with an extremely high probability) one of the paths will not face with faults, and so it is minimal. But latency with considering the contention in high injection rates, is higher than XY.

D. Deadlock Freedom

Theorem: The LCFT routing algorithm is deadlock free.

Proof: The LCFT routing algorithm is basically based on the turn model, so its deadlock freedom can be proved. All the allowed turns in LCFT is shown in Figure 4. The solid lined are paths through VC0 and dashed lines for VC1. Channel switchings is also shown in the figure. As it is evident, there is no cyclic dependency in each virtual channel alone. Because the set of allowed turns in each virtual channel separately is just equal to (or a subset of) the set of allowed turns in XY routing algorithm. Also there is not any cyclic dependency between the two virtual channels (as packets are turns in XY routing algorithm. Also there is not any cyclic dependency in each virtual channel (as packets are passed between virtual channels only in increasing order). Therefore it can be stated that there is no deadlock situation possible in LCFT routing.

![Figure 4. Allowed turns in LCFT routing algorithm](image)

E. Probability of Successful Transmission

In this section the probability of a packet being successfully transmitted to a destination is calculated and compared with two other algorithms.

At first this parameter will be calculated for XY simple algorithm in which no fault tolerance in set. The probability of XY routing algorithm for a typical packet, \( P_{xy} \), is:

\[
P_{xy} = P^m \times P^{n-1} = P^{m+n-1}
\]

In which:

\[ m = \text{abs}(x_{dst}-x_{src}+1) \]
\[ n = \text{abs}(x_{dst}-x_{src}+1) \]

And \( P \) is the probability of each single router being healthy. Equation 1 shows that \( m+n-1 \) routers must properly work in order to the typical packet be able reach its destination.

The probability of successful transmission of XYX routing algorithm proposed in [6] which send each all packets by both XY and YX, can be calculated as follows:

\[
P_{XY} = 1 - (1 - P_{xy}) \times (1 - P_{xy})
\]

\[
= 1 - (1 - P^{m+n-1}) \times (1 - P^{m+n-1})
\]

\[
= 1 - (1 - P^{m+n-1})^2
\]

Each core in DCS presented by [9] is connected to two neighboring routers, and so each packet will be sent by using the router which is better (in terms of hop count, fault status and traffic). The algorithm is a modified version of XY.

The probability of successful transmission in DCS can be formulated as:

\[
P_{DCS} = P^m \times (1 - (1 - P^m) \times (1 - P^{n-1}))
\]

(3)

The LCFT routing algorithm presented here sends each packet into two directions by two sub algorithms (Extended-XY and Extended-YX). Probability of successful transmission of each of these two (sub) algorithms is as follows:

\[
P_{EXT} = P^{m+n-1} \times (1 - (1 - P^{m+n-2}) \times (1 - P^{m+n-2}))
\]

\[
= P^{m+n-1} \times \sum_{i=0}^{w} (1 - P^{m+n-2})^i
\]

(4)

And

\[
P_{EXT} = P^{m+n-1} \times (1 - (1 - P^{m+n-2}) \times (1 - P^{m+n-2}))
\]

\[
= P^{m+n-1} \times \sum_{i=0}^{w} (1 - P^{m+n-2})^i
\]

(5)

In which \( w \) is the distance of destination node to the left or right border which one is lower and \( v \) is the distance of destination node to the top or bottom border of the network which one is lower.

The probability of successful transmission of a typical packet for LCFT routing algorithm is as follows:

\[
P_{LCFT} = 1 - (1 - P_{EXT}) \times (1 - P_{EXT})
\]

(6)

The probability of successful transmission for total network and for random traffic pattern can be calculated by averaging all the possible transmissions from all nodes as source to all other nodes as destination. As shown in equation 7.

\[
P_{Total\_Transmission} = \frac{\sum_{i=1}^{Dx} \sum_{j=1}^{Dy} \sum_{k=1}^{Dx} \sum_{l=1}^{Dy} (P_{Algorithm} | (i,j) \neq (k,l))}{\sum_{i=1}^{Dx} \sum_{j=1}^{Dy} \sum_{k=1}^{Dx} \sum_{l=1}^{Dy} (1 | (i,j) \neq (k,l))}
\]

(7)

In which \( P_{Algorithm} \) is either of \( P_{xy}, P_{XYX}, P_{DCS} \) or \( P_{LCFT} \).

F. Average Path Length Calculation

The Average path length of a transmission is the number of hops that a packet must traverse until it reaches its destination. This parameter is an estimation of packet transmission latency which can be mathematically (exactly) calculated.

Note that some of the parameters used in this section, are brought from the previous section (Probability of successful transmission) such as \( m \) and \( n \), and are not redefined here.

Let \( \Gamma_{m,n} \) be the Manhattan distance between two nodes.

\[
\Gamma_{m,n} = m + n - 1
\]

(8)

In which:

\[ m = \text{abs}(x_{dst}-x_{src}+1) \]
\[ n = \text{abs}(x_{dst}-x_{src}+1) \]

In the case of some faults in the network, LCFT routing algorithm average path length may vary (i.e. lengthens). The
path length from a source to a destination by EXY sub algorithm can be formulated as:

\[
\Gamma_{m,n,\text{EXY}} = \Gamma_{m,n,\text{EXY} \neq 0} + (\Gamma_{m,n} + 2)\times P_{\text{EXY} \neq 1} + \sum_{j=0}^{\infty} [(\Gamma_{m,n} + 2j)\times P_{\text{EXY} \neq j}]
\]

(9)

In which \(P_{\text{EXY} \neq j}\) is the probability that the packet must pass away \(j\) nodes from the destination column because of faults. Path length for EYX sub algorithm is as follows:

\[
\Gamma_{m,n,\text{EYX}} = \Gamma_{m,n,\text{EYX} \neq 0} + (\Gamma_{m,n} + 2)\times P_{\text{EYX} \neq 1} + \sum_{j=0}^{\infty} [(\Gamma_{m,n} + 2j)\times P_{\text{EYX} \neq j}]
\]

(10)

In the case that both sub algorithms find a route to the destination, the shorter one is determined here to be our path. So the probability that path found by EXY sub algorithm is better than the EYX is

\[
P_{\text{EXY preferred}} = \frac{(\Gamma_{m,n,\text{EXY} \neq 0} + \sum_{j=2}^{\infty} P_{\text{EXY} \neq j}) + (\Gamma_{m,n,\text{EXY} \neq 1} + \sum_{j=2}^{\infty} (\Gamma_{m,n} + 2j)\times P_{\text{EXY} \neq j})}{(\Gamma_{m,n,\text{EYX} \neq 0} + \sum_{j=2}^{\infty} P_{\text{EYX} \neq j}) + (\Gamma_{m,n,\text{EYX} \neq 1} + \sum_{j=2}^{\infty} (\Gamma_{m,n} + 2j)\times P_{\text{EYX} \neq j})}
\]

(11)

And similarly for PEYX being better is:

\[
P_{\text{EYX preferred}} = \frac{(\Gamma_{m,n,\text{EYX} \neq 0} + \sum_{j=2}^{\infty} P_{\text{EYX} \neq j}) + (\Gamma_{m,n,\text{EYX} \neq 1} + \sum_{j=2}^{\infty} (\Gamma_{m,n} + 2j)\times P_{\text{EYX} \neq j})}{(\Gamma_{m,n,\text{EXY} \neq 0} + \sum_{j=2}^{\infty} P_{\text{EXY} \neq j}) + (\Gamma_{m,n,\text{EXY} \neq 1} + \sum_{j=2}^{\infty} (\Gamma_{m,n} + 2j)\times P_{\text{EXY} \neq j})}
\]

(12)

It can be simply shown that the average path length for a source and destination is:

\[
\Gamma_{w,n} = \Gamma_{m,n,\text{EXY}} \times P_{\text{EXY is preferred}} + \Gamma_{m,n,\text{EYX}} \times P_{\text{EYX is preferred}} + \sum_{i=1}^{\infty} [(\Gamma_{m,n} + 2i)\times P_{\text{EXY} \neq i} \times P_{\text{EYX} \neq i}]
\]

(13)

It is only sufficient to calculate the average path length of a network for random traffic pattern by

\[
\Gamma = \sum_{i=1}^{D_x} \sum_{j=1}^{D_y} \sum_{l=1}^{D_x} \sum_{k=1}^{D_y} \sum_{i=j}^{(i,j)\neq(k,j)} \left( \Gamma_{i-j-1,i-j-1} \times |(i,j)\neq(k,j)| \right)
\]

(14)

IV. RESULTS

The evaluation of LCFT routing algorithm and comparing it with some other existing algorithms are done both analytically and with help of simulation. Analyses are done with MATLAB software, for its simplicity and accuracy. These analyses are based on the formulas presented in the paper.

Also simulations are done in a detailed simulator implemented in SystemC.

All the results are based on a \(10\times10\) mesh NoC, and wormhole scheme is used as switching strategy. Packets are divided into three types of flits: header, data and tail flits.

Figures 2, 3 and 4 show the probability of successful transmission under three popular network traffic patterns. In these figures the X axis shows the probability of routers working properly.

The figures reveal that the proposed algorithm obtains a much higher probability which means that a network based on this algorithm can tolerate more faults.

The average path length in LCFT as compared with XY routing algorithm is depicted in Figure 8. As it can be seen the average path length is near minimal.
Figure 8: Average path length under three traffic patterns for LCFT and XY routing algorithms.

Figure 9: Average latency of LCFT as compared with XY, DCS and XYX algorithms.

Figure 10: Total number of flits passed in network in LCFT as compared with XY, DCS and XYX algorithms.

Figure 9 depicts the packet latency as a function of injection rate under uniform traffic pattern and in absence of fault. It shows that LCFT routing has a higher latency than the two other routing methods: XY and DCS, but is near the latency of XYX in higher injection rates. But in lower injection rates, i.e. lower than 0.0277, the method proposed in this paper can be almost tolerable.

In Fig. 10, the throughput of the LCFT in terms of total flits passed in the network as compared with the three routing algorithms is depicted. As it can be seen in the figure, the throughput of LCFT along with XYX is much lower than the two other algorithms.

V. CONCLUSIONS

This paper proposed three new routing algorithms, Extended-XY, Extended-YX and LCFT (Low Cost Fault Tolerant). The first two algorithms (which are successors of traditional XY and YX and based on turn model), dedicate some degree of fault tolerance. But using both of them in LCFT algorithm simultaneously makes the new algorithm an extremely reliable routing algorithm. It uses two virtual channels in order to avoid deadlocks. The performance evaluation presented in this text show that its fault tolerance property can well compete with other existing fault tolerant algorithms.

The fault tolerance of this algorithm is high enough to be an option to be used in future NoCs for critical applications. A feedback method from destination to source can be used in order to inform the source, to limit transmissions in only one of the two paths. So the algorithm can be partly customized to allow tradeoff between parameters such as throughput in high traffics, degree of reliability and so on, Leading to lower traffic and higher performance. This remains to be done in future works.

REFERENCES


