A withmany rule checking steps. based on network

A Highly Resilient Routing Algorithm forFault-Tolerant NoCsGaurav Khuranay, Sadia Moriam y, Emil Matus y, Gerhard FettweisyVodafone Chair Mobile Communications Systems, Technische Universit¨at Dresden, Dresden, GermanyyCenter for Advancing Electronics Dresden cfAED, Technische Universitt Dresden, GermanyAbstract—Product reliability in complex systems such as chipmultiprocessor (CMP) and system on chip (SoC) is very difficultto achieve due to combination of factors like transistor scalingand their large numbers in each system. In order to improve therobustness in interconnect networks, a network-on-chip (NoC)algorithm is presented in this work. Connectivity and correctoperation are maintained by reconfiguring them to prevent faultycomponents. Proposed solution overcomes many faults withoutusing adaptive routing or virtual channels, thus, preventing singlepoints of failure. Moreover, this work requires no disabled routersand no particular fault restrictions.

Furthermore, less than 300gates per network router are needed to implement this algorithmin hardware. I. INTRODUCTIONNetworks on chip utilize lightweight network protocol todecentralize and distribute communication, thus, giving a scalablesolution and high throughput.

Packets of information aretransmitted between IP components such as cores, caches etc. through distributed system of routers connected by links1. Inorder to build a reliable system NoC should be able to workaround the links and faulty routers2. Furthermore networkcan even disable a faulty IP with no or little protection, thus, improving system reliability. Ideally, faults should bediagnosed by the network and mitigated by the reconfigurationprocess, whenever possible, to facilitate full connectivity. II. ROUTING ALGORITHM

OVERVIEWThe algorithm reconfigures network routing table in anoffline method and consists of a basic routing step along withmany rule checking steps. Based on network topology andexisting faults, the rules constrain the basic routing step, thus, safely redirecting traffic around failed resources.

Output portfor each destination in the network is listed by each router inits routing table. Faults are modelled as link-level hard failures, therefore, each bidirectional link can be bypassed individually, letting routers to be partially functional. Each router shouldknow which of their adjacent links are not working andbased on this information these work with their neighboursto reconfigure their routing tables collectively.

Furthermore, the algorithm follows a procedure that updates an entry for aparticular destination in all routing tables. A. Basic Routing StepRouters communicate through flags and entries in the routingtable are marked either valid if corresponding destinationis reachable or invalid at any point during the implementationof algorithm. In order to determine the best course to thedestination all routers initialize their corresponding entry toinvalid initially, except the one which is connected locally tothe destination, thus marking its entry valid and right directionin the table. Later, all routers repeat the following steps untilevery router updates its entry: 1) Flag transmission: Routers which have valid destinationentry will send flag to all of their adjacent routers whileother routers will stay silent. 2) Routing entry update: Routers with invalid entry andhave received a flag through any of their links inpreceding step will update their entry to valid with incomingflag direction.

Also, if a router receives flag frommultiple links then the preferred direction of routing isselected by priority selection procedure. Router will nottake any action if it does not receive any flag or alreadyhas a valid entry. B. Basic Routing Step RulesSet of rules for each router to avoid deadlock loops formationwhile executing basic routing step: 1) Links: It can be disallowed by not transmitting flagsacross them or by not accepting flags which are transmitted. Thus it can be disallowed by either of the routersit connects to. 2) Turns: It consists of two links connected by a routerand is disallowed by their centre router by preventingtransmission of flag from one to the other.

III. 2D-MESH ROUTINGIn 2-D mesh network, basic routing step is reused toevaluate which rules are required to prevent deadlock. Defaultrules are thus removed or adjusted depending on set of faultylinks. A. 2-D-Mesh RulesNo deadlock loop forms in the top row of figure 1 since S-EW-N priority leaves Northeast and Northwest corners naturallyunused. However, a deadlock situation arises when a faultis present (Figure 1, second row) in which loop of utilizedturns is formed. In this case, minimum path length betweentwo routers is every turn in the loop around the fault. Glassand Ni avoid deadlock situations in adaptive routing networkby disallowing a pair of turns3.

According to the paper, both anticlockwise and clockwise turns (Northeast corner)Fig. 1. Disallowing turns to prevent deadlockwere disallowed at the same corner, thus, preventing deadlock(Figure 1 , third row). B. Selectively Removing 2-D-Mesh RulesStrict adherence to the disallowed turn rule may preventsome routers to reach Northwest destination even thoughpossible

paths exist (Figure 2, left). This is caused because theEast-North turn has been disabled for the western edge routersto avoid deadlock.

Therefore, in order to reinstate networkconnectivity, West router’s turn rule is removed (Figure 2, right). Fig. 2. Selectively removing rules for consistent networkC.

Pathological CaseRemoval of the corner rule for the Southwest corner (Figure3) leads to the formation of a deadlock loop which passes twicethrough that router. Thus, with the use of S-E-W-N prioritywhich disfavours Northwest corner, this problem is reduced. Fig. 3. Pathological CaseIV.

2D-TORUS ROUTINGTorus network forms deadlock loop even if there are nofaults, thus, the basic routing step is a bit challenging. A. 2-D-Torus RulesLoops are formed around the outside of network by progressingin the same direction until the same router is reachedagain. These are addressed with the usage of link rules. Formation of loop in the same row is avoided by horizontallinks while vertical link rules (along north edge) prevent azigzag pattern to form loop around the network as well as theloops which will form in the same column. Furthermore, theserules are checked. In the case of vertical link rules, one sidepreserves the rule while other side is routed and checked ifit’s entry is ever valid. If it is invalid then the rule is removedto maintain consistency.

B. Corner Rule ConsistencyIn some cases, tori can be inconsistent because the pathsmay be blocked around the outside of the network when oneturn versus another is routed. In certain scenarios, deadlockpath appears if

particular turn rule is discarded, but resultsinto inconsistent network if it is not allowed. So, it is resolvedby introducing a new link rule, thus, maintaining networkconsistency without deadlock. V. EXPERIMENTAL RESULTSAlgorithm is implemented on both 2-D-Mesh and 2-DTorustopologies with various sizes (4×4, 8×8, 12×12). Faultsare injected and network is allowed to reconfigure.

Resultingtables verify, whether properties like consistency of routingtables, no pointless cutting off routers and absence of deadlockcondition hold true. Experiment is repeated a million times foreach point of data. Fig. 4. Reliability versus broken linksReliability of over 99. 99% is achieved for all topologieswhen a tenth of links are broken (Figure 4).

Regardless ofnumber of broken links 4×4 networks show reliability of 100%for 2-D-Mesh and 99. 99% for 2-D-Torus. However, for largernetworks, as the number of faults increased beyond this point, the probability of a faulty network configuration increased.

Fig. 5. Packet latencyPacket latency for given number of broken links and trafficdensity is investigated for performance evaluation of 8×8 2-DTorusnetwork (Figure 5). The latency is below 20 cycles forlow traffic densities, however, as density increases, networksaturation results in latency wall. Moreover, with faults beeninjected, latency wall is shifted towards lower traffic densities. This is resulted due to presence of fewer active communicationpaths among routers and longer routes around failed components.

Due to random fault injection the onset of networksaturation vary as shown by highlighted region (5th to 95thpercentile). VI. CONCLUSIONSResults show reliability of 99. 99% across all topologies with10% broken links. The presented routing algorithm enableselegant degradation of performance as network componentsfail, while maintaining network correctness and connectivityin NoC architectures. Additionally, solution requires neithervirtual channels nor adaptive routing. REFERENCES1 T.

Bjerregaard and S. Mahadevan. A survey of research and practicesof network-on-chip. ACM Computer Survey, 38(1), 2006. 2 D.

Sylvester, D. Blaauw, and E. Karl. ElastIC: An Adaptive Self-HealingArchitecture for Unpredictable Silicon. IEEE Design Test, 2006. 3 C.

J. Glass and L. M.

Ni. The turn model for adaptive routing. ACMSIGARCH Computer Architecture News, 20(2), 1992.