Design of a fast, low-level fault-tolerant protocol for Network on Chips

Muhammad Ali[1], Awais Adnan[2], Michael Welzl[1]
[1]Institute of Computer Science, University of Innsbruck, Austria
[2]Institute of Management Sciences, Peshawar, Pakistan

Abstract: *Network on a chip (NoC) has been proposed to address the inefficiency of buses in the current System on Chips (SoC). However as the chip scales, the probability of errors is also increasing, thus, making fault tolerance a key concern in scaling chips. Transient faults are becoming a major cause of errors in a packet based NoC. A transient error can either corrupt the header or the payload of packet requiring a retransmission from the source. Due to retransmissions, packets arrive out of order at the receiver side. Complex reordering algorithms are required at the receiver side to organize packets before sending them to associated resource. This adds a major overhead as the storage and logic capabilities are limited on a chip. We therefore provide a low-cost, fast and reliable end-to-end protocol for NoC which does not require reordering of the packets at the receiver end. Our protocol performs bitwise logical operations using binary representation of the addresses for the buffers to handle packets, hence making it much faster than conventional algorithms. Furthermore, the protocol is equally applicable to both static as well as dynamic routing environments on a chip.*

Keywords: Fault Tolerance, Network on a chip, Transient / soft errors.

I. INTRODUCTION

The International Technology Roadmap for Semiconductors (ITRS) projects that by the end of this decade chips would accommodate billions of transistors [1]. This means that in near future Application Specific Integrated Circuits (ASIC) will be made up of hundreds of communicating partners. However, it has been observed that the existing on-chip interconnects pose a serious threat toward achieving the billion transistor era. Buses that form an integral part of today's SoCs show serious degradation of performance beyond a certain number of communicating partners [2]. Considering the unmatched success of the Internet especially in terms of scalability, VLSI researchers have borrowed ideas from computer networks and proposed a packet based communication model for SoCs. This new paradigm is termed as *Network on Chips (NoC)* [3]. The idea is to send data in the form of packets over a network of switches/routers on a chip. This way communication and computation are kept transparent from each other, hereby achieving reusability of components and scalability of the whole network on-chip.

The increasing number of transistors on a chip is also contributing toward an increase in the faults, both temporary and permanent. With decreasing die size, cross talk, critical leakage currents and high field effects will increase the probability of permanent and temporary faults on a chip [4]. Permanent faults may cause one or more on-chip links or switches to fail, causing permanent damage to the component. Temporary faults, also known as transient failures or soft errors, do not cause permanent damage but may scramble one or more bits in a packet, hence making it invalid [5].

The frequency of transient errors is much higher than the permanent faults on-chip, making it necessary to provide built-in error recovery mechanism. With increasing transistor density on-chip, the soft error ratio is increasing exponentially [6]. Since today's SoCs are integrated into consumer products, it is important to equip them with some degree of fault tolerance. This can increase the overall yield of the chips by reducing the cost of production.

In this paper we present a fast and reliable end-to-end protocol for NoCs which ensures safe delivery of packets from source to destination. Since the protocol is implemented in the end systems, no complex logic algorithms and excessive storage capability is required at the intermediate routers. The significant aspect of our protocol is its simplicity in implementation on a chip as it uses bitwise logical operations to process and store the packets, eliminating any need for complex reordering and storing algorithms.

The paper is organized as follows; in the next section we discuss issues regarding error tolerance in NoCs along with related work. In section III, we provide a brief discussion of the NoC model we are using. Section IV elaborates the description of our protocol in detail, followed by conclusion and future work.

II. FAULT TOLERANCE IN NOCs

Traditionally, chips are equipped with error detection and correction mechanisms to deal with transient faults. However, packet based communication on-chip brings new challenges in terms of error resilience. The primary unit of packet based communication is a packet itself. A packet is usually composed of a *header* and a *payload.* The header of the packet contains identification information like source and destination addresses, routing path, CRC (Cyclic Redundancy Check) checks etc. The payload contains the actual data to be transported. A soft error can scramble either header or the payload of a packet. A bit flip, due to a soft error, in the header of the packet may cause it to be routed to a wrong destination. An error in the payload, on the other hand, may corrupt the contents of the packet, hence making it invalid although it might reach the destination. In both cases a retransmission of the misrouted or corrupt packet is required.

T. Sobh (ed.), Innovations and Advanced Techniques in Computer and Information Sciences and Engineering, 345–348.

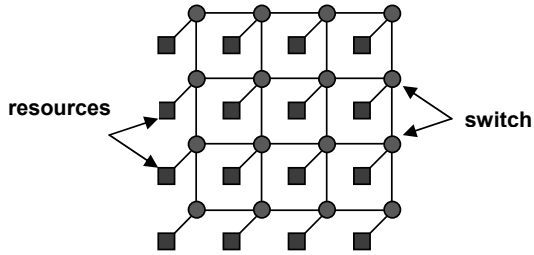

Figure 1: 2D mesh topology for NoCs

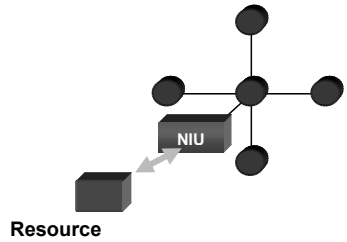

Figure 2: Abstract representation of NI

The authors of [7] have discussed and analyzed various error coding schemes for NoCs. In [8], T. Dumitras et al. present a stochastic communication model for NoCs based upon rumor spreading. In this mechanism a node spreads the packet to its neighbor which, in turn, spreads it to all its neighbors assuming that at least one packet will reach the destination. Though simple, the mechanism has high packet overhead especially when a large number of nodes are communicating. In [9], Bertozzi et al. present a link level flit[1] based retransmission protocol. Each flit is acknowledged by every intermediate router on successful reception. In case a flit is missing or is corrupt, a negative acknowledgement is generated. Such a mechanism adds complexity on part of intermediate routers besides buffering them.

We propose a fast, efficient, and fault tolerant scheme to deal with packet corruption due to transient errors in a NoC. The mechanism does not require intermediate routers to maintain buffers and thus forwards packets as they arrive. Moreover, the mechanism is equally effective in both dynamic and static routing environments of NoCs. The detailed description of the proposed protocol is discussed in section IV.

III. NOC MODEL

Various topologies are possible for NoCs like honeycomb [10], fat-tree [11], 2D mesh [12] etc. The choice of a topology can dramatically affect the network characteristics of a NoC in terms of number of hops, link delays etc. Although we intend to study the behavior of our mechanism with other topologies in future work, in this paper, we only discuss the most agreed upon topology for the sake of simplicity --- a *2D mesh*. A typical topology model is shown in figure 1 followed by its description:

> *Boxes* represent resources in a NoC which is an end system having data to send or receive. Each resource in this scenario is connected to a router. They are considered black boxes as they can be anything: a MIPS[2] processor, a DSP[3], a memory module or even a combination of any or all of these elements. Hence in terms of a set of resources, it can be termed as *network within a network.*
>
> *Circles* in figure 1 represent routers which are responsible for establishing a routing path between sender and receiver. Each router is directly connected to neighboring four routers (except for the ones at the edges), thus creating a mesh network.
>
> *Network Interface* (NI) is a special device acting as a middle layer placed between the resource and the router. NI is responsible for creating packets from bit streams obtained from the connected resource and vice versa. It is a very significant unit of the NoC as it covers the discrepancies that exist between the homogeneous router network and different kind of resources emanating from a variety of vendors. NI makes it possible to reuse not only the resources but also the network architecture of a NoC. A typical example of a NI is shown in figure- 2.

IV. DESCRIPTION OF THE PROTOCOL

As mentioned earlier, we propose an end-to-end reliable packet delivery mechanism, restricting the intricacies of the protocol to the sender and receiver nodes. The sender sends a pre-defined number of packets to the receiver and after making sure their safe delivery, sends the next set. If there are X packets in a set, it means the sender and receiver need X buffers to store them. In our example setup, we take it as 16, which means the sender and receiver have to buffer 16 packets at a single time. Each packet in a set is identified by a unique packet ID from 0 to 15 (*0000 -1111 at lower four bits of address*). The receiver associates a flag value with each buffer which is initially *zero* . When a packet arrives at the receiver, the error detection code checks it for any inconsistencies (scrambled bits, for example). If a packet arrives with incorrect payload, it is dropped. The receiver sets the flag for each received packet as 1 and otherwise for packets not received.

Consider figure 4 which is an abstract representation of block diagram at the receiver side. The packet ID is first ANDed with equal number of bits where except for first 4 bits all the rest are zeros. This way, we can retrieve the

[1] Packet divided into equal sized smaller units called *flits*
[2] *M*icroprocessor without *I*nterlocked *P*ipeline *S*tages
[3] *D*igital *S*ignal *P*rocessor