# ABSTRACT

The rapid down-scaling of silicon technology has made massive on-chip transistor integration densities possible. Consequently, parallel-processing chips have emerged as the new digital design paradigm in ultra high-performance computing. Unfortunately, buses and dedicated interconnect wiring, which present severe restrictions in their ability to scale with system size, cannot meet the inter-tile communication requirements in current parallel on-chip systems such as Chip Multi-Processors (CMPs), imposing severe limitations in the performance efficiency and scalability of the entire parallel system. In addition, long wires give rise to numerous design restrictions which must be resolved, such as routing contention and challenges in synchronizing the various components on-chip. Network-on-Chip (NoC) architectures, a scalable and modular fabric, have been proposed as an alternative way to solve the above problems, replacing buses and "spaghetti" wiring, by using a packet-based on-chip network. Hence, communication among numerous Processing Elements (PEs) which reside on a single chip now takes the form of exchanging messages over the NoC, which function like off-chip interconnects found in supercomputers.

However, a single link failure in an NoC topology, which can occur due to various damaging physical effects such as electro-migration, can prevent the communication process. This effect may eventually render the entire CMP chip useless if the routed packets are stalled indefinitely in their routers. In this Thesis, a routing algorithm capable of handling large numbers of link failures that can occur either at manufacture-time (statically) or at run-time (dynamically), named Pythia[1], is presented. As opposed to marking the entire CMP as faulty, whenever some links fail, Pythia routes packets around the faulty link(s) until a new healthy link is available. This process can lead in-flight packets towards their destinations,

[1] Due to the fact that the proposed algorithm is aware of the status of the next-hop routers, concerning their faulty link distributions, it is named Pythia after the Greek priestess at the Temple of Apollo at Delphi.

maintaining network connectivity, and guaranteeing packet delivery, albeit at a slower rate which degrades gracefully.

Pythia is an adaptive and localized fault-tolerant routing algorithm, oblivious to the global state of the network. It uses distributed graph tables, which are held by each NoC router, in order to determine the best choice in routing a packet under the presence of faulty links. Every router has a graph table, which contains only the statuses of a its own links and of those nodes it leads to. Given this information, routing is made possible by manipulating either the coordinates of a packet's destination or destination distance, and the next available router's number of faulty links, for each link available from the current router to the next ones. Graph tables that hold all the statuses of all links residing at the next nodes found in the vicinity of the route, gives the foreknowledge of a possible deadlocked route a the next-hop router(s); hence deadlocked parts of a topology are avoided. In case a deadlocked situation cannot be avoided a-priori, as its existence is not yet known, a deadlock-resolution mechanism resolves such a detrimental scenario. Moreover, Pythia introduces a new type of header flit, which holds information that helps in livelock-avoidance as well.

Pythia was simulated under uniform random and transpose synthetic traffic patterns, with a range of virtual channel per port counts using wormhole flow-control, in order to determine its performance and behavior, utilizing two faulty link spatial placement scenarios: (1) random, and (2) hotspot faulty link distributions. When compared against ARIADNE, an existing state-of-the-art fault-tolerant routing algorithm, Pythia demonstrated up to 237.5% and 166.67% improvement in throughput with a random faulty link placement, while it showed up to 165.0% and 66.7% increase in throughput with a hotspot faulty link placement, under uniform random and transpose traffic pattern usages, respectively. In addition, the proposed routing algorithm was simulated under a lightly-loaded network to determine its basic routing delay under "no stress" conditions, while further experiments exhibit its superior throughput attainment just before network saturation.