STORM: A Simple Traffic-Optimized Router Microarchitecture for Networks-on-Chip

Shalimar Rasheed, Paul V. Gratz, Srinivas Shakkottai, Jiang Hu

Department of Electrical and Computer Engineering, Texas A&M University
{rshalimar,pggratz,sshakkot,hujiang}@ece.tamu.edu

Abstract—Networks-on-Chip (NoCs) offer a scalable means of on-chip communication for future many-core chips. This work explores NoC router microarchitectures which leverage traffic pattern biases and imbalances to reduce latency and improve throughput. It introduces STORM, a new, low-latency, fair, high-throughput NoC router design, customized for the traffic seen in a two-dimensional mesh network employing dimension-order routing. Compared to a baseline NoC router with equivalent buffer resources, STORM offers single cycle operation and reduced cycle time (17% less than the baseline on 45nm CMOS). This design yields a higher overall network saturation throughput (13% higher than the baseline) in an 8x8 2D mesh network for uniform random traffic. STORM also reduces packet latencies under realistic workloads by 36% on average.

I. INTRODUCTION

Networks-on-Chip (NoCs) employ routers at each node to direct traffic. A typical, baseline router pipeline consists of two stages [2]. The first stage involves lookahead routing, virtual channel allocation, and speculative switch allocation, to compute a flit’s output port, allocate a downstream virtual channel, and a crossbar output port, respectively. These three operations are performed in parallel.

Switch allocation in the baseline NoC router involves two-level, separable, round-robin arbitration. This scheme, while simple to implement, can result in matching inefficiencies that limit network throughput [2]. Other allocation schemes such as wavefront allocation [8] and its fairer variant [1], which we term wavefront\(^+\), aim to improve network throughput, though usually at the expense of higher implementation complexity.

Under realistic workloads, traffic patterns in typical NoC designs are known to be both biased and highly imbalanced from node-to-node within the network. In this work, we propose STORM - a Simple Traffic-Optimized Router Microarchitecture - to leverage these per-node imbalances and traffic biases to lower latency and improve throughput.

This work is targeted towards a 2D mesh topology using dimension-order routing, arguably among the most prevalent of NoC architectures due to its low complexity and modularity.

II. THE STORM ROUTER DESIGN

A primary feature of STORM is destination-biased VC partitioning to facilitate simple, yet efficient matching. The concept of VC partitioning has been explored previously - for instance, partitioning has been based on X-Y dimensions [5] [4]. These works, however, primarily focus on low-overhead, low-throughput networks, or employ allocation techniques which require greater complexity than the baseline router, impacting cycle time. STORM leverages traffic pattern biases to partition VCs based on output ports; VC partitions in STORM are non-uniform across network locations and customized for location-specific traffic patterns.

Let \( V \) be the number of VCs per input port. We assume that \( V \) is the same for all input ports in all routers across the network. For a router at a particular node, we identify the following parameters:

- \( i \) - input port index
- \( P_i \) - set of output ports that can be requested by input \( i \)
- \( p \) - output port index, \( p \in P_i \)
- \( N_p \) - number of nodes that can be reached via output \( p \)
- \( d_{ip} \) - number of VCs at \( i \) which request output port \( p \)

Our goal is to find \( d_{ip} \) for all \( i \) and \( p \), for the router at a specific node in the network. We assume that all possible final destination nodes for any flit at any input port are equally likely - a characteristic of a uniformly random traffic pattern. Thus, splitting \( V \) input VCs at each input port is estimated by

\[
d_{ip} = \left( \frac{N_p}{\sum_{p \in P_i} N_p} \right) \times V, \quad \forall p \in P_i, \quad \forall i
\]

subject to

\[
d_{ip} \geq 1, \quad \forall p \in P_i, \quad \forall i
\]

and

\[
\sum_{p \in P_i} d_{ip} = V, \quad \forall i
\]

Figure 1 shows a sample microarchitecture of our proposed STORM router, for 5 VCs per port, with biased VC partitioning. The set of VCs across all input ports that are assigned to the same output port form a path-set. These per-output path-sets result in simplified, efficient VC and switch allocation. VC allocation uses two-level round-robin arbitration, with the first level picking one VC out of each path-set, and the second
level picking one downstream VC for the winner from the first level. Switch allocation involves a single level of round-robin arbitration for each path-set. Since each path-set has a unique, independent arbiter, there is no contention between different path-sets for the same output port, and thus network throughput is enhanced.

### III. Evaluation

Network-level simulations and RTL synthesis are performed to evaluate the baseline router, the wavefront+ router and STORM designs. We examine three variants of STORM:

- **STORM-2**: a 2-stage pipeline with a dedicated switch traversal stage.
- **STORM-1**: a single stage router pipeline.
- **STORM-1S**: STORM-1 utilizing a single STORM router microarchitecture used across the entire network.

**STORM-1S** is a simpler implementation of STORM, in which we perform traffic-optimal VC splitting for a single central node in the network, and use this same microarchitecture at all the other nodes.

Simulations are carried out on an 8x8 2D mesh network with dimension-order (XY) routing, using the network simulator Ocin_Tsim [7]. We simulate the designs with 6 4-flit deep VCs per port. For uniform random traffic, after a warm-up phase of 10000 cycles, the runs last until 1 million 4-flit packets have been routed.

For RTL synthesis, we use an open-source, parameterized Verilog RTL model for the baseline router [1], which also includes the description of a wavefront+ switch allocator. This RTL was modified to evaluate the STORM design. We synthesize the baseline, wavefront+ and STORM designs using TSMC’s TCBN45GS 45nm CMOS library with an operating voltage of 0.9 V. For routers using 6 VCs per port, the minimum clock periods for the baseline, wavefront, STORM-2 and STORM-1 designs are 0.69, 0.80, 0.50 and 0.57 nanoseconds, respectively.

Figure 2 shows load-latency curves for the router designs for uniform-random traffic; in the case of STORM, it is assumed that the entire network is clocked at the speed of its slowest router. STORM-1 offers a 41% reduction in zero-load latency and 13% increase in saturation throughput, relative to the baseline.

We also analyze the variation in saturation throughput for the router designs at the cycle-level, as the number of VCs per port varies from 4 to 16 (Figure 3). To serve as performance bounds, a router using maximum bipartite matching for switch allocation, and an *unrestricted* router [6] are also simulated. We see that STORM-1 consistently outperforms the baseline and wavefront+ designs in terms of throughput for VC counts of 5 and greater. STORM-1S, where the router microarchitecture is held constant over the entire network, while not the most optimal design in terms of throughput, still better the performance of both the baseline and wavefront+ routers.

We execute PARSEC benchmark traces using the Netrace [3] library integrated with Ocin_Tsim to evaluate the average flit latency offered by the baseline, wavefront+ and proposed designs on these realistic workloads. The PARSEC traces were simulated with six 5-flit deep VCs per input port. Figure 4 shows the average flit latency in nanoseconds for PARSEC traces while employing the different router designs. STORM-1 provides significant improvement in terms of latency, being, on average, 36% faster than the baseline router.

### REFERENCES