Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 94 (2016) 410 – 417

Procedia Computer Science

The 3rd International Workshop on Design and Performance of Networks on Chip (DPNoC 2016)

An Enhanced Network-on-Chip Simulation for Cluster-Based Routing

Ahmed S. Hassan[a,*], Ahmed A. Morgan[b], M. Watheq El-Kharashi[a]

[a]*Department of Computer and Systems Engineering, Ain Shams University, Cairo 11517, Egypt*
[b]*Department of Computer Engineering, Cairo University, Gize 12613, Egypt*

Abstract

Network-on-Chip (NoC) research and development have increased noticeably over the past few years. As the NoC size increases, it was an urge to group processing elements (PEs) into clusters, based on common characteristics between these elements or the spatial locality in the floorplan. NoC clustering achieved better performance, scalability, and load balancing. On another hand, many NoC simulators, like NoCTweak, were developed to provide quick and easy tool for early evaluation of NoC-based designs. In this paper, we extend NoCTweak in order to allow it to simulate NoC clustering. The presented NoC clustering simulator supports the modular design technique, which in turn offers the flexibility in configuring cluster parameters. Examples of configurable parameters that are supported by our simulator are adding a cluster manager, adding latency factors of cluster links, and adapting the routing algorithm for both inter-cluster and intra-cluster traffic.

Peer-review under responsibility of the Conference Program Chairs
Keywords: NoC; Clustering; NoCTweak

1. Introduction

Many-core System-on-Chip (SoC), and especially NoC-based platforms, have increased dramatically. These platforms are loaded with lots of processing elements (PEs) and other peripherals. As the number of connected elements increases, their communication cost increases, in terms of hop count, and packets are more prone to stalls and contention. Depending on task mapping and routing protocols, the generated traffic could result in high communication density between some PEs and over certain links. PEs maybe close to one another or in different neighborhoods. The system designer has to explore different NoC topologies and routing techniques, along with different traffic patterns, to reach the optimal performance and energy dissipation [1], or trade-off between area and average delay [2]. This early design space exploration comes with high cost, in both time and resources. Therefore, it is a common practice and more efficient to do simulation to explore different design options.

There are two types of NoC simulators, systems-level and circuit-level. In system-level simulation, the NoC system design is evaluated as a whole. Parameters, like routing algorithm efficiency and topology, are evaluated in terms of

* Corresponding author. Tel.: +20-100-437-6169.
E-mail address: ahmedsayed.elaraby@gmail.com

Peer-review under responsibility of the Conference Program Chairs
doi:10.1016/j.procs.2016.08.063

communication latency and throughput[3–7]. While, in circuit-level simulation, technology parameters are taken into consideration, like critical path delays and temperature variation[8]. For early design space exploration, system-level simulators are more suitable.

Many literatures have investigated system-level NoC simulation for early design space exploration. Noxim[3] and NoCTweak[4] are notable examples. Most of the NoC simulators focus on evaluating metrics, like throughput and energy dissipation on the NoC platform level. They further allow customizing the traffic communication pattern and rate, the routing algorithm, and the topology. NoC clustering has spread out and different clustering techniques have been investigated to provide both optimal performance and energy dissipation[9?–12]. However, those NoC clustering techniques are not included in the mainstream NoC simulators. In our work, we addressed this problem by modifying one of these simulators, in order to support for NoC clustering simulation. We preferred modifying an existing NoC simulation tool as it was more convenient to start from a NoC simulator, which had been used before and provided reasonable results. The following NoC clustering simulation related issues have been investigated in our work: NoC clusters platforms, traffic routing, and cluster management.

In this paper, we present an extension to the NoC simulation tool, *NoCTweak*, to provide support for clusters simulation. The source code is adapted from *NoCTweak*, and different clustering-related features are added to the code.These features are: the ability to create a 2-D NoC grid of clusters of PEs, specifying how this grid is interconnect, adding latency factor on the inter-cluster links, specifying the routing algorithm for the inter-cluster traffic, and the possibility to use a cluster manager.

The rest of this paper is organized as follows. section II reviews related work. Section III provides background on NoC clustering. Section IV reviews the original *NoCTweak* simulator. Section V presents the new feature added in *NoCTweak* to support simulating NoC clusters. Section VI presents case studies. The paper is concluded in section VII.

2. Related Work

A handful of NoC simulators have been developed over the recent years. Noxim was developed by Palesi et al.[3], and was implemented in SystemC. Noxim provides a variety of options to simulate 2-D NoC and to evaluate the NoC in terms of throughput, delay and power consumption. Jiang et al. developed Booksim[5], which allows configuring topology, buffer size, routing algorithm and router micro-architecture. Agarwal et al. implemented GARNET[6], which enables the evaluation of components, like caches and memory controller, along with NoC topology, router microarchitecture, and routing algorithms. Tran et al. implemented *NoCTweak*[4], which allows a wide range of configurations to be applied on the NoC platform under simulation.

The main focus of the aforementioned NoC simulators was introducing ways to select and customize the NoC parameters, like the routing algorithms, the communication pattern, the topology, the router microarchitecture, and the traffic characteristics. However, they ignored the scalability of the NoC platform, which can be achieved by clustering multiple NoC systems. Consequently, the interaction between NoC clusters is not considered. In this paper, we fill this open research gap by providing a tool to simulate NoC clusters.

NoCTweak has many configuration options that are not found in other simulators. For example, NoCTweak allows the designer to completely control traffic patterns, number of flits per packet, pipeline stages, switch and virtual channel allocation policies, and inter-router link length. These large configuration options motivate us to select *NoCTweak*, out of all simulators, to extend in this paper.

3. NoC Clustering Background

With the steadily growing number of PEs in NoC platforms, it became a common practice to employ clustering techniques that group related and heavily-communicated PEs close to one another. Moreover, the employed clustering technique should reduce hop distance between communicating clusters as possible. NoC clustering provides means for load balancing, by task distribution and relocation, and resource borrowing among clusters. Some NoC clustering techniques require a PE to act as a manager for the cluster[12].

Clustering can be static, where the shape and size of the cluster does not change at runtime, or dynamic, where the cluster size may change during runtime. Dynamic clustering can be done by borrowing PEs from neighboring