Computers and Electrical Engineering 45 (2015) 374–385

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Area-efficient snoopy-aware NoC design for high-performance chip multiprocessor systems ☆

Antoni Roca [a,*], Carles Hernandez [b], Mario Lodde [c], José Flich [c]

[a] Department of Computer Science, Universitat Politècnica de Catalunya, Barcelona, Spain
[b] Barcelona Supercomputing Center, Barcelona, Spain
[c] Department of Computer Engineering, Universitat Politècnica de València, València, Spain

ARTICLE INFO

Article history:
Received 14 February 2014
Received in revised form 7 April 2015
Accepted 8 April 2015
Available online 27 May 2015

Keywords:
Chip multiprocessor
Network-on-chip
Network architecture
Coherence protocol

ABSTRACT

Manycore CMP systems are expected to grow to tens or even hundreds of cores. In this paper we show that the effective co-design of both, the network-on-chip and the coherence protocol, improves performance and power meanwhile total area resources remain bounded. We propose a snoopy-aware network-on-chip topology made of two mesh-of-tree topologies. Reducing the complexity of the coherence protocol – and hence its resources – and moving this complexity to the network, leads to a global decrease in power consumption meanwhile area is barely affected. Benefits of our proposal are due to the high-throughput and low delay of the network, but also due to the simplicity of the coherence protocol. The proposed network and protocol minimizes communication amongst cores when compared to traditional solutions based either on 2D-mesh topologies or in directory-based protocols.

1. Introduction

Manycore systems are expected to grow to tens or even hundreds of cores in the same chip. A Network-on-Chip (NoC) is implemented to connect all of them efficiently [1]. NoCs are to replace conventional bus-based systems where throughput and latency is compromised as the number of cores increases. NoCs have been adopted in two system design approaches: multi-processor system-on-chips (MPSoCs) and chip multiprocessors (CMPs). In MPSoCs, applications are usually known in advance and the chip is customized to the applications, including the NoC. In CMPs, applications are not known in advance and thus, the chip is built with little or no information about its future use. In this paper we focus on NoCs designed for CMP systems.

Since its conception in 2001, NoC research has focused mainly in adopting the best strategies usually found in high-performance interconnects, covering aspects like topology, routing, switching, and arbitration. The main challenge found in NoC research has been the suitability of known research and solutions to the highly-constrained new domain (inside the chip). Indeed, many of the proposals have focused on providing very power- and area-efficient solutions, thus minimizing the power consumption and the area footprint of the NoC. With these constraints as a reference, the 2D mesh topology has been adopted as the baseline for NoC design, meanwhile conceptually simpler but better topologies as crossbars are discarded as they show higher area and power overheads. Real examples of simple NoCs are the ones implemented in the Polaris chip prototype by Intel [2], the Single Chip Cloud Computer by Intel [3], and the products offered by Tilera [4].

☆ Reviews processed and recommended for publication to the Editor-in-Chief by Guest Editor Dr. Terrence Mak.
* Corresponding author.

http://dx.doi.org/10.1016/j.compeleceng.2015.04.020
0045-7906/

Orthogonal to the design of NoCs for CMP systems, the memory hierarchy, and its implementation, plays a key role in the final product. A shared variable programming approach is appealing from the point of view of the programmer, instead of the message passing programming approach. The inherent simplicity when programming, however, requires a coherence protocol implementation that ensures coherency and consistency along all the memory hierarchy levels. It is typical to find approaches where the processors in CMP systems have a first level (L1) of private caches and a bank of L2 caches on each tile forming a global shared but distributed L2 cache. The third level of the memory hierarchy is main memory. Fig. 1 shows the CMP configuration we focus in this paper.

The coherence protocol implemented in the system can significantly vary from an implementation point of view. For example, snoopy protocols rely on a shared medium connecting all the processors. Typically, snoopy protocols are very simple to design and test [5]. On the contrary, the mostly-assumed directory-based protocols require a directory structure to keep the coherence information, that makes them much more complex to design and, most importantly, to test and validate, since they rely on a point-to-point network with no global visibility. This leads to race conditions of the protocol when multiple cores access the same block at the same time. The number of states of the protocol increases to an extent that prevents its validation in an affordable amount of time. Additionally, directory structures require dedicated resources at cache memories increasing area and power consumption.

When the two components (NoC and coherence protocol) are put on the same perspective we can identify an interesting conflict. Discarded topology structures like buses or crossbars offer the opportunity to implement simple coherence protocols, like the snoop-based protocol. Other preferred topologies, like the 2D mesh, do not allow snoop-based protocols thus need for more complex protocols, e.g. directory-based protocols.[1]

What we pursue in this paper is the effective co-design of both, the NoC and the coherence protocol, in order to improve performance and power meanwhile area resources remain bounded. If we analyze a typical CMP system, L1 caches and L2 banks resources and power clearly overcome resources and power consumed by the network. Thus, reducing the complexity of the coherence protocol – and hence its resources – and moving this complexity to the network, will lead to a global decrease in power consumption meanwhile area is barely affected. In this paper, we pursue the following properties to the final designed system:

- A simple coherence protocol that can be easily tested and validated. In particular, a snoop-based protocol.
- A customized NoC scalable enough for a relative large amount of nodes, compatible with the snoop-based protocol. In particular, reaching 128 processors on the same chip.

The solution we propose is called sNoC, referring to a snoopy NoC. sNoC is built with two network components, each implemented as a mesh-of-trees (MoT) [6]. The first MoT is used to broadcast requests whereas the second MoT is used to send unicast data messages. sNoC increases network resources but helps reducing the directory structure, containing overall power consumption. sNoC includes also a customized coherence protocol for the network.

System level evaluation, in terms of performance, area overheads, and power consumption, shows the viability and the higher efficiency of sNoC when compared to the typical designs of 2D meshes with directory-based protocols. The benefits of our proposal are due to the better performance of the MoT network, and the ability of the snoopy protocol to take the best of these high performance networks. In this sense, we show how a directory-based protocol underutilizes the MoT making this solution not attractive even being feasible in terms of area.

Results show that for a 64-node network sNoC reduces execution time and power consumption with respect to the 2D mesh up to 20% and 35%, respectively. Benefits are because the snoopy protocol reduces communication between nodes, in terms of number of messages in addition to the sNoC high-throughput architecture. Introducing a high-throughput MoT with a conventional directory-based invalidation protocol reduces execution time up to 13% due to the high-throughput low-latency network, but power consumption is increased up to 15% due to network inefficiency. Additionally, introducing a simple snoopy protocol improves scalability, as the sNoC just increases area by 2%, meanwhile the same network without the snoopy protocol increases area up to 10%.

The rest of the paper is organized as follows. In Section 2, the related work is presented. In Section 3, the snoopy protocol is presented. In Section 4, we introduce sNoC architecture. In Section 5, we evaluate our design and show that it overcomes previous state-of-art solutions meanwhile scalability is guaranteed. Finally, the main conclusions are presented at the end of the paper.

2. Related work

Snoopy protocols were the preferred solution for ensuring cache coherence in the first multicore designs. For example, the Pentium 4 chip used a source-synchronous protocol to handle a bus amongst 4 cores [7]. Snoop-based protocols are very simple in theory. However, to be practical, an efficient broadcast medium is required to handle ordered transactions. As the number of processor and memories to interconnect in the chip has increased in the past years, many

[1] One exception is the case of using a broadcast-based protocol in a 2D mesh network. Each snoopy action is converted into a broadcast. In this case, the network is flooded with many messages and virtual channels are required to avoid protocol-level deadlock conditions.