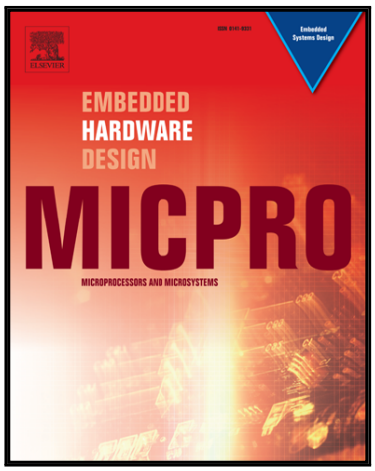

SDNoC: Software Defined Network on a Chip

Konstantin Berestizshevsky, Guy Even, Yaniv Fais,
Jonatan Ostrometzky

PII: S0141-9331(17)30153-9
DOI: 10.1016/j.micpro.2017.03.005
Reference: MICPRO 2521

To appear in: *Microprocessors and Microsystems*

Received date: 7 October 2015
Revised date: 6 March 2017
Accepted date: 8 March 2017

Please cite this article as: Konstantin Berestizshevsky, Guy Even, Yaniv Fais, Jonatan Ostrometzky, SDNoC: Software Defined Network on a Chip, *Microprocessors and Microsystems* (2017), doi: 10.1016/j.micpro.2017.03.005

SDNoC: Software Defined Network on a Chip

Konstantin Berestizshevsky, Guy Even, Yaniv Fais, Jonatan Ostrometzky

School of Electrical Engineering, Tel-Aviv University, Tel-Aviv 69978, Israel

Abstract

We present a novel network-on-chip (NoC) architecture, called SDNoC, that is based on a hybrid hardware/software approach. This approach is based on a few principles used in Software Defined Networks (SDNs). In particular, the control network and the data network are physically separated. In addition, SDNoC is controlled by a centralized Network Manager (NM) implemented in software that is executed on a dedicated core. These principles lead to many advantages. 1) Computation of paths is simple and the allocation of routes is efficient because the NM has a global view. Moreover, the NM is not limited to a small set of allowed routes and can easily deal with unexpected irregular traffic patterns. 2) The switches that forward phits in SDNoC are simple because they are configured by the NM, do not store phits, and do not have routing tables. 3) The overhead consumed by packet headers/trailers and control messages is greatly reduced. 4) There is no need for a complicated error-prone distributed protocol that avoids deadlock, starvation, etc. 5) Power consumption is proportional to the traffic in the NoC (as the NM processes requests, and the switches forward incoming phits). 6) Flexible design of the NM in software facilitates the addition of features such as security and support of priorities and deadlines for Quality-of-Service (QoS).

Index Terms

Network-on-Chip, Routing, Multi-Core Processing, Parallel Architectures.

I. Introduction

Networks-on-Chips (NoCs) have been proposed as a methodology for simplifying and improving the design of Multi-Processor Systems-on-Chips (MPSoC) [1], [2]. NoCs constitute good engineering practice since they scale well and, even more importantly, support modularity as NoCs help decouple communication from computation. NoCs need to support high performance and reliability (e.g., packets can not be lost).

In this paper we present a NoC scheme, called SDNoC, that can be viewed as a software defined network on a chip. Software defined networks (SDNs) are characterized, among other things, by a separation between the data plane and the management (control) plane, global control implemented in software, introduction of network abstractions, and programmability of network components [3]. Our SDNoC-scheme builds on this idea in order to design a low-cost high-performance architecture for aperiodic, nonsteady, low duty-cycle traffic between cores. Such irregular traffic appears in the execution of coarse-grain parallel applications in MPSoCs [4]. The SDNoC-scheme relies on a centralized network manager (NM) that can be implemented in software and executed on a dedicated core. The NM adaptively computes routes for requests based on a global view of the network. The NM allocates these routes by sending configuration instructions to the switches to employ virtual circuit switching. A few advantages are attained by using a centralized NM: (1) The network manager has a global view of the network which facilitates efficient allocation of data network resources as well as other performance measures (e.g., fairness, avoiding deadlocks, abiding priorities, etc.). (2) Computation and allocation of routes to packets is less sensitive to the distance from the source to the destination. (3) Reduced hardware networking overhead thanks to simple switches, no need for buffers, and no need for routers or routing tables. (4) Reduced communication overhead as packets and flits can be

Corresponding author: Guy Even (email: guy@eng.tau.ac.il)