Microprocessors and Microsystems 42 (2016) 215–226

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A BiNoC architecture—aware task allocation and communication scheduling scheme

Wen-Chung Tsai[a,*], Wei-De Chen[b], Ying-Cherng Lan[c], Yu-Hen Hu[d], Sao-Jie Chen[e]

[a] Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 413, Taiwan, ROC
[b] MediaTek Inc., Hsinchu 300, Taiwan, ROC
[c] Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, ROC
[d] Department of Electrical and Computer Engineering, University of Wisconsin–Madison, Madison, WI 53706-1691, USA
[e] Department of Electrical Engineering and Graduate Institute of Electronics Engineering, National Taiwan University, Taipei 106, Taiwan, ROC

ARTICLE INFO

Article history:
Received 31 May 2015
Revised 16 October 2015
Accepted 20 November 2015
Available online 31 December 2015

Keywords:
Bidirectional link
Communication scheduling
Many-core
Network-on-Chip
Task allocation

ABSTRACT

A novel real-time task allocation and scheduling scheme is proposed for a multi-core system incorporated in a Bidirectional Network-on-Chip (BiNoC) platform. Given a task graph, this scheme seeks to minimize the total execution time by allocating ready-to-execute tasks to as many available cores as possible subject to the real-time deadlines of each task. A refinement process is introduced to update the priority ranking of a task list so as to meet the timing constraints. In particular, the communication overhead is considered by incorporating the packet routing paths and delays into the overall optimization process. In doing so, the flexibility of bidirectional links of BiNoC is exploited to alleviate traffic congestion, such that more tasks could be executed concurrently at different cores and overall execution time be reduced. To validate the effectiveness of this proposed scheme, extensive simulations have been performed. The results clearly demonstrate the superior performance of this proposed scheme compared to existing approaches that did not exploit the flexibility of BiNoC.

1. Introduction

In a Many-Core Network-on-Chips (MC-NoC) [1–3], chip real estate is partitioned into a two-dimensional array of rectangular processor cores. Between the cores, a mesh-configured on-chip network (i.e., Network-on-Chip, NoC) [4,5] provides the underlying communication infrastructure. Computation tasks are allocated to individual cores and data packets are routed among different cores via the NoC fabric. Communication between cores at a distance would incur multi-hop routing and may be delayed due to traffic congestion. Hence, while it is imperative to allocate as many cores as possible to boost the parallelism, the communication overhead due to the constrained NoC infrastructure must be taken into account to achieve overall performance optimization.

Existing NoC architectures that use unidirectional links between on-chip routers offers rather limited communication capacity for applications where long streams of data must be transmitted from one processor core to the other in one direction only. The limited bandwidth often causes prolonged packet transmission delay, compromising the overall performance of multi-core computation. Recently, a novel Bidirectional Network-on-Chip (BiNoC) architecture [6–10] has been proposed. Since the transmission direction of a link can be reversed in a BiNoC, it has been shown that the communication capacity can be significantly improved. None the less, these earlier works focused on how to control a BiNoC fabric for a given trace of inter-processor communication requirements. For most real-world applications, the communication patterns over the NoC fabric depend very much on how specific tasks are allocated to individual processor cores, and how the computation of individual tasks are scheduled.

In this work, we focus our efforts on investigating the intertwined relations between task allocation and communication scheduling and the underlying NoC control on a BiNoC platform. We have made the following tangible contributions:

(1) Our integrated algorithm performs both task allocation and communication scheduling. Not only individual task is allocated to specific processor core, but its execution time also takes into account the routing path and data transmission delay over the underlying BiNoC network.
(2) A flexible communication delay model over a BiNoC infrastructure is proposed, where the inter-core communication

* Corresponding author. Tel.: +886 4 23323000x7843; fax: +886 4 23305539.
E-mail addresses: azongtsai@cyut.edu.tw, azongtsai@gmail.com (W.-C. Tsai), wedochen@gmail.com (W.-D. Chen), f94943068@ntu.edu.tw (Y.-C. Lan), hu@engr.wisc.edu (Y.-H. Hu), csj@ntu.edu.tw (S.-J. Chen).

http://dx.doi.org/10.1016/j.micpro.2015.11.014
0141-9331/

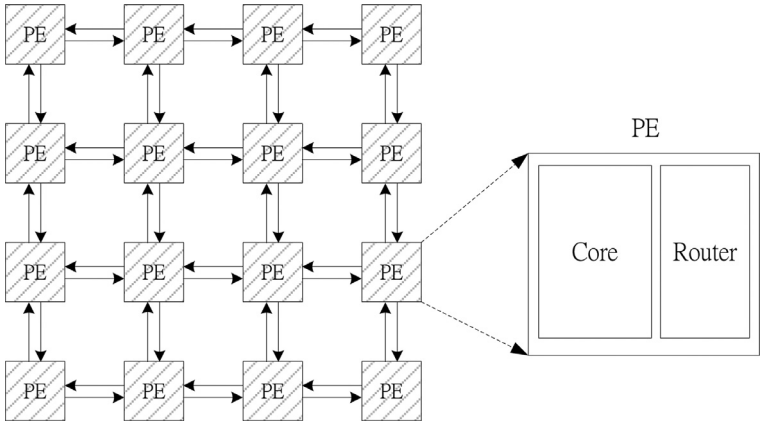

Fig. 1. Conventional NoC architecture.

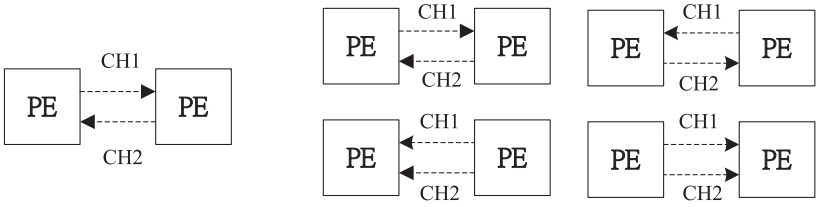

Fig. 2. Link directions in (a) a conventional NoC and (b) a BiNoC.

delay is explicitly modelled based on specific routing path and routing direction assignment.

(3) Joint optimization between task allocation and communication scheduling is realized in a refinement process.

(4) Application of hardware (BiNoC routing and control) and software (task allocation, communication scheduling) co-design facilitates a full exploitation of the flexibility afforded by the BiNoC to enhance multi-core task allocation and communication scheduling.

The remainder of this paper is organized as follows. We first introduce the preliminary aspects of the task allocation and communication scheduling problem for BiNoC architecture in Section 2. In Section 3, we present our proposed design methodology in detail. Section 4 reports extensive experimental results using Task Graphs For Free (TGFF) standard benchmark. Finally, we conclude the paper in Section 5.

2. Background

In this section, we will first introduce an intelligent NoC architecture which is called “Bidirectional Network-on-Chip” (BiNoC). Then previous works related to task allocation and communication scheduling will be reviewed. Lastly, some application specified problems will be defined and to be solved.

2.1. Many-core conventional and bidirectional NoC architectures

A typical Network-on-Chip fabric is an $m \times n$ mesh of tiles as shown in Fig. 1. Each core of a Processing Element (PE) can be a general-purpose processor, digital signal processor, data cache, or other type of hardware controller. A PE uses its router to link to the router in a neighbouring PE by two unidirectional channels with opposite directions.

In contrast to the conventional NoC architecture, where each pair of neighbouring PEs uses two unidirectional links in opposite direction to propagate data on the network as shown in Fig. 2(a), BiNoC architecture allows data to be transmitted in either direction of a bidirectional link as shown in Fig. 2(b).

2.2. Task allocation problem on conventional NoC architecture

An application can be divided into many smaller processes which are called “tasks”. A task is the basic execution unit in scheduling, thus one task can only be executed on one processor core. At a multi-processor system, each task is executed on a different processor core. This task allocation problem is known to be NP-Hard [11] and thus previous works solved this problem by different heuristic algorithms [12–16], In which, the Heterogeneous Earliest Finish Time (HEFT) [15] is the one of the best list-based scheduling algorithms in terms of robustness and schedule length [17]. Besides, Schmitz et al. [18,19] presented an iterative synthesis approach using Genetic Algorithms (GA). Kianzad et al. [20] improved the previous work by combining assignment, scheduling, and power management in a single algorithm. Chang et al. [21] proposed an Ant Colony Optimization (ACO) algorithm instead. Recently, authors of [22] provided a Predict Earliest Finish Time (PEFT) algorithm and demonstrated that it outperformed the classic list scheduling algorithm HEFT. The improvement is by introducing a look-ahead feature generated from the Optimistic Cost Table (OCT). Besides, task scheduling problem can be solved by using a basic, commonly-used Earliest Deadline First (EDF) algorithm. For example, Lee et al. improved schedulability using the prioritization polices in EDF [23]. Tan et al. provided an FPGA-based scheduler that supports EDF algorithm [24]. As mentioned, although studies have applied different heuristic methods in making profit in task scheduling. However, there seems to be no established theory to tackle the computation and communication performance issues especially for the BiNoC architecture.

2.3. Task allocation problem on BiNoC architecture

As Fig. 3(a) shows, an application task graph is typically described as a set of concurrent tasks that have been assigned and