



Dynamic task mapping for Network-on-Chip based systems



Tahir Maqsood*, Sabeen Ali, Saif U.R. Malik, Sajjad A. Madani

Department of Computer Science, COMSATS Institute of Information Technology, Pakistan

ARTICLE INFO

Article history:

Received 5 February 2015

Received in revised form 29 May 2015

Accepted 7 June 2015

Available online 15 June 2015

Keywords:

Dynamic task mapping

Network-on-Chip (NoC)

Multiprocessor System-on-Chip (MPSoC)

ABSTRACT

Efficiency of Network-on-Chip (NoC) based multi-processor systems largely depends on optimal placement of tasks onto processing elements (PEs). Although number of task mapping heuristics have been proposed in literature, selecting best technique for a given environment remains a challenging problem. Keeping in view the fact that comparisons in original study of each heuristic may have been conducted using different assumptions, environment, and models. In this study, we have conducted a detailed quantitative analysis of selected dynamic task mapping heuristics under same set of assumptions, similar environment, and system models. Comparisons are conducted with varying network load, number of tasks, and network size for constantly running applications. Moreover, we propose an extension to communication-aware packing based nearest neighbor (CPNN) algorithm that attempts to reduce communication overhead among the interdependent tasks. Furthermore, we have conducted formal verification and modeling of proposed technique using high level Petri nets. The experimental results indicate that proposed mapping algorithm reduces communication cost, average hop count, and end-to-end latency as compared to CPNN especially for large mesh NoCs. Moreover, proposed scheme achieves up to 6% energy savings for smaller mesh NoCs. Further, results of formal modeling indicate that proposed model is workable and operates according to specifications.

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

Advancements in silicon technology and developments in the nanotechnology have enabled the integration of a complete system onto a single chip, referred as System-on-Chip (SoC). The SoC helps in integrating maximum amount of technology in the smallest possible space by placing several processing elements (PEs) on a single chip leading to the development of Multiprocessor System-on-Chip (MPSoC) [1]. The PEs of MPSoC can be homogenous or heterogeneous each performing varying tasks and possibly operates at different frequencies. MPSoCs are being applied in complex embedded applications because the growing and complex computational requirements of future embedded system applications cannot be fulfilled by a single general purpose processor. Therefore, MPSoCs have been proposed to meet the growing performance requirements of future embedded systems [2]. Future MPSoCs are anticipated to be composed of dozens or even hundreds of processing cores. Integrating large number of cores on a

single chip requires a robust and scalable interconnection architecture.

Network-on-Chip (NoC) has been proposed as a modular and scalable communication interconnect to support communication among the PEs of MPSoCs [3]. The research issues and challenges for NoC based systems can be classified into four major categories that are [4]:

- **Communication infrastructure:** Research based on the design and development of efficient communication infrastructure that includes: selection of suitable network topology, router architecture, optimization of buffers, link design, clocking, floor planning, and layout design.
- **Communication paradigm:** Efforts to find best possible solutions for routing policies, switching techniques, congestion control, power and thermal management, fault tolerance, and reliability.
- **Evaluation framework:** Development of tools and techniques to get better understanding of achievable throughput, latency, and bandwidth of the network.
- **Application mapping:** Efforts to find suitable placement of set of tasks of an application onto the PEs of NoC in such a way that optimizes the one or more parameters of interest.

* Corresponding author. Tel.: +92 333 3122065.

E-mail addresses: tmaqsood@ciit.net.pk (T. Maqsood), sabeenali@ciit.net.pk (S. Ali), saif_ur_rehman@comsats.edu.pk (S.U.R. Malik), madani@comsats.edu.pk (S.A. Madani).

First three categories are extensively surveyed but the fourth category, i.e., application mapping is not much dealt with [4]. In application mapping, tasks of the application are mapped onto the suitable PEs depending upon various optimization criteria, such as energy consumption, network overhead, congestion, and latency. Application or task mapping is one of the critical factors that affect the performance and efficiency of NoC based MPSoCs. There are few surveys and comparative studies conducted to evaluate the task mapping algorithms proposed for NoCs [4–6]. However, these studies either restricted the analysis to smaller mesh sizes or have conducted qualitative comparison of presented algorithms. For instance, in [6] authors have conducted the comparison of static and dynamic task mapping heuristics for 5×5 and 9×9 NoCs. Similarly, the authors in [5] presented the analysis of selected dynamic task mapping heuristics over 8×8 mesh NoCs. The authors of [4] presented a qualitative survey of static and dynamic task mapping techniques for MPSoCs.

In this study, a detailed quantitative analysis of some of the selected dynamic task mapping heuristics is provided considering a wide range of NoC mesh sizes with varying task loads. The experimental results of comparative analysis revealed that communication-aware packing based nearest neighbor (CPNN) algorithm achieves better performance as compared to the other algorithms in terms of reduced energy consumption. However, CPNN exhibits higher communication cost, especially for large mesh NoCs. Moreover, mapping produced by CPNN algorithm results in unbalanced load distribution among the PEs leading to increased energy consumption. Therefore, in this work, we also propose an extension of CPNN algorithm that attempts to reduce communication overhead by migrating communicating tasks to the same PE for constantly running applications. Major contributions of this article are summarized as follows:

- A detailed quantitative analysis of the selected dynamic task mapping heuristics is provided under same environment, using same assumptions, and system models.
- An extension to CPNN heuristic is proposed. The proposed heuristic aims to reduce the communication cost and energy consumption by migrating communicating tasks that are already mapped using CPNN from lightly loaded PEs to the other PEs that can accommodate those tasks.
- Formal verification and modeling of the proposed technique is provided using high level Petri nets, satisfiability modulo theories, and Z3 solver.

The remaining of the paper is organized as follows: Background and some preliminary concepts are presented in Section 2. Prior research work related with our work is reviewed in Section 3. Section 4 provides the details about models used for conducting this study. Section 5 presents the working of algorithms selected for analysis along with the proposed algorithm. Formal verification and modeling is discussed in Section 6. In Section 7, experimental setup and results are presented. Finally, Section 8 concludes the paper with some directions for future work.

2. Preliminaries

The future MPSoCs can consist of hundreds or even thousands of processing elements. Therefore, there is need for scalable and robust communication infrastructure to interconnect the large number of processing elements. Traditional SoC technology uses a shared bus interconnection to connect limited number of PEs. However, as the number of PEs increases, the shared bus architecture is unable to scale with the increase in number of PEs [7,8]. In order to overcome the limitations of shared bus architecture, NoC has been proposed as a modular and scalable alternative [9,10].

The processing elements within the NoC can be homogeneous or heterogeneous. The processors belonging to same family make the MPSoC platform homogeneous and those belonging from different families make it heterogeneous [11,12]. In 2007 and 2009, homogeneous MPSoCs were proposed by Intel [13] and Tilera [14] with 80 and 100 processing elements connected through NoC, respectively. Heterogeneous MPSoCs were proposed by IBM, Sony, and Toshiba, that consisted of one manager processor with 8 floating point units [15]. The resources that can be shared by the various PEs in NoC include: memory, input/output, and links [11,12].

Network-on-Chip may comprise of computing resources, programmable logic blocks, distributed storage resources, programmable I/O, and a switching fabric that connects all the resources. The computing resources can be processor cores or field programmable logic blocks. The switching fabric acts as a communication backbone for connecting computing resources of NoC [16]. Major components of a NoC based MPSoC are depicted in Fig. 1 [18,19]:

- *Processing elements (PEs)*: Can be a general-purpose processor, digital signal processor (DSP), or embedded memory.
- *Network interface*: Interface that connects processing elements (PE) to NoC through routers.
- *Routers*: Nodes for routing data and control flits based on specified routing strategy.
- *Router-to-router links*: Include logical and physical channels that are used for transferring flits between routers.

3. Related work

To utilize the NoC resources effectively, efficient task mapping is required to place tasks onto the suitable processing elements. Task mapping in NoC refers to the process of finding suitable placement for the tasks of an application to the processing elements while considering certain optimization criteria, such as reduction in energy consumption, total execution time, packet latency, congestion, and communication overhead [1]. Task mapping decisions directly reflect on the overall system performance in terms of the aforementioned parameters. Mapping applications onto the NoC architecture is NP-hard [18,20,21]. Therefore, various heuristics based solutions have been proposed for task mapping problem. A significant amount of work has been done and is still under process to find efficient task mapping for NoC. In the subsequent paragraphs we briefly mention some of the selected task mapping techniques proposed for NoC based MPSoCs. Moreover, a brief comparison of the discussed techniques is presented in Table 1.

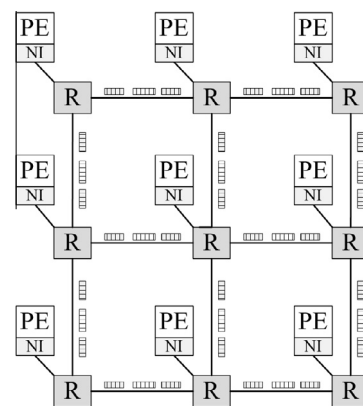


Fig. 1. NoC components [17].

Download English Version:

<https://daneshyari.com/en/article/457699>

Download Persian Version:

<https://daneshyari.com/article/457699>

[Daneshyari.com](https://daneshyari.com)