Microprocessors and Microsystems 38 (2014) 233–245

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

# System-level performance analysis of multiprocessor system-on-chips by combining analytical model and execution time variation

Sungchan Kim [a,*], Soonhoi Ha [b]

[a] Division of Computer Science and Engineering, Chonbuk National University, Jeonju, Jeonbuk 561-756, Republic of Korea
[b] School of Computer Science and Engineering, Seoul National University, Seoul 151-744, Republic of Korea

ARTICLE INFO

*Article history:*
Available online 18 February 2014

*Keywords:*
Multiprocessor
System-on-chip
Communication architecture
Performance analysis
Queuing theory

ABSTRACT

As the impact of the communication architecture on performance grows in a Multiprocessor System-on-Chip (MPSoC) design, the need for performance analysis in the early stage in order to consider various communication architectures is also increasing. While a simulation is commonly performed for performance evaluation of an MPSoC, it often suffers from a lengthy run time as well as poor performance coverage due to limited input stimuli or their ad hoc applications. In this paper, we propose a novel system-level performance analysis method to estimate the performance distribution of an MPSoC. Our approach consists of two techniques: (1) analytical model of on-chip crossbar-based communication architectures and (2) enumeration of task-level execution time variations for a target application. The execution time variation of tasks is efficiently captured by a memory access workload model. Thus, the proposed approach leads to better performance coverage for an MPSoC application in a reasonable computation time than the simulation-based approach. The experimental results validate the accuracy, efficiency, and practical usage of the proposed approach.



## 1. Introduction

With the ever-increasing complexity of embedded applications, the system complexity is also growing with an increasing number of processing elements in a single chip. Such a chip, which uses multiple processors, is called a Multiprocessor System-on-Chip (MPSoC). Hence, more communication requirements are imposed on on-chip networks, which, in turn, significantly affects the performance of an MPSoC. To cope with such a complexity, designers need to perform a system-level performance analysis in the early stages to explore various design choices before system realization.

Even though simulation-based approaches are popular for estimating on-chip communication performances [26,27], they often suffer from lengthy run times as well as poor performance coverages owing to limited input stimuli or their ad hoc applications. Therefore, recent research on the performance analysis of embedded systems has focused on analytic or semi-formal methods to estimate the worst-case execution time (WCET) [28,32,19,18,33]. In particular, in MPSoC designs, the arbitration policy is a key parameter affecting the performance over various interconnection networks such as a bus, crossbar, or Network-on-Chip (NoC). Fixed priority arbitration is still a popular choice even though it may cause a starvation problem. Related works that have addressed the performance analysis problem on bounded arbitration protocols [30] or unbounded ones [28,32] usually focus on the WCET delay for transferring a network packet or a task-level event stream. Thus, the use of such approaches for a bus transaction-level analysis may result in a severe overestimation such that every bus access undergoes the worst-case arbitration delay. As a result, they are unsuitable for soft real-time system design where the average performance is a primary concern [6,13].

This paper proposes a system-level method to estimate the average performance distribution of an MPSoC with a bus matrix (also known as crossbar switch)-based communication architecture deploying a fixed priority arbitration. A bus matrix provides high throughput while preserving the simplicity of a shared bus abstraction. It is now widely accepted as the industrial de facto standard for on-chip communication in chip multiprocessors [21] as well as in MPSoCs [2,29,31]. A bus matrix has multiple master and slave ports that are connected via multiple internal buses. Any master port can be connected to any slave port in a bus matrix. This is usually referred to as a fully connected matrix [25]. It allows multiple accesses to different slaves in order for them to be in parallel; this results in a higher performance than conventional shared bus architectures. However, there is a scalability issue with regard to the number of master and slave ports in a single bus matrix [35]. One way to resolve the scalability problem is to partially connect

* Corresponding author. Tel.: +82 63 270 2411.
*E-mail addresses:* sungchan.kim@chonbuk.ac.kr (S. Kim), sha@snu.ac.kr (S. Ha).

http://dx.doi.org/10.1016/j.micpro.2014.02.003
0141-9331/

the master and slave ports, thus avoiding resource wastage due to unused bus connections.

Packet-switched Network-on-Chip (NoC) architectures are becoming popular as a backbone communication infrastructure that connects processing subsystems and other hardware devices. They combine locally synchronous subsystems with a packet-switched network to build a globally asynchronous system. They may have various topologies such as mesh, ring, and tree, and the regularity of NoC can improve design productivity. The NoC architecture typically allows for a higher clock rate and provides higher communication bandwidth than the bus matrix architecture. However, it has the following disadvantages compared to the bus matrix architecture. First, it incurs the non-negligible overhead of additional buffers and control logics for converting memory transactions to packets because many IPs are still provided with on-chip bus standards [2,29,31]. Second, packetization/depacketization incurs additional delay. Finally, the latency for delivering packets to a destination over NoC is often more unpredictable than the bus matrix-based architectures because of the complicated transaction protocol of NoC. Hence, the bus matrix architecture is a viable on-chip interconnection scheme for systems with processor of the order of several tens. For large-scale systems with hundreds of processors, typical on-chip interconnection implies a combination of bus matrix for the subsystem and NoC for backbone communication [3,4].

Given a target application and the underlying communication architecture, the proposed technique finds a wider range of performance distribution by corner-case analysis than by a simulation-based approach in significantly less time. The proposed technique consists of two key parts: first, building an analytical model of the target system's dynamic behavior, and second, systematically exploring, based on the model, the wider performance variations as far as possible within the affordable[1] computation time. The proposed analytical model of a bus matrix architecture is based on the queuing theory and statistics of the memory access behavior of tasks. Then, it is integrated into a unified framework to enumerate task-level execution time variation of a target application, and thereby, to estimate the performance distribution with the underlying communication architecture. Because the execution time variation of tasks constitutes the huge search space of execution paths, we propose a scheme to reduce the search space by selecting a representative set of the execution times for a task. In this scheme, the execution time of a task is defined by the memory access count and access request interval.

Experimental results validate the proposed technique. First, our analytical model robustly and accurately predicts the execution time of a target application on various bus matrix architectures. In comparison with the simulation-based approach, the time taken for our analysis is an order of magnitude shorter. Furthermore, the estimated performance on average is 95% accurate. Second, the proposed technique defines a wider performance range than the simulation-based approach along with a faster analysis time by significant orders of magnitude. Experiments over various bus matrix architectures show that the performance ranges obtained by the simulation-based approach lie within the range obtained by the proposed technique. The performance range gap between the two approaches is about 21% on average in terms of the worst/best execution time, which is an acceptable overestimation for practical use. However, it is worth noting that the proposed technique does not guarantee the worst-case performance.

In the next section, we review related work and state our contributions. The overview of the proposed analysis framework is presented in Section 3. Section 4 explains the analytical model of on-chip communication architectures using the queuing theory. Then, in Section 5, the system-level performance analysis technique based on the analytical model is introduced. Experiment results on the accuracy and efficiency of the proposed approach are provided in Section 6. Finally, Section 7 presents the conclusions and addresses future work.

[1] Although no quantitative metric of *affordable time* is given, we regard a technique as affordable if it can be used in practice. For instance, in this study, if an analysis can be done in several hours, we consider it affordable.

## 2. Related work

If the communication architecture of an MPSoC is customized to provide a guaranteed latency [9], it is possible to estimate the WCET of an application at the system level. Otherwise, the system performance is usually based on the average performance of the underlying communication architecture as in this study.

Regarding the former case, a considerable number of studies have been conducted on WCET analysis at the system level. In the Modular Performance Analysis (MPA) approach [32], the stream of task-level requests to processing components are modeled as arrival curves. Processing components have their own computational capacities, which are represented as service curves. The stream of requests to a processing component has a bounded delay until the completion of service, which is calculated by the min-plus and max-plus calculi. The various arbitration policies of shared resources are considered at the task level in this approach. The SymTA/S [28] approach proposes a system-level performance analysis method based on the classical schedulability analysis for real-time systems. When the task execution in a component is analyzed by the schedulability analysis, the inter-component interaction is modeled by the abstract event model that is represented as a tuple $(p, j, d)$ indicating the period, jitter, and minimum distance of events. These approaches share a common disadvantage: overestimation caused by high-level abstraction between components.

The approach proposed in [18] uses a discrete event simulation for the schedulability analysis of distributed embedded systems with periodically invoked communicating tasks. The execution scenario of the system is represented by an execution path tree; whenever the scheduling of a new task or the completion of a task having variable execution time occurs, a new node representing a system state is defined and a branch is added to the tree. This exhaustive approach shows better performance coverage than a simulation. Our approach is similar to this approach in that we enumerate the execution paths of an application. However, the proposed technique is different from this work in various ways. First, we consider the contention of the communication architecture for more accurate estimation. Second, we consider the variability of task execution time selectively to reduce the time complexity of the design space exploration.

In order to model the average performance of on-chip networks, several formal approaches have been proposed at various levels of abstractions. Model checking-based approaches have been proposed for AMBA bus verification at the protocol level [19,5]. These approaches, however, require prohibitively huge computation times. Moreover, a nontrivial modeling effort for various communication architecture topologies is required. At the transaction level, on-chip communication architecture models using the queuing theory have been proposed for a hierarchical shared bus [15] and an NoC [10,24,23,7,8,1,14], respectively.

Several studies have proposed analysis techniques for the crossbar-based interconnections of multi-computer systems [20,22]. They, however, usually ignored the effects of an arbitration policy or focused on bandwidth rather than latency. Therefore, these approaches are unsuitable for the analysis of MPSoC architectures. The work on modeling shared bus or cascaded bus matrices of MPSoCs [15,11] is based on the $M/M/1$ queuing model or a similar