

Special issue on on-chip parallel and network-based systems

Masoud Daneshtalab¹ · Nader Bagherzadeh² ·
Hamid Sarbazi-Azad^{3,4}

Published online: 14 May 2015
© Springer-Verlag Wien 2015

1 Introduction

In order to achieve functionality with low energy speed product, on-chip parallel and network-based system design requires larger device, multi block functions, and energy evaluation schemes. Such systems, which are emerging as the architecture of choice for future high performance processors, require efficient interconnect which are necessary to satisfy the data supply needs of all cores.

This special issue attempts to cover new ideas in the design and analysis of on-chip communication technology, architecture, design methods and applications. In what follows, we will give a brief overview of the accepted papers of this Special Issue based on their topics.

2 Selected papers

Parallel task execution on multiple cores is increasingly the norm in high-performance embedded systems. Many multicore real-time operating systems (RTOSes) support a fixed task priority policy, and many applications that execute on such platforms require high performance while adhering to the fixed task priority policy. In the paper, titled

✉ Masoud Daneshtalab
masdan@kth.se

¹ Royal Institute of Technology, Stockholm, Sweden

² University of California-Irvine, Irvine, USA

³ Sharif University of Technology, Tehran, Iran

⁴ School of Computer Science, Institute for Research in Fundamental Sciences (IPM),
Lavāsani Street, Tehran, Iran

“Decentralized Task Scheduling for a Fixed Priority Multicore Embedded RTOS”, a task scheduling problem for a fixed priority multicore RTOS on an embedded system is presented. Authors has proposed a general multicore scheduler which can be applied differently according to the types of tasks to be scheduled. Based on the behavior deemed to be desirable for a fixed-priority multicore RTOS scheduler, 32 possible combinations of heuristic policies were proposed and evaluated for the L1 scheduler.

In the paper, titled “Design and implementation of counting networks”, Authors described Hamming weight counters built on counting networks that incorporate two distinctive and important features. The counting networks are composed of simple logic (core) elements with incrementally reducing numbers of elements from the inputs to the outputs. This feature provides the same performance as the best known sorting networks with radically reduced complexity. Compared to a competitive design based on parallel counters, the propagation delays of signals passing through data independent segments within the circuit are shortened, which allows faster pipelined implementations. Several types of counting networks are elaborated, namely pure combinational, partially sequential with reusable fragments, and pipelined. The correctness of the proposed concept and scalability of the networks are proven. Formal expressions to estimate the complexity and throughput of the network are given. Finally, the results of extensive experiments, evaluations and comparisons are reported that demonstrate that the solutions proposed offer better characteristics than the best known alternatives.

As the optimal resource utilization is a crucial factor for efficient Network-on-Chip (NoC) architectural designs, the Authors of the paper titled, “Using Constraint Programming for the Design of Network-on-Chip Architectures”, explored the practicality of the Constraint Programming (CP) models for NoC architecture designs. The complexity of the CP models is compared with the earlier Mixed Integer Programming (MIP) models. Practical CP-based mapping and scheduling models are developed and results are reported on the benchmark datasets. Results indicate that mapping and scheduling problems can be solved at near optimality even under relatively shorter run-time limits as compared to those required by the MIP models.

A logical extension to the homogeneous platform is to study the case of having heterogeneous PEs. The challenge of dynamic voltage frequency island (VFI) is an open problem in terms of providing robust models. Therefore, an extension of the CP model should be in the direction of studying VFI on heterogeneous platforms and adaptive routing. Authors believe that the work can be extended to real applications on various processing platforms beyond 8×8 mesh architecture in the future.

Combining three-dimensional integrated circuits (3D ICs) and NoC is expected to yield better performance and higher scalability. The paper titled, “Energy Reduction in 3D NoCs through Communication Optimization”, explored the possibility of combining these two techniques in a heterogeneity aware fashion. Specifically, on a heterogeneous 3D NoC architecture, they explored how different types of processors can be optimally placed to minimize data access costs. Moreover, they selected the optimal set of links with optimal voltage levels. The experimental results indicate significant savings in energy consumption across a wide range of values of our major simulation parameters.

This paper titled “System on Chip Failure Rate Assessment Using the Executable Model of a System”, proposed an analytical soft-error rate assessment technique for

the cores inside a System-on-Chip (SoC). The proposed method takes an executable UML-RT model of a SoC and assigns a SER to each module inside that model. The method considered simultaneously AVF and TVF as well as characteristics of an environment in which a SoC is supposed to be implemented, thereby involving raw soft error rate of storage cells to assign SER to different cores inside an SOC. Experimental results show that the proposed method is 17 % more accurate than the previous error estimations technique.

Faults at either the link or router level may result in the failure of NoC-based systems. Fault-tolerant routing algorithms attempt to tolerate faults by rerouting packets around the faulty region. However, it would be at the cost of significant performance loss. The proposed algorithm in the paper titled “A Light-weight Fault-Tolerant Routing Algorithm Tolerating Faulty Links and Routers”, is able to tolerate both faulty routers and links with the negligible impact on the performance. In fact, the proposed algorithm avoids taking unnecessary longer paths to reroute packets and the shortest paths are taken as long as a path exists. On the other hand, fault-tolerant routing algorithms might be based on deterministic routing in which all packets use a single path between each pair of source and destination routers. Using deterministic routing, packets reach destinations in an in-order manner so that no reordering buffer is needed at destinations.

Transactional Memory (TM) is emerging as a promising paradigm to simplify parallel programming for Chip Multiprocessors (CMPs). Most of Software Transactional Memory (STM) systems exploit a lock table to synchronize transactional accesses to the shared memory locations. Memory addresses map to entries of the lock table through a hash function to detect conflicts in the event of simultaneous accesses to the shared memory locations. In the current implementation of the lock table, if two distinct addresses map to the same entry of the table, they are treated as a conflict even though there is no true conflict between the two addresses. This is called false conflict. In the event of a false conflict, transactions are aborted conservatively which reduces concurrency level in programs. In the last paper titled “TurboLock: Increasing Associativity of Lock Table in Transactional Memory”, Authors studied false conflicts in STMs and propose TurboLock to reduce frequency of false conflicts. Inspired by set associative caches, TurboLock increases associativity of the lock table to reduce likelihood of aliasing-induced conflicts. While TurboLock is effective in reducing the false conflicts, it may degrade performance due to overhead of the associative lock table in software.

Acknowledgments We would like to thank all authors (of accepted papers and the papers that were not selected for inclusion in this Special Issue) who submitted papers to this Special Issue and extend our gratitude towards our reviewers for their time and diligence during the review process and for coming up with high quality reviews. Finally, we would like to thank Professor Schahram Dustdar, Editor-in-Chief of the Springer Journal of Computing for hosting this Special Issue. We would also like to thank Christine Kamper, Priya Balamurugan, and Divya Rajakumar for the great support during the course of this project.