Architectural and Physical Design Optimizations for Efficient Intra-tile Communication

A. Papanikolaou[1], F. Starzer[2], M. Miranda[1], K. De Bosschere[3], F. Catthoor[1,4]
[1]IMEC v.z.w., Kapeldreef 75, 3001 Leuven, Belgium
[2]FH Hagenberg, Hauptstr. 117, 4232 Hagenberg, Austria
[3]Universiteit Gent, Sint-Pietersnieuwstr. 25, 9000 Gent, Belgium
[4]Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, 3001 Heverlee, Belgium

Abstract— **Intra-tile communication requirements for future SoC platforms are becoming ever more demanding for new processor and memory architectures. Increased bandwidth, low latency and low energy consumption are required, which the current communication architecture solutions cannot provide. In this paper we propose the use of software-controlled, light-weight segmented buses to implement the communication between the processing elements and their working memories. We show that significant energy and delay/latency gains can be expected from the use of this communication architecture.**

I. Introduction

Energy minimization is fast becoming the major optimization criterion during the design of embedded systems, in order to maximize their battery lifetime. The System-on-Chip (SoC) architecture comprising synchronous tiles or islands is an architectural template for low-energy and high-performance application domain specific system instantiations.

The communication architectures in the SoC architectural template can be divided into inter-tile and intra-tile architectures. Inter-tile architectures take care of the communication between the tiles, which is not very frequent if the application mapping is done properly and where a significant latency can be tolerated. Intra-tile communication architectures provide the means for transferring data between the components of the same tile, the local memories and processing elements for instance.

The requirements on the intra-tile communication network are more severe than the inter-tile ones. Inside the tile, the communication bandwidth is large, for the wireless and multimedia application domain our measurements indicate a bandwidth of a few tens of Gbps is needed. Additionally, latency should not exceed one, or maximally two, cycles and the energy per transfer should be very low, given the large bandwidth. Currently, past solutions are being re-used for intra-tile communication, such as point-to-point connections and crossbars, in the case of processing architectures with a centralized local memory [1].

For global energy efficiency reasons, however, new system architectures are moving toward fully software-controlled solutions. Distributed local memory organizations consisting of software-controlled scratch-pad memories instead of caches are used. New processor architectures are also moving toward software-controlled VLIW architectures, which offer the right amount of programmability and energy efficiency to fill the gap between ASICs and general purpose processors for various application domains. The TI C54x [2], for instance, has 4 local data memories, 5 local program memories and 3 load/store units. Shared buses, based internally on crossbars, are used for the communication in this architecture. They consume, however, too much energy per bit and are not scalable in number of connected components or technology node, see [3]. Furthermore, the number of memories and load/store units will further increase to fill the massively parallel datapaths of the future in order to exploit the application level parallelism.

Thus, a more scalable and energy-efficient programmable communication architecture is required to meet the demands of such platforms. It should be software-controlled and not run-time hardware based as current advocated solutions. The software control should move the exploration penalty fully to design time so that at run-time energy and delay overhead becomes negligible. Furthermore, it should be programmable in order to be flexible enough to handle several applications, but not designed under worst-case connectivity assumptions to improve energy efficiency.

On the other hand, interconnect wire technology scaling trends further deteriorate the energy efficiency of communication architectures, because the scaled wires cannot keep up with the improved performance and energy characteristics of the scaled transistors.

In this paper we describe an application domain specific, software-controlled communication network for intra-tile communication that is based on a light-weight implementation of multiple segmented buses. This architecture is scalable to future technology nodes and to more complicated systems. We also illustrate a number of optimizations that are required at the architectural and the physical design level in order to achieve energy-efficient intra-tile communication.

II. Related work

Most of the past and current research on communication architectures has been focused on the communication between the SoC tiles. Emerging industrial standards, such as the AMBA bus [4], CoreConnect [5], STBus [6], WISHBONE [7] as well as a number of academic contributions, like NoCs [8], [9] and self-timed segmented buses [10], all target this type of communication. The architecture proposed in this paper uses a much finer-grain segmentation and much simpler control than the one proposed in [10].

In the context of intra-tile communication, however, the amount of literature is limited. As already mentioned, current industrial System-on-Chip implementations rely on textbook [11] solutions such as point-to-point connections [1], shared buses [2] and crossbars [12]. These solutions, however, are general purpose communication architectures, which cannot provide both the energy-efficiency, the programmability and the scalability that will be required by future massively parallel processing architectures [3]. Each of the existing solutions can satisfy only one of these major requirements.

Segmented buses are not a novel communication mechanism as such. They were initially developed in the context of super-computing, in order to speed up computations on parallel architectures in the mid 90's, see [13] for instance. Chen et al [14] have illustrated the potential to use them also for communication energy optimization. They did not, however, show how such an architecture can be programmed or controlled. Their switch implementation is

0-7803-9294-9/05/$20.00 ©2005 IEEE