

DMA-Aware Scheduling for Multiprocessor Systems-on-Chip

Morteza Damavandpeyma^{*,1,2}, Sander Stuijk^{*,1},
Twan Basten^{*,†,1}, Marc Geilen^{*,1}, Henk Corporaal^{*,1}

^{*} *Electrical Engineering Department, Eindhoven University of Technology, The Netherlands*

[†] *Embedded System Institute, Eindhoven, The Netherlands*

ABSTRACT

Multi-processor Systems-on-Chip (MPSoCs) are becoming increasingly popular as these systems are able to fulfill the increasing demand for more computational power requested by emerging applications. To further increase the performance and at the same time decrease the energy consumption of these systems, on-chip memories are used. [Yin09] These memories limit the number of off-chip memory accesses. On-chip memory can be used as caches or scratchpad memories (SPMs). SPMs have become an efficient replacement for caches in novel embedded systems, due to their lower energy/area cost and better predictability [Ban02]. Due to limitations in the size of on-chip memories, only some portion of the application's code can be loaded to these on-chip memories. Applications must therefore be split into several smaller tasks. During the execution of the application only a few task can be loaded to on-chip memory simultaneously. After the completion of one task, its code can be discarded from the on-chip memory and the freed-up space can be used to load another task. Many MP-SoCs require the processor to explicitly control this loading process. As a result, the loading of a new code segment to the on-chip memory cannot be done in parallel with the execution of a task. This execution scheme will obviously increase the run-time of the application. Direct memory access (DMA) was devised to liberate processors from transferring data between different memories in a memory hierarchy. Using a DMA, the transfer of code to an on-chip memory and the execution of a task on a processor can be overlapped. In this poster presentation we will show that the use of a DMA in an MPSoC can improve the run-time of applications. We will also present a novel scheduling technique that exploits DMAs in MPSoCs to optimize the run-time of applications.

KEYWORDS: Scratchpad Allocation; Multi-processor System; Memory Hierarchy; Performance Optimization;

References

- [Yin09] Y. Ying et. al. An ILP formulation for task mapping and scheduling on multi-core architectures. In *DATE*, pages 33–38, 2009.
- [Ban02] R. Banakar et al. Scratchpad memory: design alternative for cache on-chip memory in embedded systems. In *CODES*, pages 73–78. ACM, 2002.

¹{m.damavandpeyma, s.stuijk, a.a.basten, m.c.w.geilen, h.corporaal}@tue.nl

²Supported by the Dutch Technology Foundation STW, project NEST - 10346