Composable Local Memory Organisation for Streaming Applications on Embedded MPSoCs

Jude Ambrose[1], Anca Molnos[1], Andrew Nelson[1],
Sorin Cotofana[1], Kees Goossens[2], and Ben Juurlink [3]

[1]Delft University of Technology
[2]Eindhoven University of Technology
[3]Technische Universität Berlin
Email: a.m.molnos@tudelft.nl, k.g.w.goossens@tue.nl, juurlink@cs.tu-berlin.de

ABSTRACT

Multi-Processor Systems on a Chip (MPSoCs) are suitable platforms for the implementation of complex embedded applications. An MPSoC is composable if the functional and temporal behaviour of each application is independent of the absence or presence of other applications. Composability is required for application design and analysis in isolation, and integration with linear effort. In this paper we propose a composable organisation for the top level of a memory hierarchy. This organisation preserves the short (one cycle) access time desirable for a processor's frequent local accesses and enables the predictability demanded by real-time applications. We partition the local memory in two blocks, one private, for local tile data, and another shared for inter-tile data communication. To avoid application interference, we instantiate one such shared local memory block and an Remote Direct Memory Access (RDMA) for each application running on the processor. We implement this organisation on an MPSoC with two processors on an FPGA. On this platform we execute a composition of applications consisting of a JPEG decoder, and a synthetic application. Our experiments indicate that an application's timing is not affected by the behaviour of another application, thus composability is achieved. Moreover, the utilisation of the RDMA component leads to 45% performance increase on average for a number of workloads covering a large range of communication/computation ratios.

ACM Categories & Subject Descriptors

C.1.2: Multiprocessors, D.4.5: Reliability

General Terms

Design, Performance, Verification

Keywords

Composability, Direct Memory Access, MPSoC

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
CF'11, May 3–5, 2011, Ischia, Italy.
Copyright 2011 ACM 978-1-4503-0698-0/11/05 ...$10.00.

1. COMPOSABLE LOCAL MEMORY

Most state-of-the-art MPSoCs follow a tile based organisation comprising processor and memory tiles that communicate via an interconnect. Processor tiles typically embed local memories. Applications running on such platforms consist of communicating tasks that process infinitely-long input data streams. These applications share the MPSoC resources, which may result in inter-application functional and temporal interference due to resource request conflicts. In this context composability [3, 4] is a desired platform property, as it enables application design and analysis in isolation, and integration with linear effort. Prior work proposed resource virtualisation techniques to achieve composability, and demonstrated their feasibility for processors, memory tiles, and interconnect [3]. The techniques proposed for memory tiles [1] incur arbitration delay, thus do not straightforwardly apply to the local memories that have to ensure fast (one cycle, preferably) access to data.

To obtain a composable, fast local memory organisation we take four steps, as presented in Figure 1. First, we employ a Remote Direct Memory Access (RDMA) component (Figure 1(B)) to avoid that an application monopolises the processor potentially indefinitely, due to stalls on remote reads. Second, to decrease contention, we partition the local memory into a private memory and a pair of shared memories – one for the outbound transactions initiated by the local processor, and one for the inbound transactions initiated by a remote processor, as shown in Figure 1(C). In this manner each of the memories is shared by at most two access initiators. Third, to further avoid arbitration and its potentially large time penalty we propose to employ dual-ported memories (Figure 1(D)). Finally, to achieve composability, i.e, no local memory contention when multiple applications are mapped on a tile, we propose to replicate the shared memories and the RDMA (Figure 1(E)). Thus there is one RDMA and a pair of shared memories per application mapped on a tile. Predictability, crucial for real-time applications, is preserved by the four steps above.

2. EXPERIMENTAL RESULTS

The experimental platform consists of two processor tiles, one memory tile, and a NoC [2], implemented on a Xilinx ML-510 FPGA board. Each processor tile contains a MicroBlaze core and local memory. The MicroBlaze accesses the local memory via a one-cycle delay Local Memory Bus.

We assess composability using a workload consisting of a JPEG decoder, and a simple synthetic application (*A1*). Figure 2(a) presents the task graph and the task to proces-

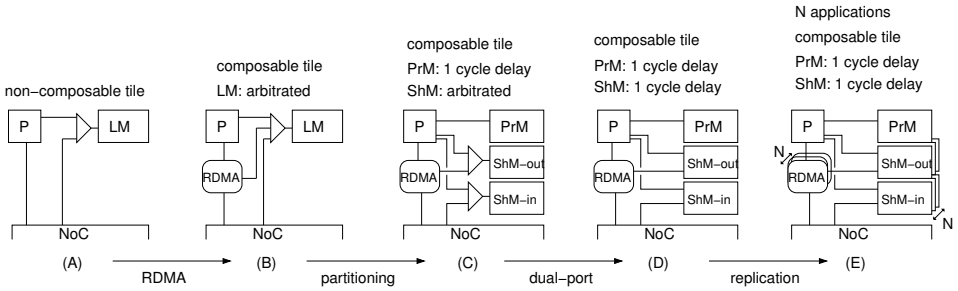
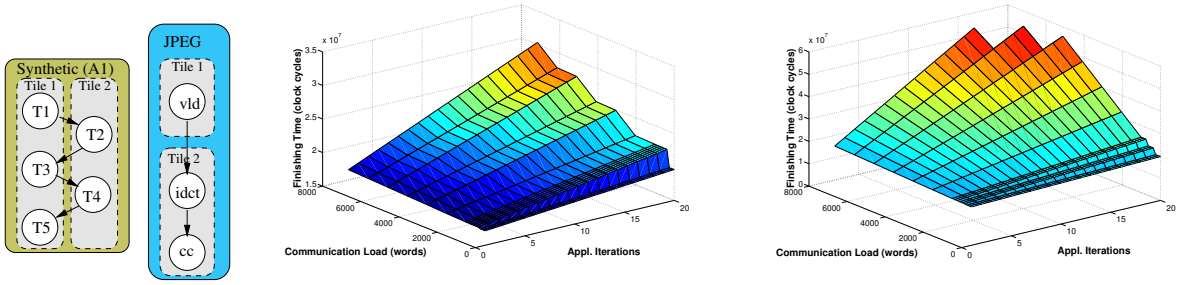


Figure 1: Steps towards a composable local memory



(a) Applications (b) A1 with RDMA (c) A1 without RDMA

Figure 2: Application workload and performance comparison

sor tile mapping of both applications. We investigate the difference in the JPEG's response time between two cases: the JPEG application executing alone (*JPEG-single*), and in combination with the synthetic application *A1* (*JPEG-multi*). We compare two tile organisations: (1) when both applications share a single RDMA engine (1 RDMA per tile); and (2) when the RDMAs are replicated (1 RDMA per application). Figures 3 presents the response time differences between *JPEG-multi* and *JPEG-single* for the *vld* task. As this figure indicates, the response times differ when using a single RDMA per tile, revealing interference, as expected. On the other hand, the response time difference is zero with an RDMA per application, as there is no interference. This suggests that our proposal is composable.

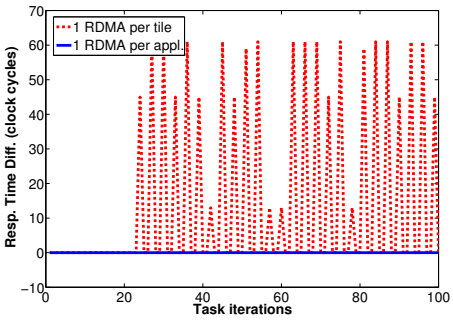


Figure 3: JPEG, *vld* response time difference (*JPEG-multi - JPEG-single*)

We compare the performance of two processor tile organisations: one with RDMAs and the other without RDMAs. The platform without RDMAs allows Microblazes to transfer data via a PLB bus to the NoC. This implies that the MicroBlaze is blocked until the entire data is transferred to the NoC, unlike in the case when RDMA performs the transfer. For these experiments we utilise only the synthetic application because its communication and computation loads can be easily varied. The RDMA increases the application performance by 70% for a communication load as large as 8000 words, as shown in Figure 2(b)(c). The RDMA is an inexpensive block as it uses very few FPGA resources, namely 332 look-up tables and 276 flip-flops, representing 1% and 0.7% from the Xilinx ML-510 FPGA's look-up tables and flip-flops, respectively.

3. CONCLUSIONS

In this paper we propose a composable local memory organisation for a tiled MPSoC executing a number of streaming applications. This organisation utilises a Remote Direct Memory Access (RDMA) unit per application and employs partitioned, dual-ported memory blocks. As a result, this scheme is predictable and it preserves fast access time to the local memory – in our case one-cycle accesses. Experiments suggest that composability is achieved and, in case communication and computation is overlapped with RDMAs, performance is increased with 45% for a number of synthetic workloads covering a large range of communication/computation ratios. Employing the RDMAs led to no significant difference in the number of FPGA look-up tables utilised and increased the total number of flip-flops's by 5%.

4. REFERENCES

[1] B. Akesson *et al.* Composable resource sharing based on latency-rate servers. In *DSD*, 2009.

[2] K. Goossens *et al.* The aethereal network on chip after ten years: Goals, evolution, lessons, and future. In *DAC*, 2010.

[3] A. Hansson *et al.* CoMPSoC: A template for composable and predictable multi-processor system on chips. *ACM Trans. Des. Autom. Electron. Syst.*, 2009.

[4] H. Kopetz. *Real-Time Systems: Design Principles for Distibuted Embedded Applications*. Kluwer Academic Publishers, 1997.