Resource-aware Computer Vision Application on Heterogeneous Multi-tile Architecture

Éricles Sousa*, Johny Paul†, Vahid Lari*, Frank Hannig*, Jürgen Teich*, and Walter Stechele†
*Hardware/Software Co-Design, University of Erlangen-Nuremberg, Germany
†Institute for Integrated System, Technical University of Munich, Germany
*{ericles.sousa, vahid.lari, hannig, teich}@cs.fau.de
†{johny.paul, walter.stechele}@tum.de

I. Description

The ever growing demand for autonomous systems has pushed algorithms to new levels of flexibility, which assures a resource-aware implementation on fast reconfigurable architectures capable to adapt at run-time according to changing requirements and constraints. Here, of great importance are computer vision algorithms used to obtain important information about the spatial position of objects in a scene. However, the efficient exploration of their capability to adapt to different environment conditions in real-time is still a challenge. Therefore, in order to provide applications more flexibility, a self-organizing computing paradigm called *invasive computing* can be used [1]. In invasive computing, applications running on an MPSoC and competing for resources have the ability to explore and dynamically spread their computations to neighborhood processors. Here, an application may dynamically claim and reserve resources (invade), employ them for parallel execution (infect), and finally release them (retreat). Our demonstration will present the benefits of invasive computing by showing the efficiency and utilization improvements in a resource-aware manner by algorithmic selection of different invasive resources, such as TCPA (tightly-coupled processor array), and RISC processors. More specific we present a dynamic load balancing between multiple RISC cores and a TCPA, based on invasive mechanisms supported by our operating system and the agent system. By exploiting information from the invasive run-time system we consider two variations of the well-known Harris Corner Detector both optimized for RISC [2] and TCPA architectures [3].

In order to demonstrate the concepts and benefits of invasive computing, a multi-tile architecture shown in Figure 1 can be partitioned up to 6 FPGAs of the *Synopsys CHIPit prototyping platform* [4]. The multi-tile architecture consists of regular RISC processor, TCPA, I/O, and memory tiles connected through a network-on-chip. The RISC tiles consist of four CPU cores including a specific reconfigurable fabric, L2 cache, local memory, a debug support unit and the network adapter. Each CPU core has 8KB instruction cache and 32KB data cache. The level 2 cache will be 128KB. The tile-local memory (TLM) is connected to SSRAM extension boards on the CHIPit system offering 8MB per tile. The TCPA tile is used as accelerators for computationally intensive tasks. That is, such processor arrays are designed to support many application-specific algorithms in image and signal processing domains. The TCPA architecture is composed of an array of VLIW processing elements (PEs). Each processing element in the array is equipped with a dedicated hardware component for invasion called invasion controller (iCtrl). Along with a processor array, each TCPA tile contains a RISC processor which is used for communication and configuration of regions of claimed processors with proper programs. It is therefore called Configuration and Communication Processor. A TCPA tile contains also a set of peripherals for supervising multiple concurrent invasions and program executions on regions of PEs. The memory tile is connected to a 256 MB DDR2 memory on an extension board. The I/O tile is connected to ethernet extension board, where input and output images are received from the host machine and transferred into to the CHIPit system which return the computed data to be displayed by the host machine. Inside the CHIPit system, the Harris Corner Detector and SIFT feature

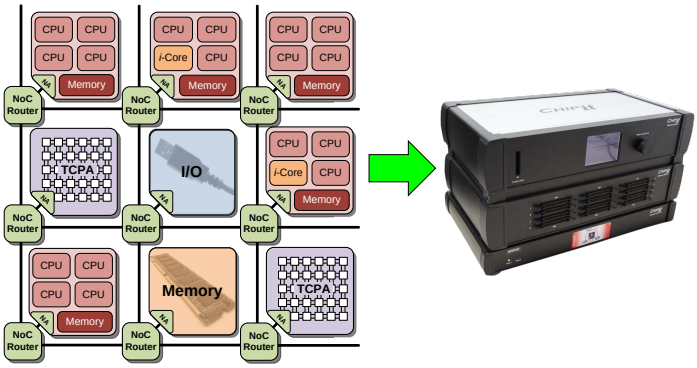

Fig. 1. Invasic demonstrator: A multi-tile architecture as prototyped on the CHIPit system.

matching are executed in parallel exploring as many cores as currently available. In our demonstration we consider the advantages of each tile to promote a synergistic environment for our computer vision application. On the one hand, given the characteristics of general purpose processors, the RISC tiles are preferable to execute irregular algorithms and floating-point operations. On the other hand, the TCPA tile is the favorite to execute regular loops, while providing a better performance as well as power consumption. Moreover, by exploring the TCPA tile applications gain the ability to fulfill constraints in two ways: (a) In case of tight throughput requirements, the number of corners detected may be adjusted depending on the number of available resources by run-time selection. (b) A certain level of quality can be guaranteed by dynamically adjusting the throughput with respect to available resources. Finally, we show that resource-aware computing can lead to significant improvements in throughput and accuracy of computer vision algorithms, as compared to a non-invasive version on the same hardware platform.

II. Acknowledgment

This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre “Invasive Computing” (SFB/TR 89).

References

[1] J. Teich, J. Henkel, A. Herkersdorf, D. Schmitt-Landsiedel, W. Schröder-Preikschat, and G. Snelting, *Multiprocessor System-on-Chip: Hardware Design and Tool Integration*. Springer, 2011, ch. 11, Invasive Computing: An Overview, pp. 241–268.

[2] J. Paul, W. Stechele, M. Kröhnert, T. Asfour, B. Oechslein, C. Erhardt, J. Schedel, D. Lohmann, and W. Schröder-Preikschat, “Resource-aware harris corner detection based on adaptive pruning,” in *Proceedings of the Conference on Architecture of Computing Systems (ARCS)*. Springer Lecture Notes in Computer Science (LNCS).

[3] É. Sousa, A. Tanase, F. Hannig, and J. Teich, “Accuracy and Performance Analysis of Harris Corner Computation on Tightly-Coupled Processor Arrays,” in *Proceedings of the Conference on Design and Architectures for Signal and Image Processing (DASIP)*. IEEE, Oct. 2013.

[4] Synopsys, http://www.synopsys.com/systems/fpgabasedprototyping/chipit/pages/default.aspx.