Lessons Learned from last 4 Years of Dynamically Reconfigurable Computing

Walter Stechele, Christopher Claus, Andreas Laika
Institute for Integrated Systems
Technische Universität München

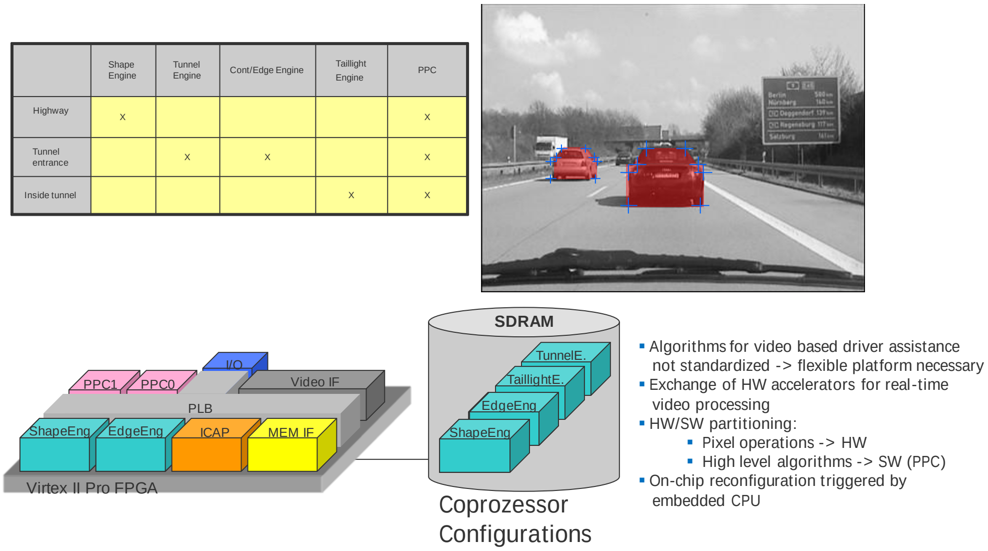
Abstract: Partial dynamic reconfiguration of FPGAs was investigated for video-based driver assistance applications during the last 4 years. High-level application software was combined with dynamically reconfigurable hardware accelerators in selected scenarios, e.g. vehicle lights detection, optical flow motion detection. From the beginning of the project, various research challenges have been targeted, including hardware/software partitioning between embedded RISC and accelerators, granularity of reconfigurable regions, as well as the impact of the reconfiguration process on system performance. This article will review the status of these research challenges and present an outlook on future challenges, including reconfiguration look ahead. Challenges will be illustrated on robotic vision scenarios with dynamically changing computational load from soft real-time and hard real-time applications.

1 Vision-based Driver Assistance

Reconfigurable computing for vision-based driver assistance application was investigated by the authors in the AutoVision project from 2005-2009. A representative scenario is illustrated in Figure 1. During daylight a driver on the highway is approaching a tunnel entrance. With changing driving conditions, from daylight on the highway to darkness inside the tunnel, the algorithms for visual analysis will change as well. In full daylight cars might be detected based on their contours, when approaching the tunnel contrast enhancement might be used for the dark tunnel entrance region, inside the tunnel vehicle lights might be detected and distinguished from tunnel lights.

Each visual algorithm may use different hardware accelerators on FPGA. Predefined partial bitstreams for hardware accelerators can be loaded into the FPGA whenever needed. The Internal Configuration Access Port (ICAP) within Xilinx FPGA is used to write the partial bitstream into the FPGA and reconfigure hardware accelerators in partially reconfigurable regions. Figure 1 shows a table with different driving conditions on top left, the FPGA with two CPUs (PowerPC PPC0 and PPC1) and two reconfigurable regions for hardware accelerators on bottom left.

Dagstuhl Seminar Proceedings 10281
Dynamically Reconfigurable Architectures
http://drops.dagstuhl.de/opus/volltexte/2010/2835



	Shape Engine	Tunnel Engine	Cont/Edge Engine	Taillight Engine	PPC
Highway	x				x
Tunnel entrance		x	x		x
Inside tunnel				x	x

- Algorithms for video based driver assistance not standardized -> flexible platform necessary
- Exchange of HW accelerators for real-time video processing
- HW/SW partitioning:
  - Pixel operations -> HW
  - High level algorithms -> SW (PPC)
- On-chip reconfiguration triggered by embedded CPU

Figure 1: AutoVision overview

2 Research Challenges

When the AutoVision project started in 2005, the authors defined various research challenges to be investigated. The status of these research challenges is reviewed and summarized here. Five challenges have been identified:

- Hardware/software partitioning
- Granularity of reconfigurable regions
- Impact of reconfiguration on system performance
- Online bitstream modification
- Reconfiguration look ahead

The first research challenge was on hardware/software partitioning of visual algorithms on CPU and hardware accelerators. Studies within AutoVision covered various representative algorithms, including feature point extraction, motion detection and vehicle light detection. In all these cases, hardware accelerators were used for the low level pixel processing only, whereas high level algorithms were implemented on PPC. For motion detection, optical flow vectors have been computed on hardware accelerators, clustering of motion vectors for motion detection on PPC. Similarly for vehicle light detection, light spot detection was done in hardware accelerators, evaluation of symmetry between light pairs in software on PPC. Results have been published in [DATE 2008, FPL 2009, IV 2009].

The second research challenge was on the granularity of reconfigurable regions. A lot of investigation was done within the Reconfigurable Computing program SPP 1148, ranging from coarse grain to fine grain reconfiguration, from application specific instruction set processors to dedicated hardware accelerators, from switching between dedicated configuration states to hyper reconfiguration. A summary of results was published in [SPP 1148].

The third research challenge was on the impact of reconfiguration on system performance. Within AutoVision, interleaving of application and reconfiguration tasks was deeply investigated. During the initial phase of AutoVision, the goal was to load new hardware accelerators without losing a video frame. After some optimization on the reconfiguration process, i.e. removing redundant information from partial bitstreams and increasing data rate for bitstream loading into ICAP, hardware accelerators could be reconfigured within a few msec, such allowing multiple reconfigurations per video frame. This could be exploited to run two hardware accelerators alternating on the same reconfigurable region, in order to use a smaller FPGA device [ARC 2010, ARCS 2006, FPL 2008, ISVLSI 2007, RAW 2007, XCell 2010].

The fourth research challenge was dealing with online bitstream modification. This has been investigated in the group of Jürgen Becker and Michael Hübner from KIT. They showed how to modify partial bitstreams in order to place hardware accelerators on various locations on the FPGA, and route their connections online [SOCC 2009].

The fifth research challenge was on reconfiguration look ahead. How to decide for new configurations during runtime of the system? This was controllable in the AutoVision scenario, where reconfiguration was related to changes in driving conditions. But this might become highly challenging in more complex scenarios. We will illustrate these open challenges in the following section.

Overall, in the AutoVision project the authors could demonstrate the benefit of dynamic partial reconfiguration in todays FPGA devices for vision-based driver assistance applications. Although vendor tool support for dynamic partial reconfiguration is continuously improving, the design effort has shown to be quite high in order to get the system running.

3 Reconfiguration Look Ahead

In a robotic scenario, soft real-time and hard real-time applications will contribute with dynamically changing computational load, e.g. robotic vision and robot control. Efficient resource utilization in Multiprocessor System-on-Chip (MPSoC) requires advanced reconfiguration planning. MPSoC might contain reconfigurable processor arrays and heterogeneous RISC cores. Various strategies for reconfiguration planning might be investigated, e.g. central vs. distributed vs. self-organizing approaches.

Imagine a robotic vision scenario with a team of rescue robots inside a burning house. Various tasks have to be computed simultaneously, e.g. 3D modeling of environment, using camera(s) on robot and combining multiple views from team members; analyzing audio input ("help" shouts); finding and localizing objects of interest (persons, explosives); walking into hazardous area; continuously updating 3D model of environment; watching for falling objects. Alarms might be triggered from own motion detection algorithms or from other robots over network I/O. The computing platform may include heterogeneous RISC clusters and an array of reconfigurable hardware accelerators.

Figure 2 shows a simplified task graph for a rescue robot. A real-time control loop for motion control is depicted on the left hand side; two vision routines are depicted in center and on right hand side. Video input is analyzed twofold, (1) by optical flow algorithm and motion analysis, in order to watch for hazards, (2) by stereo vision, detection of Region of Interest (ROI) for object hypothesis generation and verification, and motion planning, in order to interact with surrounding objects. Cooperating robots from the team might contribute video and ROI information over the network. Tasks might be mapped on an array of hardware accelerators (blue) and on a cluster of RISC CPUs (green).

One sample option of resource utilization for an array of hardware accelerators and a RISC cluster is depicted in Figure 3. Dependencies between tasks are shown according to the task graph from Figure 2. Motion control is computed in fixed, reserved time slots on a RISC core, other tasks are scheduled over all available cores on a best effort scheme. Watching for hazards consists of a sequence of optical flow (on array) and motion analysis (on RISC). An alarm might be triggered as a result from motion analysis, e.g. detection of falling objects. Search for objects of interest consists of a sequence of stereo vision, ROI detection (both on array), hypothesis verification and motion planning (both on RISC).

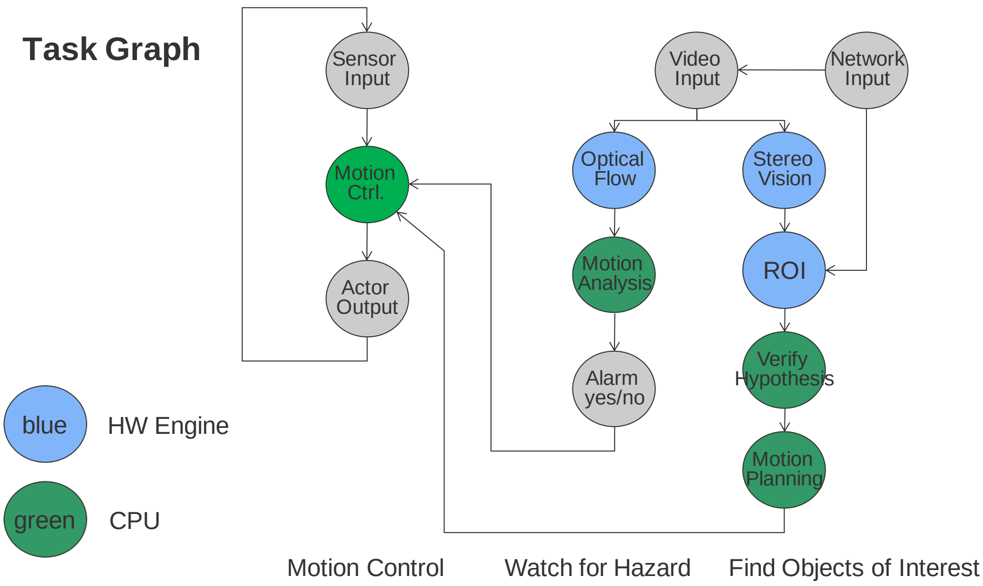


Figure 2: Task graph for rescue robot

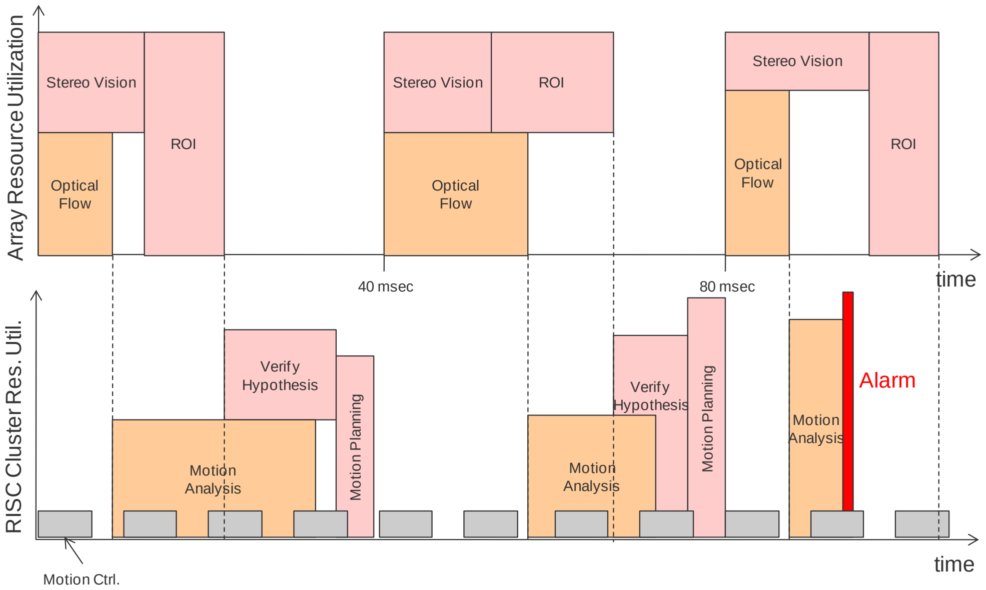


Figure 3: Resource utilization for an array of hardware accelerators (above) and RISC cluster (below).

This simplified scenario shows the complexity of possible configurations and mappings. As compared to the AutoVision scenario, where reconfiguration was mainly corresponding to external driving conditions, in the robotic scenario, reconfiguration corresponds to both external and internal conditions. Efficient resource utilization should minimize blank spaces in the resource utilization graph from Figure 3. It seems quite challenging to take runtime decisions on new configurations and to trigger the reconfiguration process, taking into account reconfiguration cost in terms of latency and power consumption. Too many reconfigurations might keep the system busy and contribute to power consumption, without executing application tasks. Modules that are going to be used again soon might not be reconfigured, but clock/power gated instead.

4 Conclusion

Partial dynamic reconfiguration of FPGA devices was exploited in video-based driver assistance scenario twofold, with change of driving conditions and with two subsequent hardware accelerators on same FPGA area. Some research challenges have been tackled, including hardware/software partitioning for vision algorithms, granularity of reconfigurable regions, impact of reconfiguration on system performance, and online bitstream modification. But reconfiguration look ahead still remains an open research challenge, as illustrated in a robotic vision scenario with dynamically changing soft real-time and hard real-time applications.

Acknowledgements

This work was funded by the German Research Foundation (DFG) under the research program on Reconfigurable Computing SPP 1148.

References

[ARC 2010] C. Claus, R. Ahmed, F. Altenried, W. Stechele, "Towards rapid dynamic partial reconfiguration in video-based driver assistance systems", 6th International Symposium on Applied Reconfigurable Computing, ARC 2010, Bangkok, Thailand, March 17-19, 2010

[ARCS 2006] C. Claus, F. Müller, W. Stechele, "Combitgen: A new approach for creating partial bitstreams in Virtex-II Pro devices", International Conference on Architecture of Computing Systems, ARCS 2006, Frankfurt, March 16, 2006

[DATE 2008] N. Alt, C. Claus, W. Stechele, "Hardware/software architecture of an algorithm for vision-based real-time vehicle detection in dark environments", Design, Automation & Test in Europe (DATE 2008), Munich, March 10-14, 2008

[FPL 2008] C. Claus, B. Zhang, W. Stechele, L. Braun, M. Hübner, J. Becker, "A multi-platform controller allowing for maximum dynamic partial reconfiguration throughput", Proceedings of the International Conference on Field Programmable Logic and Applications (FPL08), Heidelberg, Germany, September 8-10, 2008

[FPL 2009] C. Claus, R. Huitl, J. Rausch, W. Stechele, "Optimizing the SUSAN corner detection algorithm for a high speed FPGA implementation", 19th International Conference on Field Programmable Logic and Applications (FPL09), Prague, Czech Republic, August 31 - September 2, 2009

[ISVLSI 2007] C. Claus, B. Zhang, M. Huebner, C. Schmutzler, J. Becker, W. Stechele, "An XDL-based busmacro generator for customizable communication interfaces for dynamically and partially reconfigurable systems", Workshop on Reconfigurable Computing Education at ISVLSI 2007, Porto Alegre, Brazil, May 12, 2007

[IV 2009] C. Claus, A. Laika, L. Jia, W. Stechele, "High performance FPGA based optical flow calculation using the census transformation", The Intelligent Vehicles Symposium (IV'09), Xi'an, China, June 3-5, 2009

[RAW 2007] C. Claus, J. Zeppenfeld, F. H. Müller, W. Stechele, "A new framework to accelerate VirtexII Pro dynamic partial self-reconguration", Reconfigurable Architectures Workshop (RAW), Long Beach, CA, March 26-27, 2007

[SOCC 2009] M. Niknahad, M. Huebner, J. Becker: “Method for improving performance in online routing of reconfigurable nano architectures”, SOC Conference, 2009

[SPP 1148] C. Claus, W. Stechele, "AutoVision - Reconfigurable Hardware Acceleration for Video-Based Driver Assistance", In: Platzner, Teich, Wehn (Editors): Dynamically Reconfigurable Systems, ISBN 978-90-481-3484-7, Springer, 2010

[XCell 2010] C. Claus, F. Altenried, W. Stechele, "Dynamic Partial Reconfiguration of Xilinx FPGAs Lets Systems Adapt on the Fly", Xcell journal, pp 18-23, first quarter 2010