Research and implementation challenges of RTOS support for heterogeneous computing platforms

Martin Cornil, Antonio Paolillo, Joël Goossens, Ben Rodriguez

PARTS Research Centre
Université Libre de Bruxelles
HIPPEROS S.A.

I. Introduction

Nowadays, embedded systems are massively present in our live and run about every electronic equipment around us. Moreover requirements for equipments become more and more challenging, embedded systems need to become cheaper, lighter, faster, energy efficient and as modulable and flexible as possible. This last requirement, which usually applies to software, became also possible for hardware since the last 30 years thanks to the apparition of Field Programmable Gate Arrays (FPGA) and more recently with a new capability enabling reconfiguration of hardware sections on-the-fly.

With respect to Real-Time Operating Systems (RTOS) most efforts in the research community to improve embedded systems have been made around single core or partitioned multi-core platforms [1] that, as a consequence, are now fairly well understood. However, these approaches and design strategies are no longer able to follow the requirements of today's embedded systems when multi-core (with shared ressources) and/or heterogeneous systems come into play and they become therefore an interesting research area. For example, image processing is more and more used in the industry but, because of the increasing resolutions of the camera used, it becomes ineffective and power hungry to be realised on a general purpose processor. This is why adding hardware accelerators like Graphic Processing Unit (GPU) or coprocessors in these embedded systems becomes interesting.

To satisfy the high computational performance and flexibility requirements of contemporary embedded systems, reconfigurable computing seems to be a promising approach. Of special interest are those heterogeneous architectures where a microprocessor is tightly connected with a reconfigurable FPGA, constituting a so called Reconfigurable System-on-Chip (RSoC). These RSoC have the advantage to be flexible (reconfigurable) and are able to run some tasks in parallel. Making good use of hardware parallelism in real-time applications is known to be an effective approach to reduce power consumption and improve performance of real-time applications [2], [3]. However, because of the complexity of these systems and underlying applications, the usage of an Operating System (OS) is essential to provide the necessary abstraction of hybrid hardware and software computational resources. Moreover, a deep and careful understanding of the intrinsic scheduling constraints and overhead caused by the reconfiguration activities and the potential sharing of computational resources must be done, especially for a RTOS. Within this context, this work attempts to develop a solution to manage efficiently RSoC software and hardware resources from a Real-Time Operating System running a set of critical application-specific processes.

II. Context

Today, embedded systems are everywhere. They are present in our pocket under the form of a smartphones, in our household machines, cars, airplanes, medical equipment, etc. The embedded system market covers more than 95% of the processors sold today. Unlike a general purpose computer, the design of an embedded system is considerably more constrained because of their specific requirements in Size, Weight and Performance (SWaP) as well as their cost and power consumption. Moreover, contemporary embedded systems are increasingly becoming multipurpose which means they need to support more than one application which may be different in their nature and criticality. This is the reason why heterogeneous platforms which integrate multiple processing unit (e.g.: multi-core, coprocessor, GPU) seems to be a good and modern solution to deal with the requirements of modern embedded systems. Indeed the heterogeneity of these platforms will help to reduce the number and size of the processing units as well as their power consumption.

To efficiently deal with the dynamic requirements of modern embedded systems, an idea proposed in the H2020 Tulipp project [4], [5] is the use of hybrid platforms which embed a Reconfigurable System-on-Chip (RSoC) composed of a general purpose processor and a Field Programmable Gate Area (FPGA). However the heterogeneity of these platforms induces an increase in complexity to efficiently manage the resources. The support of these hardware features by a Real-Time Operating System becomes unavoidable to manage reconfiguration activities, shared hardware resource allocation and related scheduling to meet the timing requirements of the target applications.

III. Aim of the work

We started the Tulipp project in February 2016. In this context, we aim to study and implement RTOS solutions for heterogeneous embedded platforms to provide high performance yet power efficient solutions to real-time application designers. More precisely, we will focus our effort on hybrid target platforms as the Zynq-7000 and the Zynq UltraScale+ produced by Xilinx®, the main actor of today's RSoC market. The first RSoC integrates an ARM Cortex A9 symmetric dual-cores general purpose Central Processing Unit (CPU) tightly interconnected with a FPGA in a single chip. This kind of heterogeneous architecture can provide high computational performance and energy efficiency by offloading complex task to hardware as well as good cost efficiency by reusing the same hardware (CPU and/or FPGA) for different applications. However, the flexibility of the FPGA reconfiguration comes with a significant scheduling latency cost that has to be taken into account by the scheduler to optimise resource usage as well as avoid critical tasks to miss their deadlines.

We use HIPPEROS as the targetted embedded RTOS [3], [6]. Based on state-of-the-art modeling techniques [7]–[14], we will study and implement heterogeneous multi-core support in HIPPEROS and techniques to efficiently integrate dynamic FPGA reconfiguration in a hard real-time environment. The goal is to apply real-time theoretical scheduling research about heterogeneous platforms in practice and evaluate it.

IV. Research questions

To define the scope of this research project, we will try to answer to the following questions:

- How to integrate FPGA-based functions into real-time operating system tasks or services? (e.g.: what software architecture, what application programming interfaces)?
- How to integrate FPGA resources in the constrained deadlines sporadic and periodic tasks model? (e.g.: memory-like resources or "virtual" coprocessors)?
- Can we reuse and modify existing scheduling theory to include these type of components?
- What will be the impact of these heterogeneous components on scheduling analysis (e.g.: reconfiguration and execution latency, task dependencies)?
- Will the parallelism introduced by the use of FPGA resources help to improve performance, platform utilisation and energy efficiency of the targetted real-time applications?

References

[1] S. Baruah, M. Bertogna, and G. Buttazzo, Multiprocessor scheduling for real-time systems. Springer, 2014.

[2] A. Paolillo, J. Goossens, P. Hettiarachchi, and N. Fisher, "Power minimization for parallel real-time systems with malleable jobs and homogeneous frequencies," in 20th International Conference on Embedded and Real-Time Computing Systems and Applications, August 2014.

[3] A. Paolillo, P. Rodriguez, N. Veshchikov, J. Goossens, and B. Rodriguez, "Quantifying energy consumption for practical fork-join parallelism on an embedded real-time operating system," in Proceedings of the 24th International Conference on Real-Time Networks and Systems, ser. RTNS '16. New York, NY, USA: ACM, 2016, pp. 329–338. [Online]. Available: http://doi.acm.org/10.1145/2997465.2997473

[4] "The tulipp h2020 project," http://www.tulipp.eu, accessed: 2017-03-31.

[5] T. Kalb, L. Kalms, D. Göhringer, C. Pons, F. Marty, A. Muddukrishna, M. Jahre, P. G. Kjeldsberg, B. Ruf, T. Schuchert, I. Tchouchenkov, C. Ehrenstrahle, F. Christensen, A. Paolillo, C. Lemer, G. Bernard, F. Duhem, and P. Millet, "Tulipp: Towards ubiquitous low-power image processing platforms," in 2016 International Conference on Embedded Computer Systems: Architectures, Modeling and Simulation (SAMOS), July 2016, pp. 306–311.

[6] A. Paolillo, O. Desenfans, V. Svoboda, J. Goossens, and B. Rodriguez, "A new configurable and parallel embedded real-time micro-kernel for multi-core platforms," in Proceedings of the ECRTS Workshop on Operating Systems Platforms for Embedded Real-Time applications (ECRTS-OSPERT '15), July 2015.

[7] M. Pagani, M. Marinoni, A. Biondi, A. Balsini, and G. Buttazzo, "Towards real-time operating systems for heterogeneous reconfigurable platforms," OSPERT 2016, July 2016.

[8] A. Biondi, A. Balsini, M. Pagani, E. Rossi, M. Marinoni, , and G. Buttazzo, "A framework for supporting real-time applications on dynamic reconfigurable FPGAs," RTSS 2016, December 2016.

[9] T. Xia, J.-C. Prevoltet, and F. Nouvel, "Microkernel dedicated for dynamic partial reconfiguration on ARM-FPGA platform," EWiLi 14, 2014.

[10] X. Iturbe, K. Benkrid, C. Hong, A. Ebrahim, R. Torrego, and T. Arslan, "Microkernel architecture and hardware abstraction layer of a reliable reconfigurable real-time operating systems R3TOS," ACM transactions on Reconfigurable Technology an Systems, February 2015.

[11] J.-J. Chen, G. Nelissen, W.-H. Huang, M. Yang, B. Brandenburg, K. Bletsas, C. Liu, P. Richard, F. Ridouard, N. Audsley, R. Rajkumar, and D. de Niz, "Many suspensions, many problems: A review of self-suspending tasks in real-time systems," Technical Report, Department of Computer Science, Dortmund University of Technology, May 2016.

[12] M. Götz, F. Dittmann, and C. E. Pereira, "Deterministic mechanism for run-time reconfiguration activities in an RTOS," 2006 4th IEEE International Conference on Industrial Informatics, 2006.

[13] C. Kohn, "Partial reconfiguration of a hardware accelerator on zynq-7000 all programmable soc devices," 2013.

[14] D. O. Hotoleanu and J.-C. Prevotet, "Implementation of a RTOS service for the management of the reconfiguration," 2011.