Criticality-aware Design Space Exploration for Mixed-Criticality Embedded Systems

Vittoriano Muttillo
Center of Excellence DEWS
Via Vetoio, 1
L'Aquila, Italy
vittoriano.muttillo@graduate.univaq.it

Giacomo Valente
Center of Excellence DEWS
Via Vetoio, 1
L'Aquila, Italy
giacomo.valente@univaq.it

Luigi Pomante
Center of Excellence DEWS
Via Vetoio, 1
L'Aquila, Italy
luigi.pomante@univaq.it

ABSTRACT

This work[1] focuses on Design Space Exploration for embedded systems based on heterogeneous parallel architectures and subjected to mixed-criticality constraints. In particular, it presents a criticality-aware evolutionary approach integrated into a reference Electronic System Level HW/SW Co-Design flow.

ACM Reference Format:
Vittoriano Muttillo, Giacomo Valente, & Luigi Pomante. 2018. Criticality-aware Design Space Exploration for Mixed-Criticality Embedded Systems. In Proceedings of ACM/SPEC International Conference on Performance Engineering (ICPE'18 Companion). ACM, NY, NY, USA, 4 pages. https://doi.org/10.1145/3185768.3185769

KEYWORDS

HW/SW Co-Design; Heterogeneous Parallel Systems; Design Space Exploration; Mixed-Criticality Systems

1 INTRODUCTION

In recent years, there has been a growing trend for switching from single-processor/core to (heterogeneous) multi-processor/core (i.e. parallel) platforms to execute embedded applications with different levels of criticality (i.e. *Mixed-Criticality Embedded Systems*, MCES). The main problem in the management of a MCES is to ensure that low criticality applications do not interfere with high criticality ones. After Vestal model [1], which first analyzed Mixed-Criticality (*MC*) system with focus on real-time performance, a series of research papers have been published [2]. In this domain, the most critical development steps are related to the *System Specification* and the *Design Space Exploration* (DSE) activities [3] and the main differences among the various works in the literature are related to the amount of information and actions that are explicitly requested from the designer and so heavily influenced by his experience. In such a context, this work present a DSE approach to support the development of heterogeneous parallel MCES. The main differences with respect to previous works (e.g. [4][5]) are related to the system behavior model, based on CSP-like (*Communicating Sequential Processes*) *Model of Computation* (MoC) and the criticality-aware HW/SW partitioning/mapping activity that exploits an evolutionary approach.

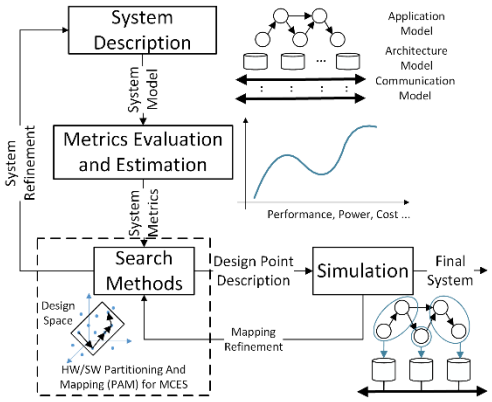

Figure 1: HW/SW Co-Design Flow.

2 HW/SW Co-Design Flow with MC Constraints

In the context of MCES, this work adopts a specific design flow (*HEPSYCODE*: *HW/SW Co-Design of Heterogeneous Parallel Dedicated Systems*) [6], based on an existing *Electronic System-Level HW/SW Co-Design Methodology* [7] (Figure 1), while introducing MC requirements. The *System Description* step defines three reference models: the *Application*, *Architecture* and *Communication Model*. The first model exploits a behavioral modeling language, named *HML* (*HEPSY Modeling Language*) [8], based on the *CSP MoC* [9]. The second model defines the basic HW components available to build the final HW architecture. These components are collected into a *Technologies Library (TL)*, a generic database that provides the characterization of the available technologies. Starting from the TL, designers define the so called *Basic Blocks (BB)*, each one composed of a set of processing units, memory units and internal and external communication links. Practically, the final HW architecture will be composed of a set of *BB* elements interconnected by means of one or more link elements (taken from the ones available in the communication model). The *Metrics Evaluation and Estimation* activities provide several metrics related to the *BB* involved in the design flow (*Affinity* [10], *Concurrency*, *Communication*, *Size*

ICPE '18, April 9–13, 2018, Berlin, Germany

ACM ISBN 978-1-4503-5629-9/18/04…$15.00
https://doi.org/10.1145/3185768.3185769

and *Load* [11]). After these steps, the reference co-design flow reaches the *DSE* step (as shown in Figure 1 and Figure 2). It includes two iterative activities: *HW/SW Partitioning And Mapping (PAM)*, that allows to explore the design space looking for feasible solution suitable to satisfy imposed constraints; *Timing Co-Simulation*, that considers suggested design points, representing mappings between application and platform models (Figure 2), to actually check for timing constraints satisfaction.

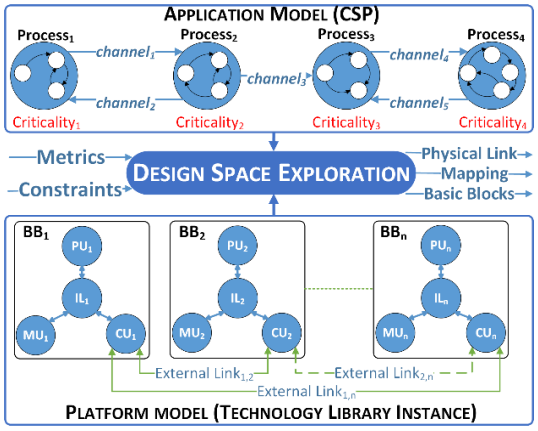

Figure 2: Design Space Exploration Approach.

The PAM activity is based on a *Genetic Algorithm* (GA) used to minimize a multi-objective cost function that quantifies the quality of each individual of the GA population. Considering the criticality levels associated to each application process, this work proposes different methods to manage MC constraints to avoid interferences derived from damages or software errors and bugs. The main idea is to drive the DSE to avoid having processes with different criticality levels allocated on the same (shared) processor/core. For this, it is exploited a metric called *Criticality Index*. The goal behind this metric is to measure how much isolated are processes with different criticality levels. So, the final cost function will have a higher value if an individual doesn't satisfy the criticality constraint. Moreover, it is also possible to constrain the initial GA population to have only feasible individuals, and/or to constrain the *crossover* and *mutation* GA operations to make the population evolving only with feasible individuals (with respect to criticality constraints) avoiding at all the generation of unfeasible solutions. With respect to the starting GA population *generation* operation, two methods are considered: **(1)** to reduce the starting random GA population deleting unfeasible individuals; **(2)** to create a starting GA population with only feasible individuals. With respect to crossover and mutation operations, different methods can be considered: **(1)** to avoid the generation of unfeasible individuals into the crossover and mutation operations. This approach generates only feasible solutions, otherwise no results will be released; **(2)** to avoid the generation of unfeasible individuals into the crossover and mutation steps while trying to generate a minimum (or maximum) amount of feasible individual; **(3)** to exploit the *Criticality Index*. So, if the constraint is not satisfied, the cost function will assume a higher value that will drive the DSE towards better individuals. Limiting the processes allocation taking into account MC has two main effects: to increase the minimum cost and to decrease the maximum execution time, because the number of BBs instances will not be less than the number of criticality levels. Figure 3 shows a subset of solutions suggested by the DSE while considering different weights and timing requirements, with and without MC constraints. As expected, the Pareto set with no MC constraints (blue rhombuses more to the left) have solutions with a lower cost with respect to the one with MC constraints (orange squares). All the steps, prior to DSE one, have been executed in few minutes (on a high-end notebook). It is worth noting that this is a one-time effort, while the time for DSE steps depends on designers experience and number of considered constraints. Future works involve the introduction into the DSE also the concept of SW partitions in order to allow modeling also hypervisors technologies.

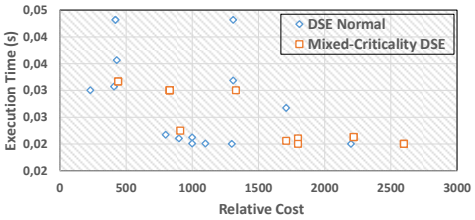

Figure 3: DSE small-set result.

ACKNOWLEDGMENTS

This work has been partially supported by the ECSEL RIA 2016 MegaM@Rt2 and AQUAS projects.

REFERENCES

[1] S. Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In 28th IEEE International Real-Time Systems Symposium (RTSS). pages 239-243. 2007
[2] A. Burns, R. I. Davis. Mixed Criticality Systems - A Review. Research report, University of York, 2014.
[3] J. Teich. Hardware/Software Codesign: The Past, the Present, and Predicting the Future. In Proceedings of the IEEE, 100:1411-1430, 2012
[4] F. Herrera, P. Pablo, and V. Eugenio. A model-based, single-source approach to design-space exploration and synthesis of mixed-criticality systems. Proceedings of the 18th International Workshop on Software and Compilers for Embedded Systems. pages 88-91, ACM, 2015.
[5] K. Rosvall, N. Khalilzad, G. Ungureanu, and I. Sander. Throughput Propagation in Constraint-Based Design Space Exploration for Mixed-Criticality Systems. In Proceedings of the 9th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO '17). ACM, New York, NY, USA, 2017.
[6] Hepsycode: A System-Level Methodology for HW/SW Co-Design of Heterogeneous Parallel Dedicated Systems, www.hepsycode.com
[7] L. Pomante, System-level design space exploration for dedicated heterogeneous multi-processor systems. Conference on Application-specific Systems, Architectures and Processors (ASAP), pages 79-86, 2011.
[8] D. Di Pompeo, E. Incerto, V. Muttillo, L. Pomante, and G. Valente. An Efficient Performance-Driven Approach for HW/SW Co-Design. In Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE '17). pages 323-326, ACM, 2017
[9] C. A. R. Hoare. Communicating sequential processes. Springer, New York, NY, 413-443. 1978
[10] L. Pomante, D. Sciuto, F. Salice, W. Fornaciari, and C. Brandolese. Affinity-Driven System Design Exploration for Heterogeneous Multiprocessor SoC. In IEEE Transactions on Computers, 55(5):508-519, 2006
[11] V. Muttillo, G. Valente, D. Ciambrone, V. Stoico, and L. Pomante. HEPSYCODE-RT: a Real-Time Extension for an ESL HW/SW Co-Design Methodology. In Proceedings of the 10th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO '18). ACM, New York, NY, USA, 2018