Injecting Hypervisor-based Software Partitions into Design Space Exploration Activities considering Mixed-Criticality Requirements

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

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

***Abstract*—This work faces the role of HW/SW Design Space Exploration for heterogeneous parallel embedded systems subject to mixed-criticality requirements, extended to consider also hypervisor technologies. In particular, it presents an evolutionary approach integrated into a reference Electronic System-Level HW/SW Co-Design flow able to consider and evaluate design alternatives while exploiting also Hypervisor-based SW partitions. Finally, some experimental results show the effectiveness of the proposed approach.**

***Index Terms*—HW/SW Co-Design, Heterogeneous Parallel Embedded Systems, Design Space Exploration, Mixed-Criticality Systems, Hypervisor technologies**

I. Introduction

In recent years, there has been a growing trend in exploiting (heterogeneous) multi-processor/core (i.e. parallel) platforms to execute embedded applications with different levels of criticality (i.e. Mixed- Criticality Embedded Systems, MCES). However, allowing these applications to safely coexist and possibly interact on the same platform becomes a very complex task that poses also several challenges from the implementation point of view [1]. In fact, embedded applications with different criticality levels can be allocated on different dedicated HW components or on different software partitions hosted on one or more shared HW components. The complexity of identifying the best architecture and mapping, especially when considering heterogeneous scenarios, is so high that heuristics Design Space Exploration (DSE) approaches are needed to help designers to identify a solution able to satisfy the requirements.

In such a context, the purpose of this work is to present a design space exploration step, integrated into an Electronic System-Level (ESL) HW/SW Co-Design framework, to support the development of heterogeneous parallel MCES, extended to consider also Hypervisor (HPV) technologies [2] and related software partitions concept. The remainder of the paper is organized as follows: Section II presents related works that consider mixed-critical requirements into the whole design flow. Section III describes the adopted design flow, while Section IV presents the main features of the proposed DSE approach. Then, Section V analyzes experimental results. Finally, Section VI closes the paper with some conclusions and future works description.

II. Design Space Exploration for Safety Critical Applications

In the last few years, a growing trend in the embedded systems domain is to run multiple embedded applications with different levels of criticality on a shared hardware platform, where the criticality of an application is an indication of the required level of "assurance" both from safety and security points of view. In such a context, the most critical development steps are related to the *System Specification* and the *Design Space Exploration* activities [3] and the main differences among the various works in the literature are mainly related to the different amount of information and actions that explicitly rely on the designer experience. For example, AUTOFOCUS3 [4] proposes a model-based development process at different levels of abstraction introducing safety-oriented constraints associated to computing components. The tool assigns the levels of criticality to application tasks and computing resources, avoiding the allocation of high-criticality tasks to low-criticality resources. CONTREP (CONTREX Eclipse plug-in, [5]) is a framework supporting UML/MARTE based modeling, analysis and design of mixed-criticality embedded systems. It is based on the CONTREX UML/MARTE modeling methodology [6] and considers safety constraints into the different design activities, integrating external tool like Multicube Explorer [7] for the DSE step. Finally, DeSyDe [8] provides a DSE tool for bare-metal applications, finding implementations for a set of tasks on a shared multi-processor platform starting from synchronous dataflow graphs (SDFGs), introducing MC requirement at scheduling level.

Considering the "software partition" concept, the work in [2] presents a state-of-the-art respect to HPV technologies into the embedded safety-critical system domain. A lot of HPVs have been developed to check and match certification requirements, avoiding interferences between partitions/tasks into a self-contained environment. In particular, PikeOS [9] provides a real-time operating system able to manage paravirtualization services, XtratuM [10] is a bare metal hypervisor

for paravirtualization and OKL4 Microvisor [11], developed by General Dynamics C4 Systems (formerly Open Kernel Labs), implements an advanced secure type-1 hypervisor, realizing a high performance Inter-Process Communication (IPC) message exchange mechanism.

In this context, this work proposes a DSE approach that is able to consider mixed-criticality issues into the development of heterogeneous parallel MCES, also exploiting HPV technologies. The main differences among the proposed approach and the previous works are related to the introduction of HPV software partitions (and virtualized environment) that allows to find cheaper (in terms of area/chip cost) and faster solutions (in term of timing, communication and concurrency performance), increasing the space of feasible final implementations. It is worth noting that none of the previous tools consider HPV solution in the design space step. A work that considers HPV methodology to identify a set of partitions, and that allocates applications to partitions is [12], but it considers fixed HW components and multi-core architecture, and it does not rely on DSE and direct (bare-metal) HW implementation, considering only HPV scenarios and not an hybrid environment and solutions involving also, for example, FPGA, DSP and so on. So, at the best of our knowledge, there are few works that introduce mixed-criticality issues directly into a HW/SW co-design flow, and there is a lack with respect the inclusion of software HPV into the whole design flow considering HW/SW Co-Design methodologies (in order to find the best sub-optimal solution in an early design stage).

III. HW/SW Co-Design Framework

In the context of MCES, this work adopts a specific Electronic System-Level HW/SW co-design flow (HEPSY-CODE: HW/SW Co-Design of Heterogeneous Parallel Dedicated Systems) [13], as shown in Fig. 1, based on an existing Methodology [14] (Fig. 1), while introducing Mixed Criticality (MC) requirements. The System Description step defines three reference models: Application, Partition, and Platform.

The Application Model exploits a behavioral modeling language, named HML (HEPSY Modeling Language) [15], based on CSP MoC [16]. By means of HML it is possible to specify the System Behavior Model (SBM), an executable model of the system behavior, a set of Non-Functional Constraints (NFCs) and a set of Reference Inputs (RIs) to be used for simulation-based activities. NFCs are composed of *Timing Constraints* (TCs), *Architectural Constraints* (ACs) and *Scheduling Directives* (SDs). In this work, the TC expressed by the designer is the *Time-to-Completion* (TTC) one. It is the time available to complete the SBM execution from the first input trigger to the complete output generation. ACs are related to the *Target Form Factor* (TFF) as *System On-chip* (SoC: ASIC or FPGA) or *System On-Board* (SoB: PCB) and to the *Target Template Architecture* (TTA) depending on the available *Basic Blocks* (BBs). Finally, SDs specify the available scheduling policies.

The partition model represents the HPV software partitions layer where, currently, only GPP are able to manage and exploit virtualization technologies (in future also ASP processors could be considered).

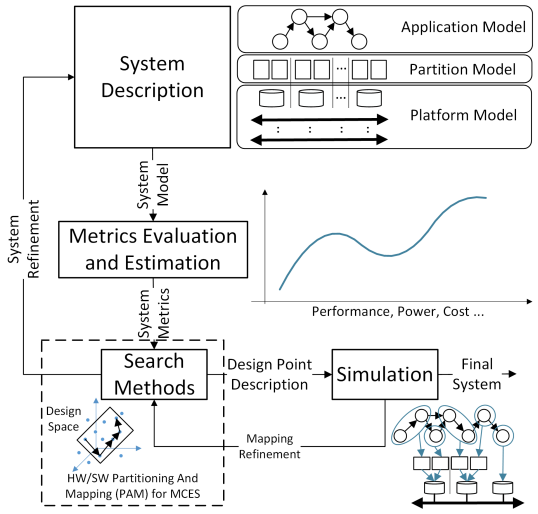


Fig. 1. HW/SW Co-Design Flow.

The Platform model defines the basic HW components available to build the final HW architecture. The target HW architecture is composed of different basic HW components. These components are collected into a *Technologies Library* (TL). TL can be considered as a generic "database" that provides the characterization of the available technologies. TL is composed by a set of *Processing Units* (PU), a set of *Memory Units* (MU) and a set of *Interconnection Links* (IL). However, the detailed characterizations depend from TFF. The main differences are related to the different attributes needed to characterize PU, MU, and IL. This work considers only TL for SOB where each PU that executes SW shall be a discrete *Commercial Off-The-Shelf* (COTS) *Integrated Circuit* (IC) mounted on a board [17].

The designer uses such components to build a set of *Basic Blocks* (BB) available during DSE step to automatically define the HW architecture. A generic BB is composed of a set of PU, a set of MU and a Communication Unit (CU). CU represents the set of IL that can be managed by a BB. BB internal architecture is dependent on TFF and TTA. The target HW architecture can be seen as a set of BB elements interconnected by means of one or more IL elements. The type of available BB is automatically defined by the selected TTA. This work focuses on *Heterogeneous Multi-Processor System with Distributed Memory* where each BB element is composed of only 1 PU element (possibly heterogeneous among BB elements), some local MU elements and 1 CU element. It is worth noting that the reference methodology is able to consider other TTA [18], but the current prototypal tools fully support only the one listed above.

The *Metrics Evaluation and Estimation* activities provide several metrics related to the BB involved in the design flow. This step aims at extracting as much as possible information about the system by analyzing the (*Application Model*) while considering the available BB (*Platform Model*)

and the use of Hypervisor technologies (*Partition Model*). This step is supported by *Co-Analysis* and *Co-Estimation* activities to evaluate/estimate several metrics related to the BB involved in the design flow. *Co-Analysis* performs evaluation of *Affinity* [19], *Concurrency* and *Communication* metrics [17]. *Co-Estimation* performs a *Static Estimation* of *Size*, and a *Dynamic Estimation* of *Load* [17].

After these steps, the reference co-design flow reaches the DSE one. Starting from *Application Model*, *Partition Model*, *Platform Model* and *HW/SW Partitioning And Mapping* (PAM) parameters associated to the evolutionary algorithm, it includes two iterative activities: "*Search Methods*", that provides HW/SW partitioning, mapping and architecture definition using a genetic algorithm that allows to explore the design space looking for feasible mapping/architecture items suitable to satisfy imposed constraints; "*Timing Co-Simulation*", that considers suggested mapping/architecture items to actually check for timing constraints satisfaction (Fig. 1).

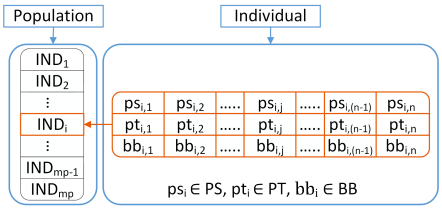


Fig. 2. Genetic Algorithm Individual Set.

IV. Mixed-Criticality Evolutionary Approach

The proposed DSE is based on a *Genetic Algorithm* (GA) used to optimize a multi-objective cost function that quantifies the quality of each individual of the GA population. In this context, the instance of an individual is defined as a matrix where the column index represents processes and the values represents BB instances and software partition elements, indicated as PT (if a BB contains GPP-type processors, 0 otherwise), as shown in Fig. 2.

The first metric considered is the ***Affinity Index***, that provides a quantification of the matching among the features of the functionality implemented by a process and the architectural features of each one of the following processor types: *GPP, DSP, SPP*. The higher the *Affinity* element value, the more suitable the corresponding processor type. The second metric is the ***Process Concurrency Index*** that provides information about how much processes pairs can be potentially concurrently "working". The third metric is the ***Process Communication Index*** that is expressed by the number of bits sent/received over each channel. Finally, the metric specifically introduced in [20] [21] and extended in this paper to consider software partition is the ***Criticality Index***, related to the criticality level associated to each process. In particular, defined the array $CRIT = \{[cr_1, cr_2, .., cr_j, .., cr_n] : cr_j$ is the integrity level associated to process $ps_j\}$, it is possible to define the *Criticality Index* as:

$$X_{CRIT_i} = \begin{cases} 1 & if cr_j - cr_k > 0 \wedge ps_j \in bb_x \wedge ps_k \in bb_y \wedge bb_x = bb_y \\ 1 & if cr_j - cr_k > 0 \wedge ps_j \in pt_j \in bb_x \wedge ps_k \in pt_k \in bb_y \wedge pt_j = pt_k \wedge bb_x = bb_y \\ 0 & otherwise \end{cases} \quad (1)$$

The goal behind this metric is to avoid having processes with different criticality levels on the same (shared) processor/core resource, using HPV technologies to fulfill MC requirements. The introduction of SW partition concept decreases the minimum cost with respect to no-partition scenario, because it is possible to use a number of BBs instances less than the number of criticality levels, increasing the number of feasible design solutions with respect to criticality requirements.

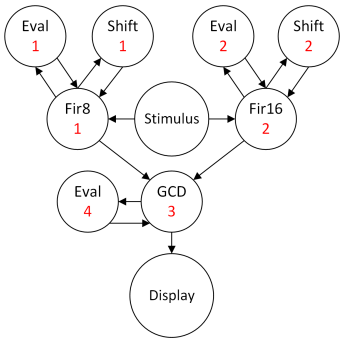


Fig. 3. Use Case Example: Fir-Fir-GCD.

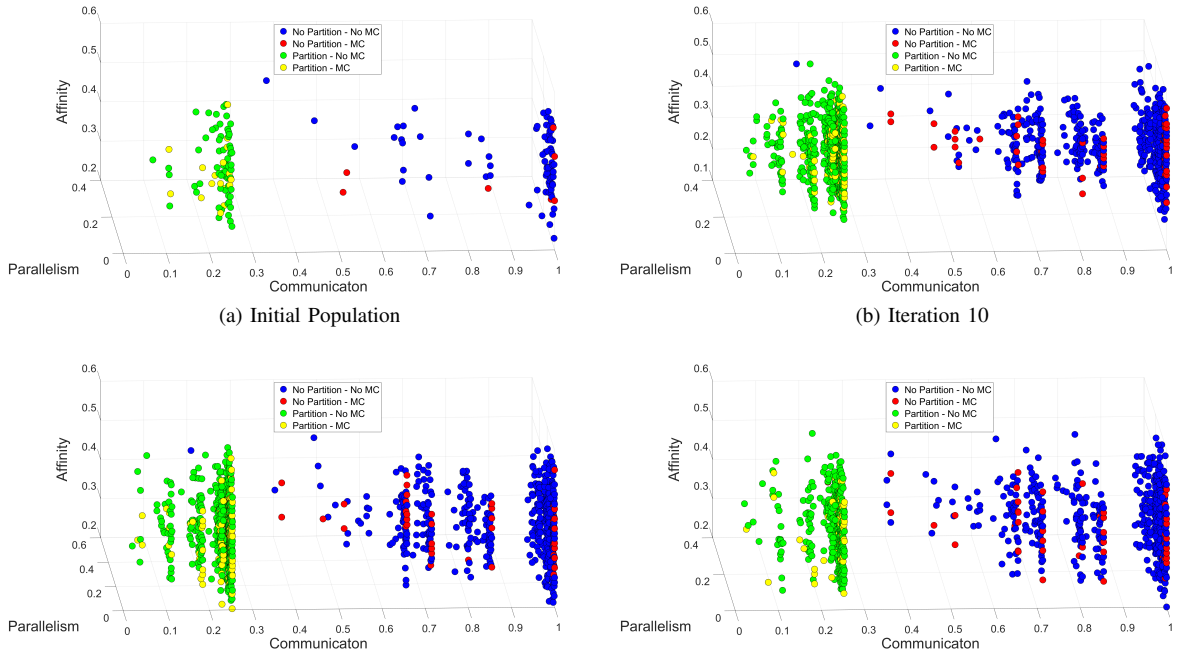
V. Validation

This section presents some results related to the DSE step. Table I shows the parameters setting. Considering AC, the maximum number of instances for each BB is 2, the maximum number of instances of BB considered into the DSE is equal to the number of processes (8) and BBs are supposed to communicate by means of a shared bus.

TABLE I
DSE Parameters Settings

Parameters	**Nr.**	**Values**
BBs	≤ 10	2 8051, 2 DSPIC, 2 LEON3, 2 Spartan3AN, 2 Virtex-7
SW Partition	≤ 4	HPV Partition
GA Selection	1	Random
GA Crossover (C)	1	One-Point
C probability (pc)	1	0.3
GA Mutation (M)	1	Random
M probability (pm)	1	0.1
Survival Selection (S)	1	Fitness-Based
S probability (ps)	1	0.15
Search Iteration (I)	40	-
Initial Population Size (P)	100	# Starting Individuals
Max Population Size (P)	≤ 1000	Max # Final Individuals

The Reference use case is shown in Fig. 3. The CSP model represents an application called Fir-Fir-GCD, that takes in input two values (triggered by Stimulus), makes two filtering actions (Fir8 and Fir16), makes the greatest common divisor (GCD), and then displays the results [22]. The red number



(a) Initial Population (b) Iteration 10

(c) Iteration 20 (d) Iteration 40

Fig. 4. Pareto Set results from the Design Space Exploration activities with respect to Affinity, Communication and Parallelism metrics. The different Pareto Points position and dispersion pattern depends on SBM application, on the number of processes/BBs/channels/criticality levels and on the specific iteration, weights values and reference inputs considered in the whole design flow.

TABLE II
DSE EXECUTION TIME RESULTS.

Iteration	No Partition - No MC			No Partition - MC			Partition - No MC			Partition - MC		
	ET[1] (μs)	# Sol.[2]	Com.[3]	ET[1] (μs)	# Sol.[2]	Com.[3]	ET[1] (μs)	# Sol.[2]	Com.[3]	ET[1] (μs)	# Sol.[2]	Com.[3]
Initial	0.17	-	-	0.16	-	-	0.33	-	-	0.24	-	-
1	8.77	89	0.9150	0.99	7	0.8426	18.3	82	0.2295	1.28	15	0.2220
5	9.29	267	0.9331	1.73	21	0.8565	19.1	218	0.2351	3.16	57	0.2354
10	19.8	941	0.9295	3.69	71	0.8183	39.4	684	0.2341	7.1	116	0.2376
20	30.7	949	0.9317	6.75	70	0.8680	59.7	939	0.2344	15.3	268	0.2277
40	56.3	913	0.9317	12.2	128	0.8215	102.0	864	0.2422	32.4	208	0.2391
Final	57.9	-	-	13.6	-	-	104.0	-	-	34.5	-	-

[1]ET: DSE Execution Time; [2]# Sol.: Number of feasible solution (from DSE); [3] Com.: average normalized value of the individual Communication indexes at each iteration.

under the name of each process represents the criticality level that has been associated to processes (the value has been assigned depending on the number of communicating channels and interactions among different processes). Fig. 4 shows some results from the DSE step, and how the DSE finds feasible solutions at each iteration step of Genetic Algorithm. Considering Affinity (provided by designer), Communication and Parallelism (taken from the Co-Analysis Activities) metrics, it is possible to note that the results, without considering partitions, are higher in Communication cost with respect to the situation with partitions. This is due to the possibility to find solutions and number of basic HW components less than the number of criticality levels, and also because the introduction of SW partitions offers the opportunity to allocate multiple processes on the same (partitioned) environment, so DSE can find solutions more suitable in terms of exchanged data with respect to the Non-SW-Partitions scenario. It is also possible to note that the Pareto Points follow a specific pattern (they are grouped into sub-sets that appear independent among each others). This behaviour could encourage the use of clustering methods into the GA steps in order to find different solutions, increasing diversity into individual generation/mutation activities. Table II shows results in term of DSE execution times. From this table it is worth noting that the MC scenarios take less time with respect to the classical non-MC scenarios. This is due to the reduced number of feasible solutions considered and the reduced number of total population size. With respect to SW partition one, the normal situation seems to be worst in terms of number of feasible (possible) solution found by DSE activity ($\simeq 50\%$). In terms of communication index, the SW partition scenario seem to be better in the average situation (as shown in Fig. 4). Other analysis related to cost area, execution

time and final (close to real scenarios) validation step will be studied in future works, but, starting from this results, considering communication, parallelism and affinity metrics, the introduction of SW partition (and HPV technologies) makes possible find solutions that behave better in term of exchanged data among processes. These solutions are not suitable in term of timing constraints, but they demonstrate how the introduction of SW partitions into DSE step improves results in terms of orthogonal metrics behavior (like communication and parallelism) and increases the number of feasible solutions by providing extended design space exploration opportunities while considering MC requirements.

VI. Conclusion and Future Work

This work has proposed a criticality-driven design space exploration for mixed-criticality heterogeneous parallel embedded systems, considering hypervisor technologies and SW partitions into the whole co-design flow. By introducing the criticality index into the evolutionary algorithm, the DSE is able to suggest solutions that fulfill constraints avoiding allocating applications with different levels of criticality on the same shared HW or SW resource. Results show that mixed-criticality solutions are typically less in term of number of feasible solutions with respect to a non-criticality scenario, and this work helps to partition processes into a heterogeneous parallel platform in a fast way. The introduction of SW partitions into the DSE step improves the number of feasible solutions (increasing diversity and avoiding to remain in a local minimum), and allows to find best solution in term of communication and parallelism allocation. The Pareto Set solution seems to follow some specific patterns (grouping each others at each iteration). Future works will analyze the GA behaviour, choosing and implementing different selection, mutation and crossover techniques (to increase diversity), and comparing results with other meta-heuristic algorithms (or introducing clustering and machine learning techniques in order to avoid unexpected behaviors), also considering the reduction of simulation time (to improve and realize a fast and efficient early co-design activity). Future works will also analyze cost and performance parameters to check the advantage of using SW partitions into the whole Co-Design Flow. Finally, SW partitions add more challenges into the scheduling activities in order to check and validate execution time by means of simulations (in fact, the final simulator shall be able to "emulate" the HPV scheduling policies, introducing a second level of scheduler for each SW partition, modeling as-much-as-possible the virtualized execution time, reducing simulation overheads).

Acknowledgment

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

References

[1] S. Baruah, H. Li and L. Stougie, "Towards the Design of Certifiable Mixed-criticality Systems", 2010 16th IEEE Real-Time and Embedded Technology and Applications Symposium, Stockholm, 2010, pp. 13-22.

[2] Z. Gu and Q. Zhao, "A State-of-the-Art Survey on Real-Time Issues in Embedded Systems Virtualization", Journal of Software Engineering and Applications, Vol. 5 No. 4, 2012, pp. 277-290.

[3] J. Teich, "Hardware/Software Codesign: The Past, the Present, and Predicting the Future", in Proceedings of the IEEE, vol. 100, no. Special Centennial Issue, 2012, pp. 1411-1430.

[4] S. Voss, J. Eder, F. Hölzl, "Design Space Exploration and its Visualizaion in AUTOFOCUS3", CEUR, 2014, pp. 57-66.

[5] F. Herrera, P. Peñil, E. Villar, "A model-based single-source approach to design-space exploration and synthesis of mixed-criticality systems", Proc. of the 18th Int. Workshop on Software and Compilers for Embedded Systems (SCOPES'15), 2015, pp. 88-91.

[6] Contrex project, https://contrex.offis.de/home/

[7] V. Zaccaria, G. Palermo, F. Castro, C. Silvano and G. Mariani, "Multicube Explorer: An Open Source Framework for Design Space Exploration of Chip Multi-Processors", 23th International Conference on Architecture of Computing Systems 2010, Hannover, Germany, 2010, pp. 1-7.

[8] K. Rosvall, N. Khalilzad, G. Ungureanu, and I. Sander, "Throughput Propagation in Constraint-Based Design Space Exploration for Mixed-Criticality Systems", Proceedings of the 9th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO '17). New York, NY, USA, 2017, pp. 1-8.

[9] PikeOS Hypervisor, https://www.sysgo.com/products/pikeos-hypervisor/

[10] M. Masmano, I. Ripoll, A. Crespo, and J. Metge, "XtratuM: a Hypervisor for Safety Critical Embedded Systems", Real-Time Linux Workshop, 2009.

[11] G. Heiser and B. Leslie, "The OKL4 Microvisor: Convergence point of microkernels and hypervisors", in Proceedings of the 1st Asia-Pacific Workshop on Systems (APSys), New Delhi, India, Aug. 2010, pp. 19-24.

[12] S. Trujillo, A. Crespo and A. Alonso, "MultiPARTES: Multicore Virtualization for Mixed-Criticality Systems", 2013 Euromicro Conference on Digital System Design, Los Alamitos, CA, 2013, pp. 260-265.

[13] Hepsycode: A System-Level Methodology for HW/SW Co-Design of Heterogeneous Parallel Dedicated Systems, www.hepsycode.com

[14] L. Pomante, "System-level design space exploration for dedicated heterogeneous multi-processor systems", ASAP 2011 - 22nd IEEE International Conference on Application-specific Systems, Architectures and Processors, Santa Monica, CA, 2011, pp. 79-86.

[15] D. Di Pompeo, E. Incerto, V. Muttillo, L. Pomante, and G. Valente, "An Efficient Performance-Driven Approach for HW/SW Co-Design", Proceedings of the 8th ACM/SPEC on International Conference on Performance Engineering (ICPE '17), ACM, New York, NY, USA, 2017, pp. 323-326.

[16] C. A. R. Hoare, "Communicating sequential processes", Springer, New York, NY, 1978, pp. 413-443.

[17] 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", Proceedings of the 10th Workshop on Rapid Simulation and Performance Evaluation: Methods and Tools (RAPIDO '18), ACM, New York, NY, USA, 2018.

[18] L. Pomante, "HW/SW co-design of dedicated heterogeneous parallel systems: an extended design space exploration approach", in IET Computers Digital Techniques, vol. 7, no. 6, pp. 246-254, November 2013.

[19] C. Brandolese, W. Fornaciari, L. Pomante, F. Salice and D. Sciuto, "Affinity-driven system design exploration for heterogeneous multiprocessor SoC", in IEEE Transactions on Computers, vol. 55, no. 5, pp. 508-519, May 2006.

[20] V. Muttillo, G. Valente and L. Pomante, "Criticality-aware Design Space Exploration for Mixed-Criticality Embedded Systems", In Proceedings of the 9th ACM/SPEC on International Conference on Performance Engineering (ICPE '18), ACM, New York, NY, USA, 2018.

[21] V. Muttillo, G. Valente, and L. Pomante, "Criticality-driven Design Space Exploration for Mixed-Criticality Heterogeneous Parallel Embedded Systems", In Proceedings of the 9th Workshop and 7th Workshop on Parallel Programming and RunTime Management Techniques for Manycore Architectures and Design Tools and Architectures for Multicore Embedded Computing Platforms (PARMA-DITAM '18). ACM, New York, NY, USA, 2018, pp. 63-68.

[22] L. Pomante and P. Serri, "SystemC-based HW/SW Co-Design of Heterogeneous Multiprocessor Dedicated Systems", International Journal of Information Systems, Volume 1, July 30, 2014.