

# Development of an interactive approach for MPSoC exploration

Sonda Chtourou <sup>1</sup>, Zied Ben Salem <sup>1,2</sup>, Mohamed-Wassim Youssef <sup>3</sup> and Mohamed Abid <sup>1</sup>, *Member, IEEE*

<sup>1</sup> CES Research Laboratory, National School of Engineers of Sfax, University of Sfax, Tunisia.

<sup>2</sup> ALPHA ENGINEERING Industrial Society, Tunis, Tunisia.

<sup>3</sup> Higher Institute of Computer Science, Tunis, Tunisia.

**Abstract**— One of the most curial issues in embedded systems is how to easily and quickly design MPSoC to get high-performance systems. We developed in this paper an interactive approach for MPSoC exploration at virtual prototyping level. This approach allows early hardware and software integration and validation in the design process and provides precise performance analysis. Exploration is performed through a graphical web-powered layer allowing fast and easy graphical exploration. The approach was validated using H264 encoder.

**Index Terms**— Interactive exploration, MPSoC, virtual platform, virtual prototyping level and H264 encoder.

## I. INTRODUCTION

Embedded systems use an increasing hardware parallelism (Multi-Processor System-on-Chip MPSoC) to deliver high computation and communication performances. The design complexity is exponentially increasing which opens up new challenges for hardware (HW) and software (SW) integration while maintaining tight time-to-market requirement. On one hand, classical design flow recommends designing HW and SW in parallel way. However, HW is always designed far too late in the product development cycle. Thus, SW can't be checked for early internships in the design cycle. This implies a huge gap between HW and SW integration. Virtual prototyping seems to be a good candidate to respond to this need. Indeed, the use of virtual prototyping simulators allows early design validation. On the other hand, embedded system SW is characterized by high-performance requirements and designer has to extract the best applications parallelism to be mapped on the best MPSoC architecture which meets constraints. Thus, designer needs interactive tool running on the top virtual platform with graphical interfaces to rapidly and easily explore others alternative solutions.

We propose in this paper an interactive approach for MPSoC exploration at virtual prototyping level. Using this approach, designer can make early HW and SW integration and validation in the design process. This approach provides also precise performance analysis and consequently helps

designer to take rapid and right decision to find optimal design meeting performance objectives. To achieve our goal, we proposed an approach to estimate MPSoC performance and we described an interactive technique of exploration based on different HW and SW variations. The major contribution of this work is that we enabled exploration through a graphical web layer in order to make the technique of exploration interactive. As case study, we realized experimentation with H264 encoder benchmark to prove the concept of the proposed approach. The experiment shows that the proposed environment permits to handle the complexity and flexibility needed in the development of MPSoCs and the early selection of the best system meeting constraints.

This paper is organized into four sections. Section II presents the problem statement. Section III details the development of an interactive approach for exploration of MPSoC at virtual prototyping level. Different results and interpretations of exploration of the parallelized H264 encoder mapped on MPSoC are discussed in Section IV.

## II. PROBLEM STATEMENT AND RELATED WORKS

The major problem faced when designing MPSoC is the difficulty and complexity to interface the HW and SW. SW validation process depends on several important details like drivers, memories, CPUs and communication. Thus, SW can be validated only on ready-to-use model of the HW platform. In addition, HW is always far too late in the product development cycle. This leads to an outstanding cycle design. The development of HW and SW components in parallel way may lead to incompatibilities and make unfeasible the early check of SW. Virtual prototyping represents a good alternative to respond this problem by allowing early design integration. Several frameworks and environments using in backend virtual prototyping platform were developed in order to speed up exploration of MPSoC design. [1] and [2] propose frameworks based on SystemC allowing higher simulation speed with early performance estimation. However, SystemC is HW-oriented language and it is not the standard language to design complex

applications at algorithm level. Simulink and UML are also used inside frameworks for high level modeling. [3] and [4] propose frameworks enabling mixed HW/SW refinement and opening new facilities for exploration. However, embedded programmers are still reluctant to adopt Simulink models because it quickly becomes unsuitable with complex designs. Qemu was also used as the foundation for Yeh Tse-Chen projects [5]. These projects show that the combination of Qemu and SystemC can make the co-simulation at the cycle-accurate level extremely fast. However, Qemu shows lack of flexibility to configure specific HW architectures and supports only homogeneous designs. Meanwhile, OVP allows simulating complex heterogeneous and homogeneous architectures at high simulation performance which can be measured by hundreds of million instructions per second (MIPS) [6]. Besides, OVP provides a large set of free open sources and standard models components providing more flexibility to configure complex HW system. OVP uses the same level of simulation but also tackles the simulation speed problem by using advanced techniques for SW simulation such as OS simulation for the former and code morphing for the latter. From the above mentioned raisons, OVP has been embraced as a primary solution to make easier system description (SW and HW integration) and faster simulation [7]. However, it uses scripts and textual-language description to integrate HW and SW parts. Therefore, its specification complexity rises with the rise of virtual platform complexity. This becomes a new bottleneck in the HW/SW integration and in the exploration of the best design meeting constraints. Indeed, designer needs interactive graphical interfaces running on the top of OVP to easily explore others solutions. In this context, we didn't find any work or environment with graphical layer allowing interactive exploration. Our contribution is to design a higher graphical layer on the top of OVP to allow an easy graphical interactive exploration.

### III. DEVELOPMENT OF AN INTERACTIVE APPROACH FOR MPSoC EXPLORATION AT VIRTUAL PROTOTYPING LEVEL

#### A. General proposed workflow

To define an interactive approach of exploration, we proposed a workflow handling different needed milestones to design system meeting performance constraints at early stage:

- Phase 1: partitioning and mapping: select a solution of partitioning and mapping tasks.
- Phase 2: configuration of the design: this phase contains two sub-actions achieved in parallel way:
  - configuration of SW prototype: select different options and features (OS, application), then generate SW prototype;
  - configuration of the virtual MPSoC HW prototype: instantiate different needed compounds and interconnections with OVP, then generate virtual HW prototype.
- Phase 3: integration: integrate and simulate SW and virtual HW prototypes with OVP to make early integration;
- Phase 4: performance estimation: use the defined approach to estimate performance of the configured system;

- Phase 5: interactive technique of exploration: allow designer to explore various integration alternatives to best satisfy defined performance requirements.

Finally, designer can proceed to the final integration.

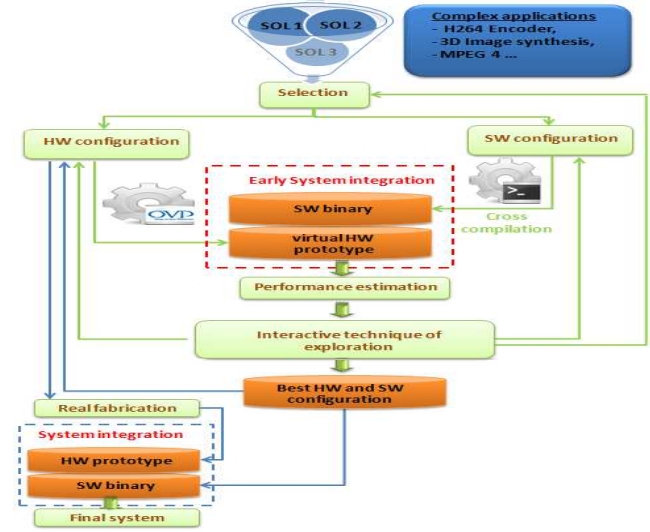


Fig. 1. Proposed workflow of interactive exploration.

#### B. Define an approach to estimate performance of MPSoC

To define an approach of performance estimation (phase 4), we have to fix performance metrics. In our project, we proposed HW and SW performance metrics. The designer can select or combine proposed metrics, evaluate each configured system and make decision.

- *Used hardware resources:* Percentage of memory (%MEM) used by the process from the available physical memory and percentage of total CPU time (%CPU) that the process consumes from the elapsed CPU time.
- *Average turnaround time:* Time elapsed by the whole application with the configured HW system depicts a widely used constraint to evaluate performance of the system.
- *Quality of Service:* The QoS provided by the system depicts an important aspect in SoC design. The QoS mainly depends of the target application.

#### C. Define an interactive technique of exploration

We proposed a technique of exploration (phase 5) to explore other solution alternatives. This technique is based on several HW and SW variations which mainly affect the performance of the system. Designer can modify the number of target CPUs and the SW parallelization levels. He can also adjust the number of MIPS and the architecture of target CPUs. Distribution, type and size of used memory or SW parameter setting can be also revised. Based on results, he can improve some parts of the design to satisfy performance requirement. We developed graphical user interfaces to make the proposed technique of exploration interactive and to control the different phases of the proposed approach. Then, we implemented different scripts automating several backend functions of these interfaces (configuration, compilation and generation of the SW, integration of configured HW and SW with OVP, storage of generated system in database, start up of

configured system). Storing configured system in database allows designer to save effort to modify configured design without repeating all steps.

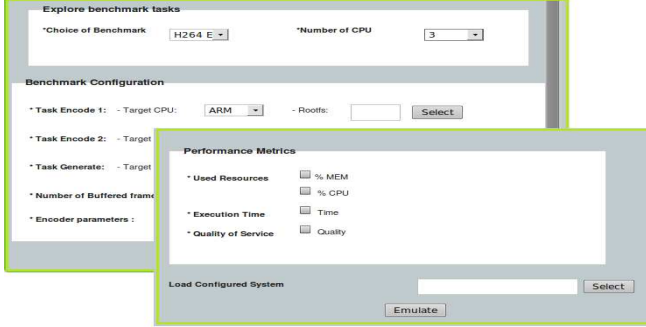


Fig. 2. Proposed graphical interfaces.

#### IV. TESTS AND RESULTS

In order to evaluate our methodology, we performed exploration with H264 encoder benchmark.

##### A. Mapping H264 on MPSoC

To parallelize H264, we split encoding the video between two processors. Each processor encodes each following buffered frames as if we have a video coming from a camera. After encoding each buffered frames, each processor puts the encoded frames in a shared memory. Then, a third processor reads encoded frames from the shared memory and generates the encoded video. For the parallelized H264, we configured with OVP an MPSoC which contains 3 sub-systems (minimal design which boots Linux) based on ARM processor.

##### B. Methodology of exploration

The performance objective of our case study is to design a system which runs parallelized H264 with minimal time and HW resources while keeping an acceptable QoS. As we mentioned in (III.B), QoS performance metrics depend mainly on target application. As we chose H264 codec, it is interesting to measure the quality of the encoded video because it is the most important characteristics of a codec. PSNR is the most widely used video quality metric [8].

##### C. Exploration of parallelized H264

###### 1) Impact of variation of the number of buffered frames

In this test, we used 4 different numbers of buffered frames (10, 15, 20 and 25). Table 1 shows effect of this variation on used memory. We remark that each processor uses a different amount of memory resources. This fact is due to the different number of load of buffered frames to encode the following buffered frames. Indeed, loading the buffered frames consumes considerable rate of memory.

TABLE I  
EFFECT OF VARYING THE NUMBER OF BUFFERED FRAMES.

Buffered frames	Processor 1	Processor 2	Processor 3
10	39.12%	26.22%	27.86%
15	31.03%	26.22%	22.53%
20	31.03%	18.10%	22.53%
25	20.26%	21.60%	15%

###### 2) Impact of variation of the encoder parameters

We used different presets (H264 encoder parameters). We operated with Bitrate/Quality by using different bitrates. H264 includes several profiles and preset options. Table II shows results of used presets for 25 buffered frames. Depending on what the user needs for his payload device, he can make a fast comparison. Thus, all conclusions ("better", "worse", "faster") will be made from this point of view.

TABLE III  
EFFECT OF VARYING OF ENCODER PARAMETERS

	PSNR	Time
Bitrate 100	29.215 (kb/s)	51.30 (s)
Bitrate 500	35.236 (kb/s)	70.02 (s)
Bitrate 1800	38.305 (kb/s)	113.10 (s)
Profile Main	36.113 (kb/s)	79.80 (s)
Profile High	36.126 (kb/s)	78.20 (s)
Profile Baseline	36.100 (kb/s)	77.76 (s)
Preset veryfast	35.863 (kb/s)	49.60 (s)
VBV	35.563 (kb/s)	82.35 (s)

###### 3) Exploration results

It is now easy to select the best system configuration satisfying the fixed performance objectives. Through previous exploration, we have to fix the number of buffered frame to 25 and preset "veryfast" in order to use the minimal HW resources and time while keeping an acceptable QoS.

#### V. CONCLUSION

We proposed in this paper an interactive exploration approach of MPSoC at virtual prototyping level. This approach helps designer to take the right decision to reach optimal design meeting performance objectives at an early stage in the design process of MPSoC.

#### REFERENCES

- [1] Gerin, P. "Flexible and Executable Hardware/Software Interface Modeling For Multiprocessor SoC Design Using SystemC", ASP-DAC '07: Proceedings of the 2007 Asia and South Pacific Design Automation Conference, 2007, 390-395.
- [2] Mello, A. "Parallel Simulation of SystemC TLM 2.0 Compliant MPSoC on SMP Workstations", DATE '10: Proceedings of the Conference on Design, Automation and Test in Europe, 2010.
- [3] Popovici, K. "Simulink based hardware-software codesign flow for heterogeneous MPSoC", SCSC'07: Proceedings of the summer computer simulation conference, 2007, 497-504.
- [4] Popovici, K. "Mixed Hardware Software Multilevel Modeling and Simulation for Multithreaded Heterogeneous MPSoC", VLSI-DAT'07: VLSI Design, Automation and Test, 2007, 1 - 4.
- [5] Chen Yeh, T. "On the interface between QEMU and SystemC for hardware modeling", ISNE'10: International Symposium on Next-Generation Electronics, 2010, 73.
- [6] <http://www.ovpworld.org/technology.php#sectionFOS>.
- [7] Ben Atitallah, R. "A fast MPSoC virtual prototyping for intensive signal processing applications", MICPRO: Microprocessors and Microsystems Embedded Hardware Design Journal, 176-189.
- [8] Wang, Y. "Survey of Objective Video Quality Measurements", WWIC'10: Proceedings of the 8th international conference on Wired/Wireless Internet Communications, 2010, 240-251.