2021 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE OCTOBER 26-27, 2021

Unified Model/Hardware-in-the-Loop Methodology for Mixed-Signal System Design and Hardware Prototyping

Martin Barnasconi, Wil Kitzen, Thieu Lammers, NXP Semiconductors, Eindhoven, The Netherlands

Paul Ehrlich, Karsten Einwich, COSEDA Technologies GmbH, Dresden, Germany

***Abstract*— The growing complexity of mixed-signal electronic systems requires more sophisticated design and verification methodologies to remain competitive in the market. To meet the time-to-market, product performance, robustness and quality requirements, innovative design and verification technologies are essential to boost design productivity and efficiency. This paper introduces a unified Model-in-the-Loop (MiL) and Hardware-in-the-Loop (HiL) methodology to unite the system-level design phase with the hardware prototyping phase, to enable early verification and validation of the product concept by combining system simulation and hardware prototyping technologies. The presented unified MiL/HiL methodology is based on the standardized languages SystemC (IEEE Std. 1666) and SystemC-AMS (IEEE Std. 1666.1), which offer a versatile and flexible flow to seamlessly integrate the SystemC-based system modeling and simulation environment with the hardware prototyping domain. The implementation is based on the application of a cost-effective and off-the-shelf hybrid CPU/FPGA-based hardware development board, which executes SystemC and/or SystemC-AMS models in real time, combined with synthesized RTL which is mapped on the FPGA. The methodology has been successfully demonstrated in a demanding mixed-signal power electronics application, combining hardware representing the power electronics application with the device-under-test running in real-time on the CPU and FPGA.**

Keywords—Hardware-in-the-loop; hardware prototyping; Model-in-the-loop; system-level design; system simulation; SystemC; SystemC-AMS;

I. INTRODUCTION

As part of the product development of mixed-signal electronic systems, interaction with the physical application environment is essential to incorporate “real-life” stimuli and conditions to early verify and validate the device-under-test (DUT) in terms of compliancy to the requirements specification, functionality and performance. For this, the application environment of the mixed-signal system must be incorporated in simulation, along with a system model representing the DUT. The Model-in-the-Loop (MiL) methodology uses system models and simulation to analyze the interaction and dependencies between the DUT and the physical environment when connected in a loop, as depicted in Figure 1a. This MiL approach is applied in the early phases of product development, to verify the product specification and validate the product concept development.

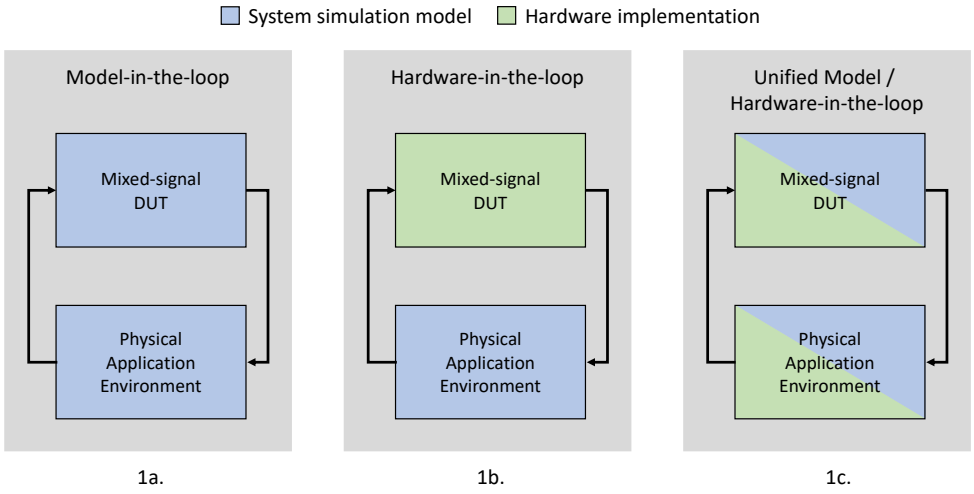

Figure 1. Model-in-the-Loop (MiL), Hardware-in-the-Loop (HiL), and unified MiL/HiL methodology

However, with system modeling and simulation not all product characteristics can be assessed, and therefore there is a need to include the real hardware of the DUT in the loop. This methodology is called Hardware-in-the-Loop (HiL), as depicted in Figure 1b. Although not depicted in this figure, the HiL approach could also capture the application environment in hardware, and the DUT system model in simulation.

This paper presents a novel unified MiL/HiL methodology and flow, as shown in Figure 1c, where the system model and simulation environment can be incorporated in the hardware environment, enabling a mixed simulation/hardware setup. This is essential for mixed-signal applications, as traditional HiL approaches are often limited to include digital functionality only, due to the limitations of FPGA-based solutions, which cannot implement analog behavior. The novelty in this methodology is to use the CPU on the hardware development board to run the actual SystemC and SystemC-AMS simulation in real-time for the analog/mixed-signal functionality, combined with the FGPA for the synthesized digital functions.

The paper is organized as follows. Section II describes the related work in the area of MiL and/or HiL methodologies and flows. Section III details the developed unified MiL/HiL methodology and flow, followed by the proof-of-concept and hardware demonstrator shown in Section IV including simulation and measurements results.

II. Related Work

HiL approaches are used for decades. Several established and dedicated hardware platforms are commercially available. HiL simulation is mainly used in system companies like automotive Tier1 supplier and OEM's [1], [2].

HiL simulation for design at circuit level is not state-of-the-art. For circuit level design of digital systems, FPGA-based prototyping is widespread. Different vendors provide dedicated hard-/software solutions. The limitations of the FPGA-based approach are that the design must be at synthesizable RTL level and thus quasi ready, an interaction with analog components or the analog environment cannot be realized, the effort to get the system running at the FPGA is high, and also design changes and design exploration is very costly. State-of-the-art is also the usage of test chips or previous generation circuits connected to the FPGA prototype for the analog components. This approach does not solve the issue of the late availability and does not allow design space exploration.

To overcome the limitation of the missing interaction with analog components, different approaches have been publicized [3], [4]. These approaches convert the analog components into digital functions by using digital filter design techniques. Those approaches require a special transformation of the analog components and have limitations regarding the analog components and effects which can be mapped. Also, the effort for this transformation is significant which also limits design changes and the exploration of different variants. There are also previous activities to use HiL simulation for verification [5], [6]. Especially the requirements for the turn-around time and processor performance are relaxed for the verification use case.

There is no approach known for the usage of a unified MiL/HiL simulation for mixed-signal circuit design.

III. Design Methodology and Flow

To implement the unified MiL/HiL methodology, the following concepts and technologies have been combined into a seamless flow:

- Using SystemC and SystemC-AMS to create a system-level model of the mixed-signal DUT.
- Using high-level synthesis and logic synthesis to map the digital (SystemC) functionality onto an FPGA in the hardware environment
- Porting the SystemC and SystemC-AMS simulation kernel to an ARM based embedded system in the hardware environment
- Using a cost-effective hybrid CPU+FPGA-based hardware development board which can simulate the AMS system model in real-time, combined with the digital synthesized gate-level description in the FPGA.

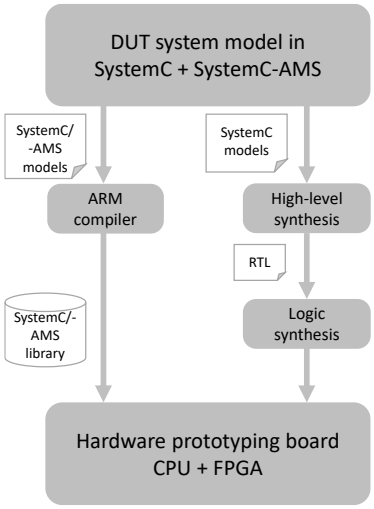

Figure 2. Unified MiL/HiL design flow

The flow is presented in Figure 2. It shows two trajectories from DUT system model to the hardware prototyping board. The left branch is applied for the SystemC and SystemC-AMS models which will be executed on the ARM processor. The right branch shows the high-level synthesis and logic synthesis path to map functionality onto the FPGA. The next section will give further details on these different steps.

A. System-level model in SystemC and SystemC-AMS

Figure 3 shows the design hierarchy and system-level models of the power electronics application. The top-level testbench in SystemC instantiates the fly-back converter, sources and DUT. The COSIDE [7] Design Environment has been used to create schematics of the system-level models using build-in library components in SystemC or SystemC-AMS. Where applicable, user-defined SystemC or SystemC-AMS models have been created.

The application of the fly-back converter is purely analog (see also Figure 5) and modelled with SystemC-AMS, using Timed Data Flow (TDF) models. Also the sources and load in the application are represented by TDF models.

The fly-back controller is a mixed-signal integrated circuit represented as schematic, which contains the digital control (valley detector and gate control) modelled in SystemC, as well as the analog PI filter modelled in SystemC-AMS using ELN primitives. The clock and reset signals are supplied to the fly-back controller via the top-level testbench.

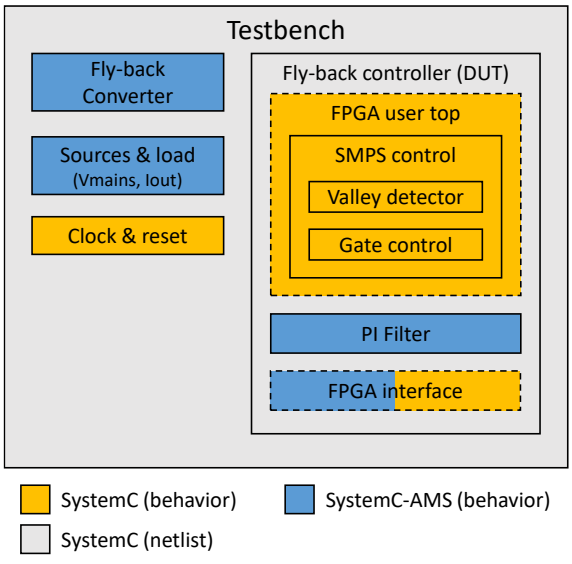

Figure 3. Design hierarchy and system-level models of the power electronics application

An extra design hierarchy called “FPGA user top” has been created to combine all digital functions which will be mapped onto the FPGA. Dedicated FPGA interface elements are made available in the COSIDE tool to interface between the CPU and FPGA.

B. Application of high-level synthesis (HLS) and logic synthesis to map the SystemC functions onto an FPGA

The digital functions (valley detector and gate control) as shown in Figure 3 have been mapped onto the FPGA. To this purpose, both functions are made HLS compliant by following the language constraints as defined in the SystemC synthesizable subset. In addition, floating point data types were translated into fixed-point data type (`sc_fixed`). The COSIDE tool supports build-script generation for the HLS flow using the Siemens Catapult HLS tool [8]. The output of the HLS flow results in RTL representations of the digital functions, which are synthesized to a gate-level description using the Xilinx Vivado Logic Synthesis tool [9].

C. Porting the SystemC and SystemC-AMS simulation kernel to ARM

To run the system model written in the SystemC and SystemC-AMS language, it has to be compiled for the target processor of the development board or more specific for an ARM A9 core in our example. As both the model as well as the libraries are written in C++ all recent compilers (> C++11) are sufficient to compile SystemC-AMS and the required libraries as well as the model.

In general, this leads to two possibilities on how to run the model on the ARM: Bare metal or via an operating system. However, since SystemC has dependencies to operating system layer, e.g., on the memory management, only the latter is feasible with reasonable amount of effort. This decision also leads to other benefits regarding the simulation control and evaluation, which come along with an OS such as driver for the ethernet stack or the SD card. The OS should be a real time or real time patched kernel to be preemptive, this ensures that the OS returns to the timing critical simulation as predictable as possible if the scheduler switches to a kernel task. The use of an OS also allows to conveniently cross compile the platform in an IDE such as COSIDE and transfer then the model onto the hardware.

When porting SystemC and SystemC-AMS to the ARM architecture, special attention regarding the resulting simulation speed should be paid on the floating-point calculation support via an external floating-point unit and mapping of SystemC thread to OS thread. The SystemC kernel uses threads to map the parallel activity within the model. Those SystemC threads never run in parallel and those do not need all typical threading features. Therefore, SystemC provides a light weight Quick Thread implementation for common platforms to avoid the mapping onto full blown system thread libraries like the pthreads under Linux. While the floating-point issue is a hardware requirement and a compiler option the threading should be done specifically for each hardware, to avoid the fallback onto slow system thread libraries. In our example we ported the assembler based SystemC QT threads to the ARM-v7 architecture.

D. Selecting an CPU+FPGA-based hardware development board

The requirements for the hardware development board can be split into different key components of the whole setup: FPGA, processor, FPGA-processor interconnect, the I/O to external hardware ant the connections towards to a control computer see Figure 4.

The FPGA will accommodate the synthesized part of the model and therefore requires sufficient I/O to the processor as well as to the external hardware, enough logic cells/gates to hold the synthesized model and speed/frequency to meet timing of the most critical path of it. The processor runs the model, usually the testbench and the analog parts of the design, which cannot be synthesized. The main processor requirement is its speed. Together with the interconnect between processor and FPGA, which is a tightly cache-coupled AXI connection, it determines the feasibility of the closed simulation loop between FPGA and processor model parts. Not only have embedded processors usually a slower core frequency but also have a comparable slower architecture (pipeline/cache/flops) compared to x86 based computers. Another important requirement is the need for a floating-point unit, as especially the analog computations benefit from it. The IO interface towards the external hardware has to have enough IO with sufficient slew rate for the required interface. Also, level shifters might be needed as external hardware and the FPGA IO might have different voltage levels.

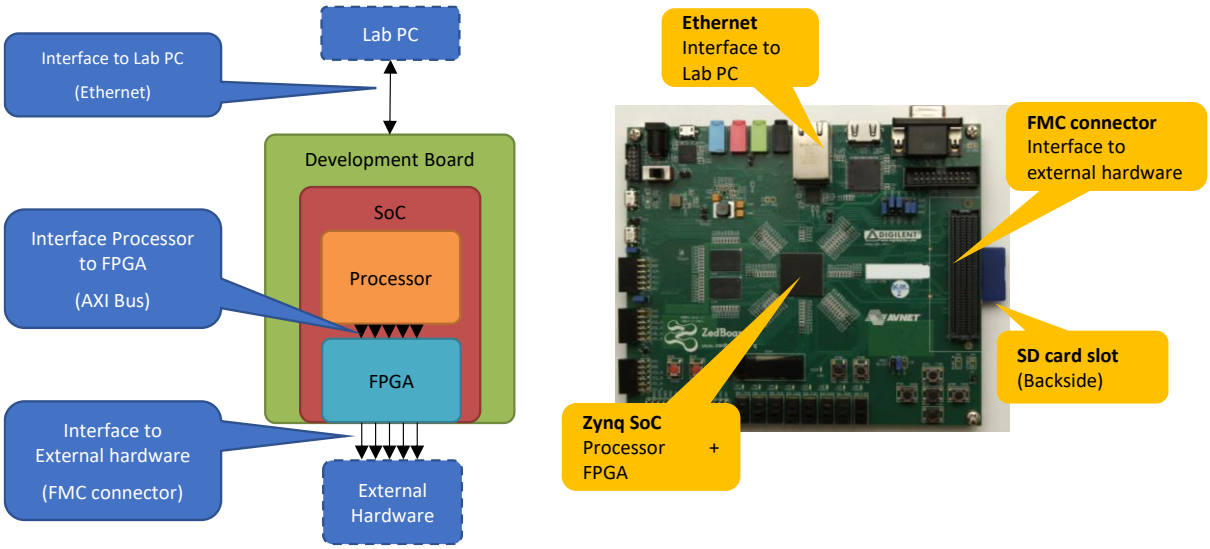

Figure 4. Hardware development general board overview and chosen ZedBoard

All those requirements are strongly application dependent, along with soft requirements such as cost and availability. Additionally, we considered the presence of an SD writer and an ethernet port as required for easy simulation control and evaluation. The SD card is used to store simulation traces and the Ethernet port is used to retrieve them as well as to deploy the model itself.

For our use case we selected the ZedBoard [10] which features a ZynQ-7000 SoC including a dual ARM A9 as well as FPGA fabric with 85.000 logic cells. The SoC provides tight coupling between processor and FPGA and offers a lot of IO like the FMC connector to hook up to further daughter boards. It features the required interfaces with an GBit Ethernet stack as well as an SD card.

IV. PROOF-OF-CONCEPT AND HARDWARE DEMONSTRATOR

The proposed methodology and flow has been validated on a mixed-signal power electronics application as shown in Figure 5. It shows the fly-back converter hardware and the DUT acting as fly-back controller.

The power electronics application was available as a custom hardware board. The DUT functionality has been mapped on the CPU+FPGA-based hardware development board and this board is connected to the application board via external interfaces (gate driver and AD converters).

The unified MiL/HiL implementation showed that the regulation loop in the application imposes strict (real-time) timing constraints to guarantee a correct, stable and safe mode of operation. This means special care should be taken when mapping the system model to the CPU+FPGA-based hardware development board, to make sure real-time timing constraints can be fulfilled. The advantage of the hybrid board was to ‘offload’ timing critical (digital) functionality to the FPGA.

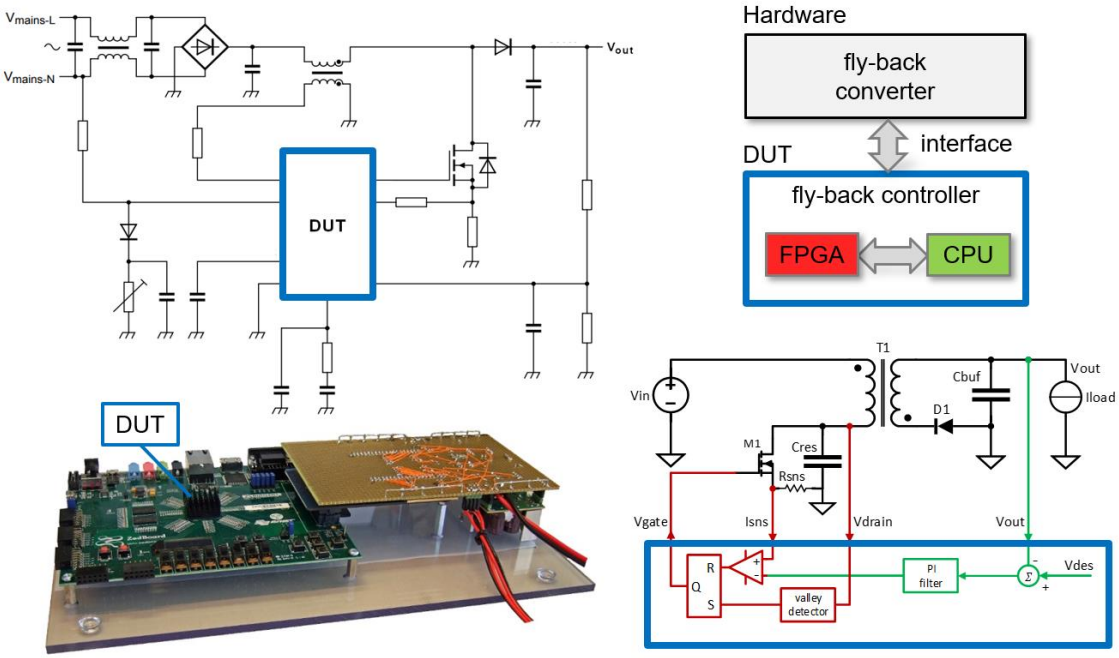

Figure 5. Application diagram of power adapter hardware demonstrator

The switching frequency of the fly-back converter is ~100kHz. The bandwidth of the PI filter is ~10 kHz, which is a factor of 10 lower than the switching frequency. The valley detector measures the transformer demagnetization sinusoidal waveform (V_{drain}), acting at a resonance frequency of ~2MHz, to switch on the mos-fet at the lowest voltage (valley). The set-reset flip-flop is part of the gate driver with a latency requirement of less than 20 ns.

The application and hardware set-up successfully demonstrated the real-time execution of SystemC and SystemC-AMS models in an embedded system, combined with digital functions running on the integrated FPGA. In Figure 6, a comparison is made between the measured (left side) and simulated (right side) mos-fet drain voltage ($V_{d,prim}$), the current through the mos-fet (V_{isns}) and the gate switching voltage (V_{gate}). Here, the mos-fet switches on at the first detected valley during the demagnetization phase. This happens when $V_{d,prim}$ reaches the lowest point of the half cosine voltage. During the mos-fet conduction phase, the current linearly rises to a level, controlled by the PI-filter, before switching off. The control loop achieves a balance, such that the exact amount of energy is transferred to the secondary side of the transformer to deliver 3A at 12V output. Also note that the ringing effects, visible at the start of $V_{d,prim}$ and at the falling edge of V_{isns}, are captured quite well by the simulation.

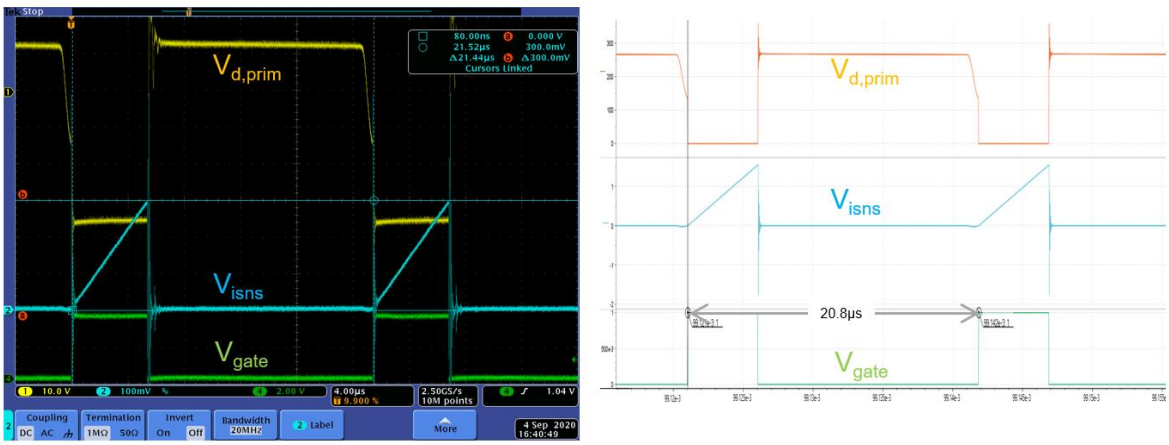

Figure 6. Closed loop measurements and associated signal traces from simulation at 12V, 3A output

V. CONCLUSIONS

This paper presented a unified Model-in-the-Loop (MiL) and Hardware-in-the-Loop (HiL) methodology where a mixed-signal DUT is implemented by using a ZynQ-7000 based hardware development board, which executes a SystemC-AMS model on its processor (dual core ARM A9) and uses a FPGA for the synthesized digital functions.

It has been demonstrated that the SystemC-AMS model and FPGA implementation can the used for closed loop measurements and are able to keep up with real-time timing constraints for a power electronics applications. Note that special care has to be taken to match the hardware development requirements, especially the processor capabilities and real-time OS constraints, with the intended application to guarantee a correct, stable and safe mode of operation. In terms of evaluated functionality and mixed-signal signal characteristics, the system simulation results and actual measurements are proven to be consistent and equivalent.

The application of these cost-effective CPU+FPGA-based hardware development boards enable '*on-board'* and real-time SystemC and SystemC-AMS based simulation and offer new means for early prototyping and system validation of mixed-signal products and applications.

REFERENCES

[1] T. Hwang, J. Rohl, K. Park, J. Hwang, K. H. Lee, K. Lee, S.-J. Lee, and Y.-J. Kim, "Development of HIL Systems for active Brake Control Systems", SICE-ICASE International Joint Conference, 2006.

[2] P. Ehrlich, "HiL Firmware Prototyping Using an COSIDE Asynchronous Motor Model," https://www.coseda-tech.com/files/Files/Dokumente/asm_engine_firmware_prototyping_final.pdf

[3] G. Rutsch, S. Fontanesi, S. G. Herbst, S. Tan Hee Yeng, A. Possemato, G. Formato, M. Horowitz, W. Ecker, "Boosting mixed-signal design productivity with FPGA-based methods throughout the chip design process", Design and Verification Conference & Exhibition (DVCon Europe), October 2020, Munich.

[4] Steven Herbst, Byong Chan Lim, and Mark Horowitz. 2018. Fast FPGA emulation of analog dynamics in digitally-driven systems. In Proceedings of the International Conference on Computer-Aided Design (ICCAD '18). Association for Computing Machinery, New York, NY, USA, Article 131, 1–8. DOI:https://doi.org/10.1145/3240765.3240808

[5] Barnasconi, M., Dietrich, M., Einwich, K., Vörtler, T., Lucas, R., Chaput, J.P., Pecheux, F., Wang, Z., Cuenot, P., Neumann, I., Nguyen, T., 2015. UVM-SystemC-AMS Framework for System-Level Verification and Validation of Automotive Use Cases. Design & Test, IEEE PP, 1. https://doi.org/10.1109/MDAT.2015.2427260

[6] P. Ehrlich, T. Nguyen, T. Vörtler, "UVM-SystemC based hardware in the loop simulations for accelerated Co-Verification", Design and Verification Conference & Exhibition (DVCon Europe), October 2014, Munich.

[7] COSEDA Technologies, System Level Design Environment COSIDE, https://www.coseda-tech.com/coside-overview

[8] Siemens EDA, Catapult HLS, https://eda.sw.siemens.com/en-US/ic/catapult-high-level-synthesis

[9] Xilinx Vivado Logic Synthesis, https://www.xilinx.com/support/documentation-navigation/design-hubs/dh0018-vivado-synthesis-hub.html

[10] Xilinx Zedboard, https://www.xilinx.com/products/boards-and-kits/1-8dyf-11.html