

FPGA based Hardware-in-the-Loop Simulation for Digital Control of Power Converters using VHDL-AMS

Abdelouahab Djoubair Benhamadouche¹
Department of Electronic, Faculty of Technology,
University Ferhat Abbas Sétif 1,
Sétif 19000, Algeria

Farid Djahli²
LIS Laboratory, Department of Electronic,
University Ferhat Abbas Sétif 1,
Sétif 19000, Algeria

Adel Ballouti³
Department of Electronic, Faculty of Technology,
University Mohamed Boudiaf,
M'Sila 28000, Algeria

Abdeslem Sahli⁴
LEPCI Laboratory, Electronics Department,
University Ferhat Abbas Sétif 1,
Sétif 19000, Algeria

Abstract—This paper presents a new approach for complex system design, allowing rapid, efficient and low-cost prototyping. Using this approach can simplify designing tasks and go faster from system modeling to effective hardware implementation. Designing multi-domain systems require different engineering competences and several tools, our approach gives a unique design environment, based on the use of VHDL-AMS modeling language and FPGA device within a single design tool. This approach is intended to enhance hardware-in-the-loop (HIL) practices with a more realistic simulation which improve the verification process in the system design flow. This paper describes the implementation of a software/hardware platform as effective support for our methodology. The feasibility and the benefits of the presented approach are demonstrated through a practical case study of a power converter control. The obtained results show that the developed method achieves significant speed-up compared with conventional simulation methods, using minimum resources and minimum latency.

Keywords—*Hardware-in-the-Loop (HIL) simulation; Field-Programmable Gate Array (FPGA); VHDL-AMS; power converter; digital controller*

I. INTRODUCTION

In a top-down development process, design tasks are critical keys in system implementation success. In technologic research or industry fields, such as electronic, automation or mechatronic, designing system details go through different steps of analysis and modeling [1], where modeling consists in developing abstract descriptions of some physical realities in such a way that they are useful for the design process [2]. Conceived models are then used as an input to a specific simulator to study systems behavior or to explore unconsidered functionalities.

Modeling at different levels of abstraction of multi-domain systems needs the use of different methodologies and tools that should manage mixed-signal design and hardware/software challenges. For that reason, mixed-signal hardware description languages such as VHDL-AMS are indispensable, it is intended to provide a unifying trend that will link the various

tasks of analog and mixed-signal design in a coherent framework to support different design methodologies and different design tools [2], [3].

On another side, Verification and validation (V&V) procedures are the solutions for a system design success, they go along with each step of the development cycle. Different techniques can be used to perform those procedures. Usually, it depends on system architecture and hardware implementation. In different engineering systems such automotive, mechatronic or control systems, best practice for V&V requests the construction of a real prototype to test the whole system or one element of the system such as a new control algorithm [4]. This prototype is subjected to several cycles of testing and re-design in an expensive facility [5].

Therefore, hardware-in-the-loop (HIL) methods emerge in system design to be an effective way to resolve that issue. HIL methods seem to be an effective alternative to accelerate verification through a specific and relatively low-cost equipment, that permit to finally build a real and an operational prototype [6], [7].

In this paper, an original method for HIL simulation is presented, this method is proposed to simplify and accelerate systems design. This method is based on analog and mixed-signal (AMS) languages and programmable devices. The idea is to use the same hardware description language to model the digital part and the analog part of an AMS system, the digital component (essentially control procedure) is then implemented in FPGA, and the analog part is simulated through the use of VHDL-AMS and dedicated simulation software.

This paper is organized as follows. Section 2 introduces the essential of our approach that includes a short description of the context of each contribution of this work. Section 3 presents the implementation aspect of FPGA prototyping. Then, section 4 describes an application case to test and to show the benefits of the effective implementation of our method, where DC/DC converter is modeled in VHDL-AMS and controlled via a digital Proportional-Integral-Derivative

(PID) regulator implemented in FPGA. The last section states the conclusion and summarizes the most salient features of our contribution.

II. DESIGN APPROACH

Modeling and simulating a multi-domain system need the use of dedicated tools that can perform multi-abstraction simulation, functional prototyping, verifications, and validations. In the present work, those different design phases would be done through the use of FPGA device and VHDL-AMS modeling language.

In this work, VHDL-AMS is used as the mainstay for the system design; it will be used as a support language for modeling and simulation in the different phases of the system design flow. VHDL-AMS is an extension of the IEEE standard VHDL language, this language allows designers to model any mixed-signal plant that can be described by a system of differential and algebraic equations (DAE's), it supports the hierarchical description and the simulation of continuous and mixed continuous/discrete systems with conservative and non-conservative semantics [2], [8].

VHDL-AMS design requests a simulation process that involves a combined simulation between the analog part and the digital part of an AMS plants model. This is done through data communication and synchronization between a discrete-time and continuous-time simulator engines. In most applications which use VHDL-AMS, the digital side which is VHDL is usually excluded from the design or used only for behavioral simulation, and the analog facet of VHDL-AMS is the only one highlighted part [9]–[12]. In the present work, we will take advantages of VHDL to enhance VHDL-AMS design and vice versa, this is done through the validation of the following two points:

- Using VHDL-AMS for mixed systems modeling for pure simulation and verification.
- Using VHDL-AMS to develop FPGA application, in order to make a functional simulation, hardware implementation, and verification.

The second part of our design approach is using FPGA based hardware-in-the-loop techniques for test and verification. Usually, FPGAs are used when we need fast processing and parallel computing. Currently, they are also used for system rapid prototyping and system verification, where they take an important part on hardware-in-the-loop platforms. Several works have been recently proposed for FPGA based hardware-in-the-loop [13]–[18], where FPGA benefits are highlighted to simplify and accelerate considerably hardware-in-the-loop simulation. This technique is increasingly used in the development of system control in several fields of application.

Therefore, in this paper, we introduce a new method to use hardware-in-the-loop techniques, where FPGA device is associated with a simulation software to make an online simulation with VHDL-AMS model as resumed in Figure 1.

In order to reach our goal, we have to find the best way to bring together FPGA prototyping and VHDL-AMS modeling in an advanced software/hardware simulation platform.

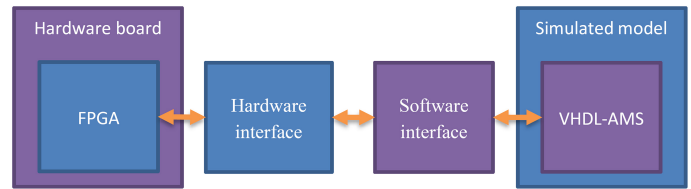


Fig. 1. AMS design for FPGA based HIL verification

III. DESIGN IMPLEMENTATION

The aim of the proposed platform is to set up a data communication between the proposed FPGA device and VHDL-AMS simulator. Figure 2 presents a simplified representation of what the HIL platform must enclose, three parts are essential; VHDL-AMS simulator, an FPGA development board and a software application to manage the simulation.

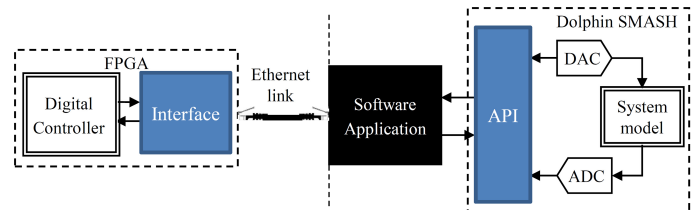


Fig. 2. Simplified representation for the proposed platform

A. VHDL-AMS Simulator

As mentioned in the last section, carrying out our approach need the use of a software which can perform VHDL-AMS simulation and have the ability to communicate with other applications. SMASH software from Dolphin integration is chosen to execute our approach, this software can support the full IEEE VHDL-AMS standard and a set of other modeling languages [19], [20]. Moreover, SMASH integrates an application programming interface (API) which gives functions and services to customize and to control simulation running.

B. The Software Application

To perform a hardware-in-the-loop simulation using VHDL-AMS, we have to control the simulation process through the SMASH API. According to Figure 3, a C++ application is built to allow data exchange between simulated models and the FPGA development board, this application performs the following tasks:

- The first task that should be accomplished is controlling the simulation process, so the application will be able to collect data from the simulated models and have to force signals values during simulation runs.
- The second task is to transfer data from and to the FPGA Board. Data collected from the simulator are assembled in one datagram then transferred in an UDP/IP packet to the FPGA through the Ethernet link. Afterward, the application is placed in listening mode and awaiting a response from the FPGA. When received, the datagram is unwrapped to extract the

needed data to force new signals values, and to allow SMASH to continue the simulation running.

- The last performed task is to synchronize and to manage data exchange between the simulator and the FPGA board.

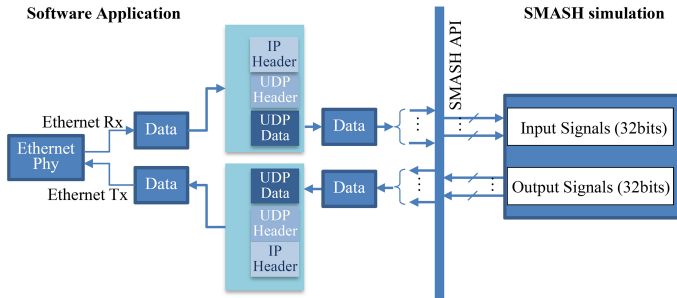


Fig. 3. Software application global view

C. FPGA Implementation

In the present approach, the HIL simulation does not require a sophisticated FPGA board, it simply needs a low-performance device with an Ethernet interface. To achieve the needed functionalities, a minimum configuration has to be implemented in the FPGA device as depicted in Figure 4.

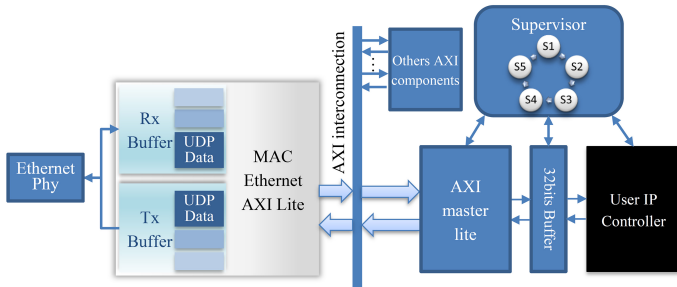


Fig. 4. Implementation of the hardware controller in the FPGA

To build a reconfigurable and an evolvable FPGA implementation, the AXI interconnection is used as the basement of our design, with a large number of compatible AXI IP core, we can extend our design to execute other useful functions. Moreover, it can integrate a processor core which is able to execute different control algorithms.

As described in Figure 4, the minimum configuration includes four essential blocks, that are:

- 1) Ethernet MAC (Media Access Controller): Many Ethernet controller cores can be used, it depends on several parameters like the targeted FPGA, connection speed or channel protocol. The proposed platform uses AXI Ethernet lite MAC from Xilinx [21], it provides an efficient Ethernet interface with least resources and can be configured to operate at the rate of 100Mb/s.
- 2) AXI Master: This module is written in VHDL, it follows the AXI protocol and provides signals to perform read/write operations in the AXI bus, this

is done to control the Ethernet module. On the other side, this module exchange data with the user controller module.

- 3) User IP Controller: This is the implementation of the digital controller under test.
- 4) Supervisor: This module manages all the operations executed by the overmentioned modules. The supervisor works under the control of a finite state machine (FSM). The FSM provides control signals to manage and synchronize data transaction between the different modules.

IV. RESULTS AND PERFORMANCES

A. Case Study

To illustrate our concept and bring out its benefits on system design, a typical example of an electronic power system is used. As shown in Figure 5, the used system consists of a buck power converter with a closed loop controller; this power converter is a typical example of AMS systems, it involves analog and digital behavior and can include other compartments from other disciplines like a thermal behavior or mechanical drive.

As depicted in Figure 5, a voltage sensor gets the digital value of the output voltage, this value is then compared to the reference voltage V_{ref} which generates an error signal $e(k)$ ready at the input of the Proportional-Integral-Derivative (PID) controller. The PID controller makes the feedback loop to regulate the output voltage across the load resistor (R). In this case, the PID runs continuously to generate a duty-cycle $u(k)$ that control the PWM generator to drive the transistor switch (SW).

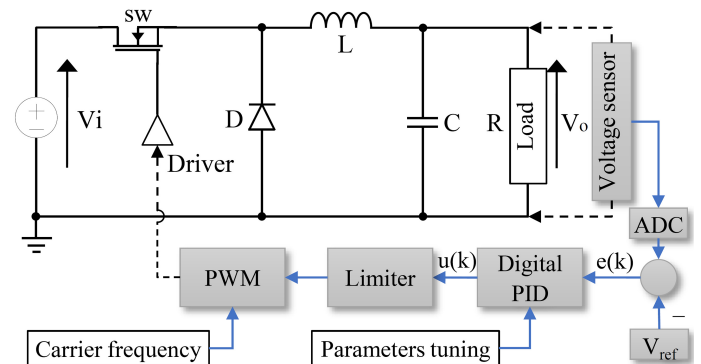


Fig. 5. Buck power converter structure with PID control loop

The simplest digital form of the PID controller in the discrete-time domain algorithm is given by the equation below:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Where the controller input $e(t)$ is the difference between the reference and the load voltage, and $u(t)$ is the controller output. The three parameters K_p , K_i , and K_d represent proportional, integrator and derivative gain coefficients.

TABLE I. SYSTEM PARAMETERS

Parameters	Values
PWM frequency	10.0 kHz
Input voltage	24.0 V
Filter inductor	1.0 mH
Filter capacitor	1.0 mF
Load resistor	1.0 Ω
Step simulation	Variable 2.0us - 50.0us

For a digital implementation, we are more interested in the discrete-time domain formula, this is given by the equation (2).

$$u[k] = u[k - 1] + K_1e[k] + K_2e[k - 1] + K_3e[k - 2] \quad (2)$$

Where: $K_1 = T_s K_i + \frac{K_d}{T_s}$, $K_2 = -K_p - 2\frac{K_d}{T_s}$, $K_3 = \frac{K_d}{T_s}$, and T_s is the sampling period.

In this work, a structural description is used to make a simulation model, each part of the system is designed in different levels of abstraction, depending on the needed accuracy and the complexity in each model. The analog part of the system is modeled in VHDL-AMS and the controller is initially described in VHDL-AMS, then a VHDL code is built to be finally implemented in FPGA.

To build a valid VHDL code intended for an FPGA implementation, first, we have to rearrange the code to make a synthesizable VHDL description according to the available FPGA resources [22]. when verified, the PID blocks are assembled with the others needed blocks for the HIL process, then timing constraints are checked for the final implementation. At this stage, an important issue is to be considered, a binary format representation of digital data has to be set, we choose a 32-bit fixed point format for input/output signals of the digital controller.

B. VHDL-AMS Simulation

First of all, the whole system is modeled in VHDL-AMS including system parameters as shown in Table I, the system is then simulated in a mixed signal mode with a variable time step, where each part of the system is verified with different scenarios especially the digital controller model.

The simulation waveforms depicted in the Figure 6.a shows the load voltage value for variable voltage reference; the result allows us to verify that the digital controller draws the output voltage of the analog buck converter according to the input reference. Also, in Figure 6.b we can see the different signals involved in the functioning of the digital PID such as the clock, Error, voltage signal...etc.

C. FPGA Simulation and Verification

In order to validate our approach, we perform numerous tests using the developed platform based on FPGA and VHDL-AMS simulator. The hardware part can be implemented in any FPGA development board that includes an Ethernet PHY. To be consistent with our objectives, we consider a low-cost development board; an Avnet LX9 MicroBoard embedding

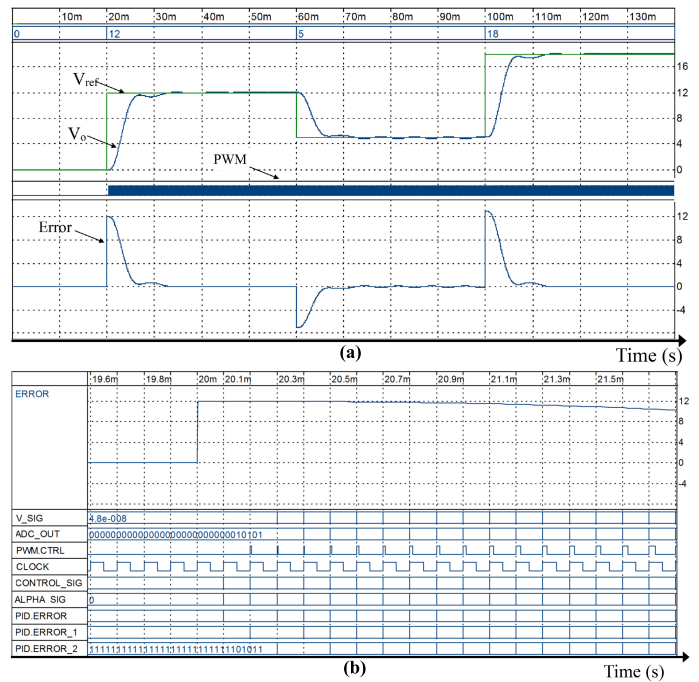


Fig. 6. VHDL-AMS simulation waveforms, (a) Voltage load and voltage error for variable voltage reference, (b) PID digital signals.

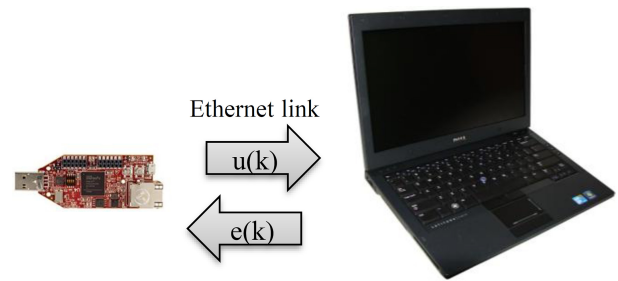


Fig. 7. FPGA based Hardware-in-the-loop plan.

a Xilinx Spartan-6 FPGA, which is suitable for low-budget design (Figure 7).

After programming the FPGA, the developed software application is executed, then SMASH software loads the plant model. After initializing the software parameters and the UDP/IP Ethernet communication channel, the user can begin the HIL simulation using a VHDL-AMS testbench, this last generates the necessary signals for the different test scenarios.

To compare efficiently software simulation and FPGA based HIL simulation, the same simulator parameters as for pure VHDL-AMS simulation (presented in the last section) are applied. Experimental results are obtained with a 100Mb/s Ethernet channel and data are transferred to, and from the FPGA board as vectors of 32 bits, and the FPGA is running at 100MHz.

For the test validation, the same scenarios are considered as used for the previous simulation. The Figure 8.a shows the result of FIL simulation, where the load voltage waveform is identical to the software simulation. In the zoomed view

TABLE II. FPGA USED RESOURCES FOR HARDWARE IMPLEMENTATION

Component	Slice Registers	Slice LUT	DSP blocks
Available	11440	5720	16
AXI_Ethernet_lite	492 [4,3%]	494 [8,6]	0
Supervisor + AXIMaster	197 [1,7%]	220 [3,8]	0
PID_dig	127 [1,1%]	217 [3,8]	12[75%]
Global	816 [7,1%]	931 [16,2]	12[75%]

at 20ms (Figure 8.b), the waveforms show a delayed response compared with the software simulation, this is due to latency in the transmission channel and the timing constraints associated with the FPGA implementation

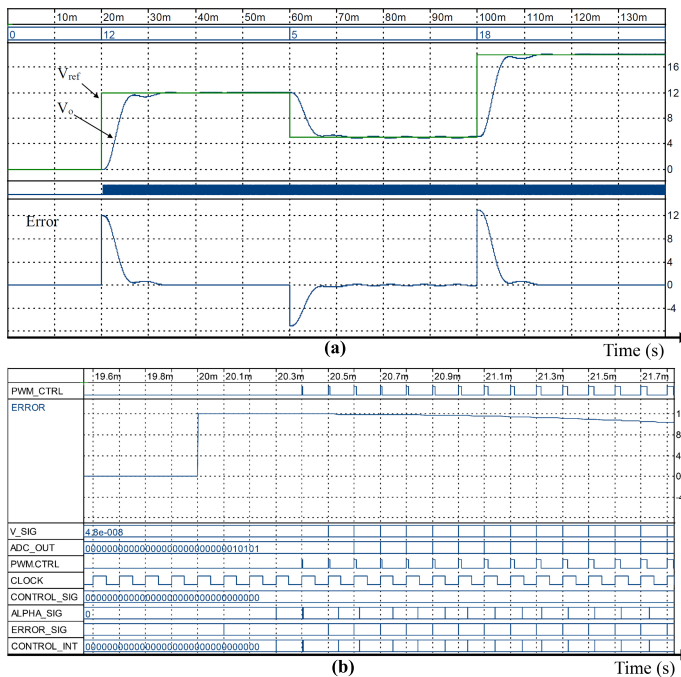


Fig. 8. HIL simulation waveforms (a) Voltage load and voltage error for variable voltage reference (b) PID digital signals.

D. FPGA Resources

Table II lists the FPGA resources used to implement the needed components, consumed resources depend on the used FPGA technology, and proportions are referred to the available resources of a Spartan-6 XC6SLX9.

In this table, the AXI Ethernet lite module is a Xilinx generated core which provides the Ethernet interface, the PID_dig represents the digital controller part, and the third component refers to the supervisor part that is necessary to connect and control all needed components via an AXI bus.

The global utilization of LUT represents only 16,2% of the available LUT, and the integration of the PID controller represent only 3,8%, those results show that the proposed HIL platform don't need large FPGA resources, and with the actual FPGA device there are enough resources to replace the digital PID with a more complex controller or to add other modules for additional functionalities.

TABLE III. TIME SIMULATION FOR DIFFERENT SCENARIOS

	Simulation Mode	CPU global time	Logic kernel time	Improvement of simulation time
Scenario 1 140ms & 10kHz	Software	7,83 s	5,38 s	83,66%
	HIL_AMS	1,28 s	0,109 s	6,55 s
Scenario 2 1000ms & 10kHz	Software	39,14 s	30,735s	75%
	HIL_AMS	9,79 s	0,733 s	29,35 s
Scenario 3 140ms & 10kHz	Software	29,22 s	21,62s	80,10%
	HIL_AMS	5,70 s	0,608 s	23,52 s
Scenario 4 1000ms & 50kHz	Software	224,7 s	182,39 s	82,45%
	HIL_AMS	39,42 s	3,76 s	184,48 s

E. Timing Performance

To bring out the benefits of the implementation of our approach in terms of time processing, several simulation scenarios are performed in software mode and in FPGA mixed mode with a variable simulation time step. The simulation times have been measured using a 2.5 GHz Intel Core i5 with 4 GB of RAM memory.

Table III summarises the CPU time and the logic kernel time required to perform for the different scenarios. Those scenarios simulate the same design for different simulation times and for different switching frequencies, this implies a different volume of analog and discrete-time simulation points which demand various hardware resources.

The needed simulation time is significantly reduced (about 80%). This is due to the fact that software mixed simulation uses big CPU resources for digital simulation, while in our approach; the digital simulation is exclusively achieved by the FPGA device.

On the other hand, we measured the time latency for different simulation cases. We note that the latency for network transmission and processing within the FPGA is about 55µs and the latency for software processing is about 25ms. Improving hardware and software latency for a specific application case can lead to real-time hardware-in-the-loop simulation.

V. CONCLUSION

The work exposed in this paper focuses on the presentation of a new approach for simulation and verification of complex systems. With this goal in mind, this work introduces a unique design tool for hardware-in-loop simulation based on FPGA and VHDL-AMS. Thereby, benefits of VHDL-AMS mixed design and FPGA rapid prototyping are combined together to improve system design and to allow fast design development time and a short time to market.

In order to validate our approach and to demonstrate its effectiveness, results of the implementation and simulation of a regulated power converter are presented, where the digital controller is coded using VHDL language then implemented in FPGA and the rest of the system is simulated on a host PC using VHDL-AMS models. The studied application gives an overview of the capabilities of this method to accomplish multiple design benefits, where several experiences have been achieved shown a significant reduction of simulation processing time and a low latency for data transfer. On another hand, hardware resources are minimized, to match any low-cost FPGA development board for a high-reliability simulation.

REFERENCES

- [1] D. Gianni, A. D'Ambrogio, and A. Tolk, *Modeling and Simulation-Based Systems Engineering Handbook*, CRC Press, 2014.
- [2] F. Pêcheux, C. Lallement, and A. Vachoux, "VHDL-AMS and Verilog-AMS as alternative hardware description languages for efficient modeling of multidiscipline systems", *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, pp. 204–225, 2005.
- [3] G. G. Gielen and R. Rutenbar, "Computer-aided design of analog and mixed-signal integrated circuits", *Proc. IEEE*, vol. 88, pp. 1825–1854, 2000.
- [4] P. G. Maropoulos and D. Ceglarek, "Design verification and validation in product lifecycle", *CIRP Ann-Manuf. Techn.*, vol. 59, pp. 740–759, 2010.
- [5] G. G. Parma and V. Dinavahi, "Real-time digital hardware simulation of power electronics and drives", *IEEE Trans. Power Del.*, vol. 22, pp. 1235–1246, 2007.
- [6] M. Bacic, "On hardware-in-the-loop simulation", *Proceeding of the 44th IEEE Conference on Decision and Control*, Seville, pp.3194—3198, 2005.
- [7] A. Bouscayrol, "Different types of Hardware-In-the-Loop simulation for electric drives", *Proceeding of the International Symposium on Industrial Electronics*, Cambridge, pp. 2146–2151, 2008.
- [8] E. Christen and K. Bakalar, "VHDL-AMS-a hardware description language for analog and mixed-signal applications", *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 46, pp. 1263–1272, 1999.
- [9] A. Laouamri, F. Djahli, A. Rabhi, A. Benhamadouche, A. Ballouti, and A. Bendjadou, "A Virtual Prototype of Proton Exchange Membrane Fuel Cell Using VHDL-AMS Language", *J FUEL CELL SCI TECH*, vol. 6, pp. 024502, 2009.
- [10] Y.-A. Chapuis, L. Zhou, H. Fujita, and Y. Hervé, "Multi-domain simulation using VHDL-AMS for distributed MEMS in functional environment: Case of a 2D air-jet micromanipulator", *Sensors and Actuators A: Physical*, vol. 48, pp. 224–238, 2008.
- [11] L. Zerioul, M. Ariaudo, and E. Bourdel, "RF transceiver and transmission line behavioral modeling in VHDL-AMS for wired RFNoC", *ANALOG INTEGR CIRC S*, vol. 92, pp. 103–114, 2017.
- [12] A. Rezgui, L. Gerbaud, and B. Delinchant, "VHDL-AMS electromagnetic automatic modeling for system simulation and design", *IEEE Trans. Magn.*, vol. 50, pp. 1013–1016, 2014.
- [13] Ó. Lucía, I. Urriza, L. Barragan, D. Navarro, Ó. Jiménez, and J. M. Burdío, "Real-time FPGA-based hardware-in-the-loop simulation test bench applied to multiple-output power converters", *IEEE Trans. Ind. Appl.*, vol. 47, pp. 853–860, 2011.
- [14] B. Lu, X. Wu, H. Figueroa, and A. Monti, "A low-cost real-time hardware-in-the-loop testing approach of power electronics controls", *IEEE Trans. Ind. Electron.*, vol. 54, pp. 919–931, 2007.
- [15] M. Matar, D. Paradis, and R. Iravani, "Real-time simulation of modular multilevel converters for controller hardware-in-the-loop testing", *IET Power Electron.*, vol. 9, pp. 42–50, 2016.
- [16] Ó. Jiménez, Ó. Lucía, I. Urriza, L. A. Barragan, D. Navarro and V. Dinavahi, "Implementation of an FPGA-Based Online Hardware-in-the-Loop Emulator Using High-Level Synthesis Tools for Resonant Power Converters Applied to Induction Heating Appliances", *IEEE Trans. Ind. Electron.*, vol. 62, pp. 2206–2214, 2015.
- [17] A. Penczek, R. Stala, L. Stawiarski, and M. Szarek, "Hardware-in-the-Loop FPGA-based simulations of switch-mode converters for research and educational purposes", *Przeegląd Elektrotechniczny*, vol. 87, pp. 194–200, 2011.
- [18] C. Foucher and A. Nketsa, "A Dynamic FPGA-based Hardware-in-the-Loop Co-simulation and Prototype Testing Platform", *Proceeding of The Tenth International Conference on Systems*, Barcelona, pp. 68–73, 2015.
- [19] P. V. Nikitin, C.-J. R. Shi, "VHDL-AMS based modeling and simulation of mixed-technology microsystems: a tutorial", *The VLSI Journal*, vol. 40, pp. 261–273, 2007.
- [20] Dolphin Integration. SMASH Simulator. http://www.dolphin.fr/index.php/eda_solutions/products/smash/overview.
- [21] Product Guide, "LogiCORE IP AXI Ethernet Lite MAC (v1.01.b). Xilinx®. Retrieved from https://www.xilinx.com/support/documentation/ip_documentation/axi_ethernetlite/v1_01_b/ds787_axi_ethernetlite.pdf
- [22] V. Sklyarov, I. Skliarova, A. Barkalov, and L. Titarenko, *Synthesis and Optimization of FPGA-Based Systems*, Springer Science & Business Media, 2014