2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE
MUNICH, GERMANY
OCTOBER 15-16, 2024

A new approach to integrated AI into analog/mixed-signal verification workflow

Long Hoang, Insight Centre for Data Analytics, School of Computer Science and Information Technology, University College Cork, T12 K8AF, Cork, Ireland (*lhoang@ucc.ie*)

George Duffy, Analog Devices Inc., D04Y8X5, Dublin, Ireland (*George.Duffy@analog.com*)

Emanuel Popovici, Insight Centre for Data Analytics, School of Electrical Engineering, University College Cork, T12 K8AF, Cork, Ireland (*e.popovici@ucc.ie*)

***Abstract*— Verification of complex integrated circuits (IC) is a significant concern for many microelectronic designs. It aims to identify and eliminate any design errors or bugs, ensuring that the system performs as expected under various conditions. However, for very large designs, it becomes the bottleneck due to cost and resource limitations. Performing simulation using hardware description language (HDL) is a common approach to verify digital circuits, but simulation times can increase when the complexity grows. Moreover, simulating accurately analog circuits is also a time consuming process. The application of AI in verification is an ongoing effort, showing promise to speed up the verification process. This paper introduces a new method to integrate AI into an analog/mixed-signal (AMS) verification environment. The main idea of the paper is to create a digital twin of an analog circuit and moving the verification of analog circuits into digital space by means of using AI. The result shows that the proposed framework reduces the simulation time while keeping good accuracy compared with the AMS/HDL simulation.**

Keywords*— *Hardware Description Language; Analog/Mixed-Signal; Artificial Intelligence; Integrated Circuits Verification; digital twin of an analog circuit

I. INTRODUCTION

Mixed-signal applications are one of the fast-increasing market segments in the semiconductor industry. As a result, most IC designs nowadays are mixed signals. Verifying Mixed-signal IC is challenging because it comprises both analog and digital blocks, which fundamentally use various paradigms in simulation. While digital simulators continuously deal with logical expressions by triggering events, analog simulators must deal with the complete system matrix at every time step [1]. The fast emergence of AI and ML techniques could benefit mixed-signal IC verification by reducing simulation time. However, integrating AI into the AMS environment remains a challenge.

This work presents a new approach incorporating AI into the analog/mixed-signal verification workflow. Specifically, the AI model in SystemVerilog will be created to predict the circuit behavior, like the voltage's output, and the testbench will be run in Cadence Virtuoso. This paper is structured as follows. Section II summarizes the related works. Section III introduces the proposed framework. The evaluation result is given in section IV. Finally, section V gives the conclusion and the application of this work.

II. RELATED WORK

Deepak Narayan Gadde et al. [2] introduced two approaches to improve design verification (DV) throughput: ranking and the new machine learning (ML) based technology. Both approaches aim to attain comparable coverage while minimizing CPU time consumption, which is achieved through the application of more efficient stimuli.

Sundeep Srinivasan et al. [3] presented a technique that builds upon existing tools for constrained-random DV environments, incorporating supervised and reinforcement machine learning to outperform random testing while remaining highly automated. In addition, the proposed method achieves full design coverage, the goal of DV, much faster and with less resource expense.

Despite significant achievements in design verification using machine learning, there is still a lack of machine learning techniques for circuit verification. This paper presents a new approach to integrating AI into the circuit verification process.

III. THE PROPOSED FRAMEWORK

The proposed framework is shown in Figure 1. Cadence Virtuoso [4], the IC tool for developing analog and mixed-signal circuits, is used as the simulation tool. Firstly, we run the analog/mixed-signal circuit simulation in Cadence Virtuoso, then collect the time series of pair data (input-output) and use them to train the AI model (Python model). After finishing the training process, the AI model predicts unseen data and compares it with the actual outputs of analog/mixed-signal circuits and hardware description language.

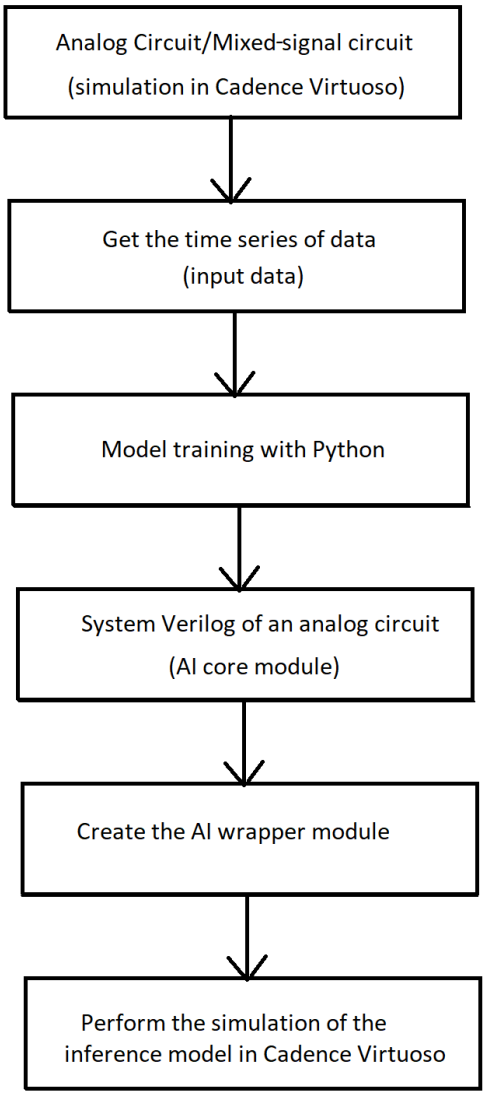

Figure 1. The proposed framework

Maciej Wielgosz et al. [5] created the machine-learning model in HDL from Python code using the MyHDL library [6]. The drawback of the proposed mapping method is that it works with VHDL and Verilog, not System Verilog. We handled this problem by modifying the original MyHDL library to perform with the System Verilog. The next step is to create the AI wrapper module, which is the interface between the analog source and the AI core

module. The wrapper module receives the value of the input source and plots the outputs based on the calculations of the core AI module.

IV. THE EVALUATION RESULT

The step-by-step procedure for integrating the proposed framework into the RC low-pass filter, a popular AMS circuit, will be presented to verify the proposed framework. Also, the numerical result and a comparison with the traditional approach will be given. The calculation followed the theory [7], with the values of R and C being 200 Ohm and 5mF, respectively. We vary the period of the input square wave from 55 to 100 seconds with a time step of 5 seconds, run the testbench in the virtuoso environment, collect the input-output pairs and export them to CSV files. Figure 1 shows an example of the input-output pair with a period of 60 seconds. The twenty exported data files are time series data. Each has two columns: one containing the time value in seconds and the other containing the voltage value. Eighteen CSV files, input1 to input9 and output1 to output9, are used for training the AI Python model. The last two CSV files, including input10 and output10, are used for testing the AI model.

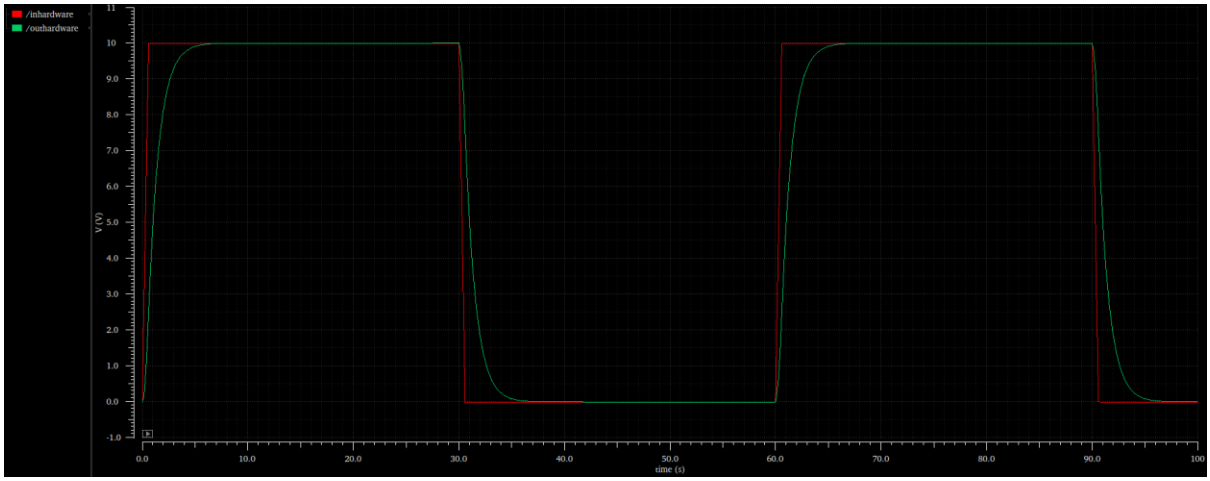

Figure 1: The input2-output2 pair with a period of 60 seconds

We adopted deep learning to hardware description language [8] and created three modules for AI Python scripts.

AItestPython: The module reads the training data (Eighteen CSV files) collected from the analog simulation and uses them as inputs for the AI model. The next step is to train the model, predict the output using the input10, and then compare the output of the AI Python model with the actual output (output10). After finishing the training process, export the trained model to the h5 file.

Transferweight: The module loads the Python-trained model in the h5 file. It converts the weights and bias from the floating point to the fixed point, which is used for the hardware description language. Finally, the module saves the weights and bias in fixed-point representation into a pt file.

ConvertAImodeltoSystemVerilog: The module takes the weight and bias from the pt file and declares them as constants in the inference model. The inference model (the AI core module) was created by the MyHDL library and is written in SystemVerilog language. The original MyHDL does not support SystemVerilog. A beta version of MyHDL [9] supports SystemVerilog and works only with basic examples. We added the bias and const into the write signal part and weight into the write memory signal part, making it work with the SystemVerilog to generate the AI inference model (the AI core module).

Next, import the core AI module file into the virtuoso environment. Create the testbench with the AI wrapper. The AI wrapper is adopted from [8], and we added the AI input and AI output signal. The AI core module will take input from the analog source in the Cadence Virtuoso environment and call the AI core module, calculate and output the value. Lastly, run the testbench and plot the result from the AI model. The testbench has the same input source for the AI, analog, and non-AI approach SystemVerilog (Figure 2). The implementation of the low pass

filter using non-AI SystemVerilog is taken from the Real Modeling with SystemVerilog Training course at Cadence. As seen in Figure 3, the input with an amplitude of 10 V and a period of 100 seconds is the same for all modules (red, pink, purple), and the output from the AI model, non-AI SystemVerilog, and analog are green, cyan, and brown, respectively. Table 1 gives the simulation time for all methods. The proposed framework reduces the time while achieving a good result compared with other approaches.

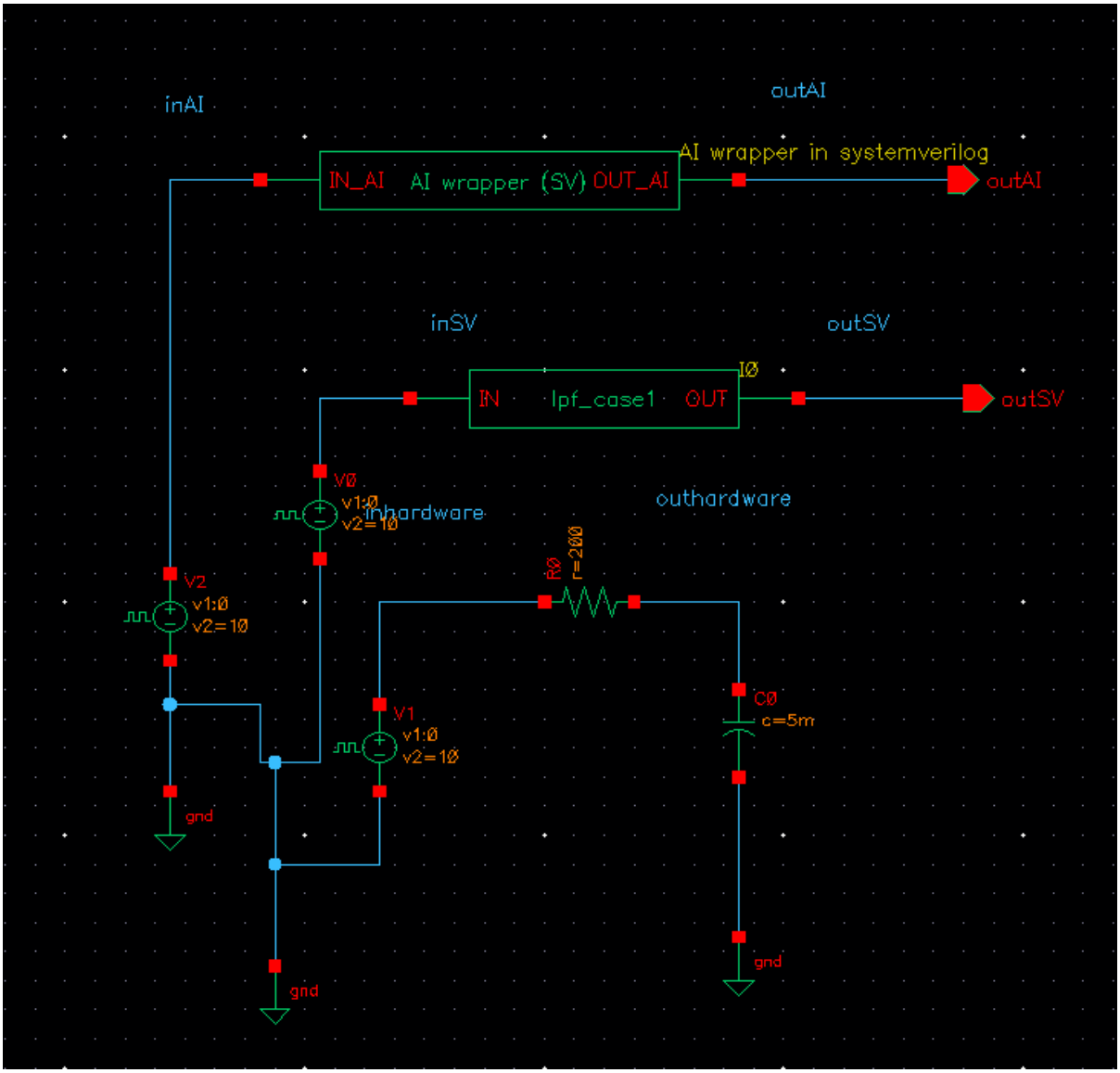

Figure 2: The testbench for RC lowpass filter with the square wave input (period of 100 seconds): AI wrapper module (top), non-AI SystemVerilog module (middle), and analog module (bottom).

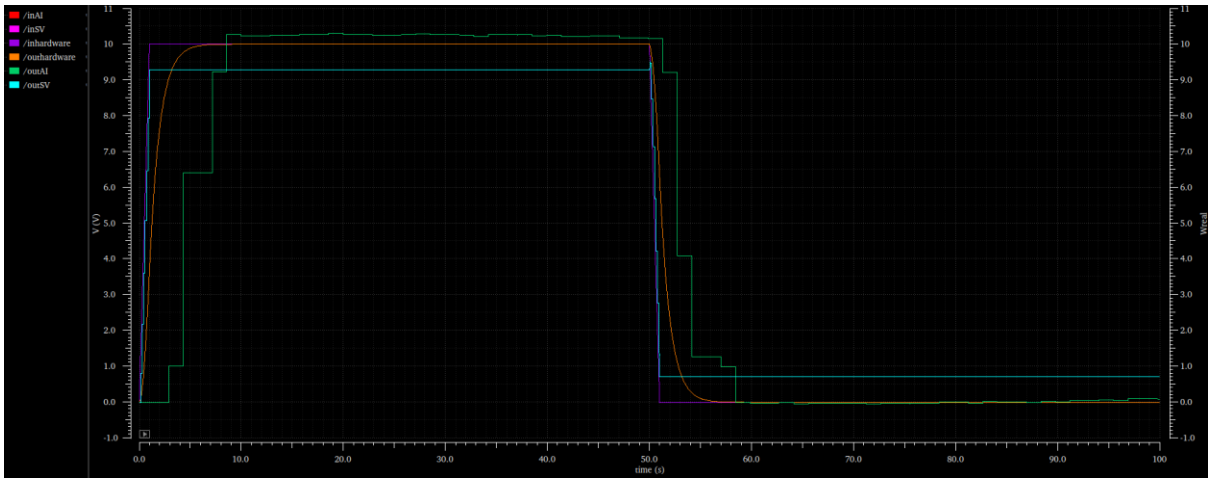

Figure 3: The output of the testbench for RC lowpass filter with the square wave input (period of 100 seconds): AI wrapper module (green color), non-AI SystemVerilog module (cyan color), and analog module (brown color).

Methods	Run time (seconds)
Analog simulation	16
Non-AI SystemVerilog simulation	15
AI methods	14

Table 1: Run times for all methods.

The proposed method is evaluated on various analog/mixed-signal circuits at Analog Devices Inc., demonstrating promising results compared to other simulation approaches. The framework achieves comparable accuracy for simple circuits with HDL simulation while requiring less running time. The framework significantly reduces the running time for complex circuits compared to traditional analog/mixed-signal simulation in the Cadence Virtuoso environment. To our knowledge, no existing works integrate AI directly into an AMS environment and evaluate the performance of an inference AI model (implemented in SystemVerilog).

V. CONCLUSION

The evaluation result shows that AI can improve the AMS verification process when integrated into the simulation environment. Our main contribution is developing the completed framework in SystemVerilog, including the inference AI model.

Circuit verification, an essential process in the early stages of IC development, ensures that each circuit meets specific requirements before becoming a part of a more complex system. With the versatility in verifying the behaviors of simple and complex circuits, the proposed method has enormous promise for a wide range of industrial applications.

ACKNOWLEDGMENT

This research is supported in part by a grant from the Science Foundation Ireland INSIGHT Centre for Data Analytics (grant number 12/RC/2289-P2), and Analog Devices.

REFERENCES

[1] Solutions for Mixed-Signal SoC Verification Using Real Number Models. URL: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/solutions/mixed-signal-verification-wp.pdf.

[2] Deepak N G, Sebastian S, Djones L, Thomas Z, “Improving Simulation Regression Efficiency using a Machine Learning-based Method”, Design and Verification Conference & Exhibition (DVCon Europe), December 2022, Munich.

[3] William H, Maithilee K, Sandeep S, Rohit S, “Optimizing Design Verification using Machine Learning: Doing better than Random”, Design and Verification Conference & Exhibition (DVCon Europe), December 2022, Munich.

[4] Cadence Virtuoso. IC design tool. URL: https://www.cadence.com/content/dam/cadence-www/global/en_US/documents/tools/ic-package-design-analysis/virtuoso-system-design-platform-ds.pdf.

[5] Maciej W, Michał K, “Mapping Neural Networks to FPGA-Based IoT Devices for Ultra-Low Latency Processing”, Sensors, 2019, 19, 2981. URL: https://doi.org/10.3390/s19132981.

[6] MyHDL. Design hardware with Python. https://www.myhdl.org/.

[7] Low Pass RC Circuit Response to a Square Wave. URL: https://www.mathforengineers.com/transients-in-electrical-circuits/low-pass-RC-response-to-square-wave.html#google_vignette.

[8] The DL2HDL mapping tool. URL: https://bitbucket.org/maciekwielgoszteam/dl2hdl/src/master/.

[9] David Johnston, MyHDL. URL: https://github.com/dj-on-github/myhdl/tree/master.