Fast and Furious

Quick Innovation from Idea to Real Prototype

Simone Fontanesi, Infineon Technologies AG, Villach, Austria (Simone.Fontanesi@infineon.com)

Gaetano Formato, Infineon Technologies AG, Villach, Austria (Gaetano.Formato@infineon.com)

Thomas Arndt, COSEDA Technologies GmbH, Dresden, Germany (Thomas.Arndt@coseda-tech.com)

Andrea Monterastelli, Infineon Technologies AG, Villach, Austria (Andrea.Monterastelli@infineon.com)

***Abstract* — Automotive customers are starting to request sensor prototypes to test new algorithms and application possibilities in the early stage of their design. Reacting fast to their requests can ensure design-wins for the next generation product (and a prosperous future for Infineon Sense and Control). For this reason, a new methodology, allowing a smooth transition from idea to implementation, has been implemented and is already being used for different products. In particular, product models, developed during concept and feasibility studies in SystemC, are translated automatically into HDL ready to be synthesized into FPGAs for prototyping by the customer. The generated HDL is first automatically validated via co-simulations with SystemC, then the FPGA prototype is tested in a laboratory system test bench and finally the measured results are again automatically compared to simulation results. The flow is highly automatized and allows time and effort to be saved when developing a prototype for the customer, without compromising on performance and reliability.**

Keywords — prototype; FPGA; model; virtual prototype SystemC; SystemC AMS; automatization; HDL; high level synthesis; automotive

I. INTRODUCTION

Electrification and autonomous driving are changing requirements in the automotive world. Even in standard applications like engine, transmission and wheel speed sensors the room for innovation is big. OEMs and Tier1s want to try new algorithms, architecture and sensor technologies in the field, and therefore ask for hardware prototypes. The hardware prototypes are normally developed within months and are characterized by dynamic changes of requirements to find the perfect fit with the application. The standard project-driven approach, where people with different backgrounds (concept, application, analog, digital design, verification…) work on the project for a number of years cannot be used; a more start-up mind approach has to be adopted. For this reason, we developed a methodology that allows a smooth transition from the idea to the real hardware prototype. Our vision is to enable concept and applications engineers to develop a hardware prototype without a big effort and the need of a complete team. Starting from a model, which is verified extensively via simulations, Hardware Description Language (HDL) is generated with a High Level Synthesis (HLS) tool and co-simulated with the golden reference SystemC model. When the simulation shows a pass, the code is synthetized onto the real FPGA (Field Programmable Gate Array) hardware and tested in the laboratory and in the field.

II. METHODOLOGY

The goal of this work was to develop a complete methodology allowing the transition from concept to implementation in a fast, flexible and lean way. Once a product model is available in SystemC / SystemC AMS (this is normally developed during concept phase), the translation to HDL is preformed automatically and the virtual prototype becomes a real prototype that can be delivered to customers and even tested in the application. In order to reach this goal various state-of-the-art tools have been used and we collaborated with some of the tool providers (i.e., COSEDA Technologies GmbH [1]) to make a one-click solution possible.

The methodology flow (described graphically in Figure 1) can be summarized in the following main points:

- Idea: Sensor model development in SystemC (COSIDE®), as usually done in Infineon Sense and Control. Sensing element, analog path and digital core are modeled in SystemC and SystemC AMS;
- Verification of the idea: Simulation and validation of the model via regressive simulations (controlled by Matlab). Thousands of simulations are run iteratively, simulating various use cases with real world inputs;
- From Idea to implementation: Automatic translation of the code to HDL, ready to be synthetize on FPGA (initially Vivado HLS, now trying other tools like Mentor Catapult). Effort to write VHDL or Verilog code is saved, performing an automatic translation of the digital core from SystemC to HDL;
- Verification of the implementation: Automatic co-simulation of HDL and SystemC, reusing the test setups written to verify the model (Cadence Virtuoso). The same stimuli generator and evaluator used to verify the SystemC are used to simulate the HDL as well. Given the same inputs, the outputs of the SystemC and HDL should match;
- From implementation to real HW: Synthesis on FPGA (Xilinx);
- Verification of the real HW: Automatic comparison between measurements and simulation results (Matlab + Labview). For selected use cases it is possible to provide the hardware and the model with the same input stimuli. Given the same input stimuli, the outputs of the SystemC model and of the real hardware should match.

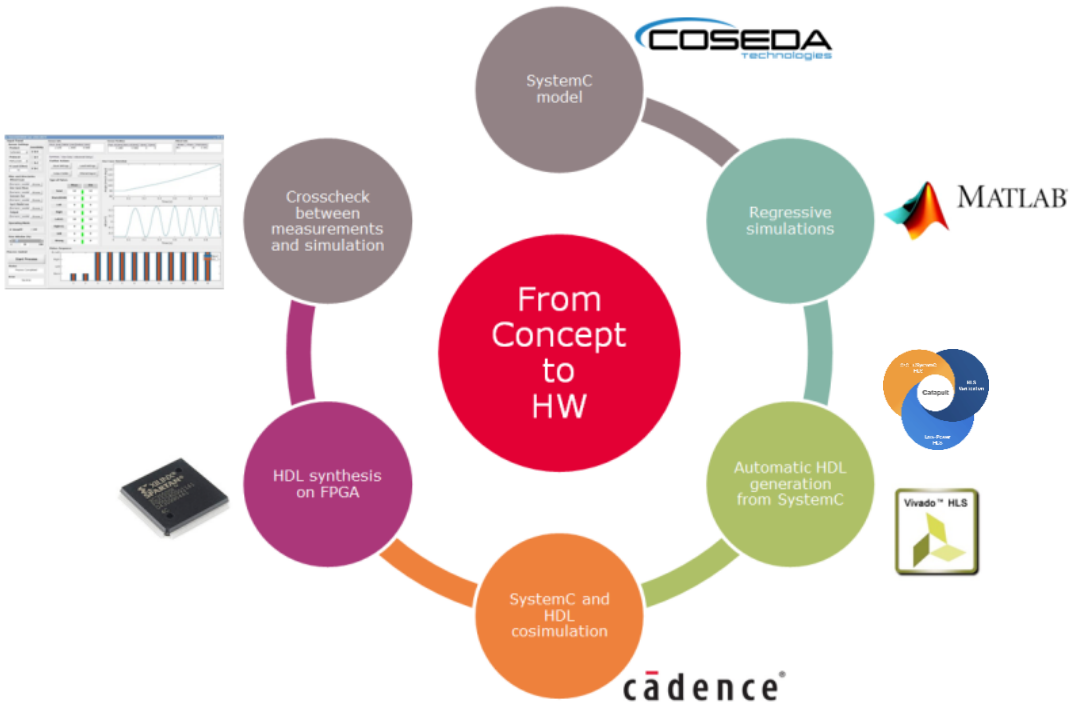

Figure 1. Fast and Furious methodology

A. *Sensor model development in SystemC*

The methodology described in this paper has been applied on a magnetic speed sensor to be used in automotive transmission application. Magnetic speed sensors sense the speed and rotational direction of a rotating wheel and are used in various applications (wheel speed, transmission, crankshaft and camshaft) because they provide a contactless, robust and low-cost solution [1]. Typically, a magnetic encoder is applied to the shaft and the sensor senses the magnetic field variation generated by the rotation. Alternatively a back bias magnet can be used to generate a constant field that is modulated by the rotation of a steel wheel. The two options are shown in Figure 1.

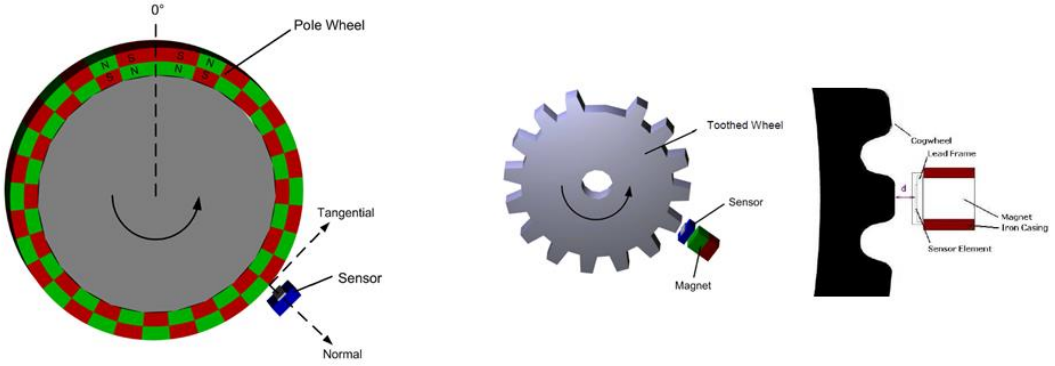

Figure 2. Magnetic encoder and tooth wheel with back-bias magnet

To sense the magnetic field, different sensing element types are used. The most common are the Hall principle [3] or via the measurement of a resistor which varies with magnetic field such as GMR (Giant Magneto-Resistance) [4], TMR (Tunnel Magneto-Resistance) [5] or AMR (Anisotropic Magneto-Resistance) [6].

As explained in [7] product models based on SystemC are widely used to simulate the behavior of an integrated circuit to reliably assess the clarity and completeness of the requirements definition, explore different possible architectures and verify the hardware requirements of each block.

Taking the internal block diagram and product requirements as a reference, a SystemC model of the next generation transmission sensor has been developed in the COSIDE® environment from COSEDA Technologies GmbH [7]. In particular, to describe the analog modules SystemC AMS has been used, while for the digital domain plain SystemC has been chosen.

SystemC is the preferred language for product modeling in our department because for correctly describe the behavior of our sensors a detailed description of the digital algorithms is needed and SystemC allows to describe digital behavior very well. Moreover, simulations are faster and easier to run compared than to hardware description languagse like Verilog or VHDL and can be executed both on Windows or Unix. Lastly, once compiled, it is possible to share the model with customers protecting the intellectual property.

B. Simulation and validation of the model via regressive simulations

The SystemC model is validated against real world use cases with thousands of simulations. The simulation flow is described in Figure 3.

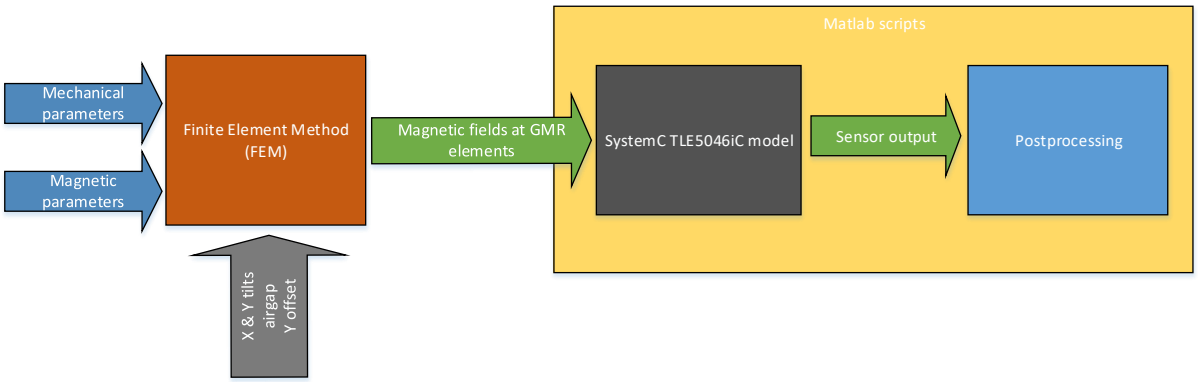

Figure 3. Simulation flow [7]

Finite Element Method (FEM) simulations are used to obtain the magnetic field vector that is used as an input for the SystemC model. In [7] the flow was already described in details. In this work, the flow has been improved increasing the number of use cases and the level of automation. In particular, the stimuli generation and output evaluation take place now in SystemC, increasing the speed of the simulations. Matlab [8] scripts control the

model parameter and test case configuration. The following picture shows the stimuli generation, sensor model and output evaluator modules in SystemC.

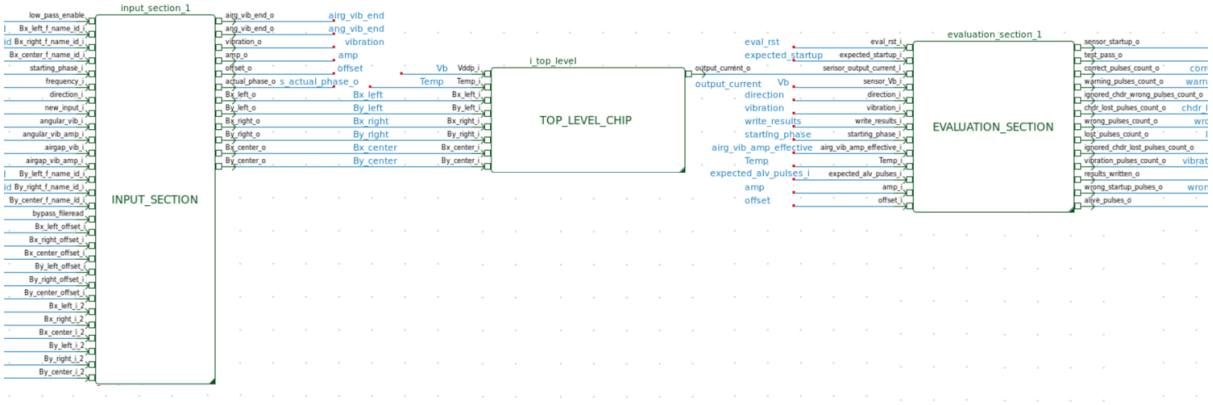

Figure 4. SystemC stimuli generator and evaluation section

The stimuli generator, is capable of generating the following use cases, taking the magnetic field from FEM simulations as an input:

- Forward / Backward rotation;
- Change of direction;
- Sudden airgap change;
- Angular and airgap vibrations;
- Wheel run-out;
- Stray field influence.

C. Automatic translation of the code to HDL

The SystemC code of the digital core has been written taking into account the need for a VHDL translation. From this perspective, all the language constructs which are possible in SystemC, but not synthesizable have been avoided, as well as the use of "double" data types and complex C++ structures.

Vivado HLS [9] software has been used to translate each SystemC module of the digital core into VHDL. The result was a set of VHDL modules, with the same hierarchy as the SystemC modules. This was possible thanks to a new option implemented in COSIDE® allowing the generation of a netlist with a simpler syntax that can work with high level synthesis tools.

In total, around fifteen SystemC modules have been converted, each of them with a complexity of around five hundred lines of code. An expert digital designers would have taken months to convert the behavior of the SystemC model into a HDL to be synthetized on the FPGA. With Vivado HLS the goal was reached within about two weeks. Later, the same approach has been used with a second tool, Mentor Catapult® [10]. In this case, the translation process was performed more smoothly in just around a day. We can definitely say that high level synthesis approach can save various man-months of effort. This is a crucial point for prototyping activities, where speed is essential.

D. Automatic Cosimulation of HDL and SystemC

To verify the translation, Cadence Virtuoso [11] environment was used to cosimulate the VHDL translation of the digital core and the SystemC model. In Virtuoso, a schematic was realized in which the SystemC "Input section" module was instantiated. The analog part of the sensor was wrapped in an "Analog section" SystemC module and fed with the signals coming from the input section; the same was done with the ADC SystemC module. The outputs of both the analog section and the ADC were connected to the SystemC digital core module

and to the VHDL translation of the same. The outputs of the two digital core implementations (VHDL and SystemC) were connected to two analog interface SystemC modules. Two "Evaluation section" SystemC modules were instantiated to evaluate the outputs of the two analog interface modules. A simplified structure is shown in Figure 5.

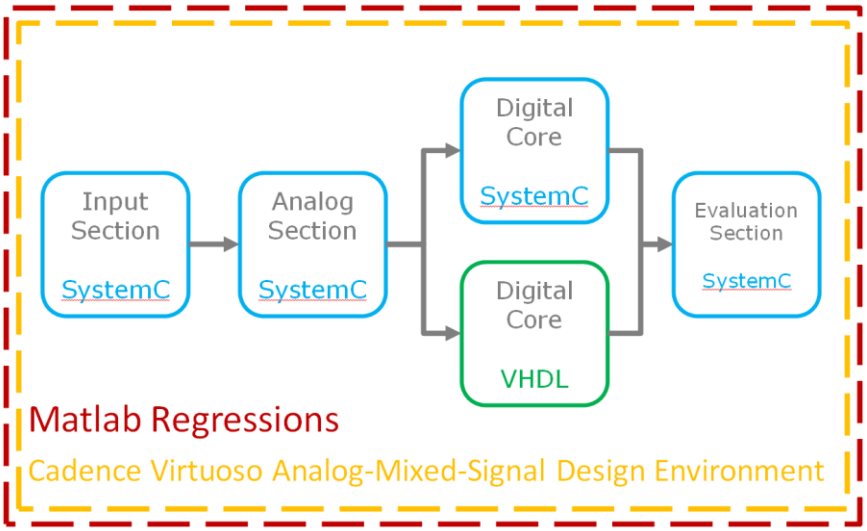

Figure 5. Cosimulation setup in Cadence

This work flow enabled the cosimulation of the SystemC and VHDL code of the digital core, by using the same inputs and the same evaluation logic for both. The result of the cosimulation consists then in two different reports, one for VHDL and one for SystemC, characterizing the output pulses of the two output signal, in order to compare them. Figure 6 shows an example of cosimulation.

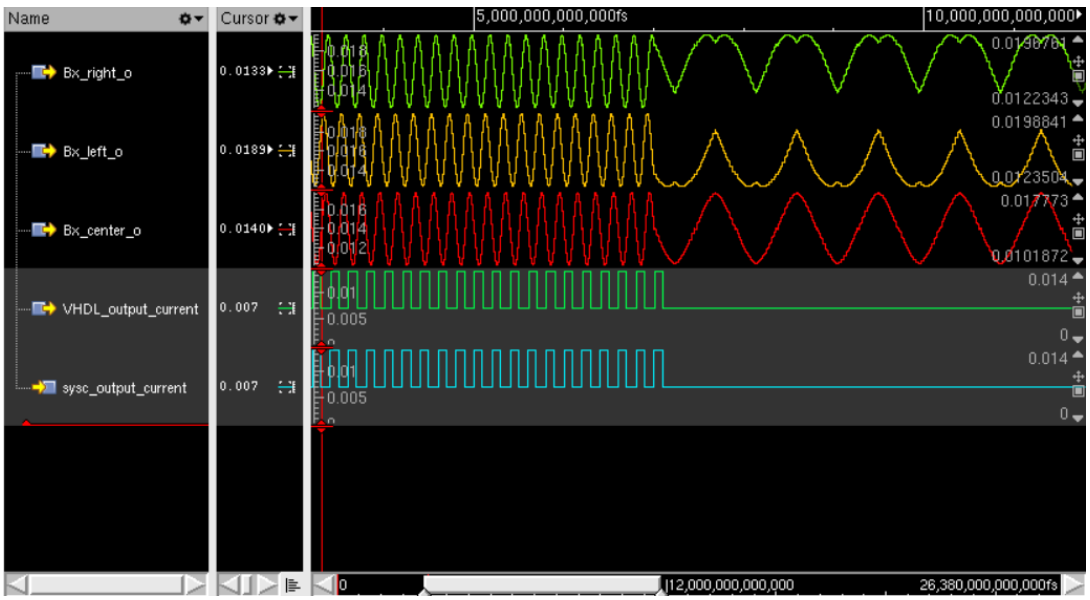

Figure 6 Cosimulation results example

The results revealed proper behavior of the VHDL translation, and a good match with the SystemC behavior, except for some small timing differences due to the way High Level Synthesis is performed by the tool. With the new tool Catapult® a full match has been obtained.

E. Synthesis on FPGA

Once the HDL code has been validated in the previous step, it is possible to move the design to the real hardware. The hardware currently in use in the laboratory consists of three boards:

- xMR board;
- Analog Front End (AFE) board;
- FPGA board.

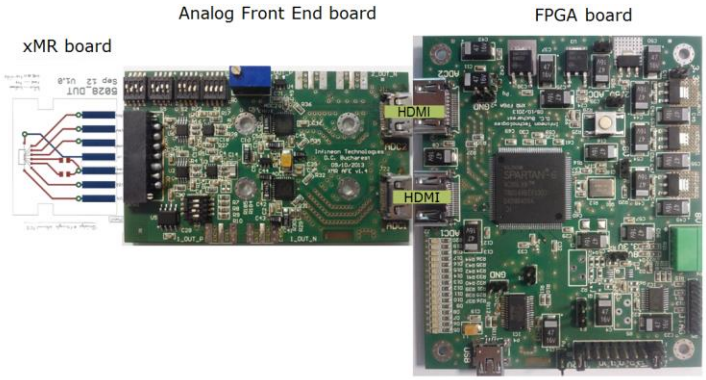

Figure 7 Hardware prototype: analog front-end and FPGA board

The xMR board consists of bare die silicon with of one or more sensing element (either GMR or TMR resistors) bonded on a small PCB. The PCB is then connected to the Analog Front End.

The AFE board models the analog front end of a speed sensor with a cascade of amplifiers and Analog to Digital Converters (ADC). This board is partially configurable using DIP switches: both the amplifier gains and the sensing element configurations can be changed.

Finally, on the FPGA board the digital core of our sensors is implemented. The board uses a Xilinx [12] Spartan 6 FPGA and provides some input and output pins for sensor configuration and debugging.

In the future, some other options could be considered, for example:

- Programmable front end instead of AFE board
- System on Chip (SoC) instead of simple FPGA

Various solutions are being analyzed at the moment, but PSOC5 [13] and Xilinx Zynq SoC based boards seem to be particularly well suited.

F. Automatic comparison between measurements and simulation results

Finally, as part of Fast and Furious methodology, an automated co-verification of the hardware is performed against the golden reference SystemC model. The validation methodology is described by the following workflow.

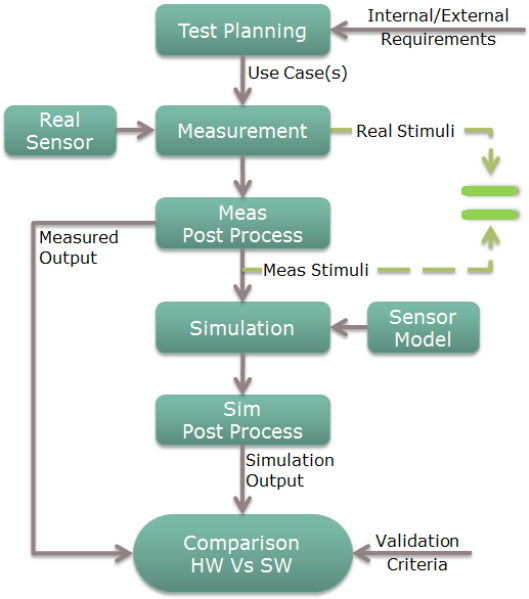

Figure 8 Co-validation methodology workflow

- ***Measurement:*** both the outputs and the stimuli of the system are measured (outputs come from the sensor while the magnetic stimuli are produced by the movement of the wheel).
- ***Simulation:*** for the simulation, the model is provided with the stimuli previously measured on the real test bench.
- ***Data Post Processing and final comparison:*** an algorithm performs a post processing of the data coming from both HW and SW for stimuli generation and final comparison purpose. Then, the post processed data is collected and assessed according to predefined validation criteria in order to crosscheck the hardware versus the model and find eventual mismatches to be fixed.

The following figure shows the Graphical User Interface (GUI) of the co-verification environment.

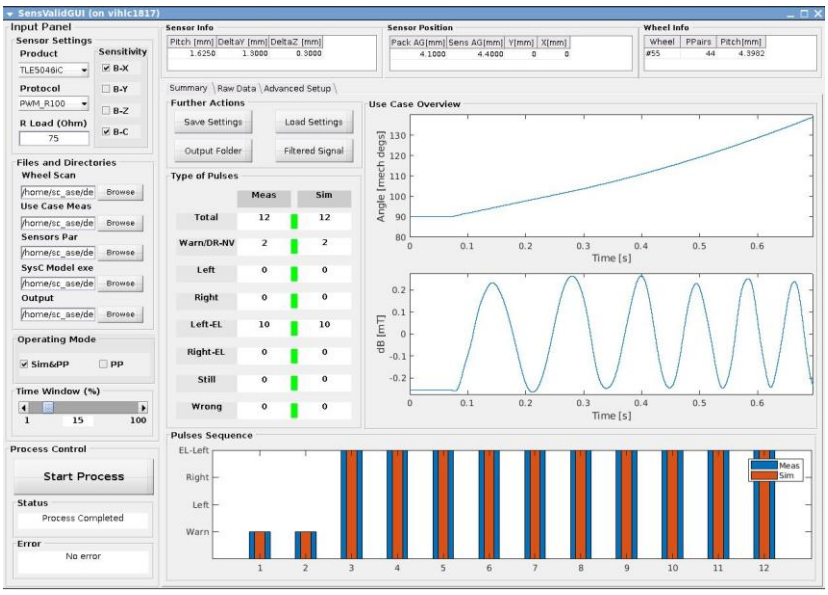

Figure 9 Sensor Validation Tool

III. RESULTS AND CONCLUSIONS

The methodology shown in this work is already in place, and is currently used for all next generation transmission sensors and for some other products. For example, the combined measurements and simulation method has already been very useful to support design-in activities with wheel speed sensors.

We have got to get the best out of our methodologies, being open-minded and willing to try new things and accepting the challenge to change the status-quo. Collaboration among different Infineon groups, with external partners and colleagues from other locations was also very important.

From an application engineering and business point of view, this methodology has enabled us to provid working prototypes to customers in a very short time, reacting fast to their requests. Moreover, it increases the level of confidence we achieve in our design, before a test chip or the final silicon is available. Concept, design and verification engineers can benefit enormously from this:

- The VHDL to be synthetized in the final silicon can be co-simulated with the reference SystemC model, reducing the number of bugs and the effort in verification;
- The prototype (and the final silicon) can be tested on laboratory test benches and the measured results can be compared with the simulations results.

In conclusion, we believe that virtual and real prototyping is a very useful and powerful tool, to validate the application requirements, via simulation and actual measurements. Using the proposed methodology, going from idea to final implementation, is faster and easier than with a standard approach. This allows us to react faster to customer requests, test new ideas and have more confidence in our design. The effort is spent doing the right things without wasting time doing repetitive tasks that could be automatized.

REFERENCES

[1] COSEDA Technologies GmbH, “COSIDE® - The Design Environment for Heterogeneous Systems”, in https://www.coseda-tech.com/

[2] C.P.O Treutler, “Magnetic sensors for automotive applications”, in Sensors and Actuators A: Physical, Volume 91, Issues 1-2, 5 June 2001

[3] R.S. Popovic, “Hall Effect Devices, Second Edition”, December 2003

[4] Reig, Candid, Cardoso, Susana, Mukhopadhyay, Subhas, “Giant Magnetoresistance (GMR) Sensors - From Basis to State-of-the-Art Applications”, 2013

[5] Hanmin Jin, Terunobu Miyazaki, “Tunnel Magnetoresistance Effect”, August 2012

[6] Michael J. Haji-Sheikh, Kristen Allen, “Anisotropic Magnetoresistance (AMR) Magnetometers”, in High Sensitivity Magnetometers, Volume 19 of the series Smart Sensors, Measurement and Instrumentation pp 167-199, September 2016

[7] Fontanesi, Kim, Binder, Monterastelli, “Mechanical mounting variation effects on magnetic speed sensor applications”, in DVCON2017

[8] Matlab, in https://de.mathworks.com/products/matlab.html

[9] Vivado HLS, in https://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html

[10] Catapult® High-Level Synthesis, in https://www.mentor.com/hls-lp/catapult-high-level-synthesis/

[11] Cadence Virtuoso, in https://www.cadence.com/

[12] Xilinx, Inc, in https://www.xilinx.com/

[13] PSoC® 5, in http://www.cypress.com/products/psoc-5