2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Virtual testing of overtemperature protection algorithms in automotive smart fuses

Thomas Markwirth[1], Gabriel Pachiana[1], Christoph Sohrmann[1],
Mehdi Meddeb[2], Gunnar Bublitz[2], Heinz Wagensonner[3]

[1] Fraunhofer IIS/EAS, Münchner Straße 16, 01187 Dresden, Germany
[2] CARIAD SE, Thurn-und-Taxis Straße 10, 90411 Nürnberg
[3] CARIAD SE, IN-Campus-Allee 18a, 85053 Ingolstadt

Abstract*— Electric vehicles are equipped with a complex network of electronics that operate at high currents and voltages. Protecting the vehicle's electrical system from damage caused by overcurrent and overtemperature is a crucial design requirement. Short circuits can have serious consequences, including potential fatalities. Therefore, all components within the E/E architecture must be safeguarded against hazardous conditions such as overvoltage, undervoltage, overcurrent, and overtemperature. At the same time, reducing the vehicle's weight and cost while increasing the battery range requires to minimize cable harness cross-sections. To achieve both goals, new devices like smart fuses are utilized, enabling more flexible and intelligent protection schemes. However, the increased functionality of smart fuses also brings about challenges in designing and verifying their performance, particularly under dynamically-changing load conditions. In this study, we propose a simulation-based verification methodology for overtemperature protection algorithms in automotive smart fuses. Our approach involves a co-simulation between the mixed-signal fuse and a thermal reduced-order model of the relevant part of the E/E architecture, using SystemC AMS. While the virtual Electronic Control Unit executes the target executable, the analog and thermal domains are simulated concurrently. The electrothermal model can be integrated into a Virtual Platform, facilitating a Software-on-Top workflow and the analysis of algorithms under complex, dynamically-changing currents.

Keywords—smart fuse; I2t; virtual prototyping; virtual testing; electrothermal; SystemC AMS; HW/SW co-simulation

I. INTRODUCTION

Smart fuses (SF) are a revolutionary development in circuit protection, playing a crucial role in various industries such as automotive, aerospace, and industrial automation [1]. These electronic protection devices integrate traditional overcurrent and short-circuit protection with advanced monitoring and control capabilities. SFs possess the ability to swiftly and accurately detect and respond to overcurrent or fault conditions, minimizing equipment downtime and mitigating the risk of damage. Furthermore, SFs offer capabilities that enable remote monitoring and control. Another pivotal focus lies in optimizing the range of electric vehicles, a parameter primarily constrained by battery capacity and longevity. A promising avenue to bolster vehicle range entails the reduction of overall car weight. This can be achieved through innovative strategies, such as diminishing the weight of the vehicle's electrical systems, particularly by downsizing the cable diameter within the E/E architecture. Simultaneously, this approach bears the potential to alleviate the burden on essential resources like copper and aluminium, thereby mitigating the automotive industry's susceptibility to external supplier dependencies. Due to the absence of degradation effects in SFs, service-life is significantly extended while reducing the plurality of components within the car.

In addition, SFs present an opportunity to employ more sophisticated methods for safeguarding the cable harness, as compared to conventional melting fuses. While conventional fuses rely solely on assessing the load current to make decisions about load disconnection, SFs introduce a more intelligent approach by considering parameters beyond just current, such as temperature of the cable insulator. Indeed, the increased functionality of SFs necessitates more advanced testing methods for both the hardware (HW) and software (SW) drivers. Power-Hardware-in-the-Loop (PHiL) simulation is commonly employed to verify standard scenarios of a SF protected system. However, assembling and conducting PHiL simulations can be time-consuming, especially when multiple scenarios need to be tested. Additionally, there is a potential risk of damaging the equipment or endangering the engineer during such physical tests.

Virtual testing [2] of SFs requires accurate electrothermal models due to the inherent coupling of thermal and electrical domains [3][4][5]. Figure 1 shows the domain interdependencies the model needs to capture.

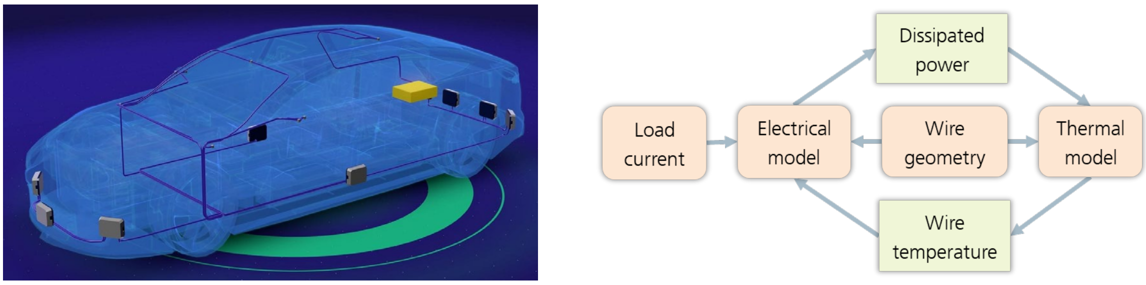

Figure 1: Modern vehicle architecture (left) and requirement for multi-physical model due to electro-thermal coupling (right).

II. FUSES FOR OVERCURRENT PROTECTION

Protection against overcurrent and overtemperature is crucially important for EVs due to the high voltage levels that can be dangerous for the driver but also damage the electronic component. Fuses are typically characterized by a time-current-curve (TCC) diagram where the points of the curve are the trigger time at a constant current level. As a compact measure for the reaction time of the fuse, the melting integral value can be calculated from the TCC as the product of tripping current squared times tripping time, commonly referred to as I2t (pronounced "i-squared-t") value. A small value corresponds to slow-blowing fuses whereas a large value indicates fast-blowing fuses. The concept of the melting integral is simply the time-integral of the internally dissipated power P, i.e. the thermal energy ΔQ generated in the fuse over time, as follows

$$\Delta Q(t) = \int_0^t P(\tau)d\tau$$

The thermal energy, i.e. heat, ΔQ gives rise to a temperature increase ΔT in the material according to the relation $\Delta T = \frac{\Delta Q}{C_{th}}$ with C_{th} being the thermal capacitance of the material. For a time-constant load current I running through an ohmic fuse with electrical resistance R_{el} the integral can be evaluated, yielding the well-known I2t relation

$$\Delta T(t) = \frac{R_{el}}{C_{th}} I^2 t$$

Using the tripping current/time from the TCC yields the previously introduced melting integral value as a measure for the reaction time of the fuse. The value of the melting integral of the fuse needs to be smaller than the melting integral of the electrical component protected, such that the fuse trips before the component overheats.

In an automotive cable harness, there is a need to address multiple levels of protection, which necessitates flexibility in fuse selection. SFs offer this flexibility by enabling the adjustment of tripping conditions, allowing them to function as either slow-blow or fast-blow fuses based on specific requirements. This adaptability ensures that the fuses respond appropriately to different fault conditions. Moreover, SFs allow for the adjustment of algorithms used to estimate the load profile of the component being protected. These algorithms are typically derived from a series of current measurements taken over time. The two major challenges for the design and parametrization of those algorithms are:

a. Selection of the correct tripping points

b. Accurate load estimation based on current measurements

By employing an electrothermal model of the component under protection, it becomes possible to meticulously design and select the tripping point and load estimation algorithms that precisely align with the specific requirements of the vehicle architecture.

III. ELECTROTHERMAL MODELLING METHOD

Modeling SFs for overtemperature protection requires an electrothermal model that allows for co-simulation of thermal and electrical effects. This is because the electrical properties are temperature-dependent, and both domains need to be simulated simultaneously. In industrial development processes, these domains are typically kept separate with distributed responsibilities. However, an electrothermal model can enhance the separation of concerns and reduce the need for frequent feedback and communication cycles during the development process. When constructing a thermal model for multi-physical simulation, there are various options available. Table I provides an overview of the thermal model types used in this work for temperature estimation.

Table I: Types of models used in this work for temperature estimation

Model name	Abbreviation	Method of model creation	Type of model	Accuracy
Full-order model	FOM	FEM software	LTI system (large)	High
Reduced-order model	ROM	Model order reduction	LTI system (small)	Medium
Equivalent circuit model	ECM	Lumped system synthesis	RC netlist (small)	Low
Melting integral model	I2tM	Known from literature	Analytical expression	Coarse upper bound

The FOM can be quite complex, incorporating advanced geometric and physical features, such as bundled wires, additional electrical devices, complex boundary conditions, or temperature-dependent material properties. Any co-simulation with such a complex model is very time-consuming. Therefore we apply a model order reduction (MOR) technique to automatically generate a simplified reduced-order model (ROM) that requires less compute resources while still being accurate enough [8]. In some cases, a reduced equivalent circuit model (ECM) can also be generated by lumped system analysis [5] and parameter fitting. Figure 2 depicts the process of creating the model using FEM software and transferring it into a model compatible with a system simulation in a Virtual Platform (VP) software, enabling co-simulation with a virtual ECU.

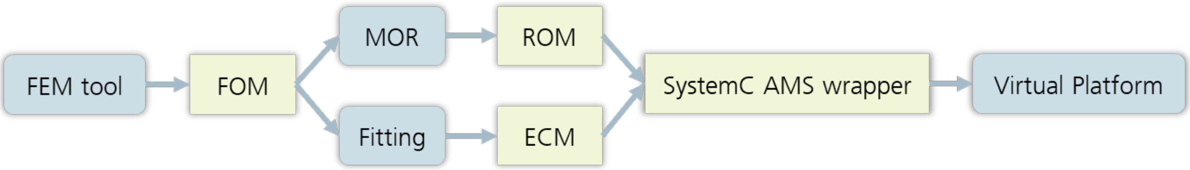

Figure 2: Generation flow for multi-physical model showing tool/algorithm (blue) and model type (green).

IV. CABLE MODEL EXAMPLE

To assess the accuracy and simulation efficiency of algorithms, a reference model is employed. This reference model consists of a one meter long PVC-insulated, single-wire copper cable. Figure 3 illustrates the geometry of the cable example and displays a reference solution obtained through simulation in COMSOL Multiphysics.

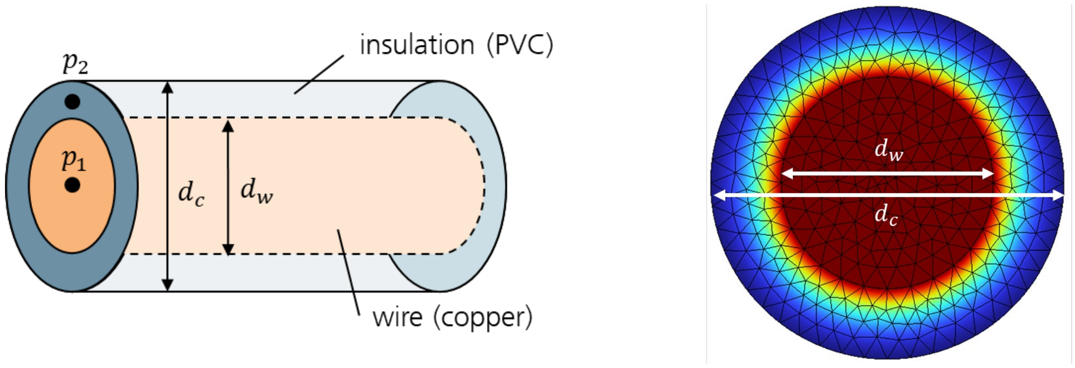

Figure 3: Geometry of cable with two observation points (left), temperature distribution over the cross-section (right).

The cable parameters used in the model are loosely based on established cable norms and standards [6][7], primarily selected for demonstration purposes. Table II presents the key parameters of this model. However, these parameters may require adjustment to accurately reflect real-world application conditions.

Table II: Cable parameters used in this work

Parameter	Value
Effective wire cross-sectional area	$A = 6\ \mathrm{mm}^2$
Cable diameter (copper+PVC)	$d_c = 5\ \mathrm{mm}$
Wire diameter (copper)	$d_w = 3\ \mathrm{mm}$
Cable length	$l_c = 1\ \mathrm{m}$
Electrical resistance	$R = 2.5\ \mathrm{m}\Omega$
Maximum temperature	$T_{max} = 125°C$
Heat transfer coefficient (thermal boundary condition)	$h = 5 \frac{\mathrm{W}}{\mathrm{m}^2\mathrm{K}}$

In a reference simulation, the temperature response caused by a rectangular pulsed wave load current switching between 10A (nominal current) and 100A (faulty current) has been analyzed. Figure 4 shows the temperature at two points in the wire and the insulation (as depicted in Figure 3) as well as the I2t integral for comparison. The corresponding TCC diagram is presented on the right hand side of Figure 4.

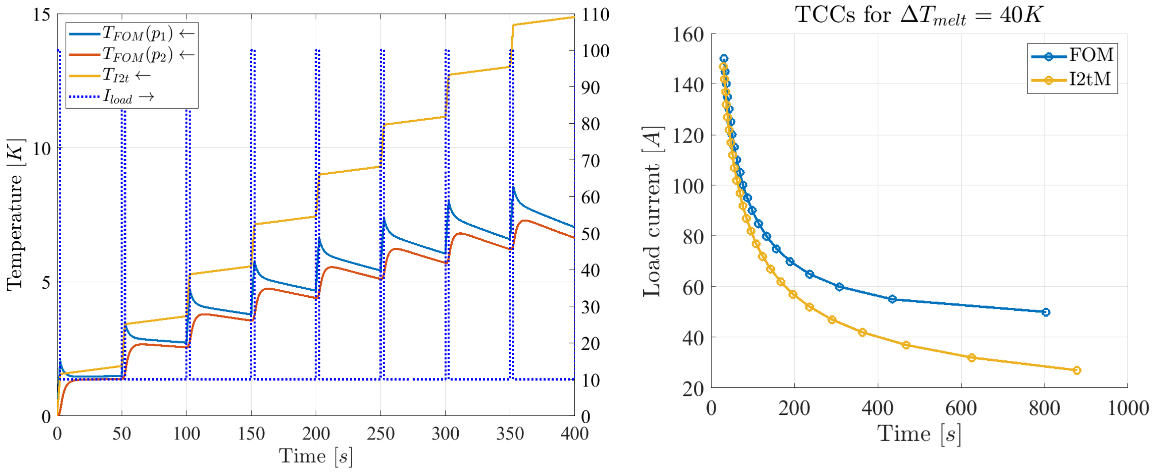

Figure 4: Temperature over time for FOM and I2tM for a periodic rectangular load (left) and TCCs for both models at $\Delta T_{melt} = 40K$ (right)

The TCC obtained from the simple I2t model shows good agreement for high currents. However, significant deviations are observed at lower currents due to the inadequate representation of heat loss to the environment in the simple analytical I2t model. To improve the I2t model, it is necessary to consider not only the stored thermal energy but also the heat loss to the environment. The thermal energy can be expressed as $\Delta Q = \Delta Q_{in} - \Delta Q_{out}$ where ΔQ_{in} represents the heat generated by the resistive load as defined before, and the new term ΔQ_{out} is determined by the rate of change according to Newton's law of cooling as $\Delta \dot{Q}_{out} = h \cdot A \cdot (T - T_{env})$. The resulting differential equation is typically replaced by a window integration of the dissipated power over a relevant time-window Δt_w, which can be written as follows

$$\Delta Q(t) = \int_{t-\Delta t_w}^{t} P(\tau) d\tau$$

The introduction of the time-window in the model acknowledges the finite memory of heat. It recognizes that heat generated in the distant past is not significant for the current temperature. However, it is important to note that this model introduces an additional parameter, namely the window length, which needs to be determined manually. The window length determines the relevant time span over which the heat dissipation is considered.

V. SMART FUSE MODEL

The SF under investigation is based on the IFX BTS70015-1ESP datasheet as smart high-side power switch. The fuse functionality is driven by a microcontroller based control logic. Figure 5 shows the schematic view of the SF model created in COSIDE IDE [11].

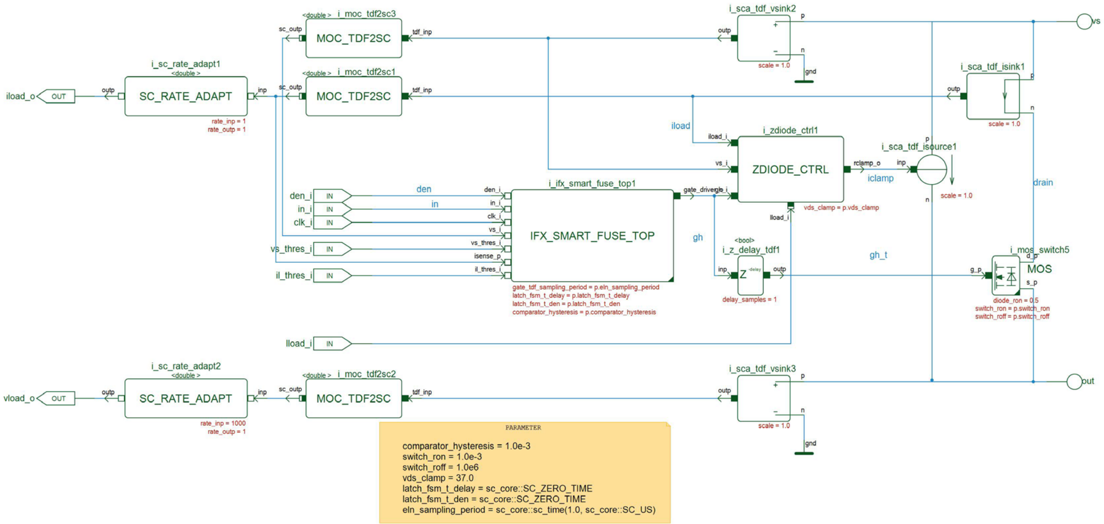

Figure 5: Smart power switch as the mixed-signal part of the smart fuse model

The smart fuse model is primarily divided into two parts: a software-based controlling algorithm running on an ECU and a mixed-signal smart power switch, shown in Figure 5. The smart power switch interacts with the analog supply and load circuits and incorporates low-level safety mechanisms. It also communicates important information, such as the load current value, to the ECU. The software-based controlling algorithm on the ECU is responsible for managing the operation of the smart power switch and implementing higher-level protection mechanisms.

VI. VIRTUAL TESTING METHODOLOGY

The functional verification of the microcontroller-based protection for the SF will focus on dynamically-changing loads, encompassing the SF's operation under normal and fault conditions. These conditions include high-current scenarios (representing normal operation near the maximum current), overcurrent situations (to test the activation of the protection mechanism and measure response time), and overtemperature conditions (to assess overheating caused by high temperatures). In this verification, a rectangular pulsed wave current load will be utilized, alternating between a nominal value of 10A and a faulty value of 100A. The pulse width will be set to $T_{pulse} = 40s$, with the faulty current occurring at three different duty cycles of $\{15\%, 50\%, 100\%\}$. Table III presents the algorithms under investigation that are tested on the described load profiles.

Table III: Load estimation algorithms under investigation

Algorithm	Description	Output
ALG-1	Numerical integration of I2t value	Temperature
ALG-2	Binned digital counter	Tripping time
ALG-3	Novel physics-based algorithm	Temperature

The particular details of the algorithms are not relevant for the methodology, but a short summary of each algorithm will be given in the following. In ALG-1, the I2tM is integrated numerically over a predetermined time-window Δt_w. The discretized summation formulation for time t_N reads as follows

$$\Delta T(t_N) \approx \frac{R_{el}}{C_{th}} \sum_{k=N-n}^{N} I_k^2 \cdot (t_k - t_{k-1})$$

The summation runs only over the past n time points that occurred in the time-window Δt_w. However, for the computation of this sum, the previous n values need to be kept in memory. It is anticipated that ALG-1 will have demanding memory requirements and generate a high processor load due to the use of floating-point arithmetic.

On the other hand, ALG-2 avoids storing many previous values or the use of floating-point arithmetic by grouping the expected current values into bins and utilizing a digital counter to estimate the thermal load on the wire. However, a drawback of ALG-2 is that it cannot provide temperature estimates but only signals the tripping time.

ALG-3 is a newly developed physics-based algorithm that offers the benefits of accurate temperature estimation while maintaining low processor requirements. The testing methodology will determine whether ALG-3 meets the performance requirements set for the microcontroller-based protection system.

A. Standalone simulation in SystemC AMS

Typically, during the early stages of the design process, a new algorithm is evaluated and its functional correctness is established in the concept phase. At this stage, the final realization, such as the SW application and the associated processor platform, may still be uncertain and therefore not taken into consideration. Instead, a purely algorithmic model is used for evaluation purposes. In this case, the entire system model is created as a SystemC AMS [9][10] model and simulated in a standalone mode. Figure 6 presents a schematic view of the model in COSIDE IDE.

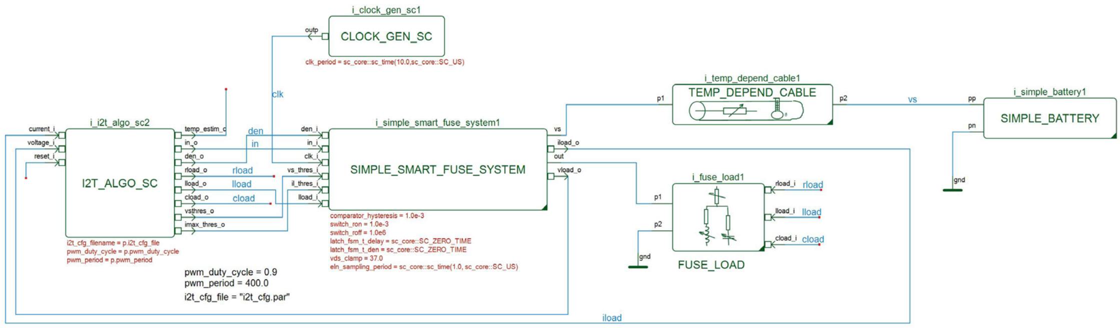

Figure 6: Test bench for evaluating the temperature estimation and tripping algorithms

Figure 7 shows the output traces of the test benches, including the input current waveform, the temperature estimates whenever possible, as well as the tripping signal triggered by each model. The melting temperature and melting time calculated from the ROM is also indicated.

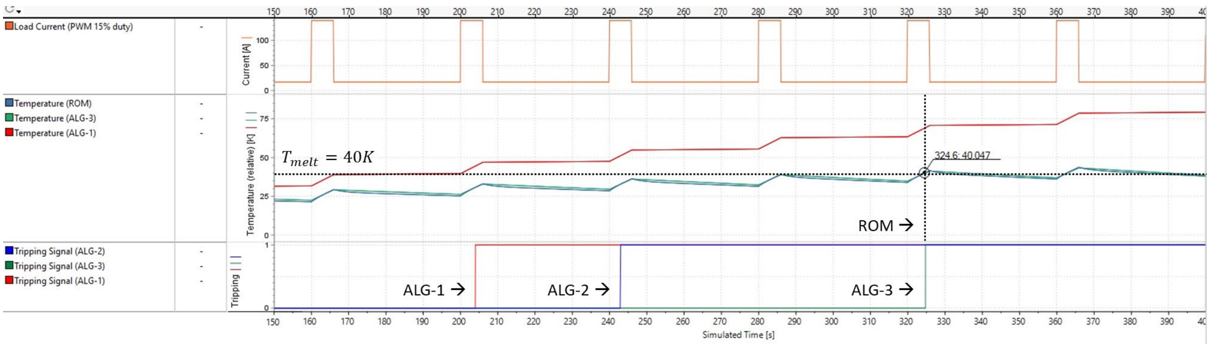

Figure 7: Comparison between reference model (ROM) and estimation algorithms for load signal with duty cycle D=15%

Based on all different load profiles described earlier, the tripping time of each algorithm is compared to the expected value from the electrothermal simulation: ALG-1 and ALG-2 systematically overestimate the temperature and trigger the fuse earlier than needed. Only ALG-3 accurately predicts the real temperature crossing with the critical temperature for all load profiles. Table IV gives a summary of these results including a qualitative assessment of the processor load implied by the algorithm based on instruction-counting in the virtual ECU.

Table IV: Results of the investigation regarding accuracy of tripping time and processor load

Algorithm	Accuracy of tripping time	Compute and memory requirements
ALG-1	Very low for medium to low load currents	High
ALG-2	Low for medium to low load currents	Very low
ALG-3	Very high for all load profiles	Low

B. Model-integration into the Virtual Platform

Once the correctness and trigger accuracy of the algorithm has been established, it becomes possible to evaluate more advanced scenarios where the ECU manages multiple SFs simultaneously. When handling the algorithms of 100+ SFs by multitasking, a critically high processor load and power consumption for the ECU can be expected. To assess the dynamic runtime behaviour on the target hardware in a more realistic setting, the microprocessor platform needs to be defined. At this stage, SW application development for the target HW commences and models of the peripherals are required. It is essential to demonstrate that the processor system can meet the expected functionality and performance requirements within the VP simulation. Cycle accurate models allow for direct integration of target software. Additionally, economic considerations can be taken into account, such as whether a platform with fewer resources can still deliver the expected functionality, particularly under high loads. For the example use case, the investigations involves integrating the mixed-signal SF model with the ROM of the cable.

For running the extended virtual prototyping simulations, the model presented was integrated into the design flow using a Synopsys Virtualizer environment. This was achieved through a co-simulation approach using the Synopsys VSI interface. Alternatively, the SystemC AMS model can be directly integrated into the Synopsys Virtualizer environment. Figure 8 shows the co-simulation scenario of the SystemC AMS model and Virtualizer.

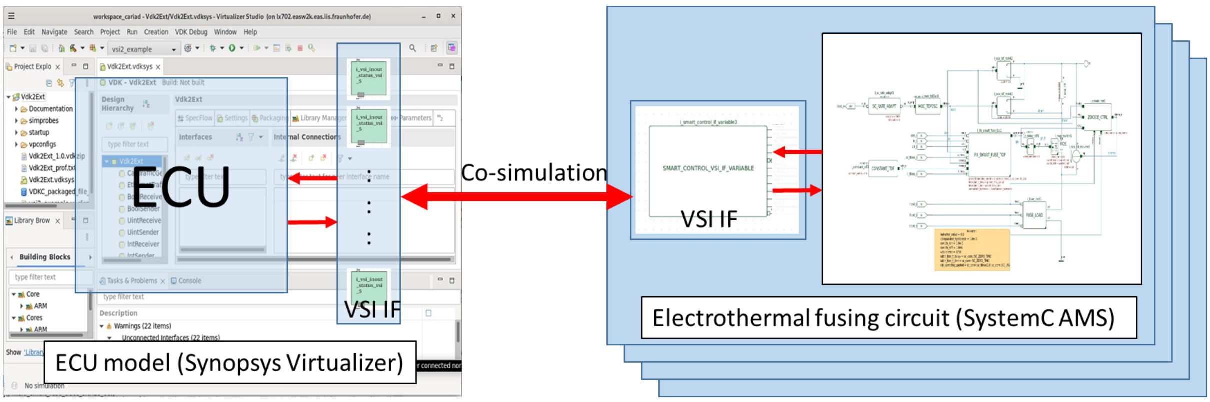

Figure 8: VSI-based co-simulation of virtual ECU in Synopsys Virtualizer (left) and multiple SystemC AMS fusing circuit models (right)

VII. Summary and Outlook

In this study, a virtual testing methodology for software drivers used in automotive smart fuses for overtemperature protection is presented. This methodology utilizes a mixed-signal, electrothermal modelling approach based on SystemC AMS, enabling comprehensive testing of the software implementation in terms of correctness, accuracy, timing, and processor load. The accuracy of the tripping behavior is validated against an analog electrothermal model of the component protected by the fuse. To demonstrate the methodology, a section of copper cable is modeled in COMSOL Multiphysics. The model is then exported, reduced using a model-order reduction algorithm, and integrated into SystemC AMS. This SystemC AMS model is subsequently co-simulated with a virtual Electronic Control Unit running in Synopsys Virtualizer, generating dynamically-changing load profiles and driving a large amount of fuses. Our investigation confirms the correctness of all tested algorithms, with one particular physics-based algorithm outperforming others. By utilizing the presented virtual testing model for smart fuses, the design and validation process is significantly accelerated, while ensuring safety. The methodology involves co-simulation of a thermally-aware analog load model, including a possible temperature-dependent resistance, with a virtual prototype.

In the future, it is envisioned that the proposed methodology will be applied to a wider range of components within the E/E architecture. To achieve this, an advanced parametric reduced-order model can be derived from the full-order finite-element model, accurately incorporating essential geometric and physical parameters. By utilizing this parametric reduced-order model, the iteration process between Virtual Prototype development and FEM modeling can be significantly reduced, simplifying virtual testing for various scenarios. The results of this ongoing work will be presented in a subsequent publication.

VIII. ACKNOWLEDGEMENT

This work has received funding from the German Federal Ministry of Education and Research (BMBF) within the project VE-VIDES (16ME0245).

IX. REFERENCES

[1] R. Letor and R. Crisafulli, "Smart Power devices and new electronic fuses compliant with new E/E architecture for autonomous driving," 2019 AEIT International Conference of Electrical and Electronic Technologies for Automotive (AEIT AUTOMOTIVE), Turin, Italy, 2019, pp. 1-6, doi: 10.23919/EETA.2019.8804538.

[2] Einwich, Karsten. "Virtual prototyping for smart systems for electric, safe and networked mobility." Advanced Microsystems for Automotive Applications 2011: Smart Systems for Electric, Safe and Networked Mobility. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011. 305-314.

[3] S. Önal and S. Frei, "A model-based automotive smart fuse approach considering environmental conditions and insulation aging for higher current load limits and short-term overload operations," 2018 IEEE International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC), Nottingham, UK, 2018, pp. 1-6, doi: 10.1109/ESARS-ITEC.2018.8607390.

[4] R. S. Olsen, J. Holboll and U. S. Gudmundsdóttir, "Dynamic temperature estimation and real time emergency rating of transmission cables," 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 2012, pp. 1-8, doi: 10.1109/PESGM.2012.6345324.

[5] S. Önal, A. Henke and S. Frei, "Switching Strategies for Smart Fuses Based on Thermal Models of Different Complexity," 2020 Fifteenth International Conference on Ecological Vehicles and Renewable Energies (EVER), Monte-Carlo, Monaco, 2020, pp. 1-10, doi: 10.1109/EVER48776.2020.9243932.

[6] VW 60306-1: Electrical Cables in Motor Vehicles, Part 1: Copper Cable; Single-Core, Unshielded, Issue 2018-09

[7] Thermal Simulation Models. ZVEI (German Electrical and Electronic Manufacturers' Association), 2020

[8] Köhler, Andreas, Sven Reitz, and Peter Schneider. "Sensitivity analysis and adaptive multi-point multi-moment model order reduction in MEMS design." Analog Integrated Circuits and Signal Processing 71 (2012): 49-58.

[9] IEEE Standards Association. IEEE 1666-2011 - IEEE Standard for Standard SystemC Language Reference Manual. DOI: http://standards.ieee.org/findstds/standard/1666-2011.html

[10] IEEE Standards Association. IEEE 1666.1-2016 - IEEE Standard for Standard SystemC(R) Analog/Mixed-Signal Extensions Language Reference Manual. DOI: http://standards.ieee.org/findstds/standard/1666.1-2016.html

[11] COSEDA Technologies. SystemC AMS Proof-of-Concept 2.3.4. DOI: http://www.coseda-tech.com/systemc-ams-proof-of-concept