2015 IEEE/ACM 19th International Symposium on Distributed Simulation and Real Time Applications

Power Estimation of an ECDSA Core applied in V2X Scenarios using Heterogeneous Distributed Simulation

Harald Bucher, Alexander Klimm, Oliver Sander, Juergen Becker
Karlsruhe Institute of Technology (KIT)
Institute for Information Processing Technology (ITIV)
Karlsruhe, Germany
Email: {bucher, alexander.klimm, oliver.sander, becker}@kit.edu

***Abstract*—Embedded systems are steadily growing in complexity and nowadays power consumption additionally plays an important role. Designing and exploring such systems embedded in its environment demand for holistic and efficient simulations. In this work we use a simulation framework based on the HLA (High-Level Architecture) and the modeling tool Ptolemy II to enable complex heterogeneous distributed simulations of embedded systems. In this context, we introduce a co-simulation based power estimation approach by integrating domain-specific simulators as well as off-the-shelf HDL simulator and synthesis tools. This enables cross-domain interaction and generation of realistic on-the-fly stimuli data for Register Transfer Level and Gate Level models as well as the gathering of power estimation data. We apply the framework to a Vehicle-2-X scenario evaluating an ECDSA signature processing core which ensures trustworthiness in vehicular wireless networks. To evaluate dynamic power reduction possibilities on application level we additionally introduce a V2X Message Evaluation technique to reduce signature verification efforts. It shows how realistic on-the-fly stimuli data obtained by the framework can improve the exploration and estimation of dynamic power consumption.**

***Keywords*—High-Level Architecture, Co-Simulation, Vehicle-to-X, E/E-Architecture, ECDSA, Power Estimation, Register Transfer Level, Gate Level**

I. Introduction

Embedded systems are steadily growing in complexity. Especially in the automotive domain the electric/electronic (E/E) architecture exists of a distributed network of embedded systems and dozens of sensor and actuators interacting with each other. Such interacting systems consists of heterogeneous components like analogous sensors/actuators, digital hardware/software, signal processing etc. This situation is even more aggravated if modern technologies like Vehicle-to-X (V2X) communication is incorporated. V2X communication forms the basis for a lot of cooperative applications to improve road traffic safety, optimize traffic flow and manage Intelligent Transportation Systems (ITS).

Typically, each system component is designed in its own domain and simulated with the use of a proper *Model of Computation* (MoC) [1], e.g. Continuous Time or Discrete Event. However, to design and evaluate the holistic behavior of heterogeneous distributed embedded systems together with complex application scenarios like V2X, it is no longer sufficient to separately use individual domain-specific MoCs. There is rather the need for methods and tools to provide a continuous heterogeneous modeling and simulation flow [2] in order to consider cross-domain interactions and influences between subsystems on-the-fly.

Another evolving yet important issue in embedded systems design is reducing power consumption. In the automotive domain, for example, to minimize the cross section of the wiring harness in order to reduce production costs. Therefore, early yet reliable power estimations are necessary to avoid expensive system re-designs because of requirement violations detected in late design phases. Nowadays, the static power consumption because of increasing leakage currents are not negligible anymore. However, the highest part of the power consumption origins from transistor switching and short-circuit currents of CMOS circuits (dynamic power). Therefore, realistic stimuli scenarios on application level are mandatory. Furthermore, closed-loop feedback influences between the model components to be evaluated (consumer) and environmental/subsystem models (producer) are mandatory to get representable power estimations [3] - independent of the abstraction level of the design description, i.e. Electronic System Level (ESL), Register-Transfer-Level (RTL) or Gate-Level (GL). This applies especially to complex and highly dynamic scenarios like V2X-based ITS or cooperative awareness applications.

In this work, we present an extension to our developed distributed heterogeneous design and simulation framework [4]. The extension is the integration of an signature processing core for V2X messages. It enables the investigation of heterogeneous embedded systems in a holistic, distributed and scalable manner. We apply the framework to an abstract automotive E/E architecture embedded in dynamic V2X communication scenarios. In this context, we also present a novel co-simulation based power estimation flow using the simulation framework in combination with domain-specific simulators and commercial off-the-shelf synthesis and simulation tools. Thereby, we rather focus on the scalability as well as on cross-domain interaction and generation of stimuli data than on fast individual SystemC [5] based ESL power models (see Section II). Regarding the latter, to cope with simulation complexity and run-time we rely on distributed simulation. Nevertheless it should be mentioned that it is conceivable to include such instrumented SystemC based ESL models into our framework (see Section III-A).

We adopt our power simulation framework to the signature

1550-6525/15 $31.00 © 2015 IEEE
DOI 10.1109/DS-RT.2015.35

processing core which ensures trustworthiness in V2X environments. A *Cooperative Adaptive Cruise Control* (CACC) application running on the E/E architecture together with a detailed vehicular network simulator are used to provide cross-domain realistic stimuli data for power estimation on RTL and/or GL. To evaluate dynamic power reduction possibilities on application level we introduce a V2X Message Evaluation technique to reduce signature verification efforts. It shows how on-the-fly stimuli data obtained by the framework can improve the exploration and estimation of dynamic power consumption.

This work is organized as follows: Sec. II gives some related works. Sec. III provides background information on heterogeneous distributed simulation and the need for secure V2X communication. In Sec. IV we describe the V2X message evaluation approach. Sec. V proposes the general structure of our co-simulation based power estimation framework and the concrete application. Sec. VI presents results. Finally, Sec. VII concludes and gives an outlook to future work.

II. Related Work

Regarding the design and simulation of heterogeneous embedded systems, big efforts were spent in recent works. The authors in [6] present a heterogeneous simulation framework also based on Ptolemy II which implements a middleware to connect a subset of PtII MoCs with Matlab/Simulink and other automation specific simulators by tool integration. However, the middleware cannot deal with distributed simulation of such systems. In [7], [8] the authors introduce a programming model for distributed real-time cyber-physical systems but distributed simulation of that MoC is not yet supported. With both, the scalability is hard to ensure as model size increases. Heterogeneous modeling in combination with distributed simulation is addressed e.g. in [9], [10], where they make use of the HLA for distributed simulation of two the MoCs discrete event and continuous time with Matlab/Simulink and DEVSim++ respectively. However, they only support the distributed simulation of two MoCs thus providing no support for the integration of further MoCs, which is possible with PtII and the HLA.

In [11] the authors also integrate PtII with the HLA. But they rather focus on the interoperability and distributed simulation of different applications modeled within PtII. A different approach is used, implementing the synchronization with two actors based on the publish/subscribe mechanism of the HLA.

Regarding power estimation of embedded systems, the trend goes towards early design space exploration and power estimation on ESL with SystemC at transaction level (TL) to speed up design cycles. As SystemC don't support native power modeling techniques, several approaches were proposed concerning the construction of proper power models. The authors in [12],[13] deal with SystemC code instrumentation and back-annotation of power and/or switching information gathered from lower-level estimations to integrate and calibrate it into the high-level power models. However, realistic stimuli generation and cross-domain interactions for representable power estimations on lower levels is not addressed. Also, the distributed simulation of such models is not issued.

The authors in [3] also dealing with co-simulation based power estimation for hardware/software systems-on-chip designs. This system-level co-simulation involves the integration of the former heterogeneous simulation tool Ptolemy I as simulation master with other hardware/software simulators at different design levels. However, they don't tackle the integration and distributed simulation of complex domain-specific simulators.

III. Background

A. Heterogeneous Distributed Simulation

To fill the lack of existing tools for a continuous heterogeneous design flow and distributed simulation in [4] we propose a novel extensible simulation tool chain that supports managing heterogeneous model composition and distributed co-simulation. The framework is composed of Ptolemy II (PtII) [14] and a simulation middleware based on the HLA [15] enabling distributed discrete event co-simulation with domain-specific simulators. In the HLA terminology the logical representation of an interconnection of different simulators is called a *federation* which includes one or multiple simulators called *federates*. PtII serves as a central design tool realizing two main goals: I) user interface for support of tool integration and configuration of control and data flow interaction via the HLA and II) central modeling tool managing heterogeneous model composition inherently for performing architectural exploration, verification and validation. A simulation library provides high-level interfaces (*HLA Interface Wrapper*) which eases integration and encapsulates control and data flow between the HLA middleware, PtII and domain-specific simulators. The necessary interfaces and HLA specific files for a certain federation and its federates to be integrated can be generated semi-automatically out of a PtII model specification.

With the help of this tool chain a framework was developed facilitating exploration, validation and verification of V2X-enabled E/E architectures. An abstract E/E architecture was modeled inside PtII. It is based on future design concepts like centralized computer architecture and standardized communication. To simulate wireless V2X network traffic and environmental road traffic accurately, Veins [16] was integrated into the co-simulation. Veins is a vehicular network simulator that integrates the OMNeT++ [17] network and the SUMO [18] traffic simulator. Detailed models for the IEEE 802.11p [19] and IEEE 1609 *Wireless Access in Vehicular Environments* (WAVE) [20] protocol stack are provided for simulating V2X communication. Furthermore, individual abstract modules in PtII can be refined from ESL down to RTL using SystemC. The integration of domain-specific simulators is done within special developed PtII actors called `HlaComposite`. These composites again contain special actors and directors responsible for configuring and conducting the control and data flow interactions via the HLA. Fig. 1 shows the simplified framework (without SystemC refinement) with one `HlaComposite`.

B. Secure V2X Communication: ECDSA

V2X communication enables great potential for e.g. improved safety [21] [22], energy efficiency [23] and infotainment [24]. Communicating the information is standardized by the IEEE 802.11p and IEEE 1609 WAVE protocol stack. In order to minimize protocol overhead and to reduce latency, WAVE defines a new protocol and message type especially

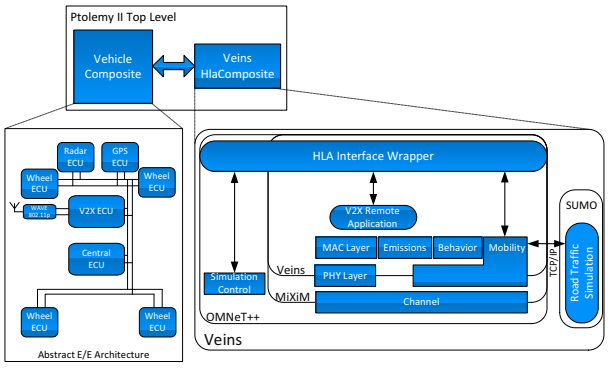

Fig. 1. Modelling and Simulation framework used for investigating V2X-enabled E/E architectures

for safety applications called *WAVE Short Message Protocol* (WSMP) and *WAVE Short Message* (WSM) respectively. Usually every vehicle is periodically broadcasting WSMs with its current state to its surrounding neighbors at a frequency of $2 - 10Hz$. Typically these messages are called *beacons*. Additionally event-driven messages can be sent to notify about special and dangerous traffic situations.

To realize such cooperative applications based on information from other vehicles and infrastructure, trustworthiness of this information is mandatory. To realize this, signature schemes are applied to protect the messages broadcasted against malicious attacks. In IEEE 1609.2 [25] the security layer of WAVE is specified and uses the *Elliptic Curve Digital Signature Algorithm* (ECDSA) [26] to guarantee validity and authenticity of a signed message to the receiver and prove the identity of the sender. ECDSA is an asymmetric signature scheme based on elliptic curves over a finite field. IEEE 1609.2 recommends and supports the two elliptic curves *p224* and *p256* based on prime fields with key lengths of 224 bit and 256 bit respectively. This allows much smaller key lengths than other asymmetric schemes like RSA [26] with comparable security strength. Also it gets along without key exchange and predistributed keys compared to symmetric ciphers. This benefit comes at the cost of much more computational complexity which imposes major problems for embedded systems with limited resources. Especially in V2X scenarios high throughputs of up to 2500 signatures per second are necessary [27].

IV. V2X Message Evaluation

Since the signature processing is a computation intensive and time consuming task [28] the number of messages to be processed during run-time should be minimized in order to reduce and meet not only the latency but also the power consumption requirements of the signature processing module. Especially on the receiving path of V2X messages this can be accomplished by only accepting messages currently relevant to the vehicle.

Therefore, in [29] we propose a V2X message evaluation methodology based on vehicle clustering and monitoring capabilities of both vehicle and internal/wireless network utilization. The approach has special focus an safety applications. It is modeled as part of a V2X Electronic Control Unit (ECU) together with a *Cooperative Adaptive Cruise Control* (CACC) safety application as well as with abstract E/E architecture components. The goal is to reduce internal network traffic of the E/E architecture to meet application's real-time constraints. This is done by only accepting (safety) relevant messages dependent on the current traffic situation and network utilization. In the following the basic concepts of the message evaluation approach and the necessary ECDSA extensions used in this work are presented.

A. Baseline Design

The V2X message evaluation methodology presented in [29] is realized as an intelligent gateway in a dedicated V2X ECU responsible for V2X related data processing. It is modeled within an abstract ECU model as illustrated in Fig. 2 and is mainly made up of five components:

1) *WAVE Interface* for decoding and encoding WSMs.
2) *E/E architecture interface* providing data to/of other ECUs (e.g. sensor and GPS data of local/remote vehicle).
3) *WSM Scheduler* for efficient selection and transmission of local application data or forwarding of safety critical incoming WSMs.
4) *Hierarchical Message Evaluation Stages* as core component for data fusion and message evaluation/filtering.
5) *Network Monitoring* capabilities allowing evaluation of both WSM channels and internal E/E network utilization.

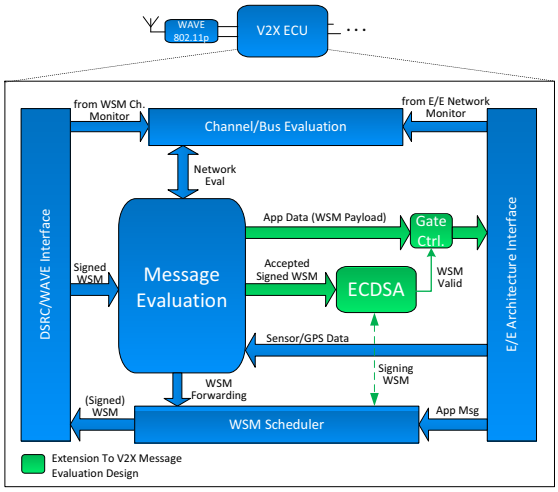

Fig. 2. V2X Message Evaluation Model with ECDSA Signature Processing Extensions

The methodology adopts a decentralized, cluster-based message evaluation using a classification into a *Farfield Network* (FFN) and *Nearfield Network* (NFN) database of surrounded vehicles. The approach is completely decentralized as it only relies on data which are part of a beacon broadcast message encapsulated in a WSM. Regarding the clustering, *NFN* considers all vehicles in close proximity within a parametrized position range, mainly dependent on the current vehicle state. All vehicles within that range which fulfill a certain acceptance policy becomes a node in NFN. All messages are considered for evaluation, since they potentially contain information for safety-critical situations and applications, e.g. CACC. In contrast, *FFN* uses a cluster-based approach, where vehicles beyond the NFN area are clustered and are represented by a

cluster master vehicle which in turn has its own NFN. Information from FFN is potentially used for long-term optimizations or warnings (e.g. premature reaction to an accident).

According to the classification into FFN and NFN, the *Message Evaluation* building block in Fig. 2 is refined towards that classification. There are hierarchical evaluation stages responsible for FFN and NFN. Both comprise two main components: A *Data Fusion Unit* and a *Filter Unit*. The *Data Fusion Unit* collects necessary sensor data from other ECUs in the E/E network, mainly GPS, speed and radar data, incoming WSMs, and evaluation data of the external/internal communication monitoring components. The reasons for the hierarchical evaluation approach are to handle the complexity of the overall evaluation (divide and conquer), a better modularity and prefiltering of non-safety messages in high-density and critical traffic situations. Out of this data fusion, dedicated *acceptance policies* for FFN and NFN are specified. They are dynamically adapted mainly dependent on the current vehicle state and monitored network utilization. In the *Farfield Evaluation Stage* the acceptance policies are additionally influenced by a FFN cost function, weighting the relation between the own vehicle and each cluster master vehicle in the FFN. This cost function is used to determine potential dangerous traffic situations, which are not within spitting distance. The actual *Filter Unit* uses those policies for comparisons of data appearing in received WSM in order to pass it into the nearfield evaluation stage, into the internal E/E network or discard it. In the baseline design no WSMs are forwarded at the output of the *Nearfield Evaluation Stage*, but application specific data frames, i.e. the WSM payload. They contain the most relevant application specific data of the nodes within the NFN. Therewith, an application specific cost function is to be defined for each application, which weights the state of each node in the NFN. We exemplarily defined a cost function for a CACC application. Finally, the application frames are sent at rate λ_{app} over the internal E/E network via the *E/E Architecture Interface* to the required safety application. The frames are sent periodically since especially safety applications depend on information updated regularly [22]. Note that the frame rate λ_{app}, at which the most relevant application data from a certain node are sent, is adjustable and could be greater or lesser than the received beacon rate λ_{beacon} of the corresponding sender node.

B. ECDSA Extensions

As depicted in Fig. 2 the green building blocks form the necessary signature processing extensions to the aforementioned message evaluation model. It outputs the most relevant application data (WSM payload) determined by the *Nearfield Evaluation Stage*. In contrast to the baseline design, the *Message Evaluation* module must also output the accepted signed WSM which belongs to the latest most relevant application data (WSM payload) in order to trigger the ECDSA signature verification. If and only if the accepted signed WSM is validated by the ECDSA core, the *WSM Valid* bit is set and the corresponding application data is forwarded to the *E/E Architecture Interface*. If the WSM Valid bit is not set the arriving application frames are discarded. This is handled by the *Gate Ctrl* module. Once a signed WSM is validated, the corresponding application frame can be forwarded even at a rate $\lambda_{app} >= \lambda_{beacon}$ as long as the contained data is not changed. The data could change for two reasons: 1) a new beacon arrives from the same sender node or 2) the most relevant application data now origins from another sender. Then, this new accepted and signed WSM is fed into the ECDSA core which resets the WSM Valid bit and verifies the signature. Finally, application data coming from other ECUs can be optionally signed by the ECDSA core and sent via the *WSM Scheduler*.

It is obvious that only messages accepted by the *Message Evaluation* need to be verified by the ECDSA core which significantly reduce the amount of messages that need to be validated. Hence, this directly leads to a reduction of the dynamic power consumption of the computation intensive ECDSA core. Furthermore the internal E/E network utilization is reduced beyond the pure message evaluation, since only accepted and validated messages are injected into the internal network via the *E/E Architecture Interface*.

Based on different parameter sets of the acceptance policies described in section IV-A and different traffic scenarios, an early power estimation of the ECDSA core can be explored. Note that in the given E/E architecture depicted in Fig. 1 also an overall relative reduction of the dynamic power consumption of the following V2X data dependent ECUs could be achieved. Because only accepted and validated messages are forwarded to that successive E/E subsystems. However, this is out of scope of this work. In order to enable early exploration and estimation of the dynamic power consumption of the ECDSA core, the co-simulation framework described in section III-A is extended by an ECDSA federate. The overall evaluation framework is detailed in the next section.

V. Evaluation Framework

In this section, the general approach used for power estimation is described at first. Afterward, the concrete evaluation framework for the V2X ECU depicted in Fig. 2 is detailed.

A. General Approach

Our approach relies on co-simulation based power estimation. Thereby, the approach is following two steps: 1) Co-simulation of the actual Design Under Test (DUT) with high-level and/or detailed and accurate domain-specific models as stimuli on application level. 2) The actual power estimation of the DUT is performed based on the monitored switching activity obtained from step 1). The switching activity is back-annotated onto the DUT according to the *Synopsys Power Compiler* flow [30].

1) Online HW/SW Power Co-Simulation: In this first step the developed framework described in section III-A is used to perform the power co-simulation.

Therewith, PtII serves as manager for the HLA data and control flow interactions between multiple federations each containing one federate (besides PtII itself). Hence, PtII serves as both stimuli generator and observer of incoming/outgoing HLA data. The federations shown on the left-hand side in Fig. 3 as well as PtII itself are used to generate stimuli data on application level, which are transferred by means of PtII and the HLA to the DUT. Note that the stimuli federations are not limited to those shown in Fig. 3, but arbitrary detailed

domain-specific models can be integrated by the simulation tool chain described in section III-A. External simulation models are used to provide detailed and realistic test data, e.g. Veins for accurate environmental V2X data or SystemC for detailed HW/SW subsystem simulation. PtII is used for abstract high-level application models, but also SystemC/TLM can be used for high-level stimulation. Since PtII manages all incoming and outgoing HLA communication, both PtII and high-level models as well as detailed domain-specific models can be mixed to provide a unified stimulus for the DUT. Furthermore, the DUT is not limited to only receive stimuli, but can even provide its own feedback data to the application/stimuli models. Hence, the application models can dynamically adjust its stimuli data based on the feedback data of the DUT. This enables even more realistic test data than unidirectional stimuli, which is important for design space exploration and power estimation on application level.

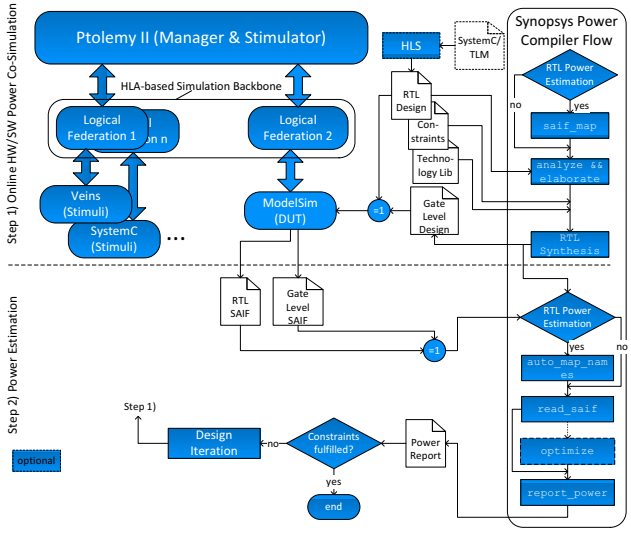

Fig. 3. Power Estimation Flow: 1) Online Co-Simulation of the DUT with mixed high-level (e.g. PtII, SystemC/TLM) and/or detailed domain-specific models (e.g. Veins) used as stimuli on application level. 2) Backward annotation of the monitored switching activity SAIF file into Synopsis Power Compiler for power estimation.

To evaluate the DUT we chose *ModelSim* [31] as common ASIC and FPGA simulator and debugger. The reasons are the support of the common hardware description languages VHDL and Verilog/SystemVerilog as well as SystemC as system description language (hardware/software). Furthermore it supports the generation of SystemC wrappers for existing VHDL/Verilog designs which enables IP core reuse and drastically eases the integration of these existing designs into the co-simulation evaluation framework by means of the C++ based *HLA Interface Wrapper* described in section III-A. Most important, it supports the online monitoring of switching activity of the DUT, which is the basis for the power estimation. The switching activity is stored in a standardized *Switching Activity Interchange Format* (SAIF) file used for the actual power estimation in Synopsys Power Compiler. Note, that any other digital hardware simulator could be used fulfilling the above features.

Finally, the DUT which is monitored by ModelSim have to be available either at RTL or at GL for the proper extraction of switching activity traces. Thereby the gate level netlist is typically generated by means of a synthesis tool, in our case Synopsys Design/Power Compiler. Nevertheless, high-level synthesis (HLS) tools, e.g. Xilinx Vivado high-level synthesis [32] can also be used to generate the initially necessary RTL design out of a (limited) high-level description (see Fig. 3). Basically, RTL simulation and monitoring of switching activity is fast but less accurate than at GL. In contrast, GL simulation and monitoring is more accurate by the extent of run-time (typically one or two orders of magnitude higher).

2) Power Estimation: In this step the SAIF file generated in step 1) is used as input for the power estimation conducted by Synopsys Power Compiler. Note that power estimation based on SAIF gives the average power of a circuit, since there is no information about the timing of the signals but the toggle rates and static probabilities. If power estimation is done on RTL, the command `saif_map` has to be invoked in the Power Compiler before elaboration and synthesis in order to properly map the RTL signal names to those at the post-synthesis gate level netlist. Independent of the RTL or GL simulation, the RTL design has to be synthesized with a given technology library and some constraints. The technology library contain all standard cell specific design rules and values like input and output capacitance or leakage currents in order to properly estimate switching power, cell internal power (equals to dynamic power) and static leakage power consumption. Constraints typically contain the timing and area constraints which should be met after synthesis. Also, *wire load models* are defined used to estimate the wire length and the effect of net interconnects on capacitance, resistance and area before the place and route design step.

After the RTL or full GL simulation in step 1) is done, the SAIF file is read into Power Compiler for back-annotation onto the synthesized design. The power is estimated based on the monitored switching activity and a power report is generated. Again, in case of RTL estimation, the command option `auto_map_names` has to be used with `read_saif` in order to properly match the RTL signal names to post-synthesis signals. Optionally, power constraints can be set after the estimation and an incremental synthesis optimizes the design meeting that constraints (represented with dashed lines in Fig. 3). After having the power reports, a design iteration can be done starting again with step 1) if the desired power constraints are not met. Finally, the Power Compiler design flow is automatized by Tcl scripts.

B. V2X-based ECDSA Power Estimation

In the following, the concrete evaluation framework for ECDSA power estimation based on the V2X Message Evaluation presented in section IV is outlined. The modeling and simulation framework depicted in Fig. 1 is extended by a further `HlaComposite` realizing the ECDSA DUT integration in ModelSim. This is shown in Fig. 4: The Vehicle Composite in PtII comprises the abstract E/E architecture of a selected vehicle communicating with surrounding vehicles via Veins.Inside the E/E architecture the V2X Message Evaluation design is modeled within the V2X ECU. The ECDSA block inside the V2X ECU now represents a SystemC refinement. The V2X Message Evaluation model together with the detailed

vehicular network traffic simulation of Veins realize a unified stimuli for the ECDSA DUT as described in section V-A. Every time a (signed) WSM from other vehicles is received by Veins and it is accepted by the Message Evaluation, the ECDSA core is triggered. The signature verification on RTL or GL is conducted while ModelSim captures switching activity. Vice versa, the ECDSA DUT provides the WSM Valid signal to the V2X ECU model which dynamically influences the subsequent PtII model, i.e. the forwarding of the accepted WSM into the internal E/E network to the appropriate V2X application.

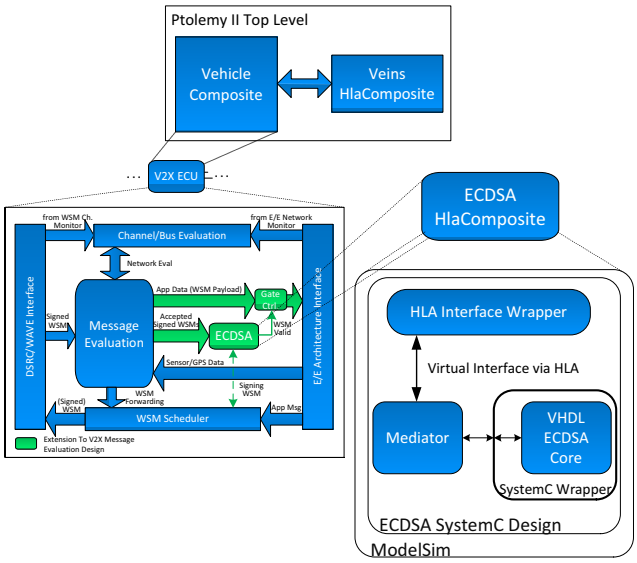

Fig. 4. Refined ECDSA Composite.

As ECDSA core we used a VHDL hardware implementation based on [28]. In order to be able to easily integrate it with the *HLA Interface Wrapper*, a SystemC wrapper for the top-level component of the VHDL design is generated by ModelSim (see section V-A). The *Mediator* shown in Fig. 4 thereby encapsulates and manages the high-level interactions and synchronization between the DUT and HLA through the *HLA Interface Wrapper*. The *Mediator* has an input buffer to store incoming WSMs via HLA, in case the ECDSA core has not yet finished the verification processing of the current WSM.

VI. Results

In this section the results of the power evaluation framework will be shown. At first, we conducted a standalone SystemC simulation of the ECDSA core for RTL and post-synthesis GL power estimation according to the flow described in section V-A. We then estimate the average power and energy consumption of signature verifications by means of the V2X message evaluation co-simulation scenario. Also the relative reduction of energy consumption of signature verification processing with and without V2X Message Evaluation is shown.

For the power estimation of the ECDSA core we used the $45nm$ standard cell technology library *tcbn45gsbwpwc* from TSMC [33]. Worst case operating conditions of the library were applied. Additionally we chose a high wire load model *TSMC512K_Lowk_Conservative* for increasing the impact of cells with high fan-out nets in order to increase estimation accuracy. The ECDSA design was synthesized and simulated with a clock frequency of $50MHz$. This setup applies for both the standalone and the co-simulation based power estimation.

A. Standalone ECDSA Power Estimation

Tab. I shows the estimated average power values of a single signature verification and generation process on RTL and post-synthesis GL with an input message of four bytes.

TABLE I. Standalone ECDSA Power Estimation on RTL and GL @ $50MHz$.

	Sig. Verification	Sig. Generation
GL	$4,064mW$	$4,507mW$
RTL	$3,364mW$ (-17,2%)	$3,332mW$ (-26,1%)
Simulated Time	$6,96ms$	$5,48ms$

The values in brackets give the relative accuracy deviation compared to GL estimation. This is because of the missing annotations of the gate level cells in case of RTL simulation.

B. Co-Simulation based ECDSA Power and Energy Estimation

As a case study for the ECDSA power estimation evaluation framework (see section V-B), the CACC application mentioned in section IV-A is used. It is modeled within the *Central ECU* of the abstract E/E architecture model shown in Fig. 1. As well, the V2X Message Evaluation and the refined SystemC ECDSA core are integrated but only signature verifications upon WSM reception are analyzed. Signature generation for internal application data to be sent via WSM is not regarded. In addition, to verify the reduction of energy consumption, only RTL co-simulation for power estimation is presented, because of extremely long run-times at GL. However, the relative reduction of energy consumption stays the same, since the same scenarios and thus the same input values for the ECDSA as on RTL are used. Only the absolute average power and energy estimation values differ compared to RTL.

In the following we show that a significant number of signature verifications can be saved at run-time by the V2X Message Evaluation compared to the case, that every incoming WSM is accepted. This drastically reduces the energy consumption of the ECDSA core. This is detailed in the following.

1) Co-Simulation Scenario Specification: We first constructed an urban traffic scenario in SUMO consisting of two adjacent lanes, where vehicles send beacons at $10Hz$. The traffic simulation is executed with a fore-run of $10s$ in order to have an initial vehicle network for evaluation. The total simulation time is $30s$. Follower vehicles are inserted in simulation every $2s$. After $30s$ of simulation time 14 vehicles actively participated in simulation, i.e. they have sent a WSM. The leading vehicle is periodically approaching intersections with a distance of $200m$ to each other. A second follower vehicle is controlled remotely by PtII, whose speed is set with the value calculated by the CACC application in the Central ECU of the abstract E/E architecture. The rest of the vehicles are following the second one. The beacons of all vehicles, each containing the WSM header data and the current vehicle's status (payload), are transferred to PtII via the wireless channel model of Veins. Therewith, the received WSMs have a total

size (header + payload) of 200 Bytes. This reflects the message size the ECDSA has to verify per WSM. The input buffer size of the *Mediator* shown in Fig. 4 is set to 10 WSMs. The mobility, network and ECDSA simulation parameters are summarized in Tab. II.

TABLE II. CO-SIMULATION PARAMETERS.

Simulation Time	$30s$
Traffic Fore-Run	$10s$
# Vehicles	14
Beacon Rate	$10Hz$
Beacon (WSM) Size	$200Bytes$
ECDSA Mediator Input Buffer Size	$10WSMs$
ECDSA Abstraction Level	RTL
ECDSA Total Message Size	$200Bytes$
ECDSA Clock Frequency	$50MHz$

2) Estimating Average Energy Consumption: In the next step we use this scenario for estimating the energy consumption of the ECDSA verifications without and with V2X message evaluation. We refer these co-simulation scenarios to S_I and S_{II} respectively. Therewith, the following steps are performed: 1) Determine average power $\overline{P}_{verif}$ of the signature verifications based on the SAIF file produced by each co-simulation scenario (according to Sec. V); 2) calculation of the average energy consumption $\overline{E}_{verif}$ of the signature verifications by means of the estimated average power, the number of conducted verifications n_{verif} during co-simulation and the duration of each verification $t_{verif,i}$.

The average energy for a single verification is calculated by

$$\overline{E}^{single}_{verif} = \overline{P}_{verif} * t_{verif} \tag{1}$$

The total average energy for all signature verifications in simulated time is then given by

$$\overline{E}^{total}_{verif} = \sum_{i=1}^{n_{verif}} \overline{P}_{verif} * t_{verif,i} \tag{2}$$

The accumulation of the energy is necessary, since the duration of a single verification is not constant. The number of cycles for the operations Point Add and Point Doubling inside the ECDSA ALU depend on the input operands, but is limited to max. $2 * |p|$ cycles [28].

3) Estimated Average Energy Consumption Without V2X Message Evaluation: In Fig. 5 the number of overall received WSMs (dark blue bars) in the V2X ECU without message evaluation is shown. The received WSMs are accumulated within a time slot of $2s$. After $30s$ of simulation time 1753 WSMs were received. Because of the missing message evaluation and filtering, this corresponds to the number of triggered signature verifications $n_{verif,S_I} = 1753$. Therewith, no buffer overflow occurred in the *Mediator* input buffer. The estimated average power based on the SAIF annotations of the whole co-simulation results in $\overline{P}_{verif,S_I} = 3,3927mW$. Applying Eq. 2 the total average energy consumption for all signature verifications results in $\overline{E}^{total}_{verif,S_I} = 42,605J$.

4) Estimated Average Energy Consumption With V2X Message Evaluation: In case of V2X message evaluation the remaining number of messages after each evaluation stage inside the V2X Message Evaluation module is illustrated in Fig. 5 (blue and light blue bars). We assume that every

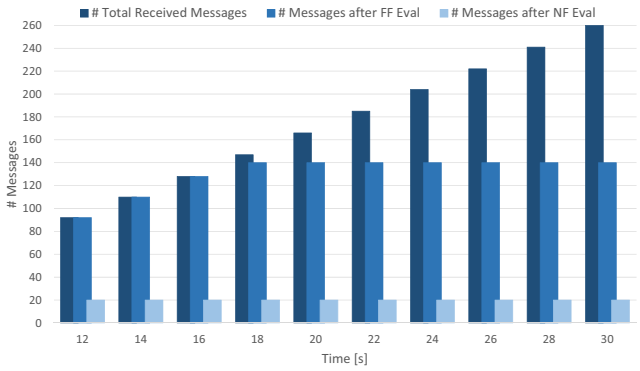

Fig. 5. WSM Histogram of the V2X Traffic Scenario.

WSM which belongs to the corresponding CACC frame (WSM payload) has to be verified by the ECDSA core. This holds because the CACC frame rate $\lambda_{app} = 10Hz$ is equal to the beacon rate λ_{beacon} of the leading vehicle and the CACC frame always contains the status data of the leading vehicle. The constant number of 20 messages in Fig. 5 (light blue bars) show the periodic CACC frames generated in the nearfield evaluation stage and correspond to the number of messages injected into the ECDSA core. Analogues to Sec. VI-B3, after $30s$ of simulation time 200 WSMs were received. This results in $n_{verif,S_{II}} = 200$. The estimated average power results in $\overline{P}_{verif,S_{II}} = 3,6244mW$. Analogues, applying Eq. 2 the total average energy results in $\overline{E}^{total}_{verif,S_{II}} = 5,195J$.

TABLE III. SUMMARY OF CO-SIMULATION BASED ECDSA POWER AND ENERGY ESTIMATION.

	Scenario	
	S_I	S_{II}
n_{verif}	1753	200 $(-88,59\%)$
$\overline{P}_{verif}$	$3,3927mW$	$3,6244mW$ $(+6,83\%)$
$\overline{E}^{total}_{verif}$	$42,605J$	$5,195J$ $(-87,81\%)$

In Tab. III the results for both scenarios are summarized. The values in brackets give the relative savings of signature verifications and energy consumption as well as the relative increase of the average power estimation of scenario S_{II} compared to S_I.

C. Short Discussion

Tab. III clearly shows that a significant amount of $88,59\%$ of all WSMs can be discarded. Furthermore, a relative reduction of $87,81\%$ of the energy consumption of all signature verifications is achieved. This proves that the energy consumption of the ECDSA core is drastically reduced by the performed V2X message evaluation and filtering. The minimal deviation of $0,78\%$ between the relative verification and energy savings arises due to the higher estimated average power of $+6,83\%$ in S_{II}. The reason for this deviation is that WSMs with different payload appear in S_{II}. This results in different toggle rates in the SAIF file which in turn delivers different input values to the power estimation. However, regarding this increase as a worst case value for the average power, the difference of $0,78\%$ between the relative reduction of the verifications and total energy consumption is negligible. Using $\overline{P}_{verif,S_I} = 3,3927mW$ as a best case value for calculating $\overline{E}^{total}_{verif,S_{II}}$ the relative reduction of verifications and the according energy

consumption even correlates to $88,59\%$.

VII. Conclusion and Outlook

In this work we presented a power estimation framework using HLA-based heterogeneous distributed simulation of domain-specific simulators and commercial off-the-shelf synthesis and HDL simulation tools. It allows on-the-fly cross-domain interaction and generation of stimuli data on different levels. A standalone power estimation of an ECDSA core is evaluated on both RTL and GL. RTL co-simulation of the ECDSA core embedded in a V2X scenario shows how mixed detailed and high-level application specific stimuli can improve power and energy estimation. Additionally, a V2X message evaluation scenario showed, that the energy consumption of the ECDSA signature verifications can be reduced by $87,81\%$. Future work comprises further evaluation in more complex scenarios as well as the integration of ESL models and performance considerations.

References

[1] A. Jantsch and I. Sander, "Models of computation and languages for embedded system design," *IEE Proceedings on Computers and Digital Techniques*, vol. 152, no. 2, pp. 114–129, March 2005, special issue on Embedded Microelectronic Systems; Invited paper. [Online]. Available: http://www.imit.kth.se/~axel/papers/2005/IEE-Proceedings.pdf

[2] E. Lee and A. Sangiovanni-Vincentelli, "Component-based design for the future," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, March 2011, pp. 1–5.

[3] M. Lajolo, A. Raghunathan, S. Dey, and L. Lavagno, "Cosimulation-based power estimation for system-on-chip design," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 10, no. 3, pp. 253–266, June 2002.

[4] C. Roth, H. Bucher, A. Brito, O. Sander, and J. Becker, "A Simulation Tool Chain for Investigating Future V2X-based Automotive E/E Architectures," in *Embedded Real Time Software and Systems (ERTS^2), 7th European Congress on*, Feb. 2014, pp. 1739–1748.

[5] "IEEE Standard for Standard SystemC Language Reference Manual," *IEEE Std 1666-2011 (Revision of IEEE Std 1666-2005)*, pp. 1–638, Jan. 2012.

[6] V. B. Michael Wetter, "A modular building controls virtual test bed for the integrations of heterogeneous systems," in *SimBuild 2008*, Berkeley, CA, USA, Aug. 2008.

[7] Y. Zhao, J. Liu, and E. A. Lee, "A programming model for time-synchronized distributed real-time systems," in *Proceedings of the 13th IEEE Real Time and Embedded Technology and Applications Symposium*, ser. RTAS '07. IEEE Computer Society, 2007, pp. 259–268.

[8] J. Eidson, E. A. Lee, S. Matic, S. A. Seshia, and J. Zou, "Distributed real-time software for cyber-physical systems," *Proceedings of the IEEE (special issue on CPS)*, vol. 100, no. 1, pp. 45 – 59, January 2012. [Online]. Available: http://chess.eecs.berkeley.edu/pubs/850.html

[9] J. H. H. Chang Ho Sung and T. G. Kim, "Interoperation of devs models and differential equation models using hla/rti: Hybrid simulation of engineering and engagement level models," in *2009 Spring Simulation MultiConf*, Mar. 2009.

[10] C. Sung and T. G. Kim, "Framework for simulation of hybrid systems: Interoperation of discrete event and continuous simulators using hla/rti," in *Principles of Advanced and Distributed Simulation (PADS), 2011 IEEE Workshop on*, June 2011, pp. 1–8.

[11] G. Lasnier, J. Cardoso, P. Siron, C. Pagetti, and P. Derler, "Distributed simulation of heterogeneous and real-time systems," in *Distributed Simulation and Real Time Applications (DS-RT), 2013 IEEE/ACM 17th International Symposium on*, Oct 2013, pp. 55–62.

[12] S. Schurmans *et al.*, "Creation of esl power models for communication architectures using automatic calibration," in *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, May 2013, pp. 1–6.

[13] S. Schürmans, D. Zhang, R. Leupers, G. Ascheid, and X. Chen, "Improving esl power models using switching activity information from timed functional models," in *Proceedings of the 17th International Workshop on Software and Compilers for Embedded Systems*, ser. SCOPES '14. New York, NY, USA: ACM, 2014, pp. 89–97. [Online]. Available: http://doi.acm.org/10.1145/2609248.2609250

[14] C. Ptolemaeus, Ed., *System Design, Modeling, and Simulation using Ptolemy II*. Ptolemy.org, 2014. [Online]. Available: http://ptolemy.org/books/Systems

[15] "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)," *IEEE Std 1516.x-2010*, Aug. 2010.

[16] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[17] A. Varga and R. Hornig, "An overview of the omnet++ simulation environment," ser. Simutools '08. ICST, Brussels, Belgium, Belgium: ICST, 2008, pp. 60:1–60:10.

[18] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo - simulation of urban mobility: An overview," in *SIMUL 2011, The Third International Conference on Advances in System Simulation*, Barcelona, Spain, 2011.

[19] "IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments," *IEEE Std 802.11p*, Jun. 2010.

[20] "IEEE Standard for Wireless Access In Vehicular Environments (WAVE) - Architecture," *IEEE Std 1609.0*, Dec. 2013.

[21] L. Le, A. Festag, R. Baldessari, and W. Zhang, "Vehicular wireless short-range communication for improving intersection safety," *Communications Magazine, IEEE*, vol. 47, no. 11, pp. 104–110, November 2009.

[22] J. Ploeg, B. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, Oct 2011, pp. 260–265.

[23] T. Tielert, D. Rieger, H. Hartenstein, R. Luz, and S. Hausberger, "Can v2x communication help electric vehicles save energy?" in *ITS Telecommunications (ITST), 2012 12th International Conference on*, Nov 2012, pp. 232–237.

[24] BMW. (2013) BMW ConnectedDrive. [Online]. Available: http://www.bmw.com/com/en/insights/technology/connecteddrive/2013

[25] "IEEE Standard for Wireless Access in Vehicular Environments: Security Services for Applications and Management Messages," *IEEE Std 1609.2-2013 (Revision of IEEE Std 1609.2-2006)*, Feb. 2013.

[26] C. F. Kerry, A. Secretary, and C. R. Director, "FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard (DSS)," 2013. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.362.5590

[27] B. Glas, O. Sander, V. Stuckert, K. D. Müller-Glaser, and J. Becker, "Car-to-car communication security on reconfigurable hardware," in *VTC Spring'09*, 2009.

[28] B. Glas, O. Sander, V. Stuckert, K. D. Müller-Glaser, and J. Becker, "Prime field ecdsa signature processing for reconfigurable embedded systems," *Int. J. Reconfig. Comput.*, vol. 2011, pp. 5:1–5:12, Jan. 2011. [Online]. Available: http://dx.doi.org/10.1155/2011/836460

[29] H. Bucher, F. Buciuman, A. Klimm, O. Sander, and J. Becker, "A V2X Message Evaluation Methodology and Cross-Domain Modelling of Safety Applications in V2X-enabled E/E-Architectures," in *Proceedings of the 8th EAI International Conference on Simulation Tools and Techniques*, 2015.

[30] Synopsys. (2015) Power Compiler. [Online]. Available: http://www.synopsys.com/Tools/Implementation/RTLSynthesis/Pages/PowerCompiler.aspx

[31] M. Graphic. (2015) ModelSim. [Online]. Available: http://www.mentor.com/products/fv/modelsim/

[32] Xilinx. (2015) Vivado High-Level Synthesis. [Online]. Available: http://www.xilinx.com/products/design-tools/vivado/integration/esl-design.html

[33] Taiwan Semiconductor Manufacturing Company Limited (TSMC). (2015). [Online]. Available: http://www.tsmc.com