Abstract—Each new embedded system tends to integrate more sensors with tight software-driven control, digitally assisted analog circuits, and heterogeneous structure. A more responsive simulation environment is needed to support the co-design and verification of such complex architectures including all its digital hardware/software and analog/multi-physical aspects using Multi-Disciplinary Virtual Prototyping (MDVP). Taking a Micro-Electro-Mechanical System (MEMS) vibration sensor as an example, we introduce a reusable framework based on the state-of-the-art technologies SystemC AMS, Finite Elements/Reduced-Order modeling, and UVM to design, simulate, and verify such systems in their real application context.

Index Terms—Universal Verification Methodology (UVM), SystemC AMS, Micro-Electro-Mechanical System (MEMS), Multi-Disciplinary Virtual Prototyping (MDVP), design and verification methodology, dimensional analysis, bond graph formalism, reduced-order modeling.

I. INTRODUCTION

Virtual prototyping is one of the key enabling technologies of the past ten years. Semiconductor companies have successfully worked on effective solutions to model their products at the architectural level of abstraction, enabling early embedded software development. At the same time, virtual platforms have evolved according to the “More Than Moore” laws. Today, manufactured products are heterogeneous in nature, as they integrate on the same die digital, analog, Radio Frequency (RF), and MEMS such as gyroscopes, accelerometers or complete Inertial Measurement Units (IMUs). The verification and validation of such an integrated heterogeneous system, especially when it integrates MEMS and its control electronics, have become a real challenge and its intrinsic complexity keeps on growing.

In practice, the main issue is to dramatically improve the design efficiency and quality of such complex systems and to develop methods and tools that can tackle the associated verification challenges. Designers are longing for an holistic system verification approach that covers simulation-based system analysis and validation based on established standards such as SystemC [1] and the Universal Verification Methodology (UVM) [2] enabling reuse of design and verification Intellectual Property (IP). By combining these two standards developed by different communities with Analog/Mixed-Signal (AMS) extensions [3] to a UVM-SystemC-AMS framework, the design and verification process can be unified enabling the Multi-Disciplinary Virtual Prototyping (MDVP) of heterogeneous systems containing MEMSs. To illustrate the advantages of such a framework, the paper applies its principles to a system containing a MEMS device, a vibration sensor, and considers it as the Device Under Test (DUT). The next section details the progressively refined modeling of the MEMS device using two different Models of Computation (MoCs) of SystemC AMS, namely Timed Data Flow (TDF) and Bond Graph (BG), and reduced-order modeling techniques. Section III shows how these SystemC AMS models can be verified by means of a UVM-SystemC-AMS-based test environment providing all the necessary vector generation, driving, monitoring, scoreboarding, and coverage estimation resources. Section IV concludes the paper.

II. MEMS CO-DESIGN USING VIRTUAL PROTOTYPING

To demonstrate the advantages for the virtual prototyping of heterogeneous systems, we have chosen the co-design of a vibration sensor and its frontend circuit, the concept of which is shown in Figure 1a. The sensing element is a MEMS resonator, which is exposed to time-varying accelerations. The movement of the inertial mass with respect to its reference frame is detected by a comb capacitor, which is biased to a certain voltage. The relative movement of the comb with respect to each other causes a displacement current, which is proportional to the speed and measured through a shunt resistor. This analog signal is processed by
a frontend circuit, which pre-amplifies it before converting it to its digital representation. To fully use the dynamic range of the Analog-to-Digital Converter (ADC), the gain of the pre-amplifier is programmable (PGA) via a digital controller, which receives an estimation of the amplitude as an input. To adapt the sensor and its frontend to the characteristics of the expected vibration signals, key parameters such as the supply voltages for the sensor, amplifier, and ADC, the sampling frequency of the ADC as well as the number of samples to consider for the amplitude estimation can be configured via a sensor control register bank.

A. Ideal sensor specification using Timed Data Flow

In a first step, the sensor and frontend circuit concept can be evaluated on the system level using a virtual prototype realized with SystemC and its AMS extensions. The signal processing aspects consisting of the vibration source, sensor element, PGA, ADC, and amplitude estimator can be conveniently modeled using the TDF MoC of SystemC AMS. At this stage, the overall dynamic behavior of the vibration sensor itself is only represented using a Laplace transfer function to convert the input acceleration to a speed-proportional voltage signal at its output. The behavior of the PGA, ADC, and amplitude estimator are described in an algorithmic way in the processing member functions of the respective TDF modules. The gain controller is implemented as a finite state machine using the Discrete Event (DE) MoC. The reconfiguration of the sensor front end via the sensor control register file to different sampling period ($T_s$), oversampling rate ($n_{os}$) and number of samples for the amplitude estimation ($n_t$) is implemented using the new Dynamic TDF features of SystemC AMS 2.0.

Already in this early stage of the system design, it is important to clearly specify the interfaces of each functional block and what information is communicated. This is especially important as different engineering/physical domains are involved. Therefore, naming conventions may not be sufficient to avoid misinterpretation of values (e.g., speed and voltage are commonly referenced with the symbol $v$). Therefore, it is beneficial to annotate systematically the measurement units to all numerical values and to specify which physical quantities are meant to be represented by signals and variables in the system model. To this end, we have implemented an integration of the Boost.Units library for compile-time dimensional analysis with SystemC AMS [4], which we systematically use in our models. This ensures the correct composition of the system model as well as coherency of the calculations done inside each block.

B. Energy-conservative refinement using Bond Graphs

To design the MEMS resonator according to the specifications of the entire system (sensitivity, bandwidth, etc.), it is beneficial in a first refinement step to study the interaction between the mechanical and electrical domains using a lumped energy-conserving model. In our case, the impact of the sensor signal readout circuit through the electrostatic force on the mechanical resonator is of particular interest. In contrast to modeling languages such as Verilog-AMS,
Verification Methodology (UVM) as part of an industry which integrates into the Fraunhofer SystemC-AMS library. VHDL-AMS, and Modelica, SystemC AMS 2.0 does not yet offer support for conservative modeling of nonlinear systems. However, it surpasses them when it comes to model complex hardware/software systems. Therefore, we propose to use the bond graph formalism for convenient multi-physical modeling. It is based on generic primitives representing sources, junction, resistive, capacitive, and inertia effects in a unified way for all energy domains [5]. We have developed a BG MoC prototype as part of our SCAX library [6], which integrates into the Fraunhofer SystemC-AMS library. Using the block diagram and bond graph primitives from the SCAX library, the complete vibration sensor can be modeled as shown in Figure 1b.

C. Geometric design with FEM and reduced-order modeling

Through the study of the interaction of the mechanical and electrical domains, suitable ranges for the key mechanical parameters of the MEMS resonator such as mass, stiffness, and damping factor as well as for the electrostatic transducer (capacitance over displacement) can be determined to satisfy the overall system specification. In a next refinement step, these have to be realized by doing the geometric design of the MEMS resonator respecting the constraints of the manufacturing technology (materials, layer thicknesses, geometric design rules). This is facilitated by using dedicated CAD tools and Finite Element Method (FEM) tools. The challenge is their integration with the system-level design tools. In our case, we use Coventor MEMS+, which enables the MEMS design at system level based on a library of 3-D-parametric components such as plates, cantilevers, springs, comb capacitors, etc. Their assembly leads to higher-order finite element models thus reducing the complexity of the overall system while preserving good accuracy [7]. Still, these models remain too complex for integration into the system-level virtual prototype. Therefore, a dedicated model export feature is under development, which further reduces the complexity of the model by applying reduced-order modeling techniques. A prototype has been realized that is capable of exporting the dynamic behavior of the MEMS+ 3-D model of the resonator (Figure 1c) to a SystemC AMS TDF module [8].

III. MEMS VERIFICATION IN THE APPLICATION CONTEXT

A. Introduction to UVM-SystemC-AMS

Several verification methodologies for digital systems have been developed and promoted by EDA vendors over the last decade, which have evolved towards the Universal Verification Methodology (UVM) as part of an industry standardization effort in the Accellera Systems Initiative [2]. UVM consolidates verification best practices by introducing a unified approach for test and sequence creation, building verification components, test bench configuration, and execution by means of simulation [9]. This UVM supports design and verification engineers with an open source class library that facilitates the creation of verification components and models for digital test benches.

The UVM principle is to build a test bench (called UVM environment) using reusable verification components (called UVC) in a structured way to ease reuse between different tests or even across different projects. In this context, the goal is to enable the description of abstract test scenarios in terms of transactions. These transactions are then refined by a driver into signals that are sent to the Device Under Test (DUT). A monitor collects the output signals from the DUT and translates them through abstraction to transactions. The subscribers are supplied with these output signal transactions as well as the collected input stimuli transactions. Figures of merit are computed by analysis functions from this collected information. The scoreboard compares the results for the DUT with the reference ones.

The cornerstone of the UVCs is the agent, which instantiates the sequencer, the driver and the monitor. The sequencer handles the transactions and sends them to the driver. The transactions are encapsulated in sequences, which form the virtual sequence. Using the UVM configuration mechanism, the UVC can be configured to match the scenario of interest.

To take advantage of the system modeling approach based on virtual prototyping of the hardware architecture as well as the UVM approach for verification, UVM-SystemC-AMS defines all the features to create a standard verification environment for system-level models. This methodology and library implementation has been developed within the EU FP7 VERDI project [10] and is in the process of being standardized by Accellera.

B. Application of UVM-SystemC-AMS to MEMS

Using UVM-SystemC-AMS, a test environment for the virtual prototype of our vibration sensor system (using one of the three refinements of the vibration sensor model described in Section II) as DUT has been built as depicted in Figure 2. Its relative complexity with respect to traditional test benches is the prize to pay for the increased reusability and configurability of its individual components. The DUT is connected to the test environment by means of three interfaces and related connections: the Input wave interface, the digital Sensor control interface, and the digital Output interface. These interfaces are used to instantiate the related connections between the DUT and the test environment. The test environment consists of three UVCs: The Stimuli wave UVC is responsible for the generation of the vibration stimuli to be sensed. To this end, it extracts from the high level sequence item (frequency, harmonics, amplitude) the necessary parameters and generates the corresponding low level physical stimulation signal on the Wave interface. The Sensor control UVC is used to control the sensor signal acquisition over time specifying different sampling periods, oversampling rates (n_{os}), number of samples for the amplitude estimation, and supply voltages. According to UVM terminology, both of these UVCs are active in the sense that they propagate stimuli from high-level sequences to the DUT. Finally, the Output UVC is responsible for reconstructing the high-level information (e.g., estimated amplitude and frequency of the ADC output signal) from the raw physical values produced by the DUT. The scoreboard subscribes to the monitor output of all three UVCs to.
aggregate the simulation results and compare them with its reference model (ideal DE-TDF model of the system).

Table I resumes the recorded scoreboard when the sensor system was exposed to a vibration signal sequence of three levels of abstractions of the sensor model even though relevant signals can be recorded for visual inspection (Fig.

In this paper, we presented a multi-disciplinary virtual prototyping, verification, and validation framework for MEMS using SystemC AMS, UVM, and Coventor MEMS+. To this end, SystemC AMS has been enhanced to model multi-physical systems using bond graphs and to ensure the consistency of the model via compile-time dimensional analysis. MEMS+ has been enhanced with a SystemC AMS model export feature to integrate reduced-order models of the MEMS into the system-level application model. Finally, UVM-SystemC-AMS allows to realize a unified, structured, configurable, and modular test environment for MEMS design. We applied our framework to a vibration sensor system and successfully achieved its verification.

**Table I.** Scoreboard of the UVM environment.

<table>
<thead>
<tr>
<th>$a_{\text{vib}}$/m s$^{-2}$</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>10</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_{\text{vib}}$/Hz</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>4001</td>
<td>8000</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>3999</td>
<td>7997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_{\text{TDF}}$/Hz</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>4001</td>
<td>8000</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>3999</td>
<td>7997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$f_{\text{ROM}}$/Hz</td>
<td>500</td>
<td>1000</td>
<td>1999</td>
<td>4001</td>
<td>8000</td>
<td>500</td>
<td>1000</td>
<td>2000</td>
<td>3999</td>
<td>7997</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{\text{TDF}}$/1</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{\text{BG}}$/1</td>
<td>15</td>
<td>13</td>
<td>13</td>
<td>12</td>
<td>10</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$k_{\text{ROM}}$/1</td>
<td>15</td>
<td>14</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>13</td>
<td>12</td>
<td>11</td>
<td>10</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_{\text{amp}}$/PEB/1</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_{\text{amp}}$/BG/1</td>
<td>8</td>
<td>3</td>
<td>7</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$a_{\text{amp}}$/ROM/1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 3.** Comparison of simulation results of the vibration sensor system using the reference (REF), bond graph (BG), and reduced-order (ROM) abstractions of the sensor.

**REFERENCES**


