WHITE PAPER

Virtual Prototyping for Electric Vehicles: From System to Software

Author

Kevin Brand
Senior R&D Manager,
Synopsys

Introduction

Road vehicles have long been considered one of the most challenging types of development projects. The tight interaction between electrical and mechanical components and the inherent safety requirements of high-speed/high-power operation are just the baseline issues. Layered on top of that are the many challenges of the physical environment: wide temperature and humidity range, noise, vibration and aging of components. The rapid rise of electric vehicles (EVs) has removed some issues, such as reliance on a highly flammable fuel, but replaced them with hazardous high voltages and currents from large battery packs, with the risk of thermal run-away and fire. It has also added new challenges. The electronic content of EVs can be considerably higher than those with internal combustion engines. The interaction among all components of the vehicle includes feedback loops unique to the EV configuration.

It is critical that these challenges be addressed during the design and verification process. Bench testing, also known as mule vehicles, is a labor intensive and expensive process, since injecting real-world failures can damage or destroy physical prototypes, and it's a slow process to consider the many 100s if not 1000s of potential fault variants that may require consideration. Validation using actual vehicles is clearly very risky since any errors detected while driving could result in injury or death. Furthermore, any electronic bugs detected after silicon fabrication or software rollout could result in project delays and massive cost over-runs to "turn" the chips. Finally, controller software may not operate correctly with respect to component tolerances, resulting in later realized subsystem failures and warranty callbacks.

This white paper presents an attractive alternative available today: virtual prototypes for EVs that can verify complete electro-mechanical-hydraulic-thermal systems, including their embedded software, purely in simulation. The simulation model of the EV is the executable specification recognized as the virtual prototype. In addition, simulation permits painless investigation of all possible variants of subsystem topologies. Implementation problems regarding component variances and other issues can be discovered and investigated early, which allows improvements before the real hardware prototype becomes available. This reduces development time and helps ensure that the first hardware prototype will be more robust. In addition, physical prototype testing time can be reduced since many of the verification tests and soft/hard fault analyses can be exercised through the simulation model of the EV. The result is a faster, more efficient and less expensive process for the design and verification of electric vehicles.

The Rise of EVs

The history of EVs is quite interesting, since many early automotive designs relied on electric power rather than combustion. As engine technology matured and gasoline became widely available, the longer range of combustion-based vehicles became their main selling point. EVs became the largely the domain of hobbyists and enthusiasts, with a minimal commercial market. All this has changed in recent years. Rising fuel costs, improved battery technology and environmental sensitivity have all helped to drive more interest in EVs. While their share of the overall market is small today, projections show that by 2040 the world will be producing 60 million EVS annually.

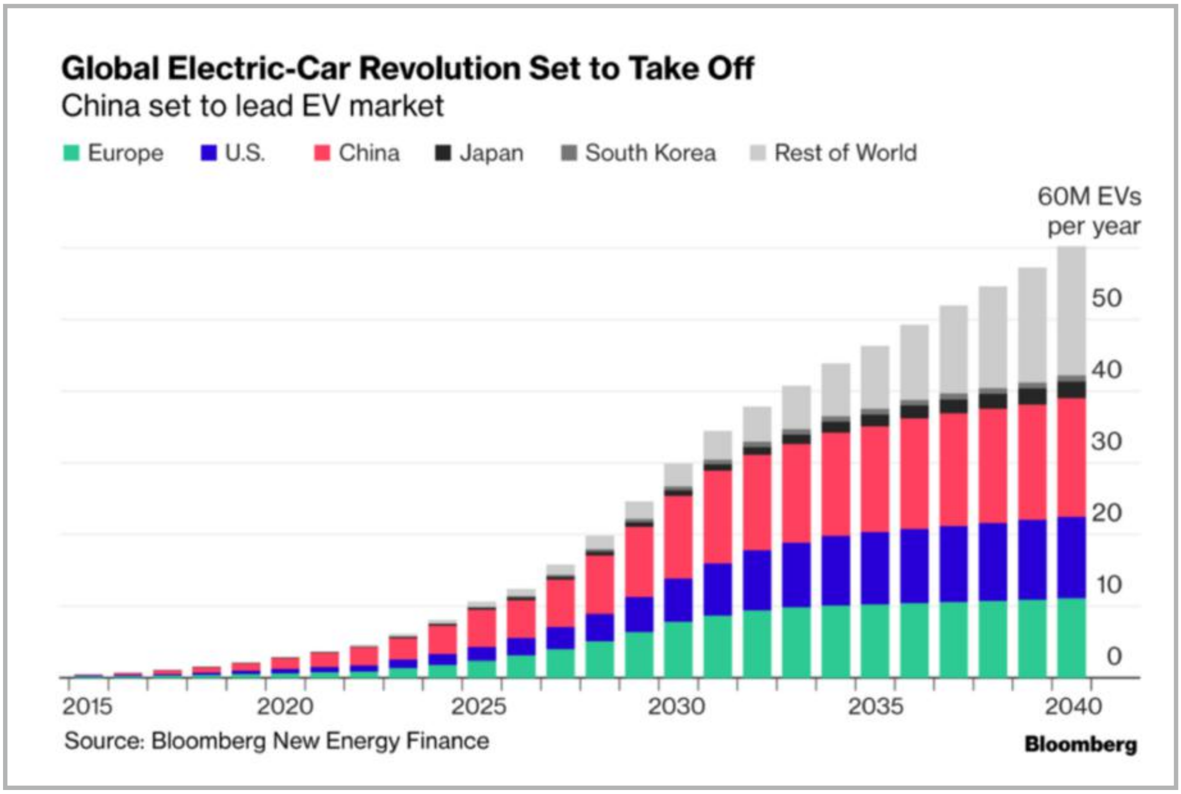

Figure 1: Current and projected EV growth

There are now numerous competitors in the EV market, including many of the traditional automotive manufacturers. Succeeding against the competition is largely about making the proper tradeoffs in the architecture and design of the vehicle. Perhaps the most obvious parameter for vehicles, especially automobiles, is performance. Acceleration, often measured as time from 0 to 60 MPH, and top speed are the two primary metrics for performance. EVs were once considered sluggish; some of today's top models meet or exceed the metrics for combustion-driven cars. However, the average driver spends little time in maximum acceleration or driving at anywhere close to top speed. Beyond the "bragging rights" of performance, other factors are more likely to drive purchase decisions.

Efficiency is clearly one of these factors. Peter Rawlinson, Chief Executive Officer and Chief Technology Officer at EV manufacturer Lucid, defines efficiency as the distance a vehicle can travel at a given speed using a given number of kilowatt-hours. The more efficient, the farther the vehicle can travel for a given battery. Efficiency and battery capacity combine to determine the driving range between charges, a major factor for drivers considering moving from internal combustion engines to full EV (no hybrid or gasoline assist technology). "Range anxiety" is a common industry term for concern that an EV will drain its battery before reaching a charging station.

As with any product, cost is also a consideration. There are always some for whom money is no object, and they will pay a large premium for maximum performance or efficiency. The rest of the buyers must weigh these metrics against cost, both for the initial purchase and for eventual battery replacement. An EV developer must make the proper tradeoffs across performance, efficiency/range and cost in order to be successful against competitors in the target market. These tradeoffs must be evaluated and iterated many times during the development process. Much like design verification, bench testing or driving of actual vehicles is too late for such efforts. Of course, a huge amount of the cost is in the development process itself, so reducing development and testing time leads to better cost margins on the final product. Software-based virtual prototypes provide the solution here as well.

EV Development Challenges

There are many aspects that must be considered in the development of EVs. As above, product differentiation involves maximizing range, performance and efficiency while keeping product costs as low as possible. Development costs and non-recurring engineering (NRE) expenses should also be minimized through an efficient design and verification process. Addressing system-level challenges, including dealing with highly connected subsystems and sufficient cyber-security to prevent unauthorized access, is paramount for a secure and safe product. Integrating many modules and subsystems is a complex task, exacerbated by both hardware and software components from a diverse supply chain. Many of these suppliers may be new to the automotive world, so they must be qualified at the same level as traditional vendors. One development challenge specific to EVs is battery management, including minimization of charge time and proper handling of thermal and safety considerations.

Development of the electrical hardware within an EV has its own challenges. The process starts with a concept design that enables the team to ramp up on emerging technologies and explore the design space to investigate alternatives and make the tradeoffs discussed earlier. Selecting the actual electrical components for the design involves additional levels of performance versus cost analysis and tradeoffs. Prior to bench testing, as much verification and analysis as possible must be performed using physical and electrical component models. Hardware in the loop (HIL) bench testing must be minimized due to the cost of replicating bench setups with expensive instrumentation and the limited availability this implies.

Modern vehicles, especially EVs, contain much more than hardware. Each electronic control unit (ECU) is an embedded system controlling one or more of the electrical systems or subsystems. The software content of embedded systems presents its own set of challenges. System-level verification and calibration are big efforts, especially for functions involving multiple ECUs or multiple cores in a single ECU. Software stacks support many complex communication protocols, including CAN, LIN, Ethernet, SPI, FlexRay, I2C and PCI Express (PCIe). In addition, the ISO 26262 standard mandates safety mechanisms and verification that the system will respond properly in the present of faults. Testing hardware by inserting faults is impractical; virtual prototypes enable safety verification using software models.

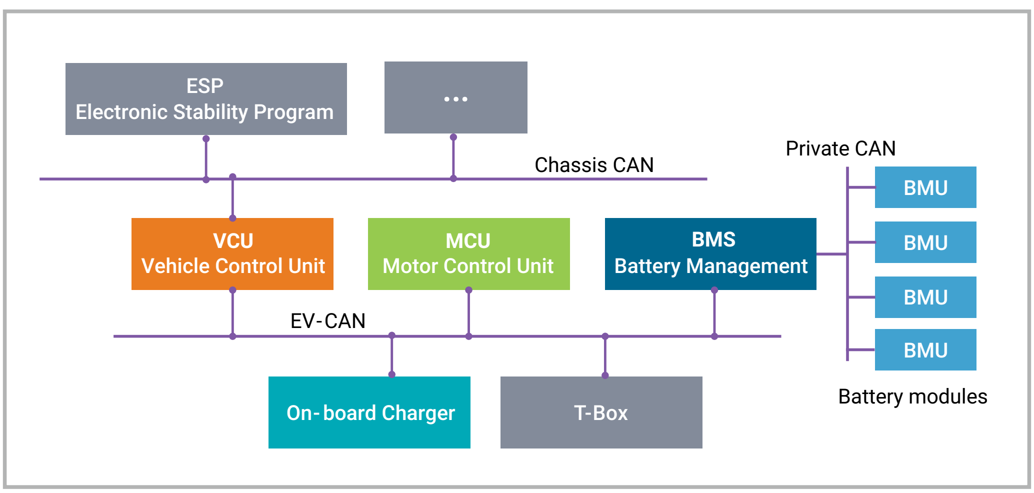

Figure 2: The typical EV architecture include numerous ECUs

Virtual Prototyping Overview

In its most basic definition, a virtual prototype is a system-level software model, generally composed from numerous models for subsystems and components, that can be simulated to replicate the behavior of the physical system. In this process, the design of the system can be verified to a large extent before building any physical prototypes or performing any bench testing. This minimizes the need for lab prototypes and bench setups and enables fault insertion and other forms of verification that could be destructive to a physical system. Because it is much easier and less expensive to provide many copies of a virtual prototype, many more engineers on a project can benefit. For EV development, beneficiaries include:

- Controls systems developers
- Application software developers
- Firmware developers

- Power electronics engineers
- Battery management system developers
- Motor drive engineers
- Reliability engineers
- Functional safety engineers
- Calibration engineers

As described earlier, the benefits of virtual prototyping for EVs extend beyond verifying that the system behaves as expected. The model also aids in the system-level architecture and high-level design process, enabling exploration of design options and evaluation of tradeoffs for performance, efficiency and cost. Because the virtual prototype spans the complete electro-mechanical design, it can model realistic responses from the complete system, including hardware/software, environmental effects and even manufacturing variability. A unified virtual prototyping solution covers system to software, including:

- Open, standards-based connectivity with industry tools.
- Simulations covering multiple levels of abstraction
- Highly abstract concept studies
- Detailed electrical response and transient analysis studies

The Synopsys Solution

As the industry leader in electronic design automation (EDA), Synopsys has a wide range of technologies and products that can be employed to address complex design and verification challenges. Building a unified virtual prototype for EV design and verification requires a mix of technologies to cover the full range of system to software, including electro-mechanical components and complex interaction between embedded software and the hardware components. Such a prototype is constructed with a stack of Synopsys products, including SaberRD, Silver, Virtualizer™ and TestWeaver®. Figure 3 represents a typical EV product showing, from the bottom up, the mechanical hardware and the electrical and power electrics required to drive the plant, and a typical AUTOSAR application software stack running on a microcontroller.

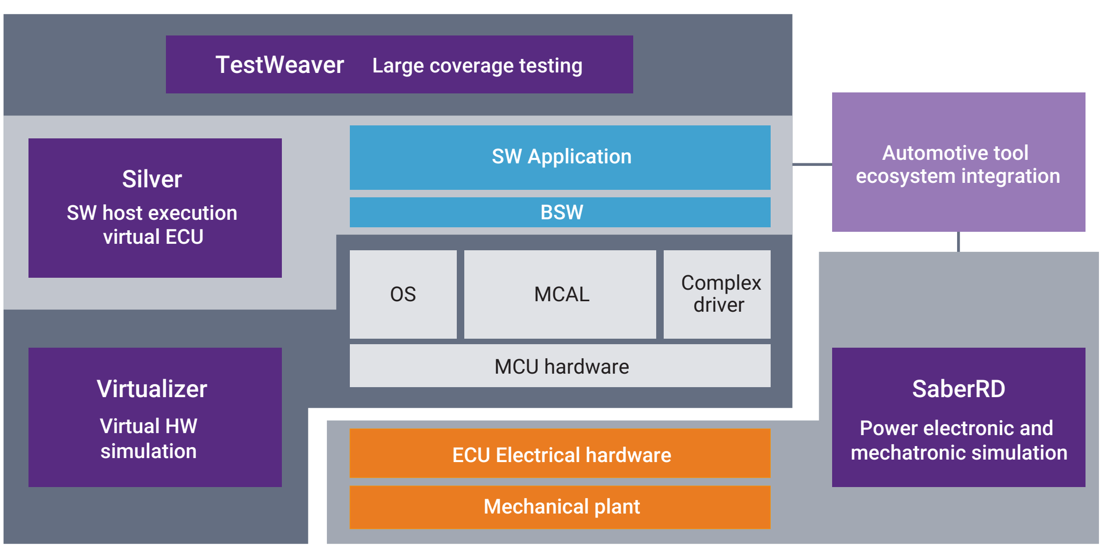

Figure 3: Synopsys EV virtual prototyping solution stack

Synopsys Saber™ is a proven platform for designing, modeling and simulating physical systems, enabling full-system virtual prototyping for applications in analog/power electronics, electronic power generation/conversion/distribution, and system/wiring/harness design and mechatronics. SaberRD provides comprehensive electro-mechanical simulation while modeling a wide range of component types at various levels of abstraction. Abstraction levels range from high-level architectural models (MIL) for extremely fast execution of architectural studies down to detailed behavioral models that mimic the physical attributes of the actual components. Synopsys provides an industry-leading electrical model database to support detailed transient analysis and efficiency optimizations.

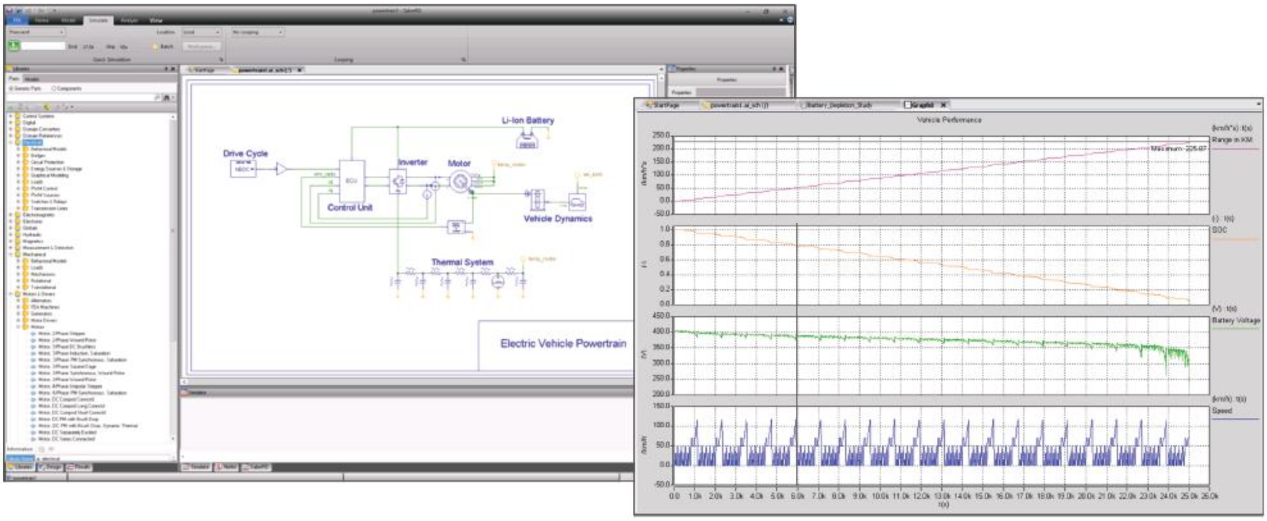

Figure 4: SaberRD detailed electro-mechanical analysis

SaberRD can provide accurate results for a wide range of EV studies such as the use of drive cycles that capture driver and vehicle operation. Drive cycles are produced by multiple countries and organizations to assess vehicle performance. Drive cycles are quite different for city driving, highway driving and rural driving. In addition, drive cycles for partially or fully autonomous vehicles are distinct from those of human drivers. Using these variations allows the EV developer to optimize the design for the most common driving profiles of the target users.

SaberRD provides a state-of-the-art functional safety solution to meet the essential requirements for functional safety by fault effect analysis in the ISO 26262 standard. Engineers can investigate thousands of fault effects and fault mitigation virtually, saving the expense of physical prototypes and handling fault types that can't be investigated in bench testing. High-level fault-effect coverage and efficiency in fault-effect simulation are achieved by automating the analysis and result validation. This enables the development and verification of safety mechanisms while building confidence in the design's ability to tolerate such effects as component tolerances, temperature dependencies, aging, parameter drifts, and stress levels. SaberRD is ISO 26262 certified and an essential part of EV safety analysis.

To address software development execution, developers can leverage either virtual hardware model (Virtualizer) or host-based execution (Silver).

Synopsys Silver is a virtual ECU (vECU) platform, used to move development tasks from road tests and bench rigs to a Windows PC for efficient software in the loop (SIL) development of ECU software. Engineers can build vECUs that closely mimic the behavior of their real counterparts. The virtual ECUs are built from C code or from target binaries. Silver is also a powerful experimentation environment for validating and testing the interaction of networked ECUs, engine, transmission and other vehicle components through simulation. Silver can run simulation models for various tools (including MATLAB/Simulink, Synopsys Saber and many others through the standardized FMI interface) on the PC without having these tools installed in the execution environment.

Depending on the use case and the availability of source code, different parts of the ECU software can be virtualized and ported to the PC: only the application software (ASW), or also parts of the basic software (BSW). A Silver configuration can be duplicated quickly, allowing every engineer to have a personal development environment without blocking scarce resources such as HIL bench rigs or test vehicles.

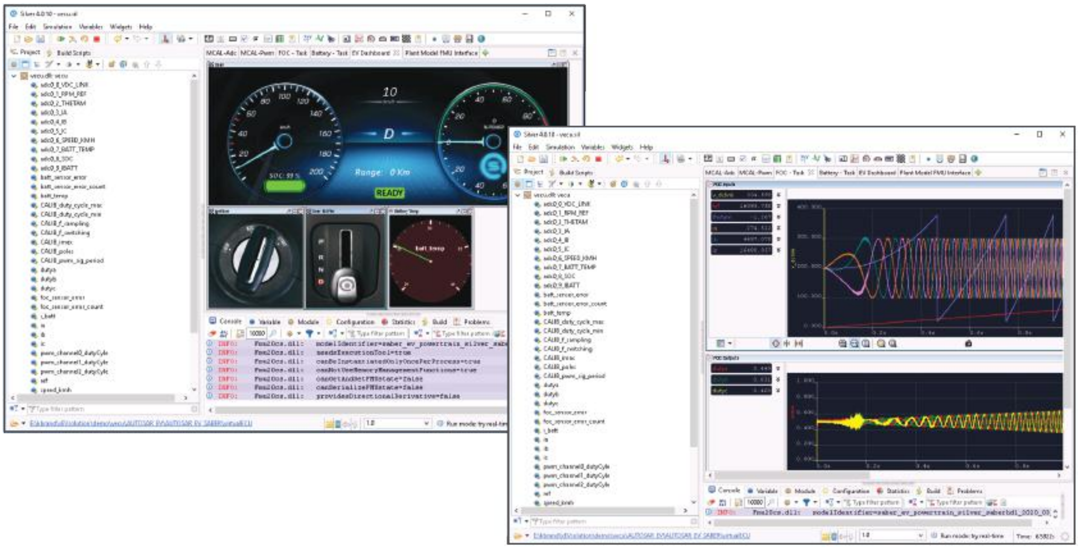

Figure 5: Silver high-level view of ECU behavior

Silver abstracts the ECU hardware and focuses on the software and bus communications. Abstracting the hardware away decouples software and hardware development and reduces simulation complexity and setup. No hardware models are needed.

Synopsys Virtualizer Development Kits (VDKs) simulate the ECU hardware, including processor and on-board peripherals. They enable the execution of the binary software stack, which allows more use cases at the operating system and hardware/drivers level. A VDK comes with significant benefits when developers need to address system level issues by observing both hardware and software behavior and are interested in full software stack verification, including ASW, BSW, MCAL and drivers. Synopsys Virtualizer includes the following features for software developers:

- Full interactive hardware/software debug visibility
- Integration with third-party software debuggers
- Powerful system-level analysis
- Python scripting to interact with the virtual prototype
- Fault injection at the hardware and software level

Several microcontroller models from leading vendors such as Infineon, Renesas, ST and NXP are supported.

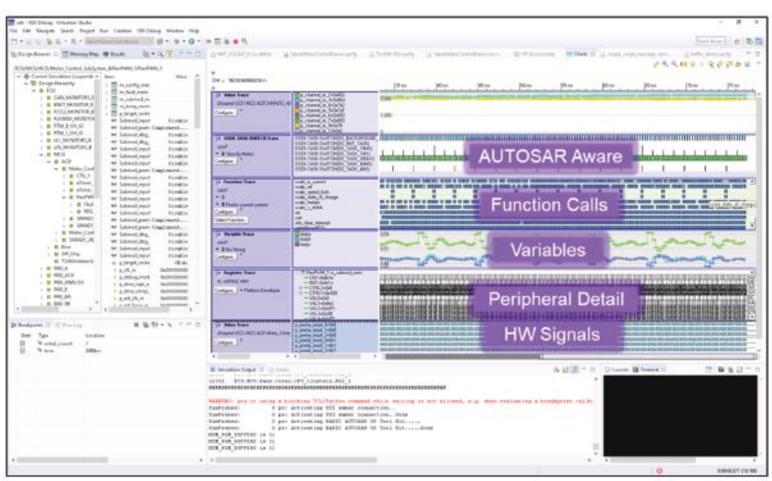

Figure 6: Virtualizer Studio embedded software development

Synopsys TestWeaver is an intelligent test automation solution, creating and running system-level tests to detect errors while achieving maximal test coverage with minimal specification effort. Functional requirements are modeled as system invariants using the simple but powerful Requirement Modeling Language (RML). RML models are translated by TestWeaver into C/C ++ code and provided in executable form to the simulation. TestWeaver generates thousands of scenario variations using self-learning algorithms to maximize the coverage of:

- Hardware states
- Software states
- Environmental conditions
- Driver maneuvers
- Parameter tolerances
- Component faults

Thousands of functional requirements are systematically modeled and continuously monitored in all scenarios. The density of the monitored conditions combined with the large coverage of the applied test yields outstanding results for the systematic verification of complex EV systems.

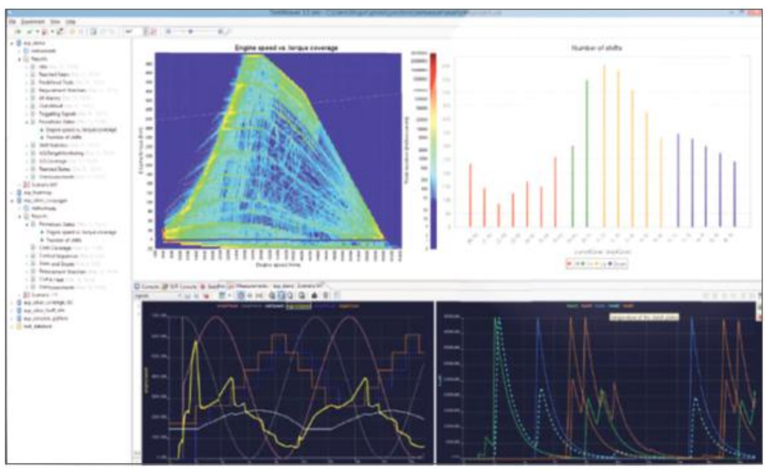

Figure 7: TestWeaver visualization of scenarios and coverage

Synopsys EV Virtual Prototyping Flow

The Synopsys unified virtual prototyping solution for EV development supports hardware, software and the complete system. Figure 8 shows a typical project flow using the Synopsys solution. The process starts with product requirements and the use of SaberRD to perform high-level, abstract MIL studies. The system model is refined as software engineering (SWE) enters the process, with Silver running SIL simulations to develop ECU software and test the interaction among EV components. As the software is refined to create the production code that will run in the vehicle, Virtualizer runs the actual binaries for highly accurate and detailed vHIL simulations of system behavior.

The use of Virtualizer continues as the entire system is integrated together. Silver comes back into play to run system-level tests for a high level of coverage, using TestWeaver to create the tests, manage the process, and maximize the coverage achieved. This verification and validation (V&V) phase would be very labor-intensive and time-consuming without the automation that TestWeaver provides. Finally, SaberRD provides high fidelity modeling capabilities for power electronics and integration with software control, providing further refinement for the final steps of verification and test.

The SWE portions of the Synopsys flow support the Automotive SPICE® (ASPICE) Process Reference Model and Process Assessment Model, specifically the following stages of the Software Engineering Process Group:

- SWE.3—Software Detailed Design and Unit Construction
- SWE.4—Software Unit Verification
- SWE.5—Software Integration and Integration Test
- SWE.6—Software Qualification Test

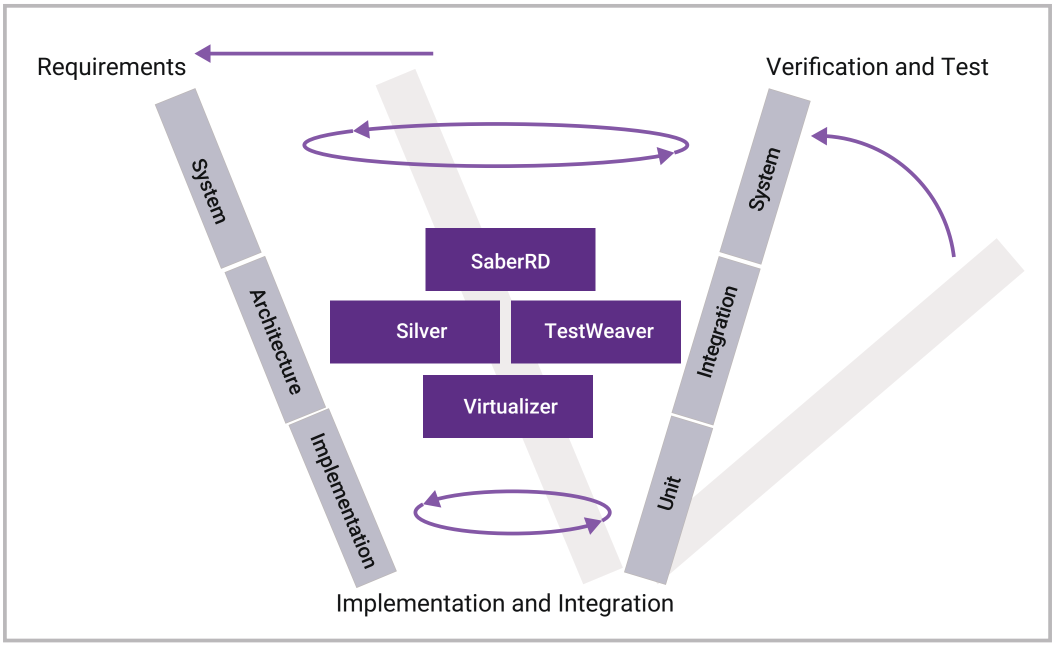

Figure 8: Synopsys EV virtual prototyping flow

The Synopsys flow offers several benefits to EV teams. They can start many aspects of design and verification earlier in the project schedule, before silicon, hardware or testbenches are available. Higher system visibility accelerates software debug and analysis, shortening the overall schedule. The flow enables development of safer vehicles by testing faults and corner cases that would be dangerous or impossible to reproduce in hardware. The virtual prototypes are scalable, easily duplicated in server farms to increase testing throughput or support additional users. These users can access the virtual environment from anywhere in the world at any time, with none of the access limitations of traditional bench testing. These benefits greatly increase the chance of successful products in the accelerating and competitive EV automotive sector. The Synopsys solution provides differentiated design tooling and technology, meets the challenges of a system to software flow, delivers a robust EV design and reduces time to market.

Conclusion

Electric vehicles are growing in popularity and will take a significantly larger share of the market over the next ten years. Development of electro-mechanical systems for vehicular applications is always challenging, but EVs raise the bar significantly. The cost of physical prototypes is too high to make them available to all engineers who can benefit, plus it's vital to "shift left" the design and verification process to much earlier in the project timeline. This white paper has presented a proven and available alternative: an EV development flow using virtual prototypes based on several industry-leading Synopsys products. This flow enables exploration of design options, evaluation of tradeoffs, development of embedded software and multiple layers of verification before any hardware is built. The virtual prototype can be replicated easily for all project engineers for much more efficient and robust EV development.

06/24/20.CS523045054 VG WP _ Virtual Prototyping for Electric Vehicles WP.

Pub: June. 2020