2021

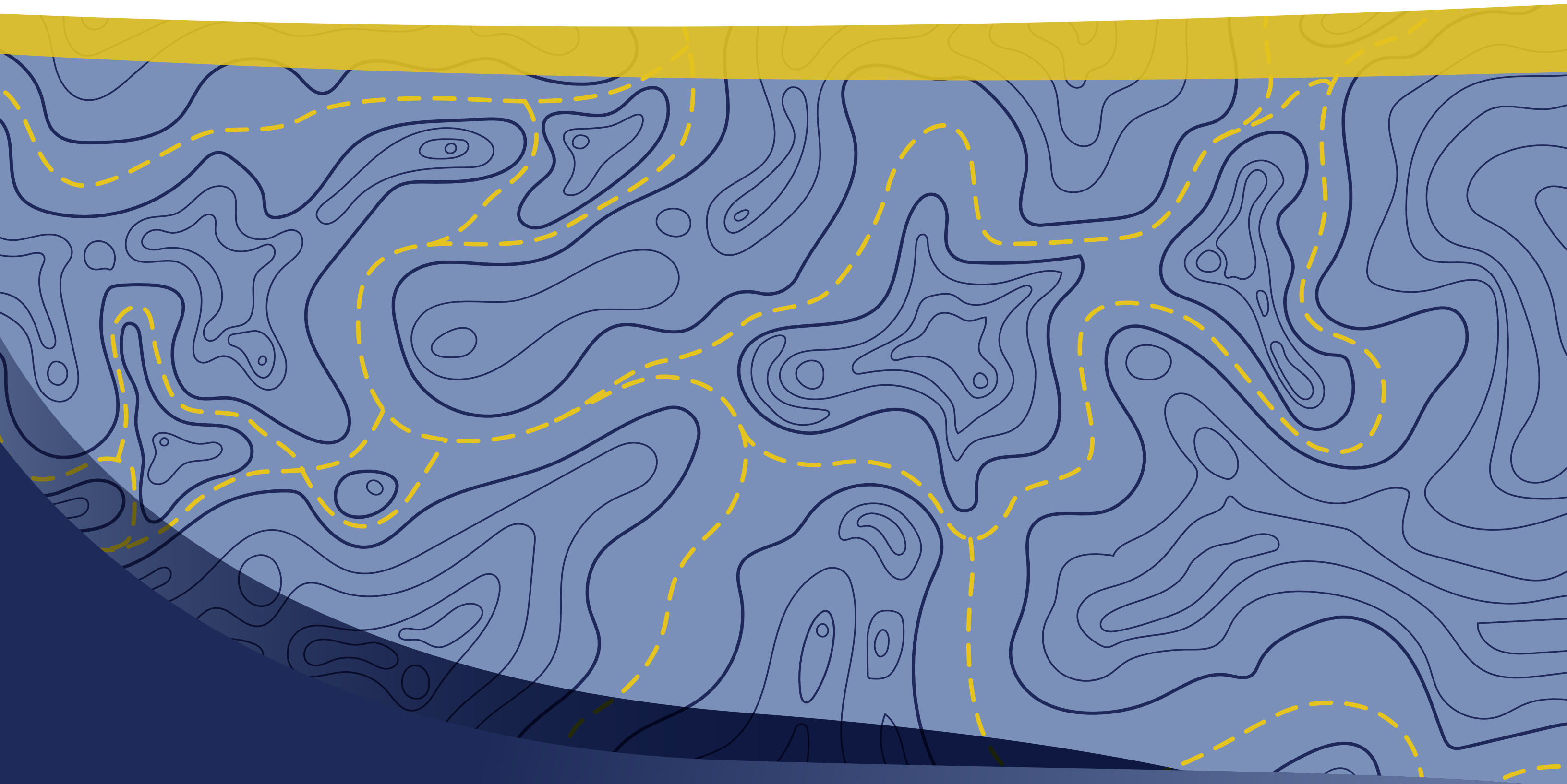
DESIGN AND VERIFICATION™

DVCON

CONFERENCE AND EXHIBITION

EUROPE

OCTOBER 26-27, 2021



DVCON EUROPE 2021 CONFERENCE PROCEEDINGS

SPONSORS AND ORGANIZERS

accellera

SYSTEMS INITIATIVE

Please visit website for
more information!

dvcon-europe.org

Table of Contents

DVCon Europe 2021

Welcome Message from DVCon Europe General Chair, Sumit Jha

General Chair - Qualcomm Technologies

A warm welcome to DVCon Europe 2021!

First of all, I hope that you and your loved ones are doing well and that all of you are in good health.

In this covid era, we have realized a new set of problems and opportunities which need to be addressed. Our generation is witnessing the steepest technological growth ever seen and innovations and modern gadgets are more of a necessity today than a luxury. This moment in time, really validates the need and the importance of semiconductor technology and how that allows us to be in much more connected world and enable completely new use cases.

And what this means is that the complexity of chip design and verification challenges are growing exponentially in the semiconductor industry. To keep us synced to the demand of advanced methodologies, tools, and techniques, we understand that DVCon Europe has a special responsibility. Especially when most of our peers are working from home office with a very little opportunity to share ideas.

The Design and Verification Conference (DVCon) and Exhibition provides attendees around the globe with the industry's most comprehensive technical program focused on the design and verification of electronic systems. It is also that time of the year when we celebrate, promote and reward some of the brilliant ideas from the authors who are carefully shortlisted from the hundreds of submissions. We also get an opportunity to hear from industry leaders in form of keynotes and panel sessions.

DVCon Europe, together with Accellera's co-located SystemC Evolution Day, is all set to provide an extraordinary stage to showcase the finest innovative ideas, use cases, methodologies, languages and standards which are helping you today and going to help our engineering community tomorrow. You will also get an insight to the latest Accellera system initiative activities, standards and working groups.

Last year, DVCon Europe 2020, was a huge success. It was the first 'virtual' DVCon with so many unknowns but despite all the challenges, the virtual conference was very well organized and well received by the wider audience from 90 organizations and as many as 28 countries.

While we missed a lot being in Munich and meeting our peers at our usual venue in the hotel, however, as every cloud has a silver lining, we also observed few blessings in disguise from the conference being virtual. It was much easier for the companies to send their engineers to DVCon last year, as the total out of office time was reduced to half and the cost to company was reduced significantly with no travel and accommodation needed. Also, at an individual level, it was very convenient for an attendee to attend the sessions live and also be able to watch some of the recordings later. Our experiment with virtual experience room was very successful and the audience appreciated the importance of being able to interact and network with peers (avatars) in (virtual) reality. Obviously, huge efforts went into bringing up this virtual conference setup from scratch and this is a testament of the resilience and the passion of our steering committee and sponsors to serve the industry even in a trying time.

Given the current scenario and uncertainty in terms of covid variants and vaccinations, we are happy to organize this 8th edition of DVCon Europe as a virtual conference and exhibition. This year we plan to have more improvements and advancements in the 3D virtual experience. As always, we will have technical paper presentation, tutorials, panel, and keynotes, with the same spirit of organizing this prestigious conference for the users by the users.

Let's join hands because if you have an idea, we will show it to the world!

Sumit Jha - DVCon Europe 2021 General Chair

CONFERENCE SPONSOR

Accellera Systems Initiative is an independent, not-for profit organization dedicated to create, support, promote, and advance system-level design, modeling, and verification standards for use by the worldwide electronics industry. We are composed of a broad range of members that fully support the work of our technical committee to develop technology standards that are balanced, open, and benefit the worldwide electronics industry. Leading companies and semiconductor manufacturers around the world are using our electronic design automation (EDA) and intellectual property (IP) standards in a wide range of projects in numerous application areas to develop consumer, mobile, wireless, automotive, and other "smart" electronic devices. Through an ongoing

partnership with the IEEE, standards and technical implementations developed by Accellera Systems Initiative are contributed to the IEEE for formal standardization and ongoing governance.

Our Mission

At Accellera our mission is to provide a platform in which the electronics industry can collaborate to innovate and deliver global standards that improve design and verification productivity for electronics products.

The purposes of the organization include:

- Provide design and verification standards required by systems, semiconductor, IP, and design tool companies to enhance a front-end design automation process.
- Collaborate with our community of companies, individuals, and organizations to deliver standards that lower the cost of designing commercial IC and EDA products and embedded system solutions, as well as increase the productivity of designers worldwide.
- Encourage availability and adoption of next-generation EDA and IP standards that encompass system-level, RT-level, and gate-level design flows.
- Collaborate with the electronic design community to deliver standards that increase designer productivity and lower the cost of product development.
- Provide mechanisms that enable the continued growth of the Accellera Systems Initiative user community including SystemC, Universal Verification Methodology (UVM), and IP-XACT.
- Standardize technical implementations developed by Accellera Systems Initiative through the IEEE.

Accellera Global Sponsors

2021 COMMITTEES

Steering Committee

General Chair

Sumit Jha, Qualcomm Technologies
sumitj@qti.qualcomm.com

Tutorial Chair

Stephan Gerth, Bosch Sensortec GmbH
Stephan.Gerth@bosch-sensortec.com

Technical Program Chair

Alexander Rath, Infineon Technologies AG
alexander.rath@infineon.com

Poster Chair

Oliver Bell, Intel Corp.
oliver.bell@intel.com

DVCon Global Coordination

Martin Barnasconi, NXP Semiconductors
martin.barnasconi@nxp.com

Virtual Platform Chair

Mark Burton, GreenSocs
mark.burton@greensocs.com

Vice Chair

Joachim Geishauser, NXP Semiconductors
joachim.geishauser@nxp.com

Panel Chair

Tran Nguyen, ARM
tran.nguyen@arm.com

SystemC Evolution Day Chair

Ola Dahl, Ericsson AB
ola.dahl@ericsson.com

Promotions & Press

Annette Bley, Annette Bley PR
annette@annettebleypr.com

Accellera Representative/Finance Chair

Lynn Garibaldi, Accellera Systems Initiative
Lynn@accellera.org

Technical Program Committee

Program Chair

Alexander Rath
Infineon Technologies AG

Vice-Program Chair

Mark Burton
GreenSocs

Rafal Baranowski
Robert Bosch GmbH

Markus Buecker
Advantest

Mark Burton
GreenSocs Ltd

Jerome Cornet
STMicroelectronics

Philippe Cuenot
IRT Saint-Exupery

Karsten Einwich
COSEDA Technologies

Mohammad A Fahim
Infineon Technologies AG

Harry Foster
Mentor Graphics
Corporation

Edgar Iglesias
Xilinx

Eric Jenn
IRT Saint-Exupery

Paul Kaunds
P&C

Thomas Klotz
Bosch Research

Tim Kogel
Synopsys, Inc.

Alastair Lefley
ams AG

Djones Lettnin
Infineon Technologies AG

Mark Litterick
Verilab, Inc.

Alessandra Nardi
Cadence Design Systems, Inc.

Jasminka Pasagic
NXP Semiconductors

Frederic Petrot
Universite Grenoble Alpes

Clemens Roettgermann
excida.com LLC

Martin Ruhwandl
Infineon Technologies AG

Christian Sauer
Cadence Design Systems, Inc.

Uwe Simm
Cadence Design Systems, Inc.

Sebastian Simon
Infineon Technologies AG

Clemens Suessmuth
ams AG

Walter Tibboel
NXP Semiconductors

Darko Tomusilovic
VTool Ltd.

Nicolae Tusinschi
OneSpin Solutions GmbH

Andrei Vintila
AMIQ

Thilo Vörtler
COSEDA Technologies

Joen Westendorp
NXP Semiconductors

Roger Witlox
ams AG

CONFlux Platform

URL: dvcon-europe2021.org

This platform will be used for the following:

- Display DVCon Europe Schedule with live Zoom links to sessions
- Provide an attendee page and opportunity to contact fellow attendees
- Exhibit pages for all sponsors & exhibitors
- Live session recordings will be posted on the platform for viewing after the sessions take place
- There will be opportunities to ask questions on presentations asynchronously so authors can respond throughout the conference
- Opportunities for chat sessions related to specific topics
- The platform will be accessible to registrants through November 27, 2021

Virtual Experience Room

DVCon Europe 2021 will bring you a true virtual experience by introducing Virtual Experience Room! This virtual space rendered in 3D are well known in the gaming industry.

DVCon Europe 2021 will use this Virtual Experience Room for the poster session, panel sessions, and to enable interaction, collaboration and networking during the 'coffee breaks' and encourage 'chatting' and discussing topics amongst the conference participants outside of the presentation sessions.

The Virtual Experience Room provides a virtual meeting space for attendees to interact and participate in the conference without the need for virtual reality gear. will be accessible directly through CONFLux.

DVCon Europe 2021 - Technical Program - Day 1 - Tutorials & Panel

26 October 2021

TIME (CEST)	Stream 1	Stream 2	Stream 3	Stream 4	Other Events
8:30 - 9:00	**Opening & Welcome**				**Virtual Experience Room & Networking**
9:00 - 10:00	**Keynote: Rashid Attar** 5G, AI and Compute Platform Innovations for What's Next				
10:00 - 10:30	**Virtual Coffee Break**				
10:30 - 11:30	T1.1 Prototyping Accelerators using Intel® Integrated Simulation Infrastructure with Modeling (Intel® SIM) **Sponsored by:** intel	T2.1 Accelerating Analog/ Mixed-Signal Design and Verification through Integrated Rapid Analysis **Sponsored by:** MathWorks®	T3.1 Accelerate Signoff with JasperGold RTL Designer Apps **Sponsored by:** cadence®		
11:30 - 12:30	T1.2 Artificial Intelligence in ASIC/SOC Verification	T2.2 Python and SystemC: A Dream Team for Building and Analyzing Virtual Platforms	T3.2 Hardware-Aware, Model-Based Software Development to Speed Up Embedded Designs	T4.1 Test Driven Hardware Design and Verification	
12:30 - 13:30	**Virtual Lunch Break**				
13:30 - 14:30	**Panel: Anatomy of a Verification Flow**				
14:30 - 15:30	**Keynote: Dr. Petra Färm** Speed Layers: Managing the Exponential Change of Technology				
15:30 - 16:00	**Virtual Coffee Break**				
16:00 - 17:00	T1.3 Deep Cycle HW/SW Verification using High-Performance Prototyping Systems **Sponsored by:** SYNOPSYS® Silicon to Software™	T2.3 AI/ML Accelerator Verification Tutorial: High-Level Verification of C-level design **Sponsored by:** SIEMENS	T3.3 Automated Code Checks to Accelerate Top-Level Design Verification **Sponsored by:** onespin assuring IC integrity		
17:00 - 18:00	T1.4 An Update on the UVM-AMS Standard in Accellera **Sponsored by:** accellera SYSTEMS INITIATIVE	T2.4 Boost your Productivity in FPGA & ASIC Design and Verification **Sponsored by:** Sigasi;	T3.4 Collaborative, Advanced Fault Analysis: Addressing the Functional Safety Verification Challenges from the Accellera Functional Safety **Sponsored by:** OPTIMA		
18:00 - 18:30	**Closure Day 1 & Outlook Day 2**				

DVCon Europe 2021 - Technical Program - Day 2 - Paper, Posters & Panel

27 October 2021

TIME (CEST)	Stream 1	Stream 2	Stream 3	Stream 4	Other Events
8:00 - 8:15	**Opening & Welcome**				**Virtual Experience Room & Networking**
8:15 - 9:15	**Keynote: Satish Sundaresan** Take a Leap: Virtualization in Future Development				
9:15 - 9:30	**Virtual Coffee Break**				
9:30 - 10:00	P1.1 Machine Learning for Coverage Analysis in Design Verification	P2.1 Achieving Faster Code Coverage Closure using High-Level Synthesis	P3.1 Bringing Reset Domains and Power Domains together – Set/ Reset Flops Augmenting Complexities in Power-Aware RDC Verification	P4.1 Method for Early Performance Verification of Hardware-accelerated Embedded Processor Systems in RTL Simulation	
10:00 - 10:30	P1.2 SimPy and Chips: A Discrete Event Simulation Framework in Python for Large Scale Architectural Modelling of Machine Intelligence Accelerators	P2.2 Detection of Glitch-Prone Clock and Reset Propagation with Automated Formal Analysis	P3.2 A Methodology for Evaluating SI Artefacts in DDR4-3DS PHY using Channel Modelling	P4.2 System-Level Register Verification and Debug	
10:30 - 11:00	P1.3 Optimizing Design Verification using Machine Learning: Doing Better Than Random	P2.3 Using HLS to improve Design-for-Verification of Multi-pipeline Designs with Resource Sharing	P3.3 Using Dependency Injection Design Pattern in Power Aware Tests	P4.3 No Country for Old Men – A Modern Take on Metrics Driven Verification	
11:00 - 12:00	**Panel: Can ML be the Driver of Next-Generation Verification?**				
12:00 - 13:00	**Virtual Lunch Break**	Sponsor Sessions AGNISYS Jade design automation LUBIS EDA SIEMENS Veriest			
13:00 - 14:00	**Poster Session**				
14:00 - 15:00	**Virtual Coffee Break**	Sponsor Sessions AMIQ EDA cādence Corigine Fraunhofer IIS			
15:00 - 15:30	P1.4 A Novel Approach to Reuse Firmware for Verification of Controller based Sub-Systems using PSS	P2.4 Testbench Flexibility as a Foundation for Success	P3.4 A Comparison of Methodologies to Simulate Mixed-signal IC	P4.4 Machine Learning Based Structure Recognition in Analog Schematics for Constraints Generation	
15:30 - 16:00	P1.5 Reuse of System-Level Verification Components within Chip-Level UVM Environments	P2.5 Handling Asynchronous Reset(s) Testing by Building Reset-Awareness into UVM Testbench Components	P3.5 Unified Model/ Hardware-in-the-Loop Methodology for Mixed-Signal System Design and Hardware Prototyping	P4.5 Democratizing Formal Verification	
16:00 - 16:30	P1.6 A Novel Approach to Functional Test Development and Execution using High-Speed IO	P2.6 One Testbench to Rule Them All!	P3.6 Accelerated Coverage Closure by Utilizing Local Structure in the RTL Code	P4.6 Netlist Paths: A Tool for Front-end Netlist Analysis	
16:30 - 17:30	**Keynote: Andreas Riexinger** The Mobility of the Future is Software defined – Can Open Technologies help?				
17:30 - 18:30	**Closing Session & Best Paper/Poster Award Ceremony**				

THANK YOU TO OUR 2021 SPONSORS

GOLD SPONSORS

AMIQ EDA

LUBIS EDA

SIEMENS

SILVER SPONSORS

cadence®

Veriest

chipright
design and verification consultants
right from the start

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Jade
design automation

Sigasi;

ACADEMIC & BEST PAPER AWARD SPONSOR

Qualcomm

EXHIBITORS

Fraunhofer IIS

SIEMENS

Sigasi;

Corigine

AMIQ EDA

chipright
design and verification consultants
right from the start

Veriest

LUBIS EDA

cadence®

Jade
design automation

hdl Design House
SoC Design and Verification Company

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

THANK YOU TO OUR 2021 SPONSORS

TUTORIAL SPONSORS

cadence®

MathWorks®

intel®

onespin
assuring IC integrity

OPTIMA

SIEMENS

Sigasi;

SYNOPSYS®
Silicon to Software™

VIRTUAL EXPERIENCE ROOM SPONSOR

MathWorks®

MEDIA SPONSORS

TECH DESIGN FORUM

newelectronics

EDACafé

Elektronik i Norden

eeNews EUROPE

Technical Program: Tuesday, October 26

Time Zone is CEST

8:30 – 9:00

Opening & Welcome

8:30 – 18:30

Virtual Experience Room & Virtual Exhibitors

09:00 – 10:00

Keynote: 5G, AI and Compute Platform Innovations for What's Next

Rashid Attar

Vice President, Engineering Qualcomm Technologies, Inc.

Abstract: This keynote will explore some of the most compelling platform innovations in our industry – 5G to connect virtually everyone and everything, AI do so intelligently, Edge Computing to enable the right level of AI and compute where it makes most sense, and the synergy among them. 5G, AI and Edge Computing are not only synergistic but also advances in one enable further advances in the other two.

We will study examples of how this synergy will affect virtually everything we use – our smartphones, our cars and transportation systems, our energy grid, healthcare, and new products such as XR headsets, all while preserving our privacy, minimizing energy consumption and improving our user experience. We will also examine how these platform innovations enable the 'next' user interface to enhance a spectrum product segments with an exploration of boundless XR.

Lastly, we will address how we can meet the energy demands due to AI and Compute and conclude with desired advances in Silicon and Design technologies to continue fueling our platform innovations.

Technical Program: Tuesday, October 26 (cont.)
Time Zone is CEST

10:00 – 10:30
Virtual Coffee Break
Meet us in the Virtual Experience Room

10:30 – 12:30
Tutorial Sessions

Tutorial 1.1: Prototyping Accelerators using Intel® Integrated Simulation Infrastructure with Modeling (Intel® SIM)

Time: 10:30 – 11:30

Stream 1

Sponsored by: intel

Speakers: Jakob Engblom, Intel, Stockholm, Sweden

Dr. Jakob Engblom works at Intel in Stockholm, Sweden, as a product manager for the Simics simulator and as a simulation evangelist. He has been working with the Simics simulator since 2002, in a variety of roles including field engineering, product marketing, and training. At Intel, he helps engineers succeed with the Simics simulator, driving training, user support, and product improvements.

Abstract: Intel® Integrated Simulation Infrastructure with Modeling (Intel® ISIM) is a versatile framework that supports functional, performance, power, and thermal simulations in a single environment. It is based on the well-known Intel Simics® simulation framework, and available as a free public download from https://software.intel.com/isim.

In this tutorial, we go through an example of how to use Intel ISIM to build a virtual platform (VP) model of an accelerator attached to a general-purpose Intel Architecture computer, and try some accelerator architecture variants in the VP. We will use the standard Simics Quick-Start Platform (QSP) virtual platform, and model and attach a virtual PCIe card that offloads computation from the main processors. The tutorial explains the process to implement an accelerator model in Intel ISIM. The model encompasses the hardware-software interface of the accelerator, as well as different internal accelerator architectures. The software stack is provided, to keep the scope limited.

The tutorial will present the basics of model building in Intel ISIM, how to attach PCIe devices to a virtual platform, and how to model parallelism and estimated computation costs. The materials presented are available in the public Intel ISIM release, and you can download, install, and try it yourself.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 2.1: Accelerating Analog/Mixed-Signal Design and Verification through Integrated Rapid Analysis

Time: 10:30 – 11:30

Stream 2

Sponsored by: MathWorks

Speakers: Ganesh Raj Rathinavel, EMEA Application Engineer for Analog & Mixed Signal Design, MathWorks

Abstract: The design and verification of Analog/Mixed Signal (AMS) Systems have become increasingly challenging with analog impairment effects in modern semiconductor technology nodes and with the integration of complex embedded digital signal processing and control algorithms. Thus, accurate modeling and rapid system-level simulation are required to verify these designs prior to production. As these designs are created in different abstractions and even design flows, exhaustive verification routines are needed to ensure the consistency in functionality and compliance to specifications.

The tutorial will guide you through the entire verification process through different abstraction, from System Modeling to post-implementation integration. This session will demonstrate:

- modeling key AMS blocks at system level and simulating with testbenches to measure characteristics and imperfections
- integrating system-level behavioral models in SystemVerilog AMS environments such as Cadence Virtuoso®
- analyzing transistor-level simulation data by interfacing the Cadence Virtuoso® ADE Explorer and Assembler environments with data analysis tools such as MATLAB® and the Mixed-Signal Analyzer app

During this tutorial we shall demonstrate on how to model and simulate systems with the emphasis on achieving early verification of your solution. The integration of modeling tools such as MATLAB and Simulink with design tools such as Cadence Virtuoso provides system-level solutions for IC and PCB design. Simulate and synergize behavioral and device models to develop and prototype ughhhigh performance systems. Visualize, analyze, and identify trends in AMS circuit simulation data.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 3.1: Accelerate Signoff with JasperGold RTL Designer Apps

Time: 10:30 – 11:30

Stream 3

Sponsored by: cādence®

Speakers:

Bijitendra Mittra is a member of the JasperGold Product Engineering team responsible for architecting the next generation JasperGold apps with focus on CDC and RTL Sign-off solutions. He has over 20 years of experience in EDA and Design Verification. Over the last 15 years, he has been working on formal verification of IPs and SoCs, methodology development and tool development. He has more than 20 publications on Formal Verification and Static Checks in various international conferences, symposia and EDA design user groups.

Kanwar Pal Singh is part of the JasperGold Product engineering team for the past 6 years and have been with Cadence for the past 20 years. Prior to his career in Cadence he worked as design engineer at ST Microelectronics. At Cadence he has been mainly associated with the front end design and verification domain and have been driving the RTL Signoff solutions including LINT/DFT DRC and CDC. He has presented various papers in industry conferences and also holds multiple patents in the Static checks domain.

Abstract:

In this tutorial, we will take the attendees through using the JasperGold Superlint and CDC applications, which add formal verification technology and functional checks to the traditional structural checks for LINT, CDC and RDC. The JasperGold technology supports the designers to identify the real problem violations, confirming fixes, and providing justification for waiving the violations that are not problematic. Additional automatic formal checks are provided for functional verification of many aspects of the design, using properties derived automatically from the RTL. Workshop attendees will learn how these JasperGold RTL Designer apps combine to "shift left" these checks, providing a much more complete level of automated verification. The result is that RTL designers are able to sign off higher quality, more robust and LINT/CDC/RDC-clean designs, months earlier in the project schedule.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 1.2: Artificial Intelligence in ASIC/SOC Verification

Time: 11:30 – 12:30

Stream 1

Speakers: Paul Kaunds

Founder and CEO of P&C, a semiconductor services and solutions company with over 20 years' experience in the industry. Knowledge across the complete ASIC development flow from micro-architecture specification, RTL implementation, through to functional verification and silicon validation. Extensive knowledge of advanced verification (including SystemVerilog, UVM, eRM) and the use of new methodologies, tools and flows to deliver the best possible verification solution.

Around 2 decades of extensive industry experience into Architecture, Design and Verification of IPs, Subsystems, SoCs, ASICs, FPGAs, Test chips & VIPs.

Abstract: With the ever-increasing complexity and the abundance of data to analyse during debugging, the ASIC /SOC verification engineers are facing the situations where the time consumption for analysis during debug is ranging from hours to days. Major contributor for that is finding out the sweet spot where the issue lies. As no part of the verification environment goes into the chip it has been a practice across verification environment developers not to streamline the testbenches in such a way that it automatically catches the sweet spot. The main hurdle to concentrate in this aspect is that it will be complex and cumbersome effort to identify the sweet spots as there will be multiple protocols involved and the complexity of the protocol also adds to that. The problem is more complex if the debug is at SoC level where the data path is not limited to IPs alone but also includes the processors. Debugging takes long because there are big-data sets to analyse without proper tools. Debugging is a series of questions, assumptions, and conclusions. Traditional debugging flow starts with huge log files from different sources. Writing down the numbers like timestamp and invoked code line numbers from the log files. Dumping the waves to analyse the correctness at each eventful clock cycle.

While semiconductor verification techniques have evolved considerably over the last 25 years, the debug of design problems found during verification has barely changed. Since the inception of Hardware Description Languages (HDLs) and simulation we have viewed verification output in the same way we analysed discrete digital circuit cards, through the lens of the logic analyser. It is true that there are new windows on top of the good old-fashioned waveform tool, and the GUI is a bit fancier, but fundamentally we look at tool output, signal-by-signal, one time tick at a time. Verification has changed radically. Designs have grown, leveraging reused intellectual property. Design complexity has also changed with multi-core onboard processing, advanced algorithms, and high-performance communication structures. Simulation has got faster and faster, and given way to emulation and formal verification, with different characteristics in terms of data storage and use models. All these developments have put a strain on the debug tools. Be it efficiently processing large data sets, visualizing complex components, or examining intricate test scenarios, all these activities are harder. This has led to debug occupying more than a third of the total verification effort. The ultimate result of a missed or incorrectly repaired bug is a full fabrication re-spin, at great expense both in terms of money and time. It is time to re-examine the entire debug process and leverage new algorithms and capabilities that may be available to us, but without throwing away the good work performed over the last twenty-five years. New algorithms including Machine Learning, visualization approaches, and problem-solving ideas allow a different approach to debug that saves up to an order of magnitude in debug time.

In this tutorial we will be presenting AI concepts and AI concepts applied to verification debug solution . It enables a logical methodology allowing engineers to visualize the essence of data and track down the root cause to problems in an efficient and error-free manner effectively verifying large-scale designs using emulation and regression simulation offering significant improvements right across the verification process.

There are seven types of bugs that can be found in any design which are

- Basic design code error
- Bad connectivity
- Mis-understood requirement
- Bad test vectors
- Mis-understood interaction with other blocks
- Timing issue
- Complex corner cases

To alleviate these issues, a rigorous verification process must be in place at both the block level and the SoC level. The debug process starts when a failing test is indicated and ends when we are sure of the root cause of the failure. In essence, it consists of a chain of assumptions, including questions and validations on those assumptions. Some assumptions are so obvious that they are not even considered as such. We simply "know" them to be true. However, if these assumptions are incorrectly made, they can sometimes create a huge delay in the root-cause analysis process. If we can decrease the number of process steps or avoid them completely while increasing the number of right answers, then our goal will be achieved.

Artificial Intelligence (AI) will be helping the verification process by making the debug faster. How it is going to make it faster and more efficient will be the game changer in verification process.AI takes abstract and visual approach to debug that solves many of the issues noted above. AI processes data from broad range of inputs including simulation UVM logs, VIP logs, Emulator logs, software messages and waveform databases and should be able to present them as one comprehensive high-level view. Additionally, it will be analysing large-scale data by applying modern Machine learning algorithms in a cognitive manner. AI is going to reduce the effort in large scale block level verification along with complex SoCs by tracking down intricate corner cases from SoC verification runs. AI minimizes the large data dumps required for traditional signal level analysis, making it ideal for emulation debug as well as large scale simulation regressions.

- All in all, AI is providing the following improvements to the debug process:
- Reduces the chance of taking an incorrect assumption for granted.
- Supporting the process of asking the right questions.
- Validating assumptions or answering questions faster.
- Revealing a possible path that could otherwise have been overlooked.
- Accelerates debug time, up to an order of magnitude, for a broad range of complex bug types. Given that debug represents 25% of the entire development time of a semiconductor, this represents a huge resource saving and time-to-market advantage.
- Improves design and verification quality over and above coverage assessment through visibility into verification scenarios, allowing a clear understanding of convoluted design code for easy team communication and cooperation.
- Extends debug for large-scale system verification on emulation, tracking down complex issues directly without the need for re-simulation, while working cohesively with existing debug and simulation environments.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 2.2: Python and SystemC: A Dream Team for Building and Analyzing Virtual Platforms

Time: 11:30 – 12:30

Stream 2

Speakers:

Eyck Jentzsch, MINRES Technologies GmbH, General Manager

Eyck Jentzsch holds a Dipl-Ing. from the Technical University Ilmenau and has more than 25 years experience in microelectronics and semiconductor design. He is working at MINRES as General Manager and focuses on virtual platform modelling, development, and application. Prior to that hw worked at Cadence Design Systems Inc. and Siemens in various full- and semi-custom as well as system level design and verification positions.

Thomas Haber, Toem GmbH, Founder

Thomas Haber holds a degree in engineering/robotics from the University of Wuppertal and has more than 20 years of experience in semiconductor development. He is the founder of toem GmbH and is involved in the visualisation and analysis of complex semiconductor and multicore software systems. Previously, he held various positions at Toshiba, Infineon and Intel.

Abstract: Today's electronics devices are dominated by 'system on chips' (SoCs). A complex system like a SoC involve many development teams with various focus areas. Examples are HW architects, HW verification teams, FW developers, SW verification teams to name just a few. Each of those teams often require a slightly different view of the full system. A 'one thing fits all' approach is very unlikely to find, at least before the HW is available. This situation drives a variety of techniques during development e.g., virtualized models, Virtual Platform models, FPGA accelerators, HW emulation, HW simulation.

The various use cases quite often require a certain amount of flexibility. For instance, to evaluate architectural variants, support product variants or partitioning of complex systems in order to keep simulation speed reasonable. Also augmenting the simulation with stimulus and analysis capabilities to gather information out of the generated data is commonly required. In practice scripting is being used to integrate all the various elements and steps addressing the VP user's needs.

This tutorial aims to give an overview of the development and deployment of VPs as well as results analysis using Python as scripting language. Use cases are not only in the semiconductor vendor domain but also at its customers. The use cases will be illustrated using (live) demos of the various approaches and the combined use of the elements. The tutorial will also show how Python can be efficiently used while maintaining the performance by connecting to high performance models written in C++.

The first part focuses on building and running VPs and variants based on Accellera official PySysC. It will highlight motivation and background to the development of the Python module. Here the use of PySysC will be demonstarted to get to a kind of 'VP Construction Box' approach allowing easy and flexible virtual platforms composition. It will also show advanced uses of the methodology to partition VPs for simulation speed improvement or quickly building user interfaces to ease working with the platform simulation e.g. for embedded software development.

Throughout the second part emphasis is put on integrating Python scripts and models like AI/ML models or connectors to host resources into a SystemC/TLM based model. This can drive flexible, fast prototyping of units to be validated in SoC context as well as designing and refining interactions of the SoC parts with these models and resources. An interesting aspect can be to use them directly as reference model for further HW implementation. The tutorial will show how such hybrid models can be enabled using adapters.

The third part of the tutorial will present ways to flexibly analyze generated data from various sources like traces and log files to extract information and root cause effects. It will demonstrate how a performance analysis framework can be constructed using Python to allow fast and flexible analysis of various aspects of simulation results. Python allows here to easily combine elements to aggregate data and distill information as well as to visualize this information in an accessible way.

Python allows a wide range of customizable visualization from waveform like traces to statistical information. Since the information presentation strongly depends on the model contents and the use case, Python enables here a new data driven information and presentation techniques. These techniques will be illustrated using an industrial model and dataset.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 3.2: Using Hardware-Aware, Model-Based Software Development to Speed Up Embedded Designs

Time: 11:30 – 12:30

Stream 3

Speakers:

Irina Costachescu, Model Based Design Software Engineer, NXP Semiconductors

Irina Costachescu is a Software Engineer with the Model-Based Design Team at NXP Semiconductors in Bucharest, Romania. She holds a Bachelor's Degree in Automatic Control and Systems Engineering and a Master's Degree in Complex Systems, both from the Polytechnical University of Bucharest. During her studies Irina deepened her control systems theory knowledge by developing robust and optimal algorithms for various systems control while also being a teaching assistant for this subject. At NXP, Irina contributes to all the development phases, from framework design to peripherals support, working collaboratively with internal teams for integrating other tools into the Model-Based Design Toolbox, dedicating her time also to creating toolbox related webinars, videos, and training sessions.

Mauro Fusco, Application Engineer - Design Automation and Code Generation, MathWorks

Mauro Fusco is an Application Engineer at MathWorks in Eindhoven. He specializes in supporting customers in Aerospace, Automotive and Machinery industries for the establishment of Design Automation workflows. Modelling, simulation, testing and implementation through automatic code generation whilst conforming to international standards are key aspects of his work.

Before joining MathWorks, he worked for 5 years at the Dutch Organization for Applied Research, TNO, focusing on the domain of Controls for Cooperative and Autonomous Driving. Mauro has a Masters in Automation Engineering from the University of Naples Federico II, during which time he conducted research at Eindhoven University of Technology. His technical expertise lies in the areas of Control Theory, Non-linear and Network Control and their implementation.

Other Contributors: Razvan Chivu, Software Engineering Manager, NXP Semiconductors

Abstract: The complexity of Embedded Software Applications is significantly increasing, especially in the automotive domain, with the number of lines of code in today's cars exceeding 100 Million. This is driving the need of embedded software engineers to work at higher levels of abstraction and to move to model-centric development workflows, such as Model-Based Design, defined as "The systematic use of models throughout the development process".

Such an approach allows engineers to model software algorithms together with the environment they are designed to interact with, validate them through simulation and generate the corresponding embedded code. This workflow provides several advantages, as it permits to shorten the development time by shifting-left verification and validation, automatically generating embedded code and abstracting from the hardware architecture. On the drawback side, this approach, if kept fully hardware-agnostic, would lead to sub-optimal and incomplete code, since the hardware-specific aspects can't be fully exploited and integrated. Two typical examples of such aspects, playing a key role in software optimization and integration, are device drivers and hardware accelerators. In this tutorial we will show how to overcome these drawbacks through a hardware-aware, model-based software development workflow.

We will present a streamlined way to prototype and develop complex applications and validate them using simulation. After that we will demonstrate how to generate code that links to platform-specific low-level software libraries (such as device drivers, optimized compute libraries, etc.) and we will also see how the generated code can be easily deployed to the target hardware, in our case a Microcontroller Unit (MCU).

Workflow and Key Features

We will start the journey by focusing on a detailed description of the tools used in this workflow, thus describing an ecosystem offering support for all the development phases of an application. We will go through:

- MCU peripherals support, illustrated in a wide range of examples
- Integration of platform software, drivers, and libraries
- Generation of C code from models
- Compiler options to build and link C code into an executable file
- Options to download a software image to the target MCU
- Simulation, Verification and Validation options: Software-in-the-Loop (SiL) and Processor-in-the-Loop (PiL)

From Concept to Embedded Application

We will then showcase the entire process of developing an application covering all the development phases. Starting from a software model, the entire logic of the example will be implemented in a visual manner, using block diagrams. Then, MCU peripherals will be configured and initialized (according to the application design).

Since verification and validation are key procedures to achieve quality results, the tutorial will focus on how they can be employed throughout all the phases of the software development workflow, by presenting the Software-in-the-Loop (SiL) and Processor-in-the-Loop (PiL) techniques.

We will conclude with the executable software image being downloaded onto the target hardware. Moreover, we will provide a visual manner of debugging the application by monitoring signals in real time on the embedded target, tracing signals and tuning parameters.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 4.1: Test Driven Hardware Design and Verification

Time: 11:30 – 12:30

Stream 4

Moderator:

Bodo Hoppe is a Distinguished Engineer at IBM Research & Development GmbH. He is responsible for the overall design quality and functionality of the IBM z Systems microprocessor. He is leading the verification methodology along with the agile transformation enabling the user experience driving the hardware development from design thinking to feature based development.

Speakers:

Georg Gläser is a digital design and mixed-signal system engineer at IMMS GmbH (Institut für Mikroelektronik- und Mechatronik-Systeme gemeinnützige GmbH). He worked on several scientific and industrial chip designs. He has a background in mixed-signal modelling methodology in his scientific work and is one of the organizers of the edaBarCamp.

Tudor Timisecu is a verification engineer at Infineon Technologies, where he works on security controllers. He is the current lead developer of SVUnit, a unit testing framework for SystemVerilog, and a proponent of test-driven design in the hardware space. He also writes about verification topics on his blog.

Matthew Ballance works as a functional verification Product Engineer at Siemens EDA, with a focus on stimulus generation techniques and the Accellera Portable Test and Stimulus language. He has previously worked in the areas of hardware/software co-verification, virtual prototyping, and IP encapsulation.

Philipp Wagner is full-stack hardware/software engineer with a keen interest in development methodologies and free and open source. During the day, he works at lowRISC on OpenTitan. Philipp is director at the FOSSi Foundation and maintainer of the cocotb verification framework.

Holger Horbach is a hardware verification engineer at IBM Research & Development GmbH. With more than 15 years of verification experience he has been exposed to many verification frameworks and methodologies -he is always curious and striving to actively shape the future of verification.

Abstract: We come from a history where a design is being developed and specified and then thrown over the fence to verify it. Verification was seen as a completely independent team. The standard method is random biased, the only metric is coverage, and it results in the wellknown design/verification gap.

New features or engineering change orders challenge this process since they may break the originally intended behavior. In the worst case, this new issue is unveiled by a full regression run. We think, it is time to change this game!

You know the feeling and the pain: You want to bring a new and essential feature online. When is it done? When do you reach full coverage? When do you exercise the new feature with all the other features? If you are lucky, the coverage is sufficient at the first run and you are done. In more usual case, you are frustrated at the end of a sprint that the new feature is still not in full regression. During the following process, you might uncover holes in the specification, window conditions or complex interactions with other features causing unexpected side effects.

Don’t get this wrong – highest quality is needed before the data is sent to build the chip. There is no second chance to first-time right silicon, no real post-production bugfixing possibilities. That’s why we are putting the effort into verification – and why we are seeking to improve and intensify this process.

Test Driven Development is a concept from the domain of software development. For achieving full functional coverage, each entity is written in the following way: First, a test for the required behavior is created. Next, the code is adapted to realize this behavior. Hence, all entities are tested itself before any integration (which is done in a similar manner) is done.

In Hardware design this would mean a logic designer and a verification engineer get to a common understanding and document it in a simple test. Then the design engineer starts the implementation while the verification engineer can develop the according verification components such as test generators, drivers and checks. Therefore, they share the responsibility of coverage in their own tests and test flavors.

These tests can serve as both specification and as acceptance criteria that a feature can be called done. The results of the test are linked to the feature as a proof-of-work and a full random biased verification with advanced coverage follows as a second step.

This demands the logic designer and the verification engineer to speak the same language. Also, the creation and execution of tests must be simple to keep frustration about tools and scripts away from engineers – but to help them create and encourage a good design. Hence, a careful design of the verification environment is required to grow it from initial tests to fully random bias verification along with the link of the test results to the overall verification process.

What are your experiences? We want to hear the good the bad and the ugly and discuss our future design and verification practice.

Instead of a traditional panel, we suggest a user experience fishbowl conversation based on liberating structures. We invite 3-5 verification leaders and leave 2-3 empty spots for the participants into the session. The remaining audience of the session can ask questions. They can put their questions in the chat – the moderator will ask them to formulate it directly. It is a conversation in the fishbowl not a presentation. And the participants can enter or leave the fishbowl during the conversation so that the conversation members change.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

12:30 – 13:30

Virtual Lunch Break

13:30 – 14:30

Panel: Anatomy of a Verification Flow

Organizer: Nanette Collins, Nanette V. Collins Marketing & PR

Moderator: Jean-Marie Brunet, Siemens EDA

Panelist:

Alex Starr, AMD
Daniel Schostak, Arm
Ty Garibay, Mythic AI
Nasr Ullah, SiFive

Abstract: Large companies, scale-ups and startups are eyeing new vertical market applications. Non-traditional semiconductor companies are entering the field as well with new business models and approaches. All are assessing markets as diverse as IoT, AI, 56/6G, HPC and healthcare, among others.

Timing is everything. Getting a chip designed, verified and ready for production requires an exceptionally talented group of knowledgeable and experienced experts. Possibly, the most important element for success is the verification flow. For a large company shifting into a fast-moving market, that could mean a major retooling effort. A startup may have more flexibility in planning its verification strategy, though it may be constrained by budget.

Join moderator Jean-Marie Brunet and panelists from Europe and the U.S. for a discussion on retooling a verification flow or designing a complete verification flow to meet the "must haves" of new market segments. They will address the types of tools and whether a standard "one size fits all" flow works in today's environment and how to budget accordingly. The discussion will include how to implement AI and ML for better verification.

14:30 – 15:30

Keynote: Speed Layers: Managing the exponential change of technology

Dr. Petra Farm
Product Thinker and Talk show host at Tolpagorni Product Management AB

Abstract: Late 19th, early 20th century city people of the world felt overwhelmed by the rapidity of change. The technology was evolving at an exponential pace and suddenly there were cars instead of horses on the streets and evening news as well as daily news doubling the rate of required information intake. Today in the 21st century there are no signs of slow down, if anything technological change is accelerating with an unprecedent speed and news from all over the world is just once push notice away. Industries are created and destroyed. Disruption is happening every day. Who remembers Blockbuster, Kodak, the state-run postal service, when you had to go to the bank to pay your bills? The complexity of the ecosystems all technology is existing in is increasing, the competition is fiercer and if you don't innovate you will be out. Yesterday it was cloud, big data, 3D printing, block chain today it is IoT, deep learning, edge computing – what will it be tomorrow? In short, we need to build more complex technologies, with more interdependencies faster. The Agile revolution is partly trying to address the challenge, but it is not enough. We need new perspectives on time and speed. Perspectives enabling short term monetization while pursuing investments in new core technologies for long term success. Enters Speed Layers, a conceptual product management framework that will allow for your product strategy, design, development and verification to operate in different speeds. All things needed to build sustainable and loved products like validation and verification, sales projects, market entries, development of new technology etc. can be placed in different speed layers. Each speed layer shall receive its unique priorities, ways of working and decision-making process. We want to have speed in execution on sales activities, in prototyping and testing and catching up with competition, but a slower more thoughtful strategic evaluation on implementing a new technology platform or entering/creating a new ecosystem. The definition of quality, financing and decision-making process for each layer will be different. The concept of Speed Layers has been successfully implemented at a number of tech companies. Lessons-learned and keys to success will be shared, but the journey will be all yours.

Biography: Petra comes from a tech background, with a Ph.D in Electronic System Design, joint research publication with Cadence Berkley Labs and formative years in the automotive industry. Eventually she started working herself up the software stack in a variety of industries such as pharmaceutical, telecom and Cloud tech. All the time remaining close to software architecture and development. When a product management position opened up at Ericsson, she seized it and has been in product management ever since. Today a thought leader in product management and one of two hosts at the popular ProductBeats' show with hundreds of attendees every Tuesday.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

15:30 – 16:00

Virtual Coffee Break
Meet us in the Virtual Experience Room

16:00 – 18:00

Tutorial Sessions

Tutorial 1.3: Deep Cycle HW/SW Verification using High-Performance Prototyping Systems

Time: 16:00 – 17:00

Stream 1

Sponsored by: SYNOPSYS® Silicon to Software™

Speaker: Andy Jolley: Principal Application Engineer Synopsys

Andy has been working with FPGA technologies for over 32 years, originally in a design capacity in the telecommunications, radar and video industries before supporting FPGA synthesis and prototyping technologies at Synplicity and then Synopsys.

Most recently, Andy has been supporting UK customers with their complex CPU & GPU IP and System Level prototyping needs on the Synopsys HAPS platforms while also providing support for worldwide engagements that deploy the same IPs and SoCs embedded into user applications.

Andy holds a 1st Class Bachelor's Degree in Electronic Engineering from the University of Brighton, England.

Abstract: Industry experts describe the amount of verification needed for processor-based system with the term 'deep cycles'. Only a high-performance prototype farm can deliver the deep cycles needed to run meaningful software workloads and system validation regressions.

In this tutorial we will showcase best-in-class prototyping methodologies harvesting all the capabilities of a state-of-the-art, high-performance prototyping system. We will show how to get full insight into the hardware running at prototyping speeds and efficiently run regressions leveraging a full range of visibility technologies and techniques including SystemVerilog assertions for an Arm-based system. We will also connect the prototype with Arm software debuggers and process data capture for full visibility of the software execution. We explain how to accelerate interface subsystems validation through the use of real-world interfaces and pre-built interface prototyping kits. Finally, we will explain how prototyping teams can enable their end users through a centralized deployment of prototyping systems using cloud-ready resource management systems. Throughout the tutorial we use examples based on the HAPS-100 prototyping system.

Tutorial 2.3: AI/ML Accelerator Verification Tutorial: High-Level Verification of C-level Design

Time: 16:00 – 17:00

Stream 2

Sponsored by: SIEMENS

Speakers:

David Aerne is a Verification Technologist within the Calypto Systems Division, focusing on HLV (High-Level Verification) solutions. His particular areas of expertise are the UVM and Verification IP. Prior to joining the EDA industry, he gained over 18 years of SoC Design and Verification experience in various roles at semiconductor companies and fabless startups. Dave received a BSCompE from the University of Illinois at Urbana-Champaign and a MSCompE from National Technological University in Fort Collins, Colorado.

Jonathan Craft is an HLS Technologist at Siemens EDA focused on development of High Level Verification (HLV) tools & methodologies. He holds a Bachelor of Science degree in Electrical Engineering from the University of Wyoming. Prior to working for Siemens, Jon held various design and verification roles performing IC and SoC development at various companies in the Denver, Colorado area.

Abstract: Introduction - One of the fastest growing areas of hardware and software design is Artificial Intelligence/Machine Learning (AI/ML), fueled by the demand for more autonomous systems such as computer vision (CV) for self-driving vehicles, voice recognition for personal assistants and many others. Many of these algorithms rely on Convolutional Neural Networks (CNNs) to implement deep learning systems. While the concept of convolution is relatively straightforward, the application of CNNs to the ML domain has yielded dozens of different neural network architectures. While these networks can be executed in software on CPUs/GPUs, the energy consumption of these software-based solutions make them impractical for most inferencing applications, the majority of which involve portable, low-power, edge computing devices.

Thus, the emergence of customized AI/ML hardware accelerators to meet the numerous, stringent, and potentially conflicting requirements. High-Level Synthesis (HLS) can provide the needed flexibility and abstraction to efficiently and quickly realize these designs in RTL. However, when working with HLS at the C-level, many have questions about what does verification look like? Waiting to verify until you have post-HLS RTL is too late and too inefficient. This workshop demonstrates how one can achieve comprehensive verification faster at a higher level of abstraction but still apply known and trusted RTL verification techniques.

Summary - Many newcomers to HLS have questions about how to take advantage of the dramatic productivity benefits of raising the design and verification abstraction but still have the confidence that they have with their verification current methodology.

This technical workshop, intended for design and/or verification teams, will demonstrate how a High-Level Verification (HLV) flow built around HLS and C-level design can dramatically speed up verification compared to a traditional RTL based flow. It will use an AI/ML hardware accelerator design example, written in C++ and using the open-source MatchLib SystemC library originally developed by NVIDIA, to step through the verification methodology and tools. This example will be provided as open source at the conclusion of the workshop.

The workshop will be a verification case study of the AI/ML accelerator design in an AMBA AXI4 subsystem. It will demonstrate how the pre-HLS simulation using MatchLib can identify and fix potential system-level performance issues that are normally not found until very late in a hand-coded RTL design methodology. Comprehensive verification on the pre-HLS design will be demonstrated highlighting both tools and flows(coverage, assertions, formal techniques, etc) gleaned from known and trusted RTL verification methodologies, as well as how the same environment can be reused post-HLS to quickly close coverage on the resulting RTL.

Tutorial 3.3: Automated Code Checks to Accelerate Top-Level Design Verification

Time: 16:00 – 17:00

Stream 3

Sponsored by: onespin assuring IC integrity

Speaker:

Nicolae Tusinschi is Product Specialist Design Verification at OneSpin: A Siemens Business. He joined the team in 2016 as a QA engineer and developed an exhaustive knowledge of OneSpin's formal verification tools before targeting his focus on the company's OneSpin 360 DV™ solution. Nicolae holds a BS Industrial Automation from "Dunărea de Jos" University of Galați (Romania) and a European Master's in Embedded Computing Systems (EMECS) from Technische Universität Kaiserslautern (Germany) and University of Southampton (United Kingdom).

Abstract: Integrated circuit designers are under constant pressure to deliver bug free code that meets evermore rigorous requirements. It is well known that the more bugs that can be detected early in the development process, the faster and easier that development effort will be. However, early bug detection requires a verification overhead on the designer that can be onerous and impact the design process.

The two major methods that designers can leverage to detect bugs are static linting and simulation. Linting requires low set up and can detect a class of bugs based on the syntax of the code, although does tend to report many potential issues that have to be analyzed and is limited when examining the sequential operation of a block. Simulation is focused on the code operation but requires a greater degree of set up, in the form of directed stimulus creation, which are usually not available at this stage in the process and will only detect issues in scenarios that the provided stimulus is examining.

What designers require for early and automated detection of implementation issues are fast and easy ways to set up static checks for the sequential operation of the code in an exhaustive fashion, without relying on user provided stimulus. Automated formal code inspection helps to rapidly eliminate errors in a piece of RTL, prior to functional verification and synthesis, while providing a fully automated, and simple use-model. Three different verification perspectives are achieved.

- Structural Analysis: Focused syntactic and semantic analysis of source code.
- Safety Checks: Exhaustively verify the absence of common sequential design operation issues and failure debugging.
- Activation Checks: Ensures that specific design functions can be executed and are not blocked by unreachability.

Each of these pieces of technology are fully automatic and require no assertions to be created by the user. There is no need to write stimulus, create assertions or understand the formal mechanisms being employed.

This tutorial will dive into how this technique can be applied. We'll explore real-world case studies that prove the effectiveness of this technology.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 1.4: An Update on the UVM-AMS Standard in Accellera

Time: 17:00 – 18:00

Stream 1

Speakers: Tom Fitzpatrick, Siemens EDA

Abstract: Based on various discussions with user companies, it was established that there was a strong interest in the standardization of UVM-AMS. An Accellera Proposed Working Group (PWG) and then a Working Group (WG) were formed, based on Accellera Board of Directors' recommendation, with a charter to develop a standard that will provide a unified analog/mixed-signal verification methodology based on UVM, with major focus on transient analysis.

The UVM-AMS standard will provide a comprehensive and unified analog/mixed-signal verification methodology based on UVM to improve analog mixed signal (AMS) and digital mixed signal (DMS) verification of integrated circuits and systems. This will encourage support by tool and IP providers, offering ready-to-use analog/mixed-signal verification IP that can be integrated easily into a UVM-AMS testbench. It will raise the productivity and quality of analog/mixed-signal verification across projects and applications, thanks to the reuse of proven verification components, and stimuli.

In this workshop, the WG would share the findings, requirements and ideas collected so far and the next step plan for the standardization and would like to receive feedback from the analog/mixed-signal verification community. The UVM-AMS tutorial will also share the latest standardization and technology developments as (being) published in the Accellera UVM-AMS whitepaper.

The following main aspects of the UVM-AMS standard under consideration will be discussed at high level in this Workshop.

1. A UVM-ASM framework for the creation of analog/mixed-signal verification components and test benches by introducing both extensions to digital centric UVM verification IP classes and also related module-based components to facilitate interactions between the class-based and structural environments.
2. A set of class-based extensions to UVM related to driver, monitor, scoreboard, etc., to support analog/mixed-signal verification
3. A set of components and/or packages in SystemVerilog and/or Verilog-AMS to facilitate interactions between the class-based and structural environments and to interface with various types of Analog Design Representations.
4. A set of Application Programming Interfaces (APIs) to enable the development of modular, scalable, and reusable verification components and test benches, including stimulus, sequence and analysis functions, etc.
5. A framework for creation of Mixed Signal Verification UVM verification components (UVCs) or extensions of existing UVCs for enhanced stimulus, analysis, monitoring and debug capabilities.

In addition, the requirements collected so far for the following elements will be presented and discussed as well.

1. Driver,
2. Monitor,
3. Sequencer,
4. Checker,
5. Types of Data and Signal Abstractions (electrical, real, …),
6. Randomization,
7. Different Types of Design Representation (SPICE Netlist, Verilog-AMS, Real Number Modeling)
8. Analog Signal Characteristics and Its Generation (RF style, Noise, Jitter, …)
9. Coverage,
10. Assertion and Assumptions.
11. Basic Libraries and/or packages of tunable checker classes, assertions, transfer functions, time-to-frequency domain transformations, and so on.

Finally, an example will be illustrated about how a UVM_AMS VIP (or UVC) Package looks like, and how the user would be able to integrate and use it in a smooth and efficient way.

At the end, we would like to hear from the audience with comments, questions, and suggestions.

Time Zone is CEST

Tutorial 2.4: Boost Your Productivity in FPGA & ASIC Design and Verification

Time: 17:00-18:00

Stream 2

Sponsored by: Sigasi;

Speakers: Bart Brosens

Kids need a passion in life and Bart's passion was always electr(on)ic devices. His talent in hardware design has transformed into a career in companies like Target Compiler Technologies (now Synopsys), Barco (now Silex Insight), Easics and now Sigasi. At Sigasi, Bart was first a customer and is now our senior Application Engineer supporting customers from Korea to California, and most countries in between. When he's not talking to Sigasi's clients, Bart loves to sing in a choir, take long mountain hikes with his family or engage himself in local social activities in his neighbourhood.

Abstract: Most EDA tools take your HDL code as a starting point and process it as efficiently as possible, focusing on the silicon. Those tools do not care about how you come up with this code in the first place.

If you want to be more productive as a hardware design or verification engineer working on VHDL, Verilog and SystemVerilog code, you deserve a software tool that helps you to create and explore code in a more efficient way. Lessons learned from designing complex software systems also apply to designing complex hardware systems.

In this tutorial, Bart Brosens (Application Engineer at Sigasi) will demonstrate how to increase your productivity using an IDE for your HDL design. You'll learn our best practices from proven methodologies such as:

- type-time feedback,
- intelligent content assist,
- various ways to explore and navigate through large projects,
- how to effortlessly document your design
- and much more.

This year, the tutorial will use Sigasi Studio and the new Sigasi extension for VS Code as examples of IDEs for HDL design.

Technical Program: Tuesday, October 26 (cont.)

Time Zone is CEST

Tutorial 3.4: Collaborative, Advanced Fault Analysis: Addressing the Functional Safety Verification Challenges from the Accellera Functional Safety

Time: 17:00 – 18:00

Stream 3

Sponsored by: OPTIMA

Speaker: Shesha Sai Kumar, Director of Applications, Optima Design Automation

Sesha Sai Kumar C.V. has 25+ years of experience in the complete VLSI eco-system ranging from concept to final silicon. Starting his career as a Scientist in India's defense R&D organization, he went on to work for Synopsys, LSI Logic, and ArchPro (acquired by Synopsys). He has extensive Applications Engineering experience in the fields of functional safety, low power VLSI verification, clock domain crossing (CDC) analysis, static analysis and linting. Sesha co-founded SkandVLSI, introducing a learn-practice-teach platform called VGuru Verilog/VHDL with interactive guidance for university undergraduates. He holds a BTech in Electronics and Communications Engineering from SV University, India.

Abstract: Fault injection simulation is necessary to quantitatively assess a safety critical design to comply with various ASIL levels, as described in the ISO 26262 standard. A passing functional simulation testbench, called a workload, is considered for the Design Under Safety Assessment (DUSA) to perform fault injection and simulation. One of the challenges with fault injection and fault simulation is that as the design grows, the number of faults increase drastically and, hence, the fault simulation time extends sometimes to weeks or months.

Most of the methodologies inject faults in the full hierarchy of the DUSA, perform the fault simulation, and classify the faults as DETECTED (output on observation signals that differ from the good machine) or UNDETECTED (the output of the fault machine and the good machine is the same after the simulation).

These methodologies end up reporting large numbers of UNDETECTED faults, and propose the use of multiple tools, for example static, formal and simulation. This raises questions on the exchange of data related to functional safety between multiple tools to minimize debug and accelerate ISO 26262 metric analysis. In this tutorial we discuss how practical tool flows might make use of the Accellera Functional Safety Working Group proposals to drive integrated methodologies that achieve ISO 26262 metric analysis more quickly.

Specifically, the recently introduced Accellera Functional Safety White Paper mentions some of the below challenges. We present in this tutorial how these may be addressed using available tools today in commercial methodologies:

1. Multiple tool setup and data exchange: This is a challenge as the suggested methodology consists of different static, formal and fault simulation tools. One source of confusion is which tool to use first. For example, is formal analysis employed to reduce the full design fault list prior to fault simulation or vice versa? Or are Design for Test (DfT) tools leveraged initially, followed by formal or simulation. All these tools have different setups, different runtime commands and different mechanisms to treat faults. Results are obviously based on the tool core functionality and are not related to fault analysis or the ISO 26262 standard. We propose to address these challenges by leveraging a single platform where setup is easy and consistent. Consistent design and fault data is maintained across the core engines. Leveraging simple commands, a user can follow a suggested methodology.

2. Intuitive Fault Classification: Faults are classified based on Observability and Detectability. This kind of classification will not provide much information if the fault is not detectable. Also the fault status for Functional Safety is generally defined by the user, which can very inconsistent across organizations. We suggest a more intuitive way of classifying faults at various analysis stages, allowing all the engineers working on the design have the same understanding of the classification. This also extends to other organizations supplied with automotive IP from the methodology. Fault classification is maintained throughout different phases of the analysis.
3. Early and accurate feedback to FMEDA: The earlier a user can start building up FMEDA information, and the ease with which design changes and failure modes can be applied to the FMEDA is very important in the ISO 26262 development cycle. We will demonstrate how this early feedback and accurate information may be used to calculate FiT rates. We will also demonstrate how the Optima platform interfaces with the Ansys' Medini product for FMEDA, showing how the Accellera FS white paper suggestions may be implemented today.

We also consider other challenges mentioned in the Accellera FS white paper and how these may be addressed leveraging advanced core verification engines.

18:00 – 18:30

Closure Day 1 & Outlook Day 2

Technical Program: Wednesday, October 27

Time Zone is CEST

8:00 – 8:15

Opening & Welcome

8:00 – 18:30

Virtual Experience Room & Networking

8:15 – 9:15

Keynote: Take a leap: virtualization in future development

Satish Sundaresan
Elektrobit India

Abstract: The automotive industry is facing various challenges, from complexities of bringing software-defined vehicles on the road to increasing technical capabilities for autonomous vehicles. Due to the processing power needed to run software-defined functions, software creation goes beyond traditional methods and often, extensively uses artificial intelligence (AI) and evolving sensor technologies. Additionally, serverbased vehicle network architecture and service-oriented base software might cause a shift of sourcing strategy to software only. This is where virtualization becomes part of the equation and leads towards a possible new vehicle lifecycle model that challenges our current understanding across car makers, suppliers, and their business models. To gain acceptance from drivers and passengers and the ever-stringent authorities, we must demonstrate all this new technology is reliable and safe. How do we decide the completed testing and validation is enough to prove our vehicles are safe and ready for mass production? What is the best balance for quality, cost, and time for a vehicle launch? Virtualization may become mandatory in future architectures to redesign the vehicle electronics networks with virtual resources ranging from processors to operating systems to data servers. Virtualization must be managed carefully to ensure the benefits outweigh the additional risks for vehicle development. There are many aspects of virtualization that need to be considered from para to full virtualization or critical to non-critical software aspect separation. It's time for the automotive industry to challenge its value and embrace virtualization in some form.

Biography: Satish Sundaresan heads Elektrobit India's subsidiary based in Bengaluru. The India centre is a R&D location for Elektrobit and addresses local sales needs.

Satish also heads a global product development team in the Validation space for Advanced Driver Assistance.

Prior to Elektrobit, Satish managed large global programs, whilst leading India based R&D centre's across automotive electronics and IT operations. Satish's experience as a business leader with outcome focused roles spans across car manufacturers, tier 1 suppliers, and services organizations with exposure in Germany, the USA, Australia, and Malaysia. During his career he led multi-cultural and geographically dispersed teams to meet organizational, technical, and market objectives.

Technical Program: Wednesday, October 27 (cont.)

Time Zone is CEST

9:15 - 9:30

Virtual Coffee Break

Meet us in the Virtual Experience Room

9:30 - 11:00

Stream 1

P1.1 Machine Learning for Coverage Analysis in Design Verification

Jayasree Venkatesh

Qualcomm India Private Limited

P1.2 SimPy and Chips: A Discrete Event Simulation Framework in Python for Large Scale Architectural Modelling of Machine Intelligence Accelerators

Daniel Wilkinson[1]; Hachem Yassine[1]; Graham Cunningham; Iason Myttas[1]

[1] Graphcore Ltd

P1.3 Optimizing Design Verification using Machine Learning: Doing better than Random

William Hughes, Maithilee Kulkarni, Sandeep Srinivasan, Rohit Suvarna

Verifai Inc

9:30 - 11:00

Stream 2

P2.1 Achieving Faster Code Coverage Closure using High-Level Synthesis

Surendhar Thudukuchi Chandrapandiyan; PhD Preetham Lakshmikanthan; Ashwani Aggarwal; Youngchan Lee; PhD Youngsik Kim; PhD Seonil Brian Choi

Samsung Electronics

P2.2 Detection of Glitch-Prone Clock and Reset Propagation with Automated Formal Analysis

Kaushal Shah; Sulabh kumar Khare

Siemens Digital Industries Software

P2.3 Using HLS to Improve Design-for-Verification of Multi-pipeline Designs with Resource Sharing

Sarmad Dahir; Nils Luetke-Steinhorst; PhD Christian Sauer

Cadence Design Systems

9:30 - 11:00

Stream 3

P3.1 Bringing Reset Domains and Power Domains Together – Set/Reset Flops Augmenting Complexities in Power-Aware RDC Verification

Manjunatha Srinivas[1]; Manish Bhati[1]; Abdul Moyeen[1]; Inayat Ali[2]

[1] Siemens Digital Industries Software; [2] NXP Semiconductors

P3.2 A Methodology for Evaluating SI Artefacts in DDR4-3DS PHY using Channel Modelling

Aditya S Kumar; Gowdra Bomanna Chethan; Shivani Maurya; Anil Deshpande; Somasunder Kattepura Sreenath

Samsung Semiconductor India R & D Centre(SSIR)

P3.3 Using Dependency Injection Design Pattern in Power Aware Tests

Mehmet Tukel[1]; Luca Sasselli; David Guthrie

[1] QUALCOMM Ireland

9:30 - 11:00

Stream 4

P4.1 Method for Early Performance Verification of Hardware-accelerated Embedded Processor Systems in RTL Simulation

Luca Sasselli[1]; Mehmet Tukel; David Guthrie

[1] Qualcomm

P4.2 System-Level Register Verification and Debug

Utkarsh Bhiogade[1]; Kautilya Joshi[1]; Puneet Goel[2]

[1] Indian Institute of Information Technology Nagpur; [2] Coverify Systems Technology

P4.3 No Country for Old Men – A Modern Take on Metrics Driven Verification

Svetlomir Hristozkov; PhD James Pallister; **Richard Porter**

Graphcore Ltd

Technical Program: Wednesday, October 27 (cont.)

Time Zone is CEST

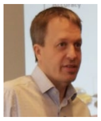
11:00 – 12:00

Panel: Can ML be the Driver of Next-Generation Verification?

Moderator: Karel Freund, Cambrian AI Research

Organizer: David Kelf

Panel Chair: Tran Nguyen, Arm Ltd.



Panelists: Tushit Jain, Machine Learning Research Qualcomm; **John Rose,** Cadence; **Ravi Gal,** IBM; **Daniel Hansson,** Verifyter; **Adnan Hamid,** Breker Verification Systems; **Darren Galpin,** Renesas

Abstract: Verification productivity has historically been largely dependent on performance tooling, and engineering ingenuity in driving these tools. The entire verification loop from test content composition, execution, debug and coverage management features engineers manually devising test programs and analyzing the results.

Verification continues to evolve with increasing functional requirements augmented by SoC integrity. 5G, Autonomous Driving, Quantum Computing and other applications drive verification complexity to manual engineering limits. Engineers need some help to contain the verification explosion driven by this expansion.

Machine Learning (ML) is proving itself a powerful weapon in the hands of engineers analyzing incredibly massive and difficult challenges. ML can be a vital aid to verification engineers struggling with these challenges under increasing schedule pressure and quality demands.

For verification, ML may be used to drive test content efficiency and reduce redundancy, predict potential bugs in big data regression output, select appropriate engines to tackle specific problems in formal and test synthesis tools, and a myriad of other tasks. However, some might argue that an overreliance on ML could reduce quality and result in missed issues.

Moderated by AI expert Karl Freund, this panel of experts, who have either developed or used verification technologies that leverage ML, will discuss and contrast the use and value of ML by examining real world applications in this area.

12:00 – 13:00

Virtual Lunch Break

Meet us in the Virtual Experience Room

12:00 – 13:00

Sponsor Sessions

AGNISYS
SYSTEM DEVELOPMENT WITH CERTAINTY

Jade
design automation

LUBIS EDA

SIEMENS

Veriest

Technical Program: Wednesday, October 27 (cont.)

Time Zone is CEST

13:00 – 14:00

Poster Session
Meet us in the Virtual Experience Room

P5.1 A Novel Approach to In-System SRAM Repair Verification in Embedded SoC to Bridge Gap between Test and Functional Mode

Harshal Kothari; Eldin Ben Jacob; Sriram Kazhiyur Sounderrajan; Somasunder Kattepura Sreenath
Samsung Semiconductor India R&D

P5.2 Advance Approach for Formal Verification of Configurable Pulse Width Modulation Controller

Sumit Kumar Kulshreshtha; Raghavendra JN
Intel Technology India Pvt Ltd

P5.3 An Analysis of Stimulus Techniques for Efficient Functional Coverage Closure

Caglayan Yalcin; Aileen McCabe
QUALCOMM

P5.4 Centralized Regression Optimisation Toolkit (CROT) for Expediting Regression Closure with Simulator Performance Optimisation

Harshal Kothari; Pavan M; Eldin Ben Jacob; Sriram Kazhiyur Sounderrajan; Somasunder Kattepura Sreenath
Samsung Semiconductor India R&D

P5.5 Chip-Level Analog Regression in Production

PhD Yi Wang
Dialog Semiconductor B.V.

P5.6 Emulation Based Power and Performance Workloads on ML NPUs

Pragati Mishra[1]; Ritu Suresh; Issac Zacharia; Jitendra Aggarwal
[1] Arm Ltd

P5.7 Five Ways to Make Your Specman Environment More Reusable and Configurable

Stefan Sljukic; Nikola Knezevic; Filip Dojcinovic
Veriest Solutions

P5.8 Formal Property Verification of the Digital Section of an Ultra-Low Current Digitizer ASIC

Katharina Ceesay-Seitz[1]; Sarath Kundumattathil Mohanan; PhD Hamza Boukabache; Daniel Perrin
[1] CERN

P5.9 Language-Agnostic Communication for SystemC/TLM-2.0 Compliant Virtual Prototypes

Smruti Khire; Kunal Sharma; Vishal Chovatiya
Infineon Technologies

P5.10 Maximize PSS Reuse with Unified Test Realization Layer Across Verification Environments

Simranjit Singh[1]; Arun K.R.; Ashwani Aggarwal; Suman Kumar Reddy Mekala; Woojoo Space Kim[2]; Seonil Brian Choi[2]; Gnaneshwara Tatuskar[3]
[1] Samsung Semiconductor India R&D; [2] Samsung Electronics; [3] Cadence Design Systems

P5.11 Resetting RDC Expectations - A Systematic Approach to Verifying Complex Configurable Designs

Eamonn Quigley[1]; Jonathan Niven[1]; Mark Handover[2]
[1] ARM; [2] Siemens

P5.12 Successive Refinement – An approach to Decouple Front-End and Back-end Power Intent

Rohit Sinha
Intel

P5.13 Virtual Prototyping of Power Converter Systems Based on AURIX™ using SystemC AMS

Radovan Vuletic[1]; Thomas Arndt[2]; Dineshkumar Selvaraj[1]
[1] Infineon Technologies AG; [2] COSEDA Technologies GmbH

14:00 - 15:00

Virtual Coffee Break
Meet us in the Virtual Experience Room

14:00 - 15:00

Sponsor Sessions

AMIQ EDA

Corigine

cadence®

Fraunhofer IIS

Technical Program: Wednesday, October 27 (cont.)

Time Zone is CEST

Stream 1

15:00 – 16:30

P1.4 A Novel Approach to Reuse Firmware for Verification of Controller Based Sub-Systems Using PSS

Vishnu Ramadas; Simranjit Singh[1]; Ashwani Aggarwal; Woojoo Space Kim[2]; Seonil Brian Choi[2]

[1] Samsung Semiconductor India R&D; [2] Samsung Electronics

P1.5 Reuse of System-Level Verification Components within Chip-Level UVM Environments

Diego Alagna[1]; Marzia Annovazzi[1]; Alessandro Cannone[1]; Marcello Raimondi[1]; Simone Saracino[1]; Mukesh Chugh[2]; Marc Erickson[2]; Cristian Macario[2]; Giuseppe Ridinò[2]

[1] STMicroelectronics; [2] MathWorks

P1.6 A Novel Approach to Functional Test Development and Execution using High-Speed IO

Marcus Schulze Westenhorst[1]; Jörg Simon[2]; Markus Bücker[1]; Klaus-Dieter Hilliges[1]; Michael Braun[1]

[1] Advantest Europe GmbH; [2] Cadence Design Systems

Stream 2

15:00 – 16:30

P2.4 Testbench Flexibility as a Foundation for Success

Ana Sanz Carretero[1]; Katherine Garden; WeiWei Cheon

[1] Xilinx

P2.5 Handling Asynchronous Reset(s) Testing by Building Reset-awareness into UVM Testbench Components

Wei Wei Cheong; Katherine Garden; Ana Sanz Carretero

Xilinx Inc.

P2.6 One Testbench to Rule Them All!

Salman Tanvir; Markus Brosch; Amer Siddiqi

Stream 3

15:00 – 16:30

P3.4 A Comparison of Methodologies to Simulate Mixed-signal IC

Simone Fontanesi[1]; Paul Ehrlich[2]; Karsten Einwich[2]; Gaetano Formato[1]; Andrea Possemato[1]

[1] Infineon Technologies Austria AG; [2] COSEDA Technologies GmbH

P3.5 Unified Model/Hardware-in-the-Loop Methodology for Mixed-Signal System Design and Hardware Prototyping

Martin Barnasconi[1]; Wil Kitzen[1]; Thieu Lammers[1]; Paul Ehrlich[2]; Karsten Einwich[2]

[1] NXP Semiconductors; [2] COSEDA Technologies GmbH

P3.6 Accelerated Coverage Closure by Utilizing Local Structure in the RTL Code

Gokce Sarar[1]; Guillaume Shippee[1]; Tushit Jain[1]; Rhys Buggy[2]; Vishal Karna[1]; Han Nuon[1]

[1] Qualcomm Technologies, Inc.; [2] QT Technologies Ireland Limited

Stream 4

15:00 – 16:30

P4.4 Machine Learning based Structure Recognition in Analog Schematics for Constraints Generation

Rituj Patel; PhD Husni Habal; Venkata Konda Reddy Rolla

P4.5 Democratizing Formal Verification

Tobias Ludwig

Lubis EDA

P4.6 Netlist Paths: A Tool for Front-end Netlist Analysis

PhD Jamie Hanlon; Samuel Kong

Graphcore Ltd

Technical Program: Wednesday, October 27 (cont.)

Time Zone is CEST

16:30 – 17:30

Keynote: The Mobility of the Future is Software defined – Can Open Technologies help?

Speaker: **Andreas Riexinger**

Bosch GmbH

Abstract: Our world is changing, and the change is visible everywhere. More than 50% of the population is living in cities and the cities are growing. More and more goods and people need to be transported, bringing the traffic infrastructure to its limits. Increasing pollution and noise levels bringing the environment to its limits.

To counter this, a transformation of the mobility is needed, who has already started, powered by new technologies and services. The mobility of the future will be electrified, connected, personalized, automated and software defined.

Software is more and more dominating our daily life, also in the automotive world. Almost every function in the vehicle is defined by software. Autonomous Driving solutions introduces a new complexity into the development of embedded systems in the vehicle. This complexity rises with each level of control and autonomy of the automated driving systems and you need deeper expertise and more software. Software is more and more becoming an important differentiator for autonomous vehicles.

Increasing embedded software complexity with an increase in level of automation is putting new demands on the existing tools, frameworks, software stacks and standards. Instead of solving these challenges alone, partnering in non-differentiating areas and collaborating is the path forward.

Biography: Since end of 2016, Andreas Riexinger is driving the Bosch autonomous driving solutions as a product manager automated driving for the Robert Bosch GmbH. He worked in different areas for the Robert Bosch GmbH for over 20 years, in which he has collected experience in the development of embedded software and management of IT projects.

Since the foundation of the OpenADx Eclipse Working Group in June 2019, he is also their speaker.

Kontakt:

Email: andreas.riexinger@de.bosch.com

Twitter: @ARiexi

LinkedIn: www.linkedin.com/in/andreasriexinger

OpenADx: openadx.eclipse.org

17:30 – 18:30

Closing Session & Best Paper Awards

SystemC Evolution Day

SYSTEMC™
EVOLUTION DAY
OCT 28, 2021 | VIRTUAL WORKSHOP

The sixth SystemC Evolution Day is a full-day, technical workshop on the evolution of SystemC standards to advance the SystemC ecosystem. In several in-depth sessions, selected current and future standardization topics around SystemC will be discussed in order to accelerate their progress for inclusion in Accellera/IEEE standards.

SystemC Evolution Day is intended as a lean, user-centric, hands-on forum bringing together experts from the SystemC user community and the Accellera Working Groups to advance SystemC standards.

Date / Time: October 28, 2021 (day after DVCon Europe 2021) | 08:30 - 18:30 CEST

Location: Virtual Workshop

Submissions / Questions: Email systemc-evolution-day@lists.accellera.org

Organization Team:

- **Ola Dahl,** Ericsson (Chair)
- **Martin Barnasconi,** NXP
- **Jerome Cornet,** STMicroelectronics
- **Christian Sauer,** Cadence
- **Mark Burton,** GreenSocs
- **Peter de Jager,** Intel

Schedule:

8:30-9:00	Welcome and Introduction **Ola Dahl,** Ericsson
9:00-10:00	How to fork threads in SystemC just like in SystemVerilog and Specman-e **Stefan Tiberiu Petre,** Independent Verification Consultant
10:00-11:00	Multi-core Debugger Integration and Suspend/Resume **Peter de Jager,** Intel Corporation
11:00-12:00	SystemC Community – GitHub, forums, web sites **Martin Barnasconi,** Accellera; **Mark Burton,** GreenSocs
12:00-12:30	Q&A **All**
13:00-16:00	Virtual Networking
16:00-17:00	SystemC in Hybrid Simulations **Mark Burton,** GreenSocs; **Martin Barnasconi,** Accellera
17:00-18:00	Achieving IC Integrity for SystemC Designs **Vlada Kalinic,** OneSpin Solutions
18:00-18:30	Summary and Concluding Discussion **Ola Dahl**, Ericsson

Save the Date!
DVCon U.S. 2022 (Virtual)
February 28 - March 3, 2022

DVCon Europe 2022
November 1-2, 2022
Munich Germany