Formal Verification Conference

United Kingdom (Reading)
Thursday, 16 June 2016

Sponsored by

Participating Companies
FORMAL Verification BOOTCAMP

Learn how to use formal verification techniques to improve verification

The 2-day Formal Verification Bootcamp is for design and verification engineers looking to enhance their knowledge of formal verification and to learn how to write effective assertions to find and fix bugs. The course is a mix of presentations and hands-on development exercises.

Key Topics Covered
+ Introducing System Verilog Assertions (SVA)
+ Advanced SVA
+ Debugging a Failing Property
+ Formal Metrics
+ Automated Formal Verification
+ Full Formal Verification of a Block
+ Formal in the Design Flow
+ Formal for Designers
+ Using Formal Apps

Course Presenter
Dr. Mike Bartley has worked in Formal Verification since completing his PhD in Mathematical Logic. He has worked in formal specification of both software (VDM, Z) and hardware (CSP and SVA), as well as formal mathematical proofs of implementations. Mike began working in property checking in 1994 and has worked continuously in both formal and dynamic simulation ever since. As well as writing and proving assertions, Mike has put together verification strategies and teams that combine both formal and simulation. Mike is a trained teacher and has written and delivered numerous training courses for Universities and commercial organisations. Mike uniquely combines theoretical knowledge, practical experience, management understanding and training expertise.

Additional Information
For detailed information on the course please contact us directly or visit our asureTRAIN website:

email: mike@testandverification.com
tel: +44 (0) 117 903 1100
asuretrain.testandverification.com
Agenda

09.25  Welcome and Introduction
       Mike Bartley, TVS (CEO and Founder)

09.30  Keynote – John Colley (Research Fellow)
       University of Southampton

10.10  Elchanan Rappaport (President)  Remote Presentation
       Gila Logic

10.40  Christian Burisch (Director Global Application Engineering)
       OneSpin Solutions

11.10  Refreshments and Networking

11.40  Holger Busch (Senior Staff Engineer)  Remote Presentation
       Infineon Technologies

12.10  Asa-Ben-Tzur (Formal R&D Software Engineering Group Director)
       Cadence

12.40  Lunch and Networking

13.50  Ashish Darbari (Principal Hardware Design Engineer)
       Imagination

14.20  Hans-Jörg Peter (Software Engineer, Senior Staff)
       Synopsys

14.50  Refreshments and Networking

15.10  Alex Orr (Master Engineer)
       Broadcom

15.50  Abdelouahab Ayari (Digital Design & Verification Solutions
       Applications  Engineer) - Mentor Graphics

16.20  Panel Session

16.45  Close
Floor Plan

Formal Verification Conference

Exhibition/Breakout Area
Mike Bartley
Test and Verification Solutions Ltd
CEO and Founder

Welcome Message

Biography:
Mike Bartley has a PhD in Mathematics from Bristol University, an MSc in Software Engineering and an MBA from the Open University, and over 25 years of experience in software testing and hardware verification. He has built and managed state-of-the-art test and verification teams in a number of companies (including STMicroelectronics, Infineon and Elixent/Panasonic) who still use the methodologies he established. Mike founded Test and Verification Solutions (TVS) in 2008 and has built the company to over 120 software test and hardware verification engineers working around the world. Since founding TVS Mike has continued his technical work and has consulted on multiple verification projects for respected organisations including ARM, NXP, Micron, Broadcom and Infineon.

Dr Bartley was formerly Chairman of the Bristol branch of the British Computer Society for ten years and is currently Chairman of the Bristol Local Enterprise Partnership (LEP). He sits on Technical committees for SNUG and DVCon in both Europe and India, had over 50 articles published on the subject of test, verification and outsourcing and regularly speaks at international conferences on these topics.

TVS would like to thank the sponsors and participants of the 2016 Formal Verification Conference
Notes
John Colley
University of Southampton
Research Fellow

Keynote Presentation
Model-based Safety and Security Analysis in High-Consequence System Development

Abstract

“Emergent behavior is that which cannot be predicted through analysis at any level simpler than that of the system as a whole. ...” (Dyson, George, Darwin Among the Machines, Addison-Wesley, 1997)

Safety and Security considerations for complex, high-consequence systems are increasingly becoming an expensive and time-consuming factor in system development. The cost of failure is high.

Safety and security are emergent properties of the system, their needs often conflict and it is not satisfactory to consider their requirements separately and in isolation from the overall requirements of the system. Trustworthiness is a measure of a system's ability to perform a wide range of roles in a variety of situations and considers not just safety and security, but also reliability, performance and resilience. Trustworthiness analysis must be conducted at the system level so that the impact of a software or digital hardware component malfunction on the system can be properly understood.

Civil aviation development and verification flows for software and digital hardware are mature and proven; the concepts underlying the requirements-based, coverage-driven DO-254 and DO-178C standards are widely adopted outside the aviation industry. These standards, however, are intended to be used with the safety analysis standard, ARP-4754A, which mandates that the safety considerations for software and digital hardware be analysed systematically from the system (aircraft and crew) perspective.

We present a method for Trustworthiness Analysis which is founded upon the civil aviation standards and leverages the model-based development and formal methods supplements to DO-178C. System trustworthiness constraints, derived from the system objectives, are modelled as formal properties at the abstract level and then refined, at the architectural level, to represent the trustworthiness constraints on the software or digital hardware component itself. Hazards and vulnerabilities are analysed systematically and the component constraints enforced to ensure that the hazards and vulnerabilities are eliminated or mitigated.
**Biography**

**Education**

2006-2010  PhD, Electronics and Computer Science, University of Southampton, UK.

1982–1985  BSc, First Class Honours, Computer Studies, University of Southampton, UK.

**Employment**

2010-present,  Research Fellow, Electronic and Software Systems Group, University of Southampton, UK.
Co-ordinator, ADVANCE(287563) FP7 project, Advanced Design and Verification Environment for Cyber-physical System Engineering.

Involved in the ongoing development of verification and validation methods for safety-critical systems, both formal and simulation based, for three UK Ministry of Defence suppliers.

Responsible for recruiting and leading a team of 12 engineers to develop a suite of verification tools for multi-core system-on-chip platforms.

As CTO, Conducted the research and led the development of property coverage tools for formal and simulation-based property checking of Verilog and VHDL designs.
As VP Engineering, worked on the IEEE standardisation of the PSL property specification language.
As Project Leader, led the development of state-machine coverage tools for simulation-based hardware verification of Verilog and VHDL designs.

Developed a language-neutral hardware simulation kernel to support VHDL, Verilog and C-based simulation.

**Publications**


Building on the DEPLOY legacy: code generation and simulation, Andrew Edmunds, John Colley and Michael Butler, DS-Event-B-2012: Workshop on the experience of and advances in developing dependable systems in Event-B.

Model-based Safety and Security Analysis in High Consequence System Development

John Colley
Formal Verification Conference
June 2016

“Humans are Slamming into Driverless Cars and Exposing a Key Flaw”
Bloomberg, Dec 2015

Introduction

• High consequence systems must be safe and secure
• but they still need to get the job done
  – Fail-safe is not always a viable option
• We are responsible for delivering software/digital hardware components. How do we assess their contribution to system safety and security?

“Emergent behavior is that which cannot be predicted through analysis at any level simpler than that of the system as a whole. …”
(Dyson, George, Darwin Among the Machines, Addison-Wesley, 1997)
“the engineering-driven actions necessary to develop more defensible and survivable systems”

“Security, like safety and other system quality properties, is an emergent property of a system.”

**Trustworthiness**

“Trustworthiness requirements can include, for example, attributes of safety, security, reliability, dependability, performance, resilience, and survivability under a wide range of potential adversity in the form of disruptions, hazards, and threats.”

“Effective measures of trustworthiness are meaningful only to the extent that the requirements are sufficiently complete and well-defined, and can be accurately assessed.”

*Where do we start?*
The Development and Certification Process

Development and Design Guidance
ARP-4754A, 2010

Guidelines and methods for Safety Assessment
ARP-4761, 1996

Software Considerations
DO-178C, 2011

Design Guidance Electronic Hardware
DO-254, 2000

DO-178B

- Software aspects of airborne systems and equipment
- Introduced in 1992
- Requirements-based Testing with Structural Coverage
  - Full, bi-directional Traceability from Requirements to Implementation
  - Modified Condition/Decision Coverage (MC/DC) sign-off for Level A software

DO-178C

- Finalised in December 2011
- Updates the successful 178-B standard
  - DO-331 Model-based Development and Verification Supplement
  - DO-333 Formal Methods Supplement
  - DO-332 Object-Oriented Technology and Related Techniques Supplement
  - DO-330 Software Tool Qualification Considerations
  - Modified Condition/Decision Coverage (MC/DC) definition clarification

DO-331

Model-based Development and Verification Supplement

- Formal Modelling Notation
  - Textual or
  - Graphical
  - Formal Semantics
- Supports a model-based methodology
  - Refinement of High-Level Requirements
  - Model-based Testing
  - Co-simulation
  - Code Generation
- Methodology already used successfully in Avionics
  - SCADE
DO-333
Formal Methods Supplement
• The outputs of a Formal Verification tool may be used to
  – Replace or
  – Augment
Testing
• Necessary to convince the Certification Authority of the validity of the Formal Verification
  – If not, must show full MC/DC coverage through testing

DO-178C Methodology for Design Assurance of Airborne Software

The Development and Certification Process

ARP-4761 Guidelines and Methods for Safety Assessment
• Strengths
  – Practical
  – Includes a detailed worked example
    • Aircraft Wheel Brake System
• Weaknesses
  – Out-of-date (1996)
  – Out-of-step with other relevant standards
    • ARP-4754A (2010)
    • DO-178C (2011)
ARP-4761A

• Updating the ARP-4761 standard
  – Currently in Progress
• Two major threads of the re-standardisation effort
  1. Model-based Safety Analysis
  2. System Theoretic Safety Analysis
     • A Comparison of STPA and the ARP 4761 Safety Assessment Process, Leveson et al, MIT, 2014

Model-based Safety Analysis

• System safety analysis techniques are well established, but
  – Highly subjective
  – Manual
  – Based on an informal System Model
• The Model-based Approach
  – System and Safety Engineers share a common system model
  – Model-based Development and Verification is already supported for DO-178C (DO-331 supplement)
  – Extending the system model with a Fault model enables automation of the safety analysis process
  – Allows Formal Reasoning
  

System-Theoretic Accident Model and Processes (STAMP)

• Systems Theory Foundation
• New Causality Model (c. 2010)
• Change in emphasis in system safety from
  – Preventing failures to
  – Enforcing Behavioural Safety Constraints
• Component Failure still Considered
• Safety is reformulated as a control problem rather than a reliability problem
• Three main concepts
  – Safety constraints
  – Hierarchical Control Structures
  – Process Models

Engineering a Safer World, Leveson MIT 2011
Safety Requirements

“Any controller – human or automated – needs a model of the process being controlled to control it effectively”

“Accidents can occur when the controller’s process model does not match the state of the system being controlled and the controller issues unsafe commands.”

Engineering a Safer World, Leveson, 2012

System Theoretic Process Analysis (STPA)

• New Hazard Analysis Technique
  – Identify potential causes of accidents, i.e. scenarios that can lead to losses
  – Eliminate or Control hazards in design or operations before damage occurs
• Augments existing methods (FTA, FMEA)
• Includes new causal factors identified in STAMP
  – Design errors including software flaws
  – Component interaction accidents
  – Human decision-making errors
  – Social, organisational and management factors contributing to accidents

Engineering a Safer World, Leveson MIT 2011

The Two Phases of STPA

1. Identify Potentially Hazardous Control Actions and derive the Safety Constraints
2. Determine how Unsafe Control Actions could occur

Engineering a Safer World, Leveson, 2012

Controlled Phenomena

For example, Landing Gear Doors
1. The Controller will open the Doors when the Pilot moves the Lever to Extend or Retract the Landing Gear
2. The Controller will then close the Doors when the Landing Gear is fully Extended or Retracted
3. The Doors will remain open while the Landing Gear is Extending or Retracting
**Step I: Identify Potentially Hazardous Control Actions and Derive Safety Constraints**

<table>
<thead>
<tr>
<th>Controller Action</th>
<th>Not Providing Causes Hazard</th>
<th>Providing Causes Hazard</th>
<th>Wrong Timing or Order Causes Hazard</th>
<th>Stopped too Soon/Applied too Long</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Door</td>
<td>Cannot extend Landing Gear</td>
<td>Not Hazardous</td>
<td>Not Hazardous</td>
<td>Damage to Landing Gear/Not Hazardous</td>
</tr>
<tr>
<td>Close Door</td>
<td>Not Hazardous</td>
<td>Damage to Landing Gear</td>
<td>Damage to Landing Gear</td>
<td>Not Hazardous/Not Hazardous</td>
</tr>
</tbody>
</table>

**Safety Constraints**
1. If the Landing Gear is Extending, the Door must be Locked Open
2. If the Landing Gear is Retracting, the Door must be Locked Open
3. A "Close Door" command must only be issued if the Landing Gear is Locked Up or Locked Down
4. An "Open Door" command must only be issued if the Landing Gear is Locked Up or Locked Down

---

**System Theoretic Early Concept Analysis (STECA)**

- **Begins at Concept Stage**
  - Requirements
    - Have multiple sources
    - Are evolving
  - No Specification
- **Model-based**
- **Starting point is Concept of Operations (Con-Ops)**

---

CONOPS

The Concept of Operations (CONOPS) is a user-oriented document that describes the characteristics for a proposed asset or system from the viewpoint of any individual or organizational entity that will use it in their daily work activities or who will operate or interact directly with it.

U.S. Coast Guard Press release, 2007

STPA-sec

1. Identifying the losses to be considered
2. Identifying system hazards or security vulnerabilities
3. Drawing the system functional control structure
4. Identifying unsafe or insecure control actions
5. Identifying intentional actions in the generation of causal scenarios

An Integrated Approach to Safety and Security* I

• Augments STPA to consider
  – External Threat
  – Inside Risks
• Added Causal Scenarios
  – Intentional Actions
• Hazards lead to safety incidents in the same way that vulnerabilities lead to security incidents
• Mission Assurance
  “the ability to complete a wide range of missions across a wide range of degradations”

Linton Wells, Former US Defense Dept Chief Information Officer, 2004-5


An Integrated Approach to Safety and Security* I

• Augments STPA to consider
  – External Threat
  – Inside Risks

Focus is on
Mission Completion
NOT Protecting the infrastructure used to complete the mission

• Mission Assurance
  “the ability to complete a wide range of missions across a wide range of degradations”

Linton Wells, Former US Defense Dept Chief Information Officer, 2004-5

An Integrated Approach to Safety and Security* II

• Top-down
  – Assuring the overall function of the enterprise
• Initial focus on controlling vulnerabilities rather than avoiding threats (strategic)
  – Likely to be far fewer vulnerabilities than threats
• Provides the context for subsequent threat analysis (tactical)
  – External and Internal
  – Know and unknown


An Integrated Approach to Safety and Security* III

“… the physical (or proximate) cause of a disruption does not really matter. What matters is the efficacy of the strategy in dealing with (controlling) the effects of that disruption on overall system function or assuring the mission.”


Model-based Trustworthiness Analysis

• Incorporates STPA, STECA, STPA-sec
  – Model-based
  – Early Concept
  – Architecture
• Augments the traditional Avionics Flow
  – Software/Digital Hardware Development
  – Verification/Validation and Acceptance
• Event-B, PSL, SVA Modelling and Formal Analysis

Case Study: Next Generation Transportation System (NextGen)

• Driven by the FAA
• 2012 – 2025 rollout
• NextGen Focus
  – More efficient flight paths
  – Shift in responsibility from ground-based crews to flight crews and flight-deck decision support tools
  – Shift of emphasis from clearance-based to trajectory-based operations

Trajectory-Based Operations (TBO) ConOps

Independent of the aircraft, the ANSP uses ADS-B position reporting for lateral and longitudinal progress, altitude reporting for vertical, and tools that measure the time progression for the flight track. Data link provides aircraft intent information. Combined, this position and timing information is then compared to a performance requirement for the airspace and the operation. ...precision needed...will vary based on the density of traffic and the nature of the operations. [JPDO, 2011]

Four Dimensional Trajectory (4DT)

- 3 dimensional space + time
- Represents not only current state of aircraft but also intent
- Specifies the tolerances in which the aircraft must operate
  - Required Navigation Performance (RNP)
  - Required Time Performance (RTP)
- Conformance
  - Monitoring the aircraft’s performance against the agreed 4DT

Conformance Monitoring

- Automated
- On the ground and in the air
- Generates alerts if an aircraft does not meet its agreed 4DT performance
- We consider security and reliability as well as safety
Hazard/Vulnerability Analysis

1. Aircraft violate minimum separation
2. Aircraft enters uncontrolled state
3. Aircraft performs controlled maneuver into ground

System-level Constraints

1. Aircraft must remain at least \( n \) nautical miles apart en-route
2. Aircraft position, velocity must remain within airframe manufacturer defined flight envelope
3. Aircraft must maintain positive clearance with all terrain (other than runways and taxiways)

Control Hierarchy

Generalised Controller Model
Composed Control Function

System-Theoretic Process Analysis (STPA)

1. Identify Potentially Hazardous Control Actions and derive the Safety Constraints
2. Determine how Unsafe Control Actions could occur

Potential unsafe control action for trajectory generation function:
- Approving a 4DT that will lead to LOS or not modifying a 4DT that will overlap

Potential unsafe control action for piloting function:
- Aircraft is following a 4DT that will lead to LOS

STPA Step 1

STPA Step 2

Potential unsafe control action for trajectory:
1. Error in ANSP algorithms (deliberate or otherwise)
2. 4DT is modified by Data Link (deliberate or otherwise)
3. Misleading or faulty display of 4DT to pilot (deliberate or otherwise)
4. Latency in communication (deliberate or otherwise)
Mitigating/controlling the Hazards/Vulnerabilities
Architectural Considerations

Add Timestamp, CRC, Encrypt

Decrypt, CRC, Check Timestamp

Safety/security/reliability Critical Modules

The Model-based Flow

• Driven by CONOPS
• System-level Constraints derived from System-level Hazard/Vulnerability Analysis
• System Control Hierarchy is defined
• Unsafe control actions are identified
• Hazards/Vulnerabilities are eliminated/mitigated
• Formal, Model-based
• Integrated within Software/Digital Hardware Development, Verification, Validation Flow
Notes
Elchanan Rappaport
Gila Logic
President

Formal Verification – Security Checking

Abstract
Secure architectures are becoming more complex. They route secure and non-secure data over the same channels, and share the same resources, but still require that there be no leakage from one domain to the other. Simulation provides no method[s?] for confirming that secure data is respected.

We present Formal methods and techniques, as well as actual user cases, which demonstrate how Formal successfully addresses this verification problem.

Biography
Elchanan Rappaport is the founder of Gila Logic, Inc. He has 35 years of experience in both simulation and formal-based verification, having worked with a number of semiconductor companies, including IBM, ARM, TI, National, Microsoft and others. For the last 15 years he has focused on high-end formal verification projects and is a common speaker at formal conferences and events. He may be contacted at elchanan@gilalogic.com.

At the request of the speaker the Slides are not available.
Christian Burisch
OneSpin Solutions
Director Global Application Engineering

Sponsor
Viewing Formal Through Simulation-Tinted Glasses

Abstract
Although Formal Verification is now actively used at many companies, it still has a reputation of being hard to use within mainstream verification environments. Often it is employed by formal specialists, and avoided by others, which is a shame as it offers a lot of benefits in this process. There are several ways in which formal can be aligned with simulation, to make it more natural and easy to use by engineers with a simulation and UVM background. This presentation will focus on operational assertions and observation coverage, mechanisms that provide a simulation feel to formal usage while still enabling the full power of the technology.

Biography
Christian Burisch is the Global Application Engineering Director for OneSpin Solutions. He was born in Germany, went to school in Kenya and to University in the UK. He started his engineering career at Fujitsu, before moving to Cadence as an AE, and then the first engineer for their new consulting business in the early 90’s. Christian has a 20-year track record working with EDA startups including Ambit (synthesis), Co-Design (SystemVerilog), Tenison (C++ IP generation), Methodics (IP management) and now OneSpin (formal verification). He also ran his own company in the software game industry. He holds a BSc in Computing Science from the University of Essex, UK, and an MSc in Microelectronics from Southampton University, UK.
OneSpin: Formal Technology Leadership

Unique, Powerful Verification Solutions
- Metric-Driven Verification
- Block Integration Validation
- FPGA Implementation Verification

Innovative Verification Solutions
- Agile Design Evaluation
- Safety Critical Verification
- SystemC/C++ Design Verification

Broad Range of Automated Apps
- Automated Inception
- Design Elaboration
- Essential ECL-RTL
- Assertion Verification
- Operational Acceptance
- Observation Coverage
- DV Apps
- xAve
- Connect
- Register
- Statement
- VIP
- Analysis
- SystemC/C++
- Sequential ECL-FPGA
- Fault Isolation
- Security Verification
- Specification

High Performance, Comprehensive Technology Platform

For the most comprehensive range of formal verification solutions available today, please visit

www.onespin.com

Proud sponsor of
TVS Formal Day 2016
The Potential of Formal-Sim

**Simulation and Formal Powerful Combination**
- The effect of combining techniques in the flow

**“Connection Points” to Improve the Flow**
- Looking at planning, coverage, assertions/stimulus

**OneSpin Pioneering Formal-Sim Integration**
- Latest work by OneSpin in this area

---

**Formal-Sim: Leveraging Orthogonality**
**Higher Coverage, Achieved Faster**

A design transitions between many states during operation.

- **Formal checks all possible states over set time to exhaustively test properties.**
- **Simulation checks one sequence of states driven by stimulus to replicate operation.**

---

**Simulation-Formal Combined Advantages**

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Formal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal Operation</td>
<td>Complex scenarios, recombination analysis</td>
</tr>
<tr>
<td>Data path</td>
<td>Control paths, data transport</td>
</tr>
<tr>
<td>High clock cycle depth</td>
<td>Shallow clock cycle depth, all states considered</td>
</tr>
<tr>
<td>Show correct behavior</td>
<td>Find unanticipated affects</td>
</tr>
</tbody>
</table>
Increasingly, Formal Layered Into Simulation Environments
• Formal being used where simulation falls short
  • Critical components that must be exhaustively verified
  • Corner cases with complex scenarios stimulus setup
  • Blocks with many operational permutations
  • Automated analysis tasks
• However, methodologies still very disconnected
  • OneSpin working on ideal methodology

Formal Users Today
Disconnected from simulation process

Tearing Down The Verification Wall
Recipe for streamlined verification methodology

Verification Planning
Consider Formal early in the process, not only at end
OneSpin Quantify™ Observation Coverage

Precise Coverage Closure

• Most precise formal coverage algorithm available
• Checking for observation as well as controllability
• Assertion development guidance towards closure
• Reduces redundant testing, shows uncovered areas
• Detects unreachable code
• Over-constrained, unreachable code
• Similar model to simulation code coverage

Observation Coverage more precise and efficient than other “Mutation” or “Cone-of-Influence”-based methods

Aligning Formal Coverage with Simulation

Common models recognizable to simulation users

• Assertions can be aligned with functional coverage from verification plan
• Need to check formal code coverage, aligned with simulation code coverage
• Need common coverage database to feedback results into planning tool

• Accellera UCIS

Available Today:
Test Generation From Formal Traces

• Research on methods to convert assertions to UVM tests
• Formal assertions hard to include simulation regression
• Useful to show verification team testing performed
• Valuable to feedback issues to designers for debug
• Formal testing re-run without formal tool in regression

Expert User Providing UVM Tests

Want assertions to be usable in regression

• Research on methods to convert assertions to UVM tests
• Formal assertions hard to include simulation regression
• Useful to show verification team testing performed
• Valuable to feedback issues to designers for debug
• Formal testing re-run without formal tool in regression
Abstract Assertions: More “Simulation-Like”
Align Operational Assertions with UVM Sequences

- Simplify assertion creation, closer to specification
- OneSpin SystemVerilog Library – industry standard
- One change required for its use with other tools
- Allow timing diagram style assertions
  - Reduce low-level detail, like UVM sequences

**Examples**

```verilog
property my_prop
    state == ! busy && transfer == 1 implies
    # (read_acc & write_acc) \[ \*2 \]
endproperty
```

---

OneSpin Formal-Sim

- High-performance/capacity via advanced platform
- Observation Coverage: precise closure metric
- Operational Assertions: abstract assertion authoring
- Integrated flow with simulation friendly use model
  New partnerships with Synopsys and Mentor

---

Powerful Solutions Require Strong Technology and Apps

Unique, Powerful Verification Solutions

- UVM
- SystemC/C++
- Specman
- Mentor
- High Performance, Comprehensive Technology Platform

---

Formal-Sim Summary

Need your feedback

**Simulation and Formal Powerful Combination**

- Orthogonal verification techniques
  - Reduces development time and reduces effort

**Several “Connection Points” Will Improve Flow**

- Verification planning including formal
  - Common coverage model
  - Common debug (by large solved issue)
    - Functional assertion authoring methods

**OneSpin Pioneering Formal-Sim Integration**

- Observation Coverage + UCDI integration
- Operational assertions and UVM connection
- Simulation integration models

For more information please visit

www.onespin.com
Holger Busch
Infineon Technologies
Senior Staff Engineer

Verifying Safety-Specification Compliance of RTL Designs

Abstract
Incorporating hardware safety measures in designs inevitably causes overhead in terms of area, propagation delays, and power consumption. The identification of safety-critical elements of hardware architectures therefore follows general principles, and needs to take into account results of various analyses at concept and design level in order to keep the undesired side-effects of the extra safeguarding logic under control. Configurable hardware safety mechanisms allow the trade-offs between increased safety and hardware optimization to be explored and adjusted.

This presentation discusses an automated formal safety verification flow for not only ensuring the effectiveness of the actually installed safety measures in instantiated designs, but also providing evidence that hardware elements are actually covered and safeguarded as specified, as needed for safety certification according to ISO26262.

Biography
Holger Busch holds a diploma degree in Electrical Engineering from the Technical University of Aachen, and a PhD from Brunel University of West London. During his professional life he has been working for different companies such as Mannesmann AG, Siemens AG – Central Research, and Infineon Technologies AG, doing research and development, application projects, project management, concept, design, software, and verification engineering in many different fields like operations research, process automation, power distribution, electronic design automation, automotive microcontrollers, formal methods, and digital hardware design. He is a certified ISO 26262 Automotive Functional Safety Professional.

Slides are available on the T&VS [website](#)
Notes
Asa Ben-Tzur
Cadence
Formal R&D Software Engineering Group Director
Sponsor
“Deep State Space Bug Hunting”

Biography
Asa got his engineering degree in computer science and electrical engineering from Technion, Haifa in 1995. Asa worked 14 years at Intel during which he was involved in post-Si verification, managed the formal verification framework and debug team and established the architectural and system level test generation group. After leaving Intel Asa managed the R&D for Bioness developing medical applications for functional neuro-stimulation. Asa Joined Jasper in 2010 and is now a software engineering group director responsible for JasperGold formal engines, BPS app and JasperGold expert system.

Slides are available on the T&VS website
Notes
Dr. Ashish Darbari
Imagination
Principal Hardware Design Engineer

Using Abstractions for Effective Formal: Lessons from the Trenches

Abstract
It has been nearly 40 years since P.Cousot and R.Cousot wrote their classic paper on Abstract Interpretations and yet abstraction is not yet used to its full potential in solving some of the hardest verification problems facing practitioners in the industry. In this talk, we will present some insight in how we have used abstractions in verifying a range of different IP blocks within Imagination. One particular abstraction we have deployed has been used to formally verify over a dozen different type of designs of varying complexity enabling an efficient bug hunting tactic. When abstractions are used in the way we did, one can also build exhaustive proofs of correctness of expected design behaviour where other forms of verification will be not only be tedious but also ineffective to generate a reasonable level of confidence often required for several specialised application domains such as functional safety and security. We will show how by using abstraction in this way one really does not need any other ‘APP’ and abstraction itself could be seen as a universal app

Biography
Dr Ashish Darbari works in Imagination Technologies where he leads the Advanced Verification Methodology (AVM) Group which he founded in 2013. Through verification consulting, hands-on project work, and grounds-up innovation in formal methods he and his team is helping multiple internal customers in different business units, across Imagination’s different sites world-wide. Dr Darbari has been working for 15 years in the field of formal and has extensive experience in a range of formal techniques including theorem proving and model checking where he has published 20+ papers and holds 5 patents. He has a Doctorate in Computer Science from Oxford and holds a Diplom Informatik from TU Dresden. Dr. Darbari is a Fellow of British Computing Society (UK), and IETE (India) and a senior member of IEEE and ACM (USA). Since March 2015, he is a “Royal Academy of Engineering Visiting Professor” at the Electronics and Computer Science department in the UK at the University of Southampton.

Slides are available on the T&VS website
Notes
Hans-Jörg Peter

Synopsys
Software Engineer, Senior Staff

Sponsor

Formal Verification Closure – Are your properties inadequate

Abstract
With growing popularity and necessity of formal verification in modern verification flows, a structured solution to measure the completeness of the formal verification environment is emerging as a critical requirement. While a formal proof on a design ensures absence of any assertion-violating behavior, an over-constrained design or inadequate assertions can give a false sense of correctness. This talk addresses this problem and presents new features in VC Formal™ that enable a methodology necessary for formal verification closure.

Biography
Dr. Hans-Joerg Peter joined Synopsys with the company’s merger with Atrenta in August 2015. In his current position, Dr. Peter is the technical lead for formal verification in Europe. Prior to joining Synopsys, he was Software Architect at Atrenta, where he was leading the development of the formal verification techniques for SpyGlass Constraints/TXV. Before that, Dr. Peter was working as a research and teaching assistant in the Computer Science Department at Saarland University, where he was also involved in the DFG-funded project AVACS. Prior to that, he was project and team leader at META-LEVEL Software.

Dr. Peter received the Dipl.-Inf. degree and the Dr. rer. nat. degree from Saarland University, Germany. He has 10+ years of experience as a researcher in formal methods and 20+ years of experience as a software engineer.

- At the request of the speaker the Slides are not available.
VC Formal
Next Generation Property Verification
Notes
Abstract
A year after sharing my first thoughts about formal verification how do I feel about the methodology - *spoiler alert* still an enthusiast - but finding its application in more and different places that I first expected. This presentation focuses on the day to day, pragmatic application of formal verification and its role in making our verification life better.

Biography
Over 20 years experience in semiconductor design and verification, worked for picoChip, Audium, Gnodal and now Broadcom. Is human, makes mistakes, considers admitting this the first step in becoming a better verification engineer!

Slides are available on the T&VS website
Notes
Abstract
What are formal property checking engines and how do they work? Why are they incredibly powerful for some properties, but not so good for others? What’s the state of the art and what's coming in the near future? This session will review the fundamentals as well as the recent breakthroughs that are driving advances in performance and capacity to instantly become a formal expert!

Biography
Abdelouahab Ayari, Ph.D. is an application engineer for formal verification, clock domain crossing, and low power verification. He received his doctor in formal verification at the University of Freiburg and worked for Micronas GmbH before joining Mentor Graphics. He has over 10 years’ experience on Assertion-Based Verification (ABV) and supporting major customers in the area of formal verification across Europe.
Questa®
Formal Verification Apps

Exhaustive, automated solutions for complex verification challenges.

For more information on the Questa Formal Verification Apps and Solutions, visit www.mentor.com/formal
Instant Formal Expert

Abdelouahab Ayari, Ph.D.
Formal Applications Engineer
EU Digital Design & Verification Solutions
TVS Formal Verification Seminar
June 16, 2016

Agenda
- Background
  - Model Checking
  - How Different from Simulation
  - Formal Friendly Design Pattern
- Technology
  - SAT-based BMC
  - IC3/PDR
- Application
  - Non-Determinism

The Promise of Formal Verification
- “Formal verification will 100%* verify your design”

Basic-Idea of Model Checking
- Synthesis generates a big FSM combining design and property logic

*This simple approach corresponds to explicit model checking used on the 80-ties. Today implementations differ.
Formal Verification Sweet Spots

- Model Checking Complexity:
  - Design and Property size
  - Run time: explore the complete state space
  - Space: produce "large" intermediate results

- "Good" Regular Structure → Small iteration number
  - Control intensive designs

- "Degenerated" Regular Structure → Large iteration number
  - Implicitly or explicitly counting
  - Complex data transformations
  - Long latency properties

Agenda

- Background
  - Model Checking
  - How Different from Simulation
  - Formal Friendly Design Pattern

- Technology
  - SAT-based BMC
  - IC3/PDR

- Application
  - Non-Determinism

Bounded Model Checking (BMC)

- Look for firings up to distance N
  - Don't keep track of reachable states
  - Simpler, but limits analysis to finding firings & bounded proofs

SAT-Based BMC

- Transform reachability question into a SAT query

- Transitions as Boolean Formulas
  \[ \Phi_i = \{ (\neg n_0 + n_1)(n_0 + n_1\neg n_0 + n_1)\} \]

- N-Path Formula
  \[ \Phi = (\Phi_0 \land \Phi_1 \land \ldots \land \Phi_n (\text{is_violated})) \]

- Satisfiability Problem (SAT-Problem)
  - Find an assignment \( x_0, \ldots, x_n \) such that \( \Phi \) evaluates to 1 (true)
BMC is a Counterexample Technology

- No Proofs
- Very effective at finding bugs
  - usually exhaustive to ~30 cycles
  - sometimes exhaustive to 100s of cycles
  - bugs have been found at 1000s of cycles
- Usage of BMC
  - New bring-up RTL (bugs are expected)
  - Bug hunting
  - Analysis up-to a given design depth is sufficient/acceptable

Agenda

- Background
  - Model Checking
  - How Different From Simulation
  - Formal Friendly Design Pattern
- Technology
  - SAT-based BMC
  - IC3/PDR
- Application
  - Non-Determinism

Induction

- Use high-school math to verify your design!
  - "Design": 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + ... + N
  - "Property": P(N) = N * (N+1) / 2
- Proof
  - Base case: P(1) = 1 * 2 / 2 = 1
  - Show that if P(k) holds, then P(k+1) also holds
  - P(k+1) = P(k) + (k+1)
    = k(k+1)/2 + (k+1)
    = (k+1)((k+1) + 1) / 2

Inductive Properties

- A property is inductive if:
  - It holds in the initial state, and
  - For any state in which it holds, it also holds in all next states
- Inductiveness easy to check with a SAT solver
  - Certainly a lot easier than running SMC or BMC
- Any provable property can be strengthened into an inductive property
  - i.e. for every provable property P(x), there is always a Q(x) such that P(x) & Q(x) is inductive
- Challenge is to discover Q(x)
IC3 / PDR-based Model Checking

- An incremental, inductive-strengthening algorithm
  - Clever use of SAT solvers to find inductive-strengthening $Q(x)$
    - Identify individual counterexamples to induction (CTI)
    - Strengthen $Q(x)$ enough to eliminate specific counterexample
    - Repeat until no more CTIs (i.e., $Q(x)$ becomes inductive)
  - Low memory consumption
  - Can also find firings and bounded proofs
    - Though usually not as effectively as BMC

Agenda

- Background
  - Model Checking
  - How Different From Simulation
  - Formal Friendly Design Pattern
- Technology
  - SAT-based BMC
  - IC3/PDR
- Application
  - Non-Determinism

Non-Determinism (ND)

- Formal property is written with extra inputs
  - The extra inputs are called ND-inputs
  - Formal can drive the ND-input values to the whatever values required to cause a violation to occur

ND Motivating Example

- How would you demonstrate that a fifo has a bug?
  - Looking at the waveform
    - Choose a time, $t$, in the waveform
    - Look at value enqueued at $t$ and the number of elements in the fifo
    - Count the number dequeues until value, enqueued at $t$, is dequeued
    - See that the value dequeued does not match the value enqueued
  - Using an ND-input, formal can pick $t$ for you
Checker State Machine Diagram

How does ND help?
- Thanks to the ND-input `start_looking_now`, the only state in the property is `idx` and `value`:
  - Much less state required than for a simulation checker:
    - 1 word + 1 counter for formal ND-property
    - N words + counter for simulation property (where N is the fifo depth)
- Using ND has reduced the property state space:
  - Reducing number of state bits by using ND-inputs makes the property easier for formal to analyze
- Examples of checkers where ND can significantly reduce the amount of verification state:
  - Data integrity checks
  - Scoreboards

Now you can talk like a formal expert!
- Background:
  - Model Checking
  - How Different from Simulation
  - Formal Friendly Design Pattern
- Technology:
  - SAT-based BMC
  - IC3/PDR
- Application:
  - Non-Determinism
Notes
At the heart of most products is a complex CPU that implements a variety of performance enhancements (pipelines, multiple instruction issue, out-of-order execution, branch prediction and caching) that can break the functional correctness of the core and can be challenging to test and verify.

T&VS have a wealth of experience in complex CPU verification both advising and supporting verification strategies and their execution. asureISG extends this consultancy service to provide a tool that verification engineers can use to accelerate the verification of CPUs with complex performance enhancements.

asureISG is an advanced Instruction Stream Generator tool from T&VS that eases CPU verification by allowing users to generate streams of assembly instructions for us in CPU verification.

Key Features & Benefits
asureISG accelerates the completion of CPU code-coverage and helps discover bugs that often go unnoticed with traditional directed testing:
- Event generation for mimicking complex system (e.g. sensor inputs)
- Random generation of structures (e.g., directed graphs)
- Developed in pure C++
- Directly generate:
  - Directed instruction assembly
  - Constrained Random instruction assembly
  - Special scenarios to verify CPU performance and corner cases
- Generate cases with sequences, constraints and policies
- Generate performance monitoring instruction streams, including:
  - Various kinds of inputs, resources, resource sharing, etc.
  - Configure specialized streams for specific unit-level instruction verification

Sample Test Case

![Sample Test Case Diagram]
We hope you have found the conference interesting and informative.

Slides and recordings will be available on the TVS website:

http://www.testandverification.com/conferences/formal-verification-conference/fv2016/