Presentation to
techUK - The Electronics Network

January 26th, 2015

Electronics Design Verification for Automotive and Autonomous Systems

Test and Verification Solutions

helping companies to develop hardware and software that is demonstrably correct, safe and secure
Scare Stories?

- Car was “programmed” to stop in fast lane
- Drone ditches into housing estate
- Criminals steal Amazon deliveries
- Disappearing cars – cars drive themselves into arms of thieves

Toyota Prius cars that are being recalled to fix a hybrid control unit glitch that can cause the cars to automatically shut down and enter a limp-home failsafe mode.

Japanese Smart Toilet Vulnerable to Hackers

FBI warns driverless cars could be used as 'lethal weapons' Internal report sees benefits for road safety, but warns that autonomy will create greater potential for criminal 'multitasking'.

Online marketplace eBay is forcing users to change their passwords after a cyber-attack compromised its systems. The US firm said a database had been hacked between late February and early March, and had contained encrypted passwords and other non-financial data.
Agenda

- **Design Verification for Automotive**
  - V&V Practices
  - Compliance with ISO 26262

- **What difference does autonomous make?**
  - System complexity
  - Connectedness & security

- **PAS 754:2014**
  - UK Trustworthy Software Initiative (TSI)
  - Defines the overall principles for effective software trustworthiness
    - What not how
  - Facets of trustworthiness
    - Safety, Reliability, Availability, Resilience, Security

- **Conclusions**
Infineon Top3 Verification Challenges

- **Top1: Mastering Verification Complexity**
  - Continuous increase in number of IP’s and embedded processors
    - 2006: 30-40 IP’s, 1 CPU
    - 2011: 80+ IP’s, 6+ CPU’s
    - 2016: 120+ IP’s, 20 CPU’s?
  - The more IP’s the higher the risk of late spec & implementation changes
  - Driving towards true Hw/Sw Co-Verification
  - Reuse of verification environments / stimulus from IP-level into big multi-CPU SoC environments
V&V Practices - Static vs. Dynamic

**Static**
- Reviews
- Code Analysis
  - Linters
- Equivalence Checking

**Dynamic**
- Simulation
  - Dynamic Formal
    - Theorem Proving
    - Silicon
    - FPGA
    - Emulation
- Prototyping

**Verification**
- Formal
  - Model Checking
The mechanics of an advanced test bench

- Test
- Functional Coverage
- Coverage
- Checker
- Monitor
- Assertions
- Code Coverage
- Design Under Test
- Test
- Design Under Test
- assert
- Driver
- Stimulus generator
- constraint
- addr data
- Active
- Passive
- Code Coverage
Examples: functional coverage and assertions

- **Functional coverage**
  - This event occurs when the FSM is in this state
  - These 2 transactions are input back-back
  - Payload data has this error

- **Assertions**
  - a_busy and b_busy are never both asserted on the same cycle
  - if the input ready is asserted on any cycle, then the output start must be asserted within 3 cycles
  - stall cannot remain high indefinitely

A liveness property
Simulation is not enough – this requires formal verification
Advanced HW Verification Techniques in Software?

- **TVS has developed libraries for**
  - Constrained random generation in C and C++
  - A functional coverage model
  - Automation of DUT output checks
  - Code coverage collection
  - A “Requirements Driven Testing” strategy

- **Applied to bubble sort as an example**

- **TBD**
  - Add assertions
Combining Coverage: Code and Functional

- Constrained random verification is now common
- Measured by coverage
  - Functional
  - Code

<table>
<thead>
<tr>
<th>Functional Coverage</th>
<th>Code Coverage</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>Low</td>
<td>There is verification work to do.</td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
<td>Multi-cycle scenarios, corner cases, cross-correlations still to be covered.</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
<td>Verification plan and/or functional coverage metrics inadequate. Check for “dead” code.</td>
</tr>
<tr>
<td>High</td>
<td>High</td>
<td>Increased confidence in quality.</td>
</tr>
</tbody>
</table>

- But is this enough?
Inputs to Property Checkers

- **3 inputs to the tool**
  - A model of the design
  - A property or set of properties representing the requirements
  - A set of assumptions, expressed in the same language as the properties
  - *typically constraints on the inputs to the design*

- **For example**
  - Usually RTL
    - Items are transmitted to one of three destinations within 2 cycles of being accepted
      - $\text{(req\_in \&\& gnt\_in)} \rightarrow \text{##}[1;2]$
        - $(\text{req\_a || req\_b || req\_c})$
    - The request signal is stable until it is granted
      - $\text{(req\_in \&\& !gnt\_out)} \rightarrow \text{##}1$
        - req\_in
    - We would of course need a complete set of constraints
Property Checking – Outputs from the tools

- **Proved**
  - the property holds for all valid sequences of inputs

- **Failed\((n)\)**
  - there is at least one valid sequence of inputs of length \(n\) cycles, as defined by the design clock, for which the property does not hold.
  - In this case, the tool gives a waveform demonstrating the failure.
  - Most algorithms ensure that \(n\) is as small as possible, but some more advanced algorithms don’t.

- **Explored\((n)\)**
  - there is no way to make the property fail with an input sequence of \(n\) cycles or less
  - For large designs, the algorithm can be expensive in both time and memory and may not terminate
Completion Criteria – Metrics to Measure Confidence

- Coverage Numbers
- Bug rate drop
- Resolution of open issues
- Reviews complete
- Regression results
- Mutation analysis
- Software running on FPGA
- No failing properties
- Are all requirements tested
  - Which ones can be dropped?

“Verification - it's all about confidence”
Mike Bartley, SNUG 2001
Requirements Driven Signoff – ISO26262 Compliance

- Stakeholder Requirements
  - (Customers and internal)
- Product Requirements
- Product Architecture
- Product Specification and Features
- Verification & Test Plans
- Verification & Test Results

- Intent to implement

- Intent to verify

- Proof of implementation

- Requirements
Complex Bi-Directional Mappings

Export Reqs Status as XML:
- Req1 [x, √]
- Req2 [x, √]

Metrics can be:
- From HW verification
- From Silicon validation
- From SW testing
Key Elements of safety

- Plans & Standards
- Requirements
- Design Specifications
- Reviews and Analyses
- Testing (against specifications)
  - Unit
  - Software Integration
  - Software System
- Test Coverage Criteria
- Traceability
- Independence

Does the move towards automated and autonomous systems pose any particular problems for ISO 26262?

how much an ‘Item’ can be confined; higher degrees of uncertainty
Concurrency Testing

Testing of parallel SW has many challenges:

- Non-determinism
- Race conditions, deadlocks, livelocks
- Heisenbugs

- How to provoke the 10% of bugs down to parallelism?
  - Making rare events happen more often

- How to know when you are done?
  - New coverage models

University of Nevada (2006)
Most bugs due to interactions between threads
Coverage metrics for parallel code

Joint Research with University of Bristol

**Figure 15** An example EIAQG with loop

Event InterAction Graph
Virtual System Level Test Environment

Test Environment

Sensor Inputs

Software Under Test

Actuator Outputs

Environment Model

Configuration Parameters

Metrics

Test Database

Metrics Database

Regression Analysis

Software Under Test
Security Concerns

2014 Commercial spending $46 billion

2015 Nation State spending $120 billion

20% more breaches
30% higher cost per breach
Attackers are using asymmetric economics

The 'Dark Web' is a marketplace for criminals
Threat growth

Number of breaches per threat action category over time

Source: Verizon
The holiday hack

- 110 million account details stolen
- 46% drop in sales
- Potentially $1 billion cost to Target
- CEO and CIO sacked
Working around security

I can see your password from here!

© January 16, 2015  News Editor

Declan O’Riordan, head of security testing, TVS

How to keep your business protected

I have no mind-reading abilities, but I’m going to make a prediction that the password you chose for your employers’ corporate website probably requires eight characters or greater, and must require three of the following four character types: upper case letters; lower case letters; numbers; special characters. Nothing unusual in that, but hackers know you probably selected the first character to be a capital letter, followed by five lower-case letters, and ending in two digits – e.g. London15. If the minimum password length is nine characters, you’ll probably add another lowercase letter at position seven – e.g. Bristol15. If instead you have to use all four character types, you’ll likely add a special character at the end – e.g. Bristol!.

This is because instead of everyone making randomly individual choices, groups of people follow patterns of behaviour that result in predictable password topologies. Let’s work through the consequences of those password topologies for cyber security and understand why 97% of LinkedIn passwords have been “cracked”:

```
password (cleartext)

hash function

S1S-4T853U855Qve4K8f/8gkP6avKQe8Q

password store
```

"hello"
Perimeter / Network defences are failing

HP alone sift through 2.5 Billion security events per day

1.2 million variants of malware per day
20%-30% are caught by anti-virus

Conventional cyber-defence is based upon reactive pattern-matching ‘known bad’
The Application layer vs. Transport layer

- Web Application Security
- Network Security
- Host Security
Hacking the Controller Area Network (CAN) bus

- CAN networks are designed for high speed, reliable communications between ECU components operating in harsh environments.
- Currently the only data security methods for CAN networks are
  - the use of proprietary CAN message IDs
  - and a physical boundary between the CAN bus and the outside world.
- This presents a serious security issue
  - anyone with physical access to the vehicle's data bus could generate spoofed CAN traffic destined for various ECUs,
  - some of which could be responsible for critical vehicle operations such as the braking system or engine control unit.
- To prevent this manufactures of passenger vehicles do not publish the proprietary CAN message IDs for various components on the vehicle network.
- However, proprietary message IDs can be identified through a reverse engineering process
The constraints we need to work with

- Shortage of skilled security staff
  - One million unfilled security jobs (Cisco report 2014)
  - 40% of security roles remain vacant (Jacob West - HP)
- In-house security knowledge concentrated in network silos.
- Limited budget
- Training is rare, knowledge retention poor, abstract from projects.
- Design flaws account for 50% of vulnerabilities (IEEE Centre for Secure Design)
- 700 kinds of software weakness (MITRE), yet 60% of developers are unconcerned with security.
- Penetration testing: usually too little, too late.

- The Project Team have to take up the slack in security!
OWASP Mobile Top Ten

- M1: Weak Server Side Controls
- M2: Insecure Data Storage
- M3: Insufficient Transport Layer Protection
- M4: Unintended Data Leakage
- M5: Poor Authorization and Authentication
- M6: Broken Cryptography
- M7: Client Side Injection
- M8: Security Decisions Via Untrusted Inputs
- M9: Improper Session Handling
- M10: Lack of Binary Protections
OWASP Web Application Top Ten

• A1 Injection
• A2 Broken Authentication and Session Management
• A3 Cross-Site Scripting (XSS)
• A4 Insecure Direct Object References
• A5 Security Misconfiguration
• A6 Sensitive Data Exposure
• A7 Missing Function Level Access Control
• A8 Cross-Site Request Forgeries (CSRF)
• A9 Using Components with Known Vulnerabilities
• A10 Unvalidated Redirects and Forwards
OWASP Draft Cloud Hosting Top Ten

• Accountability & Data ownership
• User Identity federation
• Regulatory compliance
• Business continuity & resiliency
• User privacy & secondary usage of data
• Service and data integration
• Multi tenancy and physical security
• Incidence analysis and forensic support
• Infrastructure security
• Non production environment exposure
  – + Patching & Vulnerability management
  – + Lack of transparency of security controls and difficulty auditing
  – + Integration between cloud and internally hosted services
Compliance models

PAS 754: 2014 Introduction

- Sponsored by the Trustworthy Software Initiative (TSI)
- Supported by the UK Government National Cyber Security Programme (NCSP)
- Facilitated by The British Standards Institution

Objective

to provide a consensus specification for software trustworthiness, either as a stand-alone document or as a companion and complement to other relevant standards
PAS 754: Five aspects of Software Trustworthiness

Figure 1 – Facets of trustworthiness

- **Safety**: The ability of the system to operate without harmful states
- **Reliability**: The ability of the system to deliver services as specified
- **Availability**: The ability of the system to deliver services when requested
- **Resilience**: The ability of the system to transform, renew, and recover in timely response to events
- **Security**: The ability of the system to remain protected against accidental or deliberate attacks
Trustworthiness Levels

- Role of Software
- Impact of Software

**Figure 6 – Trustworthiness level matrix**

<table>
<thead>
<tr>
<th>Role</th>
<th>None</th>
<th>Routine</th>
<th>Significant</th>
<th>Critical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Paramount</td>
<td>N/A</td>
<td>TL3</td>
<td>TL4</td>
<td>TL4</td>
</tr>
<tr>
<td>Explicit</td>
<td>N/A</td>
<td>TL3</td>
<td>TL3</td>
<td>TL4</td>
</tr>
<tr>
<td>Implicit</td>
<td>N/A</td>
<td>TL2</td>
<td>TL3</td>
<td>TL3</td>
</tr>
<tr>
<td>Ancillary</td>
<td>TLO</td>
<td>TL1</td>
<td>TL2</td>
<td>TL3</td>
</tr>
</tbody>
</table>
Aligning with existing compliance practices

- **Existing safety standards**
  - IEC61508: Functional Safety of Electrical/Electronic/Programmable Electronic Safety-related Systems
  - DO178: Software considerations in airborne systems and equipment certification
  - EN50128: Software for railway control and protection systems
  - IEC60880: Software aspects for computer-based systems performing category A functions
  - IEC62304: Medical device software -- Software life cycle processes
  - ISO26262: Road vehicles – Functional safety

- **Security standard**
  - BS ISO/IEC 15408, Information technology - Evaluation criteria for IT security
  - ISO/IEC 27034 Application security standard

- **Do we ask organisations to comply with multiple standards?**
  - Mapping between Safety Integrity Level (SIL) and trustworthiness level
  - V&V activities differ according to levels
    - E.g. independence only required at higher levels

*PAS 754 provides a companion and complement to other relevant standards*
Conclusions

- **Correct**
  - System level software testing
  - Parallel software testing

- **Safe**
  - Conform to appropriate standards

- **Secure**
  - Connectivity raises major concerns
  - No suitable standard
Q&A

- mike@testandverification.com
- www.testandverification.com

Available Whitepapers

- Find out what every well-formed tester should know about application security is, why it is important, who should be doing it, and how. This whitepaper is based on the award winning paper presented at EuroSTAR 2015. More …

Development Guidelines for Reducing the Top-Ten Most Critical Web Application Security Risks
- Essential for any members of the project team who design, code and debug web applications this whitepaper details the good practice for avoiding security vulnerabilities on any Web application project More …

Testing Procedures for Reducing the Top-Ten Most Critical Web Application Security Risks
- Managers and Testers reading this report will have a good understanding of how to test a web application project for security vulnerabilities. More …

Test and Verification Solutions

helping companies to develop hardware and software that is demonstrably correct, safe and secure