Can We Trust Hardware?
Detecting malicious logic and Trojans in IPs and SoCs

Sergio Marchese
Technical Marketing Manager
OneSpin Solutions
sergio.marchese@onespin.com
IC Integrity
Functionally correct, safe, secure, and trusted SoCs/ASICs/FPGAs

OneSpin provides certified **IC Integrity Verification Solutions** to develop functionally correct, safe, secure, and trusted integrated circuits.
Agenda

Hardware vulnerabilities

Malicious insertions during IC design

Trust assurance and detection of hardware Trojans
Trust Assurance for Integrated Circuit (IC)
All stages of the supply chain are vulnerable

Source: DARPA
ASIC Projects with Security Features

FPGA Projects with Security Features

Hardware Vulnerabilities

6488 vulnerabilities recorded in **2015** (CVE-MITRE)

*Data from MITRE/NIST CVE website*

*Source: DARPA*
Trust Assurance and Security Verification

Require new solutions, metrics, processes

**Functional Correctness**

Does the IC do what it is supposed to do?

**Safety**

Is the IC tolerant to faults?

**Trust and Security**

Does the IC do anything that it is not supposed to do?

Use Cases

Random Faults

Trojans, Misuse Cases
Hardware Trojans Taxonomy
Anatomy: triggers on rare, hidden condition - delivers damaging payload

Source: Trust-Hub
IC Development – Malicious Insertions

Functional Trojans in complex designs are hard to detect

- Undocumented functions, inconsistencies
- Modifications through manual changes, EDA tool scripts
- Optional

Specification → SystemC/C++ → RTL → Synthesis Netlist → P&R Netlist → GDSII

Fabrication → ASIC → Program → FPGA

Additional code, intentional code errors (in-house, 3PIPs)

Modifications through manual changes, tool scripts
Trust Assurance Challenges
Functional hardware Trojans are NOT bugs

IPS are complex and support a variety of configurations

Code review unlikely to spot malicious code

Functional verification targets bugs, not deliberately stealthy Trojans

ASIC/FPGA/SoC developers need automated processes to increase confidence in IP trustworthiness
Automated Trust Assurance Technology
Applicable to any IP, 3PIP

<table>
<thead>
<tr>
<th>Example</th>
<th>Source</th>
<th>Trigger</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>BasicRSA-T100</td>
<td></td>
<td>Input Value</td>
<td>Secret Leaking</td>
</tr>
<tr>
<td>RS232-T300</td>
<td></td>
<td>Large Counter</td>
<td>Data Corruption</td>
</tr>
<tr>
<td>RS232-T700</td>
<td>Trust Hub</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RS232-T800</td>
<td></td>
<td>Input Sequence</td>
<td></td>
</tr>
<tr>
<td>RS232-T900</td>
<td></td>
<td></td>
<td>DoS (Output stuck)</td>
</tr>
<tr>
<td>DeTrust</td>
<td>Conference Paper (CCS)</td>
<td></td>
<td>Privilege Escalation</td>
</tr>
<tr>
<td>SDRAMC-T100</td>
<td>OneSpin</td>
<td></td>
<td>Control Signal Corruption</td>
</tr>
</tbody>
</table>

SystemC/C++, RTL, Netlist

Script

Automated Trust Assurance

Trust Assessment Report

Debug
Automated Trust Assurance Technology
What’s under the hood

Formal is the only technology that can prove the absence of certain behaviors

Intelligent structural analysis identifies dependencies between hardware functions in the design and reduces proof complexity

Detection rules encapsulate information about known Trojans and suspicious or uncommon code patterns
Proof of Trojan Absence

GapFreeVerification
Develop executable specification in the form of assertions
Prove that executable specification has no gaps or inconsistencies
Prove that executable specification and RTL are functionally equivalent

Abstraction
Specification, RTL

Scope
New IP, critical IP, untrusted 3PIP

Benefits
Detect errors and inconsistencies in the specification
Prove 100% equivalence between specification and implementation
Detect unknown, hidden logic
Example: RISC-V Rocket Core
RISC-V Verification App - Part of RISC-V Integrity Verification Solution

Design Issues Reported:

• **Issue 1757**: Jump instructions store different return PC - instruction fetch unit responsible to prevent this issue
• **Issue 1752**: DIV result not written in register file - confirmed and fixed in RTL
• **Issue 1861**: Replay of illegal opcodes / generating memory accesses - Illegal opcodes not throwing an exception
• **Issue 1868**: Undocumented non-standard instruction - opcode 32’h30500073 / CEASE instruction
• **Issue 1949**: Undocumented CSR that reads back 0

Highlights

• Ensure correctness of RISC-V ISA specification
• Two hours runtime
• Unbounded proofs
• Low effort (few days) to set up
• Detect unknown/undocumented/hidden logic

* https://github.com/freechipsproject/rocket-chip/issues/

• 64-bit core
• 5-stage pipeline
• Single-issue, in-order pipeline
• Out-of-order completion of long latency instructions (e.g., DIV)
• Branch prediction
• Instruction replay
• Verified
• Taped-out multiple times
Prove of Trojan Absence in Netlist/Bitstream
Maintain trust through design transformations

Combinational/Sequential Equivalence Checking
Prove that no Trojan has been inserted during design transformation

Abstraction
Netlist, Bitstream

Scope
Any IP, SoC

Effort
Low
Trust Assurance with OneSpin
Applicable to many classes of Trojans

- Insertion Phase
  - Specification
  - Design
  - Fabrication
  - Testing
  - Assembly and Package

- Abstraction Level
  - System
  - Development Environment
  - Register Transfer
  - Gate
  - Layout
  - Physical

- Activation Mechanism
  - Always On
  - Triggered
  - Internally
    - Time Based
    - Physical Condition Based
  - Externally
    - User Input
    - Component Output

- Effect
  - Change Functionality
  - Degraded Performance
  - Leak Information
  - Denial of Service

- Location
  - Processor
  - Memory
  - I/O
  - Power Supply
  - Clock Grid

- Physical Characteristic
  - Distribution
  - Size
  - Type
  - Parametric
  - Functional
  - Structure
    - Layout Same
    - Layout Change
Additional Information
Learn more about trust assurance and detection of hardware Trojans

onespin.com/solutions/trust-security

Hardware Trojans and The Problem of Trust in Integrated Circuits
onespin.com/blog

Complete Formal Verification of RISC-V Processor IPs for Trojan-Free Trusted ICs
Government Microcircuit Applications & Critical Technology (GOMACTech) Conference
Albuquerque, NM, USA, 2019

sergio.marchese@onespin.com
Drop me an email to request a copy of the GOMACTech paper and for additional information