SYNOPSYS®

Security Verification of SoC Hardware

NIST Hardware Security Workshop – Enhancing Security of Devices and Components Across the Supply Chain

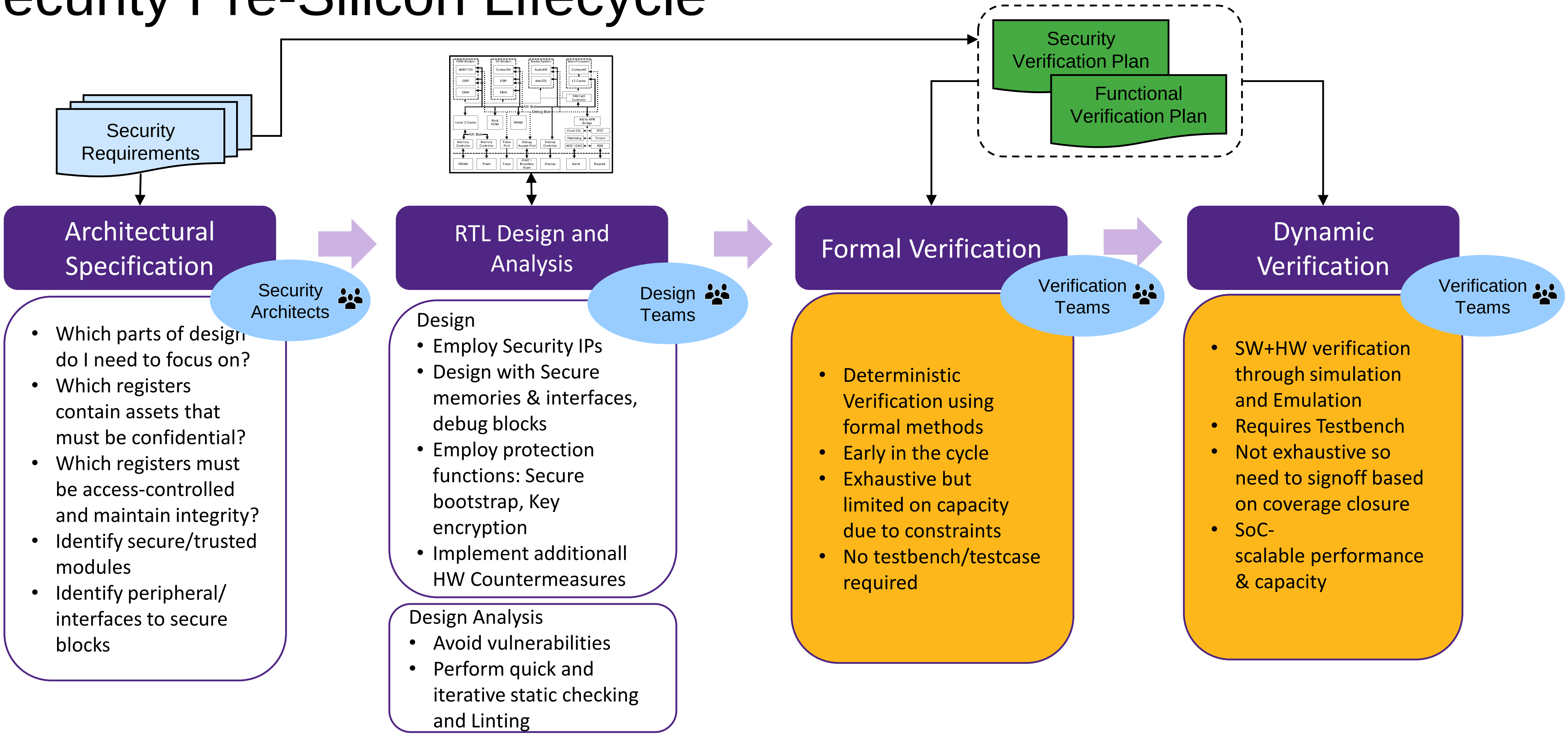
Mike Borza, Synopsys Scientist

February 27, 2024

Where Are We Now?

- Awareness of the need for security in SoCs has been steadily rising
  - Now drives design requirements, with a corresponding need for verification
  - Increasing acceptance that **hardware** must be foundation for security
- EECAD and verification tool vendors are now adapting and adding features to existing products and product lines
- Some specialty products and suppliers exist
  - To be most effective, these need to work well with existing SoC development and manufacturing flows
- Point solutions and dissimilar development flows abound
  - Lack of standards for security collateral in a form that can consumed by tools impedes rapid and thorough verification
- Some progress on standardization
  - Accellera SA-EDI (Security Annotation for Electronic Design Interchange) → IEEE P3164
  - Mitre CWE database public APIs in development

Security Pre-Silicon Lifecycle



Architectural Specification

Security Architects

- Which parts of design do I need to focus on?
- Which registers contain assets that must be confidential?
- Which registers must be access-controlled and maintain integrity?
- Identify secure/trusted modules
- Identify peripheral/ interfaces to secure blocks

RTL Design and Analysis

Design Teams

Design
- Employ Security IPs
- Design with Secure memories & interfaces, debug blocks
- Employ protection functions: Secure bootstrap, Key encryption
- Implement additionall HW Countermeasures

Design Analysis
- Avoid vulnerabilities
- Perform quick and iterative static checking and Linting

Formal Verification

Verification Teams

- Deterministic Verification using formal methods
- Early in the cycle
- Exhaustive but limited on capacity due to constraints
- No testbench/testcase required

Dynamic Verification

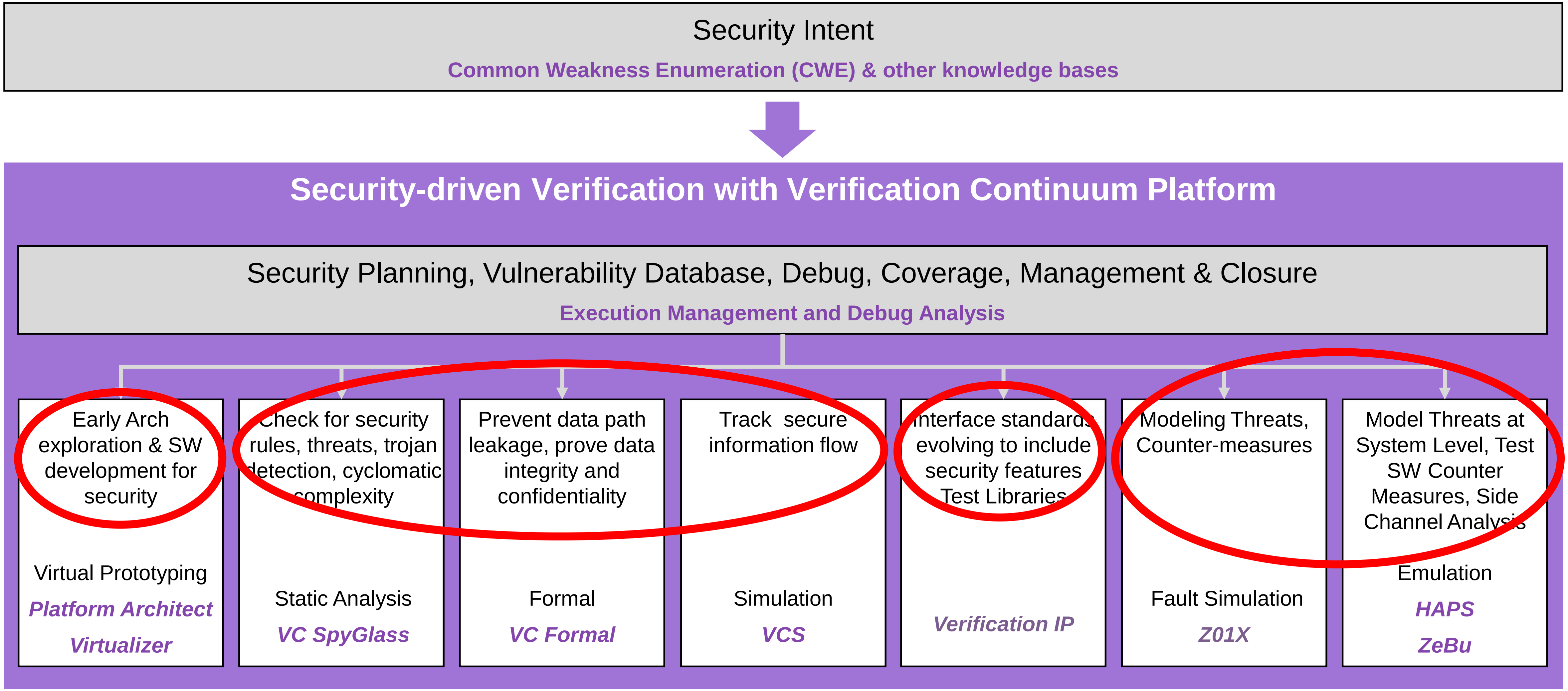
Verification Teams

- SW+HW verification through simulation and Emulation
- Requires Testbench
- Not exhaustive so need to signoff based on coverage closure
- SoC- scalable performance & capacity

Security Verification plan augmenting the Functional Verification plan is essential

Security Verification Continuum Over the Pre-Silicon Lifecycle

Intensive automation to provide security verification completeness and robustness



Security Intent

Common Weakness Enumeration (CWE) & other knowledge bases

Security-driven Verification with Verification Continuum Platform

Security Planning, Vulnerability Database, Debug, Coverage, Management & Closure

Execution Management and Debug Analysis

Early Arch exploration & SW development for security

Virtual Prototyping

Platform Architect

Virtualizer

Check for security rules, threats, trojan detection, cyclomatic complexity

Static Analysis

VC SpyGlass

Prevent data path leakage, prove data integrity and confidentiality

Formal

VC Formal

Track secure information flow

Simulation

VCS

Interface standards evolving to include security features Test Libraries

Verification IP

Modeling Threats, Counter-measures

Fault Simulation

Z01X

Model Threats at System Level, Test SW Counter Measures, Side Channel Analysis

Emulation

HAPS

ZeBu

Aspects of Security Verification

CONFIDENTIALITY
AVAILABILITY
INTEGRITY

A. Functionality of Security Features

- Security IPs, Root of Trust, Secure boot, Secure keying, Secure processing, Secure interfaces, Encryption, Decryption, Authentication, …

→ Regular Functional Verification of security functions!

→ Augmented by Security Verification based on threats

B. On-Chip Data Propagation

- Security of data at rest and in motion
- *Leakage:* Security-critical data (e.g., keys) cannot reach non-secure modules
  - Are keys contained within the secure module?
- *Integrity Violation:* Data cannot be over-written with non-secure data
  - Can data be corrupted via the system's interfaces/bus, debug and test logic?

C. Malicious Attacks "Tampering"

- Silicon can be probed or faulted directly – not via system's interfaces/buses
- Can the HW security be subverted by an external probe or induced fault?
  - Probing sensitive data storage
  - Flipping supervisor bits

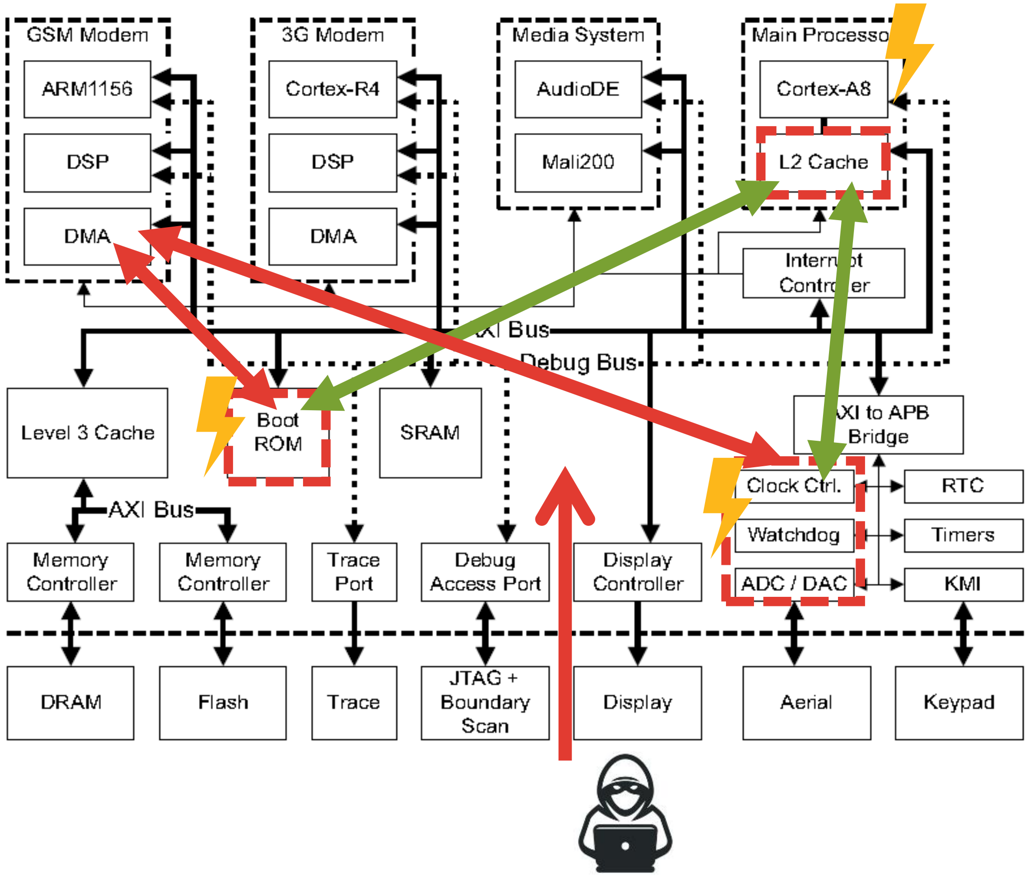


Diagram courtesy ARM ®, Building a Secure System Using TrustZone ® Technology

→ Ensuring Security objectives requires a mixture of established and novel verification approaches!

Security Verification Plan

Functional Verification Plan

What’s Next?

- More and better static and dynamic analysis tools to automate reasoning about security
  - Guaranteed high levels of coverage
  - Formal tools that can better reason about **real** physical realizations (not idealized or incomplete models)
- Standardizing collateral to describe and communicate security information about IPs and SoC subsystems among producers and consumers (people and tools)
  - Extend SA-EDI and eventual P3164 standard beyond IP to better deal with SoCs
  - More complete and concrete refinement of CWEs
    - ✓ CWE-1277: Firmware Not Updateable
    - ✗ CWE-1331: Improper Isolation of Shared Resources in Network On Chip (NoC)
- AI-based tools will emerge that encode knowledge of threats and weaknesses and raise productivity
  - But raise questions about completeness of coverage of the threat and mitigation spaces

Thank You