ANALOG DEVICES

AHEAD OF WHAT'S POSSIBLE™

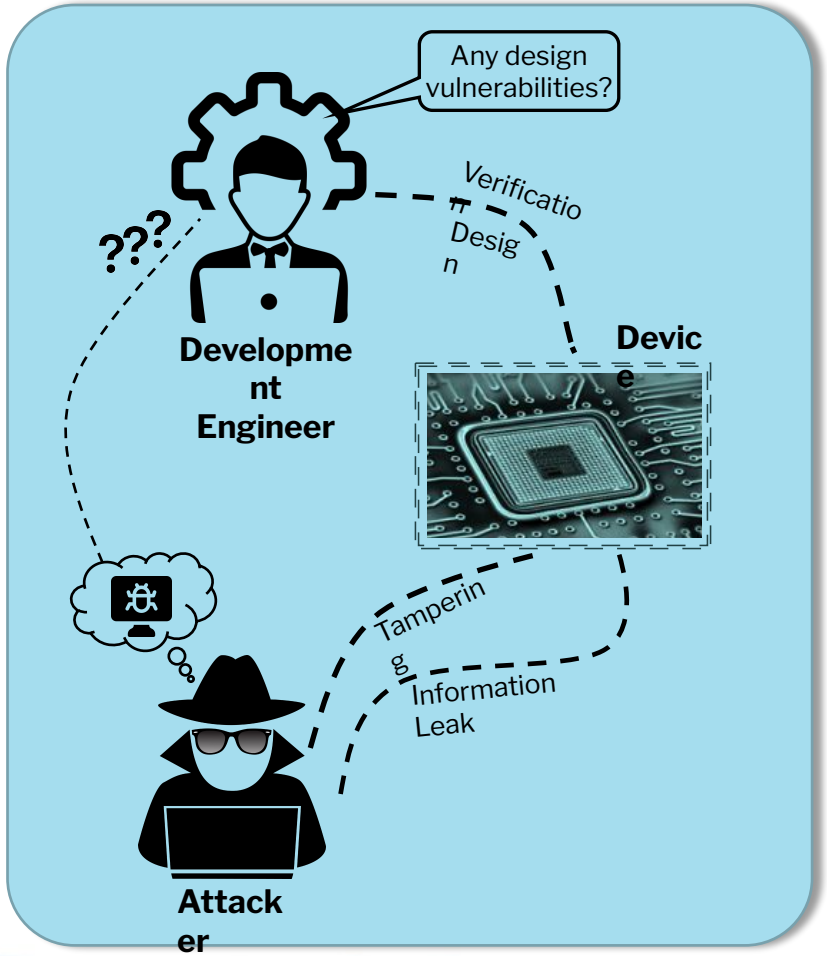
Security Verification of Hardware System Using Formal

Lee Anthony Grajo

Analog Devices

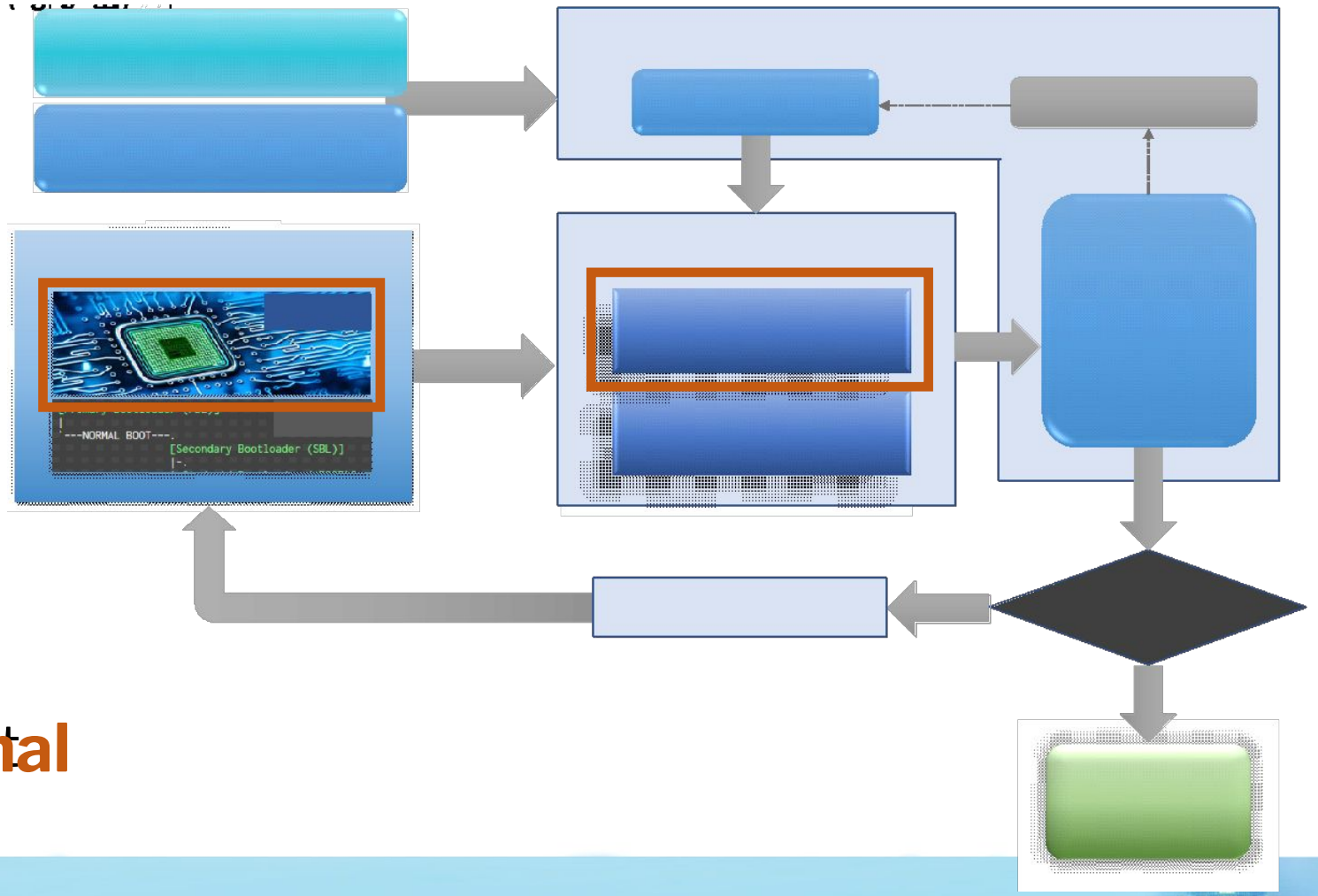
The Challenges of Developing Secure Systems

- There is an increasing need for secure systems that is more robust against attacks.
  - Entry points for attackers to compromise a device
    - Functionality weakness
    - Security implementation flaws
    - Configuration bugs
- In pre-silicon verification, functional verification is no longer enough. Methodologies should also focus on the security aspect of verification.



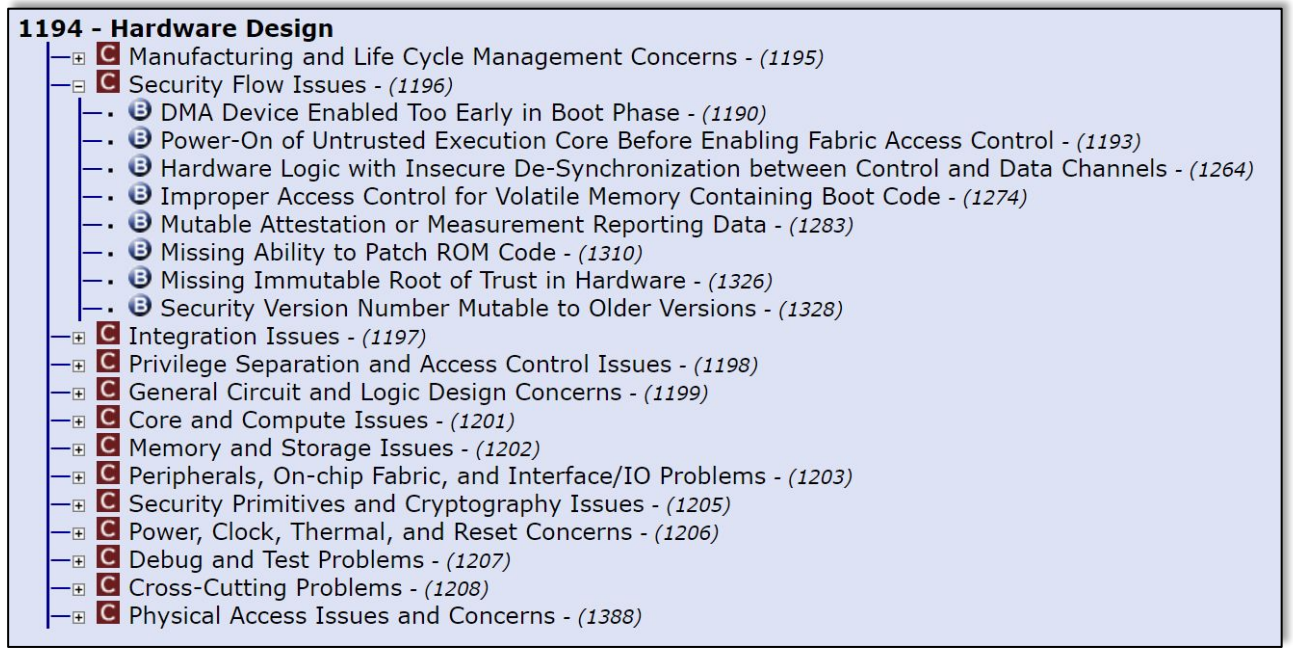
Security Verification Solution

- Use a list of weaknesses developed by a community
  - Target the source of vulnerabilities
  - Convert each applicable weakness to test items
- Use the right verification tool
- Our focus is on using **Formal** as verification tool for **hardware**
  - Different tools might be suitable for each weakness



Common Weakness Enumeration

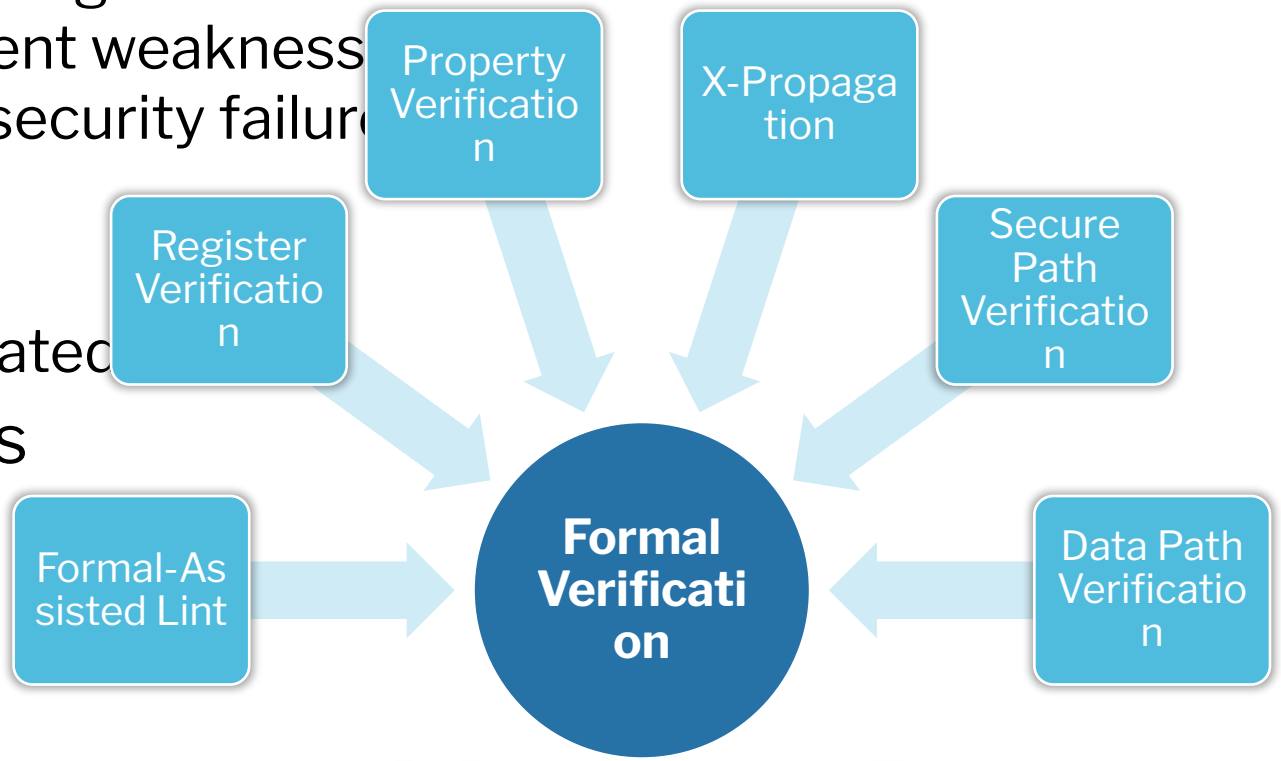
- A list of hardware and software weakness types seen in products in the field
- A weakness is a condition in a software, firmware, hardware, or service component that, under certain circumstances, could contribute to the introduction of vulnerabilities



CWE 1194 List – From MITRE CWE Site

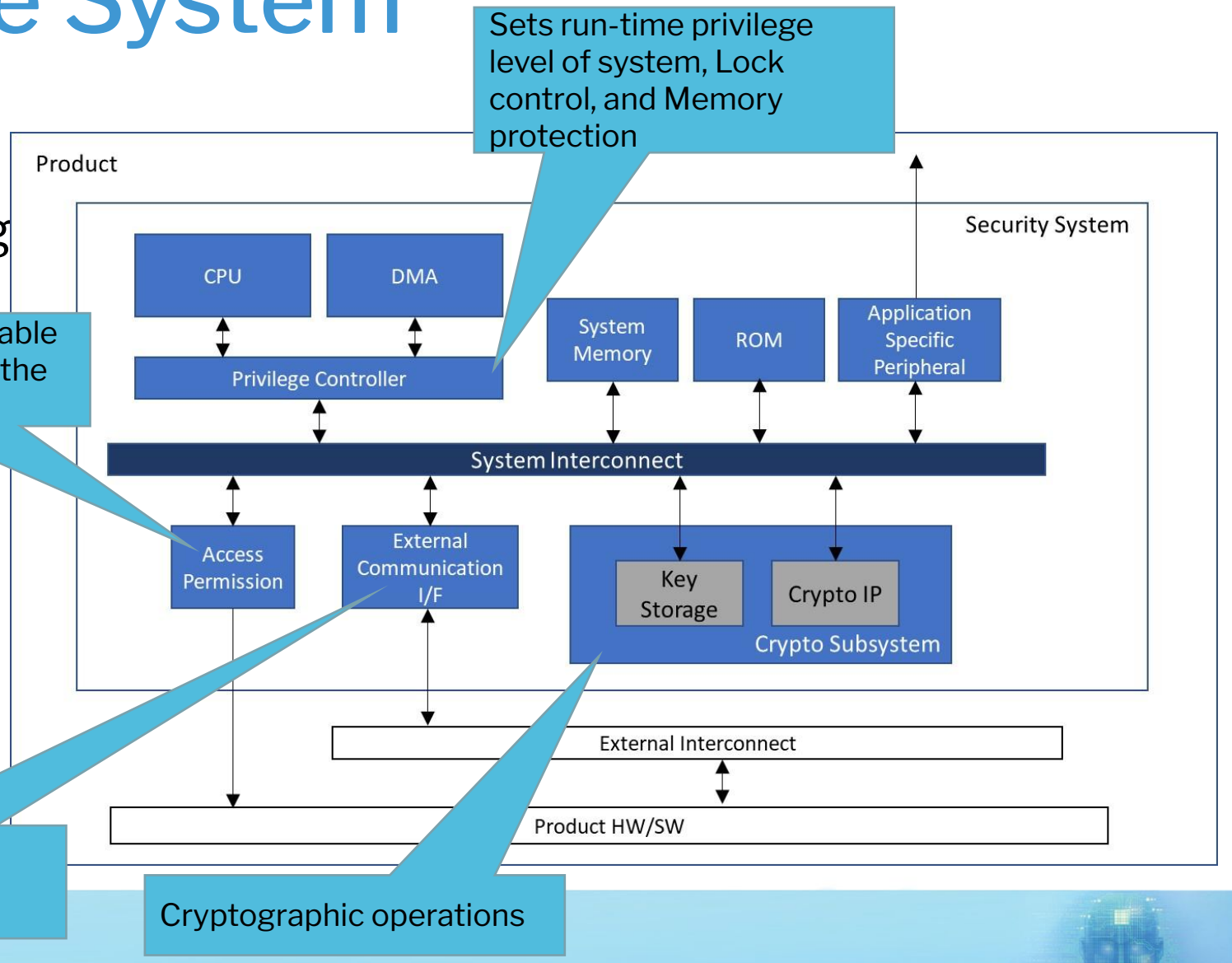
Why Formal Verification?

- Exhaustive in nature
  - Uses all possible stimulus as input to design
  - Can hit corner cases which can represent weakness
  - Helps reduce the chances of a critical security failur
- Easy to Setup
  - No need for testbench
  - Some properties are automatically created
- CWEs focus on certain design issues
  - Data Transfer
  - Register Access
  - Security-related control

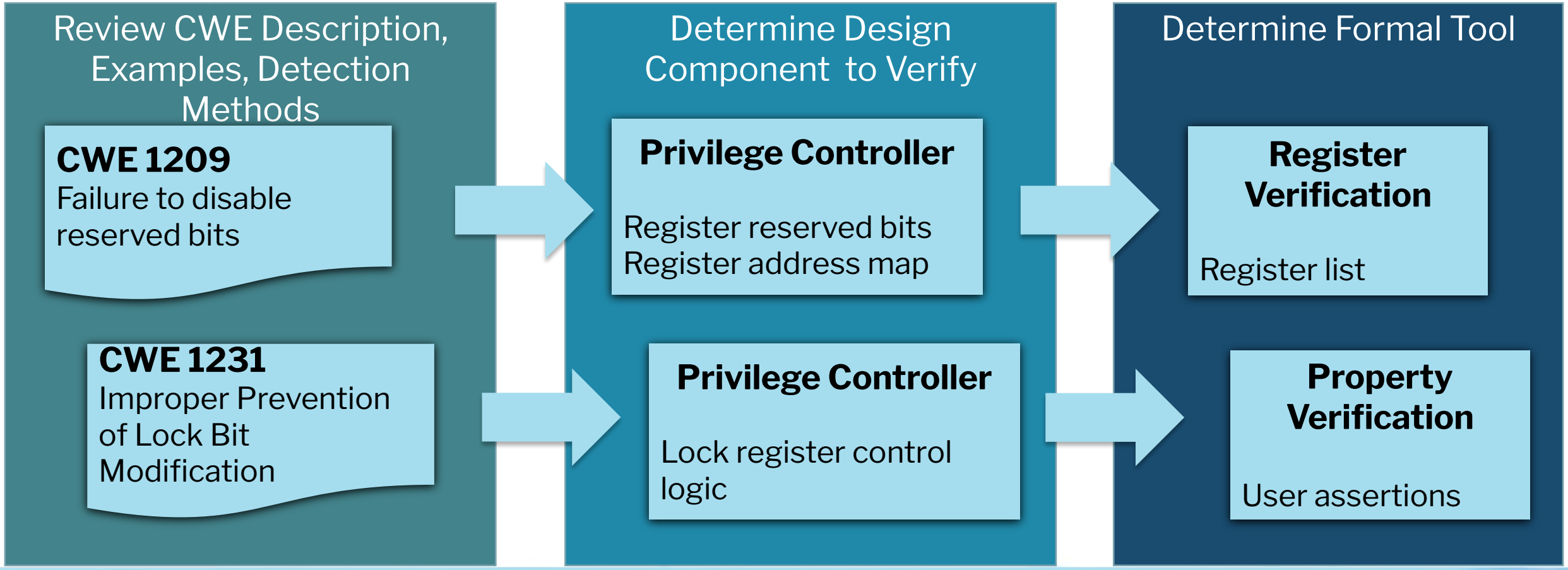


Security Hardware System

- Reusable security solution that provides isolation to processing sensitive data
- Features:
  - Secure Boot
    - Security System secure boot
    - Host secure boot authentication
  - Feature enablement
  - Identity and Authentication
  - Key management



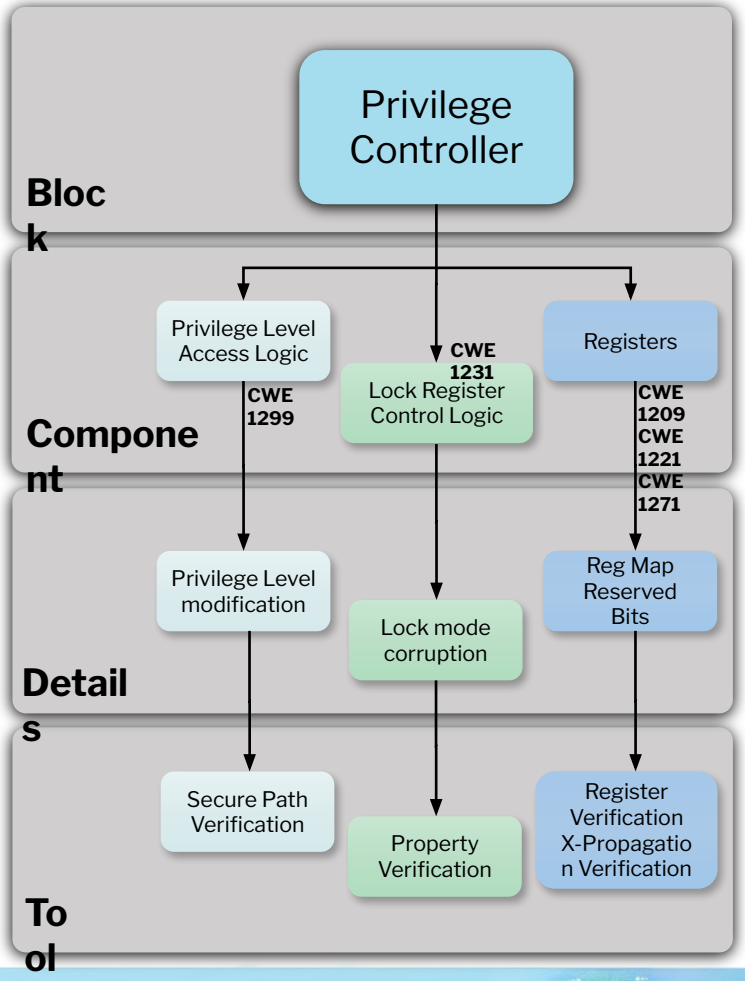
Mapping CWE to Formal Tool



Security Verification Details

- CWE List, Assets and Formal Tools

CWE	Description	Security Asset	Formal Tool
1209	Failure to Disable Reserved Bits	Privilege Controller	Register Verification, Property Verification
1221	Incorrect Register Defaults or Module Parameters	Privilege Controller, Security System Top	Register Verification, Property Verification
1231	Improper Prevention of Lock Bit Modification	Privilege Controller	Property Verification
1271	Uninitialized Value on Reset for Registers Holding Security Settings	Privilege Controller	X-Propagation Verification Register Verification
1299	Missing Protection Mechanism for Alternate Hardware Interface	Privilege Controller	Secure Path Verification
1314	Missing Write Protection for Parametric Data Values	Privilege Controller, Access Permission, Security System Top	Property Verification Formal-Assisted Lint
1317	Improper Access Control in Fabric Bridge	Security System Top	Secure Path Verification



Results

- Shorter setup time in Formal
  - Formal setup took 1 week compared to 1 month coding if using simulation
- Fast completion of tests in Formal
  - Block-level formal test runs in less than a minute compared to 1-2 mins in simulation
- Formal challenges
  - Convergence issues
  - Setup refinement for some formal tools

CWE	Formal Tool	Property Count	Result	Security Asset
1209	Register Verification	1176	PASS	Privilege Controller
	Property Verification	1		
1221	Register Verification	3797	PASS	Privilege Controller
	Property Verification	13		Security System Top
1231	Property Verification	132	PASS	Privilege Controller
1271	X-Propagation Verification	2600	PASS	Privilege Controller
	Register Verification	843		
1299	Secure Path Verification	22	PASS	Privilege Controller
1314	Property Verification	1	PASS	Security System Top
	Formal-Assisted Lint	1116	PASS FAIL	Privilege Controller Access Permission
1317	Secure Path Verification	6	FAIL PASS Undetermined	Security System Top

Defects Found

- Security verification was done on a relatively stable hardware design
  - Prior functional verification was performed on the design
- Some issues were still found which proved the value of using CWE list
  - The CWEs prove to be helpful in verifying the weaknesses in the design that is not the focus of functional verification

CWE	Security Asset with Defect	Details
1231	Privilege Controller	The privilege level switching to the desired level does not occur when a particular address is accessed
1314	Access Permission	Out-of-bounds indexing issue was found when register setting is outside valid range
1317	Security System Top	Non-secure CPU I/F can read bus data which can contain decryption key information

Conclusion and Next Steps

- Using Common Weakness Enumeration (CWE) list as a reference to identify vulnerabilities in a design helps in determining important security assets (design) and test cases (formal properties) to test the security features of a design
- The use of Formal verification:
  - Helps accelerate bring-up of verification setu
  - Exhaustively checks the design
  - Helps detect issues in a shorter time
  - Increases confidence in the design

Next Steps:

- Additional checks on other blocks of the Security system
- Enable coverage collection and mapping to test items

Acknowledgements

- Albert Landicho – Analog Devices Inc.
- Larry Getzin – Analog Devices Inc.
- Nimay Shah – Analog Devices Inc.
- Ponnambalam Lakshmanan – Analog Devices Inc
- Ameya Mulye – Analog Devices Inc.
- Tom Weiss – Cadence Design Systems
- Adam Sherer – Cadence Design Systems

Thank You!