

8/31/26

COMP 435:

Computer Security Concepts



Lecture 6: Hardware as the Root of Trust



The University
of North Carolina
at Chapel Hill

<https://tinyurl.com/comp435-fa26>

Logistics Update



- First Written Assignment
 - One part of worksheet due Thursday (9/3) at 11:59
 - Second part due in class

Written Assignment Details: Security in Practice



- Choose a real cybersecurity incident from the last 5 years
- Use credible sources to figure out what happened
- Analyze it using concepts from class:
 - assets + security policies
 - adversary + capabilities
 - threat / weakness / attack / harm
 - attack tree
 - risk
 - one relevant MITRE ATT&CK technique
- You do not need to understand every technical detail
- Be clear about what is known vs. inferred vs. assumed
- You'll share your incident with classmates on Friday

Plan for Class Today



- 10:10-10:20: Review Trust
- 10:20-11:00: Intro to Hardware Security, HW as the Root of Trust!

Guiding Question 😊

What must we trust at the lowest level in a computing system?



UNIT 1: TRUSTING THE MACHINE

Trust

Trusted vs. Trustworthy



- **Trust:**
 - relied upon to meet its security specifications
 - Trust is about one system depending on another system
- **Trustworthy:**
 - deserving of trust; meets its security specifications
 - Trustworthiness is about a system's security posture

only trustworthy systems
should be trusted

Trusted Computing Base (TCB)



Def'n: those components upon which the security of the system relies

“Trusted system: a subsystem that is permitted to violate some global security policy”

--Goguen and Meseguer

minimize →

C.I.A.



Mail this for me, please



I'm trusting...
the TA,
The postal service
The roommates of my friend

secret

(postcard)



Trustworthy



- secure against intentional, malicious actions
- secure against accidental actions
- secure against 3rd party actions

C, I, A



Mail this for me, please



I'm still trusting the same entities
BUT the amount of trust required is
reduced → **accidental reading is
no longer a concern**

secret

To: Orange

(sealed
envelope)



Honest, but Curious



Def'n: an adversary that tries to glean information while adhering to the protocol

E.g., In an honest-but-curious threat model, the TA does not open the sealed envelope



LC-IA

Mail this for me, please



wi3.s-

To: Orange

*(encrypted and
signed secret in
envelope)*

TA, postal system, housemates no
longer trusted for confidentiality or
integrity... only for availability!

Some crypto now part of the TCB!





Practice Questions

1. Is it better for the TCB to be big, encompassing most of the system? Or small, with very little of the system counted as part of the TCB?

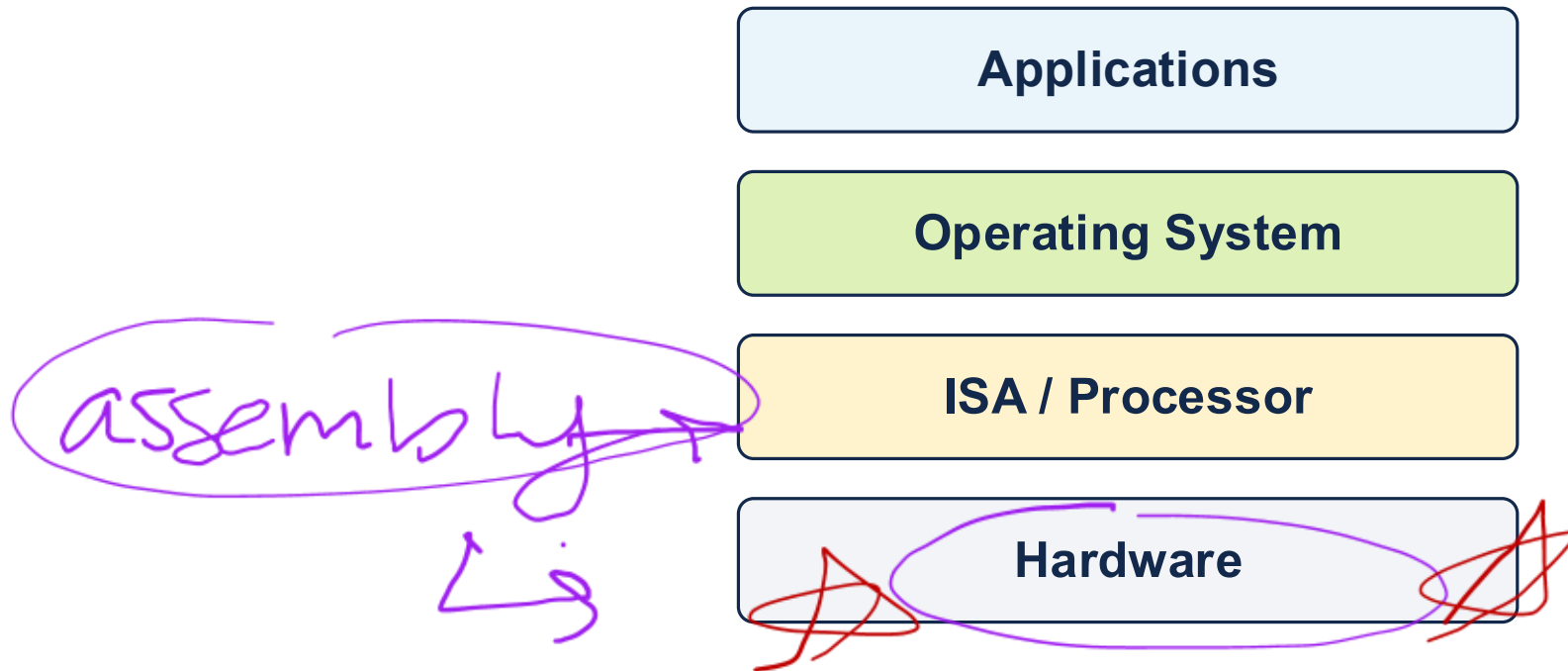
Small

2. It is instructive to consider an "honest, but curious" threat model because...

The Stack We Are Trusting



- Every software security mechanism rests on assumptions about the layers below it.



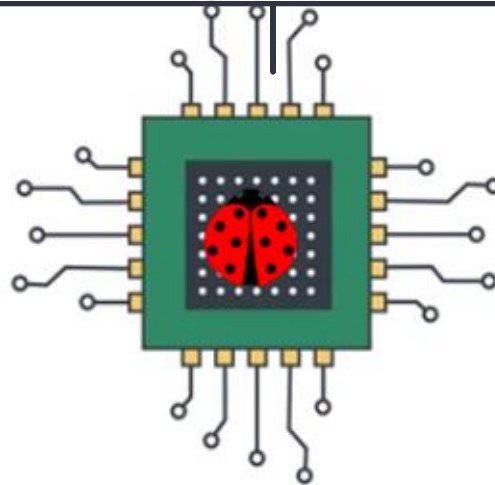
UNIT 1: TRUSTING THE MACHINE

Hardware Security 😊

Our lives depend on secure systems!



Hardware is the root of trust in all systems!

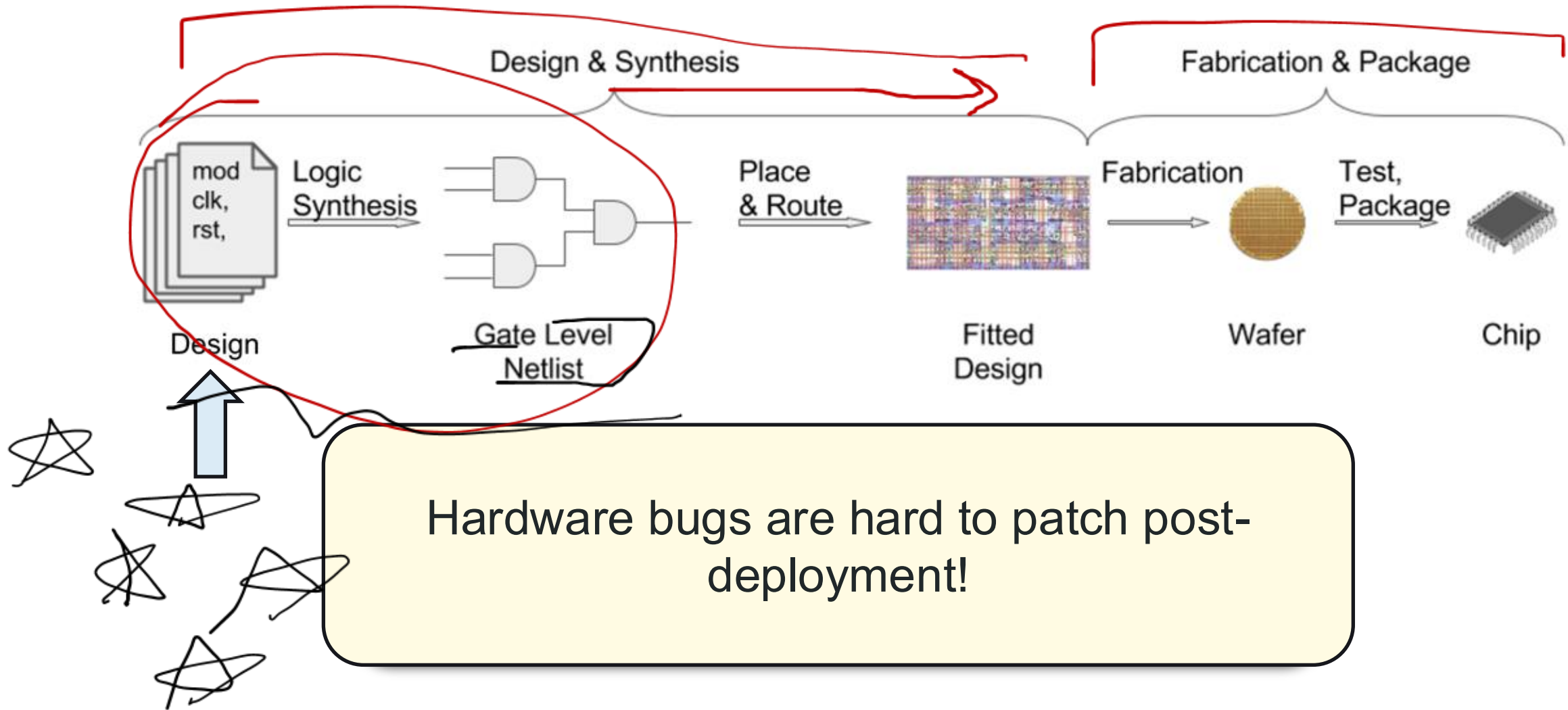




Hardware as part of the TCB?

- Our security relies on it! So **yes, it is part of the TCB.**
- → has to be proven trustworthy.
 - Hard to do for a full piece of HW (or a complex piece of code like an operating system).

The Problem(s)...



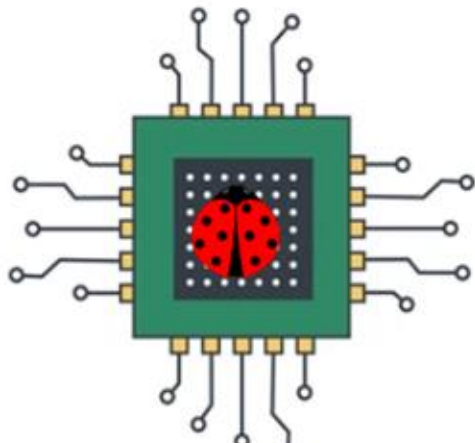
The Problem(s)...

Number of Transistors in CPUs over Time

Increasing design complexity!



↑ attack surface



Hardware Backdoor Discovered in RFID Cards Used in Hotels and Offices Worldwide

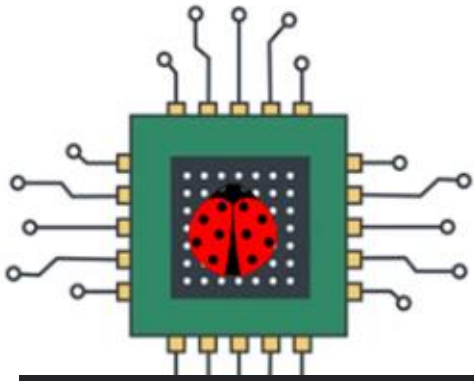
📅 Aug 22, 2024 👤 Ravie Lakshmanan

Unpatchable vulnerability in Apple chip leaks secret encryption keys

Fixing newly discovered side channel will likely take a major toll on performance.

Hardware vulnerabilities in Hitachi Energy, ABB, B&R ICS devices pose critical infrastructure threat

APRIL 07, 2025



Hardware Backdoor Discovered in RFID Cards Used in Hotels and Offices Worldwide

📅 Aug 22, 2024 👤 Ravie Lakshmanan

Unpatchable vulnerability in Apple chip leaks secret encryption keys

Fixing newly discovered

High-assurance verification is critical!

Hardware vulnerabilities in Hitachi Energy, ABB, B&R ICS devices pose critical infrastructure threat

APRIL 07, 2025

Security Boundaries

(Trust boundary)



Definition: a separation between parts of a system that have different levels of trust, privilege, or allowed access.



Examples:

one process vs. another process
user code vs. kernel code
one user's memory vs. another user's memory
software vs. privileged hardware operations



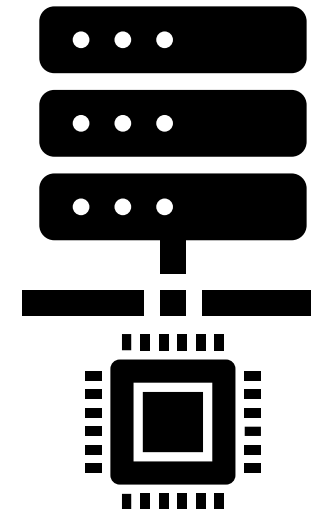
Crossing a security boundary should require a controlled check →

Complete mediation!

Hardware is the Security Foundation!



- The processor decides which **instructions** actually **execute**
- Hardware **mediates access** to memory and devices
- Software relies on hardware checks to **enforce security boundaries**
- If hardware allows software to bypass a boundary, higher-level protections may fail





UNIT 1: TRUSTING THE MACHINE

Trusted Systems

Trusted Systems Standards



- Orange Book
 - official name: the Trusted Computer System Evaluation Criteria
 - Developed in 1970s by US DoD.
 - 6 rankings, coupled security features w/ assurance features

- ✱ Common Criteria
 - Official name: Common Criteria for Information Technology Security Evaluation
 - Developed in 1990s by group of International Governmental Agencies
 - 7 assurance levls, decouple security features from assurance requirements

Trusted Systems' Process



Assurance

- specification
- design process
- evaluation

Trusted Systems' Process



- ***specification*** ← defines the desired security policy and functionality
- design process
- evaluation

Security spec vs. functional spec

*Security
policies

stating what
system does

What Guarantees Do We Expect?



• Correct Execution

- Instructions behave according to the architecture.
- **Restricted Operations**
 - Sensitive actions cannot be performed by ordinary programs
- **Controlled Transitions**
 - Crossing into privileged code happens through defined mechanisms
- **Memory Checks**
 - A program cannot freely access memory



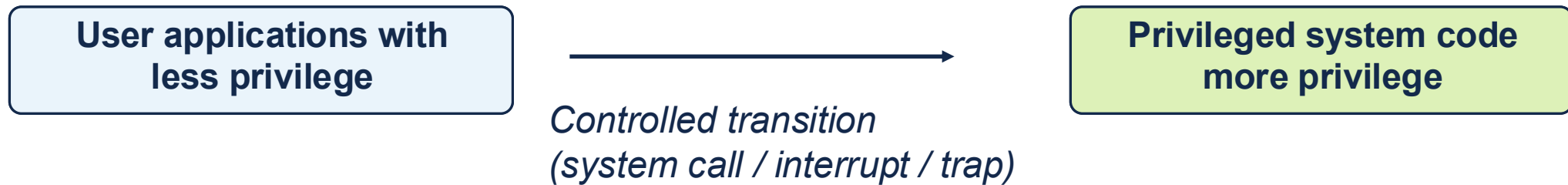
Trusted Systems' Process

- specification
- ***design process*** ← 1. model
2. design
3. implement
- evaluation

Ex: Hardware-Enforced Privilege



⚠ Not all code should have the same authority. ⚡



1. CPU tracks the current privilege level
2. Some instructions/resources are only accessible in privilege mode
3. If user requests privileged operation, CPU traps it to the OS

Thought Experiment: No Hardware Protection!



- Suppose every program could...
 - read any memory
 - modify any memory
 - execute any instruction
 - directly control devices
 - interrupt or stop other programs

Q: Which properties are at risk: C.I.A, Authorization?

Are there security guarantees the OS could still provide?

Properties at Risk → All!



- **Read any memory**

- confidentiality is basically gone. A program could read passwords, private files, encryption keys, or another process's data.

- **Modify any memory**

- integrity is gone. A program could alter another program's data, change security checks, or modify OS state.

- **Execute any instruction**

- privilege boundaries disappear. Untrusted code could perform operations that should be restricted to the kernel.

- **Directly control devices**

- confidentiality, integrity, and availability can all fail. A program might read disk contents, interfere with the network, or disable hardware.

- **Interrupt or stop other programs**

- availability fails. One program could prevent others from running or make the system unusable.

OS Cannot Provide Strong Security Guarantees!



- If hardware does not enforce restrictions, malicious software can simply bypass the OS's rules!
- The OS can define security policy, but it needs hardware mechanisms to actually enforce some of those rules.



Trusted Systems' Process

- specification
- design process
- ***evaluation***

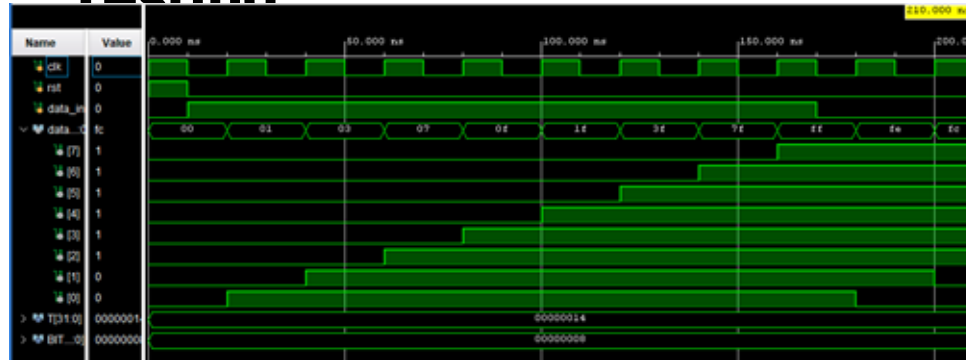


- testing
- red-, blue-team exercises
- formal verification

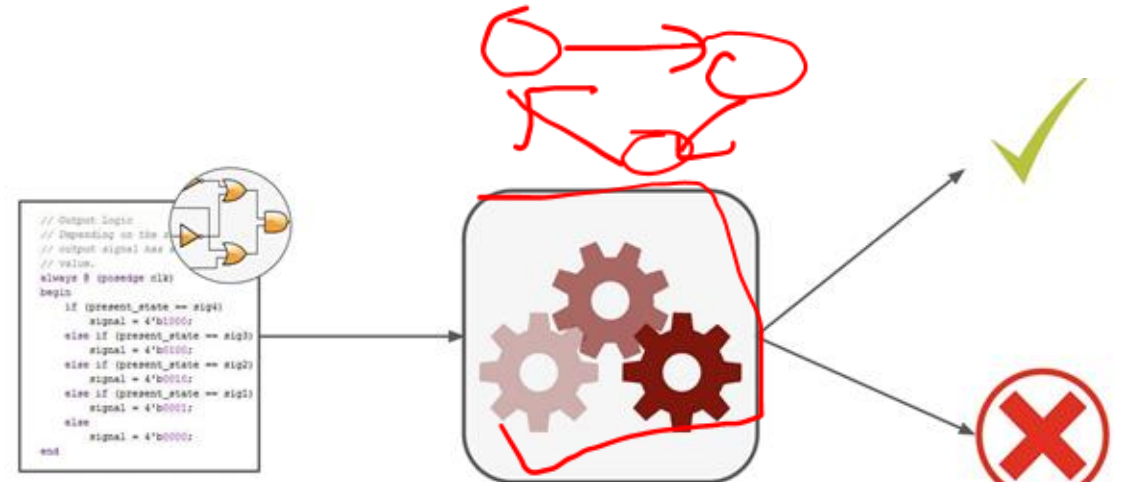
** regression
testing **

Current State of the Art in Hardware Verification/evaluation

Simulation Based Testing



Model Checking



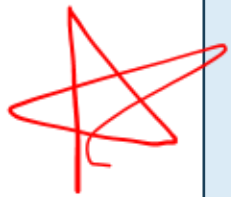
Formal

Hardware Is Not Perfect!



** Side channel attack*

- Correct architectural behavior is not the whole story
- Attacks can exploit side effects of implementation details
- Timing ⌚, caches 🗄, power ⚡, and ⌞ speculation can leak information



Hardware is both a security mechanism and a potential source of vulnerabilities!