CSE/CEN 598
Hardware Security & Trust

Trusted Digital System Design: Hardware Trojans

Prof. Michel A. Kinsy

Hardware Trojans

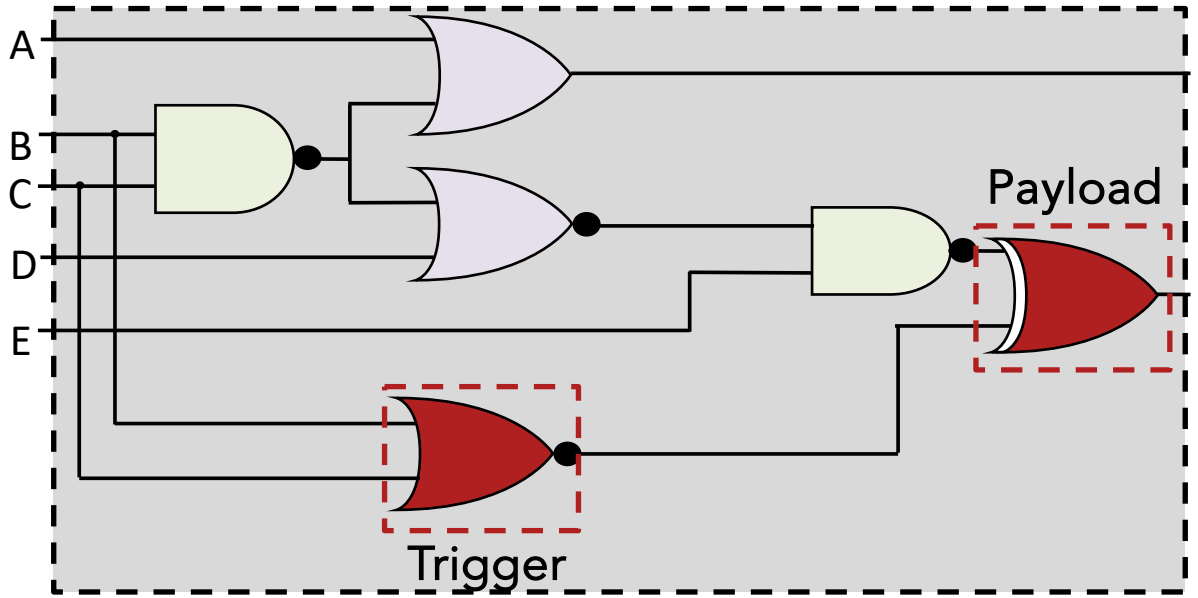
- A Hardware Trojan (HT) is a malicious insertion or alteration to an integrated circuit design
  - Possible effects ranging from leakage of sensitive information to the complete destruction of the circuit itself
- Main system targets
  - Military systems
  - Critical Infrastructures
    - Power Grids, Nuclear Power Plants, etc.
  - Transportation and Telecommunication
    - Trains, Satellite, etc.
  - Banking Systems

Hardware Trojans

- Illustrative Example
  - Cryptographic capability of the processor was compromised
  - To reduce the entropy of the random number generator from 128 bits to just 32 bits
  - Engineered by changing the doping polarity of a few transistors
  - Undetectable by built-in self-test and physical inspections

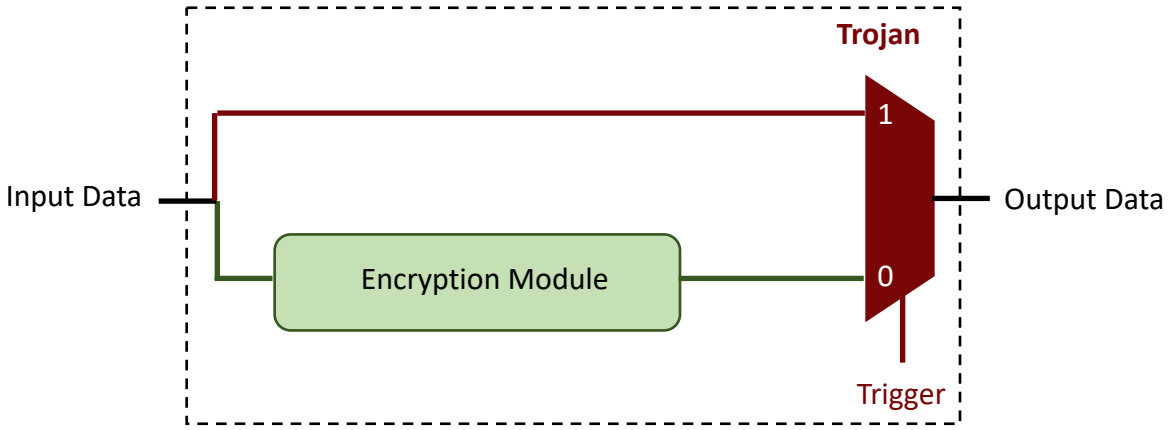
Hardware Trojans

- Malicious changes to a design
- These changes be inserted at any stage of the design and manufacturing process
  - Specification stage, RTL, manufacturing, supply chain
- Often there are two components, a trigger and a payload

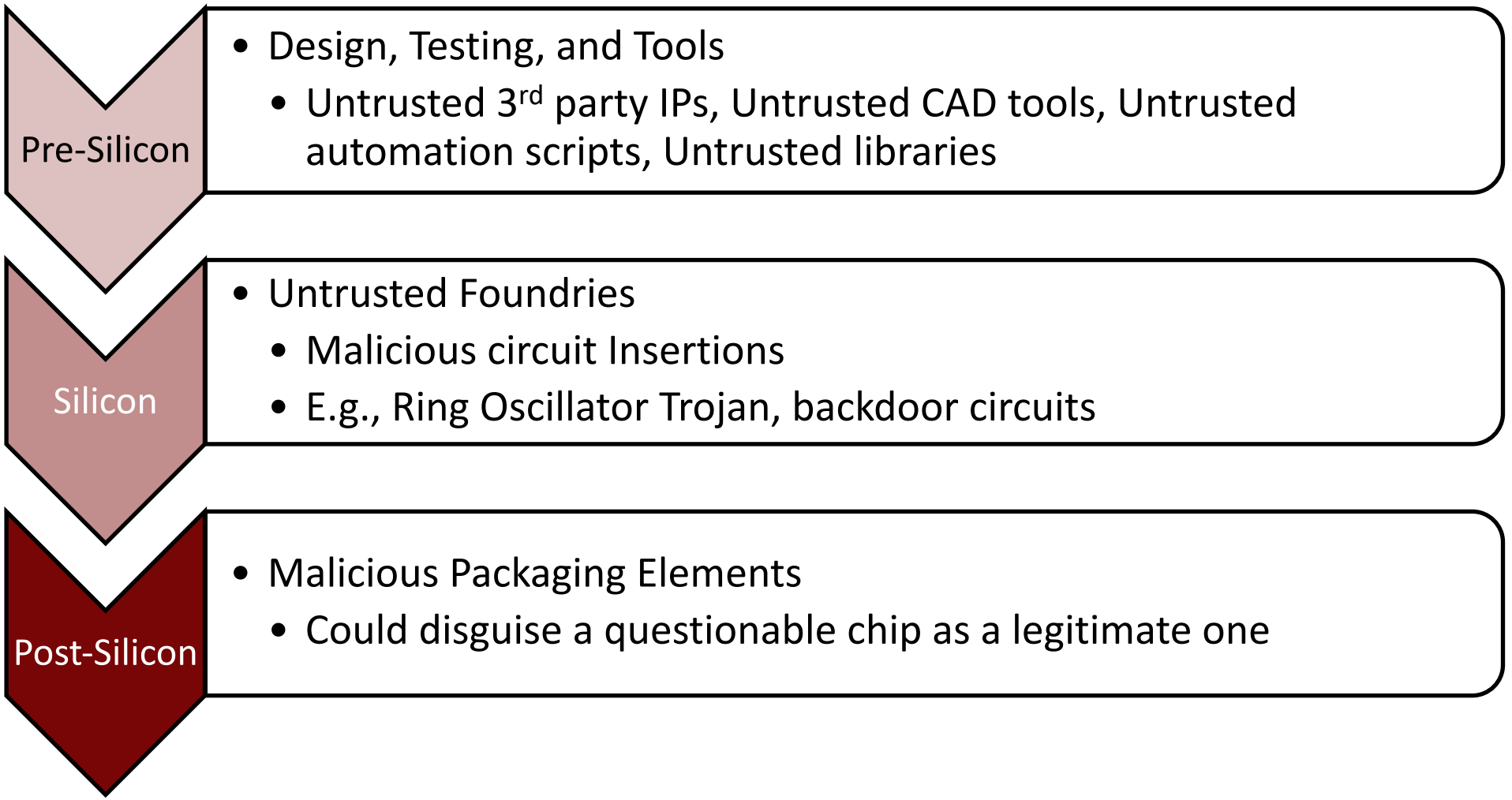


Hardware Trojans

- Malicious changes to a design
- These changes be inserted at any stage of the design and manufacturing process
  - Specification stage, RTL, manufacturing, supply chain
- Often there are two components, a trigger and a payload
  - Low possibility of occurrence
  - Very small hardware overhead
  - Extremely hard to detect

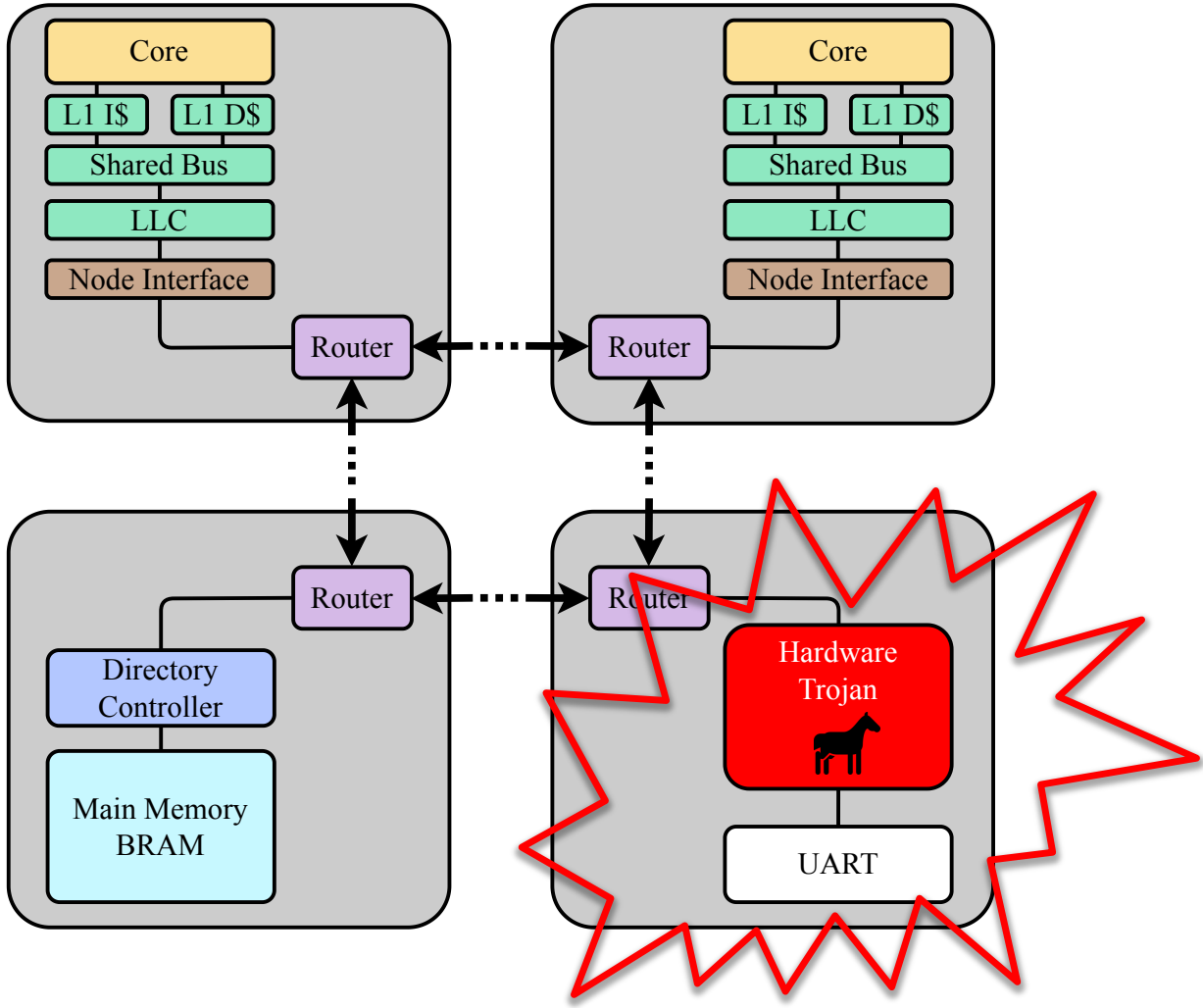


Design Flow Process & Vulnerabilities



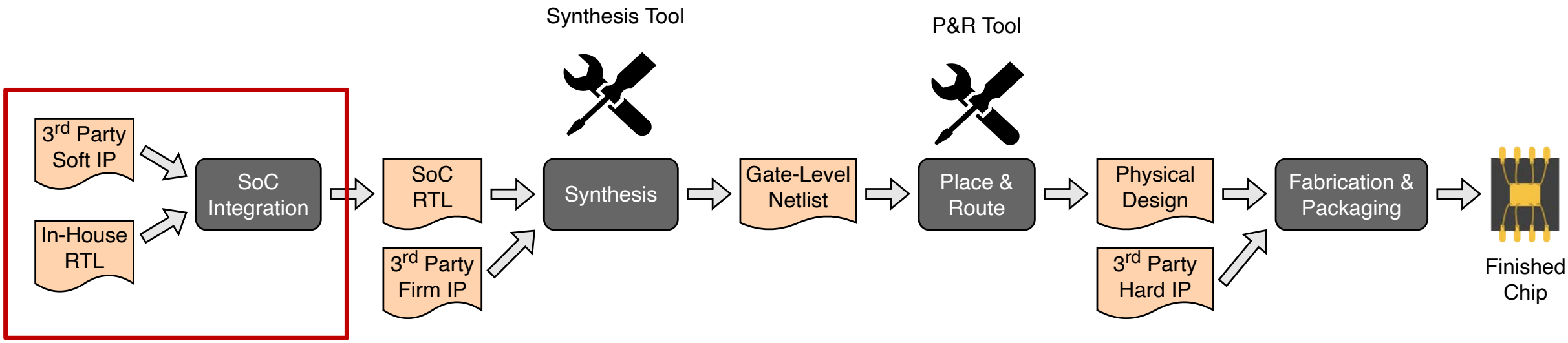
Hardware Trojans

- Extra circuitry added to specified design
  - To cause malfunction
  - To steal secret information
  - To create backdoor for attack
- HT has two distinctive parts:
  - Trigger
    - To activates the HT
  - Payload
    - For the delivery of the malicious effect
- Malicious behavior
  - Leak information
  - Degrade performance
  - Violate specifications
- Complexity of ICs make them hard to detect



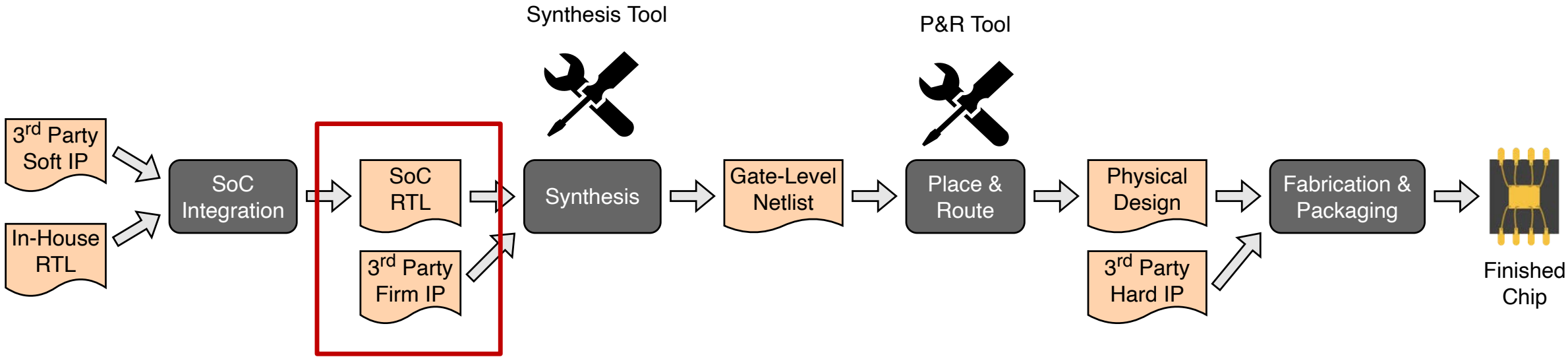
Integrated Circuit Design Flow

- Source-level RTL
  - In-house and 3rd party



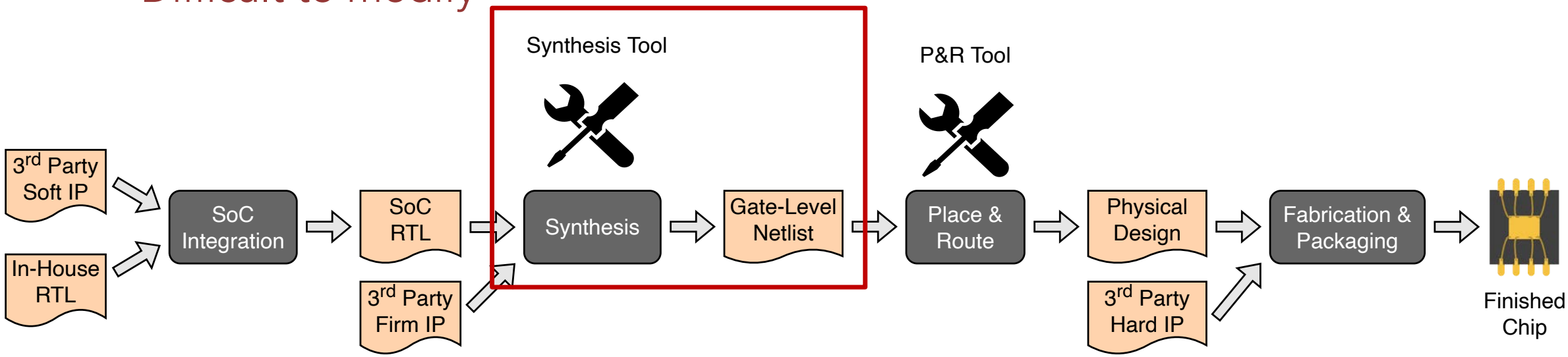
Integrated Circuit Design Flow

- Source-level RTL
  - In-house and 3rd party
- "Firm" RTL
  - Synthesized Verilog netlists
  - Difficult to modify



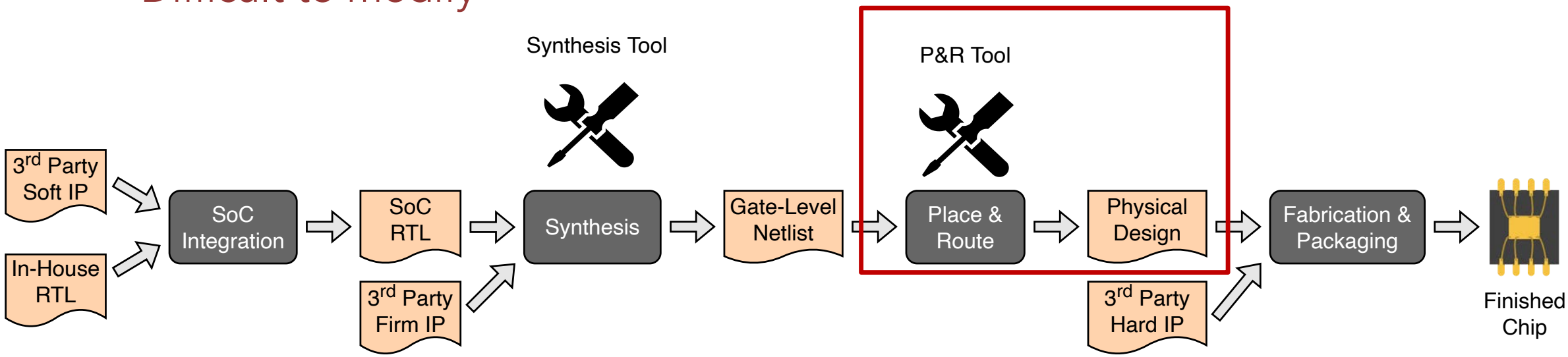
Integrated Circuit Design Flow

- Source-level RTL
  - In-house and 3rd party
- "Firm" RTL
  - Synthesized Verilog netlists
  - Difficult to modify
- Gate-Level Netlists



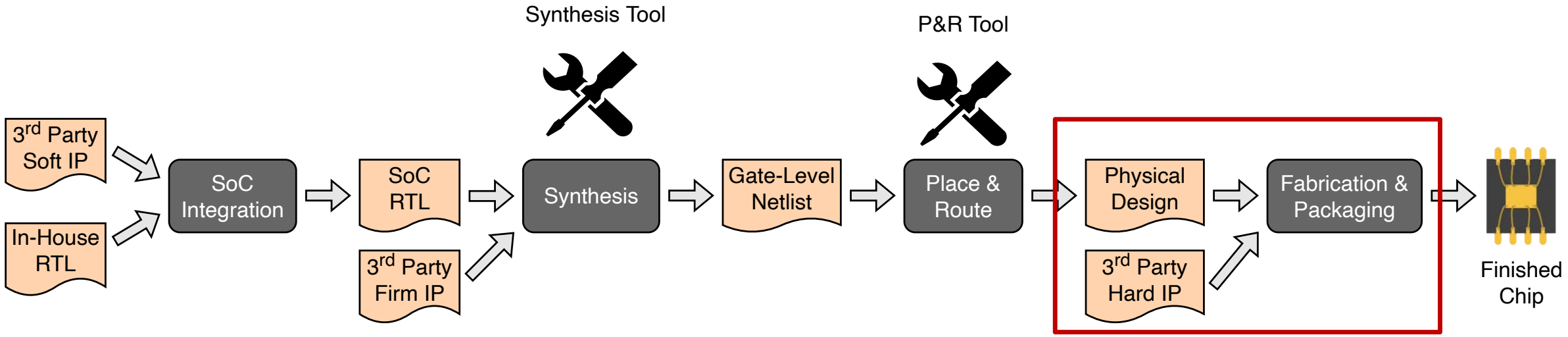
Integrated Circuit Design Flow

- Source-level RTL
  - In-house and 3rd party
- "Firm" RTL
  - Synthesized Verilog netlists
  - Difficult to modify
- Gate-Level Netlists
- Physical Designs
  - Hard 3rd Party IP

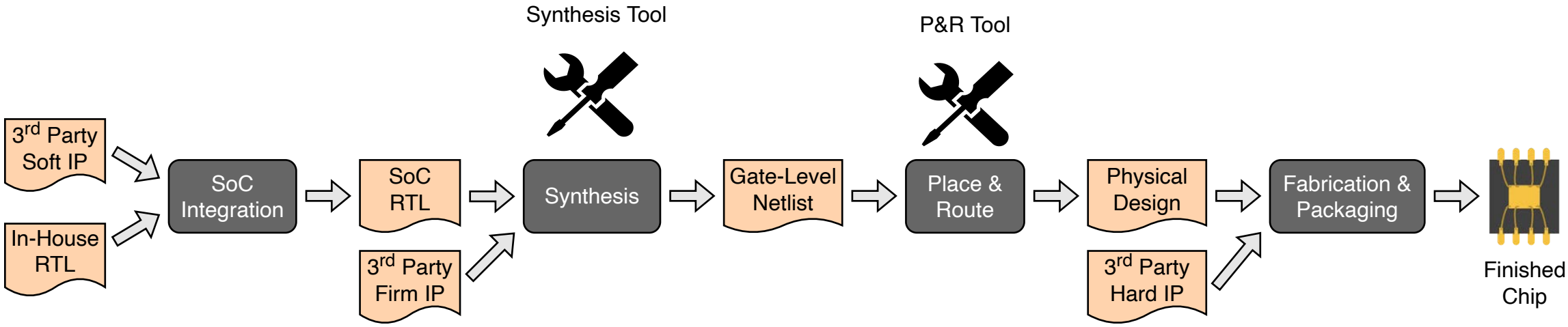


Integrated Circuit Design Flow

- Source-level RTL
  - In-house and 3rd party
- "Firm" RTL
  - Synthesized Verilog netlists
  - Difficult to modify
- Gate-Level Netlists
- Physical Designs
  - Hard 3rd Party IP
- Packaging

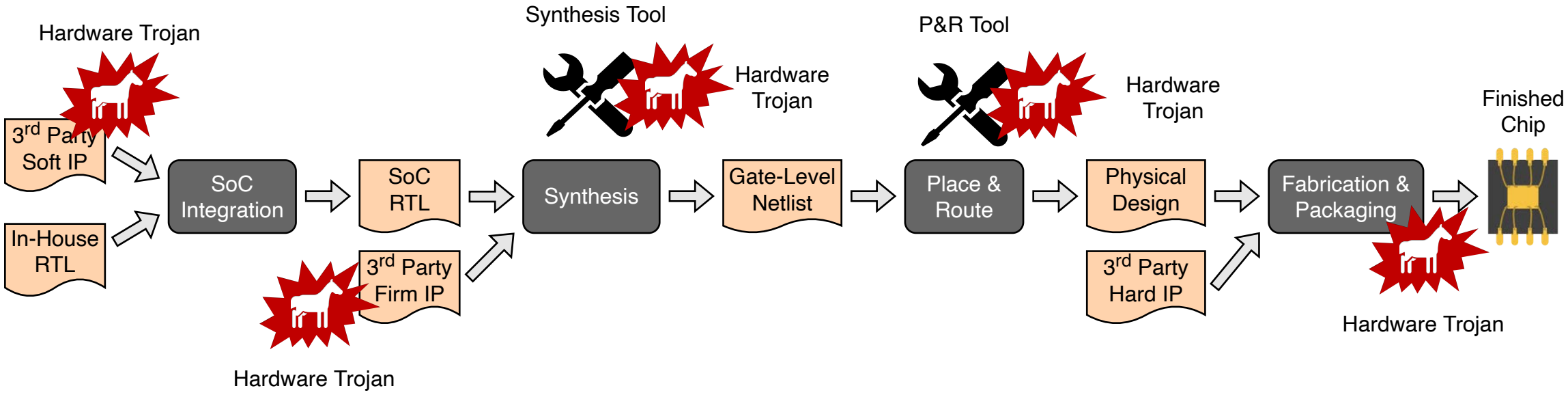


Where Can Trojans be Added?



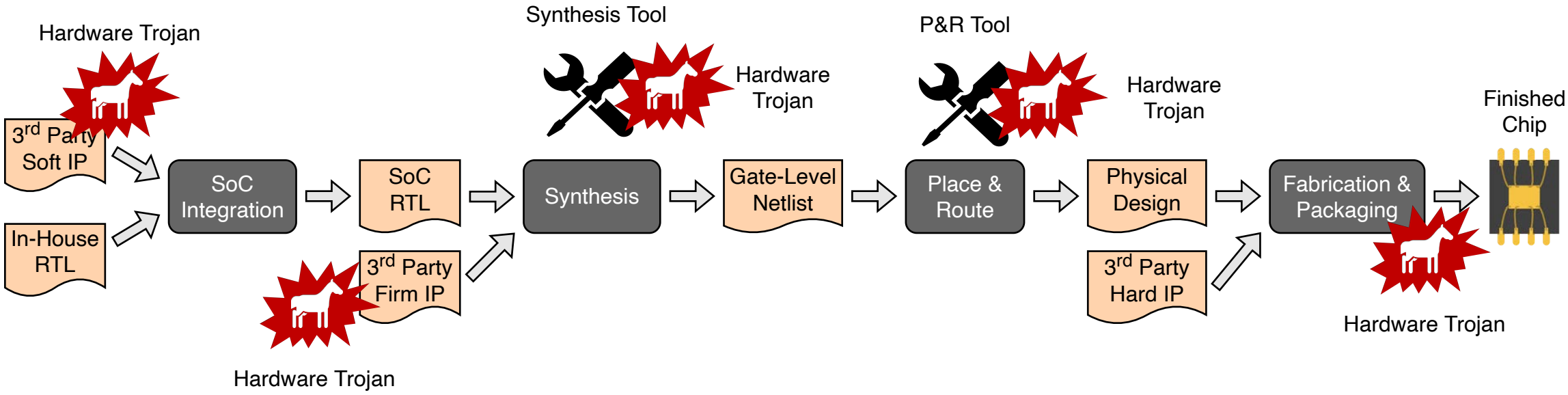
Where Can Trojans be Added?

- Lots of places!



Where Can Trojans be Added?

- Lots of places!
- Soft, firm RTL
- Hard IP
- CAD Tools
- Packaging/Chiplets



Soft IP Trojans

- Written directly in RTL
  - 3rd Party IP
  - Insider threats
- Undocumented functionality
  - Hard to spot in large projects
  - Code reviews are tedious & expensive
- Still hard to detect with testing/simulation

```
13 always@(posedge clock) begin
14   if(read)
15     rd_data <= ram[address];
16   if(write)
17     ram[address] <= wr_data;
18 end
19
20 hardware_trojan(
21   clock,
22   read,
23   write,
24   address,
25   wr_data,
26   rd_data
27 );
28
```

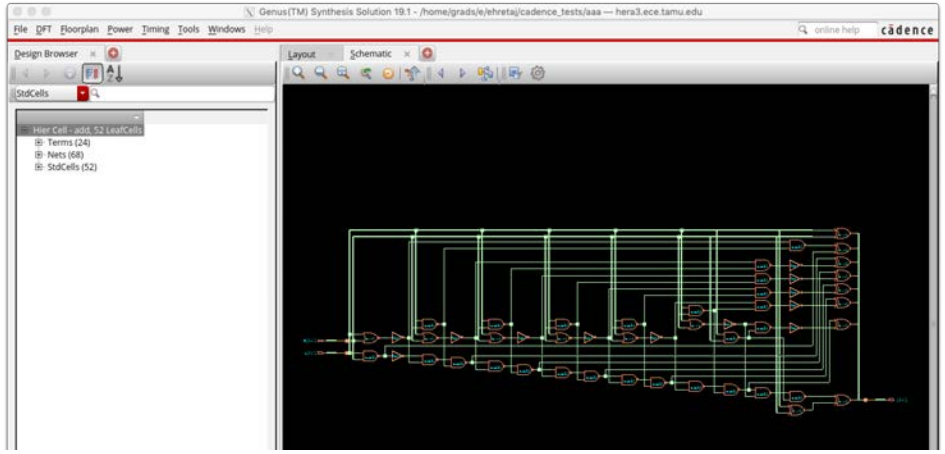
Firm 3rd Party IP Trojans

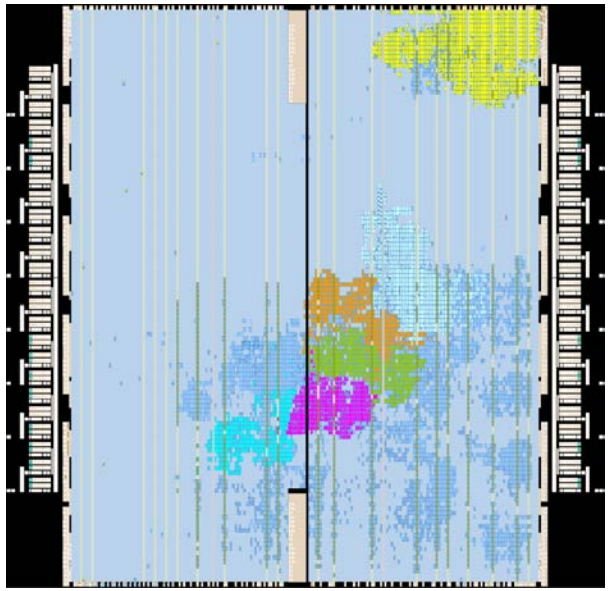
- Inserted into synthesized netlists
  - From source-level RTL
  - Manually at netlist
- Potentially obfuscated
  - Hard to reverse engineer

```
assign _19_ = _48_(8'h00, { _29_[7:0],
assign _22_ = _23_ ?  _19_ : 8'h00;
assign _24_ = _25_ ?  8'h00 : _22_;
assign _26_ = _27_ ?  8'h00 : _24_;
assign _28_ = |TX89tb;
UXmg  CJTDIIx (
   .ET5uS(pRm6vGPowaWGO),
   .Jmv(m6Powv),
   .Kogc(1'h0),
   .X8qqE(GooRagLjgP4dw),
   .ffkGpLq4E(MKHJv93PQOdt0rllG),
   .jT1j(TX89tb),
   .k7ZJ(UMIaxeJUmGYR),
   .kfno(ewTNYHIdEpVjO),
   .sMrlR(NXKSJycPvnPOa),
   .vgGEfgslfn(Zl6sD57Nd43NzyGl9f)
);
```

Malicious CAD Tool Trojans

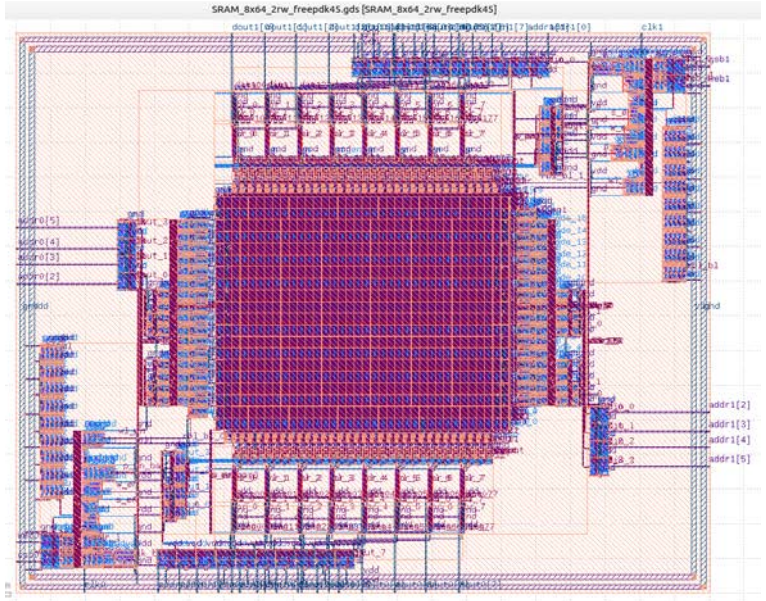
- CAD Tools are complex
- Difficult to reason about logic optimizations
  - Where is each gate/register in the hardware descriptor language?
  - Difficult to know for each
- Possible to insert additional logic undetected
  - Limits to trojan payload or triggers
  - Tool must understand design enough to place trojan

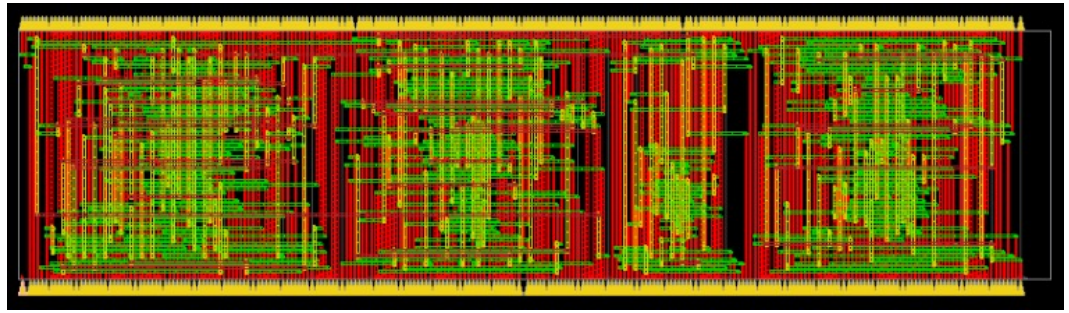




Hard IP & Foundry Trojans

- IP blocks received as VLSI black box
  - Complete physical design IP
- Foundry may edit circuit design
  - Add wires to mask
  - Insert additional standard cells
- Difficult but not impossible
  - Typically requires reverse engineering physical design





Hardware Trojan Structure

Each hardware trojan has two parts

- The Payload - Determines what the trojan does
  - Denial of Service
  - Information Leakage
  - Performance Degradation

Hardware Trojan Structure

Each hardware trojan has two parts

- The Payload - Determines what the trojan does
  - Denial of Service
  - Information Leakage
  - Performance Degradation
- The trigger – How is the payload activated
  - Always active
  - Internal Triggers
    - Software controlled trigger
    - Time delay
  - External trigger
    - Environmental factors

Hardware Trojan Payload - Goals

Leak Information

- Transmit secret keys
- Exfiltrate computed data
- Send message to attacker

Hardware Trojan Payload - Goals

Leak Information

- Transmit secret keys
- Exfiltrate computed data
- Send message to attacker

Denial of Service

- Disable clock
- FSM "Sink State"
- Short-circuit Vcc and GND

Hardware Trojan Payload - Goals

Leak Information

- Transmit secret keys
- Exfiltrate computed data
- Send message to attacker

Denial of Service

- Disable clock
- FSM "Sink State"
- Short-circuit Vcc and GND

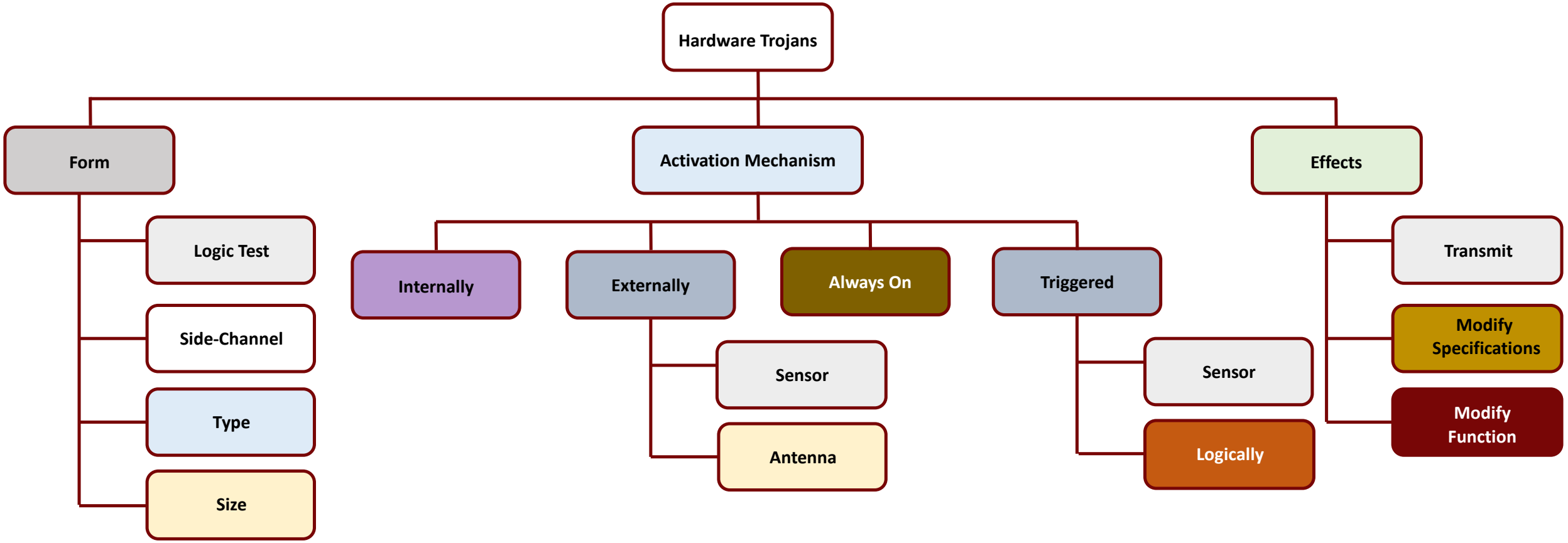
Corrupt State

- Elevate privileges
- Degrade performance

Hardware Trojan Classifications

	Trigger		Actors		
Types of Trojan	**Actor**	**Action**	**Input Channel**	**Output/Leaking channel**	**Payload / Consequence of attack**
Trigger Activated	Attacker with physical access to the device	▪ Particular legitimate input sequence ▪ Particular illegitimate input sequence	Standard Input ▪ I/O pins ▪ Keyboard ▪ Serial/Parallel protocols	Standard / Unused Outputs ▪ I/O pins ▪ LCD ▪ LEDs ▪ Serial/Parallel protocols Side Channels ▪ EM Waves ▪ Hidden in standard output	Leaking sensitive information ▪ Encryption Key ▪ Plain text Denial of service ▪ Generating incorrect results ▪ Make the device stop working Reduce the reliability of the device ▪ Drain the battery
		▪ Taking control through unused functional units or interfaces	Unused Inputs ▪ I/O pins ▪ Serial/Parallel protocols		
	Legitimate User	▪ Normal operation for certain n>N ▪ Particular legitimate input sequence ▪ Illegitimate input sequence by mistake ▪ Certain time interval between two legal inputs	Standard Input ▪ I/O pins ▪ Keyboard ▪ Serial/Parallel protocols		
Always Active	N/A	N/A	Internal IP Core	Side Channels ▪ EM Waves	Leak the Encryption Key

Hardware Trojan Classifications



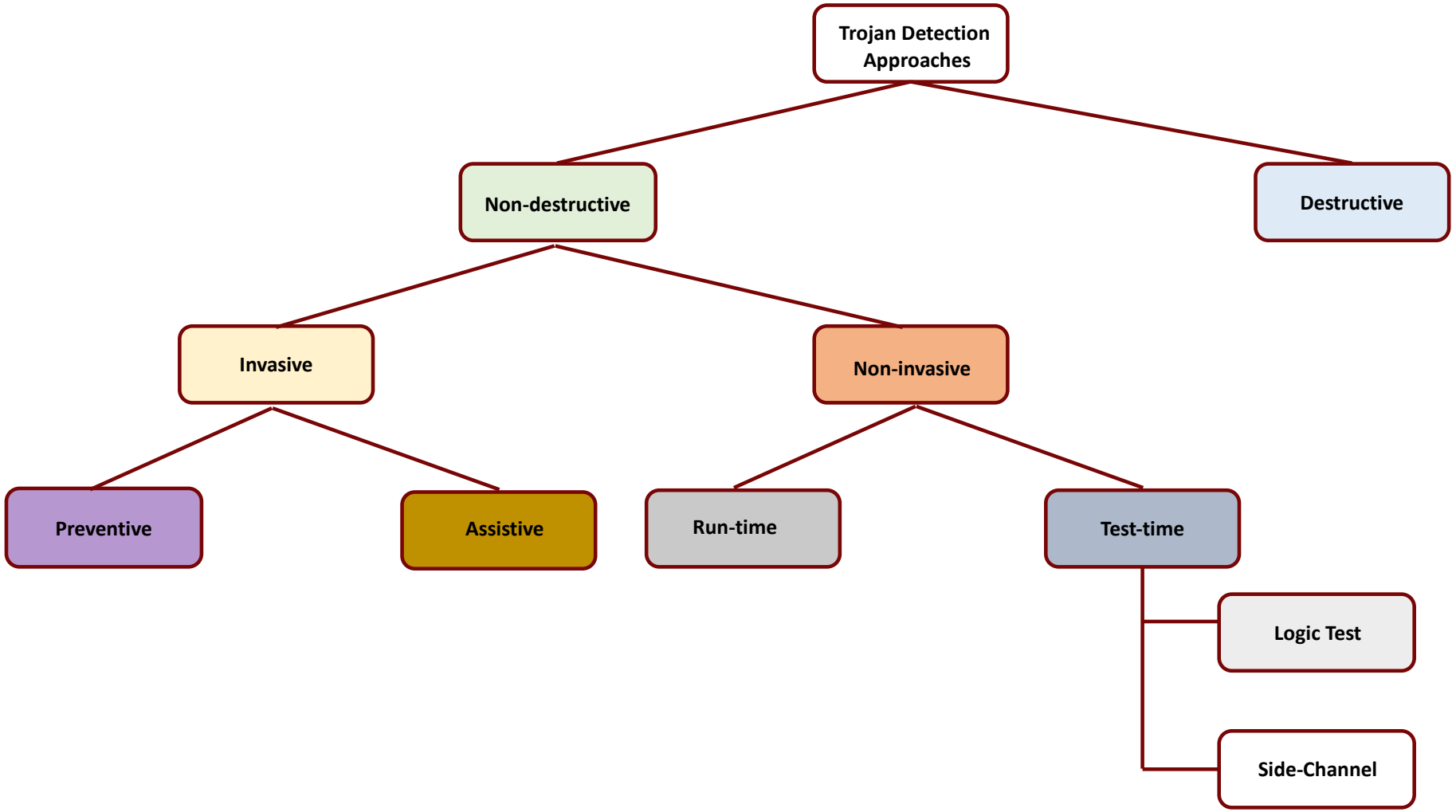
Hardware Trojan Summary

- Modification of Functional Logic
  - Third party corrupt IPs
  - Tampering with clocks
  - Tampering with voltage control logic
- Modifications of the Layout
  - Thinning of the conducting wires
  - Weakening of the transistors
  - Using spare gates
- Modify and Exploit Operating Conditions
  - Temperature
  - Power
  - Frequency

Hardware Trojan Detection Challenges

- Detecting a hardware trojan requires overcoming numerous challenges
  - Handling a large number of designs
  - Being non-destructive to the IC
  - Being cost effective
  - Ability to Detect trojans of different sizes or complexities
  - Authenticating chips in as small a time frame as possible
  - Robust to variations in manufacturing processes
  - Among others
- Current Approaches
  - Lack of general detection techniques or frameworks
  - Most techniques cannot guarantee detection
- Test time is expensive
- Trojan are designed to be stealthy

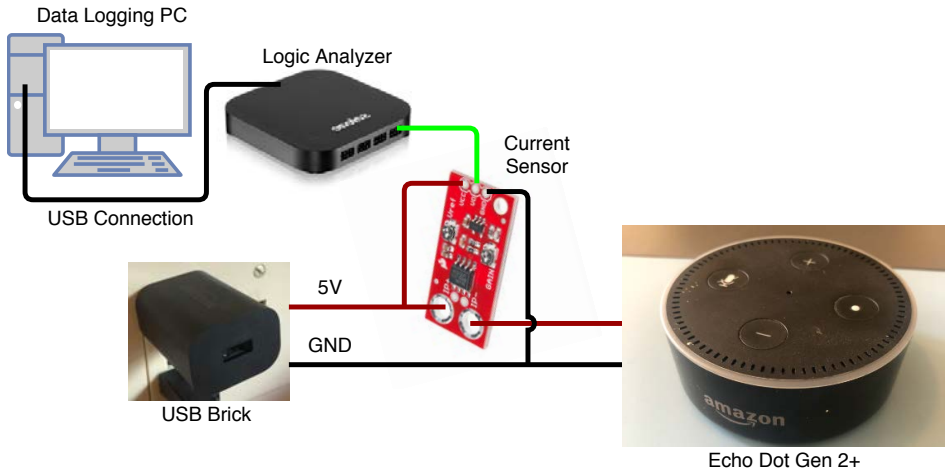
Hardware Trojan Detection Challenges

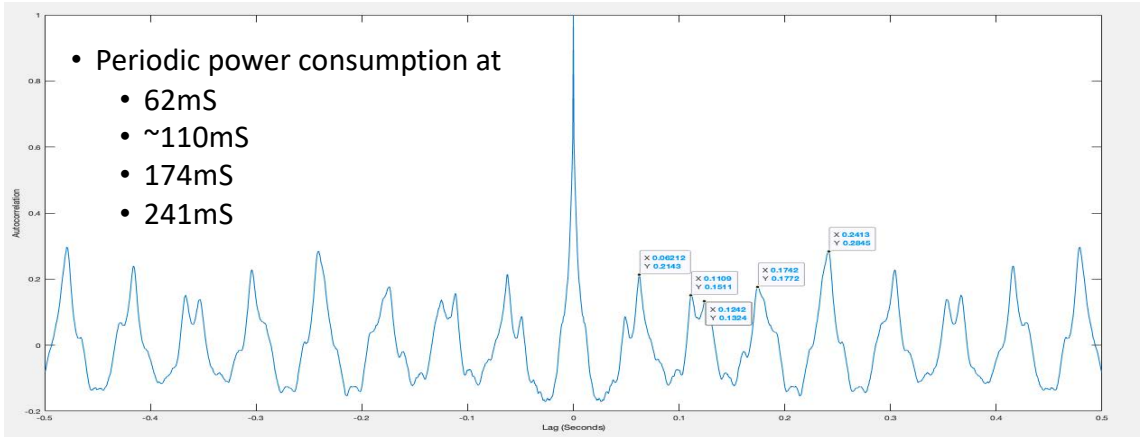


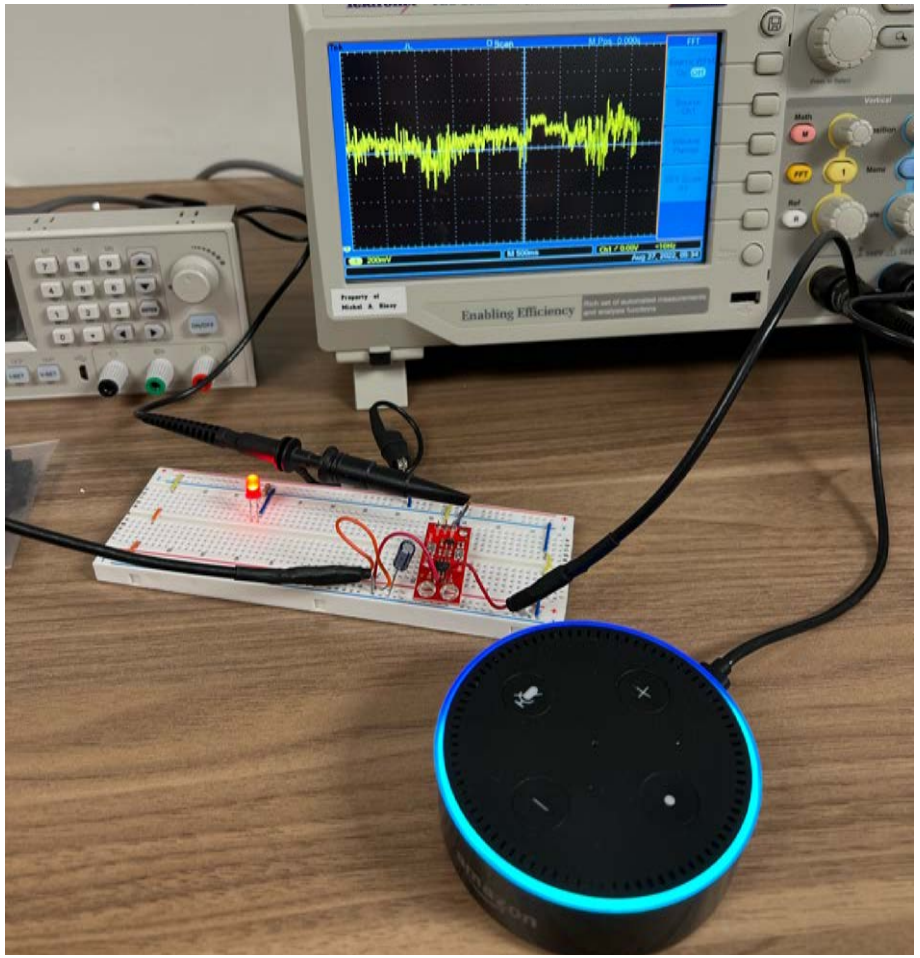
Direct I/O Access

- Direct I/O Usage
  - Obeying Protocol
  - Send extra data
- Undocumented I/O functionality
  - Negative clock edges
  - Transmit during “settling” period

Data Exfiltration Techniques – Power Side Channels







EM Side Channel Information Leakage

- Measure electromagnetic radiation radiating from IC
- Each wire in the IC acts as a mini antenna
  - Lots of wires in a chip!
- Circuit could be designed to radiate specific pattern/frequency

Power Side Channel Measurement

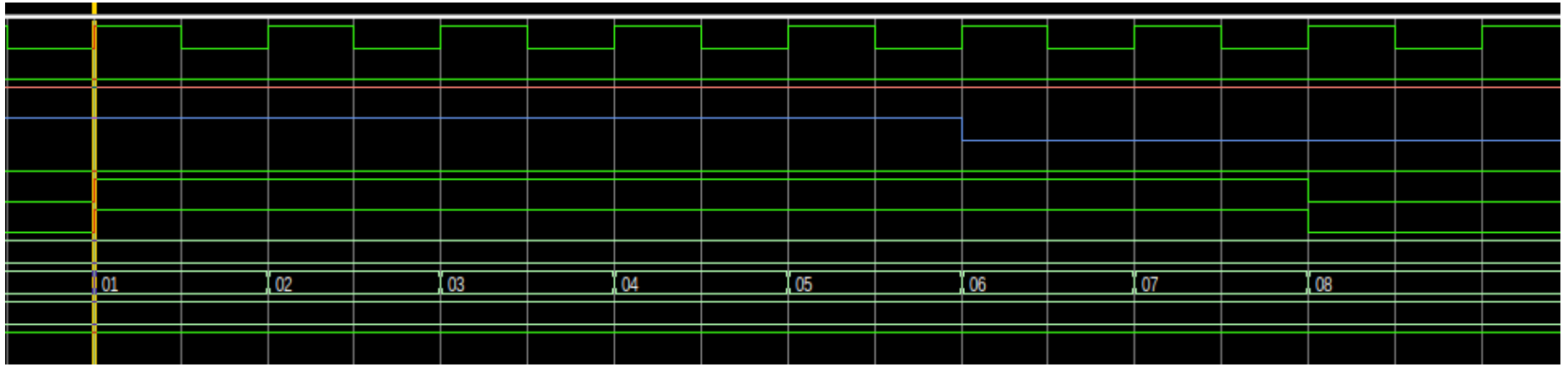
Source: Swarup Bhunia, Mark Tehranipoor , "Hardware Security: A Hands on Learning Approach"

Hardware Trojan Triggers

- Trigger selection
  - Must not be detected during regular testing
  - Attacker still needs to activate at chosen time
- A good trigger
  - Is controllable by an attacker
  - Has low toggle rate – unlikely to be accidentally triggered

Bad Triggers – Frequent toggles

Good Triggers – Rare toggles

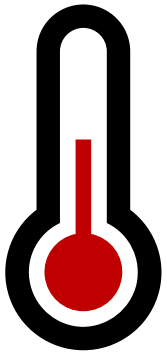


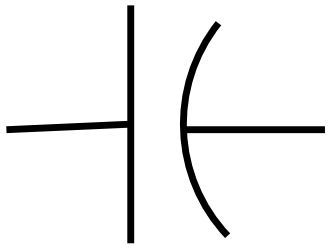
Trigger Examples

- Digital Signals
  - Uncommon flags
    - Divide by 0
  - Secret/uncommon values
    - Perform a specific operation with specific values
- Physical Characteristics
  - Temperature
  - Analog Circuits (Capacitor)
- Constantly Active
  - No trigger, always on

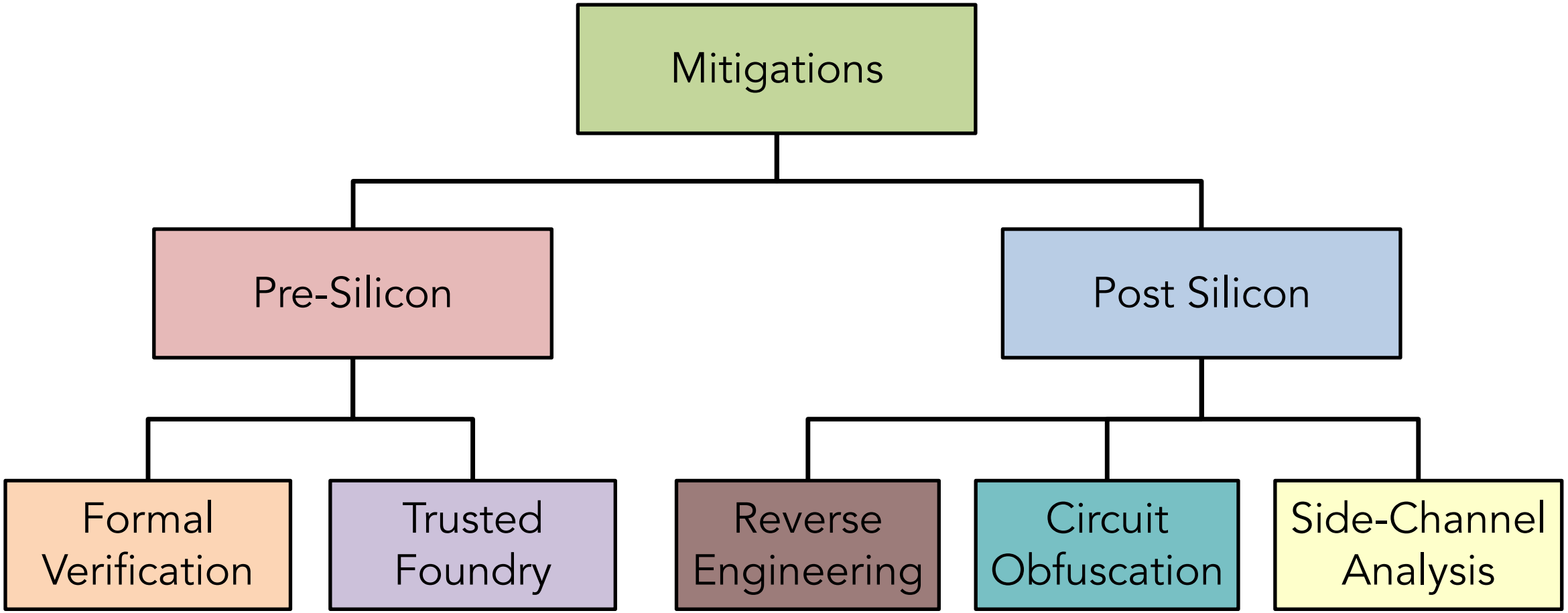
$$\frac{1}{0} = NaN$$

$$0xDEADBEEF$$



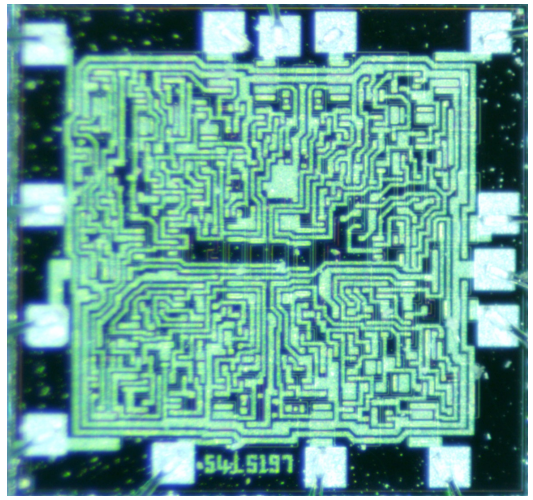


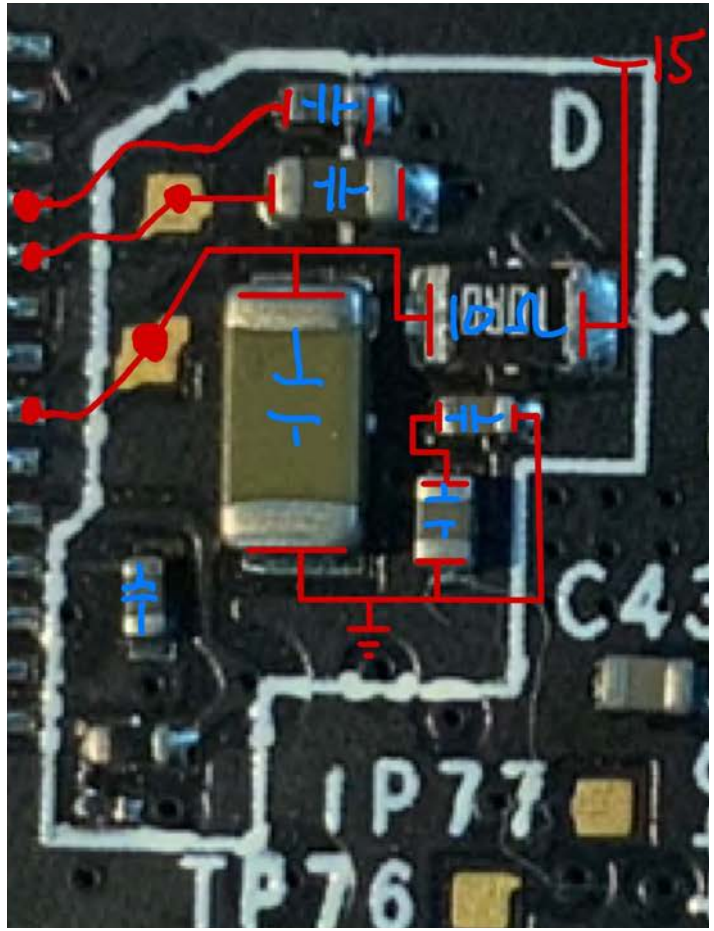
Hardware Trojan Mitigations



Mitigation – Reverse Engineering

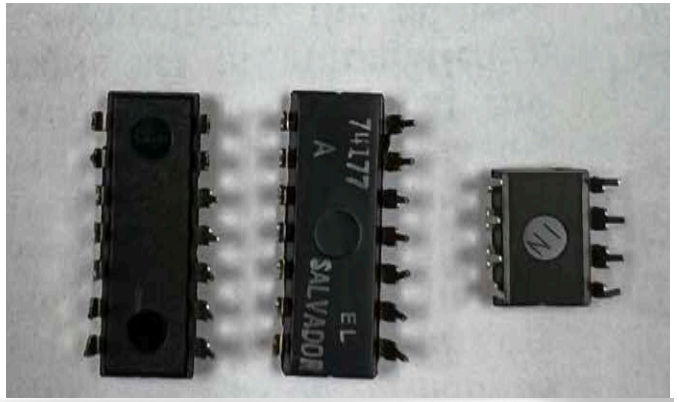
- Visually check that manufactured design matches schematic
  - Useful for PCBs and ICs
- IC reverse engineering
  - X-Rays – next slides
  - Decapsulation - later

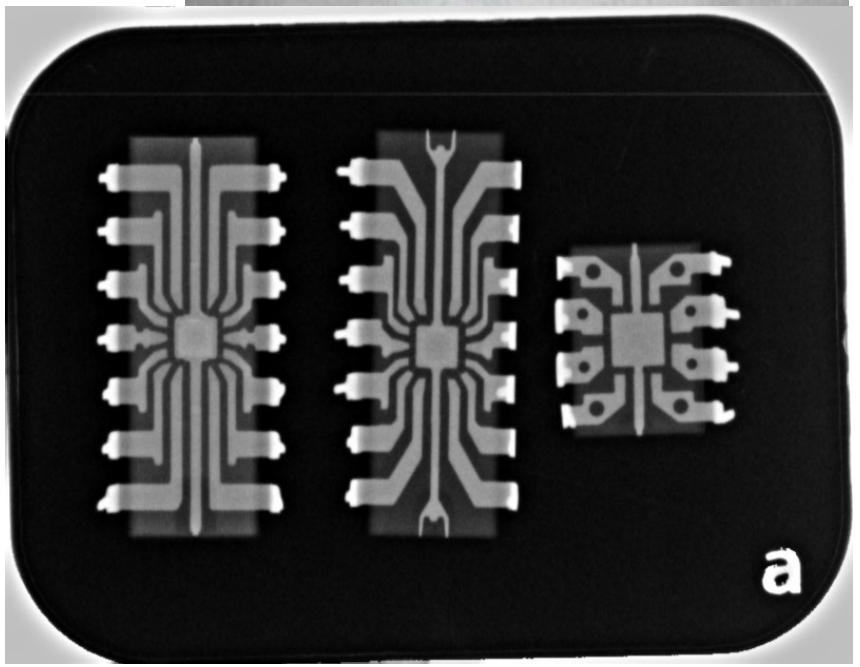




IC X-Ray Reverse Engineering

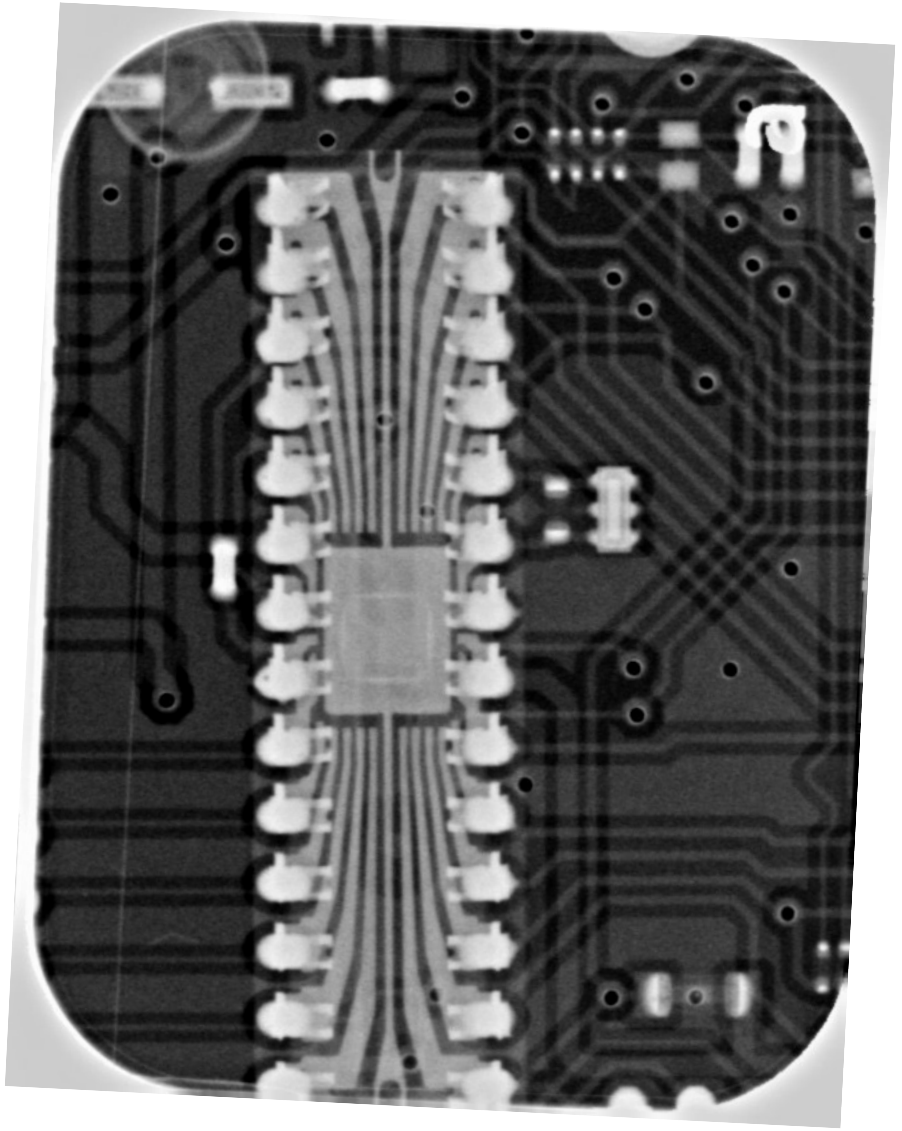
- Commercial PCB/IC X-Ray Inspections machines available
  - Cost $25k or more
  - But a dentist X-Ray machine works too!
- Post-Manufacturing Inspection
  - PCB Traces
  - IC bond wires

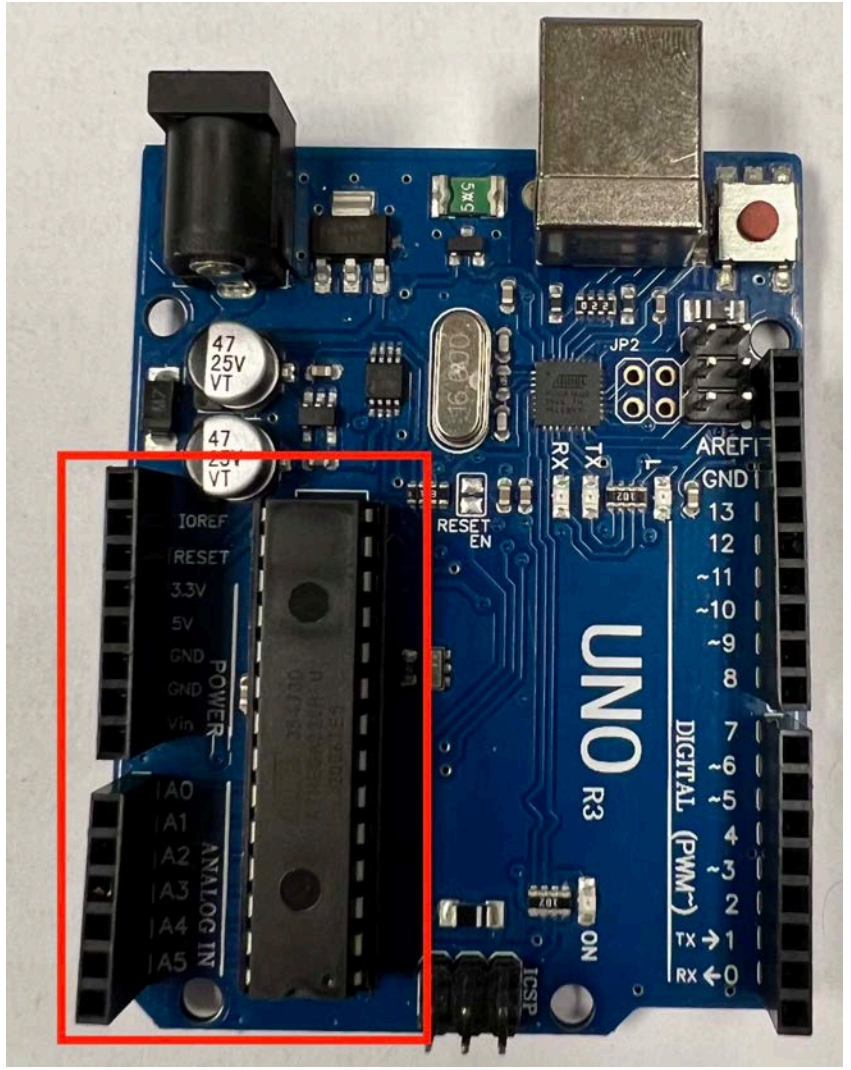




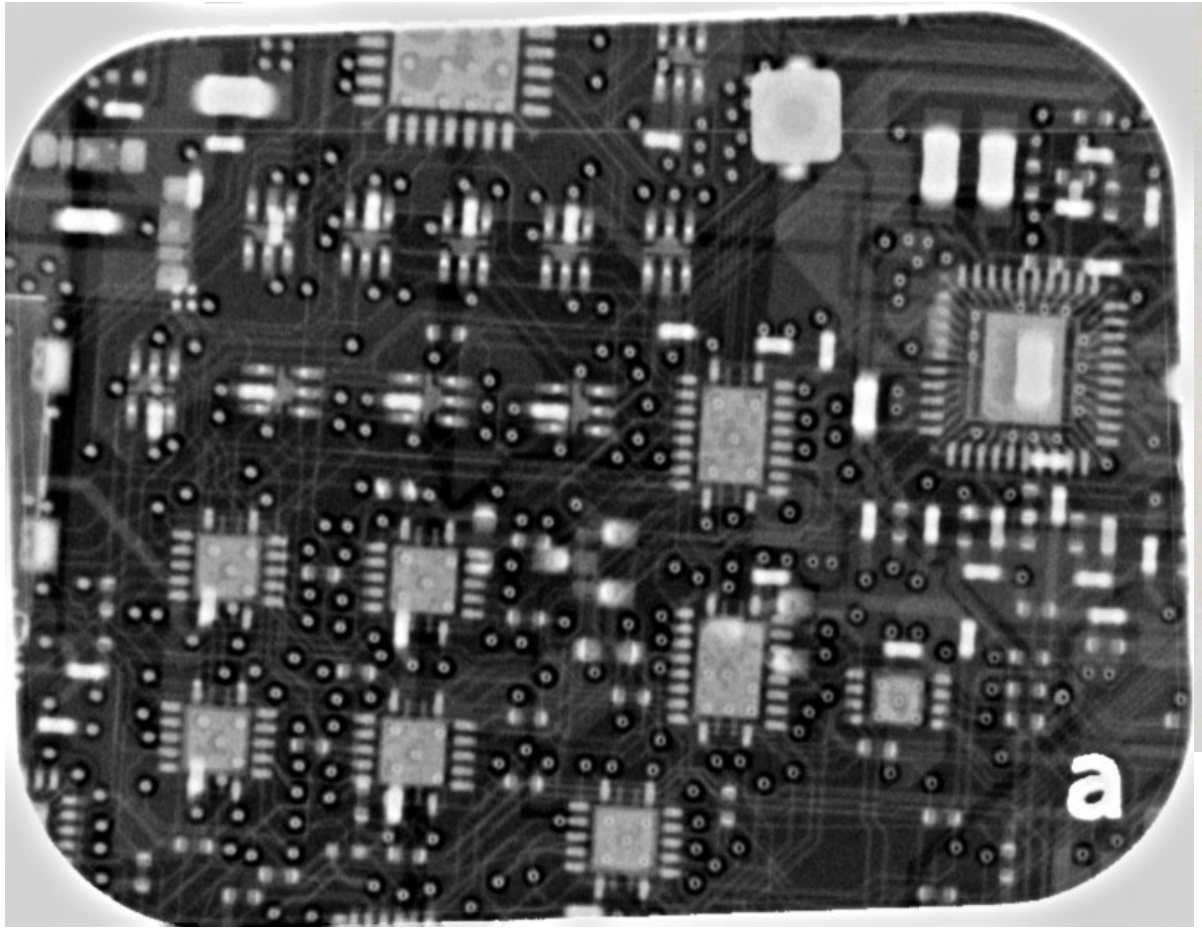
Arduino Uno X-Ray Example

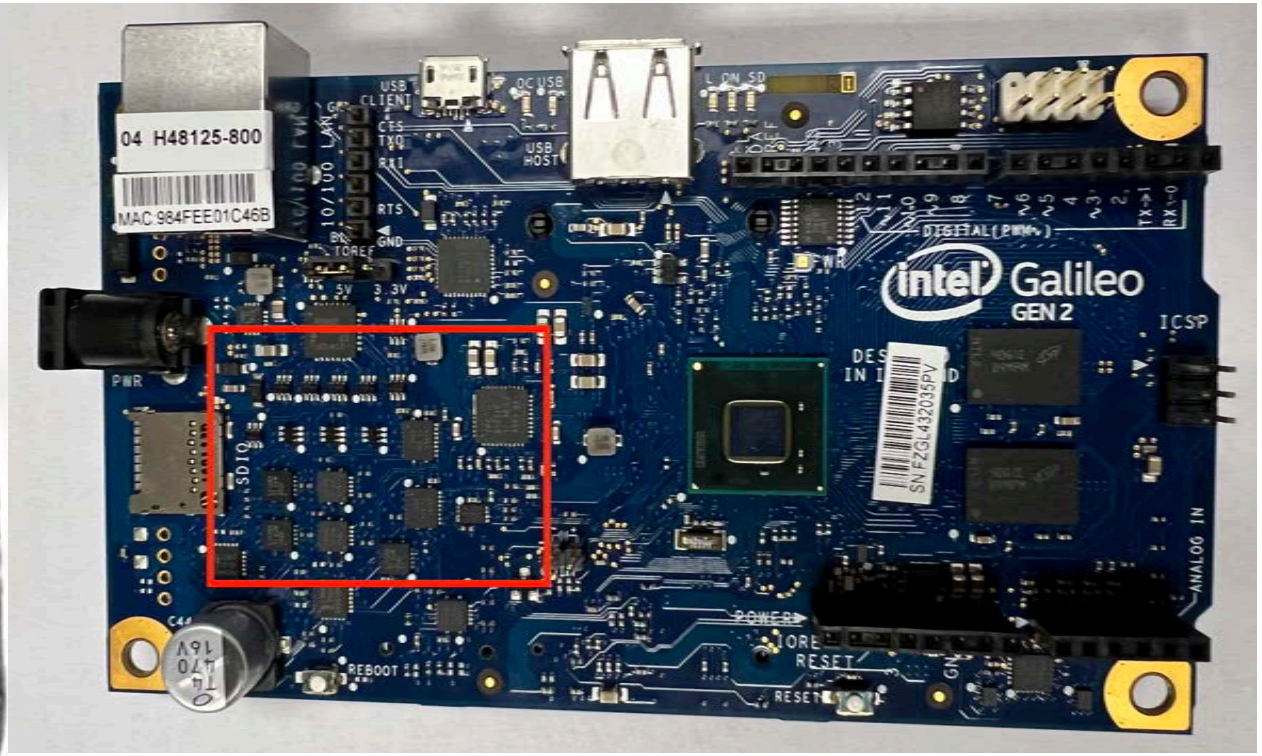
- 2-Layer PCB
- 28-DIP Package
- Misc. discrete components





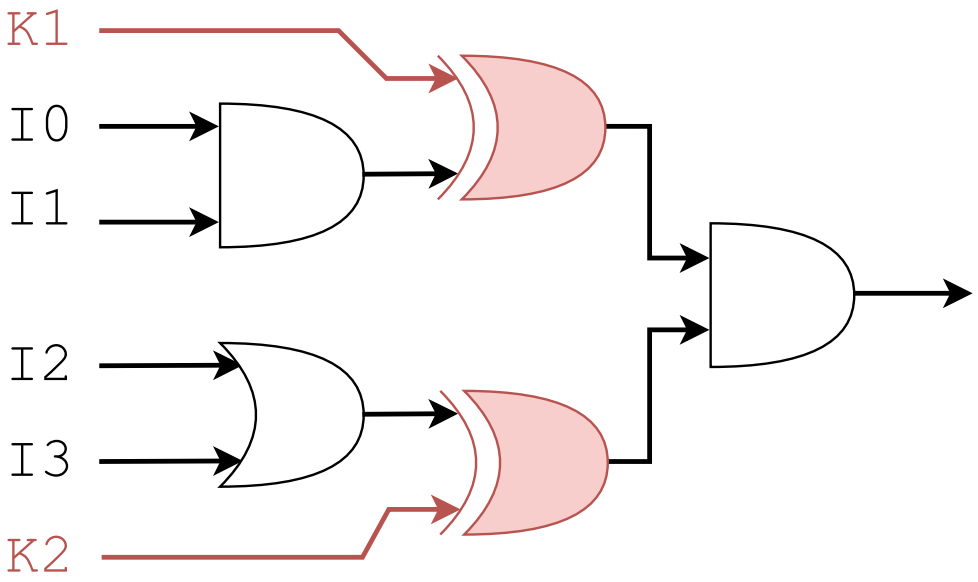
Arduino Galileo X-Ray Example





Circuit Obfuscation

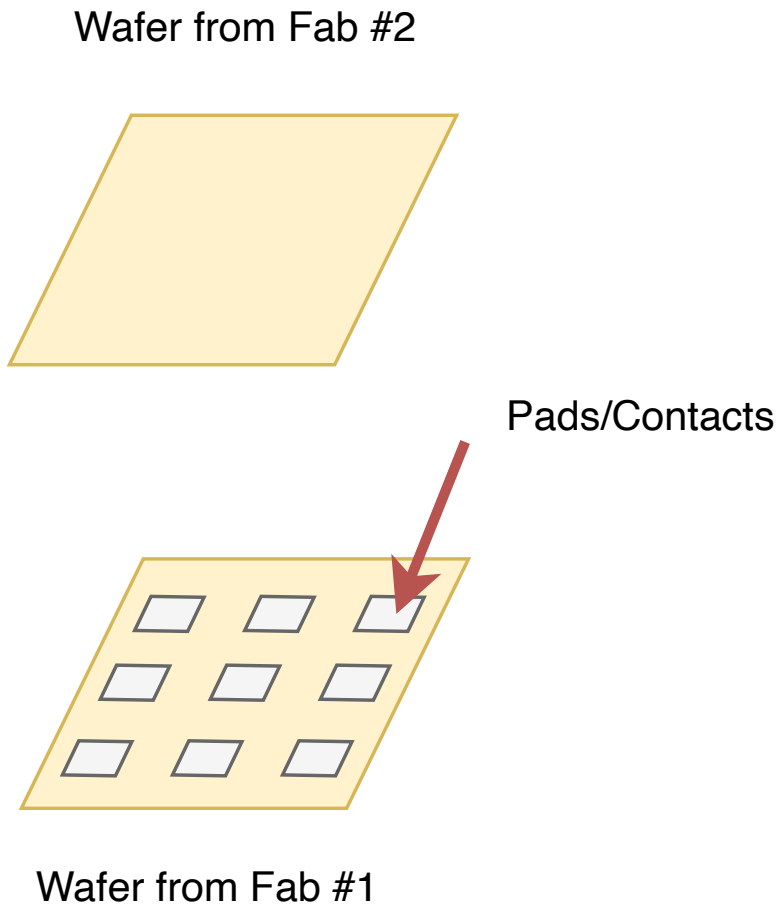
- Hide true purpose of circuit from manufacturer
- Range of options
  - FPGA based implementation – foundry never gets design
  - Xor obfuscation – foundry gets design but not xor key
    - Whole circuit, or just part of circuit



Circuit obfuscation with XOR gates

Split Manufacturing

- Fabricate chip in multiple layers
  - No single foundry has whole design
- One foundry can be "trusted



Trusted Foundry Comparison

MIT Lincoln Laboratory

- Trusted Foundry
  - Assessed integrity of people and processes
- 90 nanometer process
  - 90nm processes first commercialized around 2002

TSMC

- State-of-the-Art Foundry
  - Complex global supply chain
  - Integrity not assured
- 3, 5 nanometer processes
  - 5nm First commercialized in ~2020
    - Still best process commercially available in 2022
  - 3nm starting to ship in 2022

IC Reverse Engineering Overview

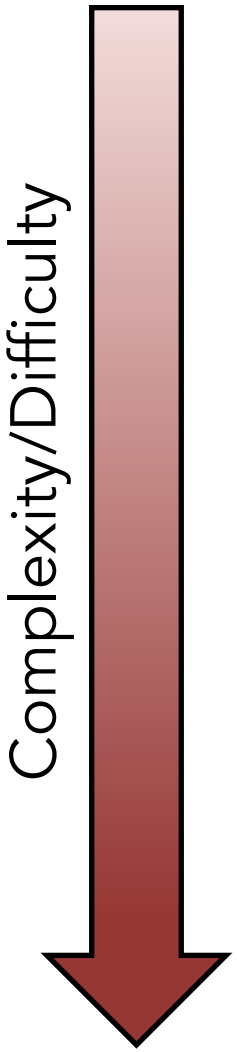
Destructive

- Most Decapsulation
  - Removing the packaging
- Observe circuit layout
  - Etch top layers off with acid
  - Image each layer of chip

Non-Destructive

- X-Rays
  - View packaging internals
  - Assist with destructive reverse engineering
- Some decapsulation methods
  - Possible to operate exposed circuit with open packaging
- Side channel analysis

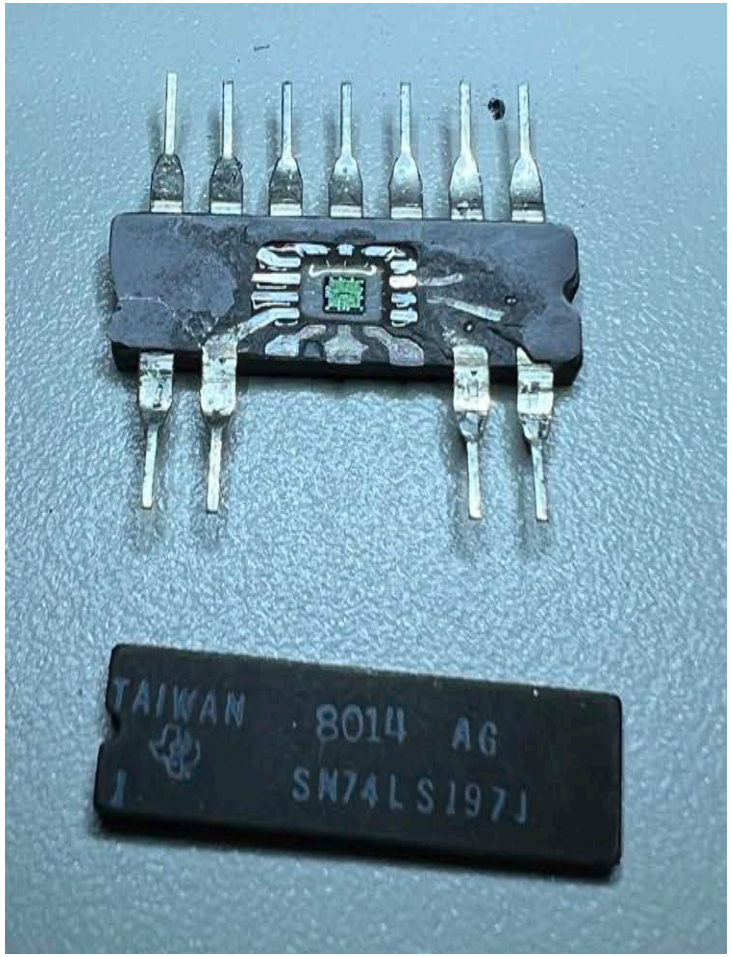
IC Decapsulation Techniques

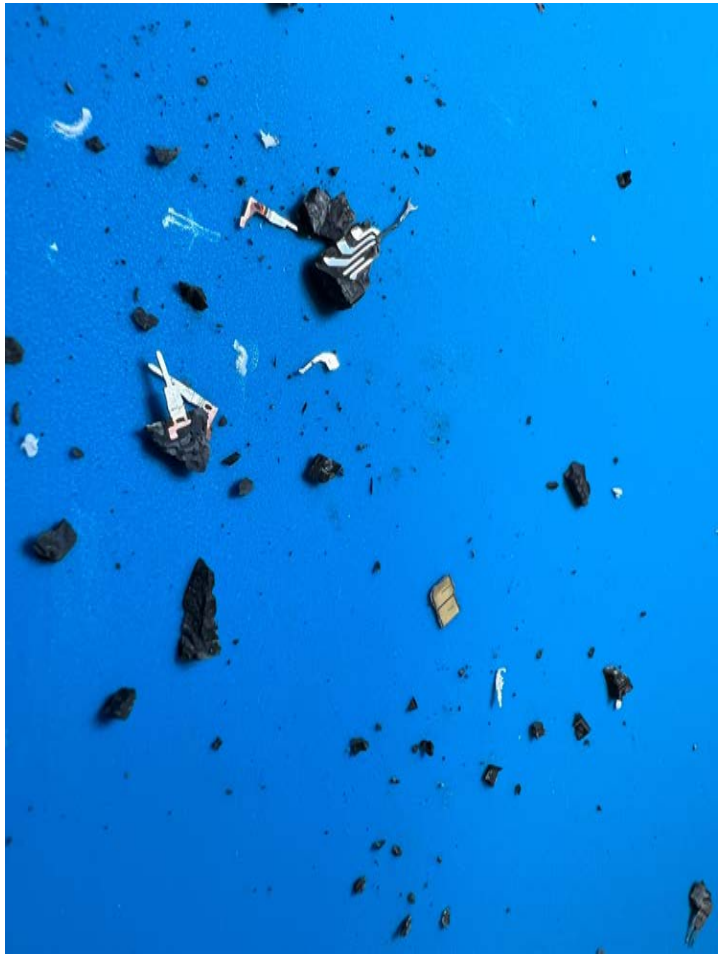


- Hammer & vice method
  - Squeeze ends of DIP package in vice until it cracks
  - Very destructive, but easy to do
- Weak Packaging
  - Carefully crack open certain types of packaging
  - Difficult with modern epoxy
  - Old ceramic packages are easier
- Acid etching
  - Remove epoxy with acid
  - Requires highly concentrated acid
  - Difficult to get acid
  - Lots of PPE/Lab infrastructure required

IC Decapsulation Example – TI 7400 Series

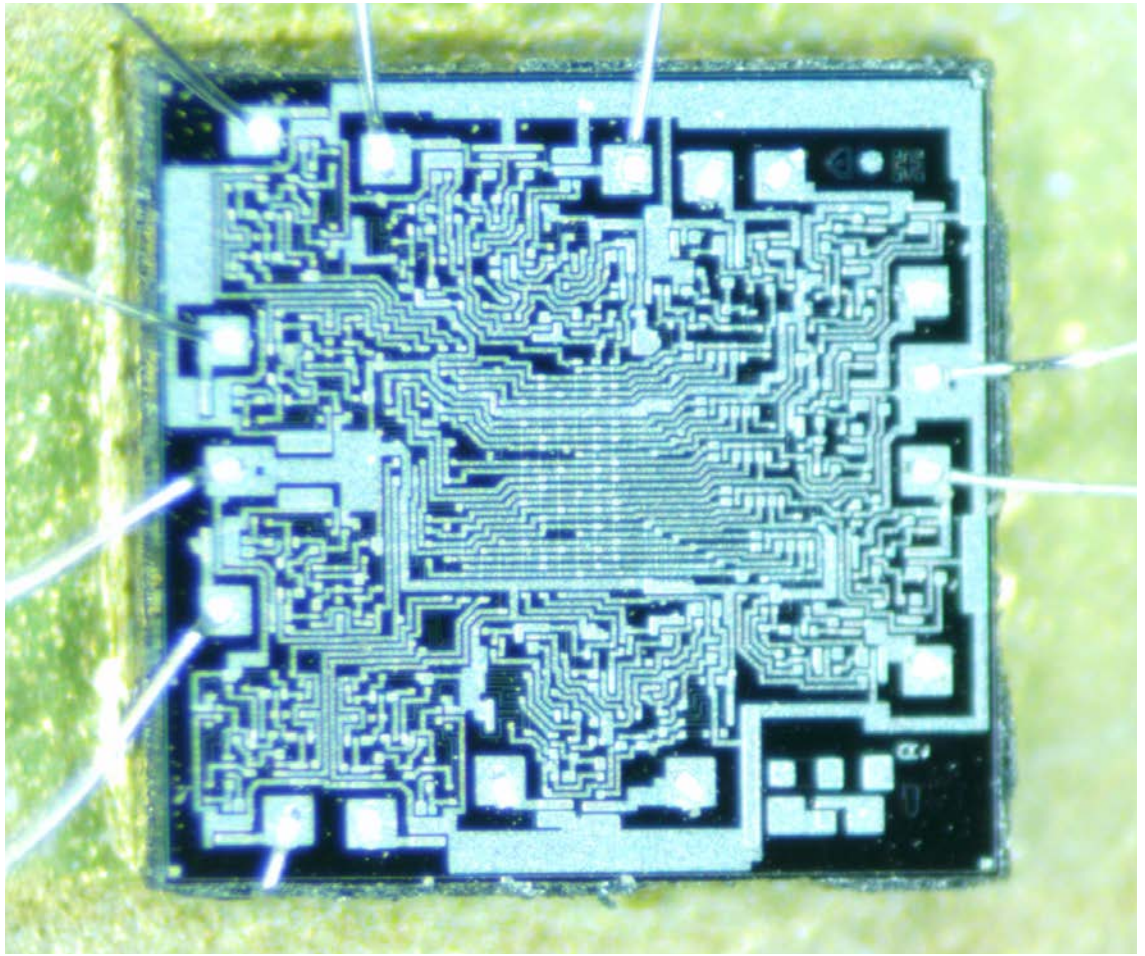
- Old Package with glued top/bottom halves
- Just smashed open with a wrench
  - A Messy process!
  - Not 100% successful



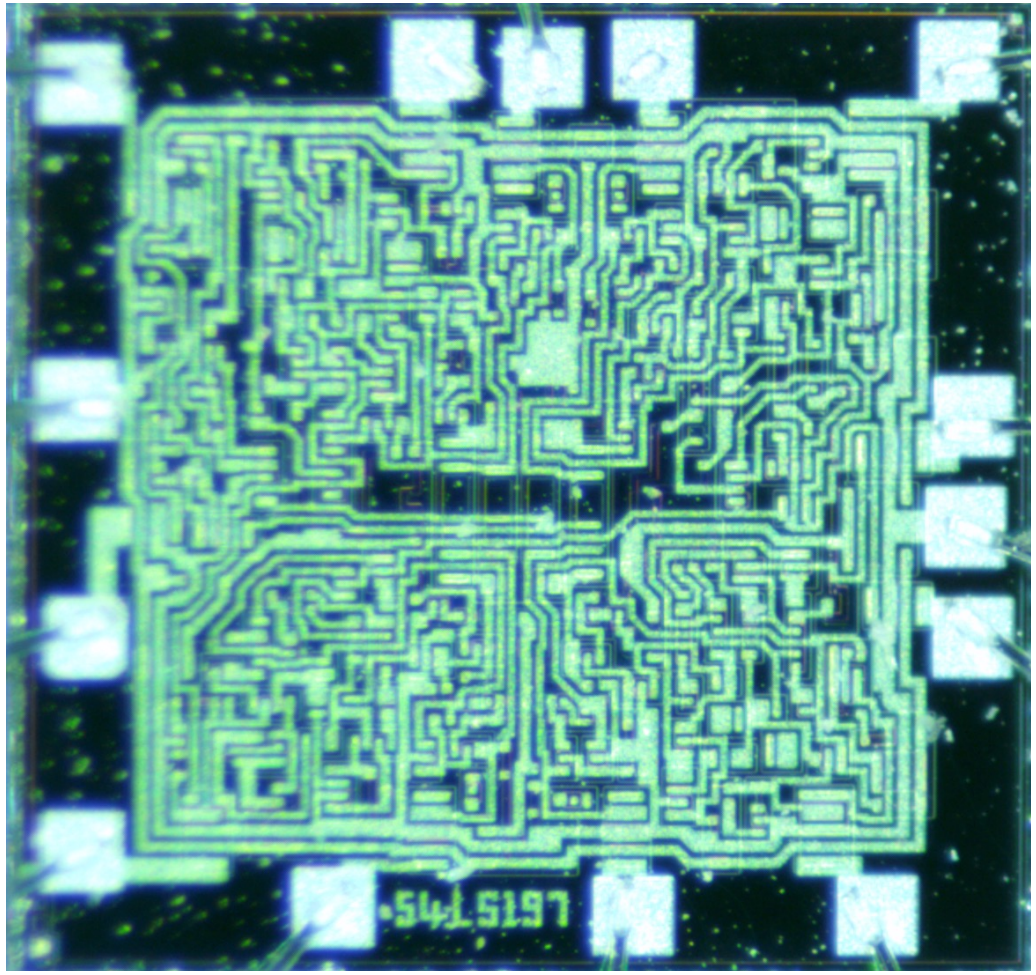


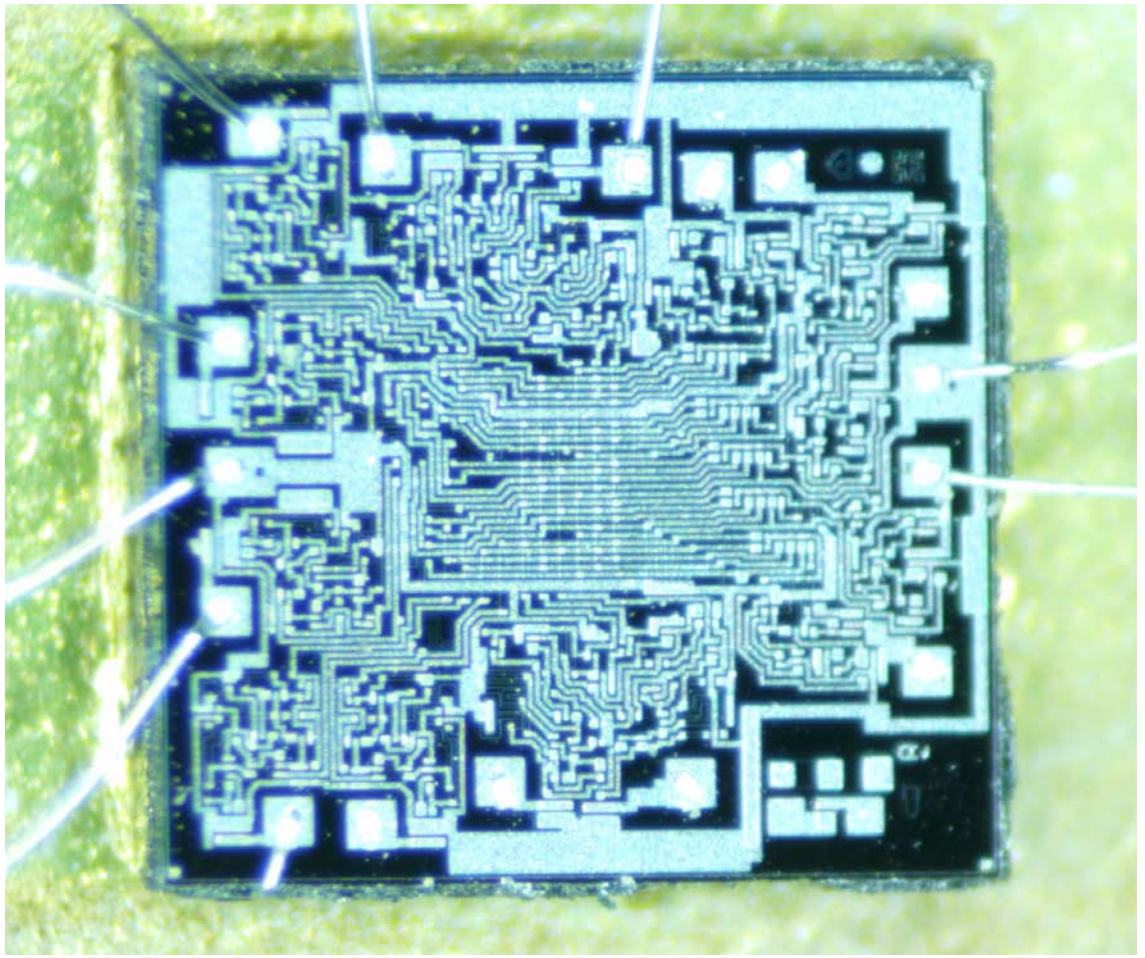
IC Decapsulation Example – Microscope Shots

- This is an old chip with large process node
- Wires visible under magnifying glass
- Only can see top layers here
  - Typically power distribution
  - Obscures interesting circuits
- Notice the bond wires around the edges



Comparison – Two Different 7400 Series Chips





Upcoming Lectures

- Secure Hardware Primitives
  - Anti-Tamper