Achilles: A Hardware Cybersecurity Platform

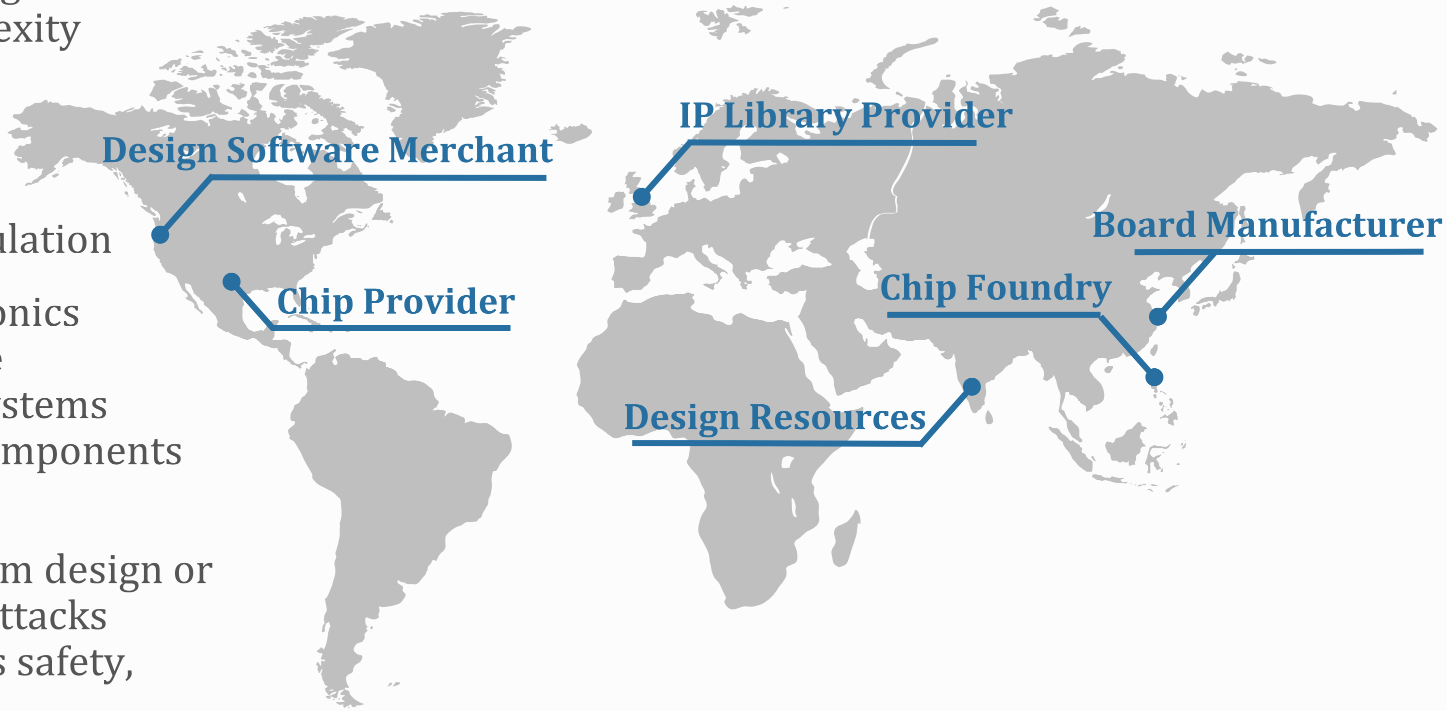
Alfred L. Crouch
W. Layton Ellington
Peter L. Levin

October 2022

The Semiconductor Supply Chain is Vulnerable to Attack

Complex electronic systems are assembled using components and processes from entities all over the world. Every party in this global supply chain creates an additional risk of potential security issues, especially when economic and ideological interests are misaligned. This is particularly true in the semiconductor value chain.

- **Design level:** Commercial semiconductors are designed for performance, not security; higher design complexity increases the threat surface exponentially
- **Manufacturing level:** Distributed intellectual property, testing, manufacturing, packaging, and shipping processes create opportunities for manipulation
- **Procurement:** Low-volume or custom-built electronics are too expensive to create in modern, high-volume manufacturing processes; thus, national security systems rely on cheaper commercial off-the-shelf (COTS) components that are prone to counterfeit
- **Product:** Inadvertent vulnerabilities that result from design or manufacturing defects – and deliberate hardware attacks by malfeasant actors – all threaten the end system's safety, reliability, and security



Infected systems remain permanently vulnerable until compromised components are physically replaced. Familiar approaches to software security – such as patches and updates – do not apply in this space. ***As a result, hardware attacks have become more common and present a serious risk to national security.***

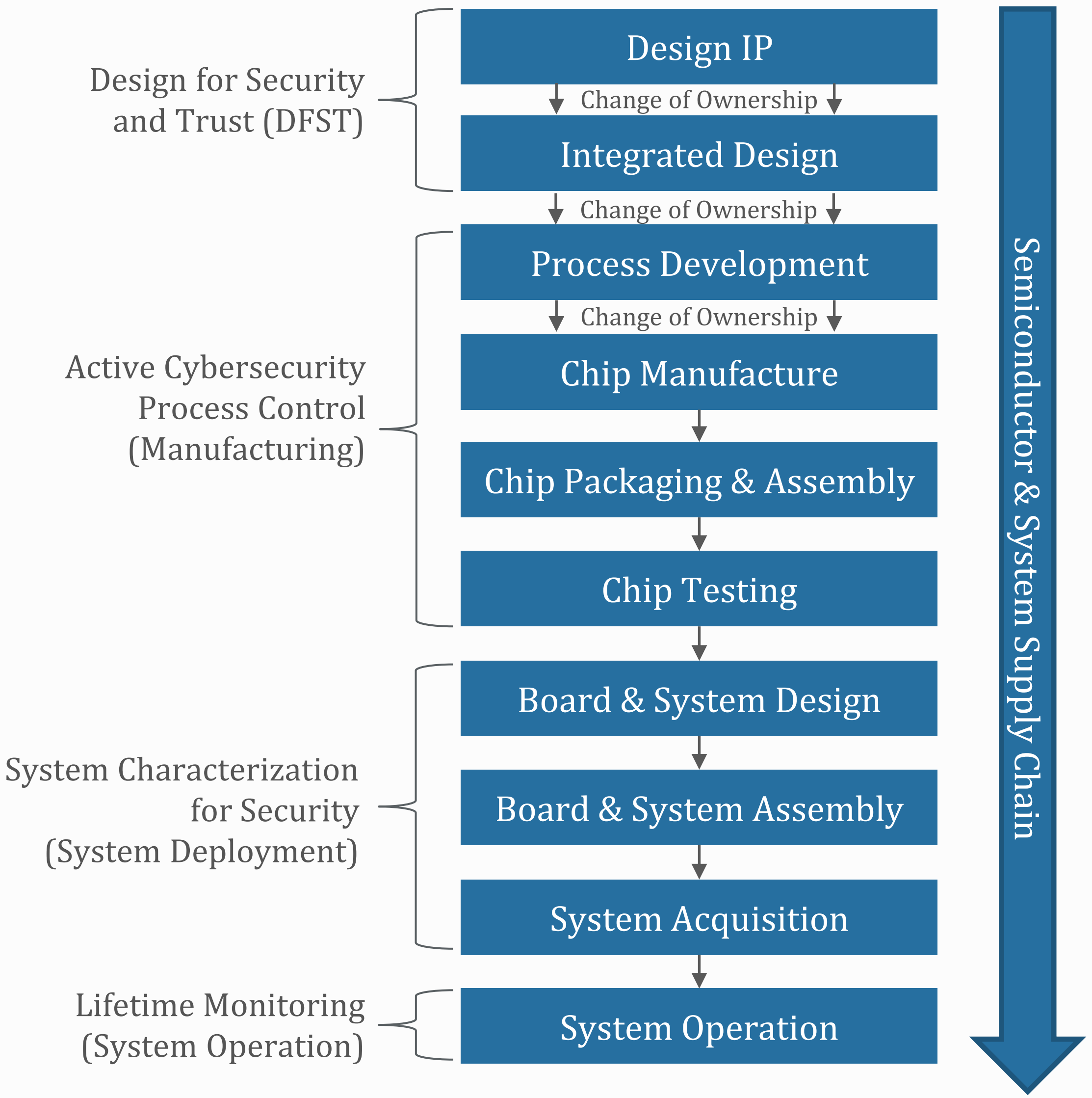
Semiconductor Supply Chain Security

Amida is building Achilles, a prototype security technology, to span the entire supply chain and detect incipient attacks, regardless of their origin. This process begins in the design phases and follows devices throughout their final deployment and operation.

Our goal is to provide an industry-wide methodology for the design, manufacture, distribution, and monitoring of secure devices for all safety-critical applications. We are developing:

- Electronic Design Automation (EDA) software tools that empower chip and system designers to evaluate and improve the security of their devices
- System characterization tools to study hardware attacks on large systems and create models to recognize anomalies in the future
- Real-time monitoring protocols that allow system administrators to observe the operation of devices throughout their lifespans

***Physical security in the manufacturing supply chain alone is insufficient to combat hardware attacks*.**


Design for Security and Trust (DFST)
Design IP
Change of Ownership
Integrated Design
Change of Ownership
Active Cybersecurity Process Control (Manufacturing)
Process Development
Change of Ownership
Chip Manufacture
Chip Packaging & Assembly
Chip Testing
System Characterization for Security (System Deployment)
Board & System Design
Board & System Assembly
System Acquisition
Lifetime Monitoring (System Operation)
System Operation
Semiconductor & System Supply Chain

The Hardware Threat Surface

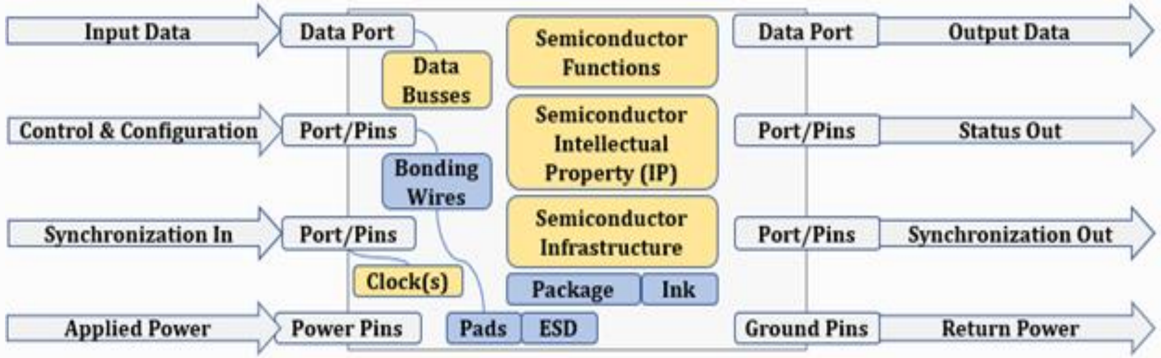
A semiconductor device has a manageable threat surface that can be categorized into three broad segments:

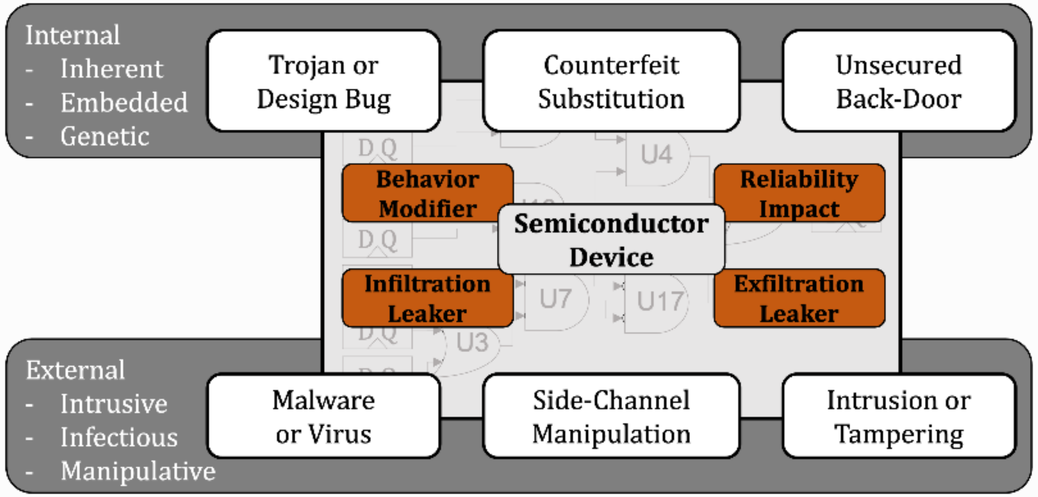
1. **Internal silicon content** – Functional components of the device implemented in semiconductor materials
2. **External connections** – Interconnection signals used to move information into or out of a device
3. **Device packaging** – Physical container used to protect the silicon device, distribute heat, and identify the component

Most significant attacks fall within three fundamental categories:

1. **Behavior modification** – Alters core functionality of a device to create malicious activity, like a kill switch
2. **Information leakage** – Infiltrates or exfiltrates protected information into or out of a system, such as leaking encryption codes
3. **Reliability impact** – Degrades the operation of a system or weakens user trust in its ability to complete the mission

The Achilles technology will enable design groups, manufacturers, and end users to secure their devices, regardless of the attack method or its origin.

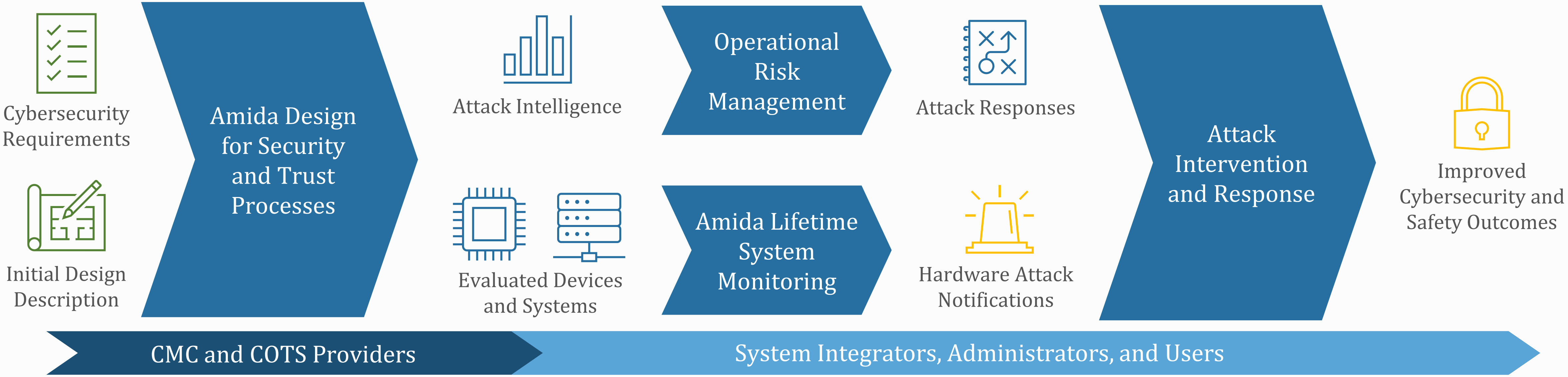




Operational Use and Improvement – Creating Secure Electronics

Amida's mission is to help critical-system designers and chip providers characterize risks and threats to custom microelectronic components (CMCs). Our tool enables previously unattainable cybersecurity and safety outcomes.

- Our Design for Security and Trust (DFST) methodology allows CMC providers and system designers to make more robust devices and systems available to the semiconductor ecosystem, using a pre-silicon analysis tool that is available today
- Once integrated into critical systems, administrators and users will be able to monitor their devices for previously undetectable hardware attacks and mitigate the risk of bodily injury, loss of life, degraded operational effectiveness, or harm to information security


Cybersecurity Requirements
Initial Design Description
Amida Design for Security and Trust Processes
Attack Intelligence
Evaluated Devices and Systems
Operational Risk Management
Amida Lifetime System Monitoring
Attack Responses
Hardware Attack Notifications
Attack Intervention and Response
Improved Cybersecurity and Safety Outcomes
CMC and COTS Providers
System Integrators, Administrators, and Users

The application of Amida's approach to hardware cybersecurity processes will help ensure national security, information privacy, and the safety of all users.

Forensic Assertions Are Good; Predictive Approaches Are Better

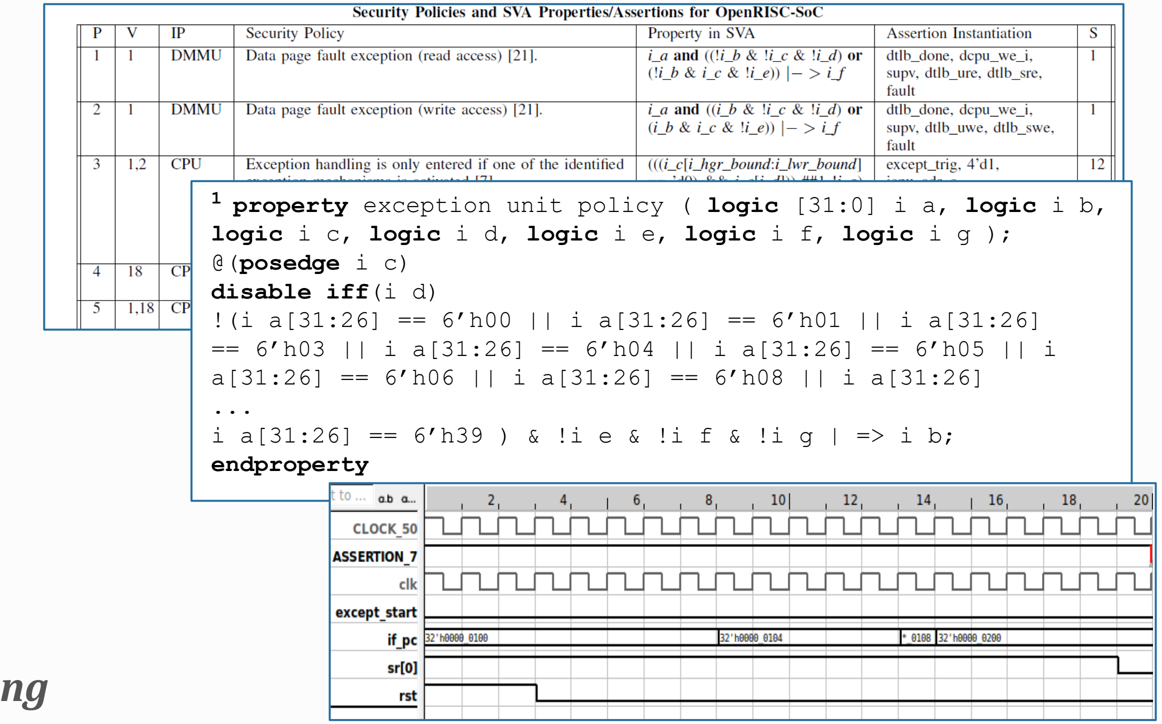
Malicious inclusions (MIs) are looking for special events

- Policy violation of behaviors that are permitted or prohibited
- These must be known and asserted prior to test
- “Narrow and deep” design-specific evaluation

“Security checkers for an entire design is a great deal of work . . . and adds to the significant effort of ordinary functional verification.”[2]

“Although most behavior requirements will be dynamically enforceable, some may not be. . .”[2]

Attackers are going to find a way through defenses, they are looking for vulnerabilities and exposures that we have not thought of yet.



Security Policies and SVA Properties/Assertions for OpenRISC-SoC

P	V	IP	Security Policy	Property in SVA	Assertion Instantiation	S
1	1	DMMU	Data page fault exception (read access) [21].	*i_a* **and** ((!*i_b* & !*i_c* & !*i_d*) **or** (!*i_b* & *i_c* & !*i_e*)) |− > *i_f*	dtlb_done, dcpu_we_i, supv, dtlb_ure, dtlb_sre, fault	1
2	1	DMMU	Data page fault exception (write access) [21].	*i_a* **and** ((*i_b* & !*i_c* & !*i_d*) **or** (*i_b* & *i_c* & !*i_e*)) |− > *i_f*	dtlb_done, dcpu_we_i, supv, dtlb_uwe, dtlb_swe, fault	1
3	1,2	CPU	Exception handling is only entered if one of the identified	(((*i_c*[*i_hgr_bound*:*i_lwr_bound*]	except_trig, 4’d1,	12
4	18	CP				
5	1,18	CP				

```
1 property exception unit policy ( logic [31:0] i a, logic i b,
logic i c, logic i d, logic i e, logic i f, logic i g );
@(posedge i c)
disable iff(i d)
!(i a[31:26] == 6’h00 || i a[31:26] == 6’h01 || i a[31:26]
== 6’h03 || i a[31:26] == 6’h04 || i a[31:26] == 6’h05 || i
a[31:26] == 6’h06 || i a[31:26] == 6’h08 || i a[31:26]
...
i a[31:26] == 6’h39 ) & !i e & !i f & !i g | => i b;
endproperty
```

[1] Security Analysis of a System-on-Chip Using Assertion-Based Verification, Bhamidipati et al, 2021

[2] Evaluating Security Requirements in a General-Purpose Processor by Combining Assertion Checkers with Code Coverage, Bizor et al, 2012

There's a Gap Between Truth and Proof . . .

We know that there are expressions (properties, inferences, axioms) that cannot be derived by formal methods:

- These expressions are true, but unseen in any conventional approach
- These expressions are correct, even if they cannot be proven

The analytical trade-space becomes:

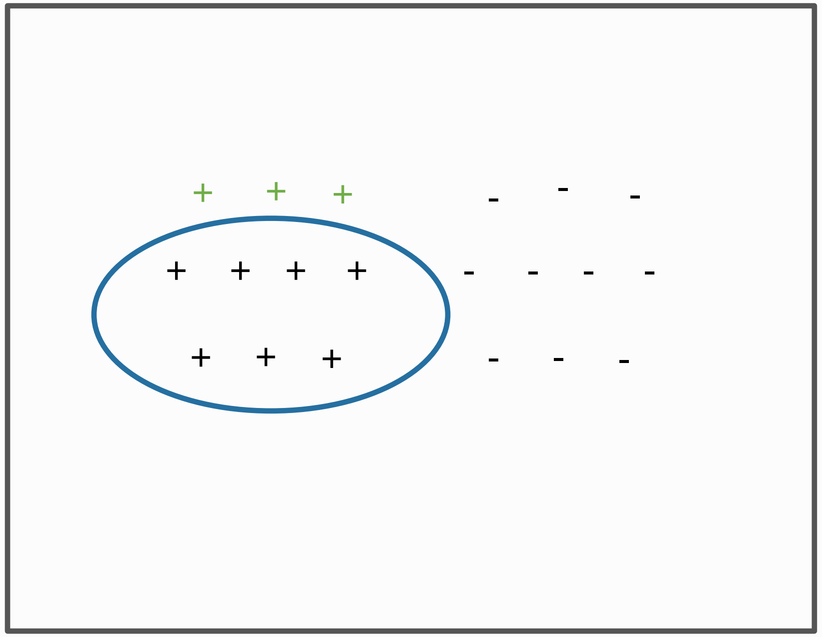
- A self-consistent and derivable set of expressions (that are incomplete)
- Or a complete set of expressions (that are not self-consistent, i.e., "provable" in a formal sense)

"Gödel showed there's a gap between truth and proof." – Marcus du Sautoy

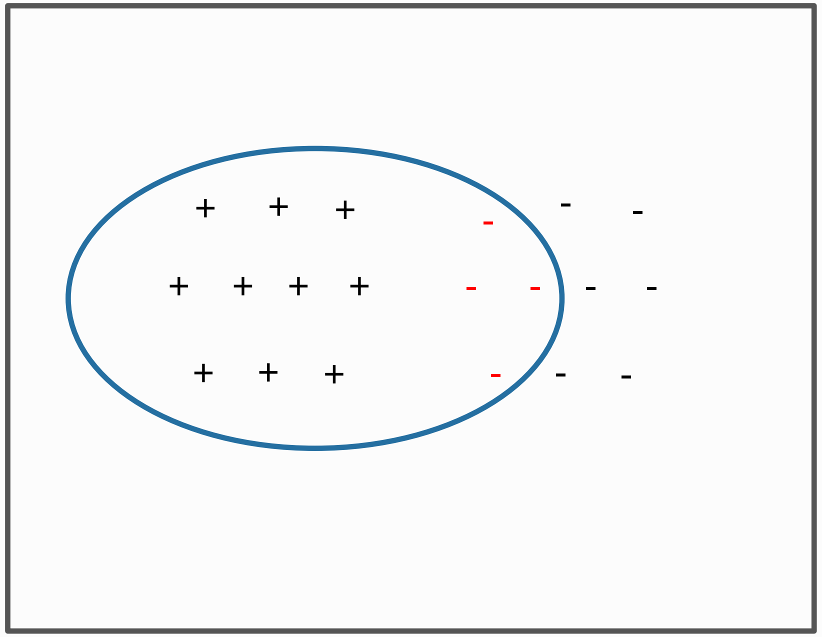
"You cannot formalize your understanding in a scheme which you can put on a computer." – Roger Penrose

"No axiomatic system whatsoever can produce all number-theoretic truths, unless it is an inconsistent system."
– Douglas Hofstadter

Formal Systems are Fundamentally Limited



incomplete, consistent



complete, inconsistent

A hypothesis is said to be **complete** if it covers all the positive examples of an object (expression) and **consistent** if it covers no negative examples.

From "Induction of Logical Relations Based On Specific Generalization of Strings," Yasin Uzun, Bilkent University, January 2007

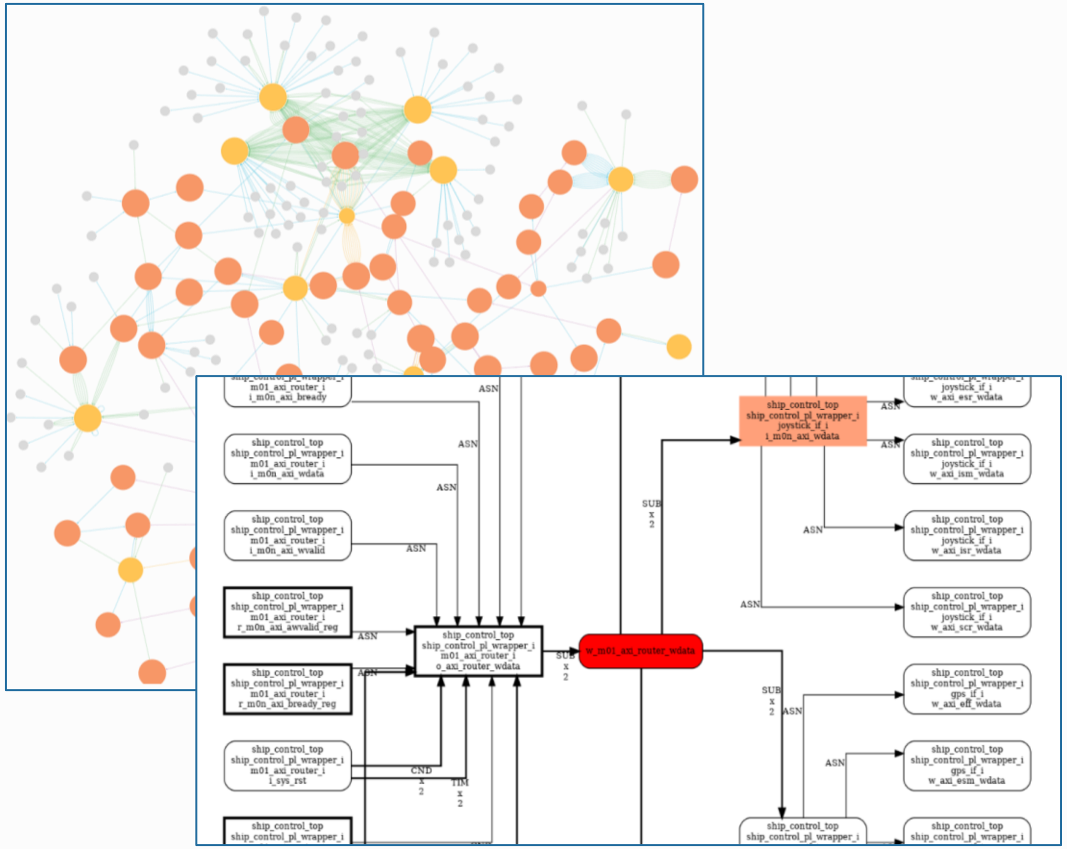
A Novel Approach to Design Analysis

Supply-chain-spanning security solutions must start in the design phase and carry through a device’s entire lifecycle. This approach – the alteration of a design to meet security objectives – is called Design for Security and Trust.

The Achilles Vulnerability Analysis Tool identifies potentially vulnerable locations in order to cyber-harden the RTL model. It enables wide exploration of vulnerabilities that are not discoverable through assertion-base protocols.

Use of our analysis software leads to hardened application-specific integrated circuits (ASICs), system on chips (SOCs), application-specific standard products (ASSPs), intellectual property (IP) cores, and field-programmable gate array (FPGA)-based systems, with an emphasis on design organization and customer security concerns.

Amida has developed a novel vulnerability analysis software product that examines behavioral designs to identify and predict locations of potential exploit before devices are manufactured.



m01_axi_router_i i_m0n_axi_bready
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i i_m0n_axi_wdata
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i i_m0n_axi_wvalid
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i r_m0n_axi_awvalid_reg
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i r_m0n_axi_bready_reg
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i i_sys_rst
ship_control_top ship_control_pl_wrapper_i m01_axi_router_i o_axi_router_wdata
w_m01_axi_router_wdata
ship_control_top ship_control_pl_wrapper_i joystick_if_i i_m0n_axi_wdata
joystick_if_i w_axi_esr_wdata
ship_control_top ship_control_pl_wrapper_i joystick_if_i w_axi_ism_wdata
ship_control_top ship_control_pl_wrapper_i joystick_if_i w_axi_isr_wdata
ship_control_top ship_control_pl_wrapper_i joystick_if_i w_axi_scr_wdata
ship_control_top ship_control_pl_wrapper_i gps_if_i w_axi_eff_wdata
ship_control_top ship_control_pl_wrapper_i gps_if_i w_axi_esm_wdata
ASN
SUB x 2
CND x 2
TIM x 2

The Achilles Vulnerability Analysis Tool converts the Verilog design model into a structural graph and applies algorithms to identify structures within the RTL susceptible to hardware attack.

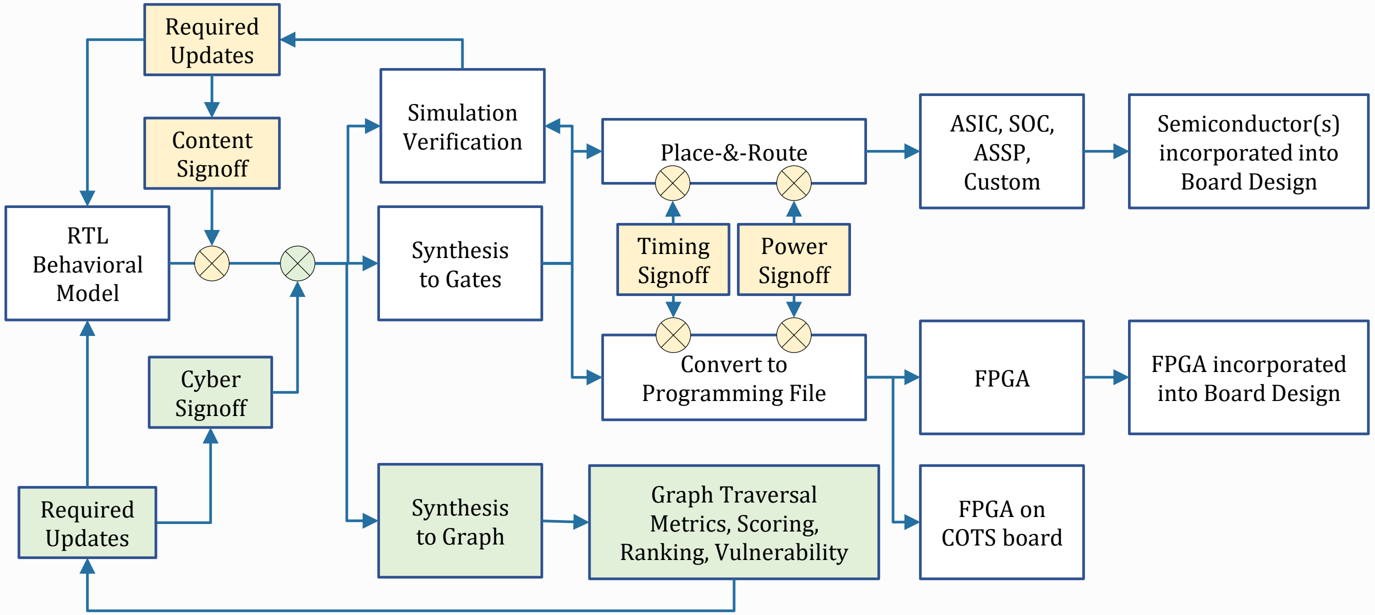
Achilles Vulnerability Analysis Use Model

The Achilles Vulnerability Analysis Tool fits within the semiconductor device development flow and the board-level programmable device flow. The software is best applied to a cybersecurity signoff process, similar to that used for device timing and power signoff.

The Vulnerability Analysis Tool:

- Operates in the flow between the generation and editing of the behavioral design model and the synthesis process used to generate a gate-level netlist
- Iterates along with the design model as IP and design-ware is updated and completed to become bug-free through verification
- Identifies threat/attack types and locations
- Informs remediation decisions and prioritization
- Provides verification that the vulnerability has been neutralized or downgraded

Cybersecurity hardening becomes part of the design flow through DFST and signoff.



The Achilles Vulnerability Analysis Tool iteratively evaluates the Verilog design model as part of the design flow to identify structures susceptible to hardware attack. This enables remediation to be incorporated incrementally as the design progresses toward manufacture. Cybersecurity-hardened devices can then populate boards and systems.

RTL Vulnerability Analysis Benefits

The Achilles Vulnerability Analysis Tool identifies weaknesses and exploits within behavioral Verilog RTL. If they escape cybersecurity evaluation, these vulnerabilities could become operationally dangerous in a silicon device.

With our RTL Vulnerability Analysis, a designer or design organization can:

- Identify areas of interest through hardware security insights provided by the analysis process
- Assess which vulnerabilities are of concern or high risk, and help to prioritize their remediation
- Generate targeted analysis based on certain regions of the design or specific security concerns
- Continuously improve security throughout design iteration processes

Users can re-evaluate their designs based on the Achilles Vulnerability Analysis, to:

- Accept vulnerability risks when they are inconsequential or do not warrant remediation
- Redesign or refactor logic to manage vulnerabilities, or modify synthesis and implementation constraints to ensure that found vulnerabilities are managed
- Insert instrumentation to surveil in-field behavior for real-time attack identification
- Insert additional logic to obviate the impact of vulnerability exploits

Achilles enables design organizations and IP vendors to improve the overall cyberhealth of their designs.

Vulnerability Reporting and Vulnerability Management

The Achilles Vulnerability Analysis Tool provides critical feedback to enable vulnerability mitigation and management. The tool generates a report informs the cost-tradeoff decisions that designers must make when they secure a device.

The tool returns two main views of the results:

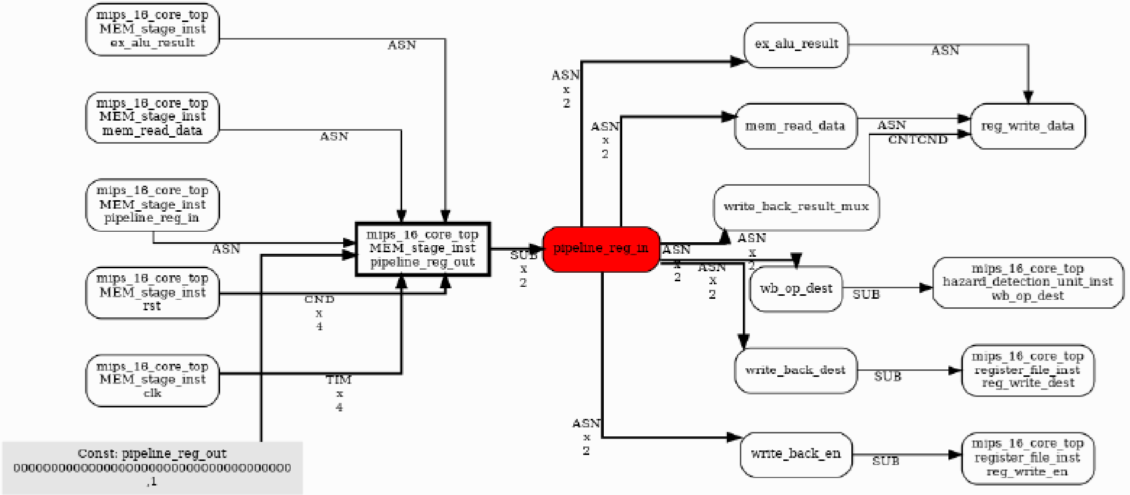
1. A set of design-level vulnerability overviews that show statistics about the design as a whole. This allows designers to choose areas to focus on and prioritize.
2. A set of individual explanations that provide important contextual and location information for each identified vulnerability, as well as a graphical representation of the influencing signals around each vulnerability.

The user can specify the types of vulnerability and the number of results to report. This enables focused, digestible analysis that can be used as a basis for mitigation actions.

Coupled with the Hardware Cybersecurity Handbook, the analysis report is the starting point to harden a digital design.

mips_16_core_top.WB_stage_inst.pipeline_reg_in

Source File	WB_stage.v
Attack Type	WID-SCL-COR
Rank	3
Score	33.00 / 49.00 (67%)
Node name	mips_16_core_top.WB_stage_inst.pipeline_reg_in
Module Name	WB_stage
Module Instance	mips_16_core_top.WB_stage_inst



mips_16_core_top MEM_stage_inst ex_alu_result
mips_16_core_top MEM_stage_inst mem_read_data
mips_16_core_top MEM_stage_inst pipeline_reg_in
mips_16_core_top MEM_stage_inst rst
mips_16_core_top MEM_stage_inst clk
Const: pipeline_reg_out 0000000000000000000000000000000000000 ,1
ASN
ASN
ASN
CND x 4
TIM x 4
mips_16_core_top MEM_stage_inst pipeline_reg_out
SUB x 2
pipeline_reg_in
ASN x 2
ASN x 2
ASN x 2
ASN x 2
ASN x 2
ASN x 2
ex_alu_result
mem_read_data
write_back_result_mux
ASN
ASN
CNTCND
reg_write_data
wb_op_dest
SUB
mips_16_core_top hazard_detection_unit_inst wb_op_dest
write_back_dest
SUB
mips_16_core_top register_file_inst reg_write_dest
write_back_en
SUB
mips_16_core_top register_file_inst reg_write_en

Example vulnerability explanation content for a widespread corruption-based kill-switch attack (shown in red) that could be used to cripple circuit functionality by injecting errors into a critical location.

Vulnerability Analysis as Part of Design for Security and Trust

The Achilles Vulnerability Analysis Tool is the first component in our overarching Achilles Design for Security and Trust Methodology.

We start with the RTL description and analyze the design structure for potential sites of exploitable vulnerabilities. We then embed a simple set of instruments – the focus of our next product roadmap milestone – into the design to provide visibility into real-time device operation.

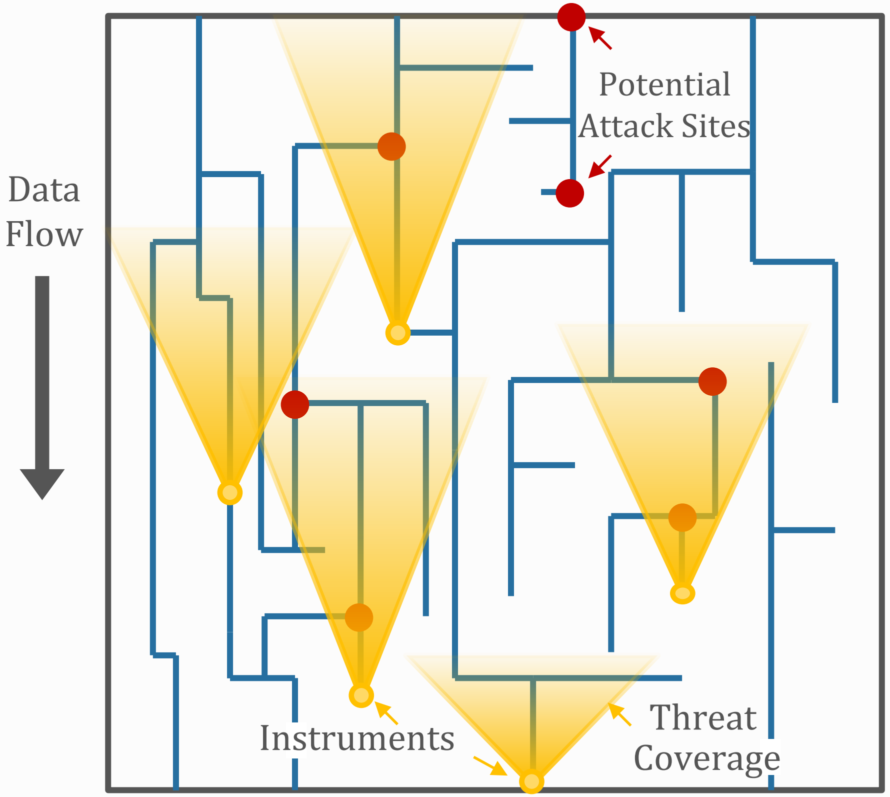
Next, we emulate the design and capture data on a sandbox platform to study the impact of attacks against the identified weaknesses.

Achilles leverages this data to build an automated attack-recognition system that:

- Detects anomalous behavior in the circuit to differentiate nominal from suspicious activity
- Identifies anomalies and classifies them according to learned predictive algorithms, from which it attributes attack behaviors to likely causes

Using insights from this process, designers can better mitigate threats with:

- Refactored designs that obviate high-risk vulnerabilities and harden their implementations to attack
- Embedded instrumentation that enables real-time monitoring during system operation to assess in-the-field health and security



Attacks cause effects downstream; instruments observe effects upstream. A recognition model can then use collected instrument data to detect and identify attacks according to their behavior.

Amida's Design for Security and Trust Methodology

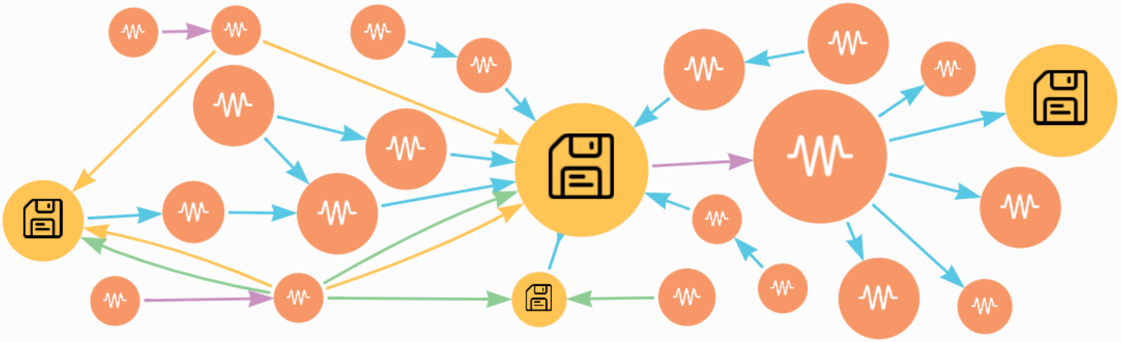
We have developed two DFST process stages to help secure chips, IP, and CMC-based systems.

Achilles Vulnerability Analysis Tool

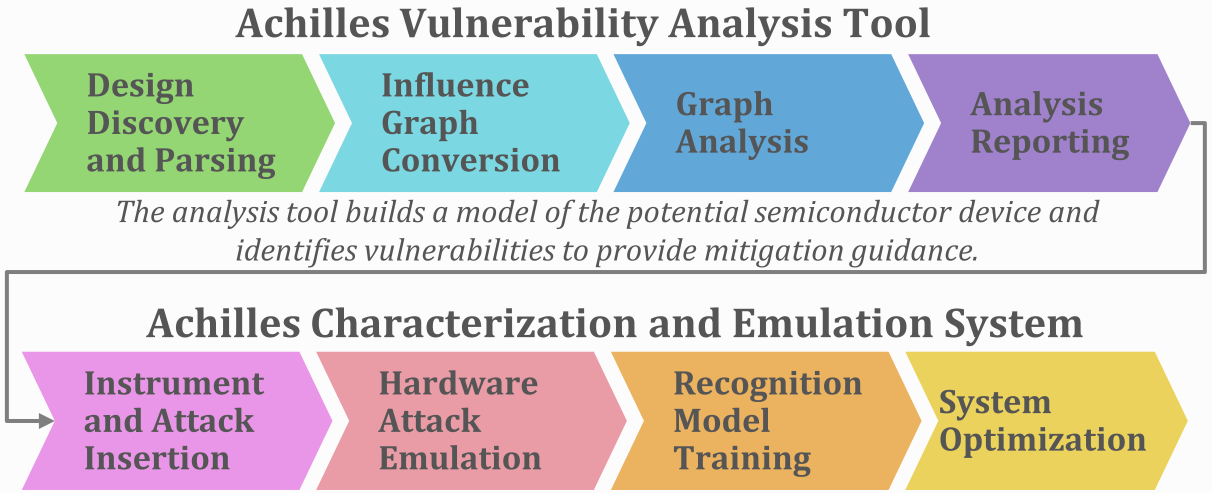
- Analysis software used to identify weaknesses and vulnerabilities present in design (RTL) source code
- Leverages a novel Influence Graph representation that models how device elements interact with one another
- Generates a vulnerability report that informs the device designer of potential problem areas that need to be hardened
- *Available today as a ready-to-use software product, with major capacity updates coming in 2023*

Achilles Characterization and Emulation System

- Inserts cybersecurity instruments into the design
- Collects operational data during emulated attacks to train threat recognition models to detect and identify malicious behaviors
- Trains real-time attack recognition algorithms
- Optimizes cost considerations to meet security objectives
- *Future software products are currently in development*



A novel influence graph structure is used to model the interactions between signals within a device. Different-colored nodes represent types of elements, and different-colored arrows signify varieties of relationships.



Achilles Vulnerability Analysis Tool

Design Discovery and Parsing → Influence Graph Conversion → Graph Analysis → Analysis Reporting

The analysis tool builds a model of the potential semiconductor device and identifies vulnerabilities to provide mitigation guidance.

Achilles Characterization and Emulation System

Instrument and Attack Insertion → Hardware Attack Emulation → Recognition Model Training → System Optimization

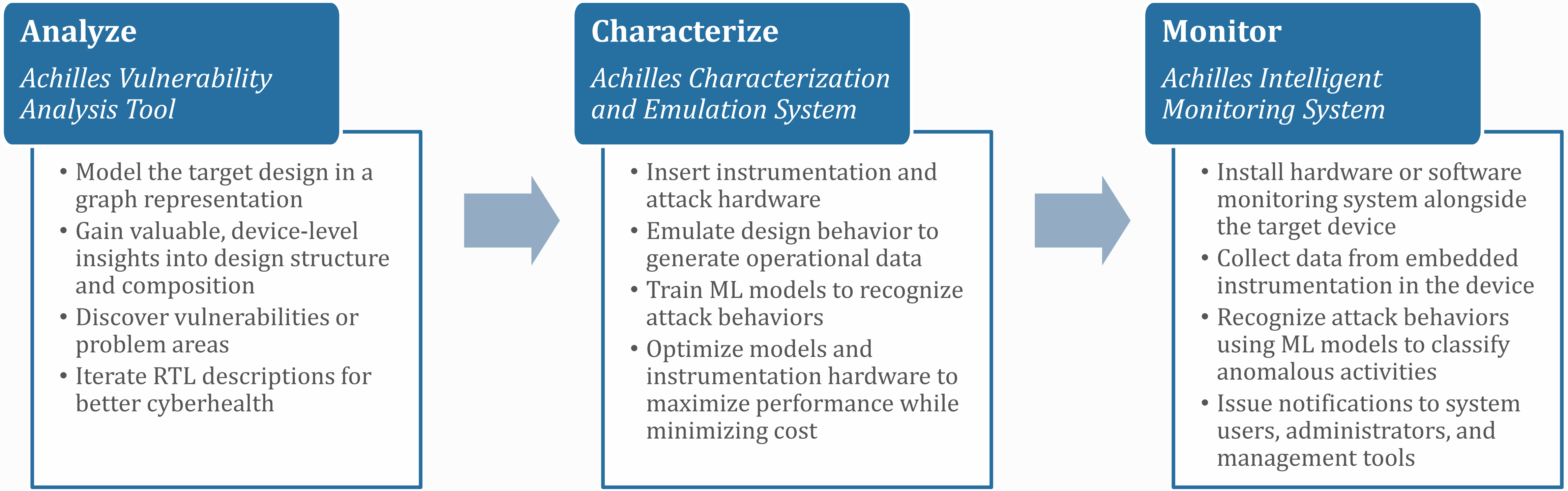
The characterization tool modifies the target design and emulates the effects of attacks against the device. Data is collected to build attack recognition models that can be used to mitigate attacks in real-time.

The Complete Achilles Solution Suite

Our overall approach draws many parallels to design-for-testability (DFT), which adds features during the design process to validate, post-manufacturing, that a device contains no *accidental* defects. Our method adds cyber-testability to reveal *deliberately* inserted defects throughout the device's entire lifespan.

When complete, our system will evaluate a design for impactful cybersecurity vulnerabilities, insert monitoring instruments, and train an initial set of attack recognition models that monitor the end-user device once it is integrated. A monitoring system installed alongside the target device can use the ML models to supply real-time attack notifications.

Our complete solution will provide three products that implement the following actions:



Analyze

Achilles Vulnerability Analysis Tool

- Model the target design in a graph representation
- Gain valuable, device-level insights into design structure and composition
- Discover vulnerabilities or problem areas
- Iterate RTL descriptions for better cyberhealth

Characterize

Achilles Characterization and Emulation System

- Insert instrumentation and attack hardware
- Emulate design behavior to generate operational data
- Train ML models to recognize attack behaviors
- Optimize models and instrumentation hardware to maximize performance while minimizing cost

Monitor

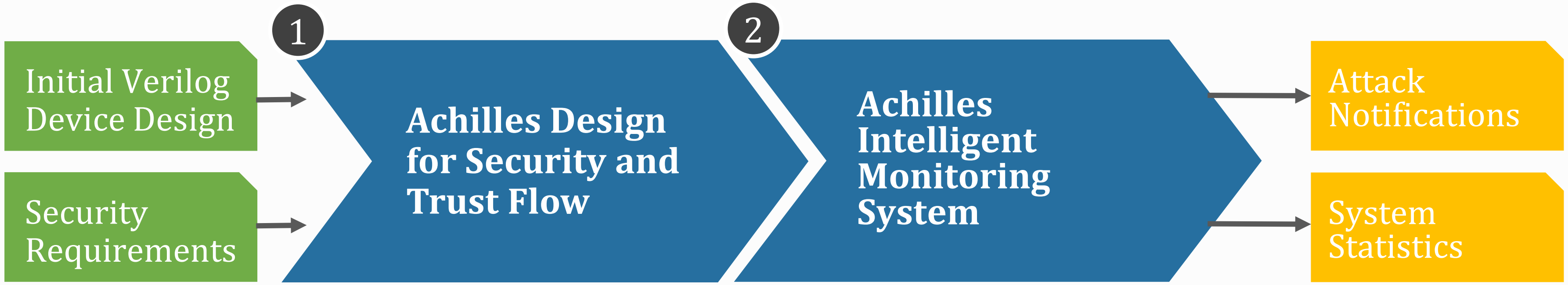
Achilles Intelligent Monitoring System

- Install hardware or software monitoring system alongside the target device
- Collect data from embedded instrumentation in the device
- Recognize attack behaviors using ML models to classify anomalous activities
- Issue notifications to system users, administrators, and management tools

The Complete Achilles Technology Suite

Together, the Achilles Vulnerability Analysis Tool and the Achilles Characterization and Emulation System implement a complete DFST flow. Devices created using this flow will be manufactured with embedded instrumentation and ready-to-use ML models that can used by the future Achilles Intelligent Monitoring System to watch for anomalies in the field-deployed device.

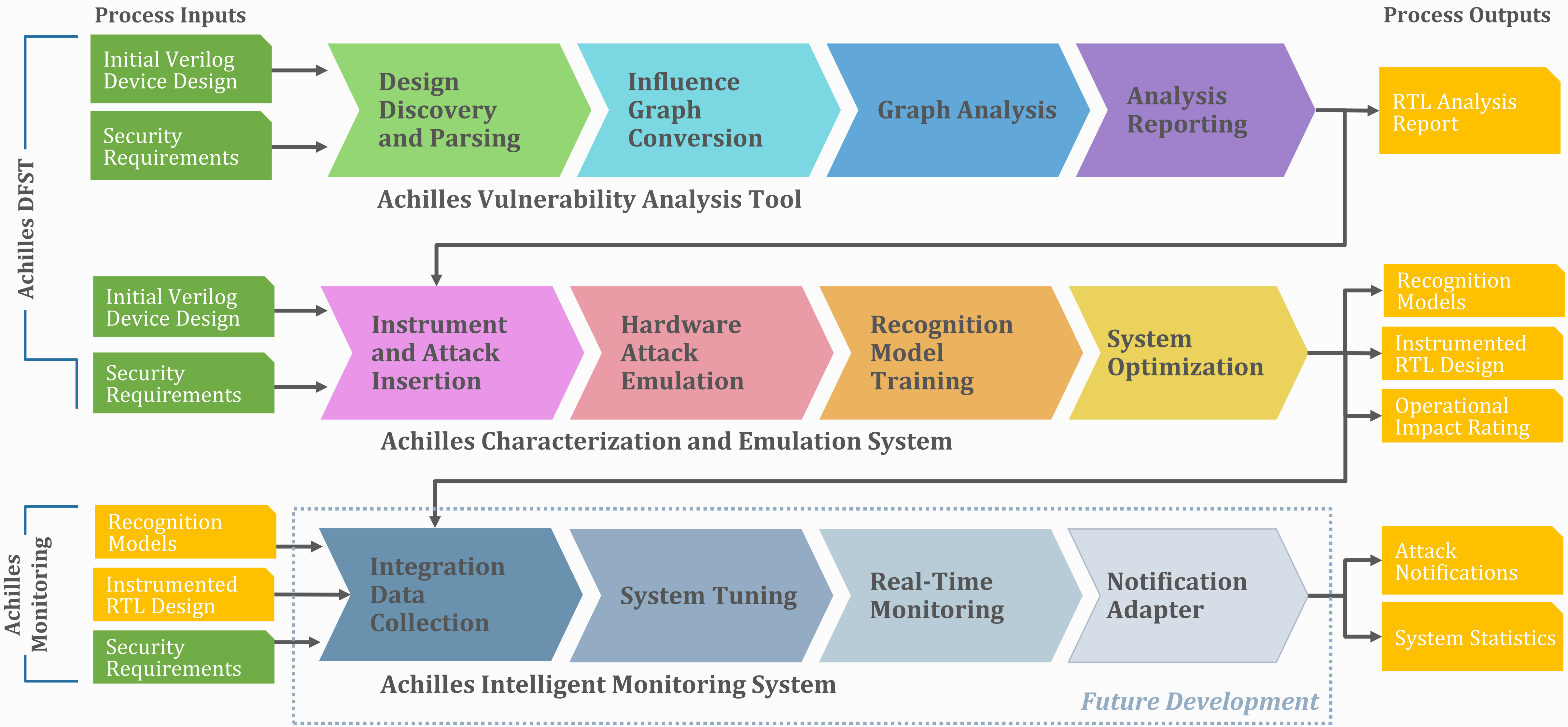
Complete Achilles Technology Suite



Initial Verilog Device Design
Security Requirements
1
Achilles Design for Security and Trust Flow
2
Achilles Intelligent Monitoring System
Attack Notifications
System Statistics

	1) Achilles Design for Security and Trust	2) Achilles Intelligent Monitoring System
Application space and embodiment	Software that implements our DFST methodology during the device design stage (IP, SOC, or FPGA-based system). This space is covered by the combined functionality of our first two products.	Software and hardware appliances that accompany the deployed system, which is integrated into the hardened device. A planned monitoring software solution will cover this space.
Function	Exposes vulnerabilities that are highly detrimental to key operations and embeds monitoring capabilities into the final design. This enables later attack recognition and notification.	To monitor the system for hardware attacks, applies predictive modeling (trained during both device design and system deployment) to operational data in real-time.
Outcomes	Provides feedback about design components for device-hardening during development stages and creates necessary monitoring elements.	Attack recognition models operate in predictive mode alongside a manufactured device to detect and identify threats, alerting users to activity.

The Complete Achilles Technology Suite – Process Overview



Achilles Moves Toward More Secure Electronics

The Problem

From large industrial and military systems down to consumer goods, severe risks threaten many areas of national security. Core infrastructure, transportation, personal devices, medical equipment, and information technology systems are all at risk of attack.

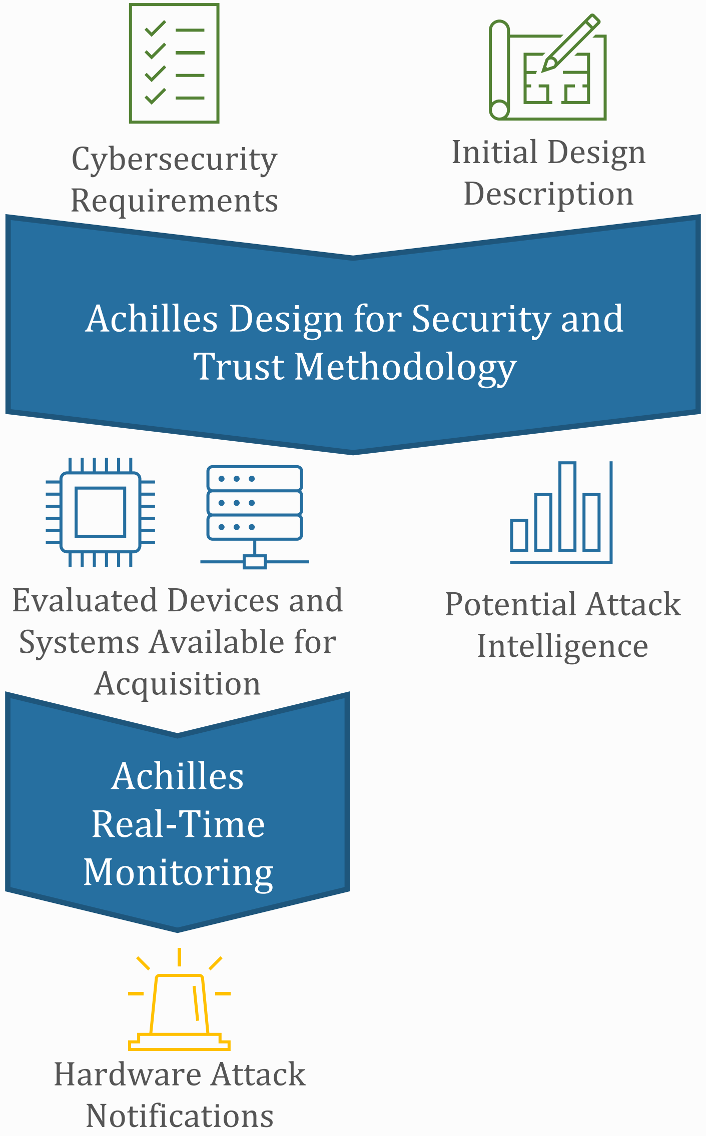
Due to the complexity of the supply chain needed to produce these devices and systems, merely locking down the manufacturing process cannot, by itself, remediate all hardware-assurance and cybersecurity issues. Secure fabrication facilities are not enough.

The semiconductor supply chain needs an end-to-end process to meet safety and cybersecurity objectives throughout a device's lifetime.

The Solution

Amida's Achilles technologies will help secure and protect semiconductor devices from attacks, regardless of their origin. Predictive analysis and machine learning improve and characterize systems during their design and validation. A real-time monitoring solution deployed alongside target hardware informs users of attack activity. Our methodology fights hardware threats throughout the entire system lifecycle.

We currently offer the first stage of this process through the Achilles Vulnerability Analysis Tool, which helps designers evaluate and manage vulnerabilities in their RTL designs.


Cybersecurity Requirements
Initial Design Description
Achilles Design for Security and Trust Methodology
Evaluated Devices and Systems Available for Acquisition
Potential Attack Intelligence
Achilles Real-Time Monitoring
Hardware Attack Notifications

Thank You.