Managing Security Trade-offs Against Economics of Chip Design

ZERO TRUST HARDWARE ARCHITECTURES WORKSHOP

11/03/2022

Serge Leef | Head of Azure for Secure Microelectronics | Microsoft

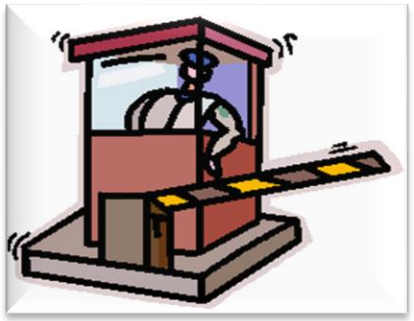
Abstract

While achievement of a zero-trust design, implementation and fabrication processes is a great aspirational goal, it is important to recognize that security comes at a cost for ASICs, SoCs, and Heterogeneous Integration Platforms. A nuanced model of the attack surfaces must be considered in the context of intended applications and trade-offs need to be assessed against performance, silicon area and power dissipation. This challenge calls for a fresh look at the design tools, IP management and re-use practices along with examination of new attack surfaces arising when multiple chiplets are integrated on a silicon interposer.

Designed-in Security

Automate inclusion of scalable **defense** mechanisms into chip designs to enable **security vs. economics** optimization

Cost and Complexity of Attack Resistance Mechanisms

Section 1/3

Securing Silicon in…

FUNCTIONAL DESIGN

Direction: *Platform-based Design + Security + Optimization…*

Large attack surface resulting from design complexity can be substantially reduced by automating inclusion of security into architecture and design tools operating at higher abstraction

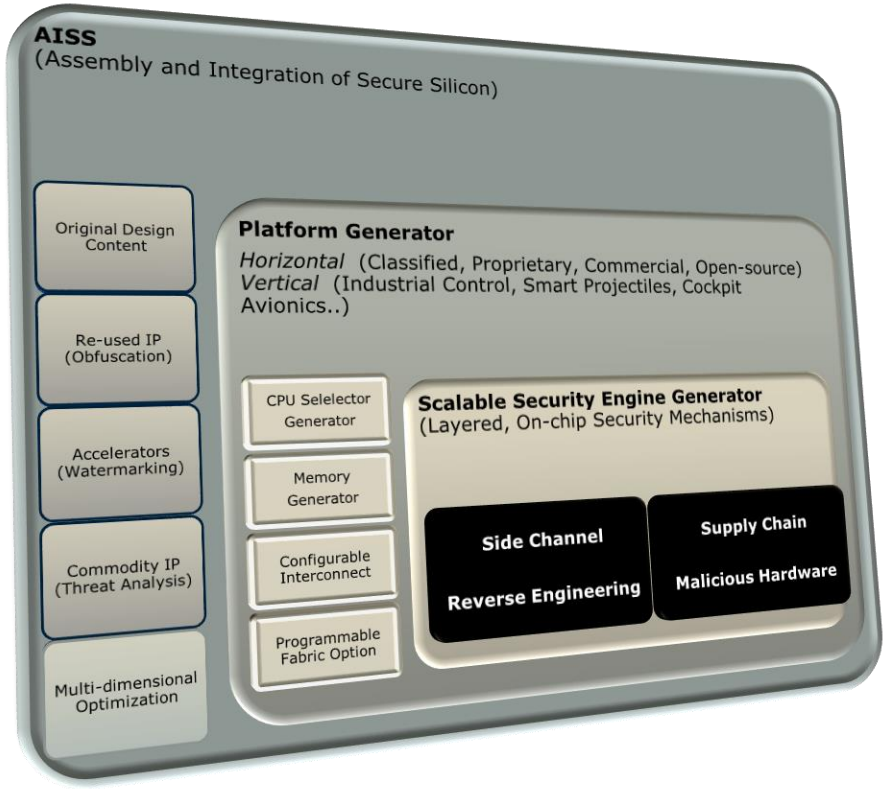
Reduce human induced security risks by abstracting away details & complexity

Security

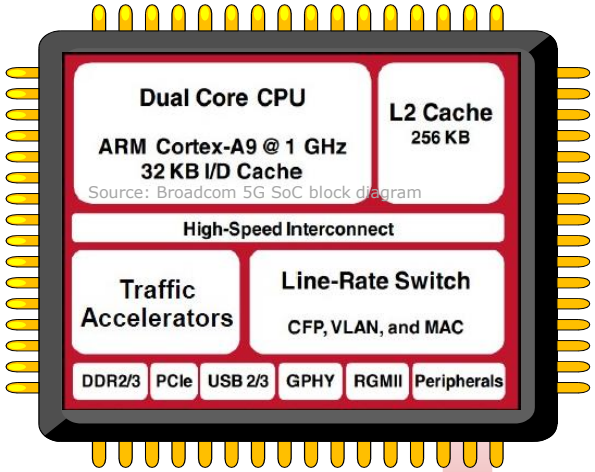
Incorporation of security into next generation of system chips, using platform-based design techniques & advances in high level synthesis

Automation

Need for automatic injection of scalable security creates an opportunity for tools & IP that enable semi automated and **automatic** approaches to assembly and integration that can substantially improve design productivity

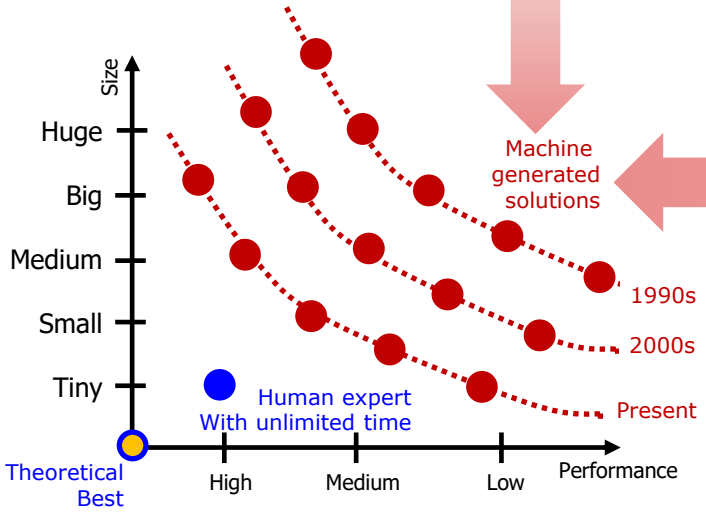


Background: System on Chip Design Process

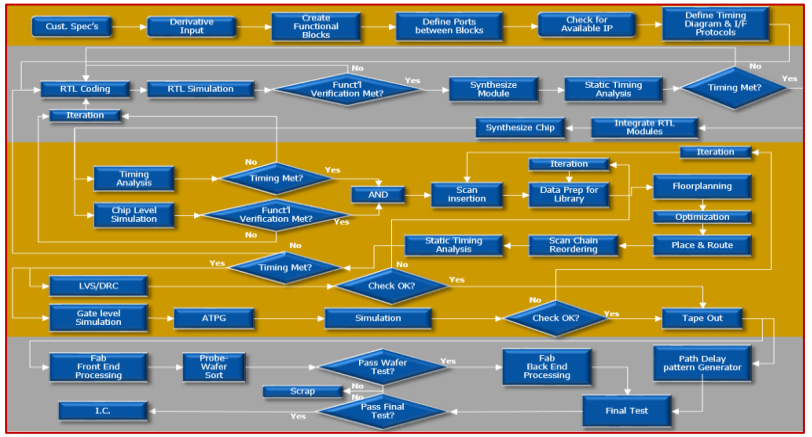


Source: Broadcom 5G SoC block diagram

(source: Broadcom)



Simplified View of SoC Design Process
(source: Mentor)



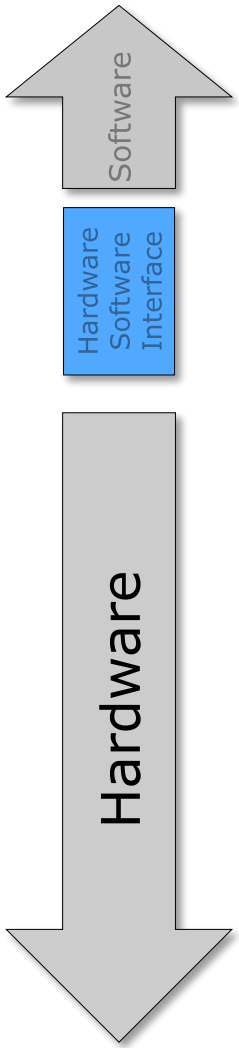
Current Practice

- Manual system integration
- Lengthy and complex simulation runs
- Block level synthesis & optimization

Limitations

- $30M+ cost for low complexity SoC
- 9-12 month design cycles
- Many human introduced errors
- **Unpredictable power and no security**

Background: Attack Surface Based Reference Model



Moving Target (I20)

- Substantial efforts are on-going in the software community

In Progress (SSITH)

- Alteration of system behavior based on software-accessible points of illicit entry that exist due to hardware design weaknesses or architectural flaws

AISS Focus Areas

- **Side Channel** – extraction of secrets through physical communication channels other than intended
  (assumption: attackers are able to "listen" to emissions)
- **Reverse Engineering** – extraction of algorithms from an illegally obtained design representation
  (assumption: attackers have access to design files)
- **Supply Chain** – Cloning, counterfeit, recycled or re-marked chips represented as genuine
  (assumption: attackers can manufacture perfect clones)
- **Malicious Hardware** – insertion of secretly triggered hidden disruptive functionality
  (assumption: attackers successfully inserted malicious function(s) into the design)

Who is the Target User?

Huge merchant semiconductor companies *(Intel, Broadcom, Qualcomm…)*

- See the critical need and have large expert teams to create custom solutions

Mid-size semiconductor and system companies *(NXP, Cisco, Nokia…)*

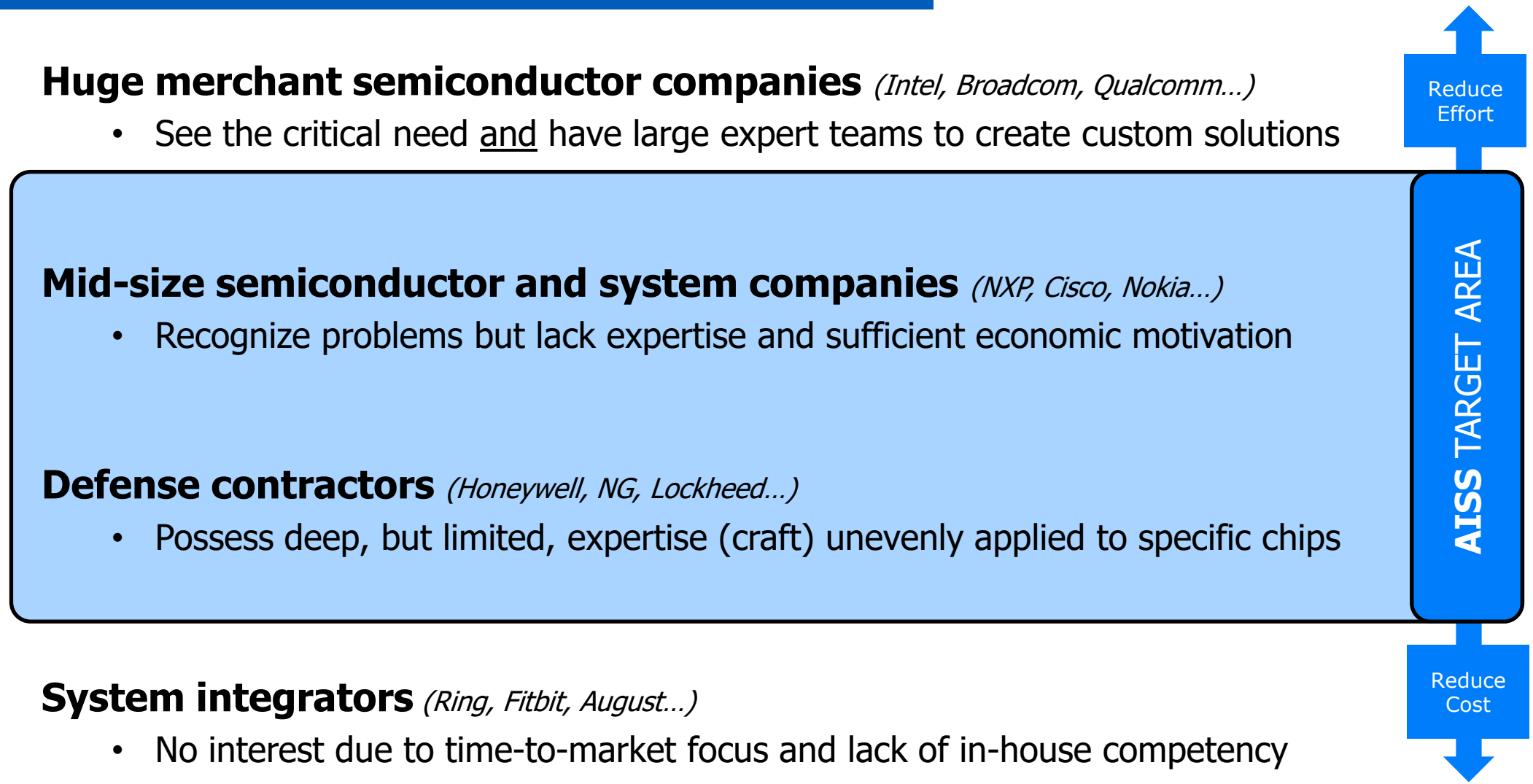
- Recognize problems but lack expertise and sufficient economic motivation

Defense contractors *(Honeywell, NG, Lockheed…)*

- Possess deep, but limited, expertise (craft) unevenly applied to specific chips

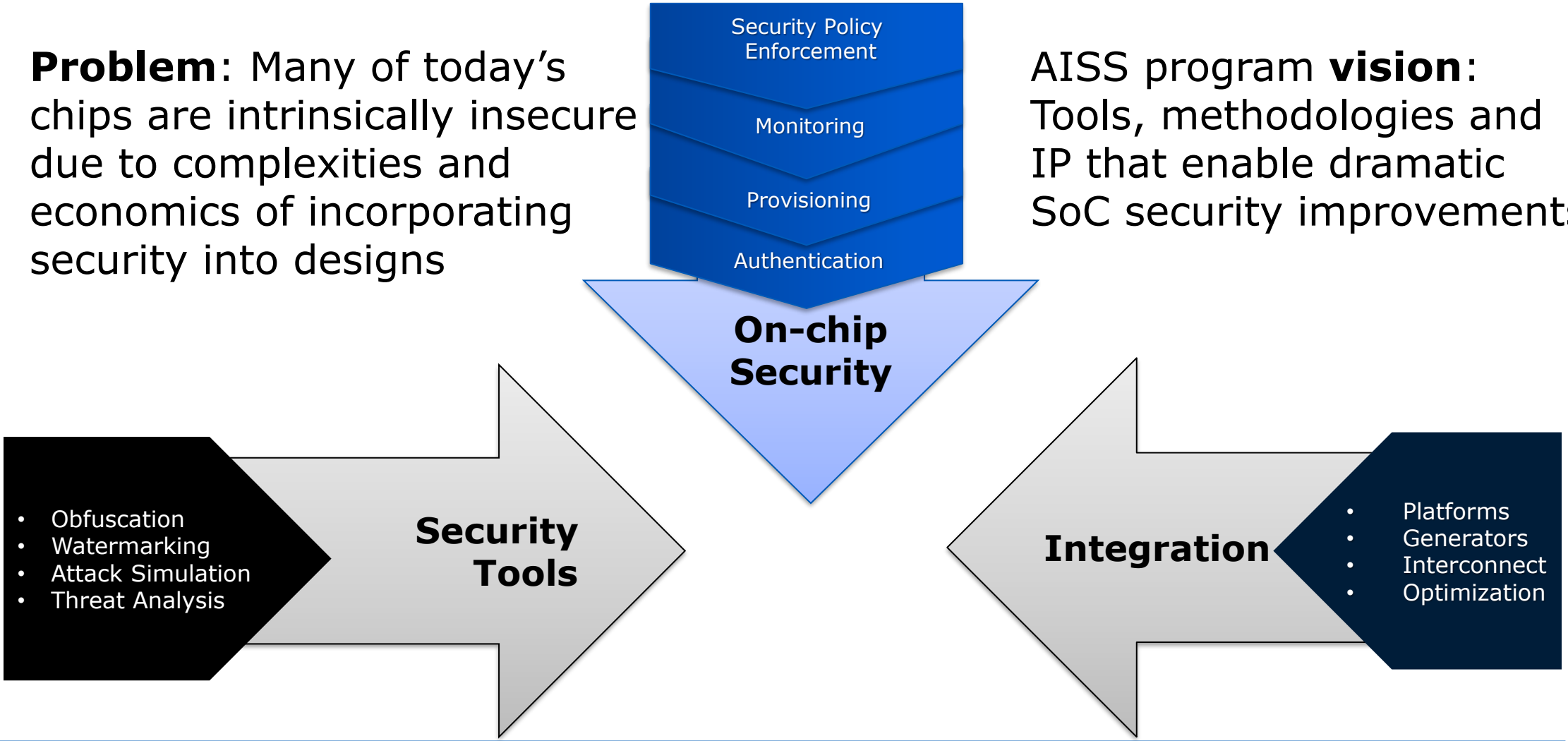
System integrators *(Ring, Fitbit, August…)*

- No interest due to time-to-market focus and lack of in-house competency



Vision: Make Chip Security Pervasive

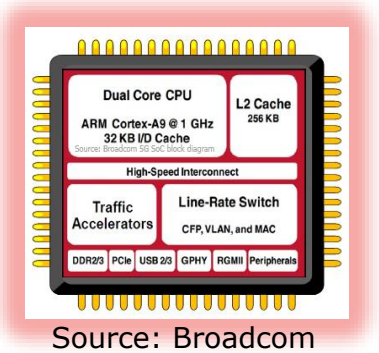
Problem: Many of today's chips are intrinsically insecure due to complexities and economics of incorporating security into designs



AISS program **vision**: Tools, methodologies and IP that enable dramatic SoC security improvements

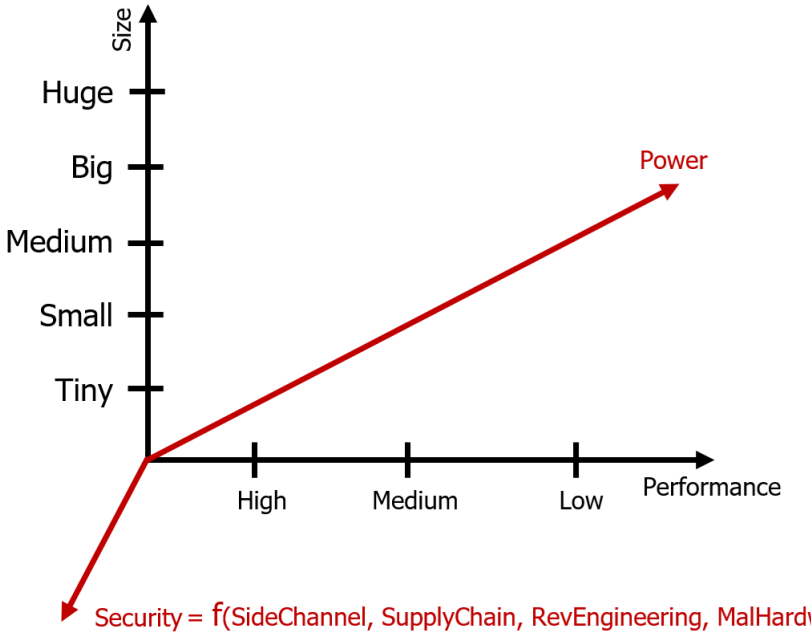
System Synthesis of Secure Chips

System synthesis & optimization



Source: Broadcom

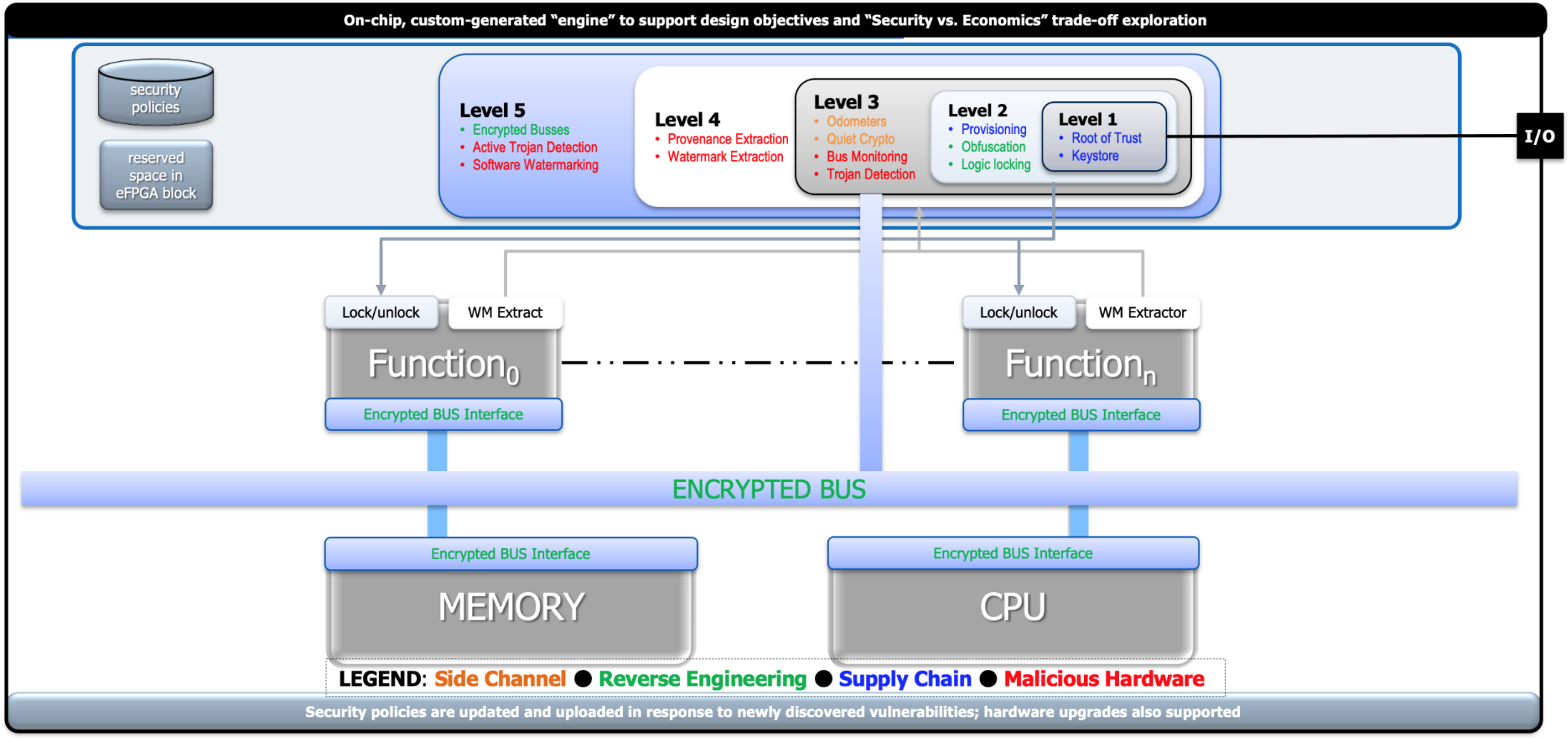
1. Σ(a*Performance, b*Size)
2. Σ(a*Performance, b*Size, c*Power)
3. Σ(a*Performance, b*Size, c*Power, d*Security)
4. Σ(a*Performance, b*Size, c*Power, {d*SideChannel, e*SupplyChain, f*RevEngineering, g*MalHardware})



Key challenges:

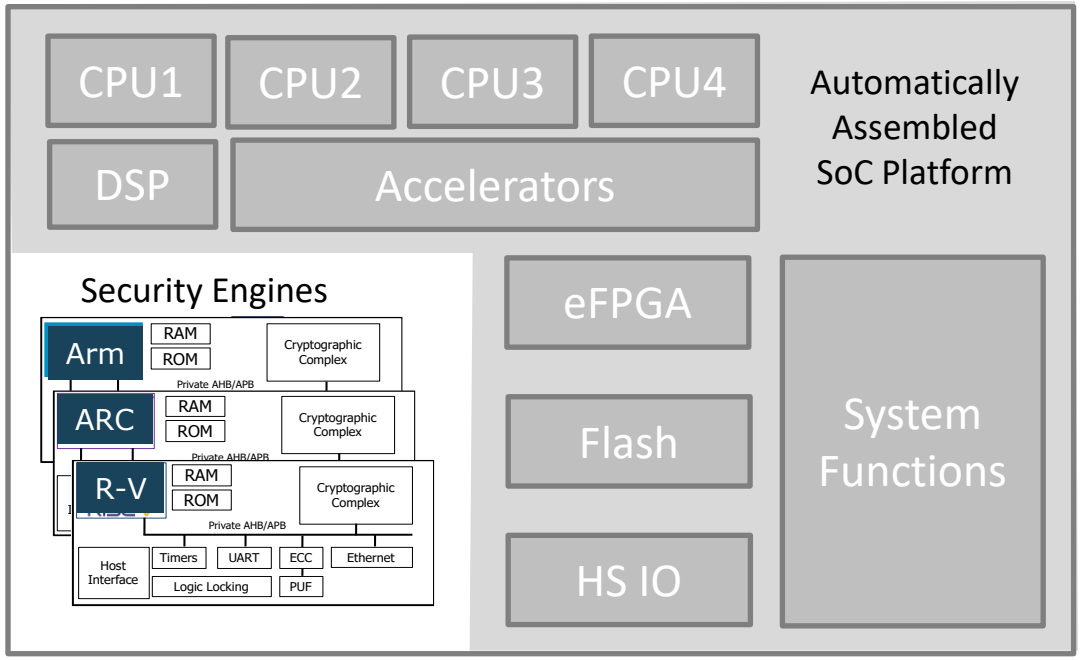
- *Quantification of security*
- *Rapid estimation of attack resistance*
- *Multi-dimensional optimization*

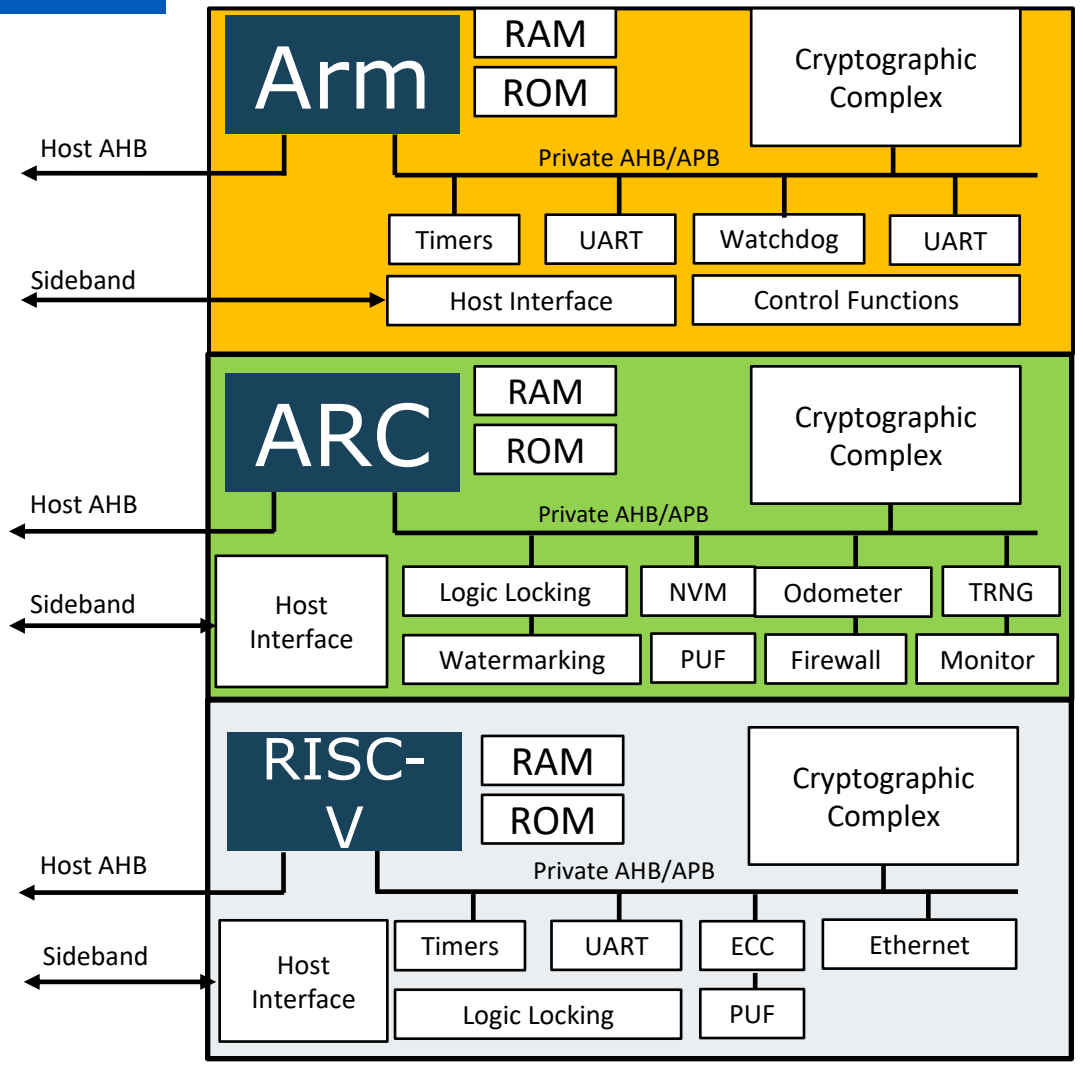
Security Engine: High Concept



Security Engine: Architecture and Implementation(s)

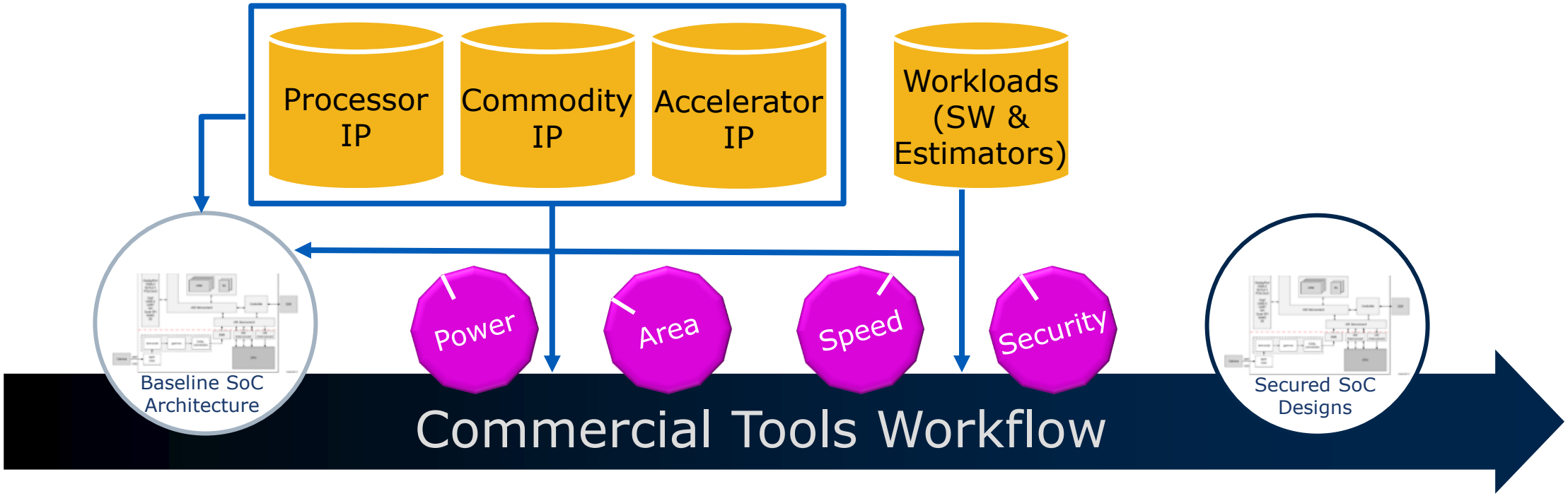
- Security Engines serve as a RoT & much more
- Modular on hardware and software level
- Three ISA's: Arm, Synopsys ARC, RISC-V
- Each SE has its own unique architecture/features
- Each SE configurable based on PASS constraints
- Field upgradable security policies





AISS: Solving the HW Security Problem

Security threat resistant design | **Easy to Use Without Deep Expertise** | **Configurable security levels**



MISSION: Make hardware designs more resilient to modern-day security concerns

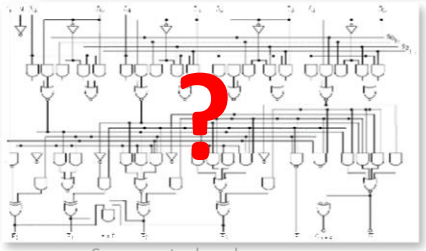
Automated mitigation against IP theft, counterfeiting, and malware insertion

Static Threat Analysis & Mitigation

- **Suspect Hardware -** Rogue hardware injected into the chip
  - **Watermarking** of circuits that survive transformations & can be extracted from all design representations & from live silicon
  - **Analysis of Threats** by examination of designs for suspect Trojans based on “low activity” and “hard to activate” circuits
- **Emission of Meaningful Data -** Accessible side-channels
  - **Simulator** to execute attacks while design is soft and fix identified data leakage issues before it is finalized
- **Reverse Engineering -** Algorithmic intent derived from design
  - **Obfuscator** of key portions of the design to make interpretation impossible or economically impractical

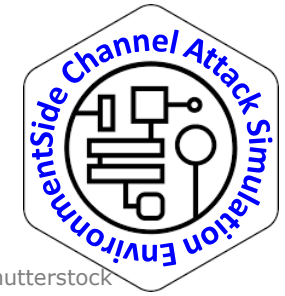
Source: exostivlabs.com

Has the design changed since trusted state?



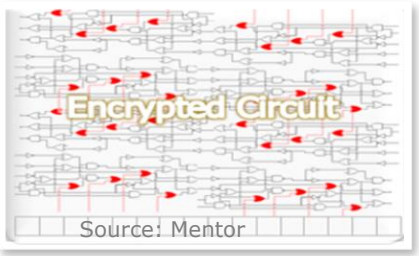
Source: stackexchange.com

Are there suspect circuits present in the design?



Source: Shutterstock

Does the design emit “secret” information?

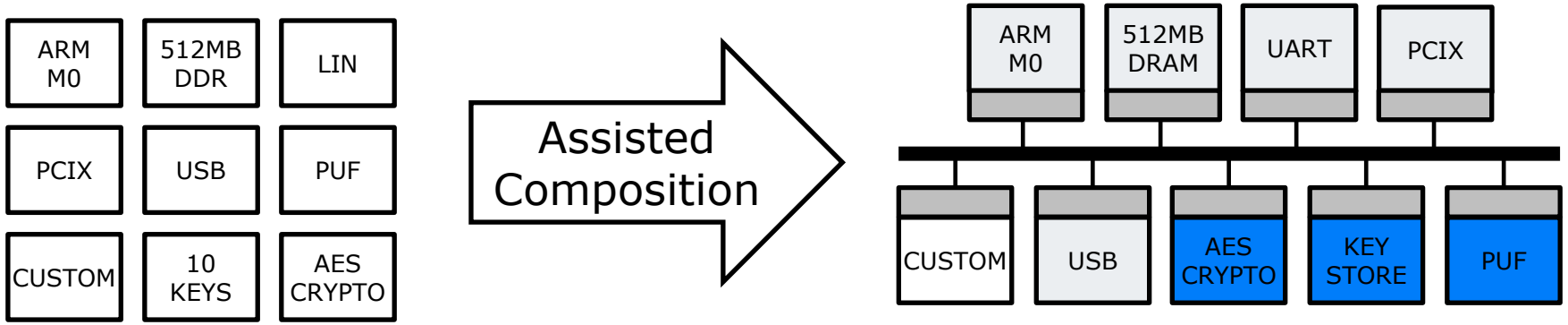


Source: Mentor

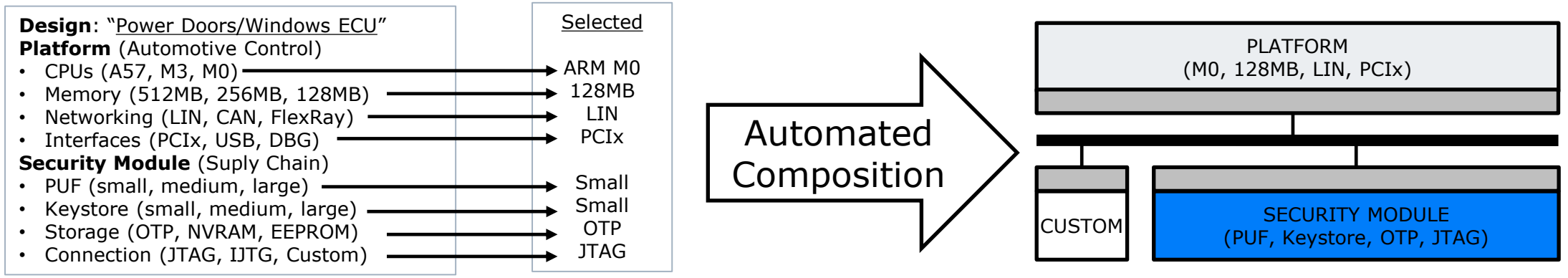
Can operational intent be derived from design files?

Assisted and Automated Composition

- **Assisted Composition** – Components are specified
  - Processor & security related components are user selected & automatically integrated



- **Automated Composition** – Configuration is specified
  - User selects a platform and provides configuration to a tool that automatically generates an integrated system



Optimized Composition

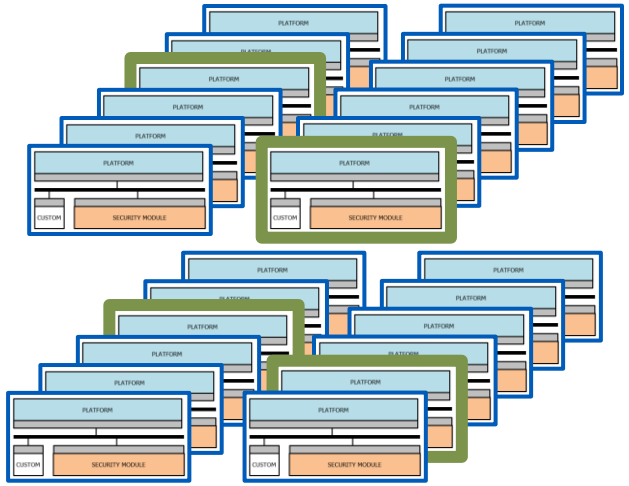
- **Optimized Composition** – Objectives are specified
  - User selects a platform and supplies a cost function with size, performance, power and security goals to guide combinatorial optimization to find **best architectures** which are presented to the user for assessment and selection

Design: "Power Doors/Windows ECU"

Platform (Automotive Control)

- Performance = 2
- Size = 9
- Power = 3
- Security = 3
  - Supply Chain = 7
  - Side Channel = 2
  - Reverse Engineering = 5
  - Malicious Hardware = 1

Optimized Composition



Combinatorial Optimization explores HUGE solution spaces (billions), but requires rapid estimation of "goodness"

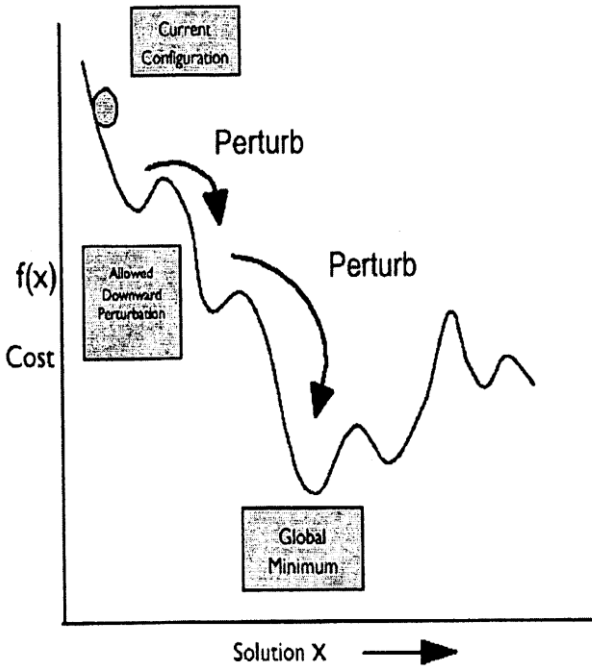
Performance and *Size* estimators are well understood and incorporated in modern tools

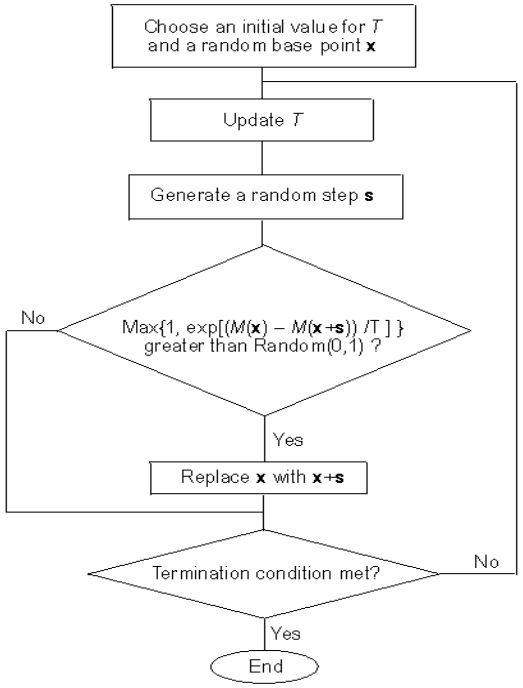
AISS will drive discovery of rapid estimation of **power** and **security**

$$f(a,b,c,d) = \sum(a*\textbf{Performance},\ b*\textbf{Size},\ \underbrace{c*\text{Power}}_{\text{estimate}},\ \underbrace{d*\text{Security}}_{\text{estimate}})$$

Optimization Cost Functions

f(a,b,c,d) = Σ(a***Performance***, b***Size***, c*Power, d*Security)





Source: The 80s
Point: *Technology for 2-dimensional optimization has been around for ~40 years*

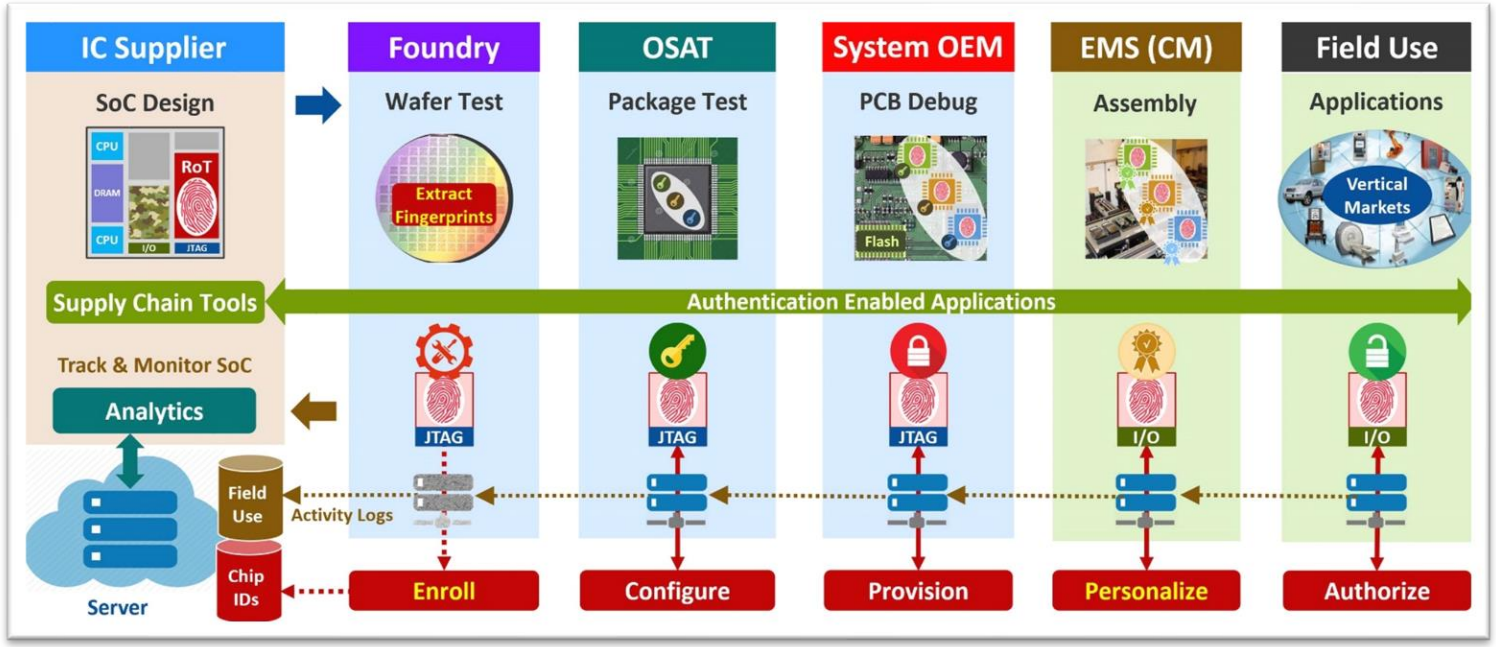
Cost Function Examples

Application	Perf.	Size	Power	Security
Lawn Sprinkler	2	7	9	1
Engine Control	6	5	1	3
Guided Projectile	5	1	9	7
Network Router	9	5	1	8
Mobile Phone	7	9	9	7
Smart Watch	3	6	9	3

Security Cost Function Expansion

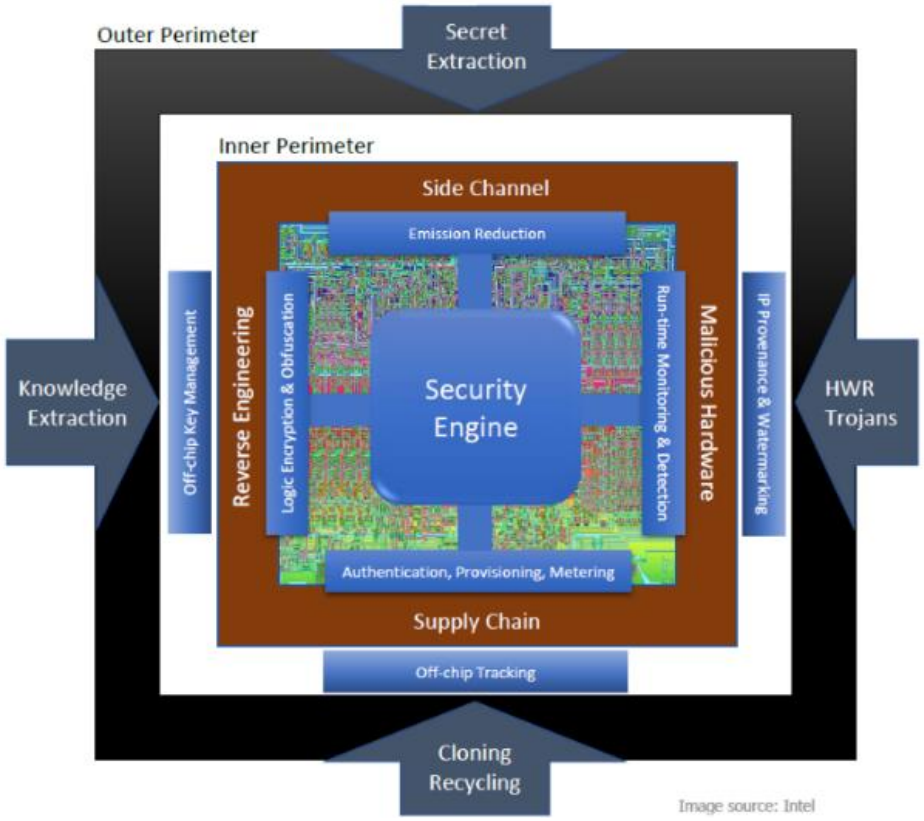
Application	Side Channel	Reverse Eng'g	Supply Chain	Malicious Hardware
Lawn Sprinkler	1	1	9	1
Engine Control	1	7	5	2
Guided Projectile	3	9	5	9
Network Router	9	7	8	9
Mobile Phone	8	9	9	6
Smart Watch	6	8	9	1

Enabling SoC Supply Chain Security



- **Design**: Create secure-reconfigurable SoCs with a unique ID based on an inborn Root of Trust
- **Enroll**: Extract chips unique ID into a secure server during first power up at wafer test
- **Configure**: Inject keys to encrypt, sign, or decrypt content for devices or end-applications
- **Provision**: Program SKUs downstream to reduce inventory risk and exploit volume ramp
- **Personalize**: Enables secure device identity during PCB assembly based on the chip's Root of Trust
- **Authorize**: Allow authorized parties to securely sign devices based on the SoC Root of Trust
- **Update**: Securely update firmware and provision SOC hardware features in the field
- **Monitor**: Track field use and evolve Big Data analytics on field failures, intrusions, counterfeits

Functional Design Security Summary



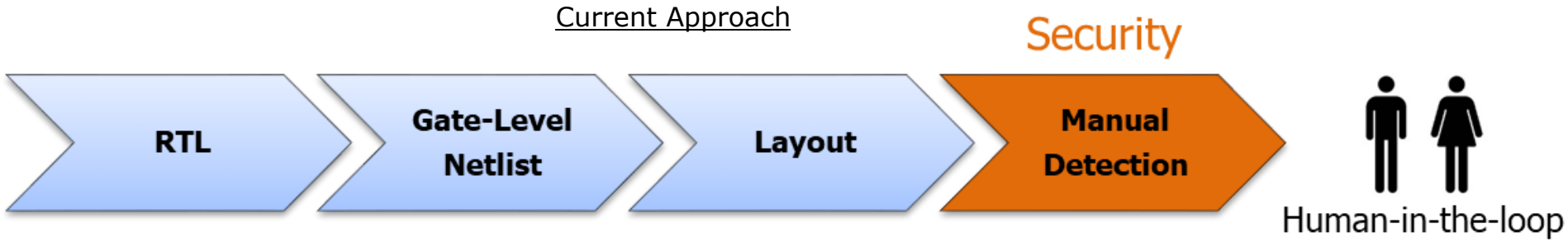
- Automated Security Inclusion/Integration
- As Easy as Design for Test
- Measurable relative "goodness"
- Scalable to secure any application
- Trusted and Zero Trust flows should be supported throughout the life cycle

Section 2/3

Securing Silicon in…

PHYSICAL DESIGN

Securing Physical Design from Post-silicon Attacks - *Present*



- Chips contain security assets that are vulnerable to exploitation at the physical design (layout) level
- Identification and mitigation of vulnerabilities is manual, difficult to scale to large designs, requires expertise, and lacks a formal process
- Detection of layout-based susceptibility to physical attacks requires elite expertise, is not scalable to modern design sizes, and cannot rapidly adapt to perpetually evolving threats

Security Challenges in Giga-size/Nano-scale Layouts

Deterministic identification

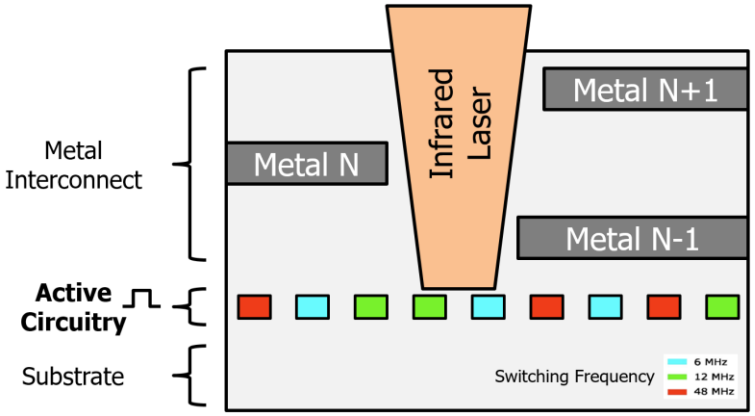
- Chip security assets' physical vulnerabilities cannot be determined until after the chip layout exists
- Comprehensive identification of vulnerabilities in billions of nanoscale structures
- Number of polygons can be 10B+, across many mask layers,
  - Hundreds of potential attack types on
  - Thousands of potentially exposed assets, resulting in a
  - Geometrically growing computational problem

Automated mitigation of with minimal PAS degradation

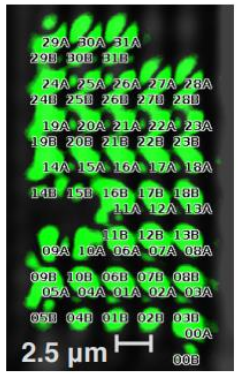
- Accomplished by rearranging the chip layout to inhibit physical, optical, & electrical access to assets
- Evaluation of billions of geometric possibilities and evaluation of the resulting security and PAS
  - Tight interdependence between the layout and corresponding PAS characteristics
  - Resulting in computationally difficult task of optimizing across so many variables

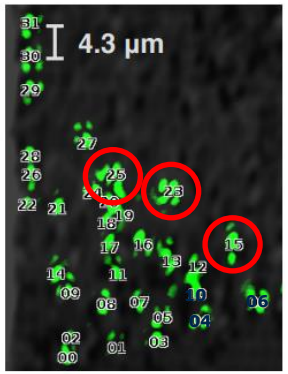
Physical Attack Surfaces

Optical Probing



Case Study: Optical probing enables extraction of encrypted FPGA bitstream on Xilinx Kintex 7



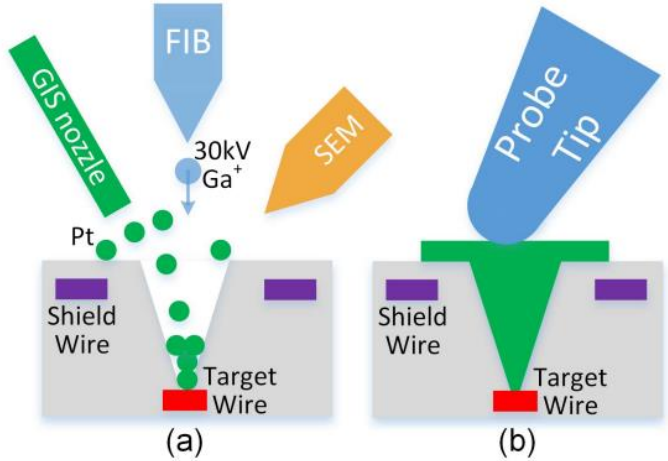


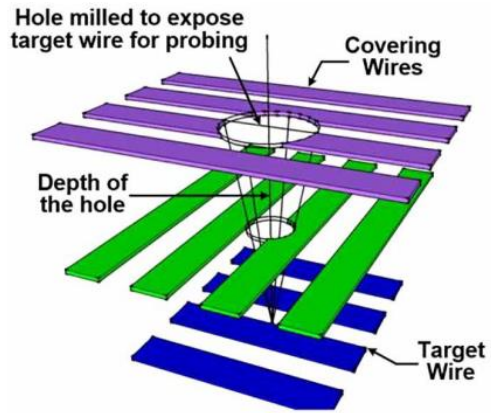
Localized logic locations storing decrypted bitstream

Decrypted plaintext values

[Source: S. Tajik et al. 2017 ACM SIGSAC Conference]

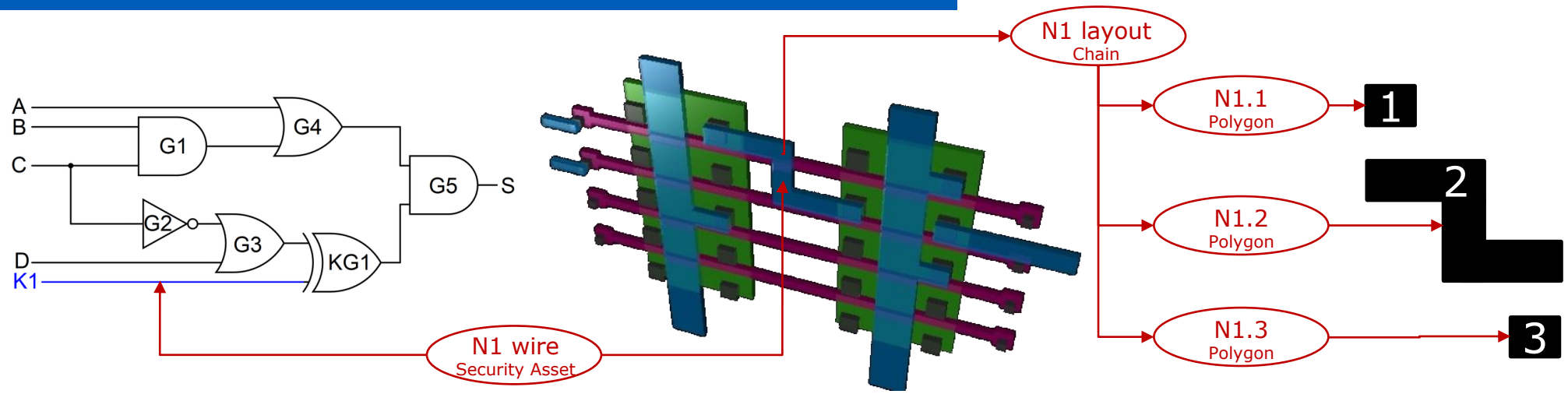
Fault Injection via Focused Ion Beam





[Source: H. Wang et al. 2019 IEEE Transactions on VLSI Systems]

Security Taxonomy



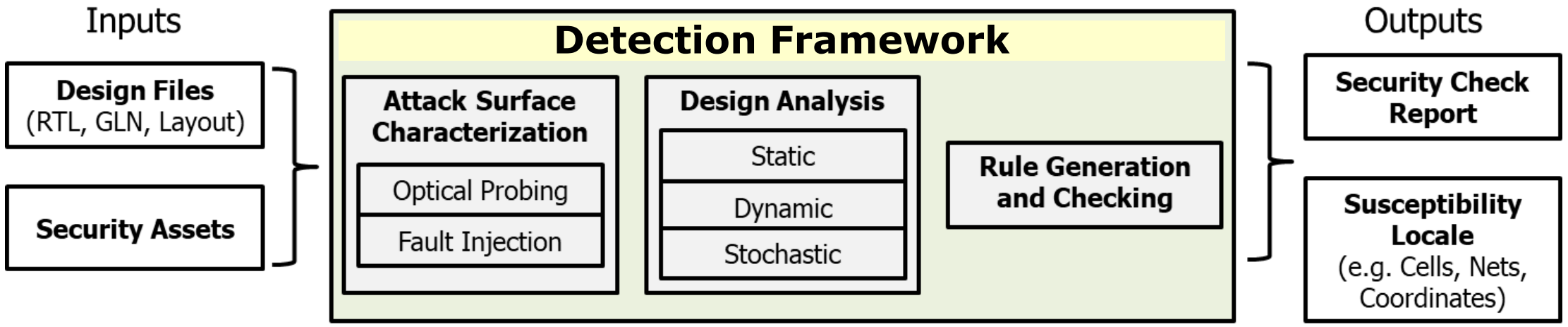
- **Security Asset** – an electrical object that can store or convey run-time data or control signals relevant for secure operation of the chip
- **Polygon** – a geometry representing a shape of a material implanted or deposited on a particular layer of a chip in a course of semiconductor manufacturing process
- **Chain** – a connected collection of polygons that represent a single security asset
- **Attack Surface** – a collection of strategies for accessing security assets with specific equipment
- **Attack Vector** – a tactic or sequence of tactics used to exploit and/or attack a particular surface

Protecting Data in Motion & at Rest

attack surface == ∑ **{vulnerabilities}**
vulnerability == ∑ **{security assets}**
security asset == ∑ **{structures}** spanning multiple layers
structure == ∑ **{polygons}** on a single layer
polygons store or convey **data**

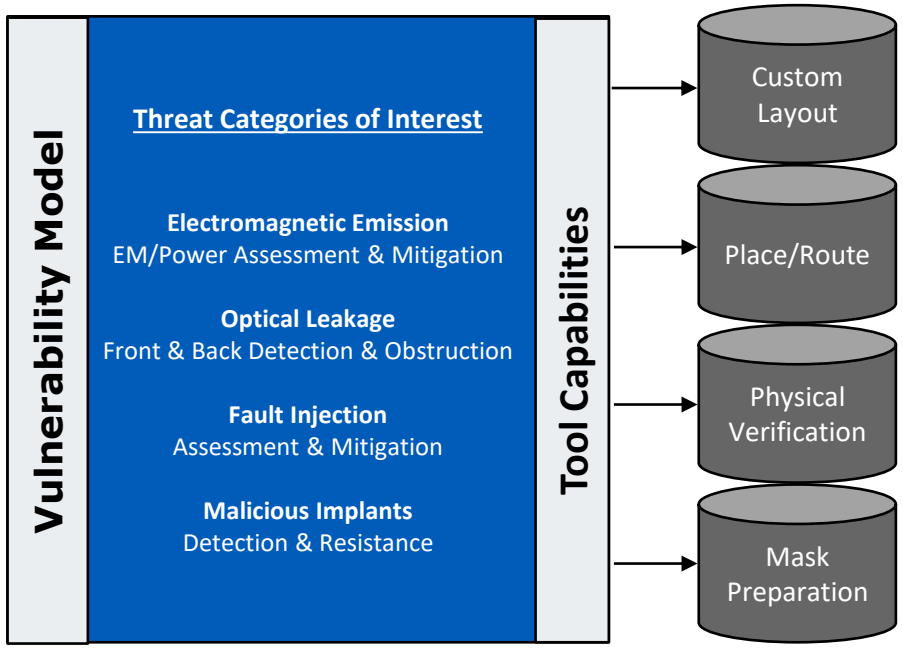
- **Detection -** assess **ACCESSIBILITY** of individual polygons
  - **Direction** == **{down | up}**
  - **Angle of Attack** == **{90 | 45 | 33}**
  - **Invasiveness** == **{destructive | non-destructive}**
  - **Access_Type** == **{read | read/write}**
  - ***Accessibility Metric*** *== function(Accessibility, Direction, Angle_of_Attack, Invasiveness, Access_Type);*
- **Mitigation -** reduce **ACCESSIBILITY** of polygons found in the Detection stage
  - Shielding – insert blocking shapes with no electrical, timing, or functional impact
  - Correction – stretch, move, or alter the surrounding polygons with no timing, or functional impact
  - Re-routing – produce routing directives for selective blocking with no functional impact
  - ***Blocking Metric*** *== % area of specified polygons blocked after mitigation*
- **Optimization -** minimize aggregate **IMPACT** of mitigations on PPA (Performance, Power, Area}
  - Objective – minimize **ΔPPA** while maximizing **Blocking** effectiveness
  - Optimization Strategy (for a possible simulated annealing approach)
    - Cost_function is driven by: min(degradation in PPA), max(improvement in Blocking)
    - Low disruption move: Shielding
    - Medium disruption move: Correction
    - High disruption mode: Re-routing
  - ***Optimization Metric*** *== 10% degradation in PPA with 90% of specified polygons blocked*

Securing Physical Design from Post-silicon Attacks - *Direction*



- **Vulnerabilities**: Comprehensive and extensible database of layout vulnerabilities
- **Detection & Mitigation**: Technologies detecting and mitigating all known vulnerabilities
- **Quantification**: Methodology to quantify layout hardness of chips with 20B+ transistors
- **Security Closure**: Framework for automated, iterative security closure

Physical Design Security Summary



- Comprehensive physical vulnerability dbs
- Novel strategies for vulnerability detection & mitigation
- Scale to giga-size/nanoscale chips: 20B+ transistors at sub-10nm nodes
- No human-in-the-loop, iterative reconciliation of security goals vs. PPA

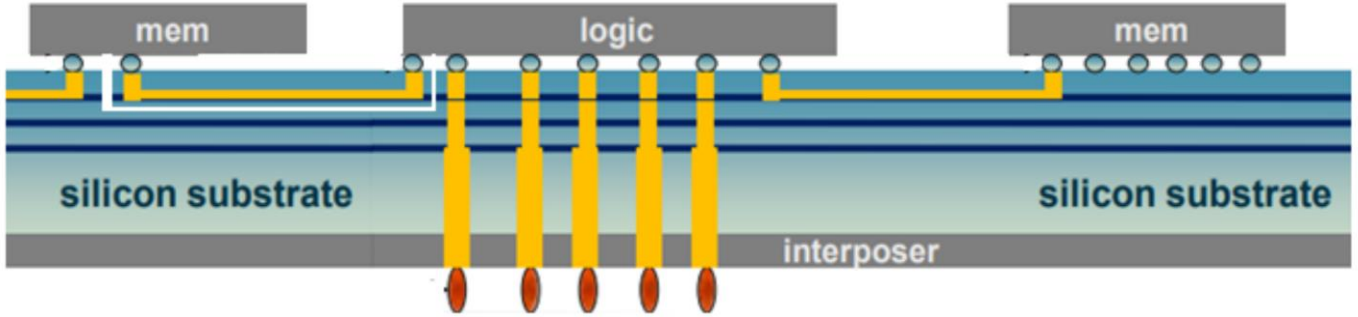
Section 3/3

Securing Silicon in…

HETEROGENEOUS INTEGRATION

(DISCUSSION LIMITED TO 2.5D WITH A SILICON INTERPOSER)

Securing 2.5D HI Design from Attacks



- Individual chiplets are subject to IC level functional and physical attacks
- Exposed signals traveling between chiplets and the interposer are exposed to physical probing and fault injection attacks
- If silicon interposer is active and holds security assets or mechanisms, it is MORE exposed to IC attack vectors due to larger geometries and fewer layers

[Image source: Surender Singh and Taranjit Kukal - Signal Integrity Analysis on High-Density Silicon Interposer Package Technology for Next Generation Applications]

HI Security Threats

- Tampered/compromised chiplets
- Compromised interconnections and interposer
- Physical attacks (X-ray, EOFM probing, direct probing and others)
- Side channels via interposers
- Covert channels among the chiplets

  (electrical, electro-magnetic, shared power line/converter, thermal)
- Chiplet firmware compromises
- DoS attacks (particularly 3D stacks)
- Run-time compromises of other kind

[Source: ASIC Alliance HI Security Working Group]

Challenges

- Development of 100% tamper-resistant or tamper-identifiable chiplets and interconnections for known exploits
- Development of pre-silicon analysis techniques in EDA tools that catch all of the package-level security threats before fabrication
- Development of facilities within the package that make the system inherently secure by detecting (during run-time) isolating compromised components to preserve partial or full functionality; this includes the detection and elimination of side and covert channels, the establishment of a fully-certifiable root-of-trust mechanism and prevention of any DoS
- Development of packaging techniques that provide complete immunity against all types of probing

[Source: ASIC Alliance HI Security Working Group]

Security Opportunities with Active Interposers

Attributes of a security-relevant interposer

- Fabricated in a trusted facility
- Design databases never visible externally
- Implements counter-measures to resist post-silicon attacks

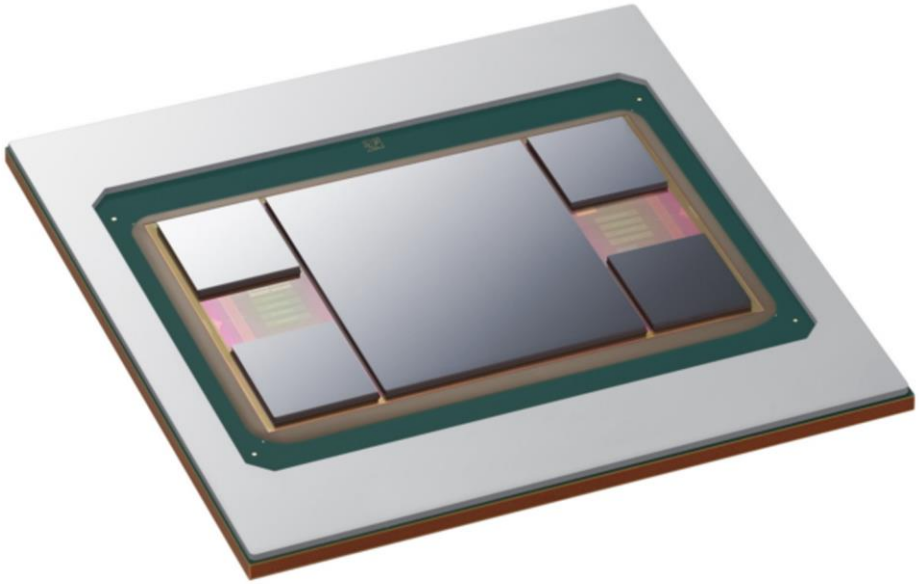
Top level of security hierarchy

- Play a security management and orchestration roles
- Unify and coordinate operations of security capable chiplets
- Host selected security-critical design modules (PUF, Key Registers, etc.)

Carrier of withheld security assets

- Implement portions of routing withheld from the chiplet manufacturers
- Manage injection of activation packages (routing tables, configuration bits, etc.)

HI Security Summary



- Identify vulnerabilities in newly exposed data in motion among tiles and interposer
- Protect active interposers from post-silicon attacks (invasive & non-invasive)
- Explore HI security architectures managed from the active interposer
- Utilize interposer as the holder of design elements intentionally withheld from fabs

[Image source: Samsung]

Q&A