

A Leading Provider of Microcontroller, Security, Mixed-Signal, Analog & Flash-IP Solutions



Securing the New Golden Age of Computer Architecture
Ted Speers, Head Of Product Arch & Planning
March 13, 2019



About Microchip FPGAs

Number One from Low Earth Orbit to Beyond Pluto

Rosetta

Orbits and Lands on Comet 2014



Legacy RT FPGAs

Pluto New Horizons

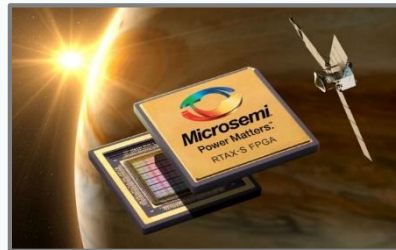
Pluto Images 2015



RTSX32SU, RTSX72SU

JUNO

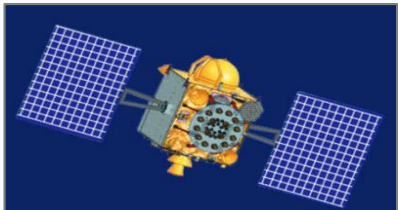
Entered Jupiter Orbit 2016



RTSX32SU, RTAX250S, RTAX2000S

IRNSS

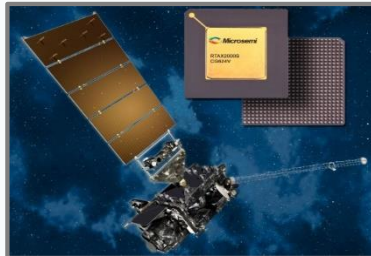
7 Satellites Launched 2013-2016



RTAX2000S

GOES-R

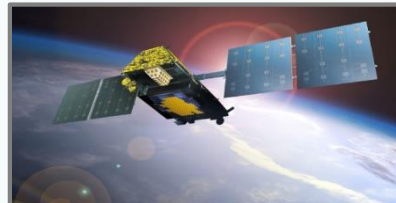
Climate Satellite Launched 2016



RTSX72SU, RTAX2000S

Iridium Next

First 10 Satellites Launched 2017



*RTSX32SU, RTAX250S, RTAX1000S,
RT3PE3000L*

Number One Above 30000 Feet



Airbus A380

- APA, A500K, SX-A, AX FPGAs
- Flight computers, cockpit displays, engine controls, power distribution, . . .



Boeing 787 Dreamliner

- APA, A3P, AX FPGAs
- Flight computers, cockpit displays, engine controls, braking, power distribution, cabin pressure, flight surface actuation . . .



Boeing 777-300ER

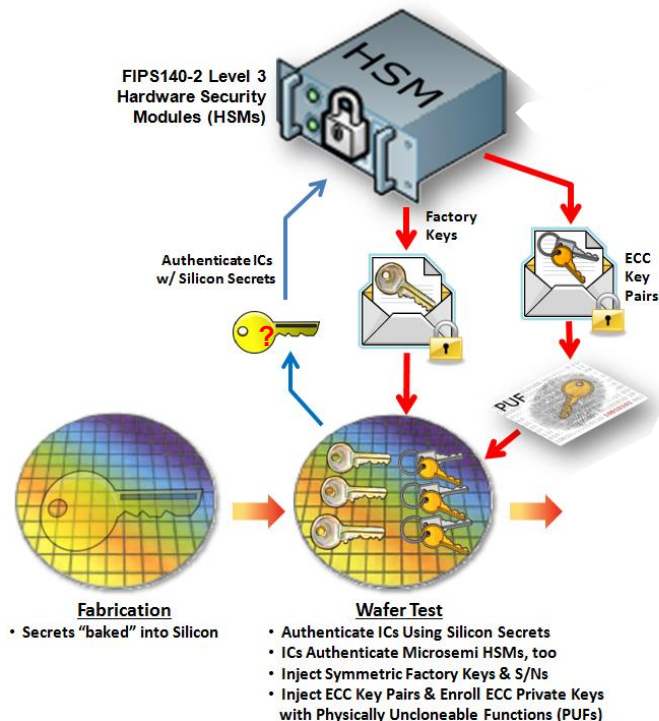
- A3P, Igloo2 FPGAs
- Flight computers, power distribution, engine controls, electronic control networks, flight surface actuation. . .



Airbus A350 XWB

- APA, A3P FPGAs
- Flight computers, cockpit displays, braking, engine controls, power distribution, cabin pressure, flight surface actuation . . .

Comprehensive Womb-to-Tomb Security Architecture



Award Winning PolarFire FPGA as an SoC platform

Lowest Power

Low static power technology
Power optimized transceivers
Up to 50% lower than SRAM
FPGAs

Proven Security

Defense-grade security
DPA safe Crypto coprocessor
Built-in anti-tamper

Exceptional Reliability

SEU immune configuration
Block RAM with ECC
Extended temperatures



**10G Bridging &
Aggregation**



**Video & Image
Processing**



**Portable
Equipment**



**Signal
Processing**



Control Plane



**Hardware
Acceleration**



**Low Power
Optics**



Who joins the RISC-V Foundation?



RISC-V Foundation: 200+ Members





RISC-V IP Providers





Academia & Research





EDA, IP and Support





Modern fabs





The New Golden Age of Computer Architecture

2017 Turing Award Lecture

A New Golden Age for Computer Architecture:

Domain-Specific Hardware/Software Co-Design,
Enhanced Security, Open Instruction Sets,
and Agile Chip Development

John Hennessy and David Patterson
Stanford and UC Berkeley
June 4, 2018

<https://www.youtube.com/watch?v=3LVeEjsn8Ts>

1



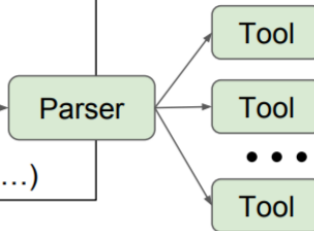
**Building a secure world
from the ground up**

Activity of Note: Formal Spec

What is an ISA Formal Spec?

A specification of the semantics of each instruction in the ISA, and of code execution:

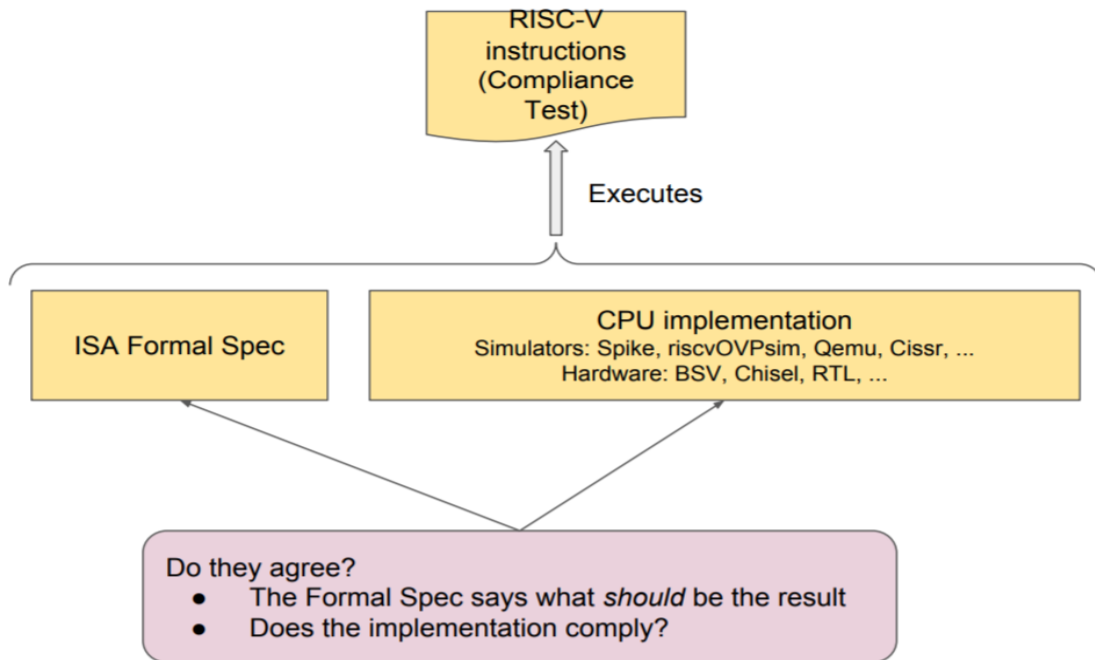
- Simple, clear and understandable to the human reader
 - Not cluttered with implementation considerations; no micro-architectural detail
 - Accessible to the practicing engineer: implementers of hardware, simulators, compilers, OSs, libraries of concurrent data structures, ...
- Precise and complete
 - (including allowed non-determinism of some instructions)
- Machine readable
- Executable (run RISC-V programs, boot an OS, compare with implems)
- Usable with various formal tools (theorem provers, model checkers, verifiers, ...)



English-language text specs, and instruction-set simulators (like Spike, riscvOVPSim, Qemu, Cissr, etc.) can be regarded as specs, but they typically do not meet many of these goals.

Activity of Note: Formal Spec

Of What Use is an ISA Formal Spec? Major use case: Compliance Testing



Activity of Note: Formal Spec

There are six efforts within TG Formal ISA, all quite advanced

(in free and open source repositories)

- **riscv-semantic**s: Adam Chlipala group at MIT
 - In Haskell, connecting to Coq formal tools in particular.

<https://github.com/mit-plv/riscv-semantic>
- **SAIL-RISCV**: Prashanth Mundkur and Peter Sewell group at U. Cambridge and SRI International
 - In SAIL DSL (domain specific language), which has also been used to model production ARMv8 (and others)
 - Has most experience in addressing concurrency.

<https://github.com/rem-s-project/sail-riscv>
- **riscv-formal**: Clifford Wolf
 - In Verilog

<https://github.com/cliffordwolf/riscv-formal>
- **GRIFT**: ("Galois RISC-V ISA Formal Tools") Ben Selfridge group at Galois
 - In Haskell

<https://github.com/GaloisInc/grift>
- **Kami**: Murali Vijayaraghavan group at SiFive
 - In "Kami", a DSL in Coq for HW description.

(hoping to publish soon)
- **Forvis**: ("Formal RISC-V ISA spec") Rishiyur Nikhil et. al. at Bluespec
 - In "Extremely Elementary" Haskell for extreme readability.

https://github.com/rsnikhil/Forvis_RISCV-ISA-Spec¹⁰

RISC-V Members Through a Security Filter

Defense Companies

BAE SYSTEMS

THALES

NORTHROP GRUMMAN

aselsan

Security IP

**inside
secure**

INTRINSIC ID

Rambus
Cryptography Research

SECURE RF
Securing the Internet of Things®

DOVER
MICROSYSTEMS

Chip Companies

NVD



LOWRISC

CHIP



NVIDIA

Micron

Security and Tools

**Tortuga
Logic**

**DATA
51**



Blockstream

galois

bluespec

INTRINSIX

Application Denied



Activity of Note: Security Standing Committee

Security Steering Committee Main Goals

- Promote RISC-V as an ideal vehicle for the security community
- Liaise with other internal RISC-V committees and with external security committees
- Create an information repository on new attack trends, threats and countermeasures
- Identify top 10 open challenges in security for the RISC-V community to address
- Propose security committees (Marketing or Technical) to tackle specific security topics
- Recruit security talent to the RISC-V ecosystem (e.g., into committees)
- Develop consensus around best security practices for IoT devices and embedded systems

Speaker Program: Gernot Heiser, Data61

Temporal Interference

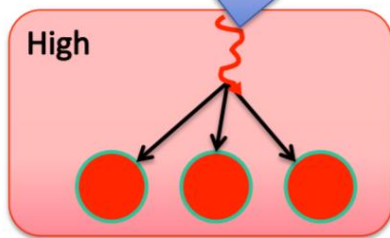
Safety: Timeliness

- Execution interference

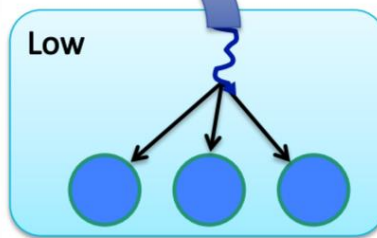
Security: Confidentiality

- Leakage via timing channels

Affect execution speed:
Integrity violation



Low



Observe execution speed:
Confidentiality violation

Timing Channels

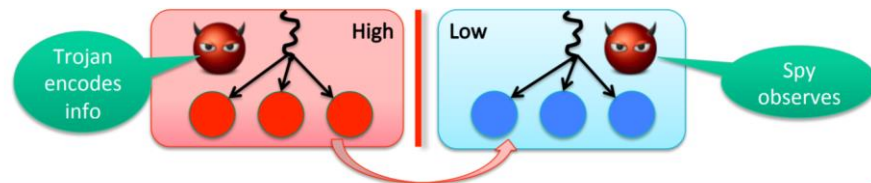
Covert Channel

Definition:

- Information flow that bypasses the system's security policy

Note:

- CC exploits mechanisms not intended for information transmission



3 | RISC-V SSC | July'18 | Gernot Heiser

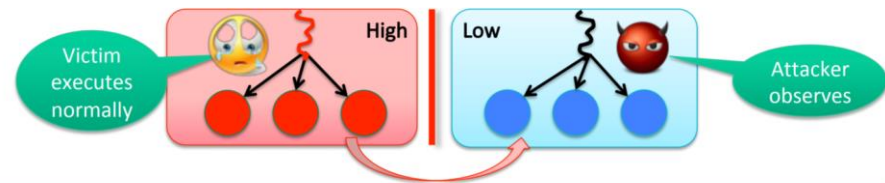
Side Channel

Definition:

- Covert channel that can be exploited without insider help

Note:

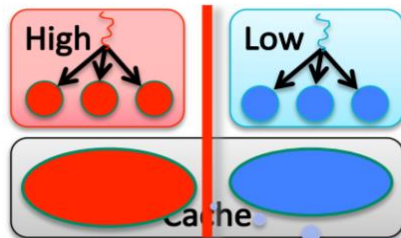
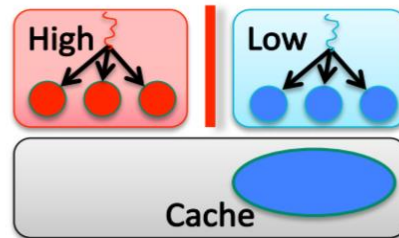
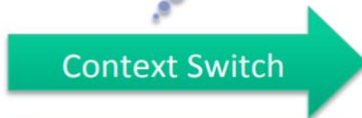
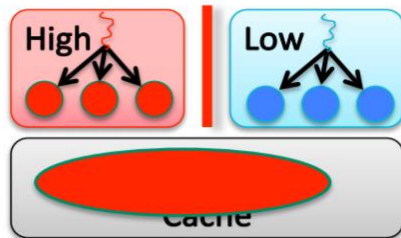
- Attacker probes for traces left by the execution of an innocent victim



4 | RISC-V SSC | July'18 | Gernot Heiser

Mitigating Timing Attacks

Mitigation: Prevent Sharing of State



Flushing useless for concurrent access

- between harts
- between cores

Cannot partition on-core caches (L1, TLB, branch predictor, prefetchers)

- virtually-indexed
- OS cannot control access

Partition through page colouring

New Hardware-Software Contract!

Need New Hardware-Software Contract!



- The ISA is a purely functional contract
 - sufficient to ensure functional correctness
 - abstracts away time
 - insufficient for ensuring either timing safety or security
- For security need an abstraction of microarchitectural state
 - essential for letting OS provide time protection

Remember: Timing channels can be closed *iff* all shared hardware state can be partitioned or flushed

Augmented ISA

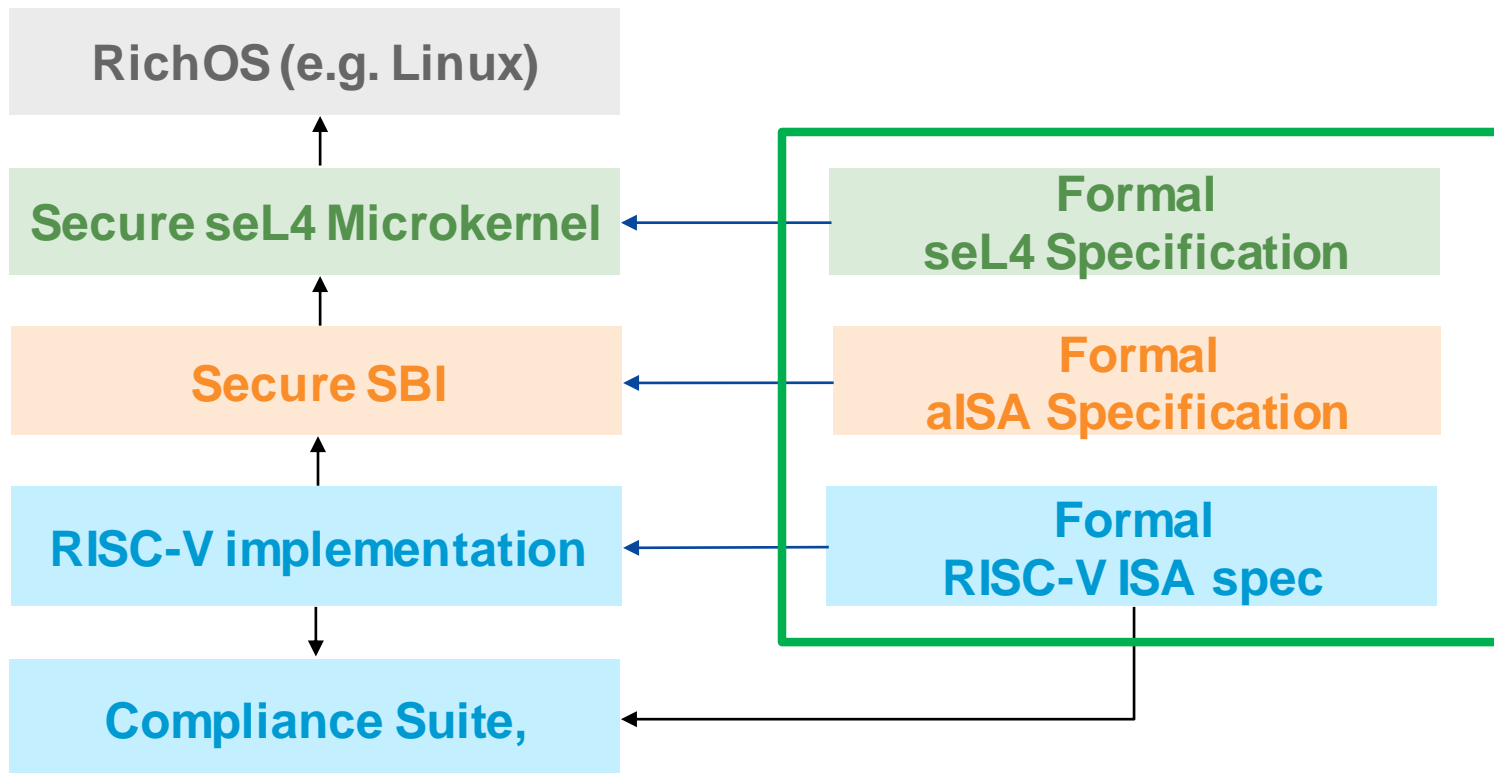
New Hardware-Software Contract: AISA



Augmented ISA must provide abstractions that support time protection:

1. Identify partitionable state and how to partition
 - Generally physically-addressed caches, memory interfaces
 - Mostly there, just make it part of the contract
2. Identify existence of non-partitionable state and how it can be flushed
 - Can probably lump all on-core state into single abstraction
 - A single flush-on-core-state operation may be sufficient

Putting it all Together: The RISC-V Security Stack





**Start creating a secure future today
with Microchip and RISC-V**

PolarFire SoC

RISC-V-based SoC FPGA

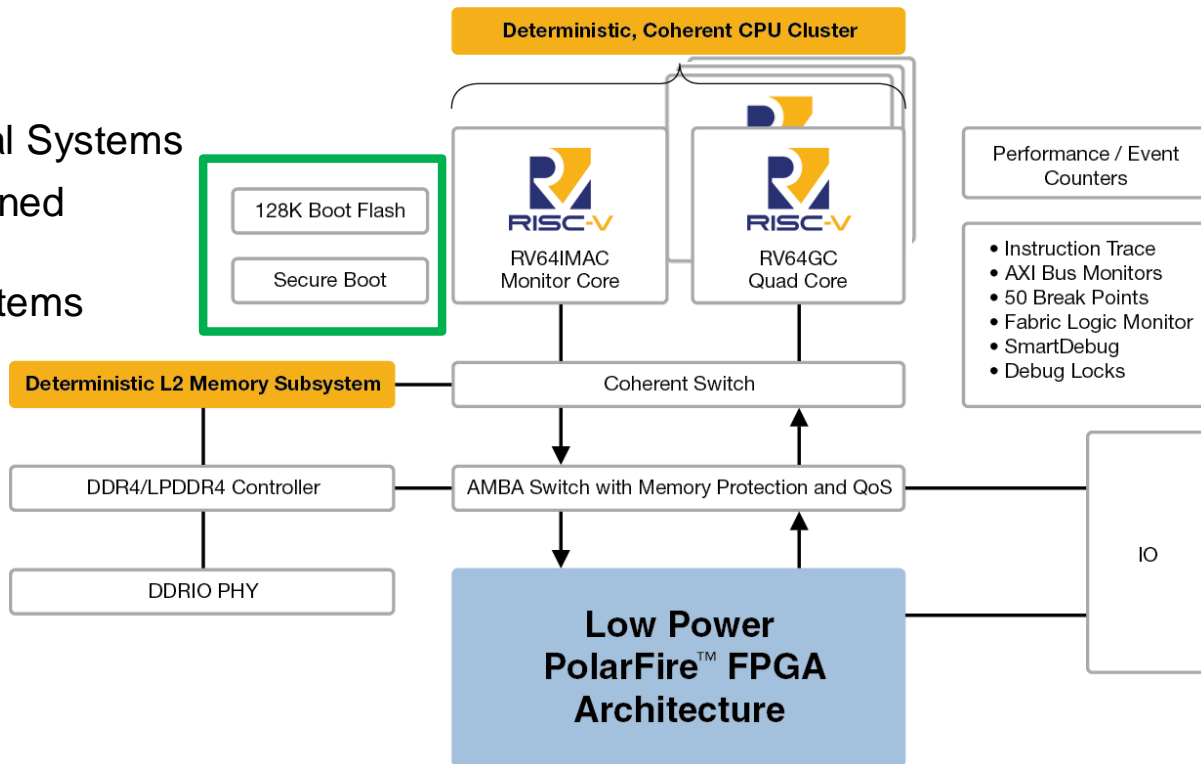
Freedom to Innovate in

Linux and Real-Time

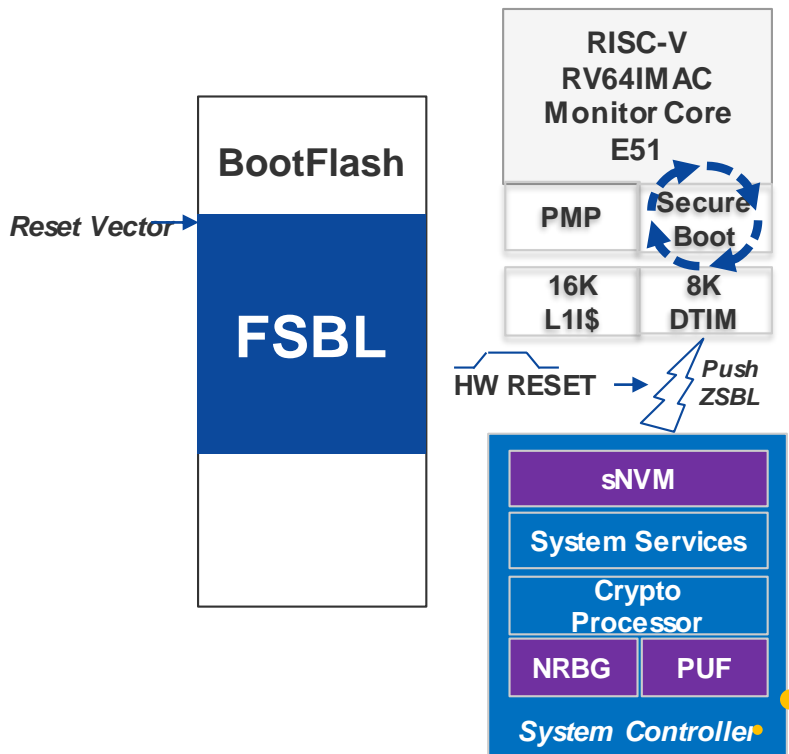
High-Reliability Safety Critical Systems

Thermal and Power Constrained Systems

Securely Connected IoT Systems

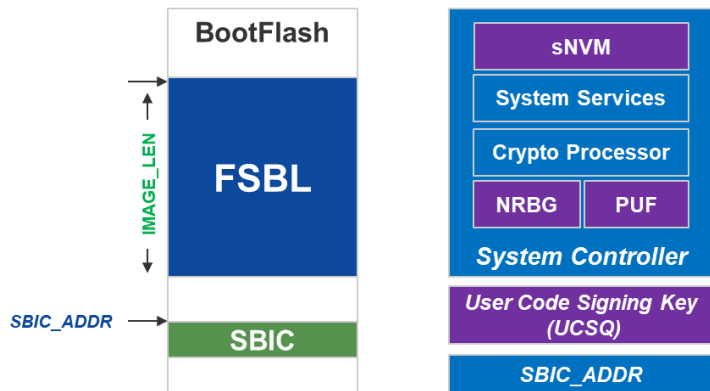


Secure Boot



- Guards against sophisticated methods of attack whereby a malicious external agent tampers with the boot image stored in bootflash (e.g Linux FSBL)
- Authenticates the image in bootflash before transferring execution control to the OS boot loader pointed to by reset vector
- **FPGA system controller (root of trust) manages the authentication process and certifies boot image using crypto functionality built into the FPGA backbone**
 - Push “zero state boot loader” (ZSBL) upon detecting HW reset.
 - Release monitor core from reset and executes authentication on FSBL image pointed to by reset vector.
 - If authentication is successful, transfer execution control back to FSBL, otherwise abort.

Authentication Framework



Value	Description
IMAGE_ADDR	Address of FSBL in SOC Memory map
IMAGE_LEN	Size of FSBL in Bytes
BOOT_VEC₀	Boot Vector in FSBL Monitor Core
BOOT_VEC₁₋₄	Boot Vectors for User Cores
H	FSBL Image Hash (SHA512-256)
CODESIG	SBIC Digital Signature (ECDSA)

- **ZSBL bootloader authenticates FSBL image in bootflash which contains:**
 - Actual FSBL image
 - SBIC data structure generated during bootflash programming and stored @ SBIC_ADDR
- **Authenticity of SBIC is verified by FPGA system controller using ECDSA:**
 - UCSQ is a public key programmed on the device by the user
 - Corresponds to UCSK private key used to sign the SBIC during programming

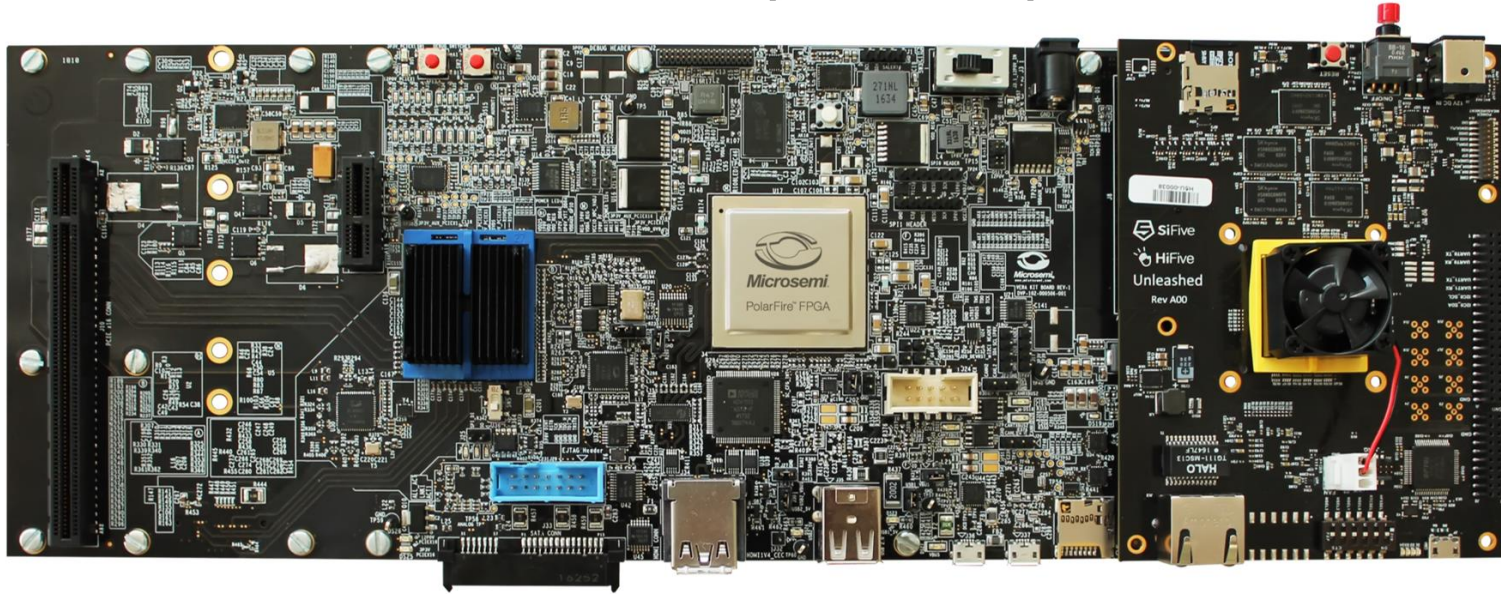
$$\text{CODESIG} = \text{ECDSA}_{\text{SIGN}} (\text{UCSK}, \text{IMAGE_ADDR} \mid \text{IMAGE_LEN} \mid \text{BOOTVEC}_{0-4} \mid \text{H})$$

$$\text{ECDSA}_{\text{VERIFY}} (\text{UCSQ}, \text{IMAGE_ADDR} \mid \text{IMAGE_LEN} \mid \text{BOOTVEC}_{0-4} \mid \text{H}, \text{CODESIG})$$

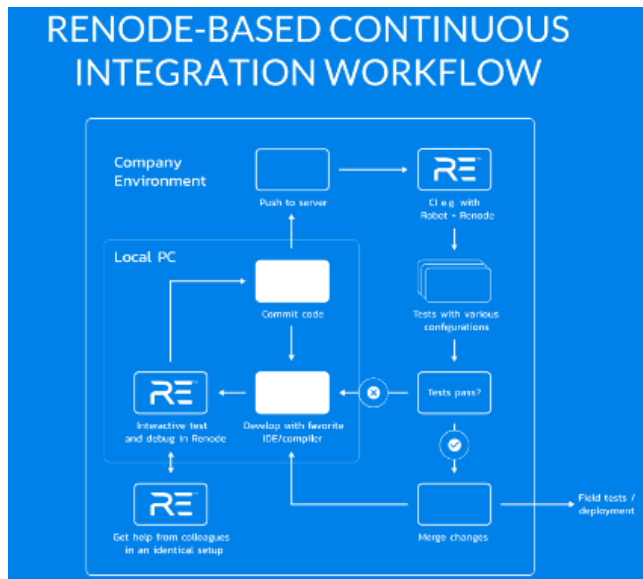
*Elliptic Curve Digital Signature
Algorithm (ECDSA)*

Freedom to Begin Hardware Development

PolarFire SoC Embedded Experts Development Platform



Freedom to Start Software Development

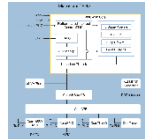


RENODE™

- Free Rapid Software Development and Debug Capabilities without Hardware
- Complete PolarFire SoC Processor
- Subsystem Model



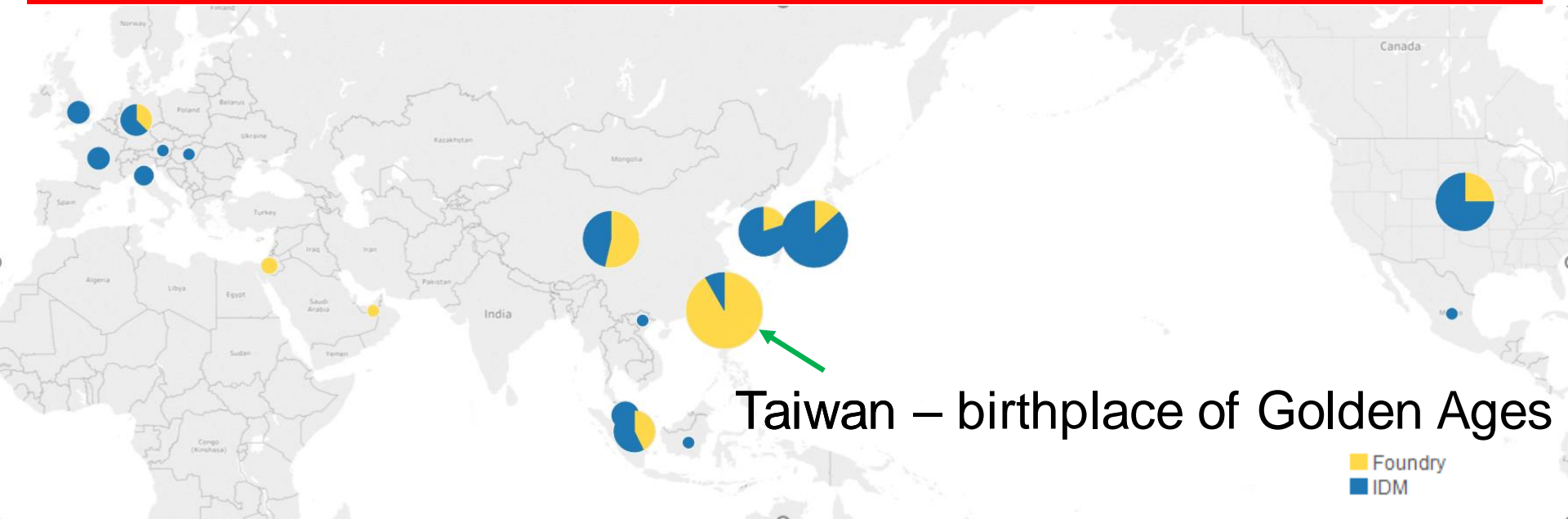
Building Out the Mi-V RISC-V Ecosystem



Where IDMs have Fabs



Where foundries have their fabs



Taiwan – birthplace of Golden Ages



MICROCHIP

Thank You