Specific details purposefully withheld due to sensitivity issues

Microprocessors for Roots-of-Trust

Kristopher Carver

Technical Director

BlueRISC, Inc.

HotChips 2013 – August 27th

Current Target Systems

- **High value Defense systems**
  - **Contain security critical algorithms**
  - **Embedded systems**
  - **Primarily FPGA-based**
    - Hybrid processor/hardware designs
- **High security commercial systems**

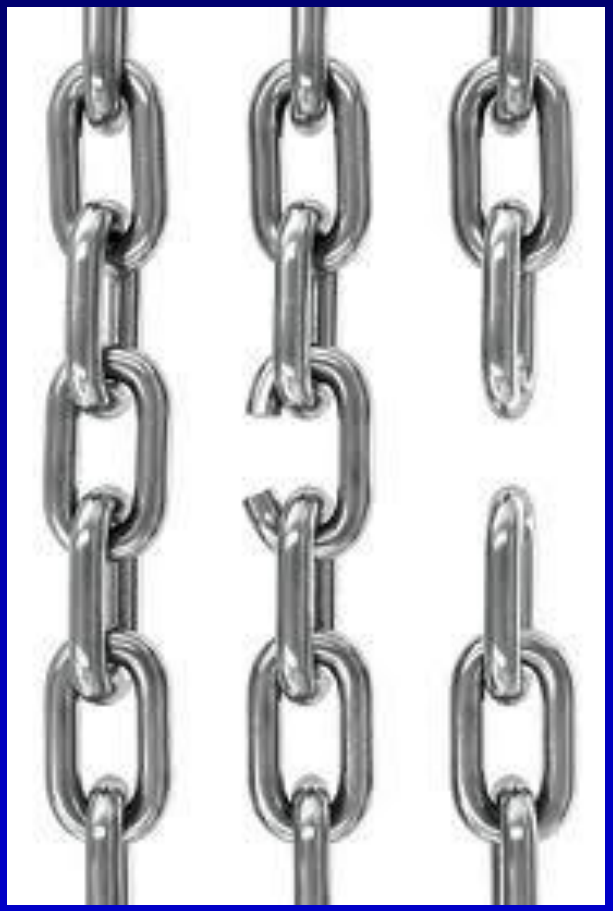
Traditional Security Mindset

- **Based on conventional software and hardware**
  - Operate on a known ISA
  - Susceptible to:
    - **Software reverse engineering and modification**
    - **Fault injection**
    - **Power analysis (DPA, SPA, PEA, etc.)**

- ***Attack surface not actually reduced***
  - **In many cases, it is increased through simply patching security holes**
  - **Typically through new software**

Goal to Achieve Roots-of-Trust

- **Break attackers assumptions**
  - ISA Encoding to Operation
  - Power profile to Execution
  - Timing
- **Securely root system-level security approaches**
- **Increase time and cost for attack**

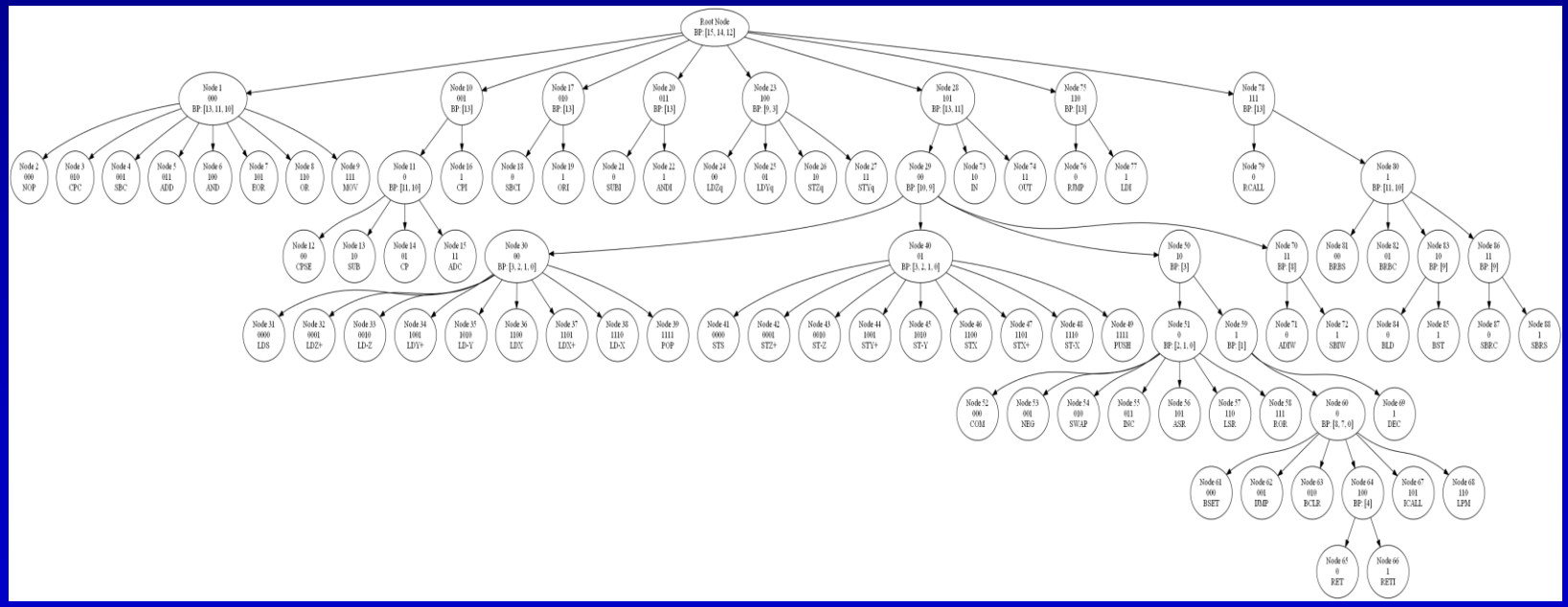


Microprocessors for Roots-of-Trust

- **Address security without introducing new attack surfaces**
- **Move Roots-of-Trust into the architecture itself**
  - Core Root-of-Trust
    - **Randomized ISA Support**
    - **ISA support for power and timing control**
    - **Secure Storage**
  - Shared Roots-of-Trust
    - **Root-of-Trust for Co-Execution**
    - **Root-of-Trust for Key Management**
    - **Root-of-Trust for Measurement**

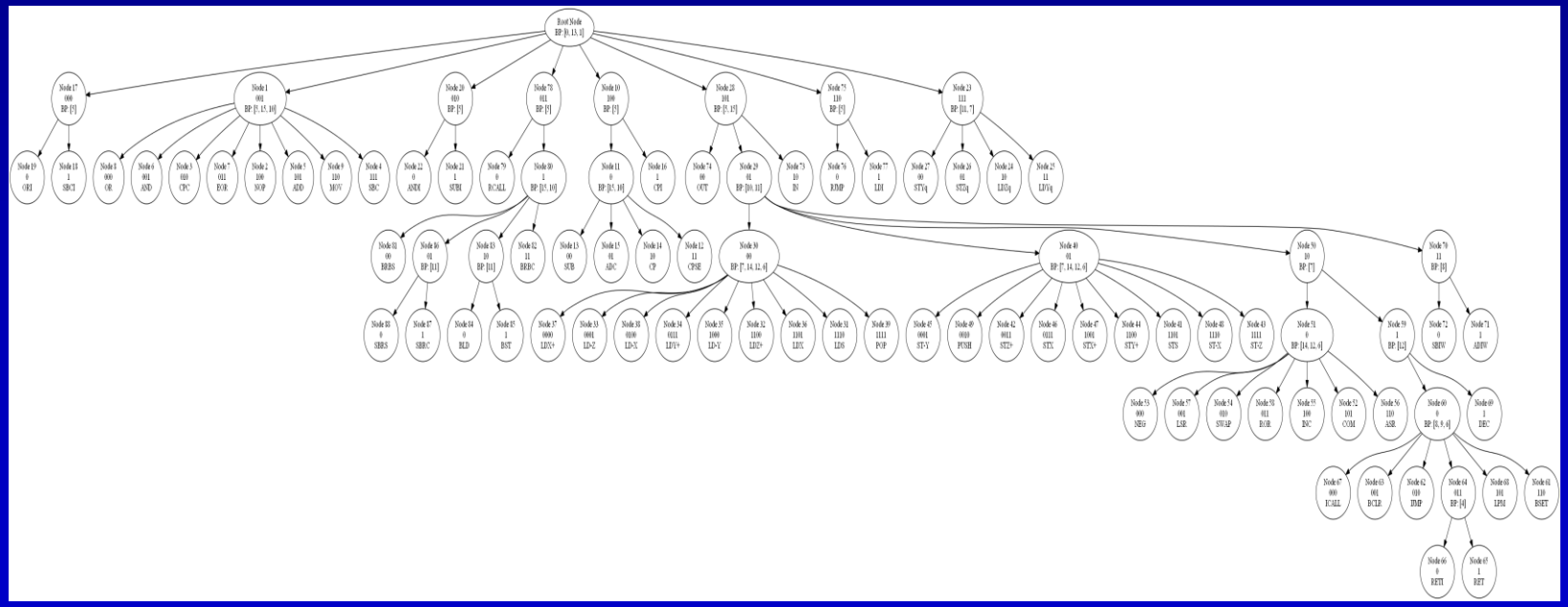
ISA Definition and Encoding

- ISA can be represented by a tree structure
- Tree defined by bit positions and meanings



ISA Definition and Encoding

- Transformations to meanings and positions
- Underlying operations do not change but new ISA



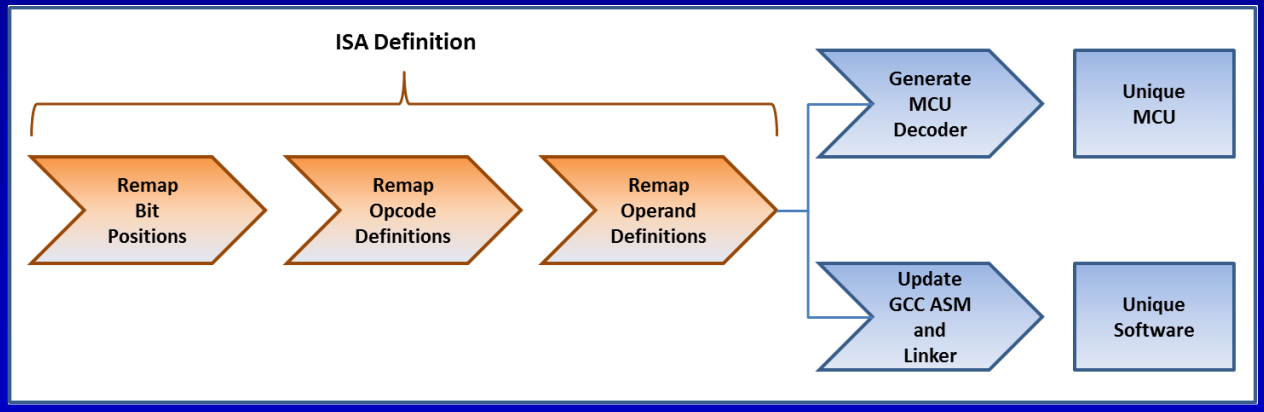
Unknown ISA Attack Methodology

16-bit ISA complexity estimate ≈ 340 bit key + time consuming trials

1. **Define Test ISA**
2. **Create simulator**
3. **Run target code**
4. **Determine if results "make sense" → Step 1**

tinyTrustGUARD

- 16-bit processor architecture
- Per-compilation processor with unique ISA
- Software toolkit that generates both software encoding and unique microcontroller/ISA



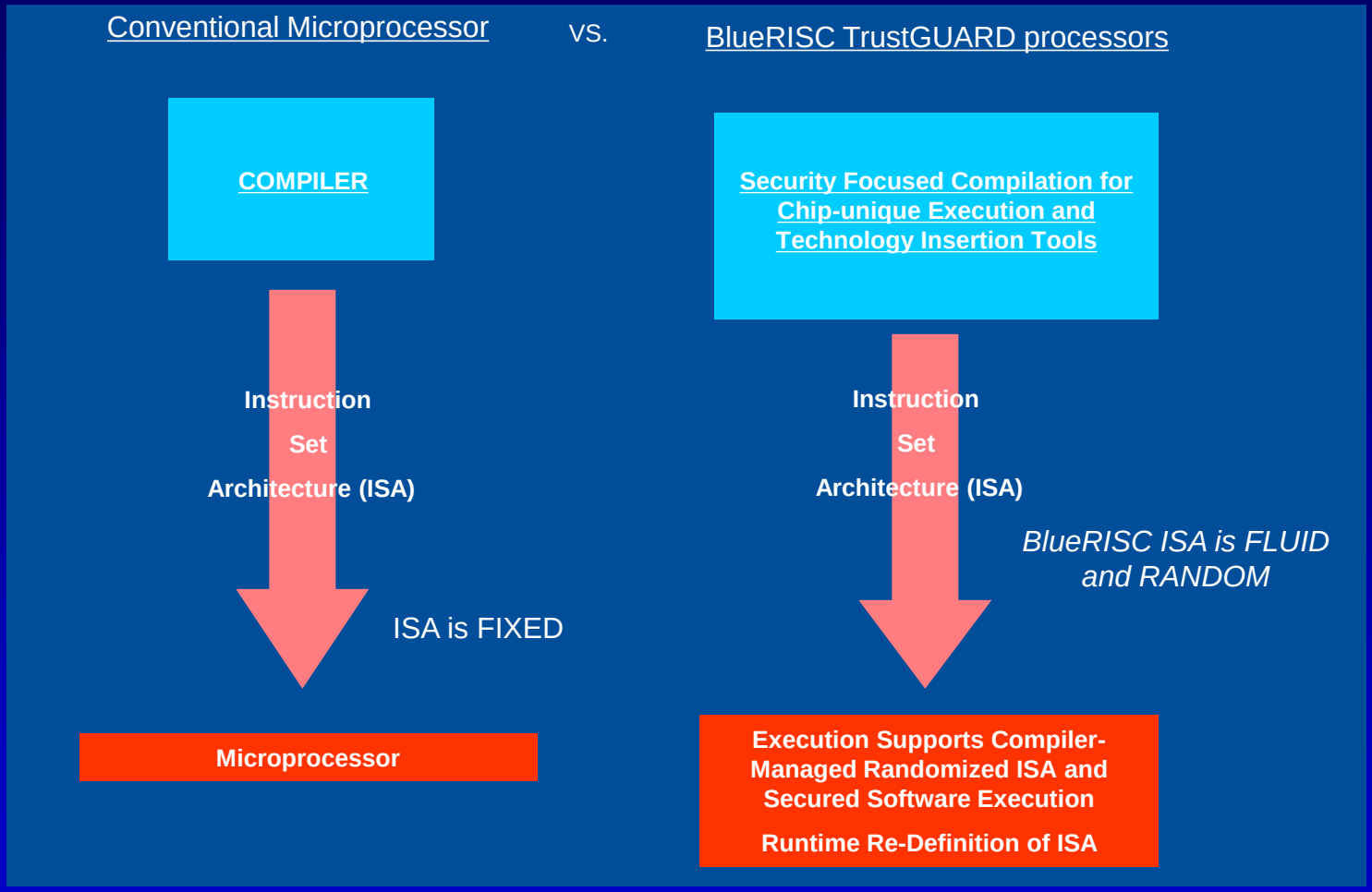
Make it device-unique

Compilation-unique ISA is great but you can also make it device-unique!

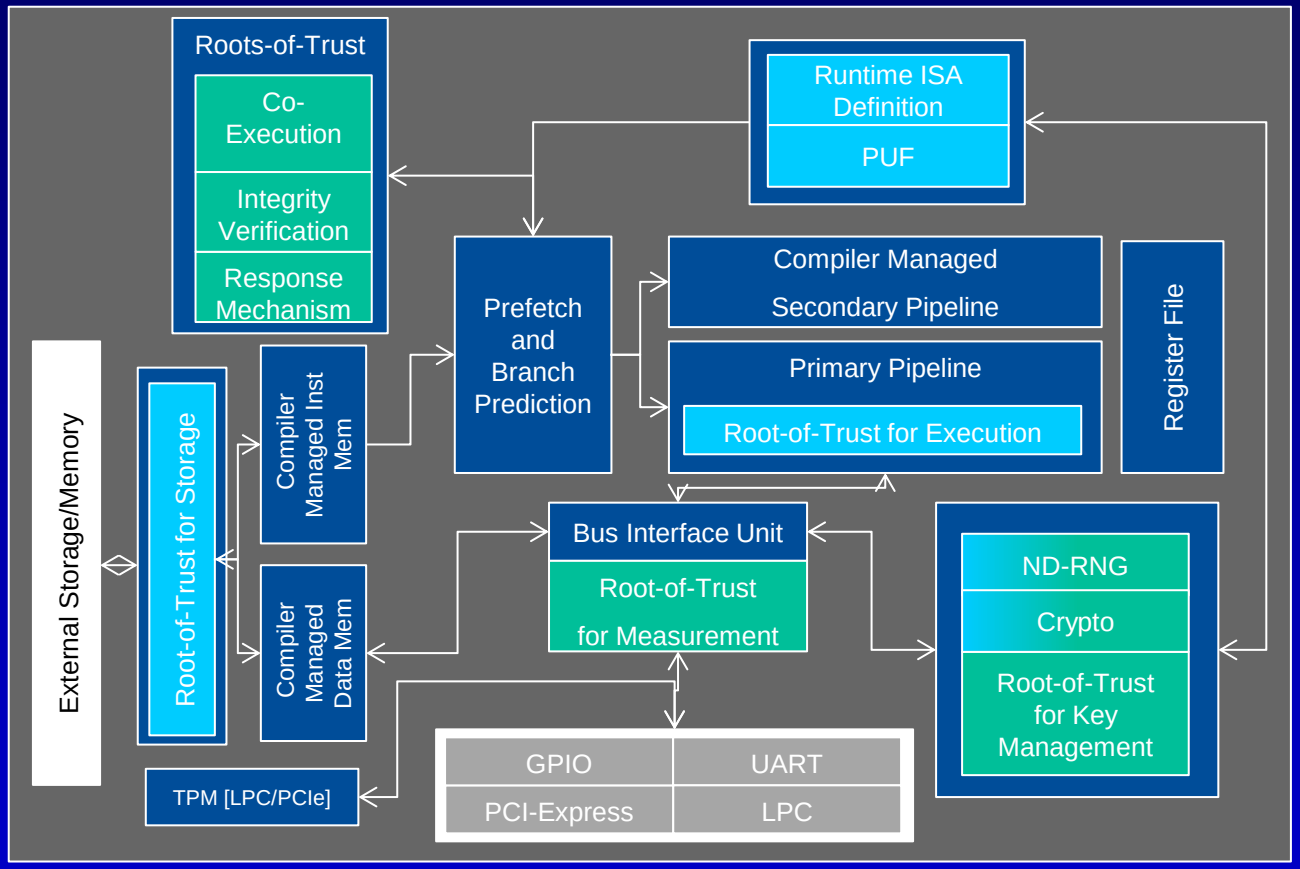
TrustGUARD Secure Processor

- Enables device-unique randomized ISA and execution environment
  - Ties ISA encoding and execution to physical properties
- Adds fluidity to ISA encoding/decoding
  - ISA supported; enabled by compilation approach
- Extends tinyTrustGUARD to 32-bit, dual issue core
  - Extends shared Roots-of-Trust to external systems
- Cryptographic Acceleration
  - DPA resilient AES128/256, SHA1/256, HW-accelerated RSA with Montgomery Multiplier, etc.

Conventional vs. TrustGUARD

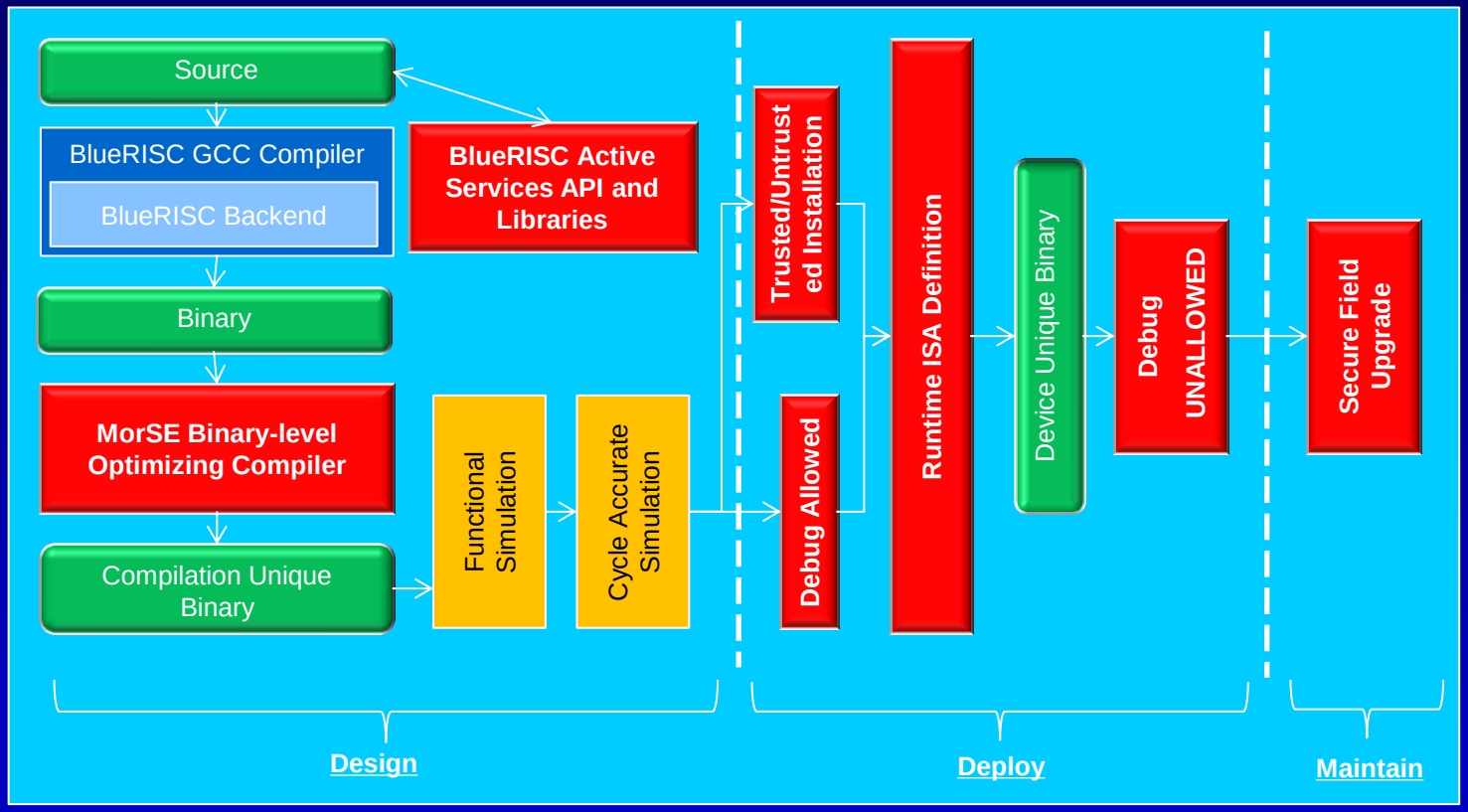


TrustGUARD Roots-of-Trust Diagram



More information can be provided in proper setting

Tooling and Infrastructure



: Security-Focused steps

Performance Implications

- tinyTrustGUARD
  - Parametric decoder results in minimal, cycle-time overhead
- TrustGUARD
  - Minimized through architectural support
  - Compiler managed fluidity control
  - 1-time runtime ISA re-definition
  - <1% performance impact relative to compilation-only techniques
  - 1.5 DMIPS/MHz with new mode of secure execution

More information can be provided in proper setting

Integration Options

PCI-Express Add-on Card

ExpressCARD Form-Factor

Embedded IP

- **Instantiated in FPGAs**
- **Completely synthesizable core**
- **TrustGUARD as main processor or security processor**

- **ASIC flow developed and available**

Questions?

Web: http://www.bluerisc.com
Email: sales@bluerisc.com
Phone: (413) 359-0599