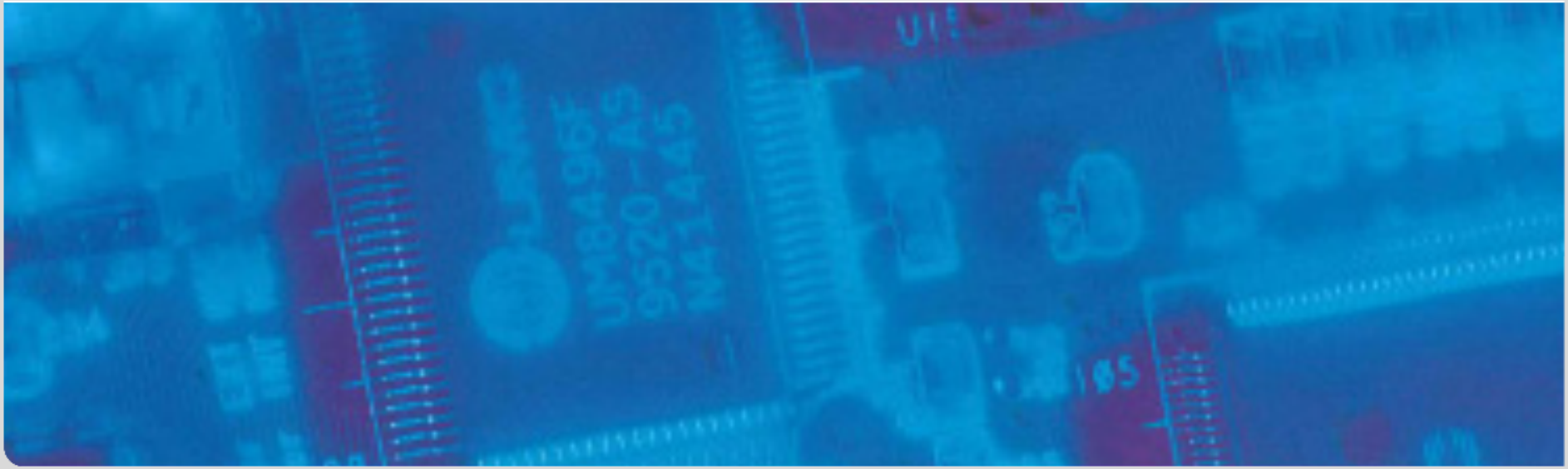
Hardware and Platforms

An overview of SPP-1500 projects

Mehdi Tahoori

Karlsruhe Institute of Technology

SPP1500.itec.kit.edu



Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

Scope of the Projects

	Device and Circuit	Microarchitecture RTL	Architecture and System
Permanent Failures		OTERA	CRAU
Transient Errors	PERICES	PERICES, OTERA, GetSURE	CRAU, GetSURE
Aging Effects		Ambrosia, OTERA, GetSURE	Ambrosia, GetSURE, Term3D

	Device and Circuit	Microarchitecture RTL	Architecture and System
ASICs	PERICES	PERICES	
Processors	PERICES	PERICES, Ambrosia, GetSURE	Ambrosia, GetSURE
Reconfigurable Fabric		OTERA	
Heterogeneous System		OTERA	CRAU, Term3D

Projects

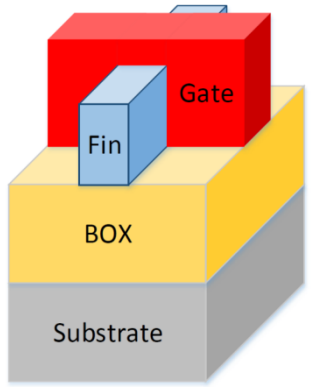
- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

PERICES Overview

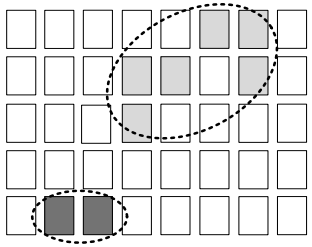
- **Goal:** Technology relevant cross-layer soft error modeling and mitigation

- **Challenges:** Soft Errors are technology and workload dependent
  - Error generation:
    - FinFET vs. Planner
    - MET error-site identification
  - Error propagation:
    - Accurate electrical masking model for nanoscale technologies
    - Correlated error propagation
    - Modeling error propagation in systems in regular (reg file, cache) and irregular (random logic, controller) structures
  - Cross-layer soft error estimation
    - Considering both error generation and propagation
    - In a unified platform

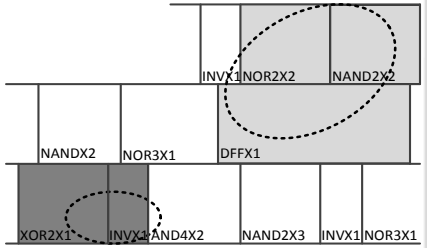
Error Generation



- FinFET Technology
  - Immunity to short channel effects
  - More 3-D compared to MOSFET
- **Error Generation at SRAMs**
  - Physical-level (Geant4) : Passage of particle in the matter
  - Circuit-level (SPICE): SRAM characterization for different parasitic current
    - SRAM cell failure probability (SEU vs. MBU)
- **Error Generation at Logic**
  - Remarkable fraction of particle strikes results in Multiple Transients (MTs) [Harada11IRPS][Rossi05DFT]
  - **Our method:** MT patterns extracted from MBU patterns
  - Using fast and accurate layout analysis (generation)
  - Projecting the effect at the netlist level (propagation)



MBU patterns in SRAM

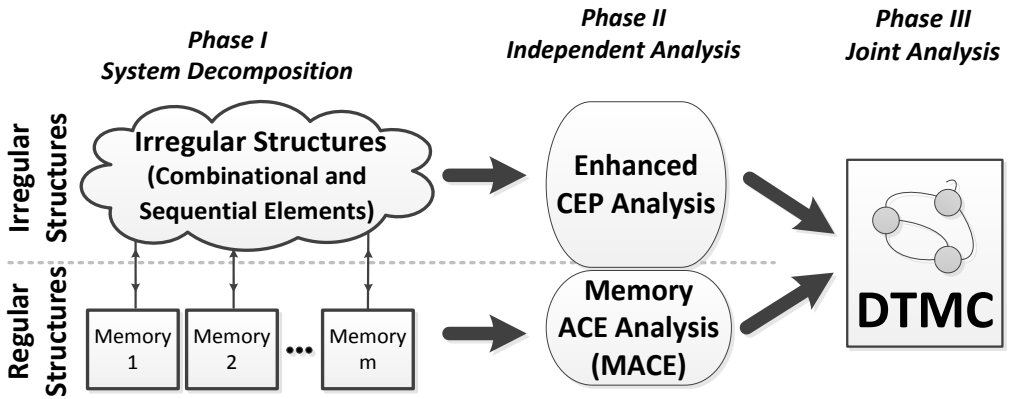


Equivalent MT error sites in circuit layout

Error Propagation

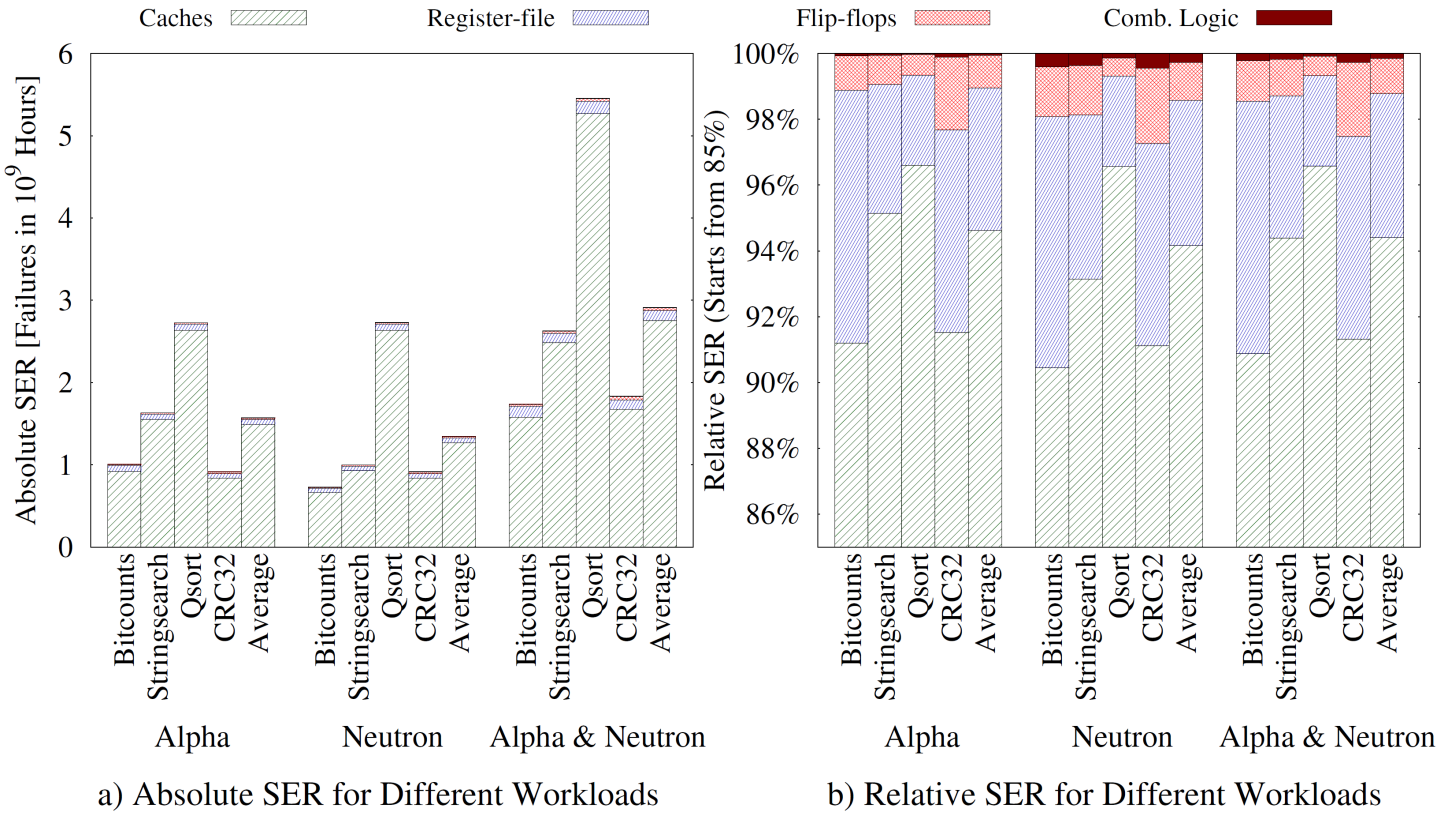
- **Single-cycle error propagation at circuit-level**
  - Electrical Masking [Kiamehr13VTS]
    - In 45 nm and beyond: Voltage fluctuation >10% → affects electrical masking
    - **Proposed:** Trapezoidal LUT-based model → 99.7% accurate compared to SPICE
  - Correlation in logical masking [Chen13JETTA]
    - Correlated error propagation considering input and internal correlation
    - Very useful for error abstraction at higher levels
- **Multi-cycle error propagation at architecture-level**
  - **ACE analysis** → Cannot model circuit-level masking
  - **Circuit-level error propagation** → Cannot model architecture and application-level masking
  - **CLASS: Combined Logic Architecture Soft error Sensitivity analysis** [Ebrahimi13ASPDAC]
    - Discrete time Markov chains

    for handling different error propagation and masking scenarios Between logic and memory units



Case Study: OR1200 Processor

- OR1200 has ~ 50,000 gates/flip-flops, 4 memory units, one register-file
- Entire processor analysis results



a) Absolute SER for Different Workloads

b) Relative SER for Different Workloads

Overall SER Where Caches and Register-file SRAMs Have Different Interleaving Distances (IDs)

Contribution of Different Types of Components on the Overall SER

Achievements & Future Work

- **Industrial Collaborations with iRoC, IBM, and ARM**
- **Achievement:** Cross-layer soft error rate analysis platform
  - Device: Failure in Time (FIT) rate analysis
  - Layout: Multiple transient error sites extraction
  - Circuit: Single-cycle error propagation
  - Architecture-level: Combined logic and architecture analysis

  - **Fast and accurate full system soft error rate analysis**

- **Future research direction: Cost-efficient soft error mitigation**
  - Soft error-aware high-level synthesis
  - Circuit-level mitigation of soft errors
  - Efficient error correction codes (ECCs) for small memory arrays
  - Protecting configuration bits of FPGAs

Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

Overall Project Objectives

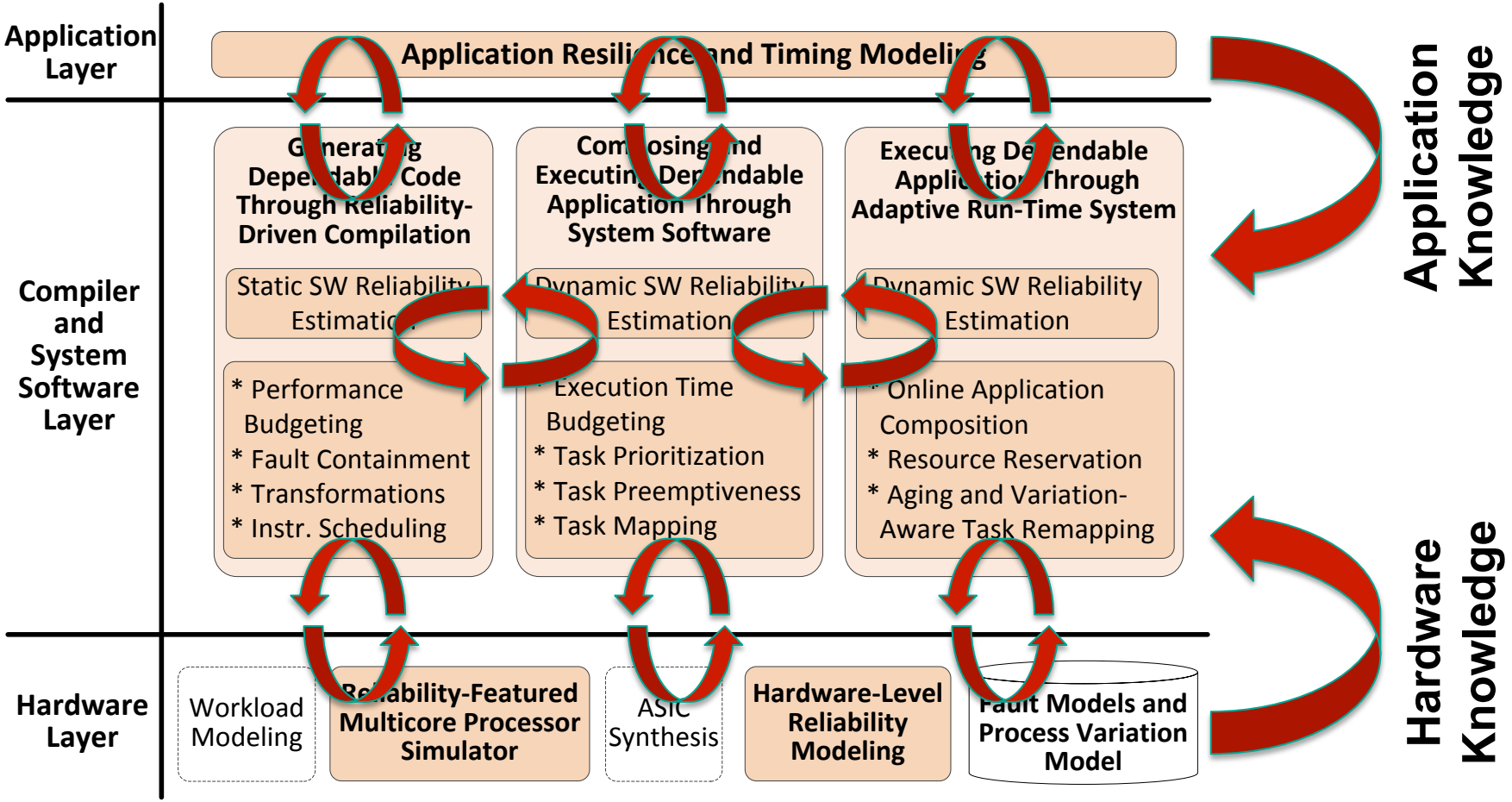
- **Reliability-Driven Code Generation and Execution**
  => Engage Multiple System Layers, i.e., Compiler, Offline System Software, Run-Time System

  - **Objective 1:** Developing Reliability-Driven Compilation and System Software Techniques for Dependable Code Generation and Execution
  - **Objective 2:** Developing an Adaptive Run-Time System for Dependable Code Executions
  - **Objective 3:** Modeling Reliability and Resilience

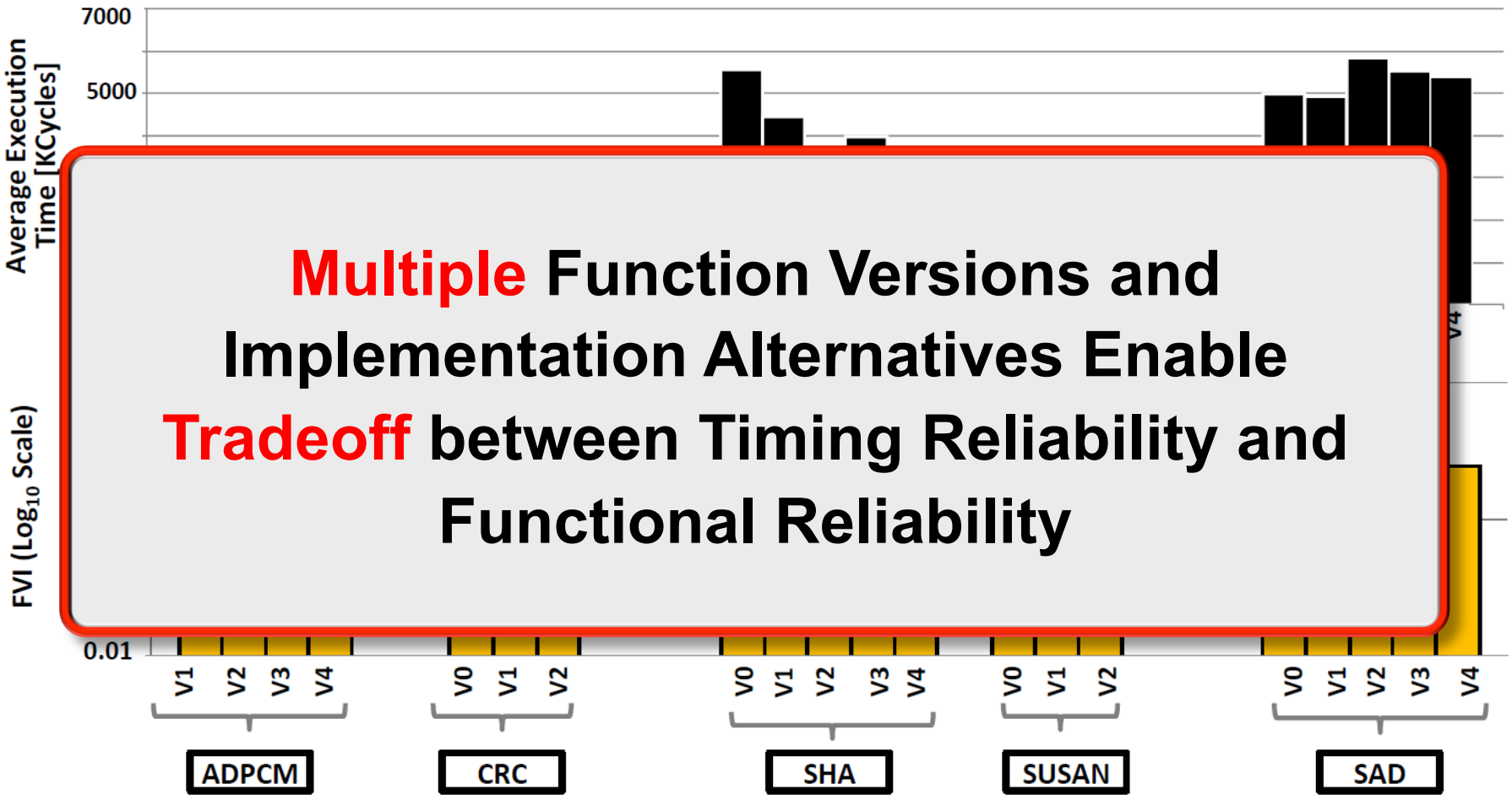
- **Overall Objective:** How to jointly exploit the tight integration of system software, compilation techniques, and adaptive run-time system to enable a highly-dependable software system for embedded systems

Our Novel Contributions and Goals

- Joint reliability optimizations at compilation and system software layers using both offline and online techniques



Performance vs. Reliability Tradeoff



Multiple Function Versions and Implementation Alternatives Enable **Tradeoff** between Timing Reliability and Functional Reliability

Conclusion

- **Bridging the gap between hardware and software** is necessary to enable software-level reliability estimation and optimization schemes

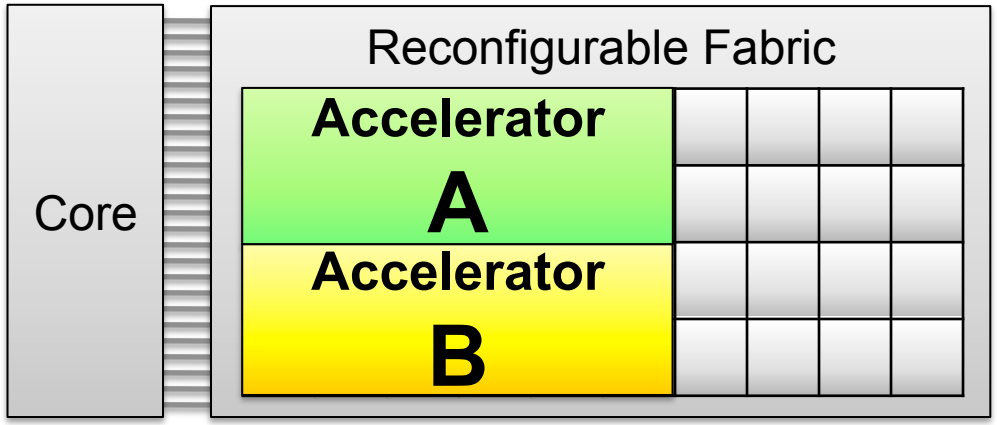
- Software can't erase the problem of unreliable hardware; **BUT: it can contribute and relieve the problem => Reliability increase basically comes for free (probably some performance overhead)**

Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

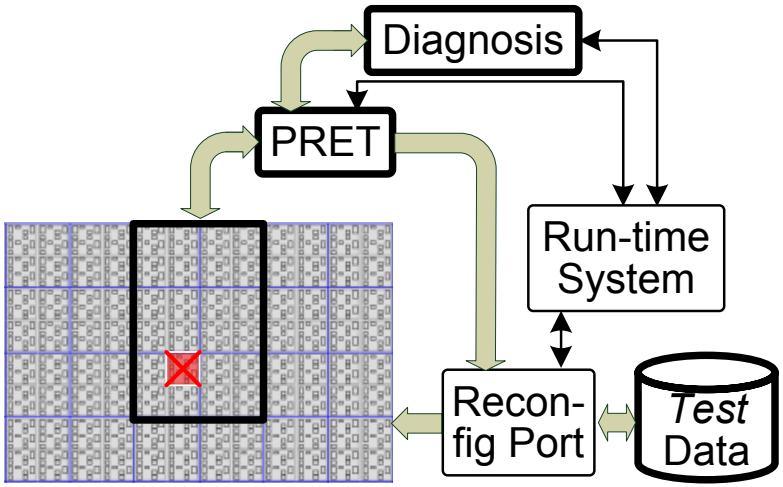
Scope and Focus: Reliability for Reconfigurable Systems

- **Phase-I:** Online Test Strategies for Reliable Reconfigurable Architectures
  - Reliable Reconfiguration: assure that FPGA structure and reconfiguration process work correctly
  - Focusing on permanent faults
- **Phase-II:** Pro-Active Self-Defense by Monitoring, Testing, and Failure Prediction
  - Broadening scope towards transient faults
  - Runtime reconfigurable reliability/performance trade-off
  - Aging mitigation methods for reconfigurable architectures

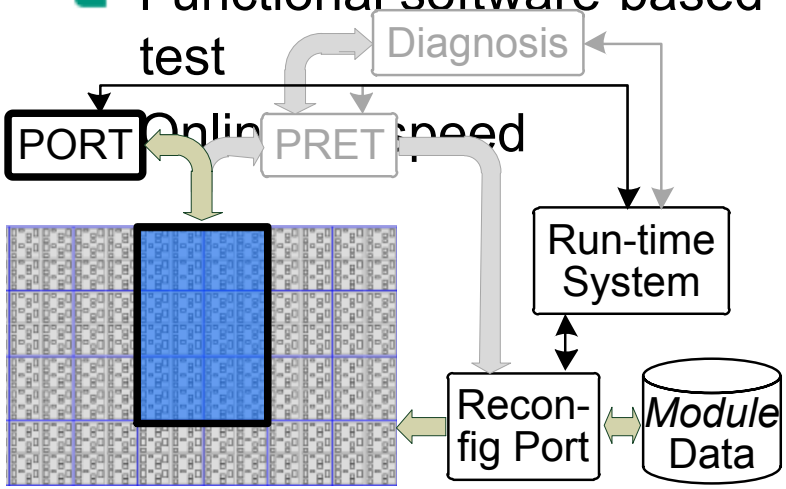


Pre-Configuration Test (PRET) and Post-Configuration Test (PORT)

- PRET: Test structural integrity of reconfiguration fabric
  - Reconfigure several structural tests at runtime before reconfiguration with mission logic



- PORT: Test correct reconfiguration and interconnection
  - Functional software-based test
  - Online speed



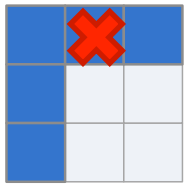
DFG SPP 1500
寺 OTERA
Online Test Strategies for Reliable Reconfigurable Architectures

Module Diversification for Fault tolerance and Stress Balancing

Example with 3x3 CLBs:

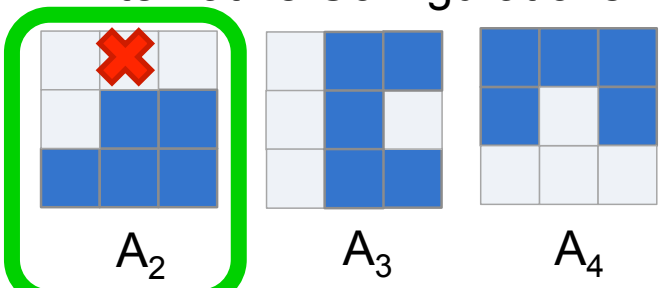
- used
- unused

Original Config.



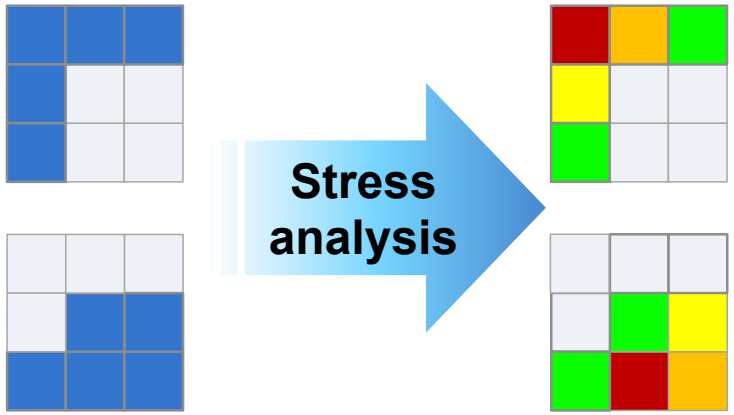
A_1

Alternative Configurations

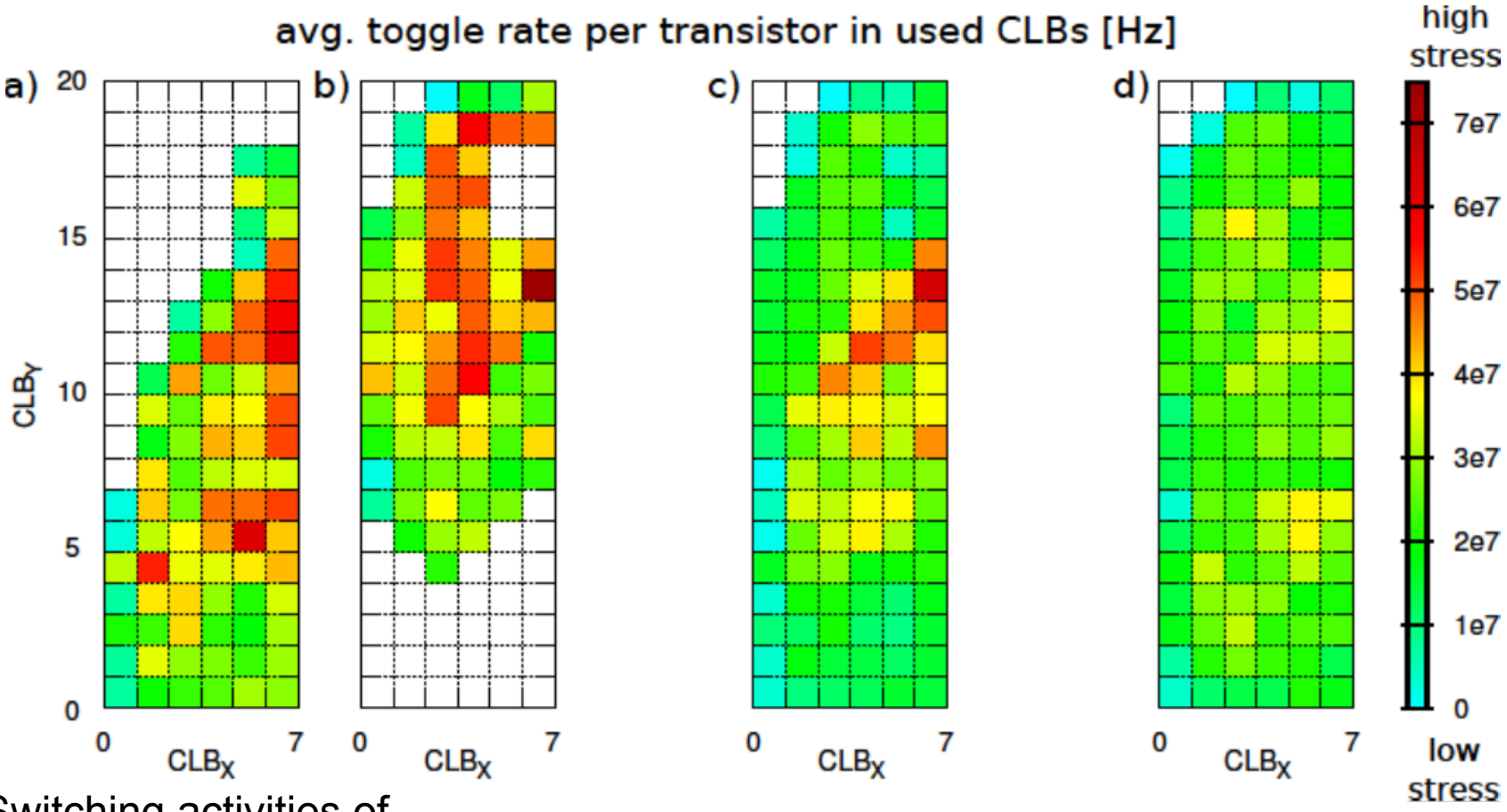


A_2 A_3 A_4

- Goals:
  - Tolerate permanent faults
  - Mitigate aging process
- Principle:
  - Implement functional modules in different ways in terms of CLB usage → *diversified* configurations
  - Find *optimal* scheduling to minimize the stress per CLB



Experimental Results for Module Diversification



Switching activities of
a), b) two diversified configurations, c) an alternating schedule thereof,
d) a balanced scheduled with min. number of 4 configurations

DFG SPP 1500
寺 OTERA
Online Test Strategies for Reliable Reconfigurable Architectures

Conclusion

- Pro-active self-defense to defend against short-, mid- and long-term faults

- System adaption based on modeling, monitoring, diagnosis, and prediction
  - Incorporate efficient online test methods at system level (Phase I)
  - Reactive and pro-active system hardening, resource management (Phase II)

DFG SPP 1500
寺 OTERA
Online Test Strategies for Reliable Reconfigurable Architectures

Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

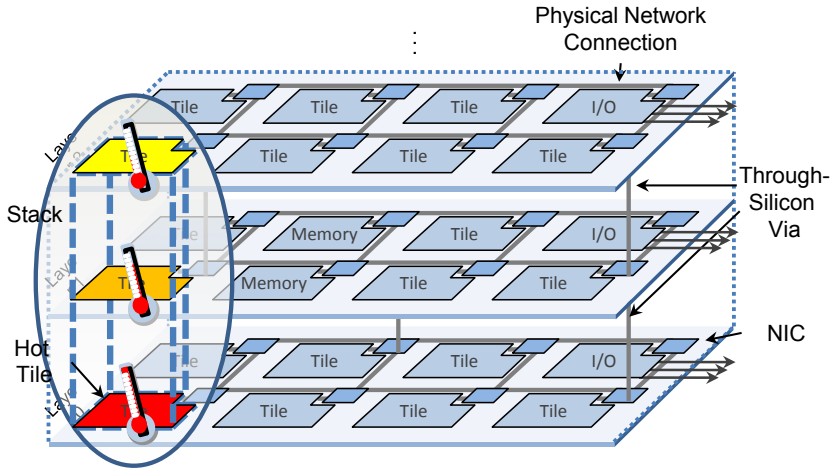
Motivation & Challenges

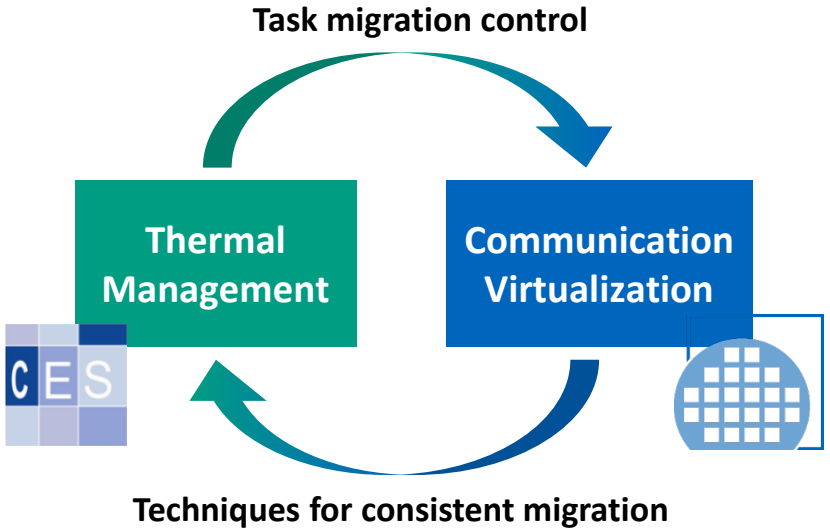
- Thermal problems on current and future chips lead to decreased dependability (short/long term effects)

➔ Problem worsens with 3D stacked manycore architectures

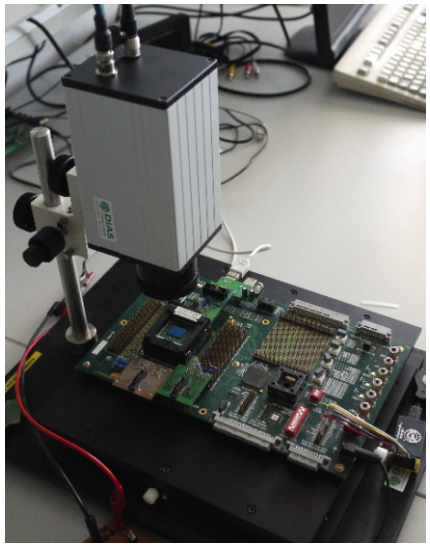
  - Centrally controlled thermal management not scalable

- Providing thermal-aware task allocation
  - complexity handled by distributed decision techniques through dynamic clustering concept
- Agent-based task migration from hot cores to cooler ones and keeping their (I/O) communication reliable, secure and transparent

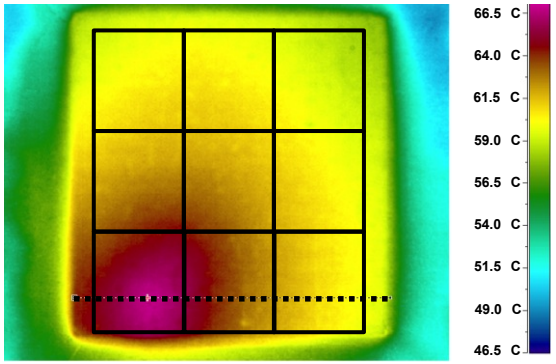




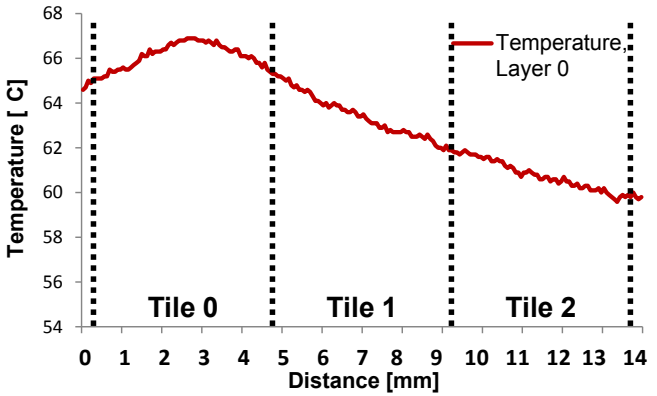
Results ThermalMgmt



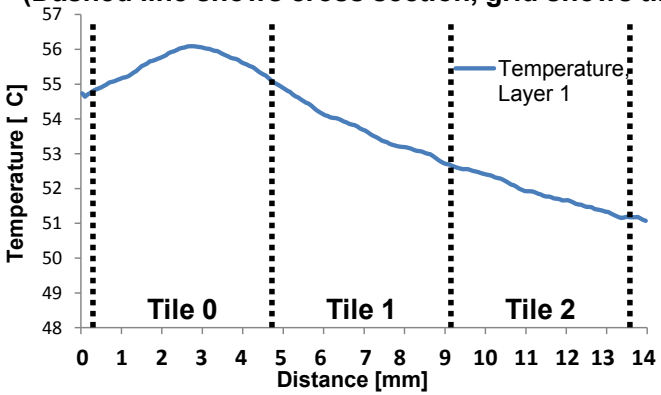
Experimental setup conducted in our lab during 1st project phase



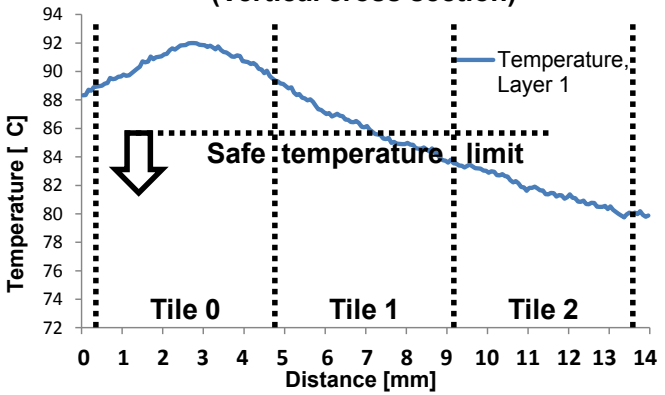
a) Computational layer
(Dashed line shows cross section, grid shows tiles)



b) Thermal distribution computational layer
(Vertical cross section)



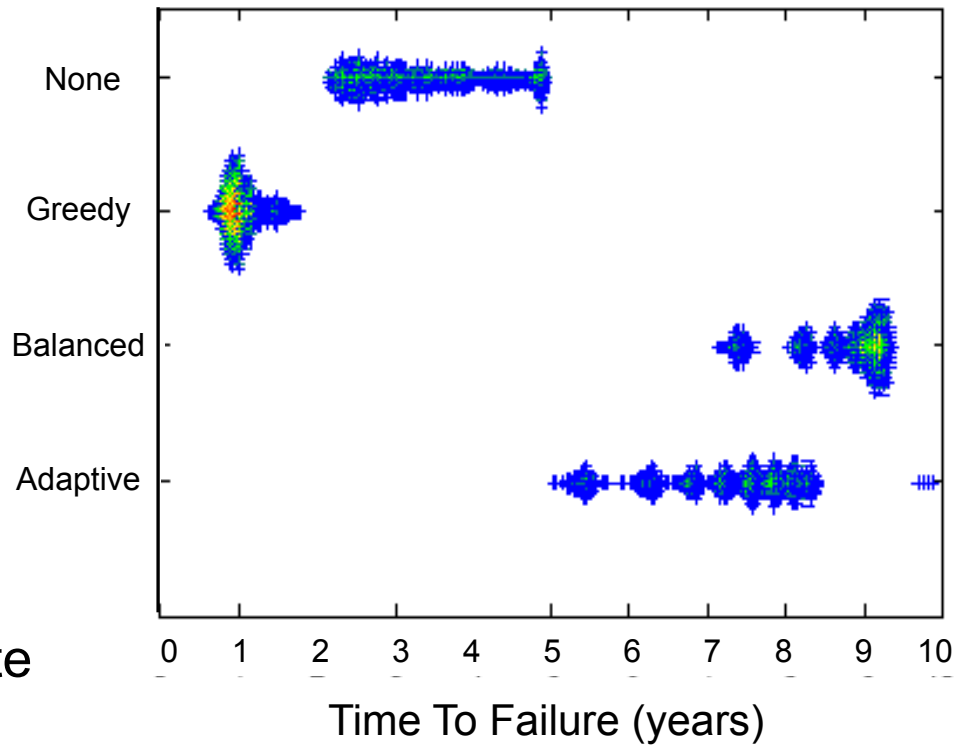
c) Layer 0: computation; Layer 1: idle
Temperature Layer 1



d) Layer 0: computation; Layer 1: computation
Temperature Layer 1

Results ThermalMgmt

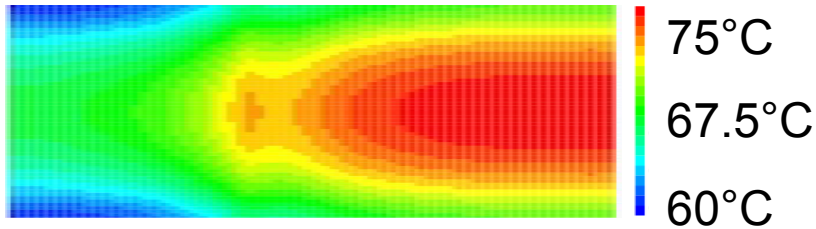
- Different thermal management strategies highly influence the expected time to failure
- Assumption: two types of task execution profiles:
  - high performance (HP)
  - Low performance (LP)
- Policies:
  - Greedy policy: limit HP tasks to small subset of cores
  - Balanced policy: evenly distribute HP tasks over cores
  - Adaptive policy: adaptively change the size of the set of cores running high performance tasks depending on aging budgets
  - None: random task placement (as long as aging budget allows)



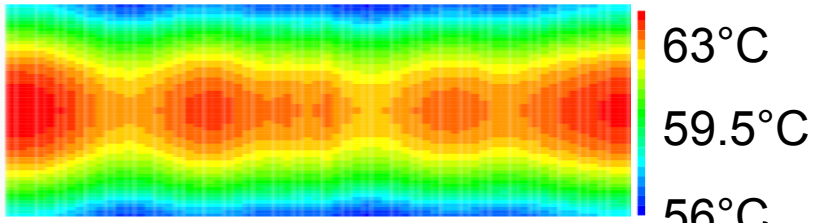
Sytem Management

- High-level thermal simulation assumes one overall power value per component (variations due to conductivity)
- Low-level simulations reveal varied temperatures inside register file

64°C
60°C
56°C
52°C



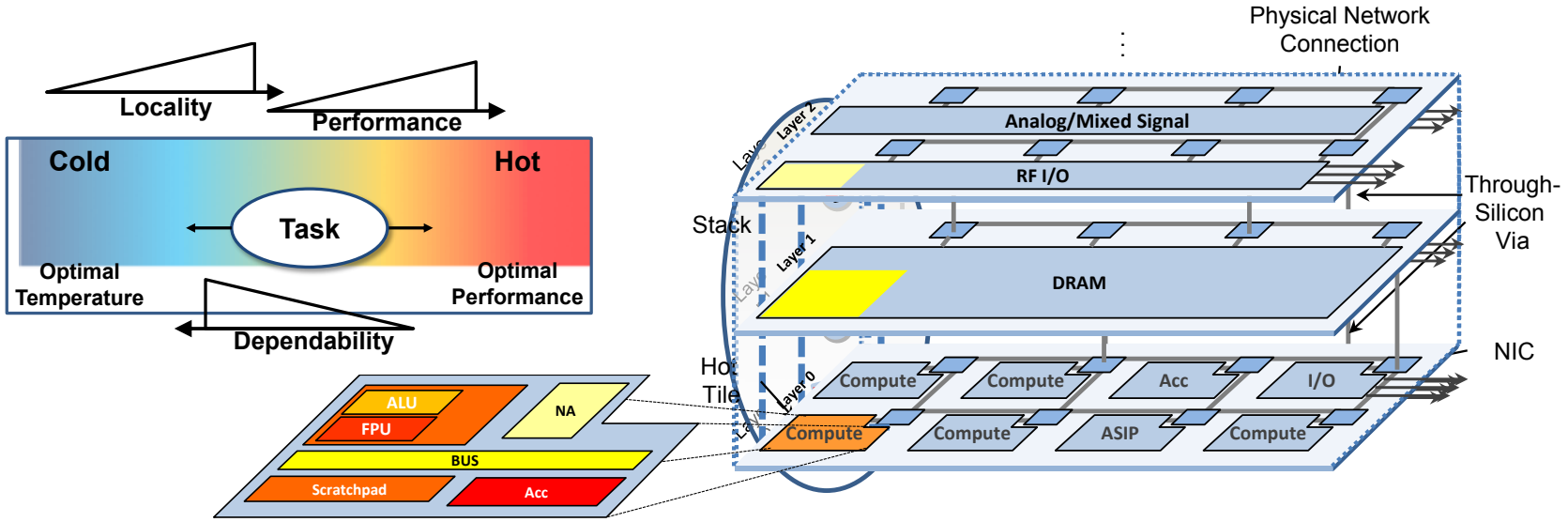
- Controlling accesses inside register file reduces temperature without reducing overall power consumption



- → System management must be performed on multiple levels (fine and coarse-grained)

Ongoing and Future Work

- Generalization of thermal management for (embedded) 3D MPSoC
  - Homogenous ➔ heterogenous architecture
  - Tile-centric ➔ mixed-grain, multicritical modeling, management and communication virtualization approach
  - Thermo-centric ➔ multi-fault error model (incl. aging)
  - Mixed-granularity levels and abstraction of fault-models



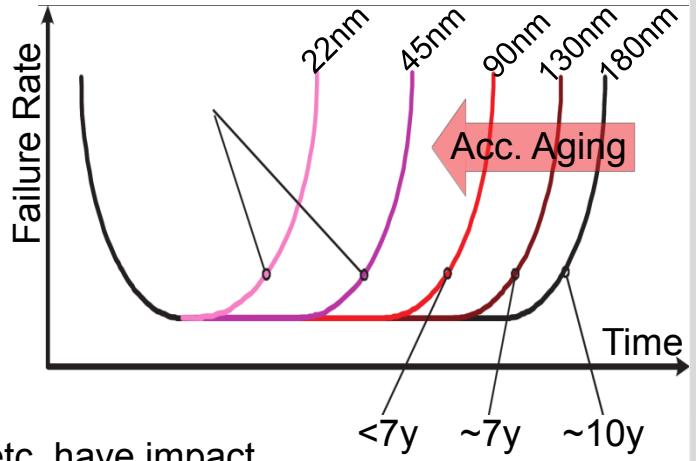
Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

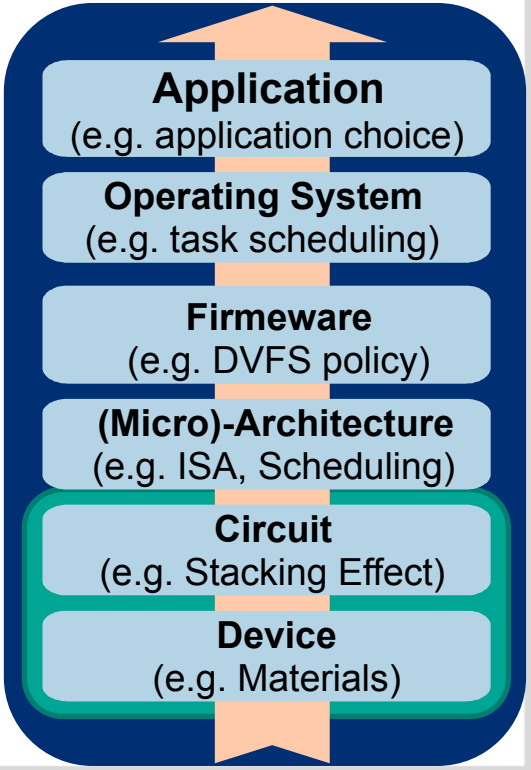
DFG SPP 1500

Ambrosia

Overview

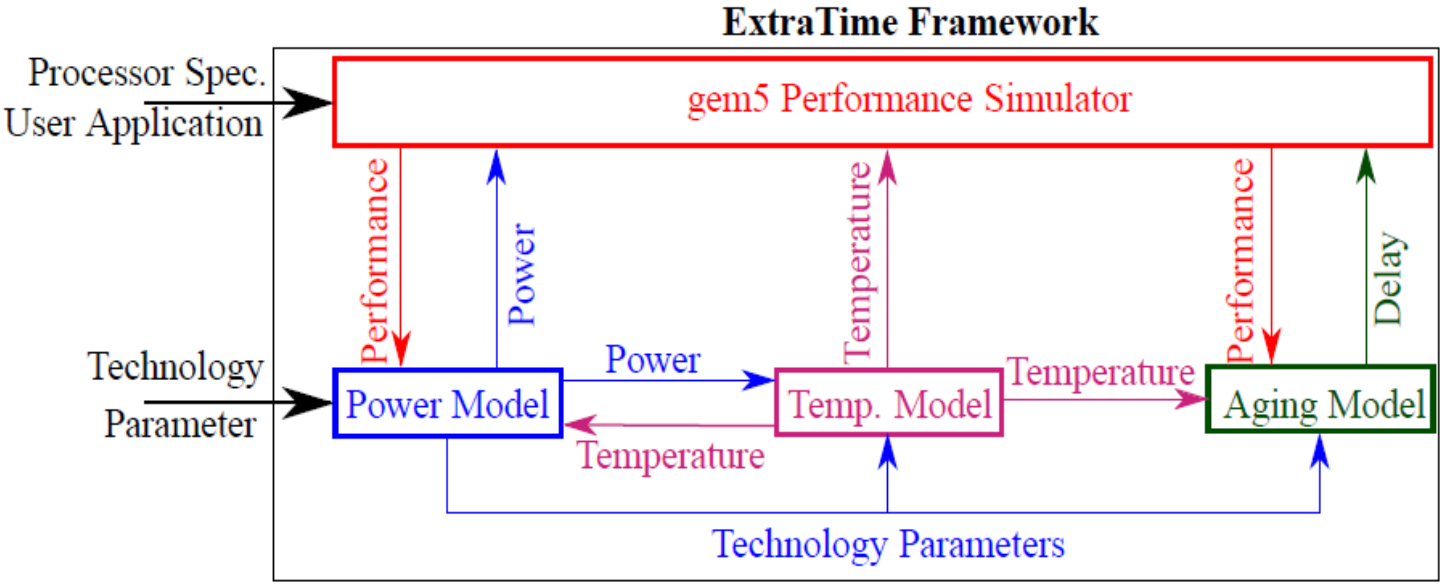


- Device aging at nanoscale technologies
  - BTI, HCI, TDDB, etc.
  - Chip lifetime is significantly reduced
  - Prediction is very challenging
    - Temperature, Workload, Voltage, Runtime, Frequency, etc. have impact
- Higher levels of design stack
  - Considerable impact on aging → opportunities
- Objectives
  - Design-time/runtime modeling & mitigation
    - Circuit-level mitigation techniques
      - e.g. input vector control, path balancing
    - (Micro)-Architecture-level mitigation
      - aging-aware NOP, instruction scheduling, pipeline stage lifetime balancing
    - Modeling at microarchitecture-level
    - Proactive aging-aware adaptation (→ DVFS)
    - Compiler-level techniques for aging reduction



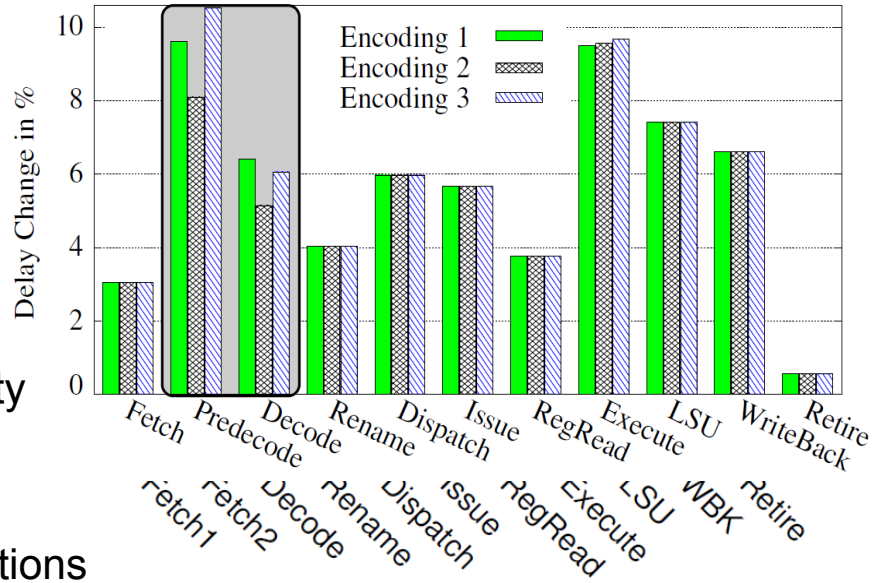
Wearout Modeling

- ExtraTime Framework
  - Microarchitectural simulation framework containing:
    - Power and temperature models
    - Contains (micro)-architectural aging models for BTI and HCI
      - Input: usage (performance statistics), temperature, tech. parameters
      - Output: delay degradation, mean time to failure (MTTF)



Static Mitigation Techniques

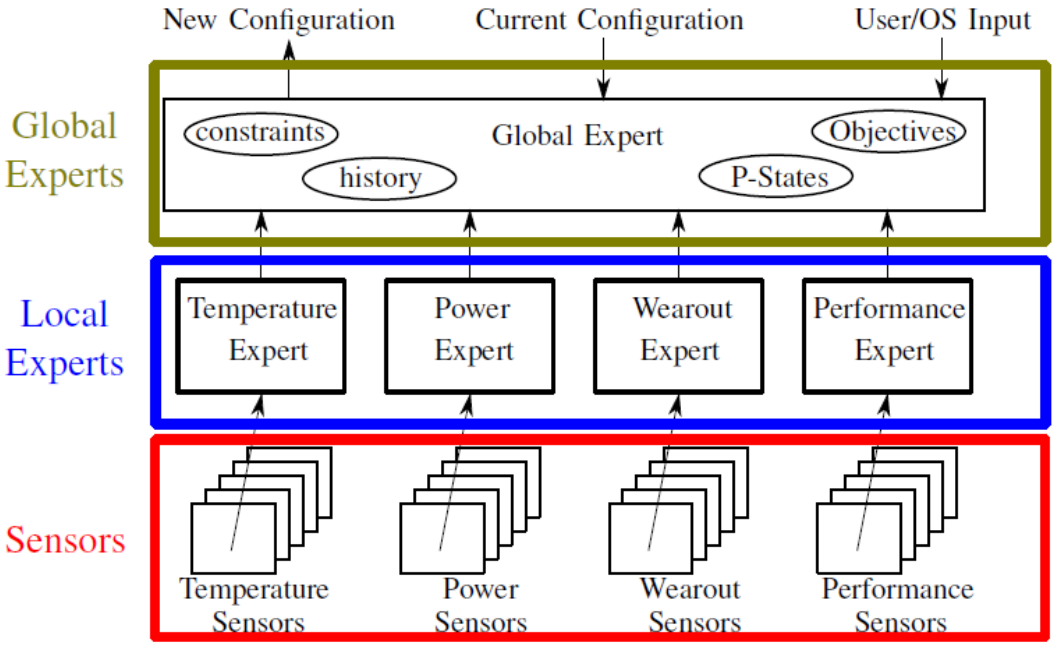
- NOP-Instruction
  - NOP-Instruction does not affect program execution
  - Idea: Replace default NOP with aging-aware NOP
- Instruction Scheduling
  - Instructions have different timing criticality Critical instructions will fail first
  - Only 1/5 instructions area critical
  - Idea: Use special units for critical instructions & adapt scheduling accordingly



- MTTF-Balanced Pipeline Design / Path-Balancing
  - Non-uniform aging rates among different pipeline stages → Huge optimization potential
  - Idea: Balance pipeline stage delays at end of lifetime and not at the beginning
- Instruction Set Encoding
  - Instruction opcodes have significant aging impact
  - Idea: Design the instruction set encoding in an aging-aware way

Dynamic Mitigation Techniques

- Fine-grain, pro-active dynamic voltage & frequency scaling
  - Based on available sensor/monitoring infrastructure + novel expert system
  - Using machine learning techniques
- Platforms
  - Real experiments using on-chip sensors and monitors of IBM POWER 7+
  - Simulated using ExtraTime framework



Achievements & Future Work

- Achievements
  - Modeling & Evaluation platform at (micro)-architecture-level
  - Various static & dynamic cross-layer aging mitigation techniques

- Industry Collaboration and Future Work
  - Collaboration with IBM
    - System health prediction and adaptation using on-chip monitors
      - POWER 7+ system
  - Collaboration with IMEC
    - Aging-aware statistical timing analysis
  - Exploiting application-level knobs for better aging mitigation

Projects

- PERCIES: Providing Efficient Reliability in Critical Embedded Systems
- Get-SURE: Generating and Executing Dependable Application Software on UnReliable Embedded Systems
- OTERA: Online Test Strategies for Reliable Reconfigurable Architectures
- VirTherm-3D: Communication Virtualization Enabling System Management for Dependable 3D MPSoCs
- Ambrosia: Cross-layer Modeling and Mitigation of Aging Effects in Embedded Systems
- CRAU: Compositional System Level Reliability Analysis in the Presence of Uncertainties

Goals

- Develop a framework providing formal models and techniques for *compositional* and *uncertainty-aware* reliability analysis (CRA)
- Combine and extend well-known reliability analysis techniques across different levels using *adapters*
- Use CRA during design space exploration to consider reliability/cost trade-offs cross-level

DFG SPP 1500

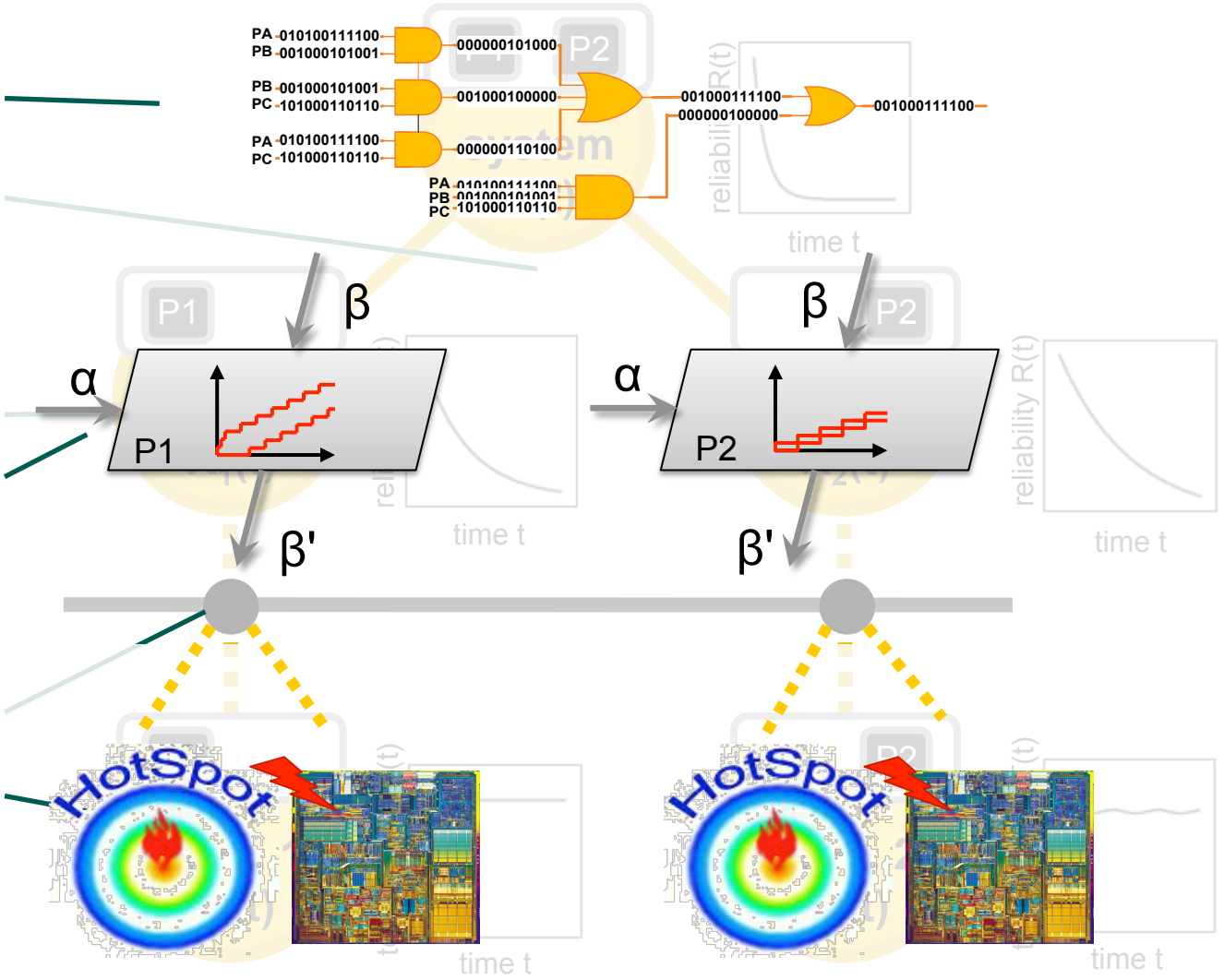
CRAU

CRA Concept & Example

Success Tree-based analysis

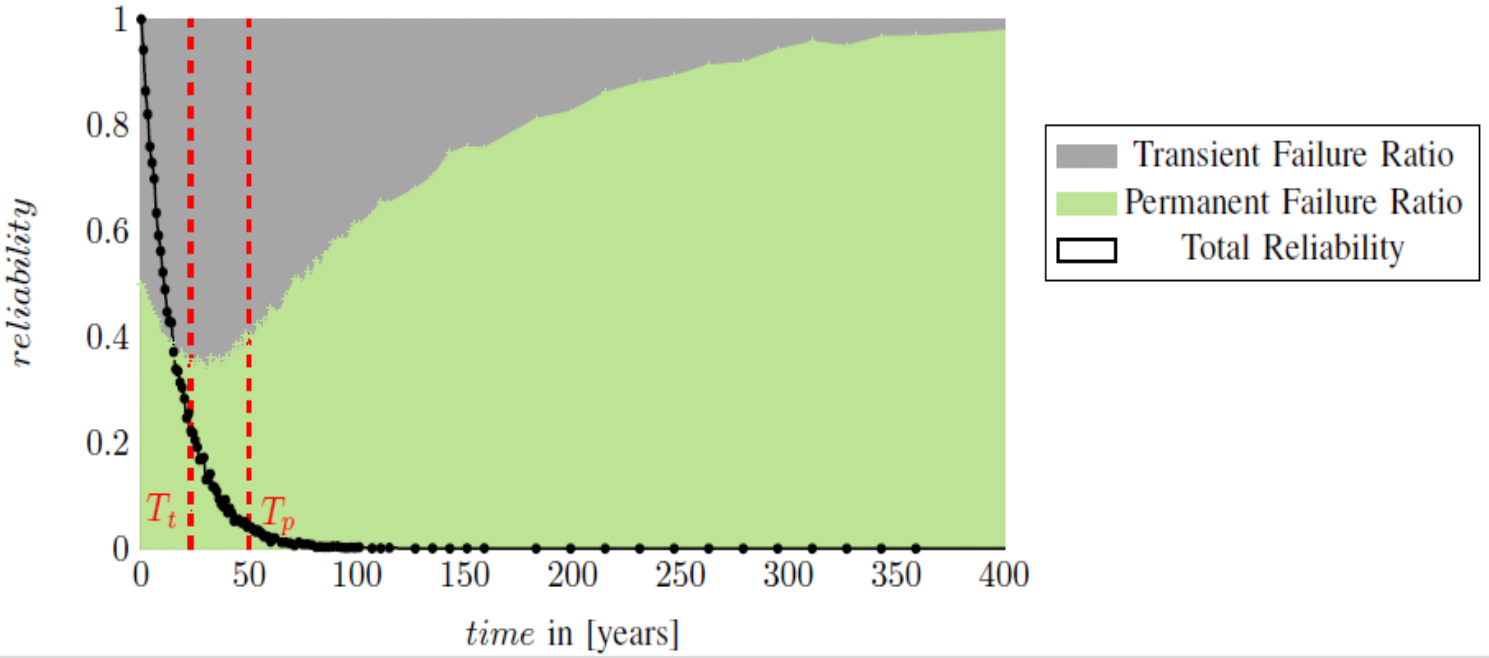
workload analysis using Real-time Calculus

temperature simulation and transient fault consideration



Perm. & Tran. Faults Consideration

- The ratio of permanent and transient faults that induce system failure of a real-world system
- Enables to carefully select the means to increase reliability, e.g., wrt. to the planned lifetime



Future Work

- Focus on advanced uncertainty modeling
- Feedback analysis results to the optimization in greater detail
  - (not just classic reliability measures like MTTF etc.)