Cross-Layer system-level reliability Estimation

Alberto Bosio, Associate Professor – UM
bosio@lirmm.fr

Clereco
Cross-Layer Early Reliability Evaluation for the Computing cOntinuum

SEVENTH FRAMEWORK PROGRAMME

http://www.clereco.eu

Agenda

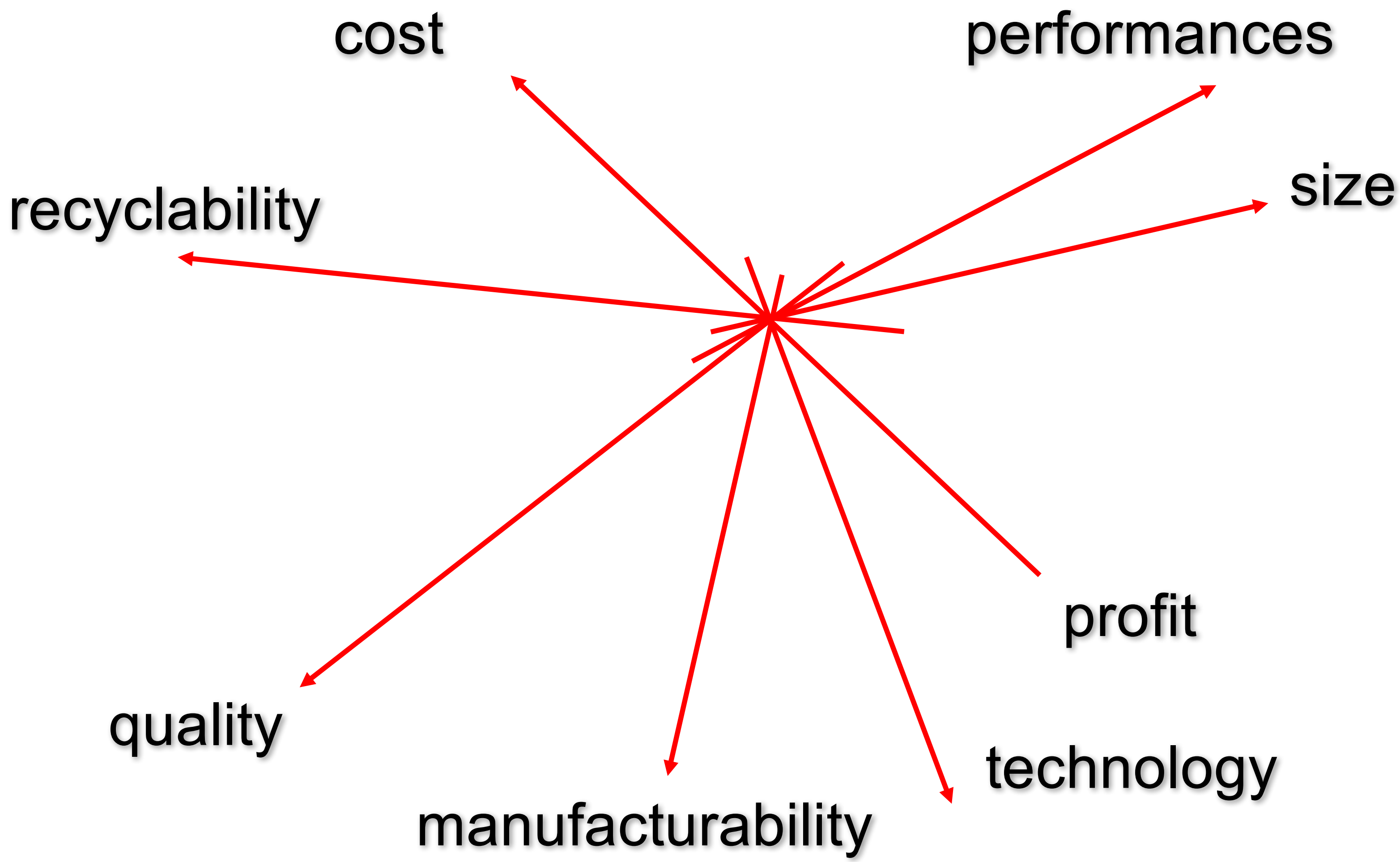
- Context
- Problem
- Proposed Approach
- Validation
- Conclusion

Agenda

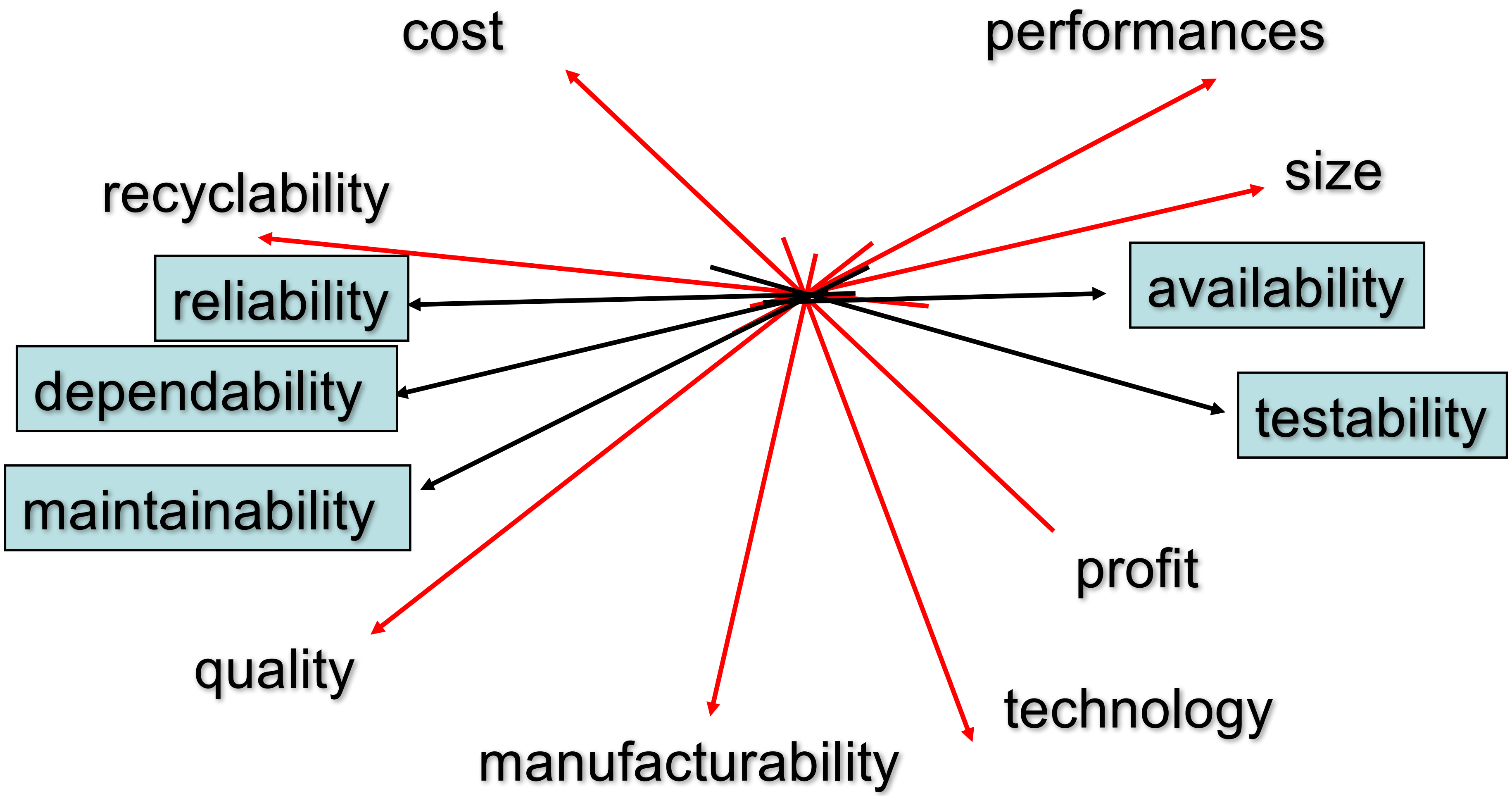
- Context
- Problem
- Proposed Approach
- Validation
- Conclusion

Today's Life with Computing Systems

Design Space



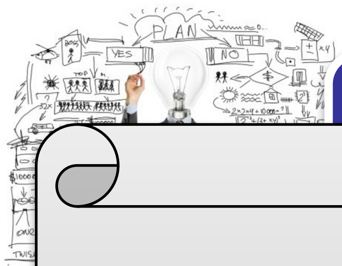
Design Space



Agenda

- Context
- Problem
- Proposed Approach
- Validation
- Conclusion

Computing-System Life Cycle



User's

Re-Cyclin

Design

"If anything can go wrong ...
... it will !"

[Murphy]

In-field Operation

Production

…

Computing-System Life Cycle

User's Requirements

Design Errors

Re-Cycling

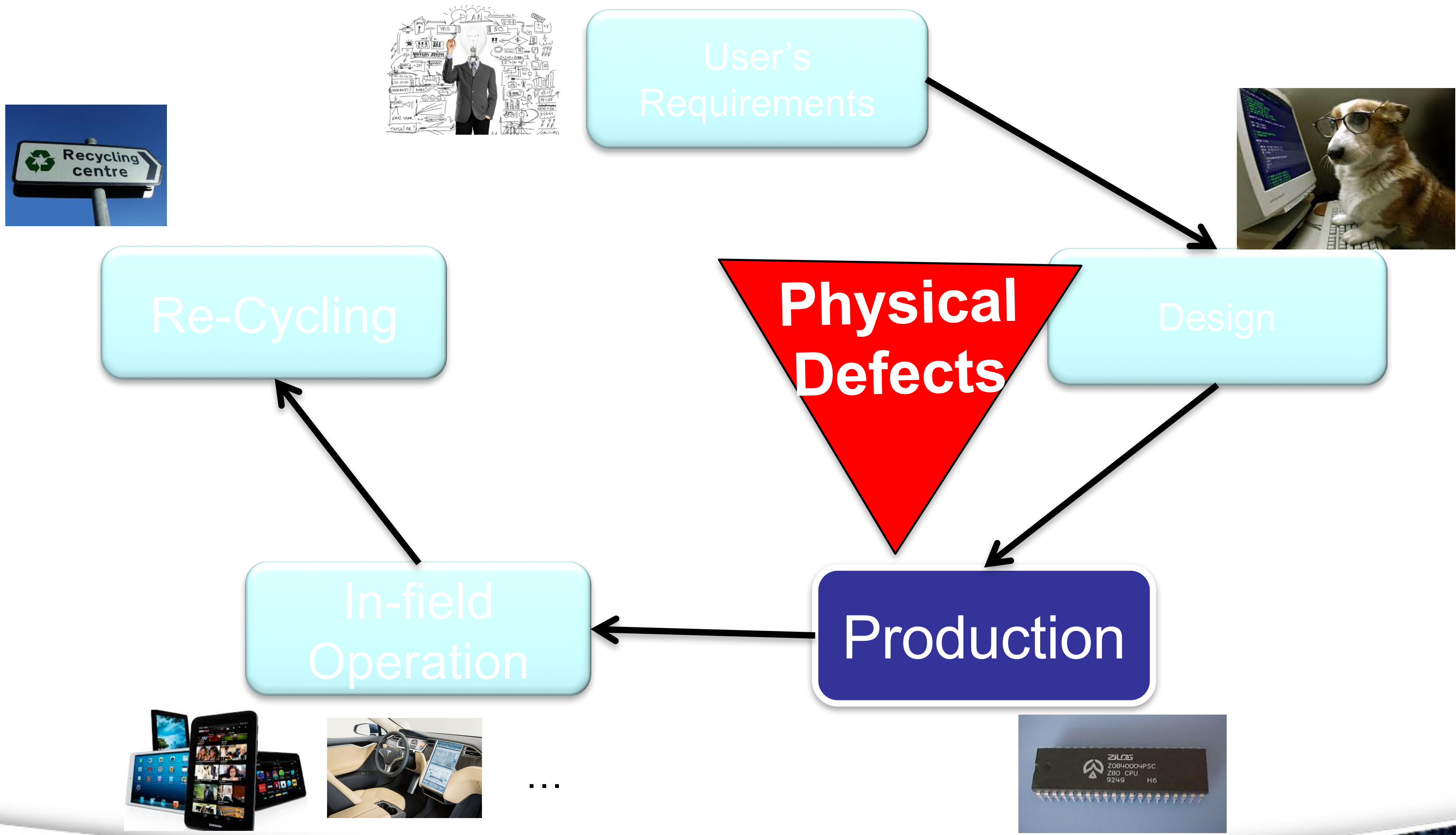
Design

In-field Operation

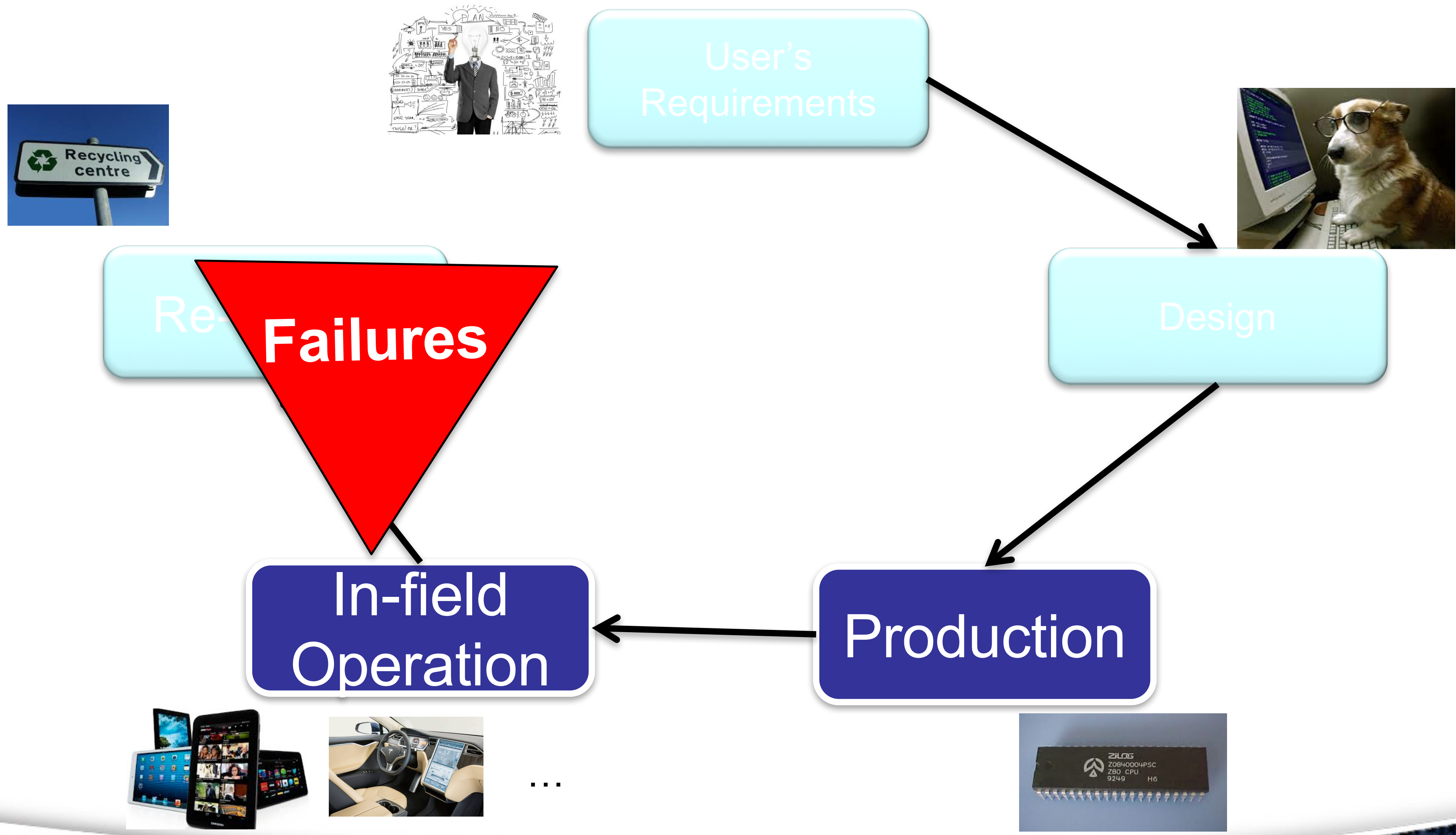
Production

...

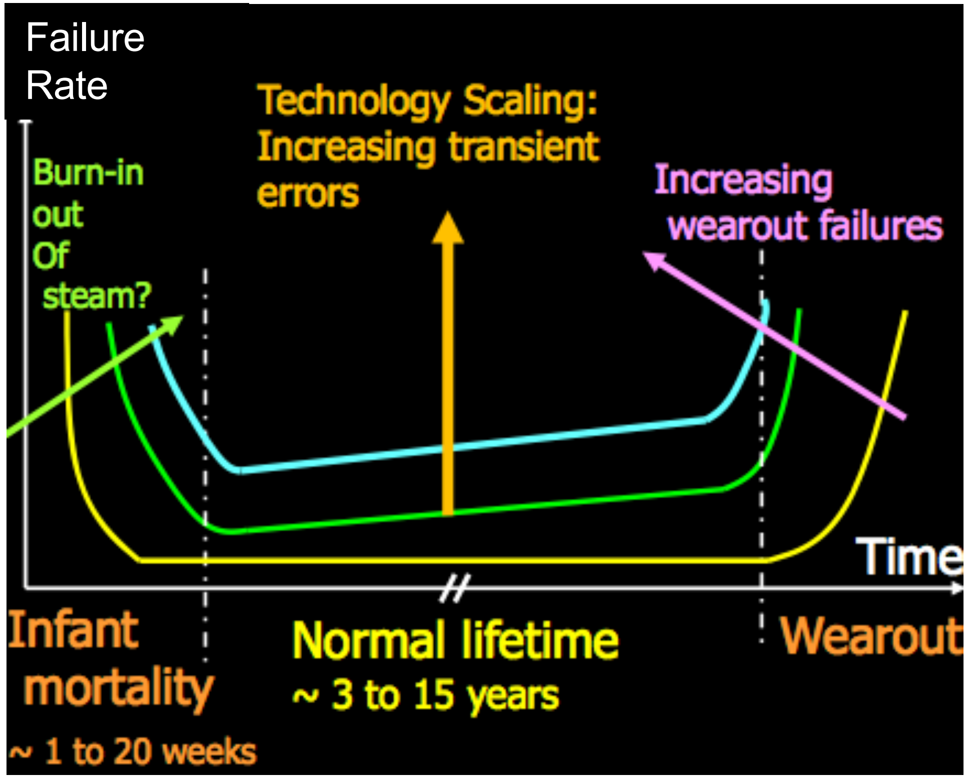
Computing-System Life Cycle



Computing-System Life Cycle



What is the source of problems?

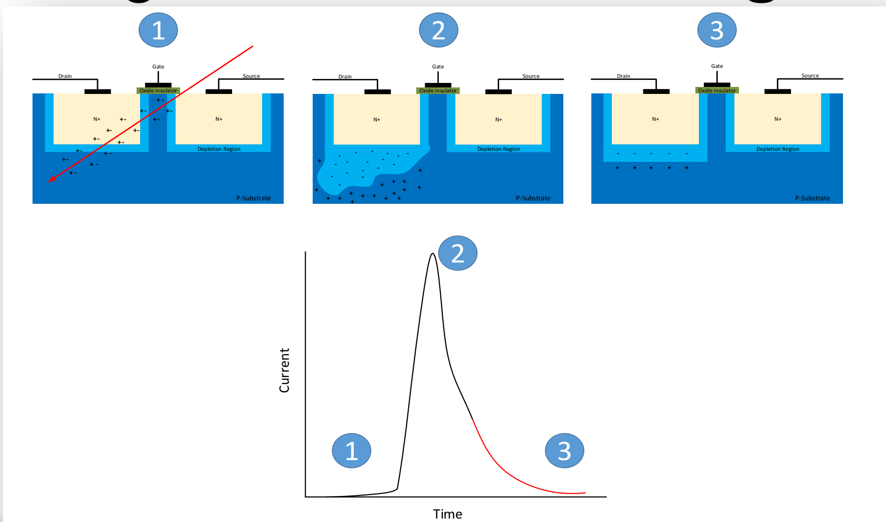


Hamdioui DTIS’15

- Faulty behaviors induced by defects are more complicated
  - Intermittent, transient
- Reduced life time
- Components are becoming unreliable
  - Problems can appear even during operational life….

What is the source of problems?

- Harsh Environment:
  - Neutron radiations from cosmic rays, alpha particles from packaging materials and environmental/design variations are common causes of **perturbations**
  - If the particle strike happens in the hold state of a memory cell or in a flip-flop, the content of the storage element is flipped, causing a **soft-error** or **Single-Event Upset** (SEU)



Example

Trinity (Los Alamos National Lab): 19,000 Xeon Phi

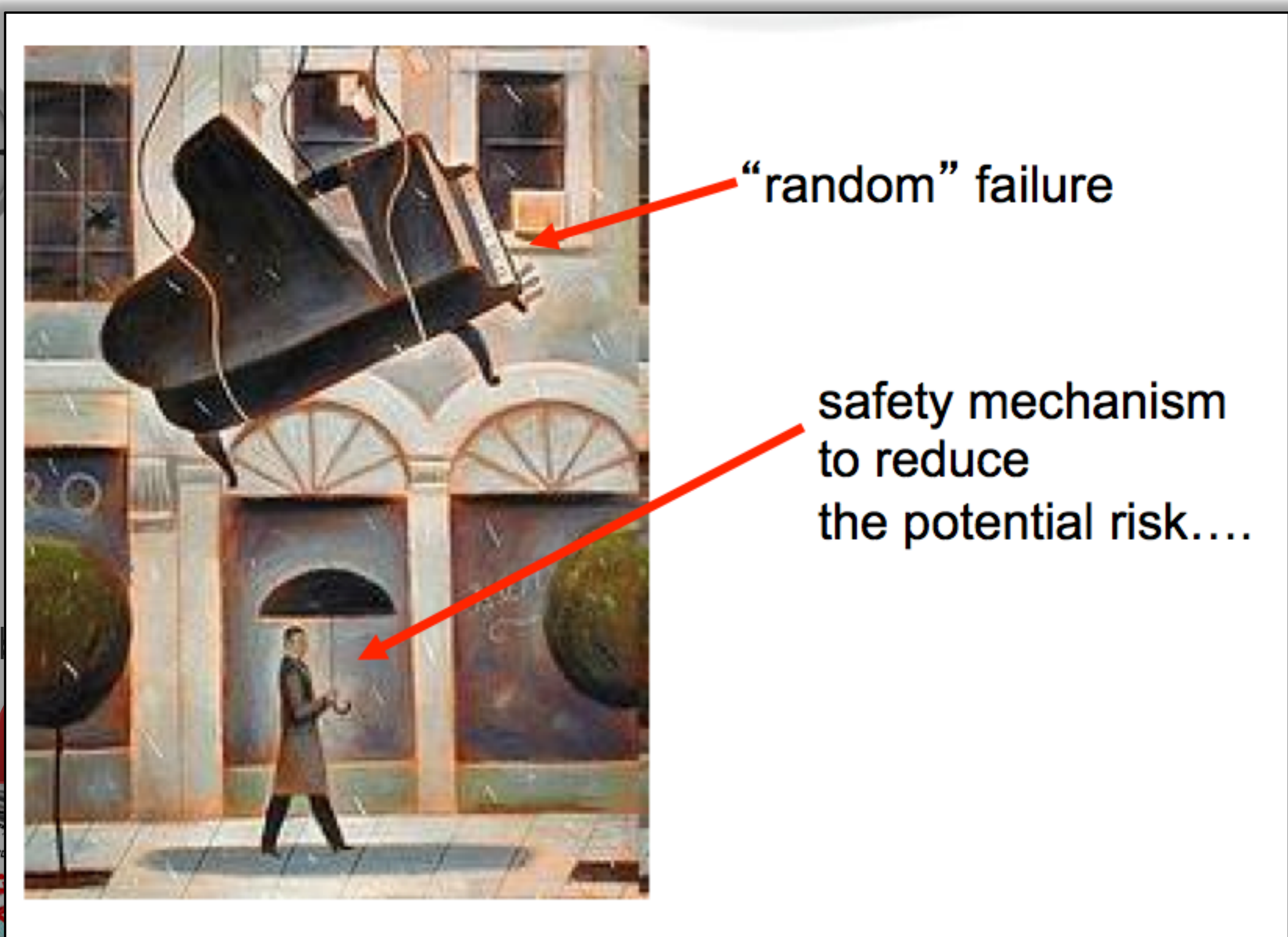
High probability of having a node corrupted
Trinity Mean Time Between Failure is ~12h*

*(data from SC'17)

P. Rech's Courtesy

Los Alamos NATIONAL LABORATORY EST. 1943
UFRGS

The problem



Relia

oach:

N

ndancy

How to Quantify the Reliability

- Reliability metrics[1,2]:
  - Failure rate (λ)
  - Mean Time To Failure (MTTF)
  - Mean Time Between Failure (MTBF)
  - Mean Work to Failure (MWTF)
  - Mean Instructions to Failure (MITF)
  - Architectural Vulnerability Factor (AVF): as the probability that a fault in that particular structure will result in an error.
  - Failure In Time (FIT): defined as a failure rate of 1 per billion hours. A component having a failure rate of 1 FIT is equivalent to having an MTBF of 1 billion hours.

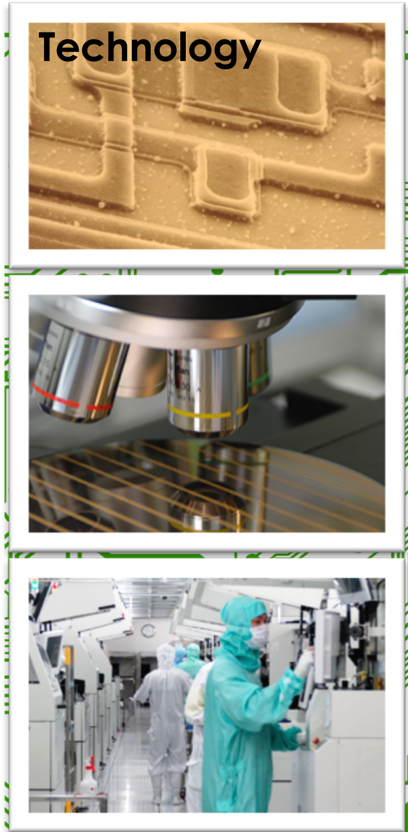
[1] IEEE Transactions on Dependable and Secure Computing, Vol. 1, N. 1, 2004
[2] IEEE Micro, 2003

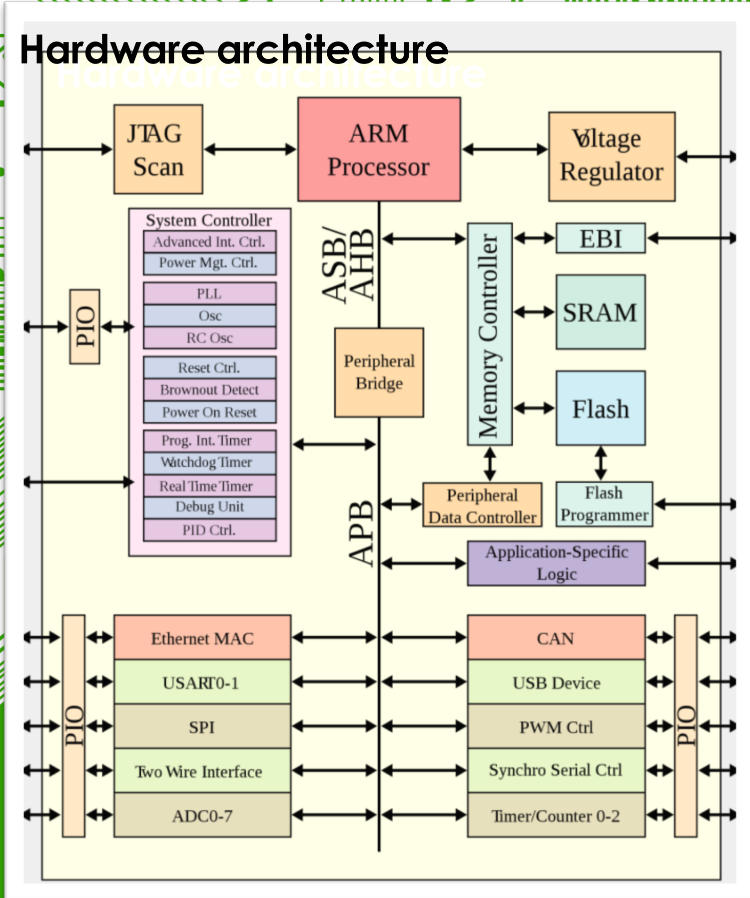
How to Quantify the Reliability

- Reliability metrics[1,2]:
  - Failure rate (λ)
  - Mean Time To Failure (MTTF)
  - Mean Time Between Failure (MTBF)
  - Mean Work to Failure (MWTF)
  - Mean Instructions to Failure (MITF)
  - **Architectural Vulnerability Factor (AVF):** as the probability that a fault in that particular structure will result in an error.
  - **Failure In Time (FIT):** defined as a failure rate of 1 per billion hours. A component having a failure rate of 1 FIT is equivalent to having an MTBF of 1 billion hours.

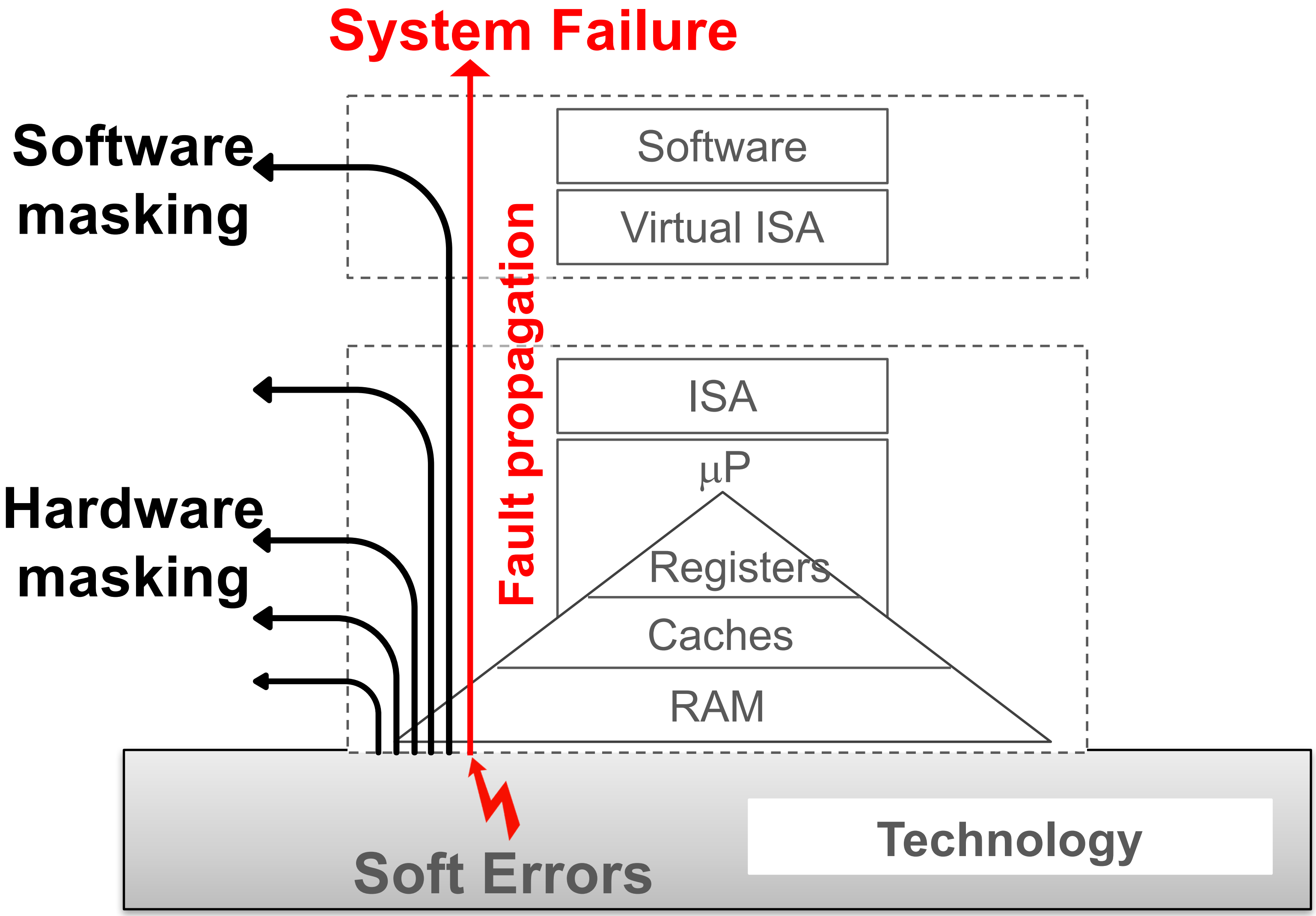
[1] IEEE Transactions on Dependable and Secure Computing, Vol. 1, N. 1, 2004
[2] IEEE Micro, 2003

System-Level View

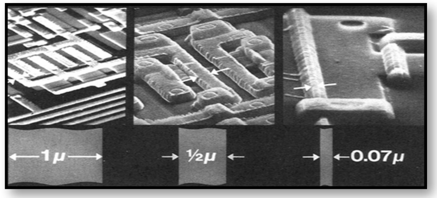




Cross-Layer Reliability

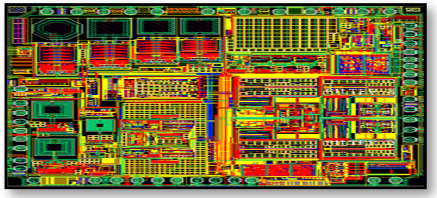


Cross-Layer Reliability



TECHNOLOGY: DEVICE/CELL LEVEL FAULTS

- Radiation effects(soft-errors)
- Ageing (NBTI, HCI, electro-migration)
- Test escapes

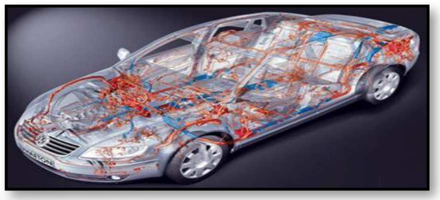


ARCHITECTURE: ISA LEVEL FAULT MODELS

- Wrong data or instruction
- Control Flow Error
- Execution timing Error

SOFTWARE: COMPLEX FAILURE MECHANISMS

- SDC (Silent Data Corruption)
- DUE (Detected, Uncorrected)
- Interrupts, resets, safety fail-over



SYSTEM: USER VISIBLE FAULTS

- Server reboot
- Brake failure
- Mission failure

State-of-the-Art

	Architectural Correct Execution (ACE) analysis & Probabilistic models [1,2]	RTL injection [3]
Simulation Time	Low	High
Estimation Accuracy	Low/Medium	High

[1] N.George, *et. al.* "Transient fault models and AVF estimation revisited", DSN 2010
[2]N.J.Wang, *et. al.* "Examining ACE analysis reliability estimates using fault injection", ISCA 2007
[3]S. Mitra, *et. al.* "CLEAR: Cross-Layer Exploration for Architecting Resilience", DAC2016

Statistical Fault Injection (SFI)

- **Scenario:**
  - program of **1B (10^9) dynamic instructions** (SPEC benchmark)
  - hardware structure of **10K bits** (a physical reg.file)
  - simulation throughput (microarchitecture) of **300K instructions/sec**
  - using **10 servers**

#Injections*	Fault Injection Campaign Time
384	1.5 day
1843	1 week
16,587	9 weeks
23,873	3 months
95,493	1 year

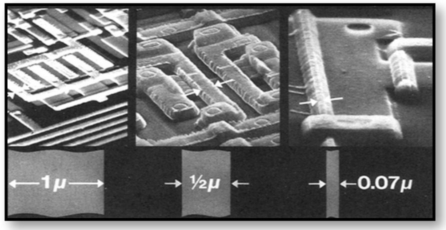
*(Leveugle, *et. al.*, DATE, 2009)

Agenda

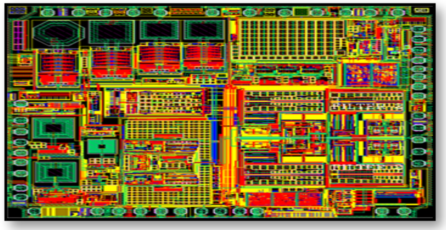
- Context
- Problem
- Proposed Approach
- Validation
- Conclusion

Proposed Approach

- *Divide et Impera* approach:
  - Target each component alone



AVF_T



AVF_{Arch}

AVF_{SW}

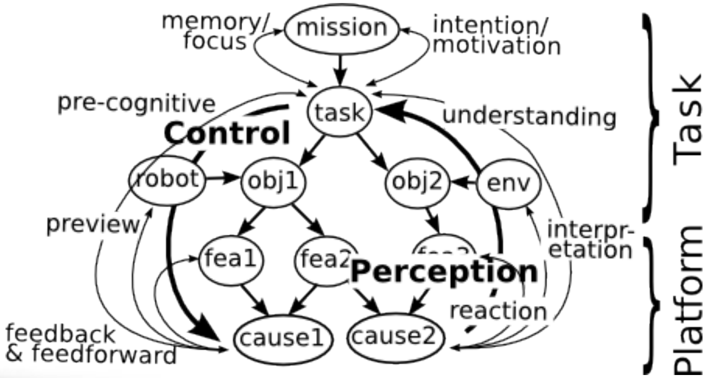
Proposed Approach

- How to combine the different results in order to estimate the reliability at system level?
- We exploit a kind or **reasoning** approach
  - **Bayesian Nets**: A statistical model representing multivariate statistical distributions. They model relations among random variables

Bayesian Nets

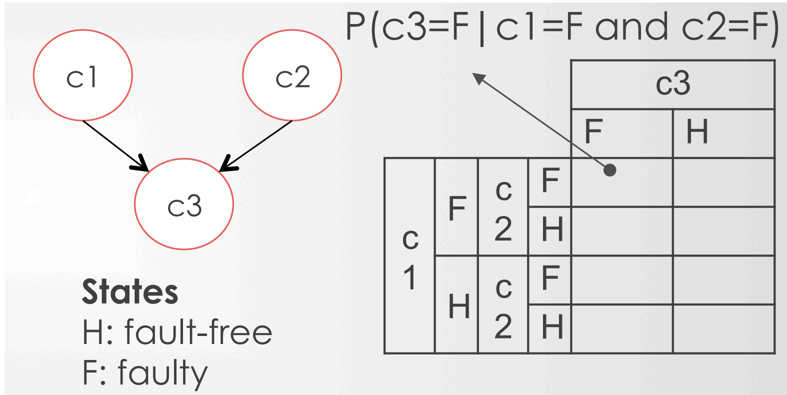
Qualitative Model

- Models the architecture of the system:
  - **Nodes** correspond to components,
  - **Arcs** define temporal or physical relations among components

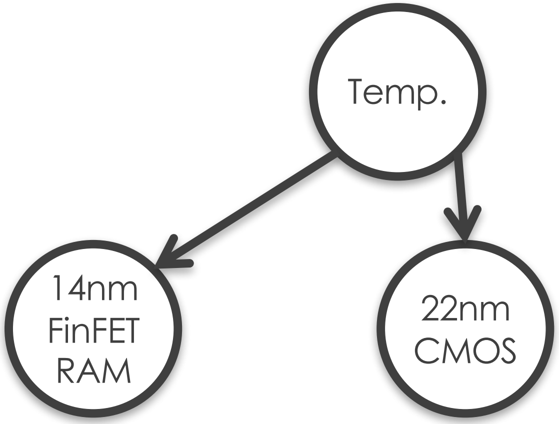


Quantitative Model

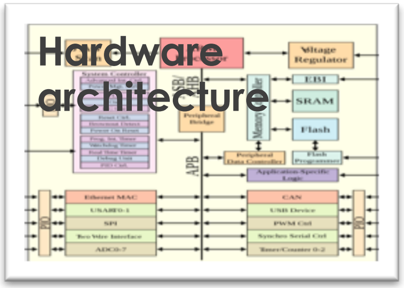
- Models state probabilities as a set of Conditional Probability Tables (CPT).

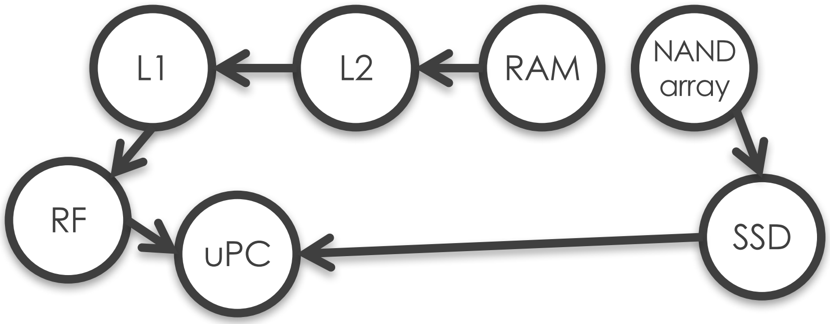


System Modeling: Topology

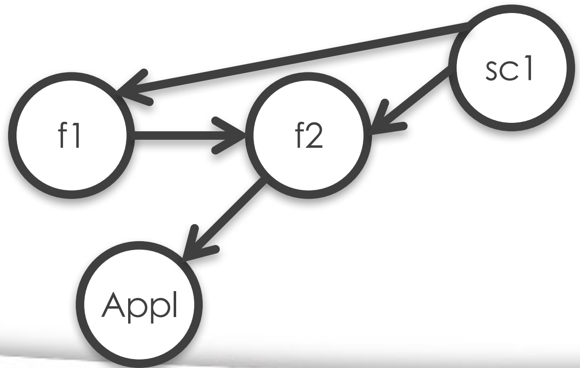


Technology nodes model raw error rates, environmental conditions, etc.



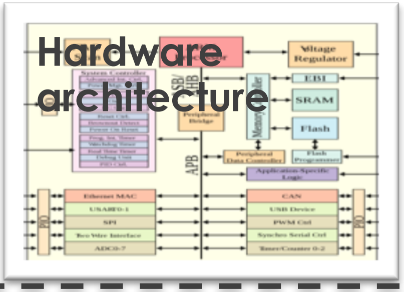


HW blocks are nodes of the network. Complex blocks can be split into sub blocks (e.g., uPC). Arcs are candidate error propagation paths.



SW blocks (e.g., functions or portions of a function) are nodes of the network. Arcs are candidate error propagation paths. Also concepts such as concurrency can be easily expressed.

Example



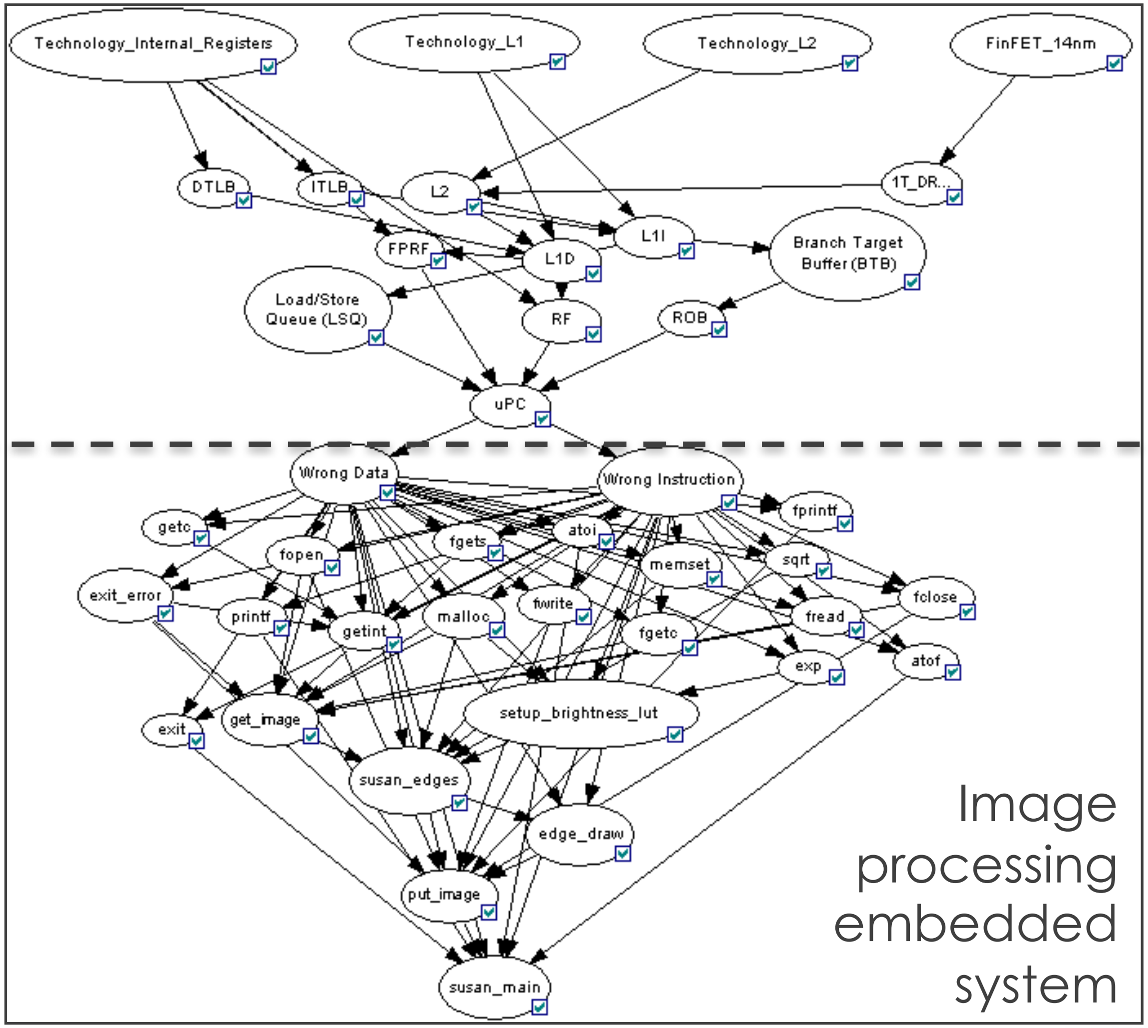
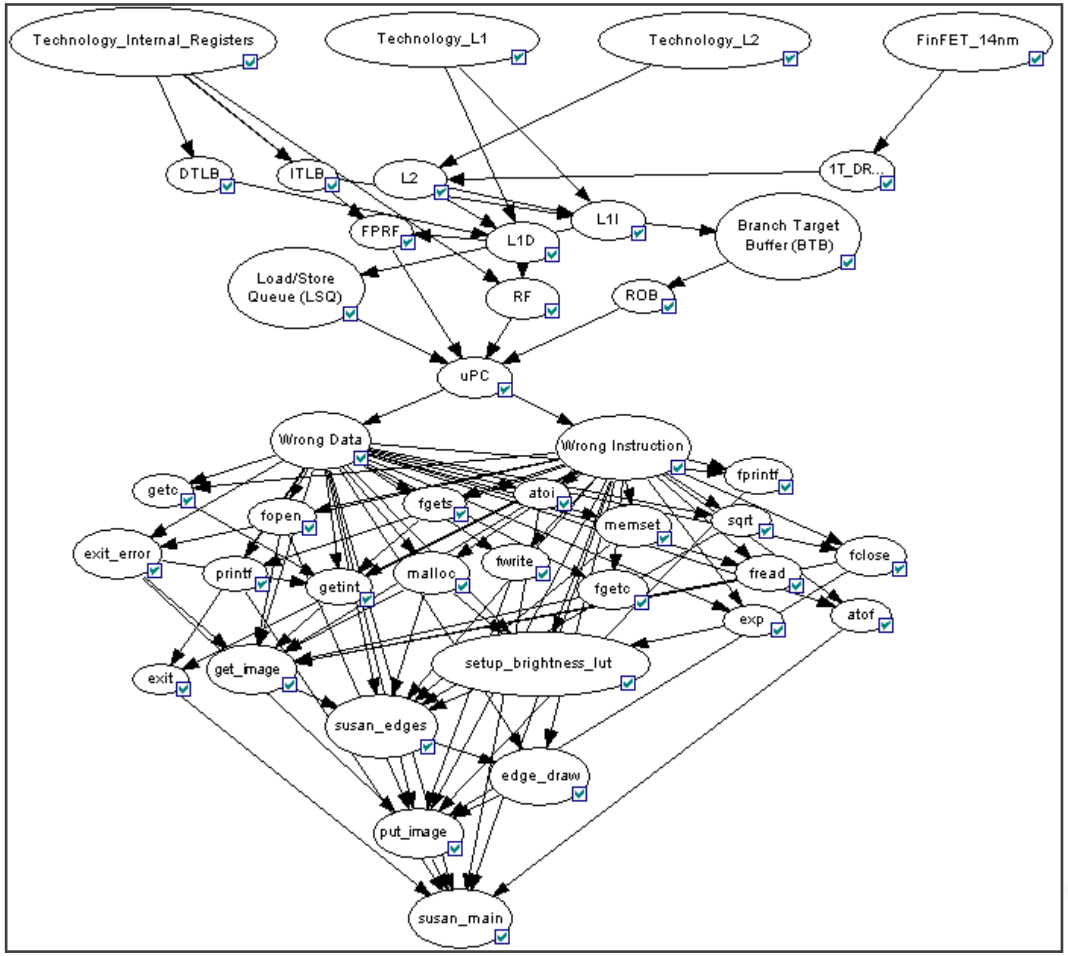


Image processing embedded system

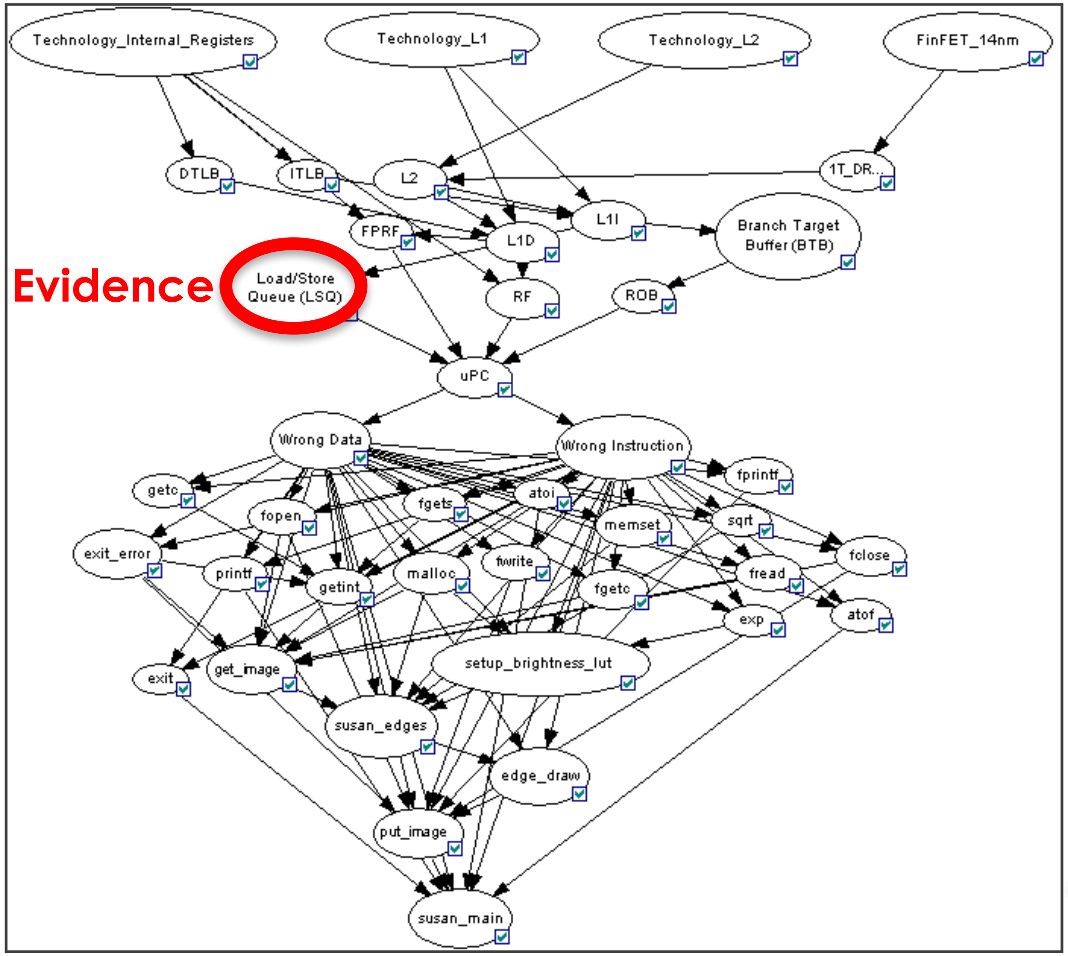
How does it work?



1 Global rel. analysis

System level reliability inference (e.g., MTBF, MTTF, FIT, etc.) taking into account raw errors and propagation/masking of raw-errors

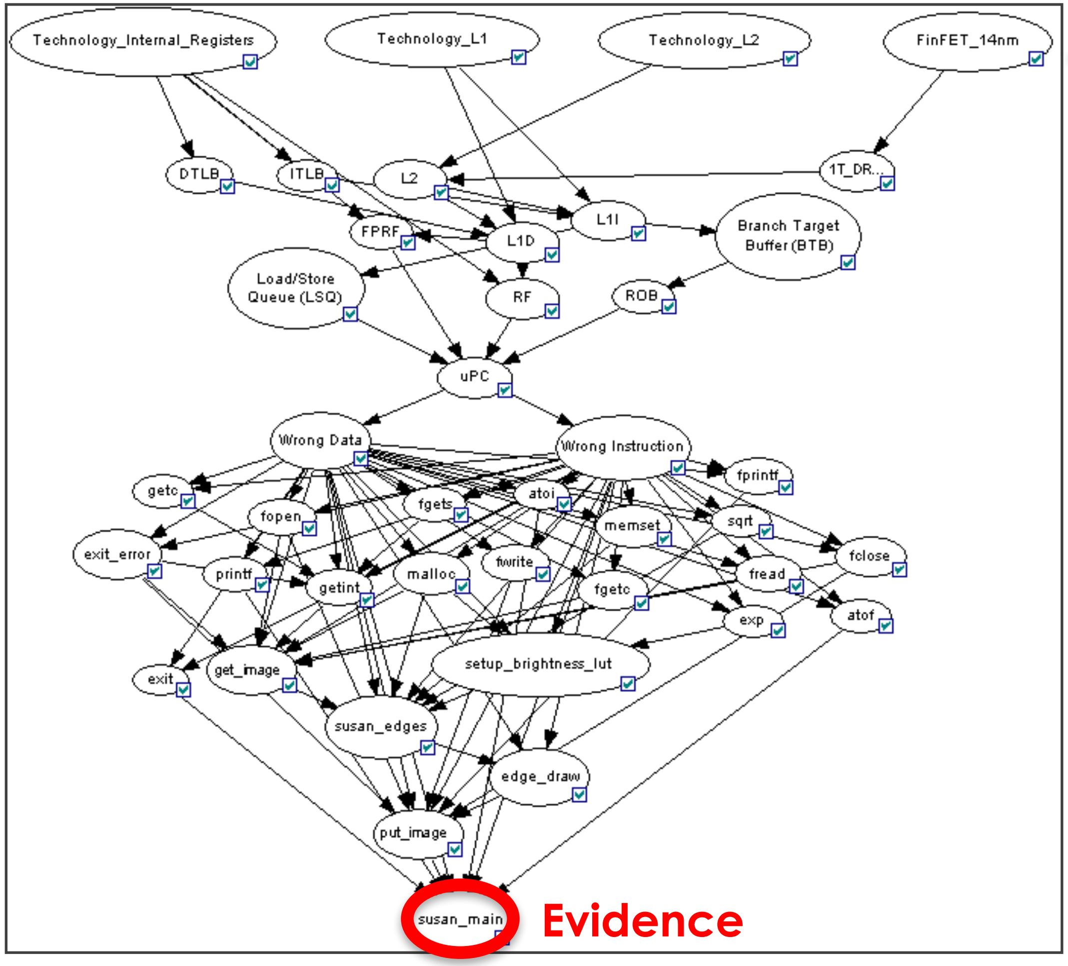
How does it work?



2 Forward inspection

Given the evidence that a node is in a given state (i.e., failure) which is the probability of correctness/failure observed at the application layer?

How does it work?



3

Backward inspection

Given the evidence that the application fails, which are the most probable roots of failure?

Agenda

- Context
- Problem
- Proposed Approach
- Validation
- Conclusion

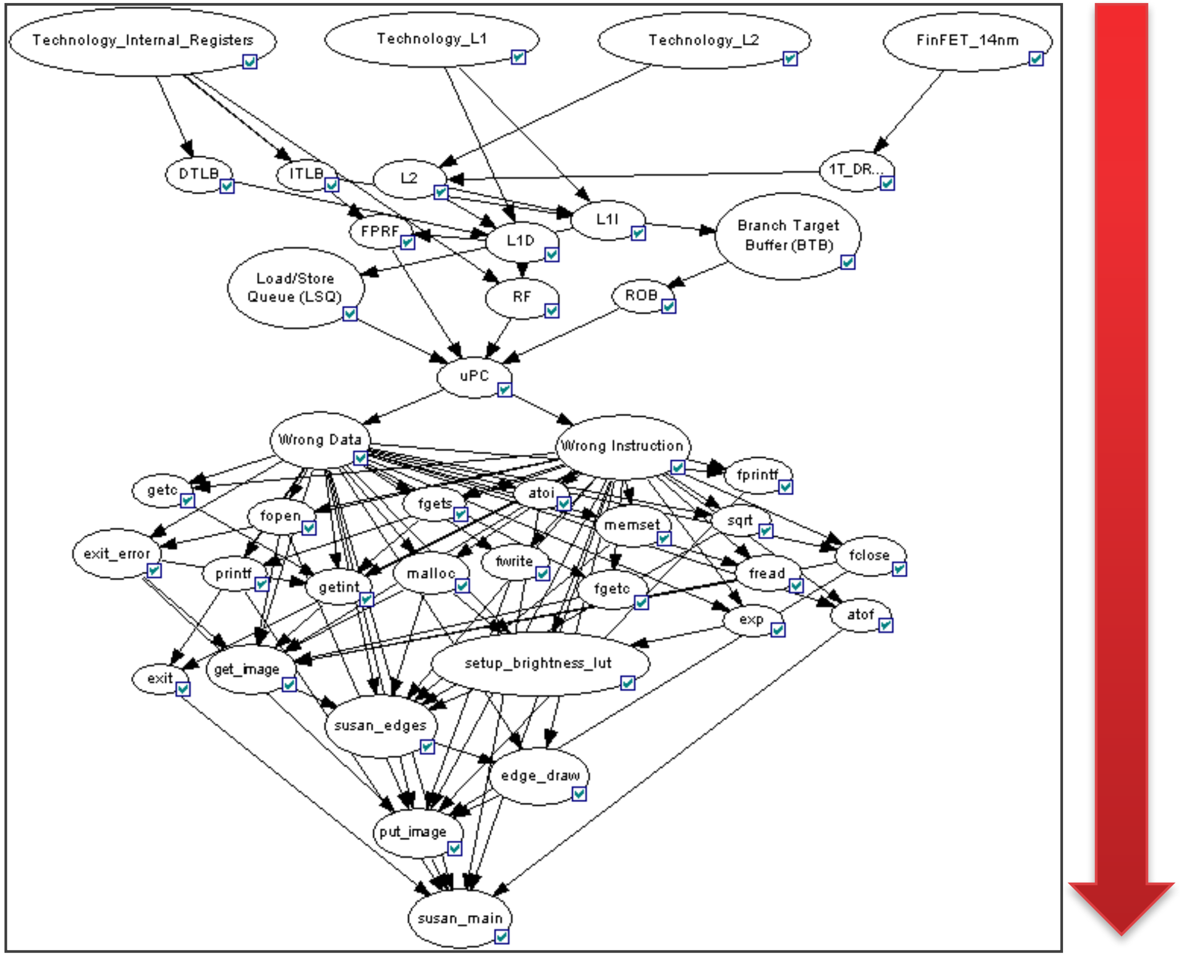
Validation

- Comparison with a uA Fault Injector [1]
- Case studies:
  - **MiBench [2]** is a suite of open-source software benchmarks that have been extensively used in reliability studies

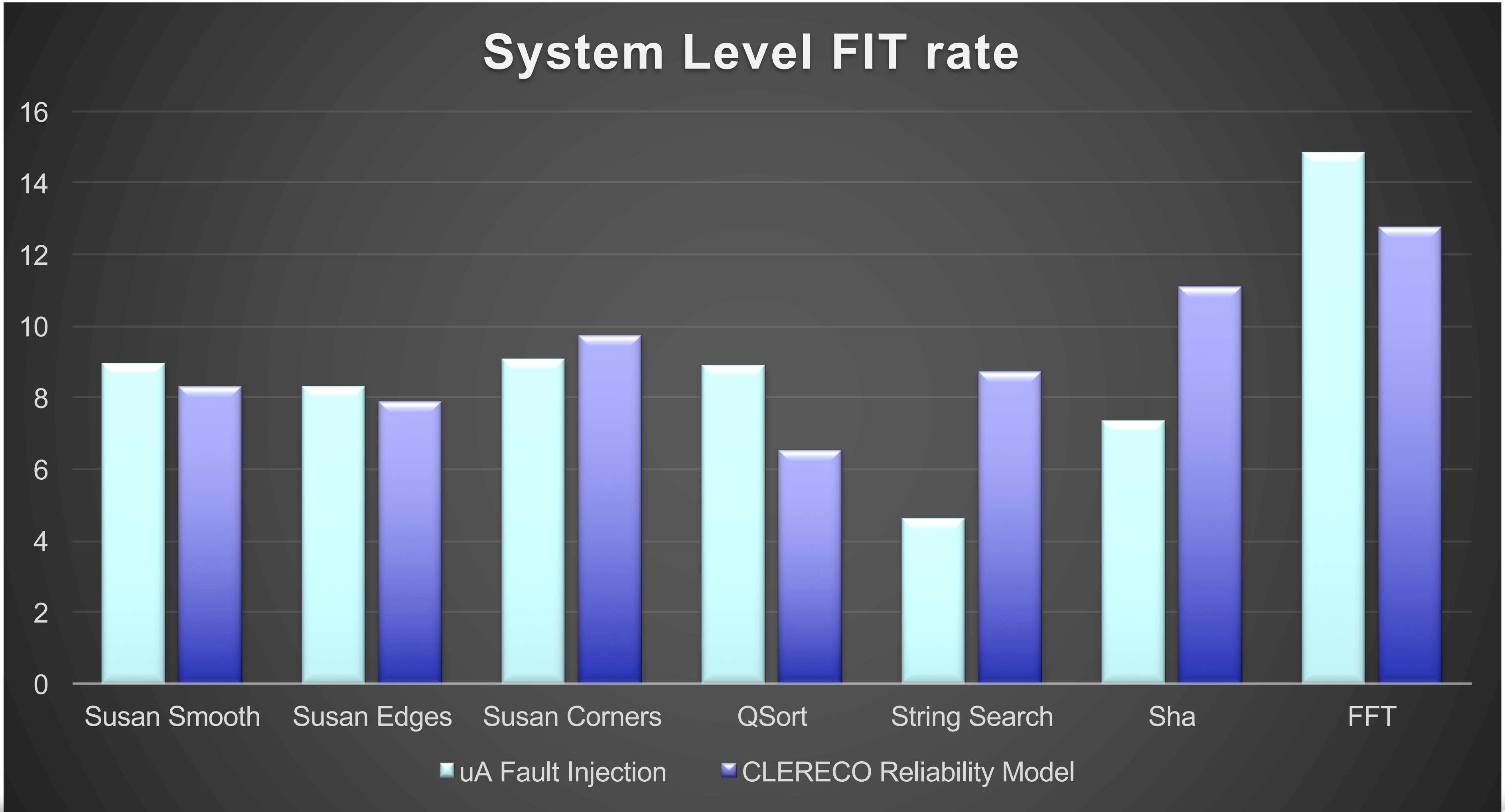
[1] GeFIN, IISWC 2015

[2] http://vhosts.eecs.umich.edu/mibench/

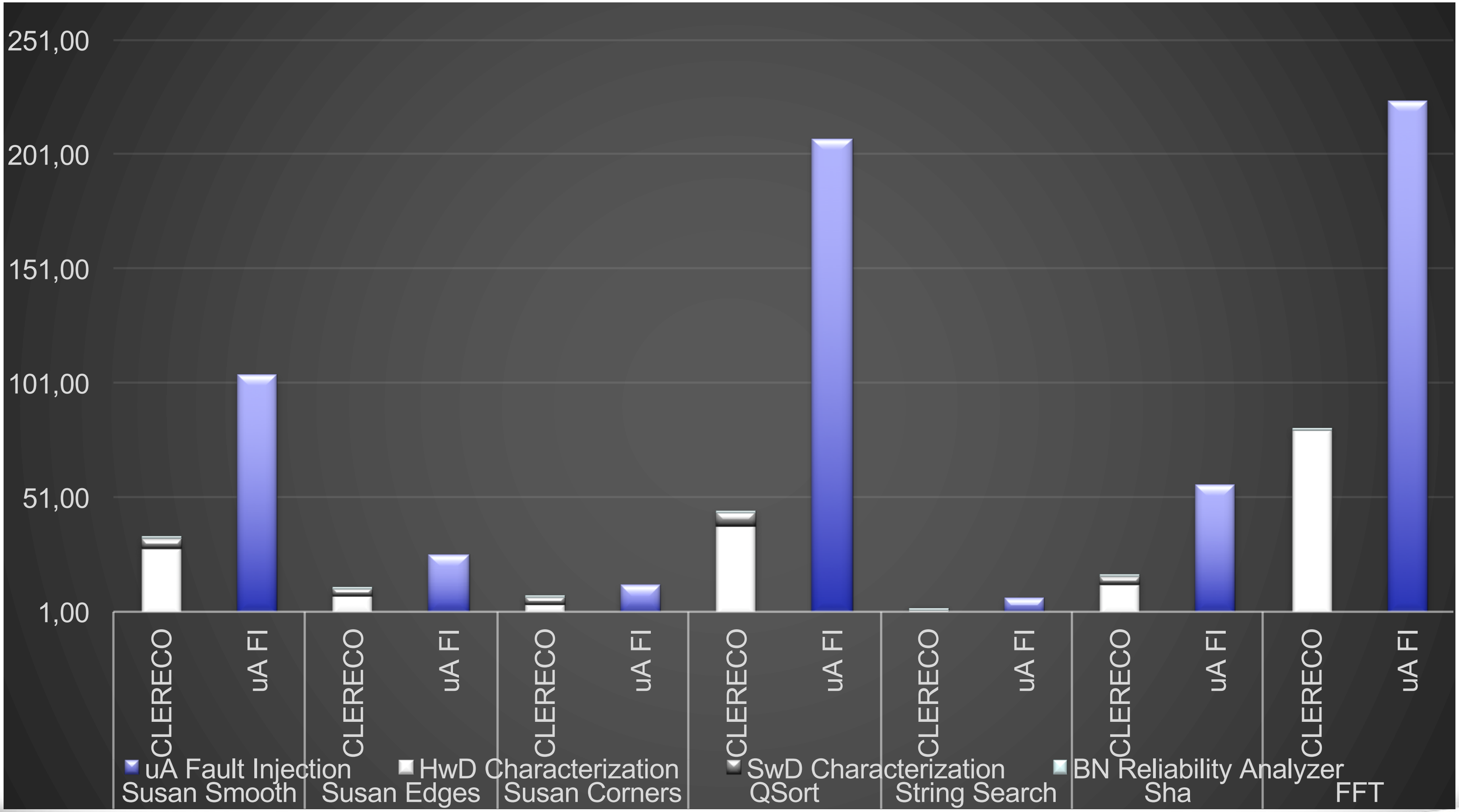
Global rel. analysis



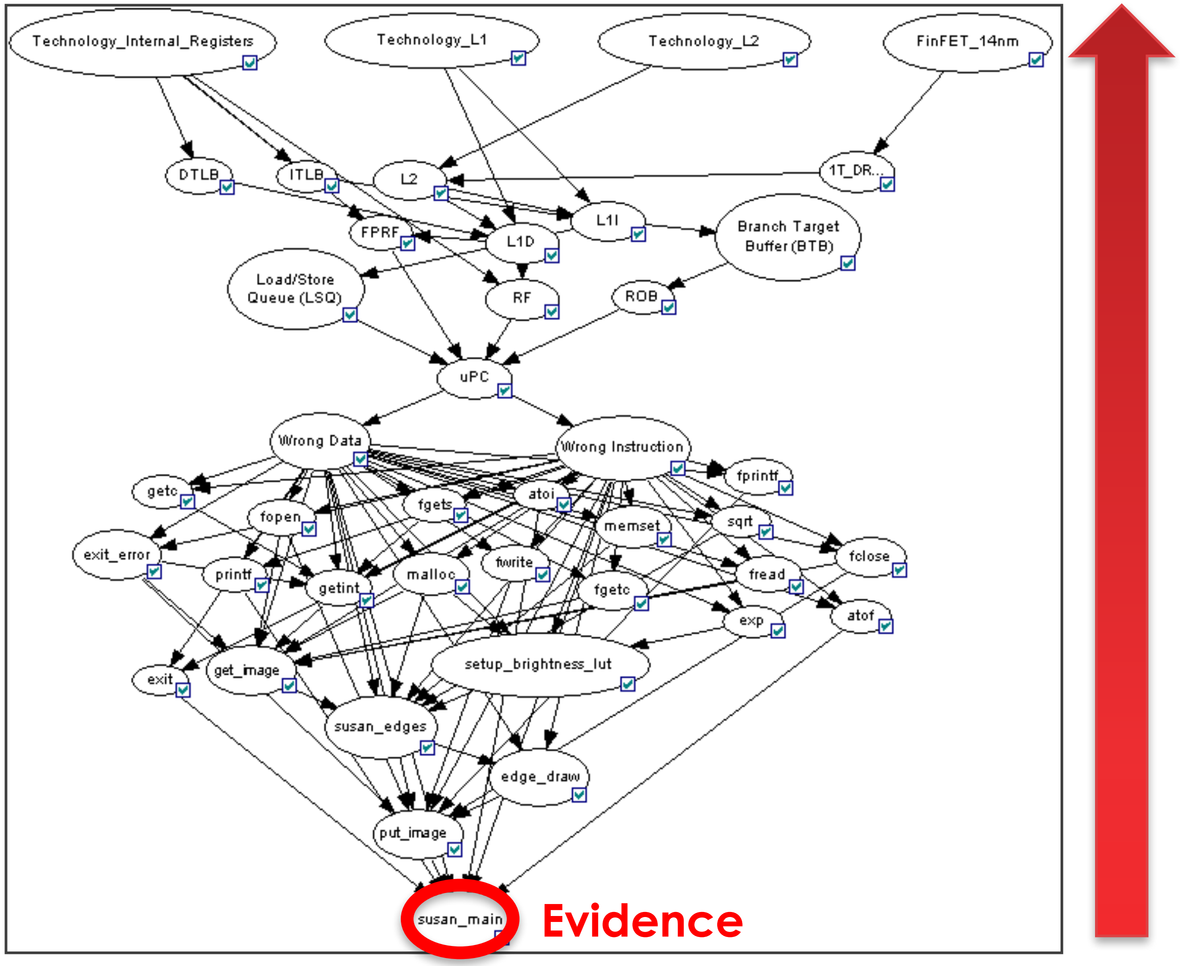
Experiments



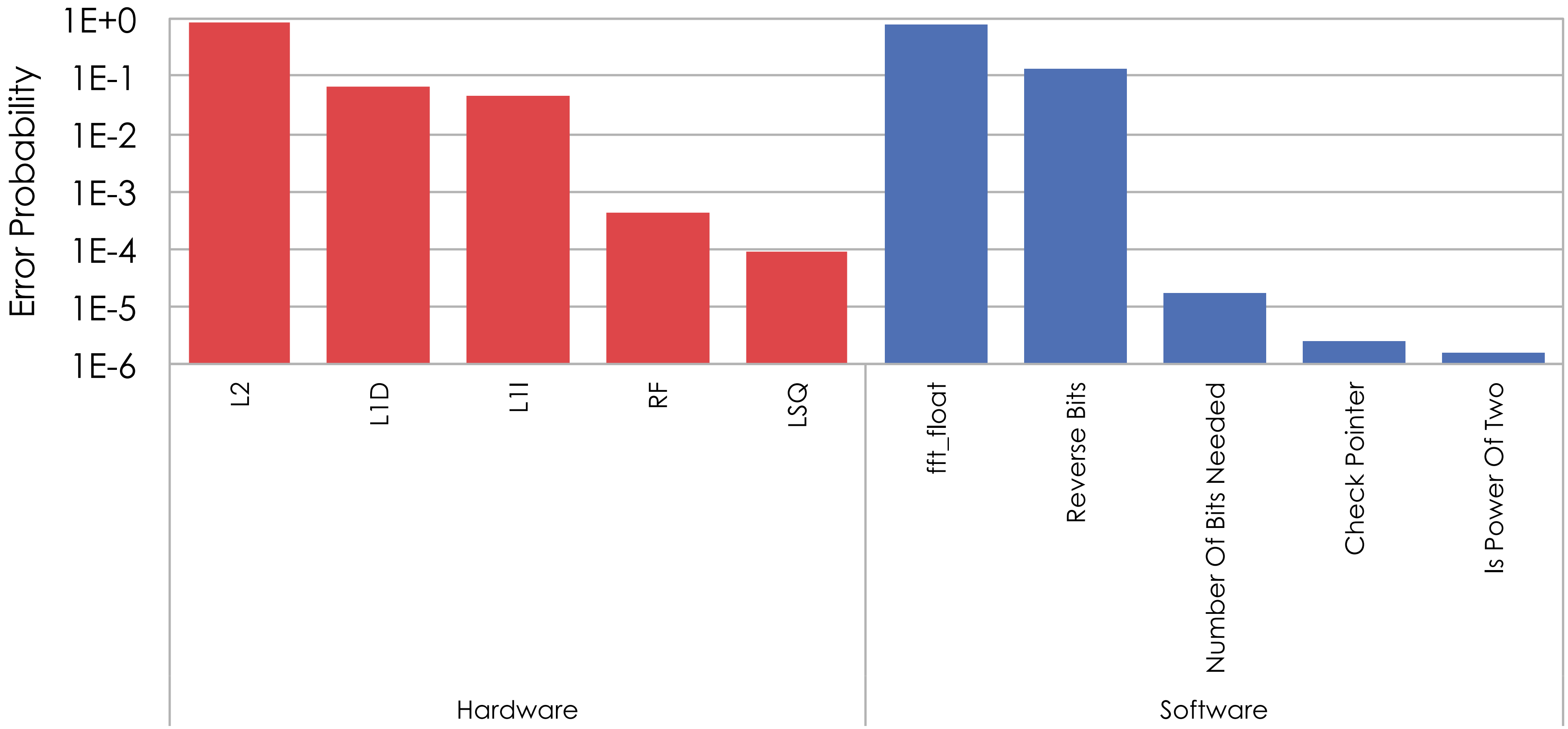
Experiments (hours of simulations)



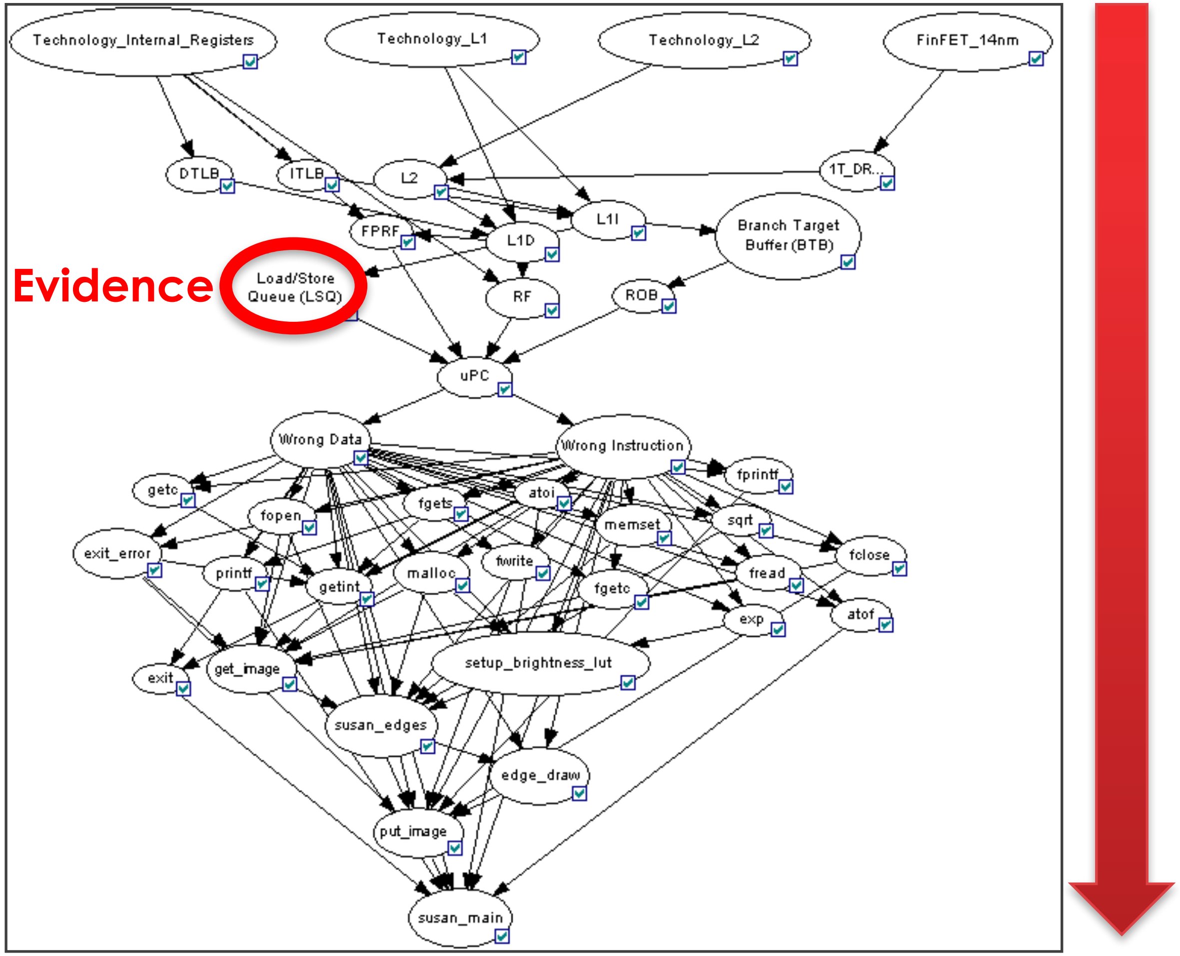
Backward inspection



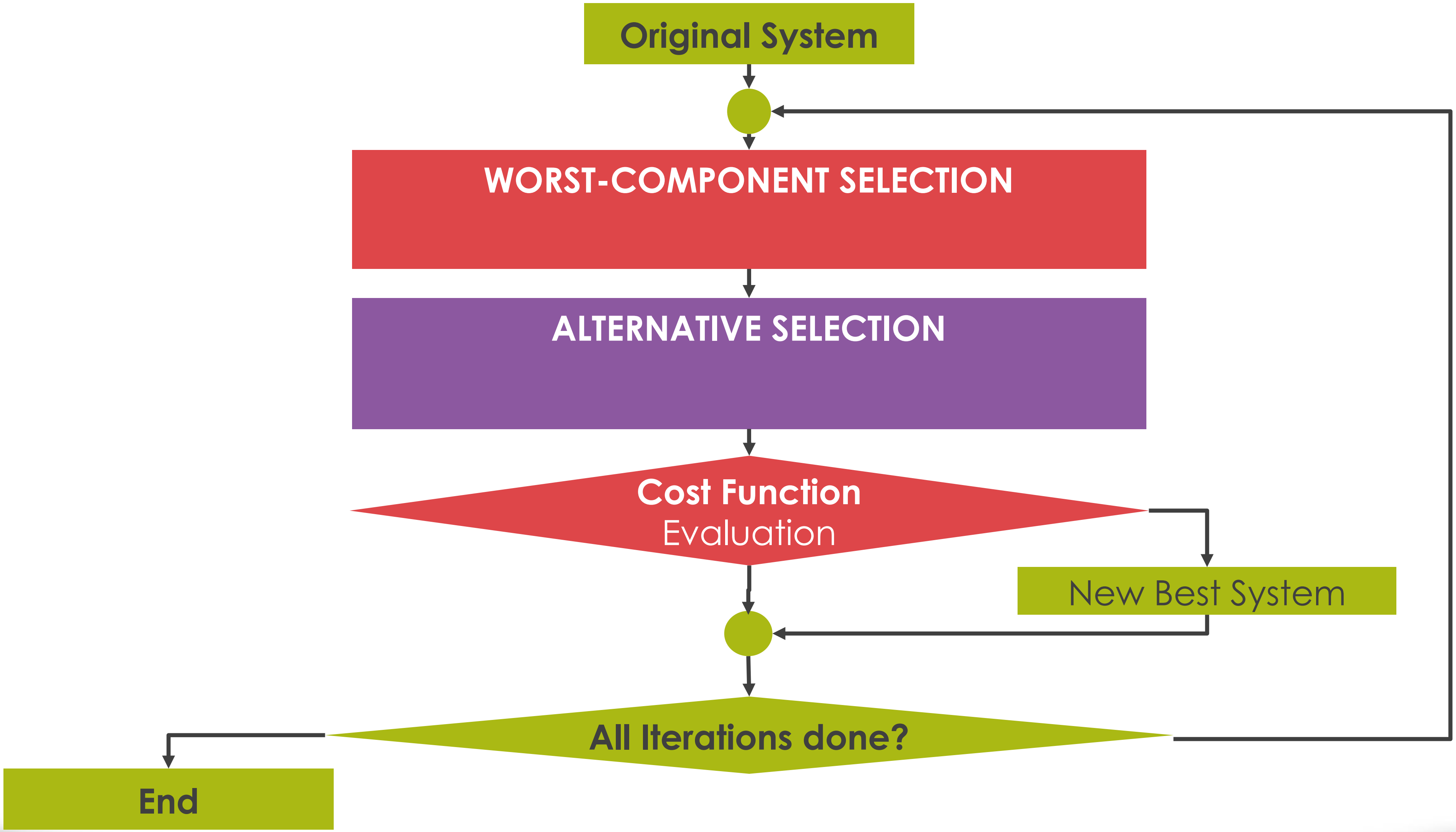
Experiments



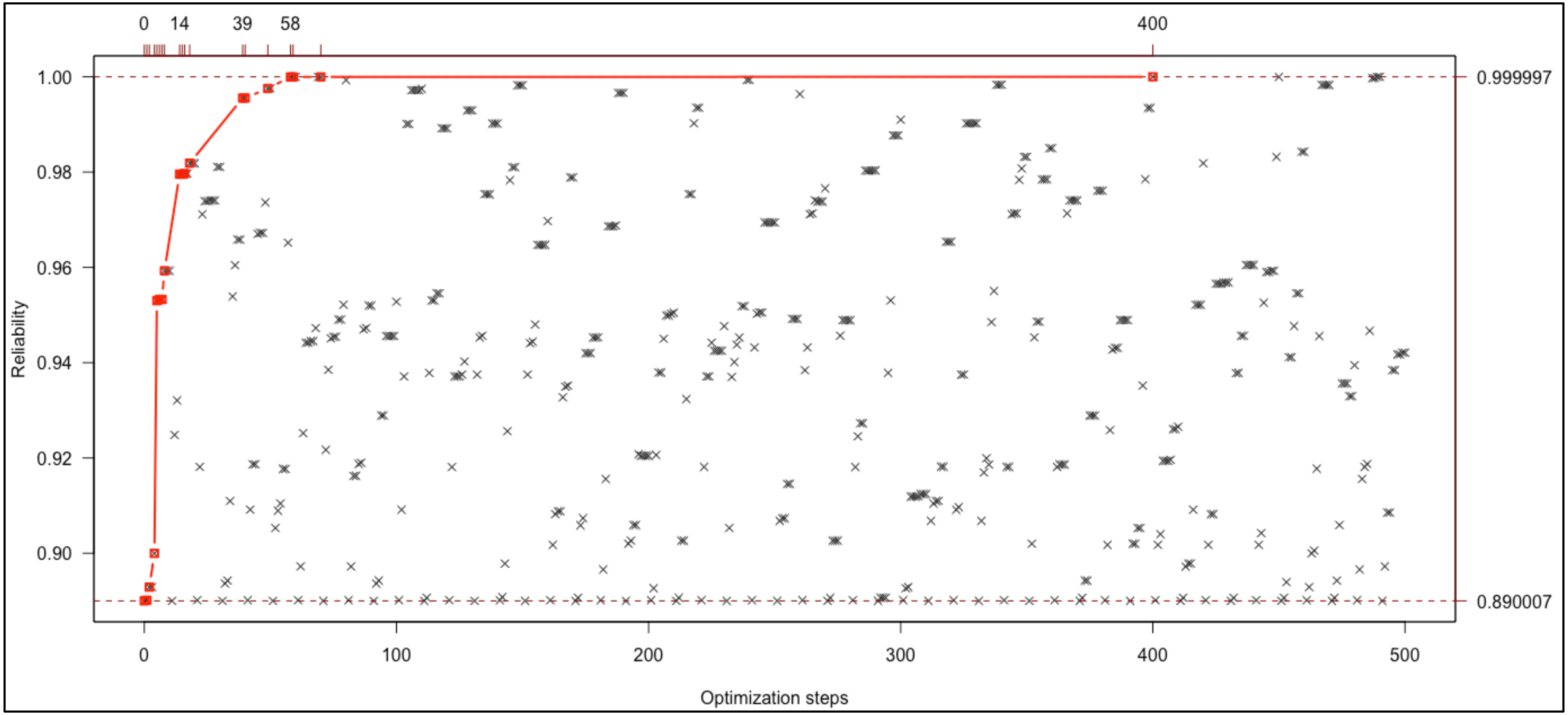
Forward inspection



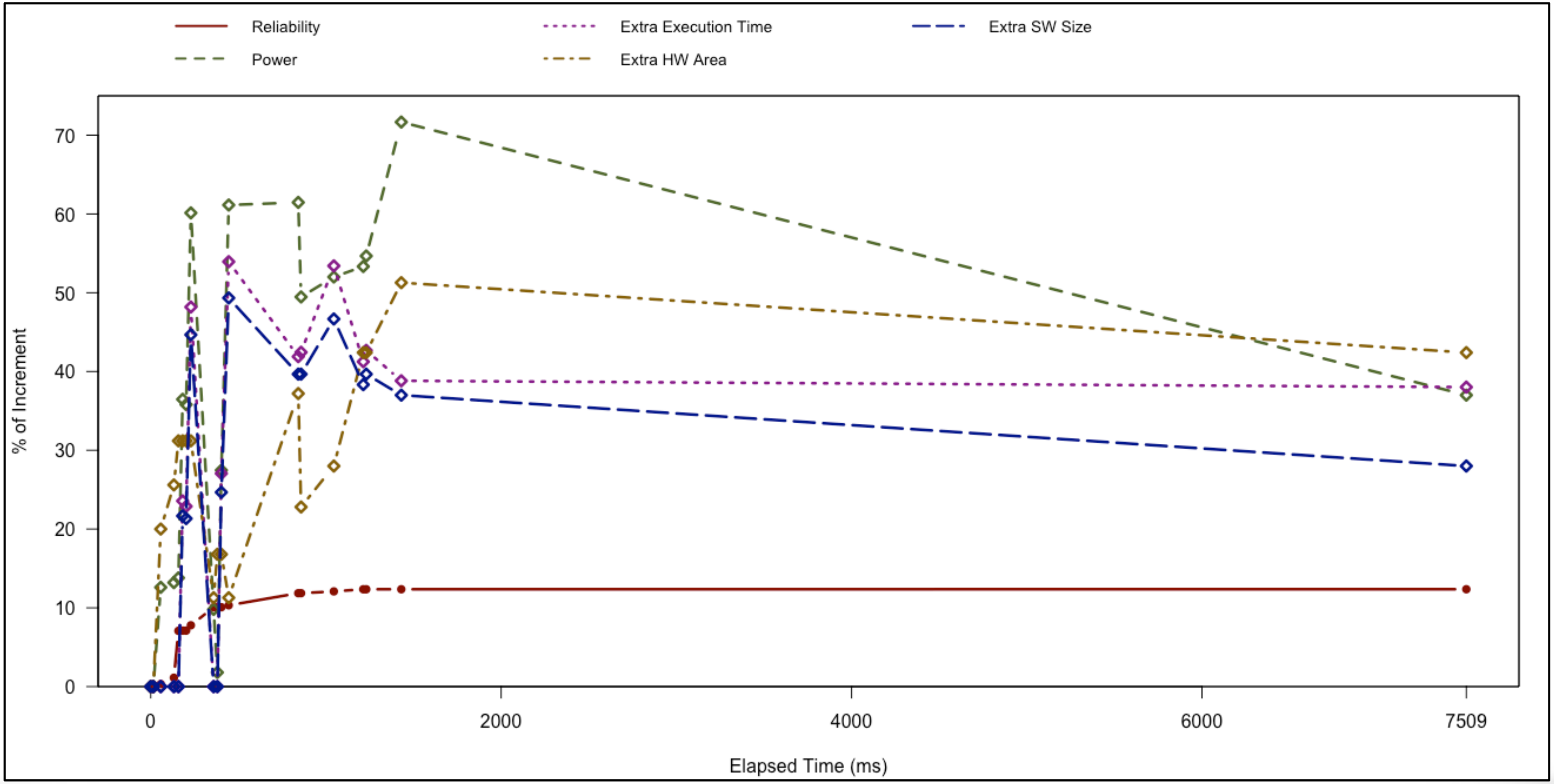
Identify the Best Implementation



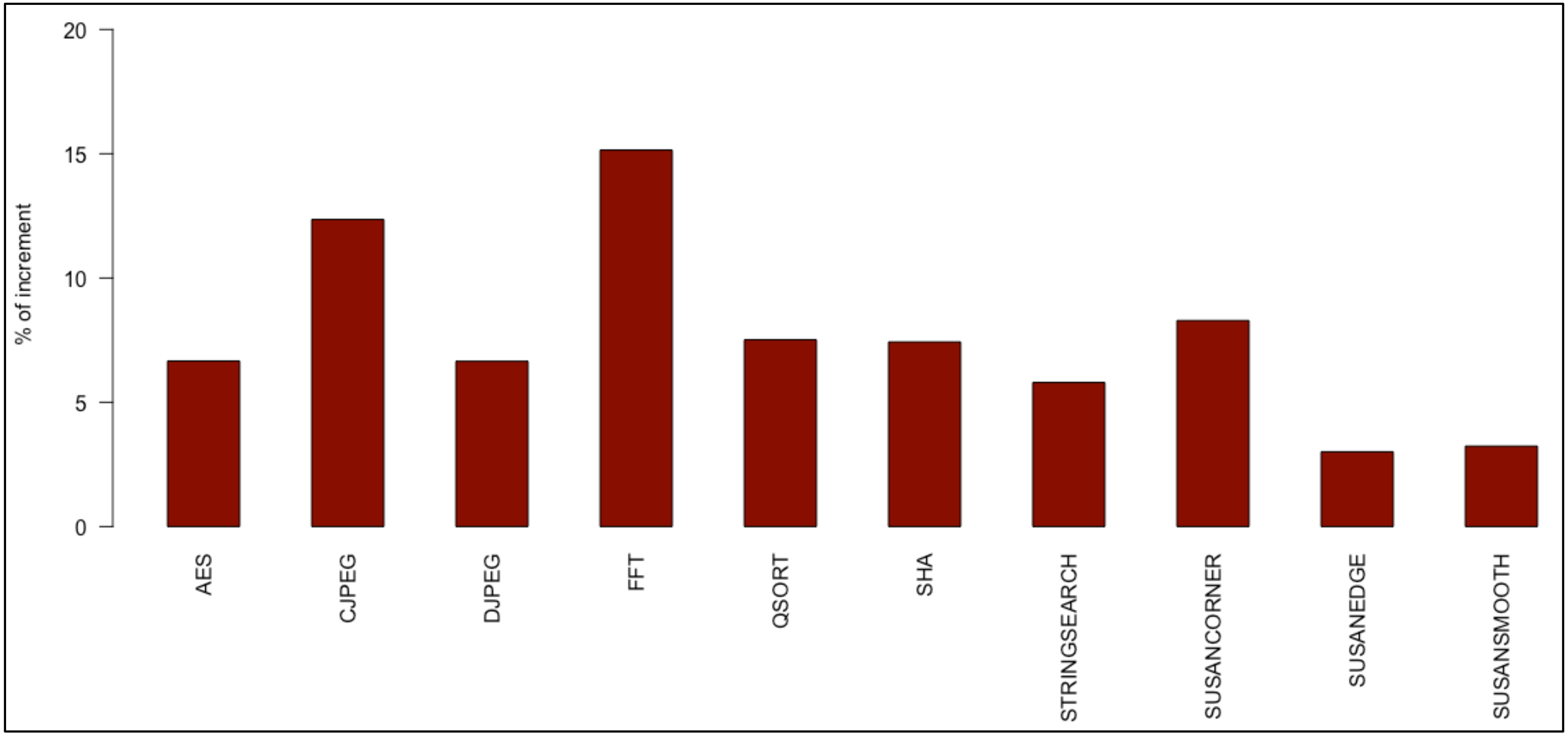
JPEG (MiBench)



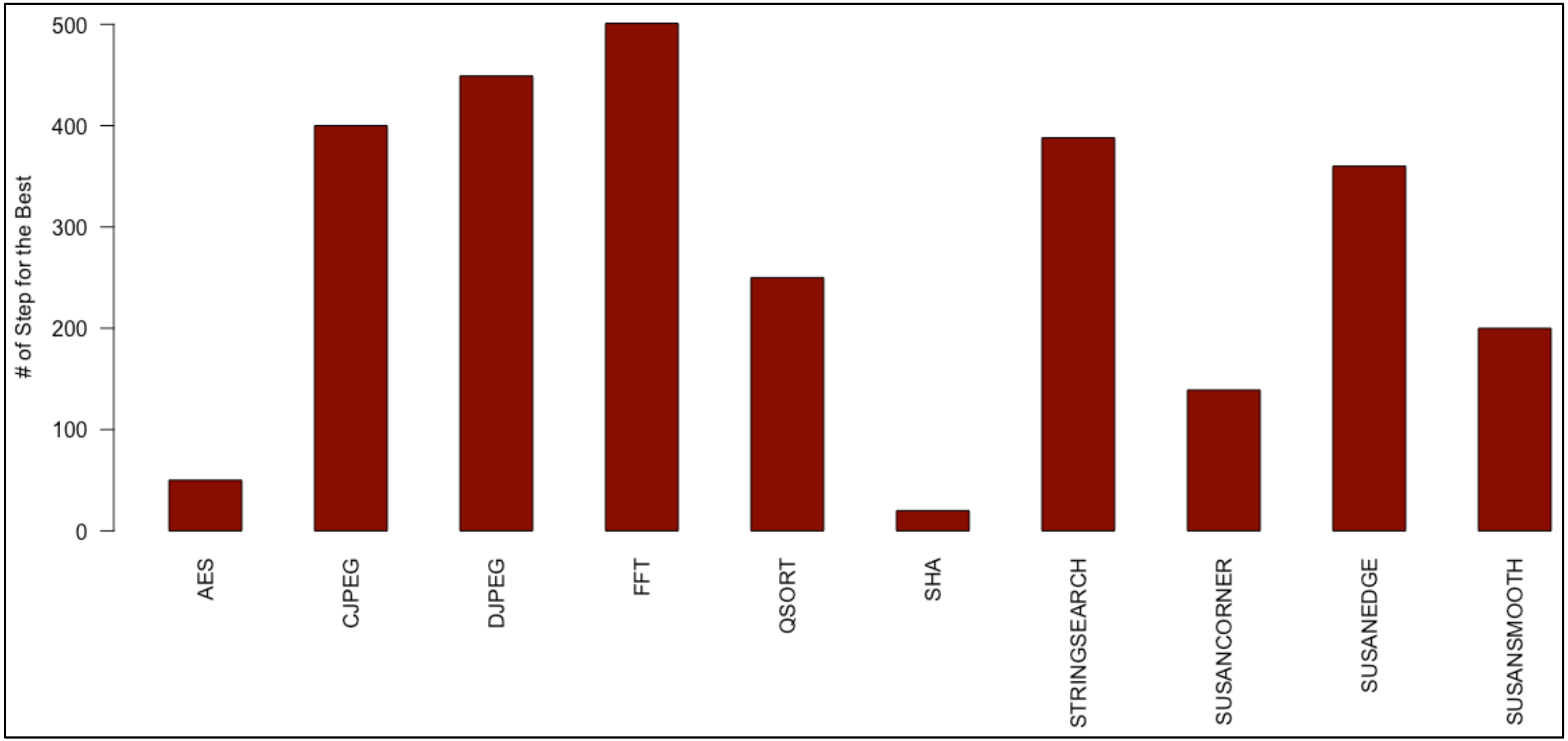
JPEG: Cost of Reliability



Experiments



Experiments



Conclusions

- A Comprehensive solution for System-Level Reliability analysis has been presented

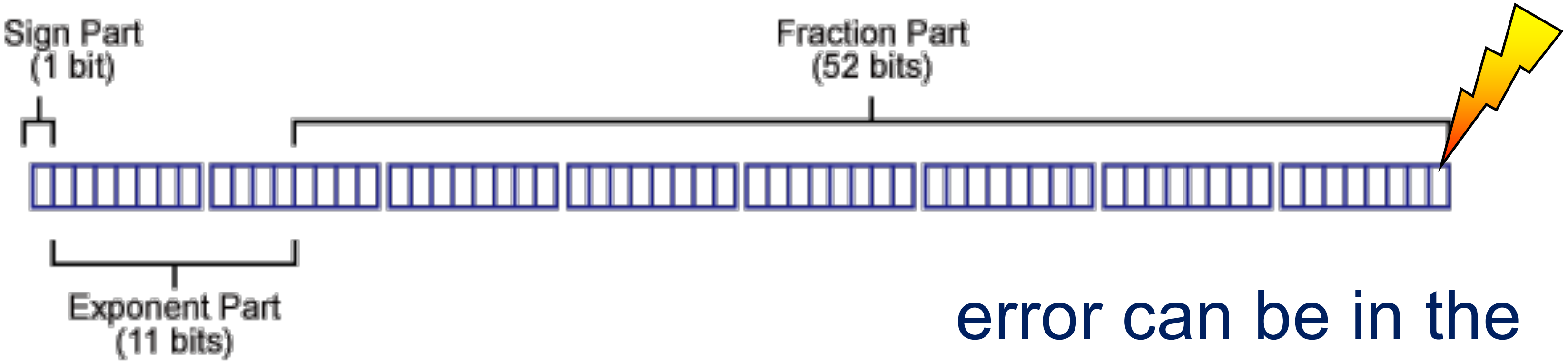
FOLLOW US

http://www.clereco.eu

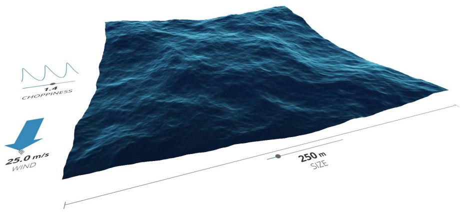
Clereco.eu

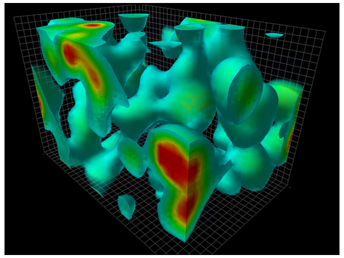
What’s next

Not all errors are critical!



error can be in the float intrinsic variance





Values in a given range are accepted as correct in physical simulations

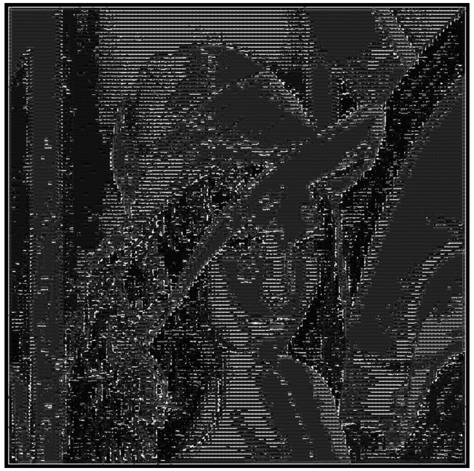
What’s next

Non-critical Error **Critical Error**

Golden

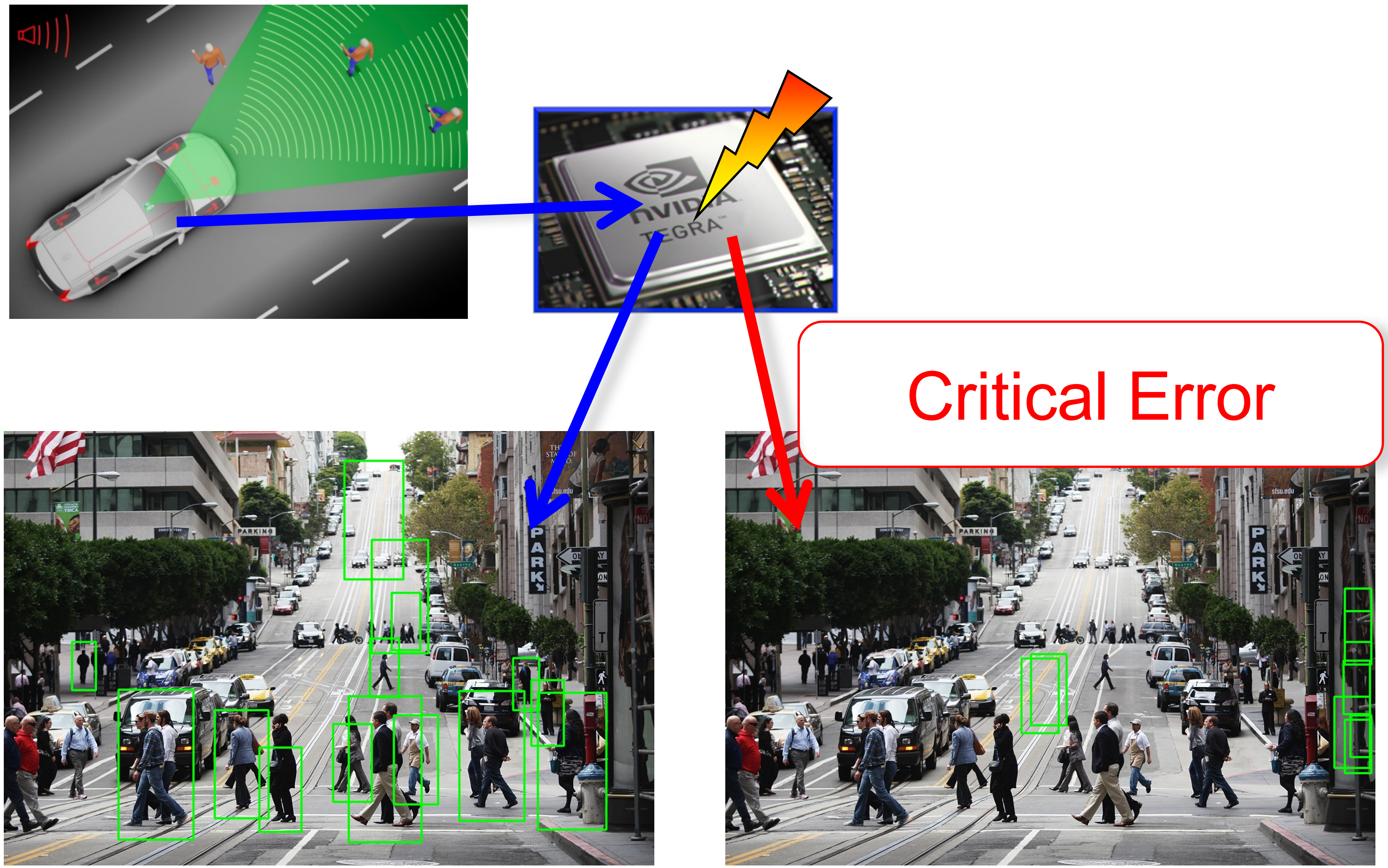
40dB

30dB



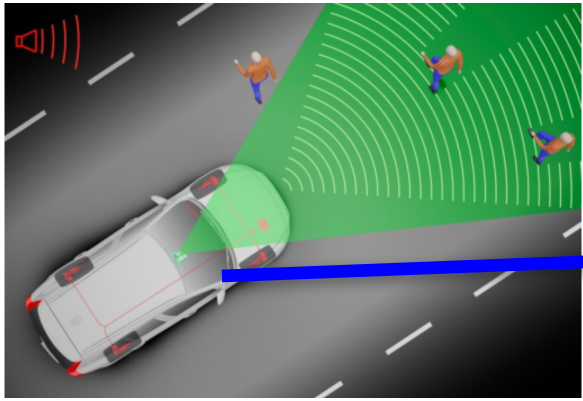
5dB

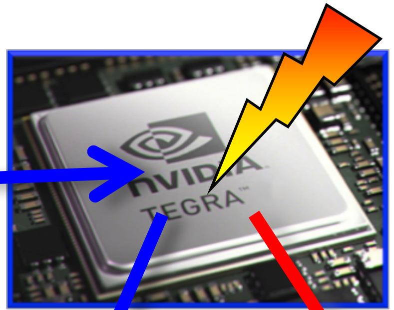
What’s Next



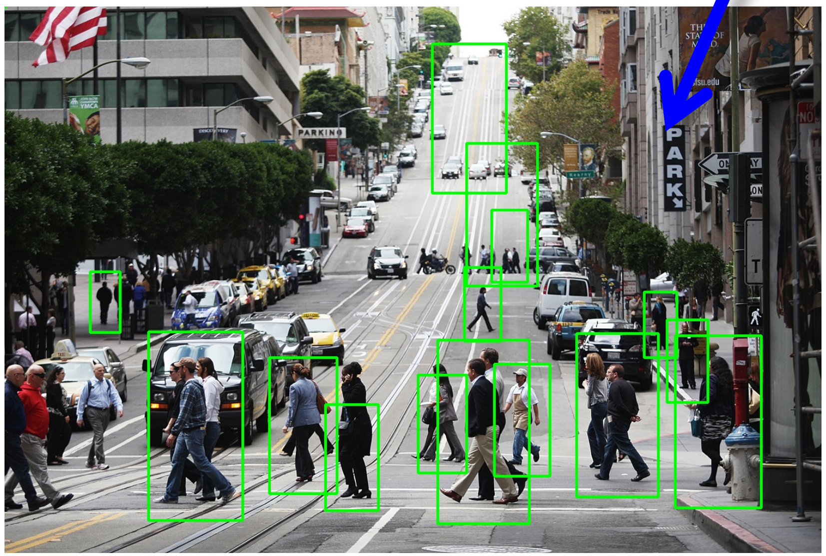
P. Rech’s Courtesy

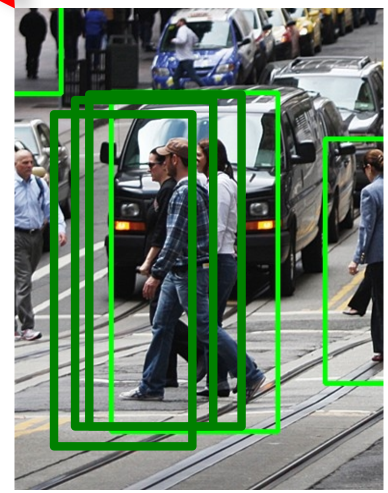
What's Next





Non-Critical





P. Rech's Courtesy

Advertising

http://www.lirmm.fr/DDECS2019