

Fault/error injection Basics

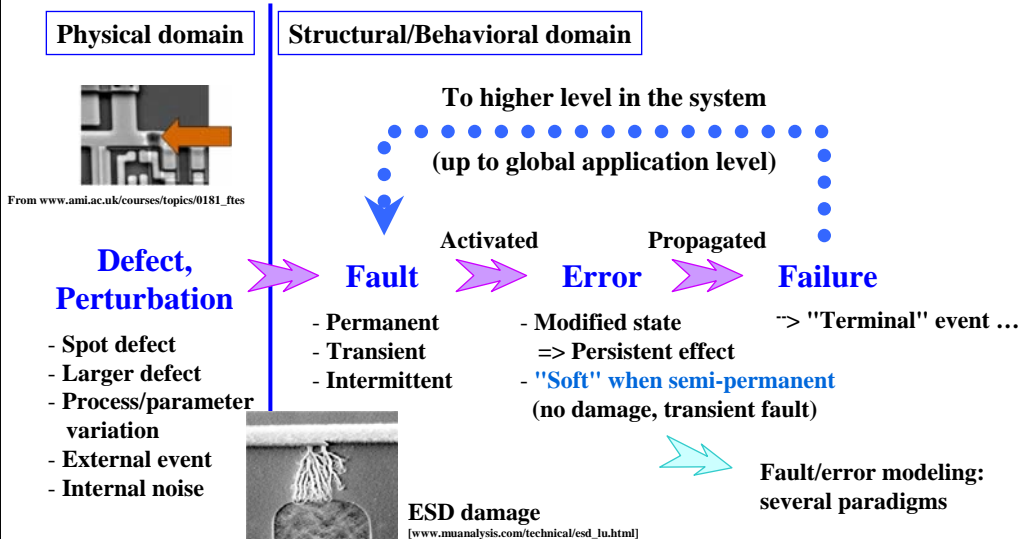
Régis Leveugle



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Cause and effect: some terminology



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Why fault (error) injections ?

- Need for early analyses to evaluate the criticality of faults in the various parts of a circuit
 - => First step towards "justified user confidence"
 - => Early identification of problems means less \$ or € for modifications
 - => Identify the real locations to be protected
 - => Minimal hardware/software hardening with respect to the application requirements ("pragmatic hardening": 10% overheads ...)

- Evaluation of the system-level failure rate (from raw SER to system failure rate) => FIT

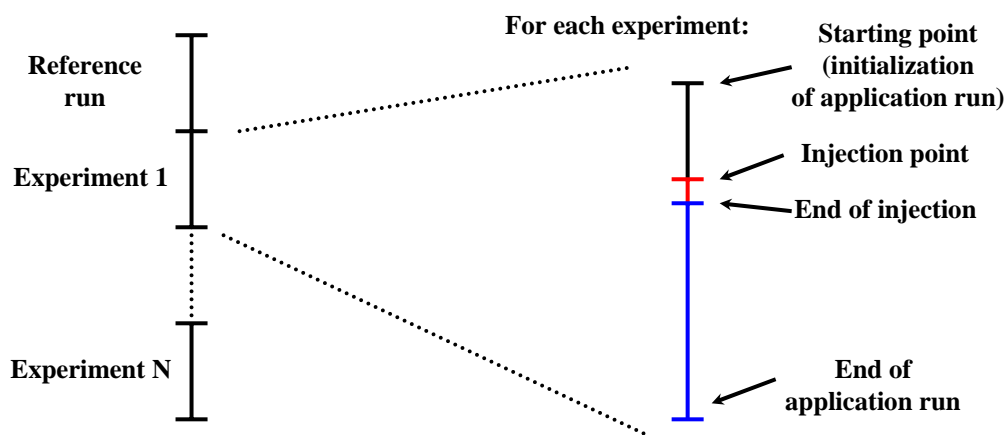
- Validation of built-in protection mechanisms



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

A fault injection campaign (basic principle)



Injection point: may be controlled/specified or not (e.g. particle beam)



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Parameters ...

- **Where ?**
 - ◆ Critical locations ?
 - ◆ Registers vs. combinatorial logic

- **When ?**
 - ◆ Critical cycles
 - ◆ Time within cycles (SETs)

- **What ?**
 - ◆ Fault/error model (bit-flip, stuck-at ...)
 - ◆ Spatial multiplicity
 - ◆ Temporal multiplicity

Exhaustive vs. random

Application-oriented vs. general



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Many fault injection approaches

- **Hardware-based**
 - ◆ Pin-level, particle flux, laser, CEU, ...

- **Software-based (SWIFI)**
 - ◆ Using a debugger, specific OS services, modifications in programs, ...

- **Simulation-based**
 - ◆ Various levels: system (SystemC) and co-design, ISS, RTL, gate, switch, electrical, physical ...
 - ◆ Precision vs. simulation time and complexity: trade-offs

- **Emulation-based**
 - ◆ Prototyping equipments, FPGAs, SoPCs

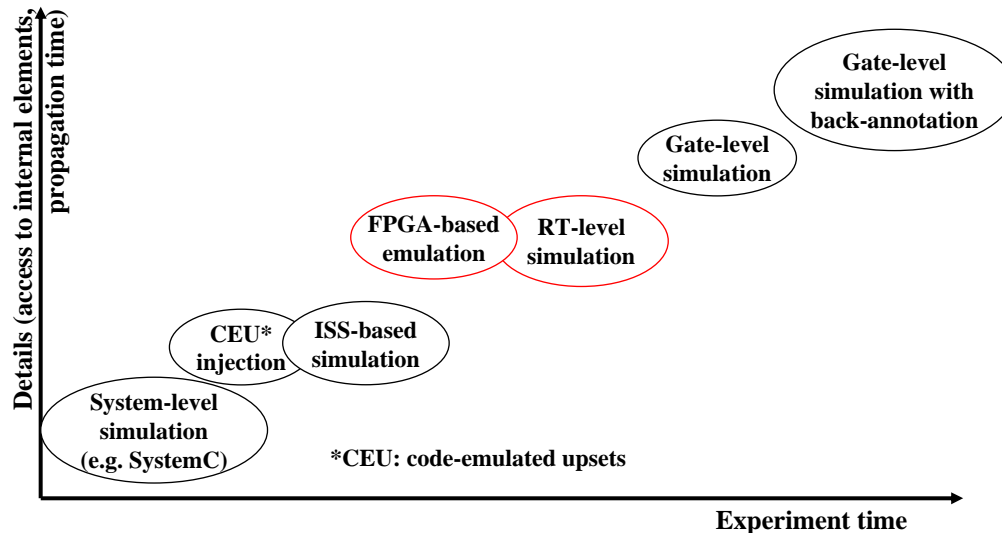
- **Hybrid**



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Fault-injections: alternative approaches



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Why early fault injections ?

- Need for early analyses to evaluate the criticality of faults in the various parts of a circuit
 - => First step towards "justified user confidence"
 - => Early identification of problems means less \$ or € for modifications
 - => Identify the real locations to be protected
 - => Minimal hardware/software hardening with respect to the application requirements ("pragmatic hardening": 10% overheads ...)

- Evolution of circuit sensitivity => evolution of fault/error models
 - ◆ From single to multiple bit flips
 - ◆ Keeping experiment time compatible with required TTM ...
 - ◆ RTL vs. gate-level simulation time: two orders of magnitude



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Are early fault injections worthwhile ??

- Lack of structural knowledge ...
 - ◆ Case study [JETTA 03] – Same behavior, two description levels with noticeable differences in implementation details (e.g. counter vs. shift reg.)
 - ◆ Very good agreement Err.Trans./Bit-flips and Archi / μ -Archi
 - ◆ Higher level tends to be pessimistic
- Actually useful data ?
 - ◆ Small percentage of possible erroneous configurations and possible sequences actually recorded => hardening targets pointed out !
 - ◆ Case study [IOLTS 04, JETTA 05] – 8051 μ controller running several applications, classification of results
 - Few percents of incorrect RT-level -> correct gate-level
 - Just a few correct RT-level -> incorrect gate-level
- Simulation breaks at RT level ("crash") ...
 - ◆ Do not preclude useful outcomes



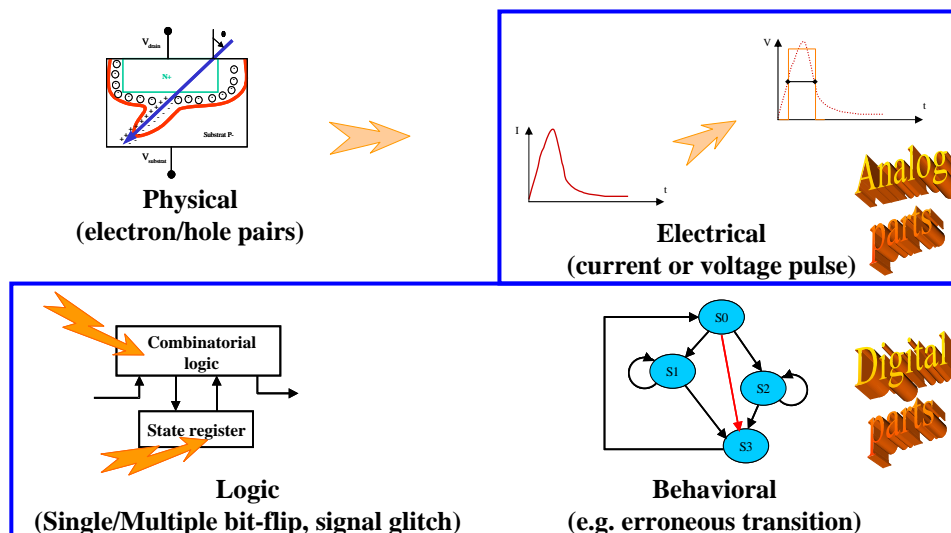
Problems due to don't cares ...



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Levels of modeling for SEUs/SETs



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Injections: summary up to soft errors

- **SET/SEU Sensitivity Evaluation Strategy: one simulation methodology per abstraction level**

- **2D/3D Physical Level Simulation**
 - ◆ Each PMOS/NMOS transistor is characterized
 - ◆ Id output current prediction for any logic gate

- **SPICE/Logic Simulations**
 - ◆ Duration of transient voltage pulses (depending on output capacitance)
 - ◆ Transient pulse propagation analysis through the circuit (masking analysis)

- **Logic Level Simulation**



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle

Higher levels towards the application

- **Extension of the logic simulations to the whole circuit**

- **Higher abstraction level: RT-level simulation (or emulation)**
 - ◆ New masking types related to the application
 - Unused parts/functions in the circuit
 - Lifetime of memorized variables
 - ...
 - ◆ Can start from soft-error configurations obtained at lower levels

- **Recently: system-level analyses (Co-design framework or SystemC descriptions)**

- **Trade-offs between description level, experiment time and accuracy of the analyses results**



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle


Link between analysis levels

Description level	Analysis	Qualitative info.	Quantitative info.
Behavioral/RTL Soft errors	Behavioral simulation (emulation)	Error -> failure (application point of view)	P(failure error)
Gate level SET (voltage), SEU	Gate level simulation (timed)	Glitch -> Error (latched) + refinement previous analysis	P(error glitch)
Electrical/Physical SET (charge, current)	Electrical/Physical simulation	Particle or physical event -> glitch or bit-flip	P(glitch particle) P(bit-flip particle)

Top-down: design flow
 Bottom-up: libraries

Estimation principle of application failure (limitations to be considered at high levels):

$$P(\text{failure}) = \underbrace{P(\text{failure}|\text{error})}_{\text{Critical nodes}} * \underbrace{[P(\text{bit-flip}|\text{particle}) + P(\text{error}|\text{glitch})]}_{\text{Sensitive nodes}} * \underbrace{P(\text{glitch}|\text{particle})}_{\text{Sensitive nodes}} * \underbrace{P(\text{particle})}_{\text{Environment}}$$


 TIMA Laboratory
 46 av. Félix Viallet - 38031 Grenoble Cedex - France
 R. Leveugle

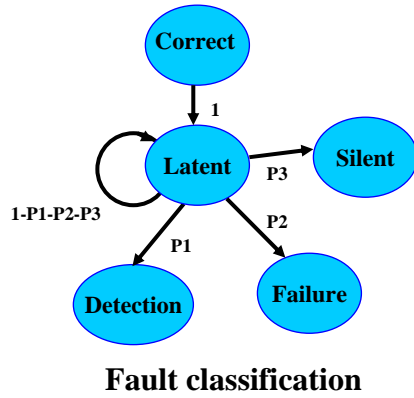
RT-level dependability analysis

- Why RTL ?
 - ◆ Early in design flow, but with detailed enough knowledge of the circuit (cycle accurate, list of registers)
 - ◆ Preliminary functional analysis for quick corrections at architectural level
- Faults/Errors
 - ◆ Single and multiple bit-flips / erroneous transitions
 - ◆ Transient stuck-ats on selected nodes
- Classification or error propagation analysis
 - ◆ Identification of feared events and evaluation of their probability
 - ◆ Analysis of error configuration sequences / Identification of critical configurations when combined with classification

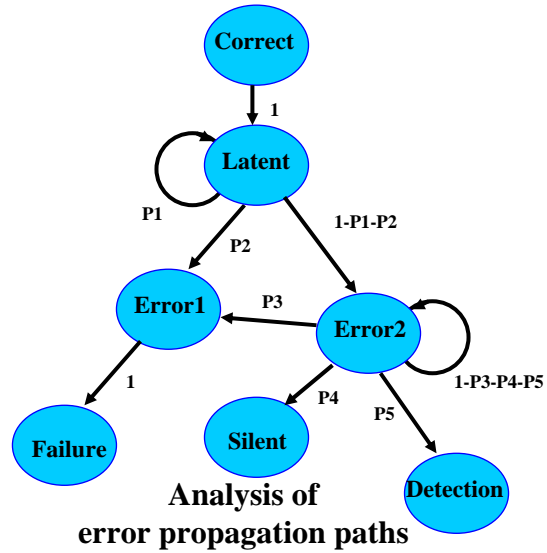


Dependability analyses: possible outcomes

Cycle-by-cycle comparisons



Fault classification



Analysis of error propagation paths

+ specific "crash" state for RTL analysis



TIMA Laboratory
46 av. Félix Viallet - 38031 Grenoble Cedex - France

R. Leveugle