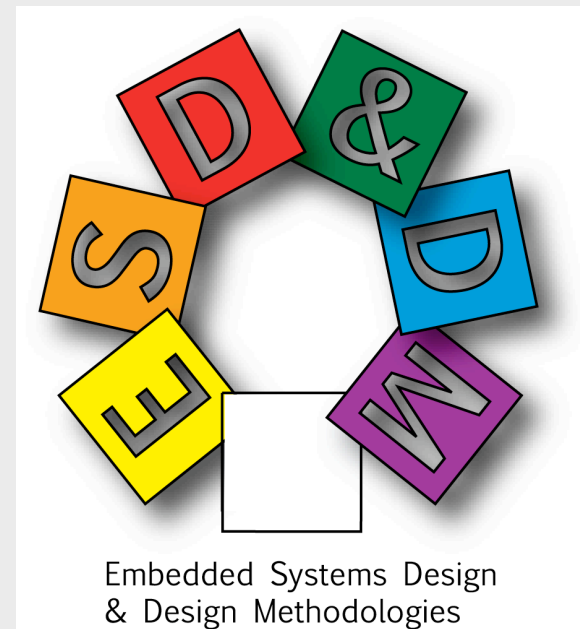


Dependable Systems & Design Methodologies

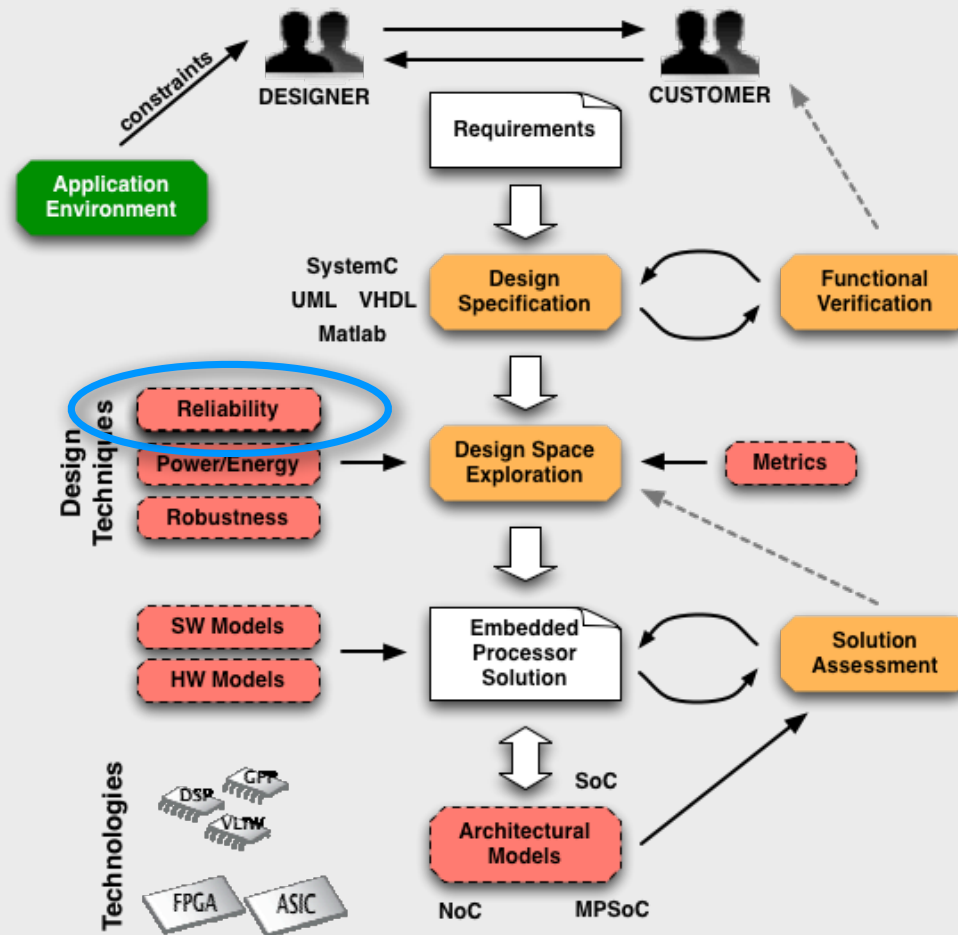
Cristiana Bolchini - bolchini@elet.polimi.it

who we are

- Part of the System Architecture group, working in the area of Embedded System Design and Design Methodologies



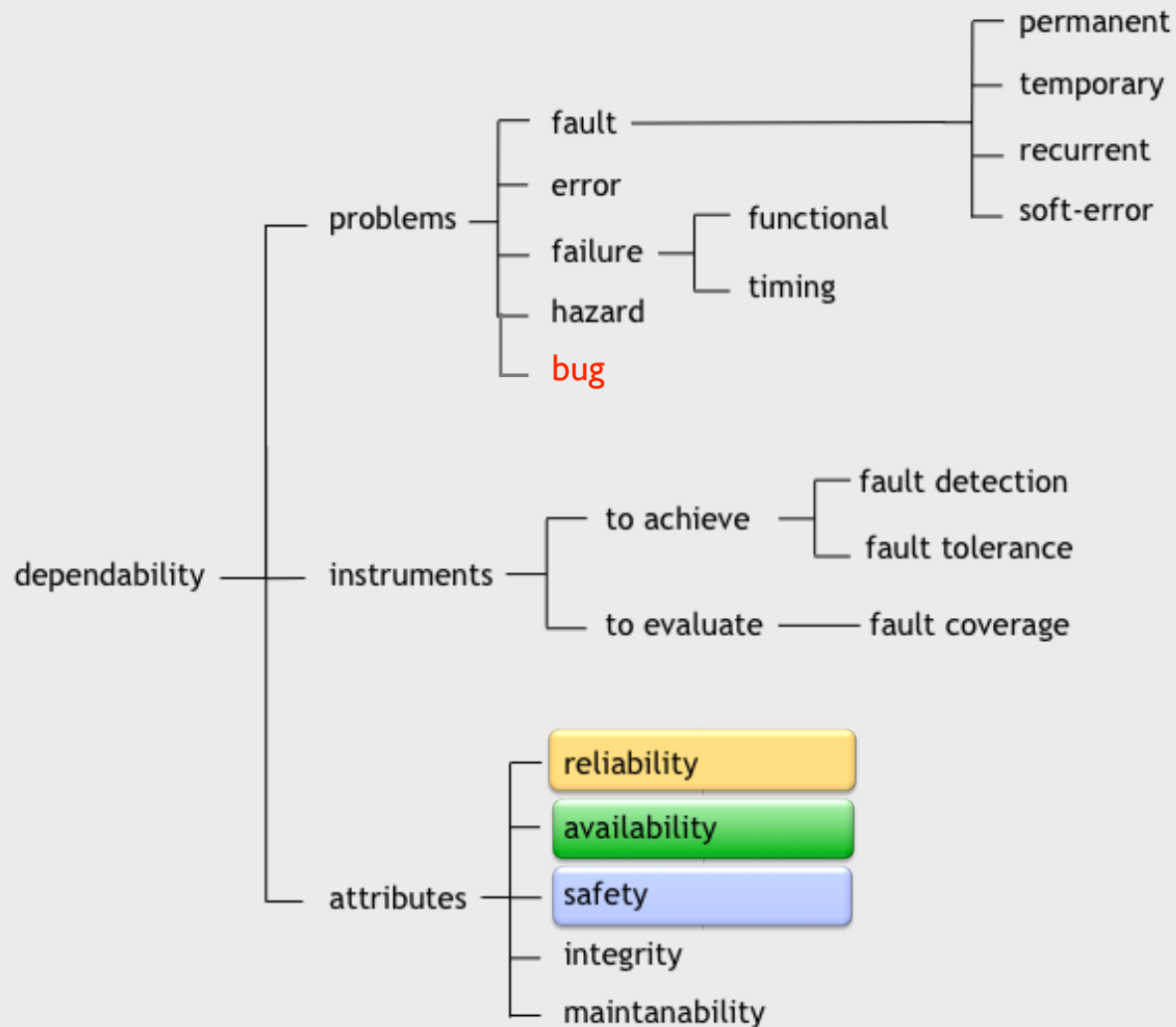
focus



dependability

- basic elements and properties:
 - defect, fault, error, bug
 - reliability, availability, safety
 - fault detection & tolerance

context: dependability²

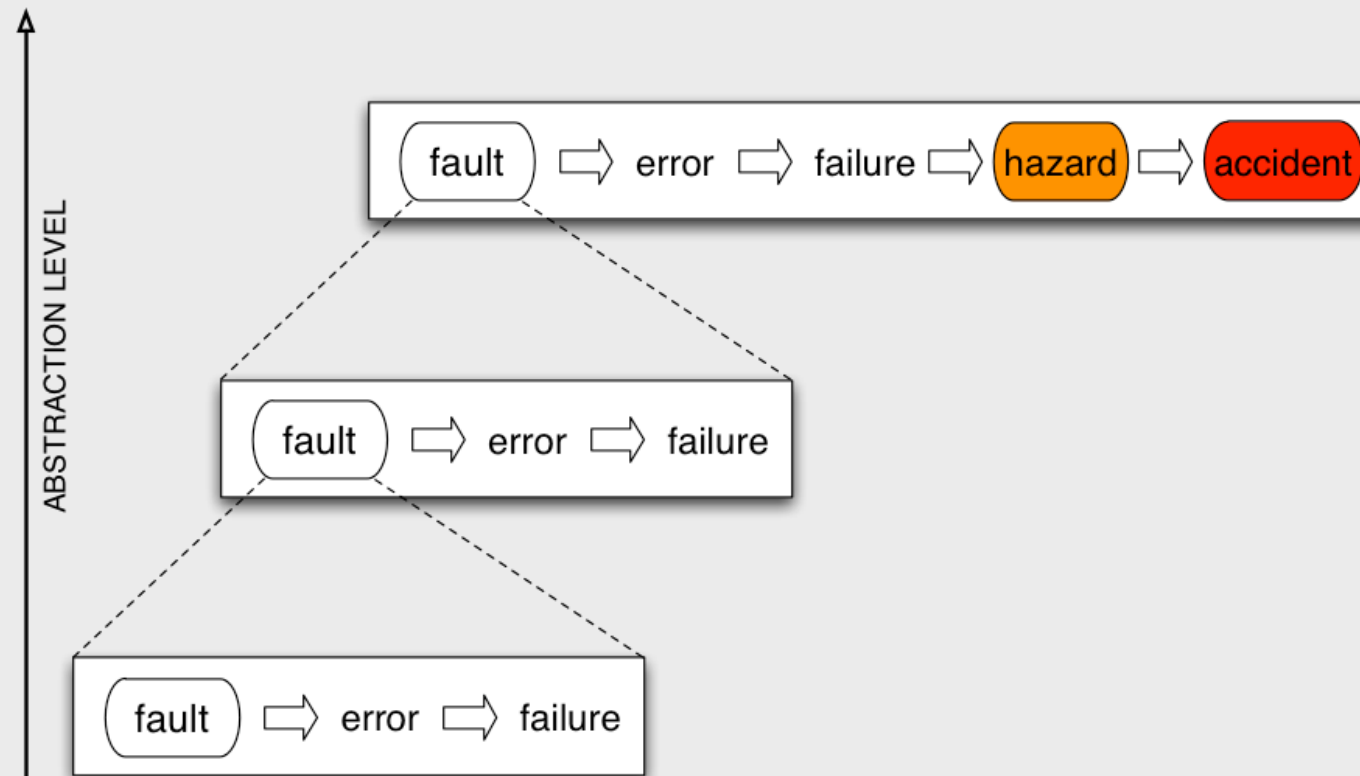


defects, faults, errors

- physical defects:
 - fabrication defects (missing or extra material)
 - material degradation over time and/or environment, wear-out
- faults
 - behavior due to defects, and
 - abstract model due to defects
- errors ...

abstraction levels²

fault hierarchy

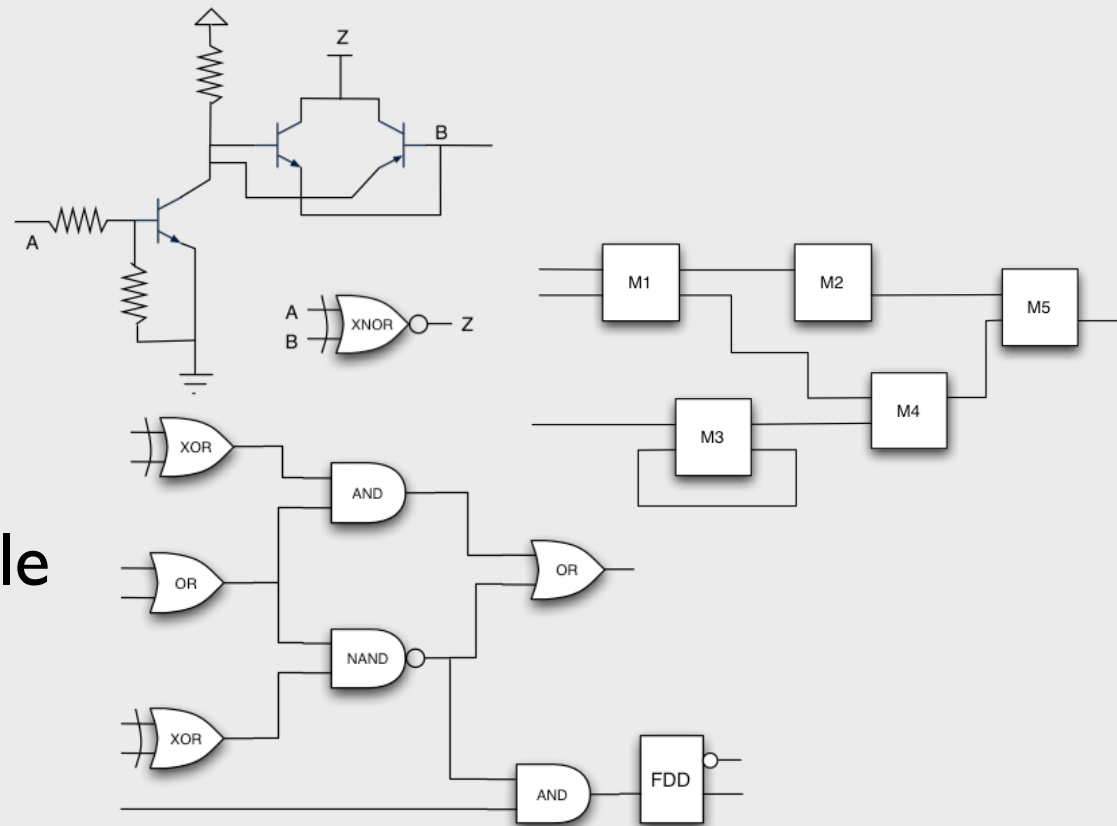


abstraction levels

- Depending on the abstraction level we look these defects at, we have different fault models

- Levels:

- Transistor
- Gate
- RTL/Module



scenario

reliability has always been an important issue for critical environments:

- design reliable components able to deal with the occurrence of faults during the operational life
- arrange reliability-aware components in more complex systems, guaranteeing that the result system will be reliable itself

goals

1. provide solutions to:


- detect faults on-line
- mask / tolerate faults
- mitigate / repair / recover from faults

2. support for reliability-related analysis:

- fault models characterization
- qualitative estimation
- probabilistic fault coverage evaluation

strategy

- a wide spectrum of fault-detection/
tolerance mechanisms available
 - from transistor to system level
 - platform tailored or general
 - hardware or software

 choosing the most appropriate approach is becoming an issue

our contribution

- embedded systems design with focus on the dependability & reliability issues
 - fault/error modeling
 - studying how a device will react to a fault
 - enabling the device to autonomously detect and deal with the fault
 - test & diagnosis
 - providing qualitative fault coverage

reliability issues

- fault modeling, fault-error analysis
- design techniques to provide fault detection and/or tolerance properties
- hw/sw co-design of dependable systems
- SRAM-based FPGAs with SEU mitigation
- fault injection strategies
- test & diagnosis
- architectural fault/error/coverage analysis

fault-error analysis

- fault models
(stuck-at, stuck-on/open, SETs)
- analysis of how the fault manifests itself
(error)
- fault classification
- fault/error at different levels of abstraction
(behavioral, RTL, gate, switch)

fault-error analysis²

- identification of the available instruments:
 - controllability
 - observability
- basic properties analysis:
 - fault redundancy
 - self-testing / fault secure properties
 - quality of the fault-error relation

fault-error analysis³

goals:

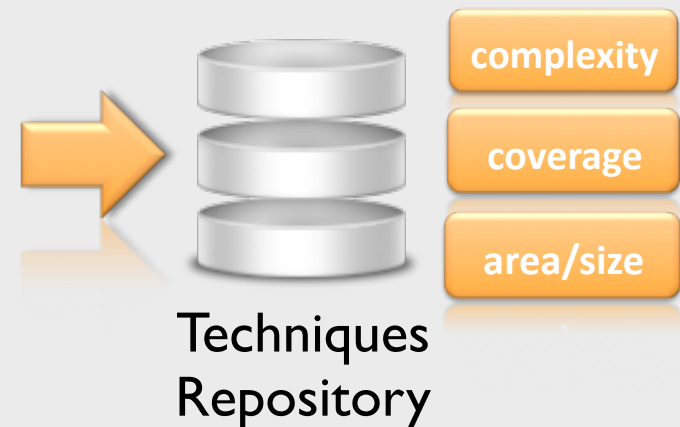
- better understanding of the dynamics under fault occurrence
- improvement of the design from the testability point of view
- post-production testing support for complex systems for diagnosis and localization

design techniques

- definition of design techniques to provide autonomous fault detection / tolerance properties
 - information redundancy
[ED/C codes & related synthesis constraints]
 - hardware redundancy
[replication and comparison/voting]
 - time redundancy
[software techniques]
- or a mix of the above ...

design techniques²

- defined and applied at different levels of abstraction
- constraints identification
- fault coverage estimation (for low level models)
- area & performance cost evaluation



design techniques³

two main issues:

- automatic application is seldom supported
[I hot encoding, TMR from Xilinx]
- design space exploration is necessary to identify the most convenient trade-offs

hw/sw co-design

- proposed for coping with the complexity of embedded systems' design
 - exploring different alternative solutions
 - supporting unified design specification
 - facilitating component reuse
- several metrics and objective functions are taken into account (area, performance, power)

hw/sw co-design²

in the past, reliability-related issues:

- taken into account at the end of the design space exploration
 - techniques and approaches are generally applied at a fixed level of granularity
- strong limitation when considering modern systems (very complex and heterogeneous)

hw/sw co-design³

goal:

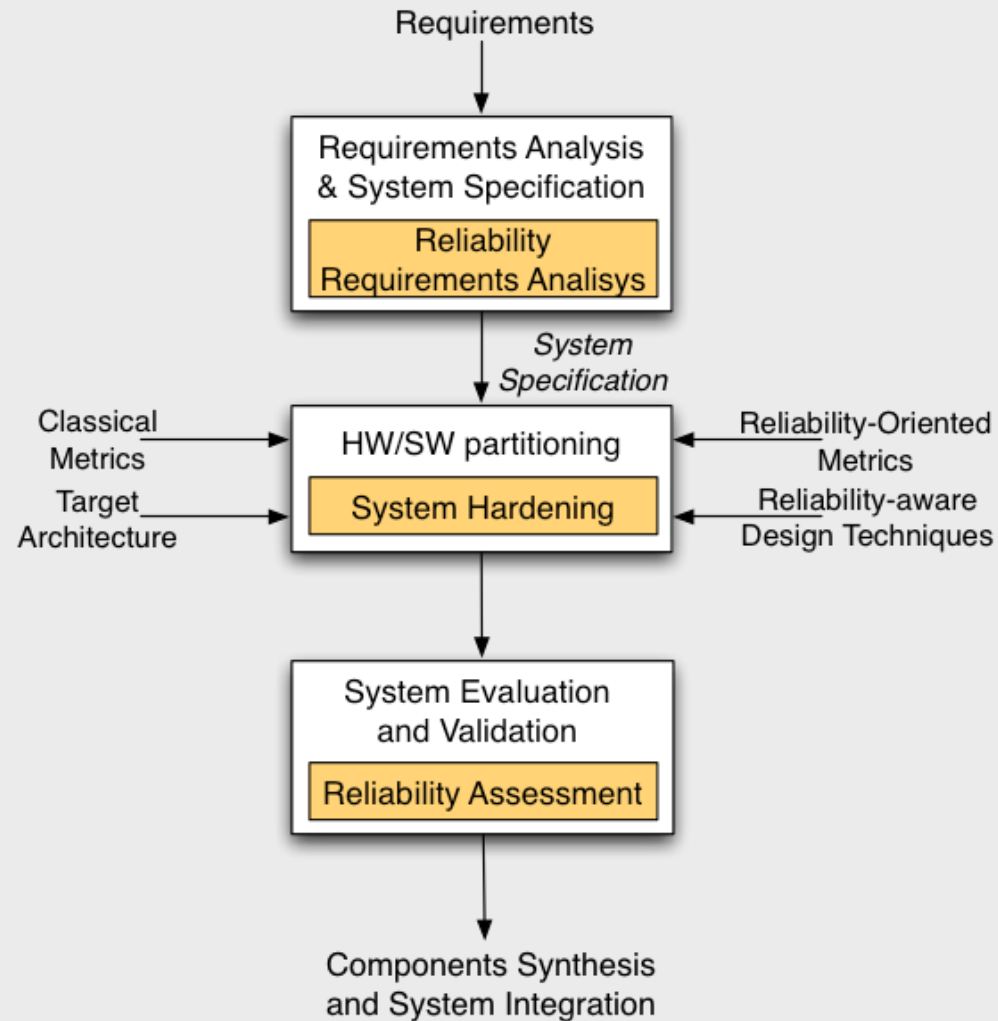
- extend the HW/SW co-design to take into account reliability issues
- a flexible and versatile methodology for reliable embedded systems' design



main issues:

- reliability requirements analysis/specification
- reliable solution design space exploration
- evaluation of the solution with respect to classical metrics and reliability oriented ones

hw/sw co-design⁴



hw/sw co-design ⁵

requirements' analysis/specification:

- functional, non-functional and reliability-related ones
- critical sections' identification
 - eventually using automatic tools (fault simulation and fault tree analysis)

hw/sw co-design⁶

hw/sw partitioning:

- enhanced with reliable techniques' application:
 - different techniques,
 - different granularity levels,
- evaluated by using classical metrics and reliability oriented ones
- multi-objective exploration for balancing classical goals with reliability-oriented ones

hw/sw co-design ⁷

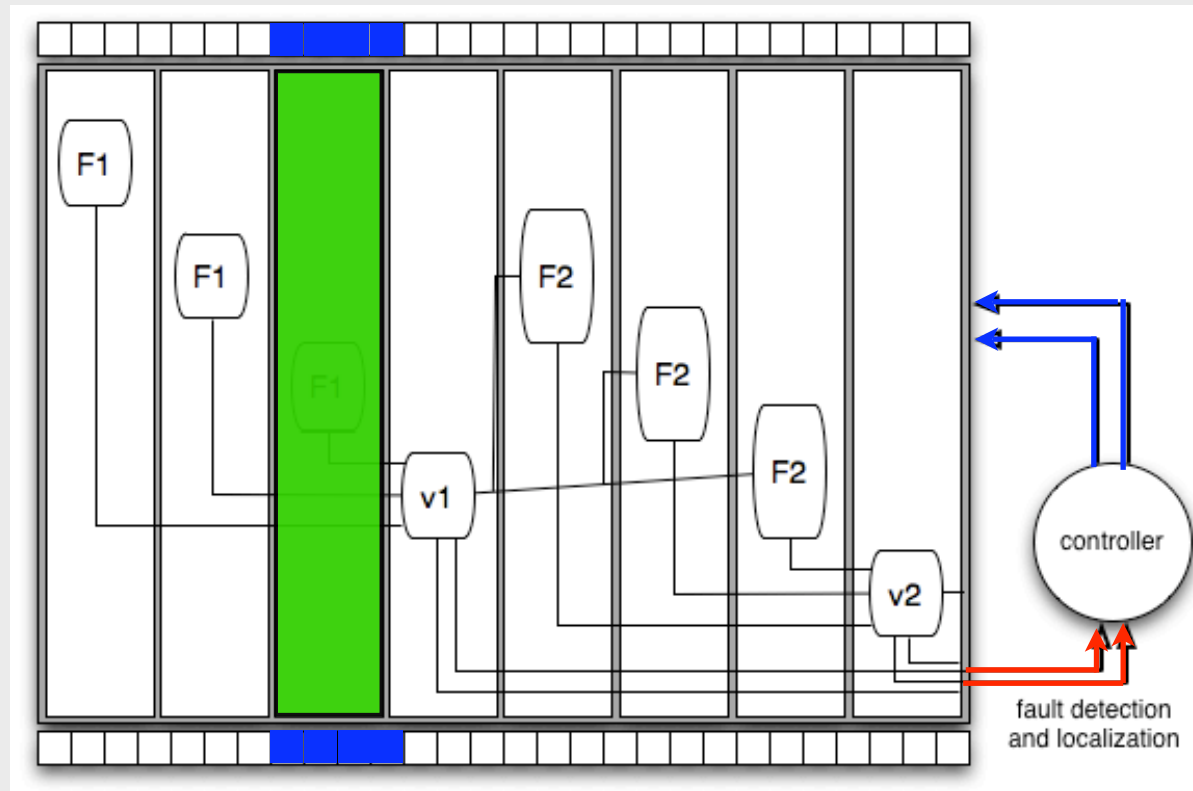
reliability evaluation/assessment:

- by design, for some techniques
- by simulation and fault injection, for complex systems

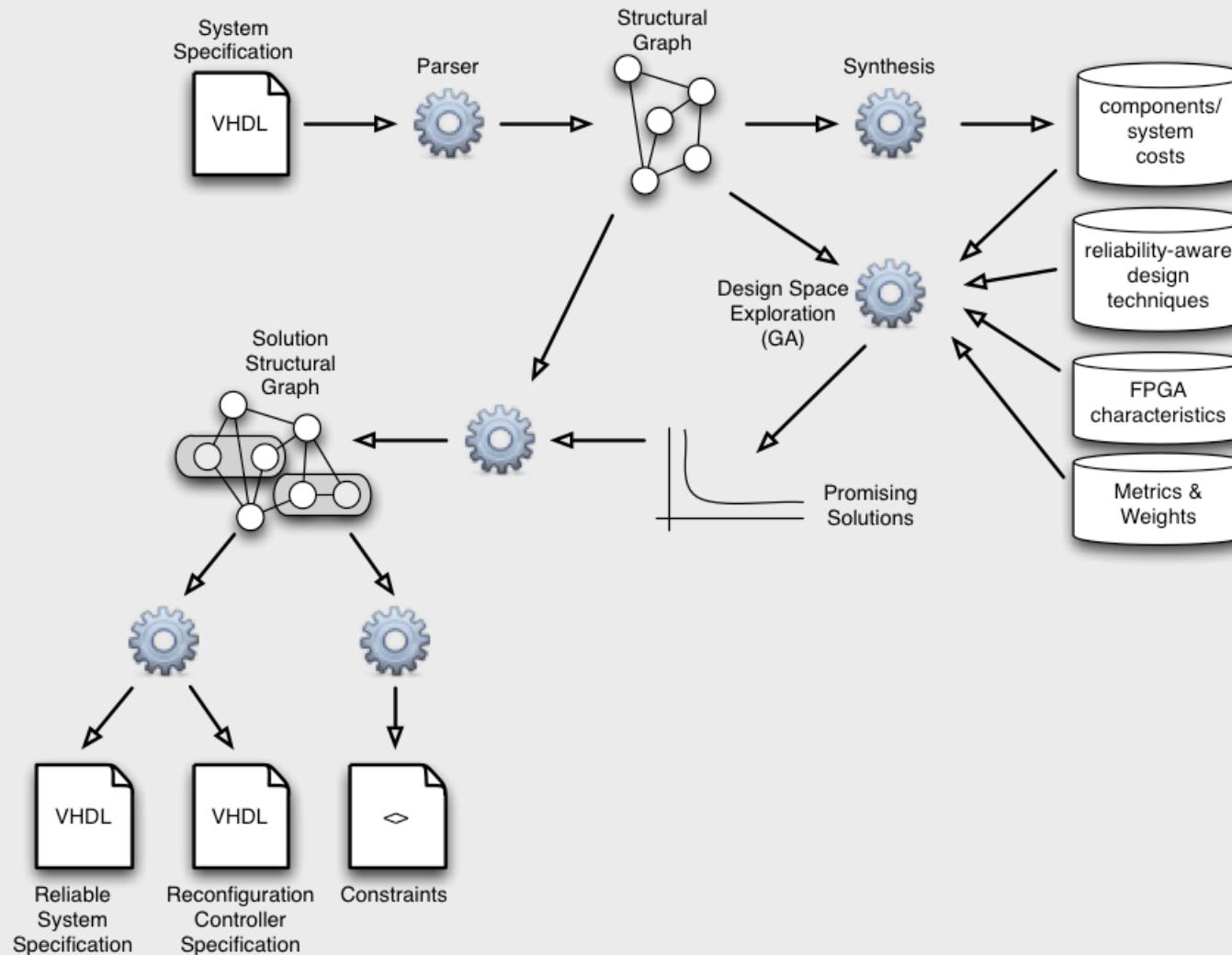
SEU faults in FPGAs

- design techniques to provide SEU mitigation properties in systems implemented on FPGA
 - fault detection & localization
 - partial, dynamic reconfiguration
- design space exploration to reduce costs and reconfiguration time

SEU faults in FPGAs²



SEU faults in FPGAs³



Permanent faults in FPGAs

- Fault detection/tolerance techniques to identify and localize the faulty portion, either
 - Totally Self-Checking - continuous monitor
 - Partially Self-Checking - periodic monitor
- Dynamic reconfiguration to avoid the corrupted logic (either partial or complete)

architectural analysis

- fault error analysis with respect to specific architectures
(microprocessors, functional modules)
- design techniques
 - software-level
 - hardware-level

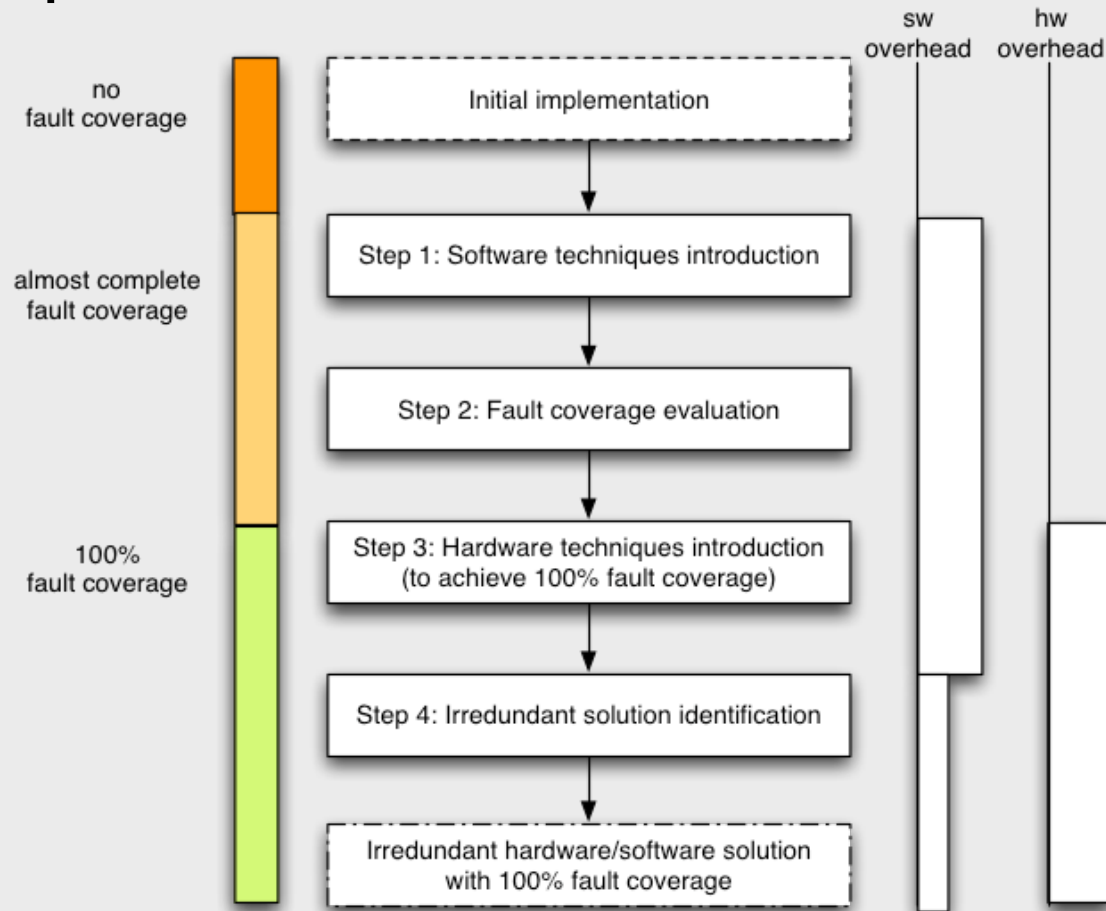
architectural analysis ²

- LEON-II microprocessor core & software techniques (duplicate & compare)
- random fault injection campaign: 100% coverage
- *specific, theory-driven* fault injection campaign: undetected faults

program	version	pipeline stage					total	exec time	data size	code size	exec time ratio	data size ratio	code size ratio	memory ratio
		FE	DE	EX	ME	WR								
FIR	plain	190	142	67	39	11	449	6235	200	860	1.00	1.00	1.00	1.00
	sw	3	0	0	0	0	3	29293	400	2864	4.70	2.00	3.33	3,08
	sw+hw	0	0	0	0	0	0	29293	400	2864	4.70	2.00	3.33	3,08
ELPF	plain	210	110	69	36	41	466	9854	164	724	1.00	1.00	1.00	1.00
	sw	3	0	0	0	0	3	110955	328	4408	11.26	2.00	6.09	5.33
	sw+hw	0	0	0	0	0	0	110955	328	4408	11.26	2.00	6.09	5.33
KLMN	plain	220	157	99	64	16	556	32811	6472	1216	1.00	1.00	1.00	1.00
	sw	2	0	0	0	0	2	258445	12944	6184	7.88	2.00	5.09	2.49
	sw+hw	0	0	0	0	0	0	258445	12944	6184	7.88	2.00	5.09	2.49

architectural analysis³

- processor customization to achieve 100%



fault injection

- structural fault injection in FPGAs to analyze the effects of SEUs
- functional fault injection in SystemC descriptions, simulated with a Reflective Simulation Platform (ReSP)



functional fault injection

strategy:

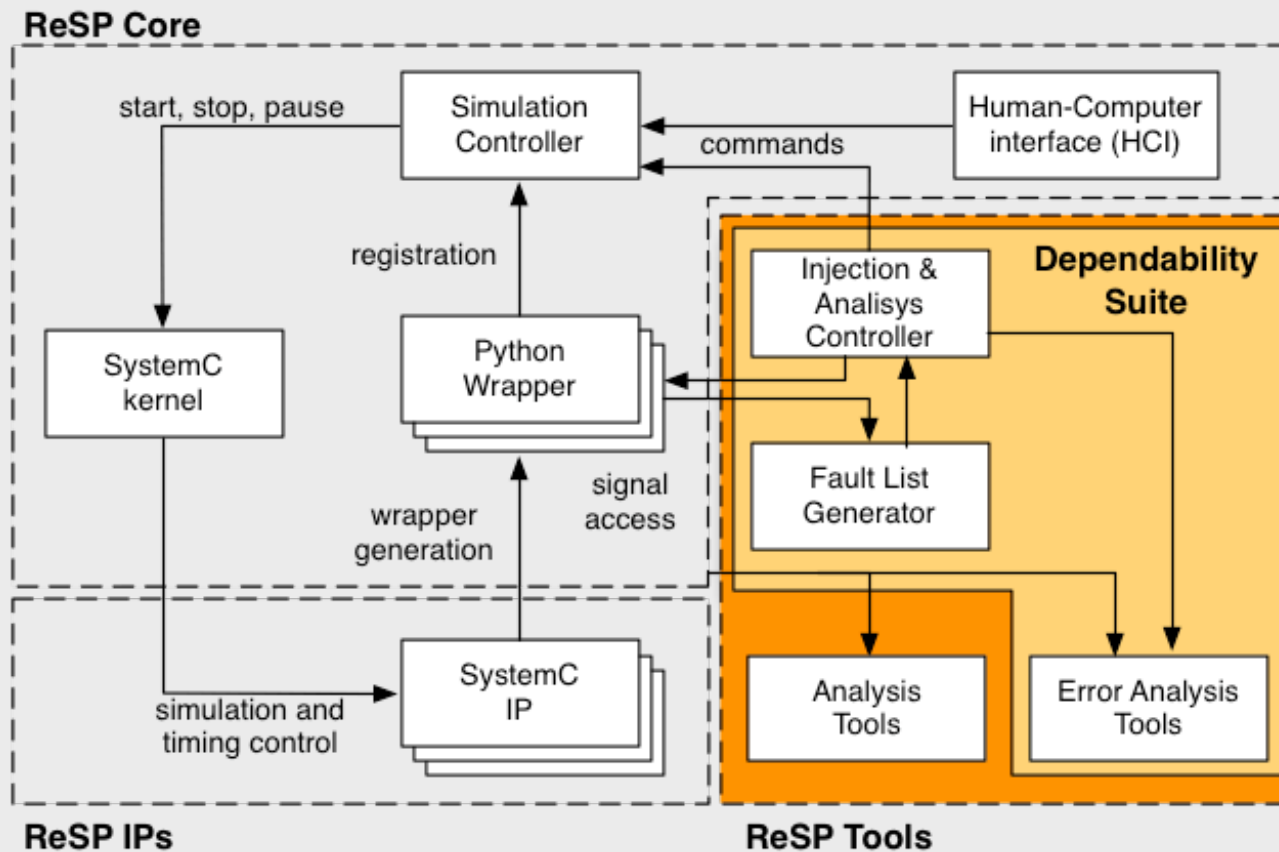
- Software-Implemented Hardware Fault Injection (SWIFI) approach
 - acts on the specification of a component functionality
 - injects faults, by modifying at run time the specification as a fault would do, and monitors the effects it produces and the system behavior

functional fault injection²

characteristics:

- transparent with respect to the nominal specification
- support for different fault models
- support for several injection strategies

functional fault injection ³



test & diagnosis

- Exploitation of test suites to perform post production debug and diagnosis
- Test suite analysis



some references ...

- Fault-error relation

C. Bolchini, F. Salice, D. Sciuto “Fault Analysis for Networks with Concurrent Error Detection Properties,” IEEE Design and Test of Computers, Vol. 15, no. 4, (1998), pp. 66-74.

<http://dx.doi.org/10.1109/54.735929>

C. Bolchini, L. Pomante, F. Salice, D. Sciuto “The Design of Reliable Devices for Mission Critical Applications,” IEEE Trans. on Instrumentation and Measurement, (TIM) Vol. 52, no. 6, (2003), pp. 1703–1712.

<http://dx.doi.org/10.1109/TIM.2003.818736>

some references ... ²

- hw/sw co-design for reliability & assessment

G. Beltrame, C. Bolchini, L. Fossati, A. Miele, D. Sciuto, “A Framework for Reliability Assessment and Enhancement in Multi-Processor Systems-On-Chip,” Proc. IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI Systems, Rome, I, 2007, pp. 132-140.
<http://dx.doi.org/10.1109/DFT.2007.6>

C. Bolchini, L. Pomante, F. Salice, D. Sciuto “Reliability Properties Assessment at System Level: A Co-design framework” Journal of Electronic Testing - Theory and Application, Kluwer Academic Publishers, Vol. 18, no. 3, (2002), pp. 351–356.
<http://dx.doi.org/10.1023/A:1015047524985>

G. Beltrame, C. Bolchini, L. Fossati, A. Miele, D. Sciuto, “ReSP: A Non-Intrusive Transaction-Level Reflective MPSoC Simulation Platform for Design Space Exploration,” Proc. IEEE 13th Asia and South Pacific Design Automation Conference (ASP-DAC), Seoul, Korea, 2008, pp. 673-678.

some references ... ³

- **Architectural reliability analysis**

C. Bolchini, A. Miele, M. Rebaudengo, F. Salice, D. Sciuto, L. Sterpone, M. Violante, “Software and Hardware Techniques for SEU Detection in IP Processors,” *Journal of Electronic Testing: Theory and Applications*, Springer Vol. 24, no. 1-3, (2008), pp. 35–44.
<http://dx.doi.org/10.1007/s10836-007-5028-0>

- **SEU fault mitigation in SRAM-based FPGAs**

C. Bolchini, A. Miele, M. D. Santambrogio, “TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs,” *Proc. IEEE Intl. Symp. on Defect and Fault Tolerance in VLSI Systems (DFT)*, Rome, I, 2007, pp. 87-95.
<http://dx.doi.org/10.1109/DFT.2007.63>

C. Bolchini, D. Quarta, M. D. Santambrogio, “SEU Mitigation for SRAM-Based FPGAs through Dynamic Partial Reconfiguration,” *Proc. ACM Great Lakes Symposium on VLSI (GLSVLSI)*, Stresa, I, 2007, pp. 55-60.
<http://doi.acm.org/10.1145/1228784.1228803>

C. Bolchini, F. Salice, M. D. Santambrogio, “Exploring Partial Reconfiguration for Mitigating SEU faults in SRAM-Based FPGAs,” *Proc. Intl. Conf. of Engineering of Reconfigurable Systems and Algorithms (ERSA)*, Las Vegas, USA, 2007, CSREA Press, pp. 199-202.

people involved

Faculty:

- cristiana bolchini
- fabio salice
- donatella sciuto

PhD & Post Doc:

- luca fossati
- laura frigerio
- antonio miele
- marco santambrogio