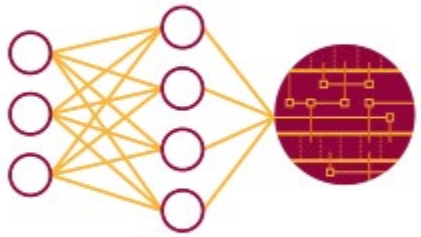


Advances in System-Level Test

Paolo BERNARDI
Politecnico di Torino

History of testing

- Since the inception of IC design in the mid-1960s, IC test has been an integral part of the manufacturing process.
- Initially, tests were of the **Functional** nature of either randomly generated or created from verification suites.
- But as chips got larger, testing required a more targeted approach, one that needed to be easily replicated from one design to another.
- This led to the invention of **Structural** methods like scan, which made designs combinational and simplified the test generation process.

Structural vs Functional

- Structural methods based on Scan almost completely replaced Functional methods
  - **Structural methods are easier to automate** as test patterns are often generated by automatic tools
  - «Ancient» **Functional test metrics** that enabled a kind of «black box» testing were based on circuit specifications and they were **not enough to satisfy the production process quality** with large designs
  - Automatic Test Equipment **(ATE) architectures are simpler** to construct **if structural methods** are used.

Invention of Scan and Logic BISTs

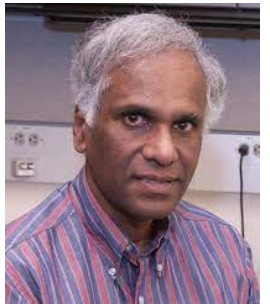
- Scan invention is attributed to dr. *Thomas Williams*, Synopsys
- **1989 W. Wallace McDowell Award**
  "For developing the level-sensitive scan technique of testing solid-state logic circuits and leading, defining, and promoting design for testability concepts."
- Logic BIST's first implementation is accredited to prof. *Bernd Könemann*, at Aachen University (LogicVision afterwards)
- **1975 "Built-In Logic Block Observation Technique."**
- **1983 -** Prototype of the first self-tested microprocessor was presented by prof *Joachim Mucha*, Hannover University

But scan methods are not free of issues

- Need of **additional hardware** modules, potentially impacting silicon area and device performance
- May be **inaccurate for delay fault** models unless very expensive testing equipment is used
- Potentially inducing **over-test**, in particular for delay testing
- Introducing many **power and thermal concerns**.
- Therefore, Functional testing was not fully forgotten and continued to play a (minor) role.

Software Testing methods did not extinguish

- With the advent of the processor, new and more structured Functional approaches were invented.
- The first solution was provided during the early ’80s by prof Jacob Abraham, Austin Texas University
- **1980: Test Generation for Microprocessors, IEEE Transactions on Computers**



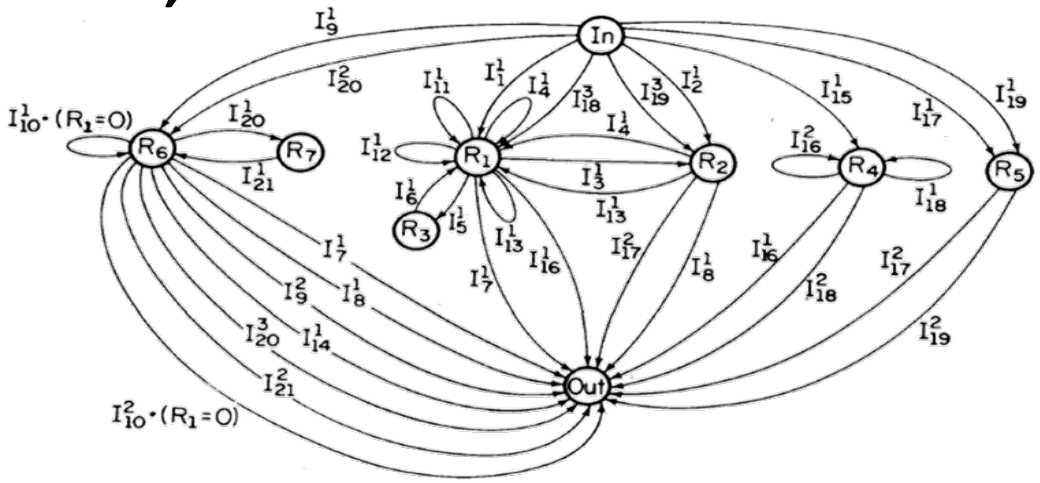


Fig. 1. *S*-graph of the example microprocessor.

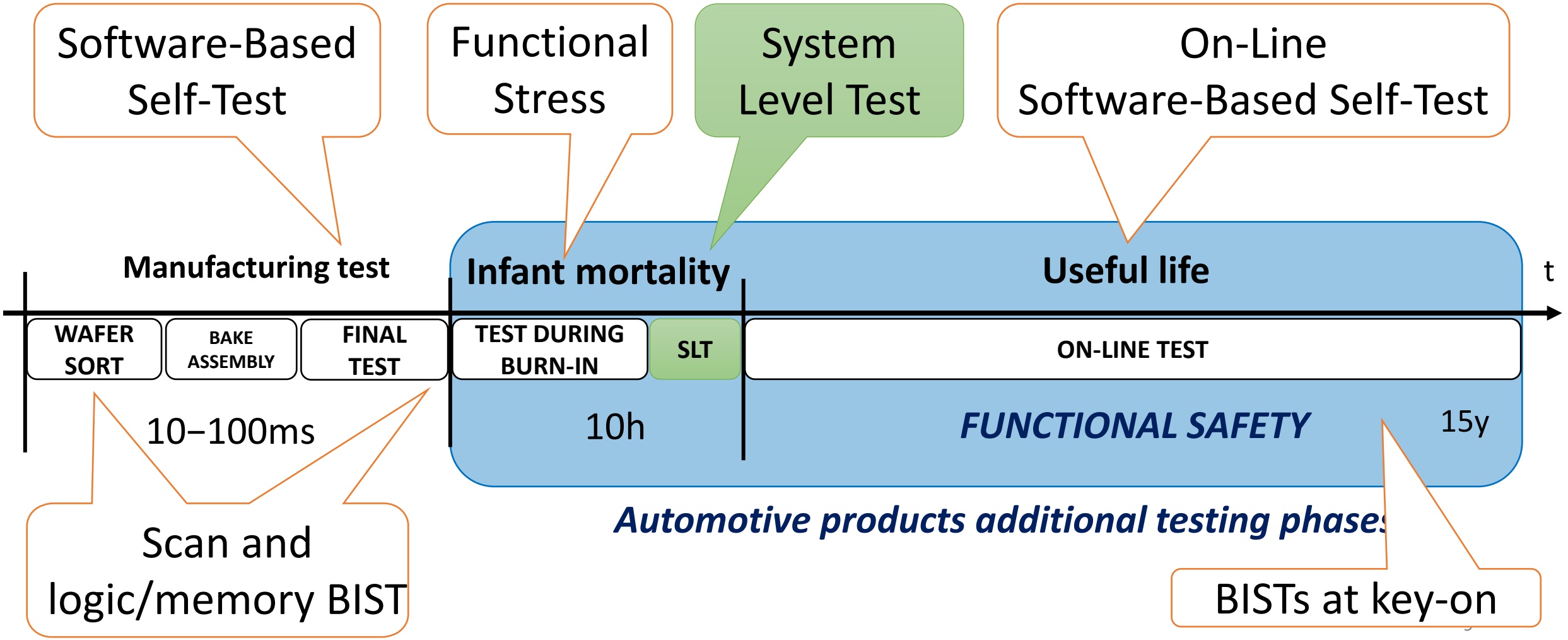
What about today?

- After almost 50 years, the testing scenario just slightly evolved, following technology trends currently led by the complexity of the circuits under test and the field of use (i.e., Automotive)
- Structural methods are still dominant, at least during the manufacturing test process
- Functional techniques are recognized to be
  - Useful to complement structural techniques during the manufacturing test process
  - Able to mitigate thermal issues that may originate during stress phases like along Burn-In
  - Very helpful along with the useful life of the components in the mission field.

Is all this test/stress activity enough?

- It is a very recent question about the effectiveness of the testing flow
- The major source for this doubt is the elevated number of field return.

A possible modern testing flow



Motivations from literature

- S. Biswas and B. Cory, "An Industrial Study of System-Level Test," in *IEEE Design & Test of Computers*, vol. 29, no. 1, pp. 19-27, Feb. 2012, doi: 10.1109/MDT.2011.2178387.
- H. H. Chen, "Beyond structural test, the rising need for system-level test," *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, 2018, pp. 1-4, doi: 10.1109/VLSI-DAT.2018.8373238.
- S. Letchumanan, T. H. H. Tan, Y. P. Gan and S. L. Wong, "Adaptive test method on production system-level testing (SLT) to optimize test cost, resources and defect parts per million (DPPM)," *2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT)*, Hsinchu, Taiwan, 2018, pp. 1-3, doi: 10.1109/VLSI-DAT.2018.8373239.

Is System-Level Test a solution?

- What exactly is SLT?
  1. Test of a whole system (e.g., a smartphone or an automotive electronic control unit), focusing on interactions between its components: ICs, sensors, mechanical parts, etc.
  2. Incoming quality control of ICs by a system integrator, to sort out defective ICs and to uncover systematic quality problems of a supplier. The ICs under test are put on a board that imitates the full-system setup and applies to the IC a workload that mimics real-life operation.
  3. Outgoing quality control by the IC manufacturer to prevent defective ICs from delivery and to reinforce its own quality control. The procedure is similar to 2), except that the IC manufacturer has less knowledge about the full-system setup but more knowledge about the manufactured IC.

[I. Polian et al., "Exploring the Mysteries of System-Level Test," 2020 IEEE 29th Asian Test Symposium (ATS), Penang, Malaysia, 2020, pp. 1-6, doi: 10.1109/ATS49688.2020.9301557.]

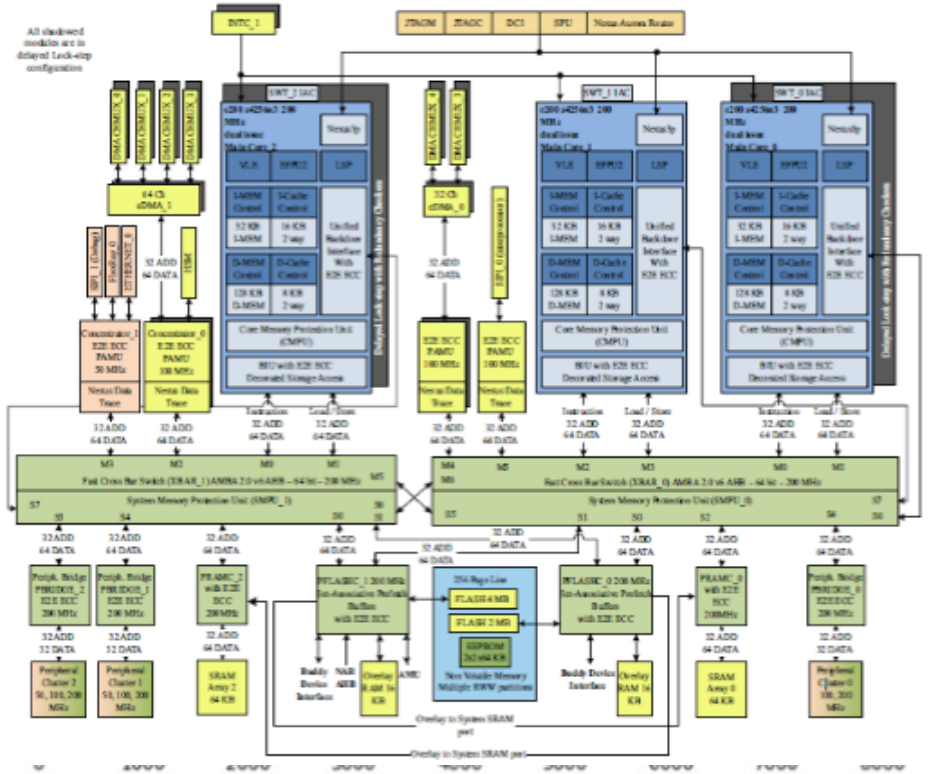
More details about SLT

- It usually consists in applying to the device under test (usually, a complex system-on-chip IC) workloads that originate from its intended usage.
  - A popular SLT example is booting an operating system and running several software applications known to stress the system; if the system does not behave as expected, SLT has found a failure.

An example

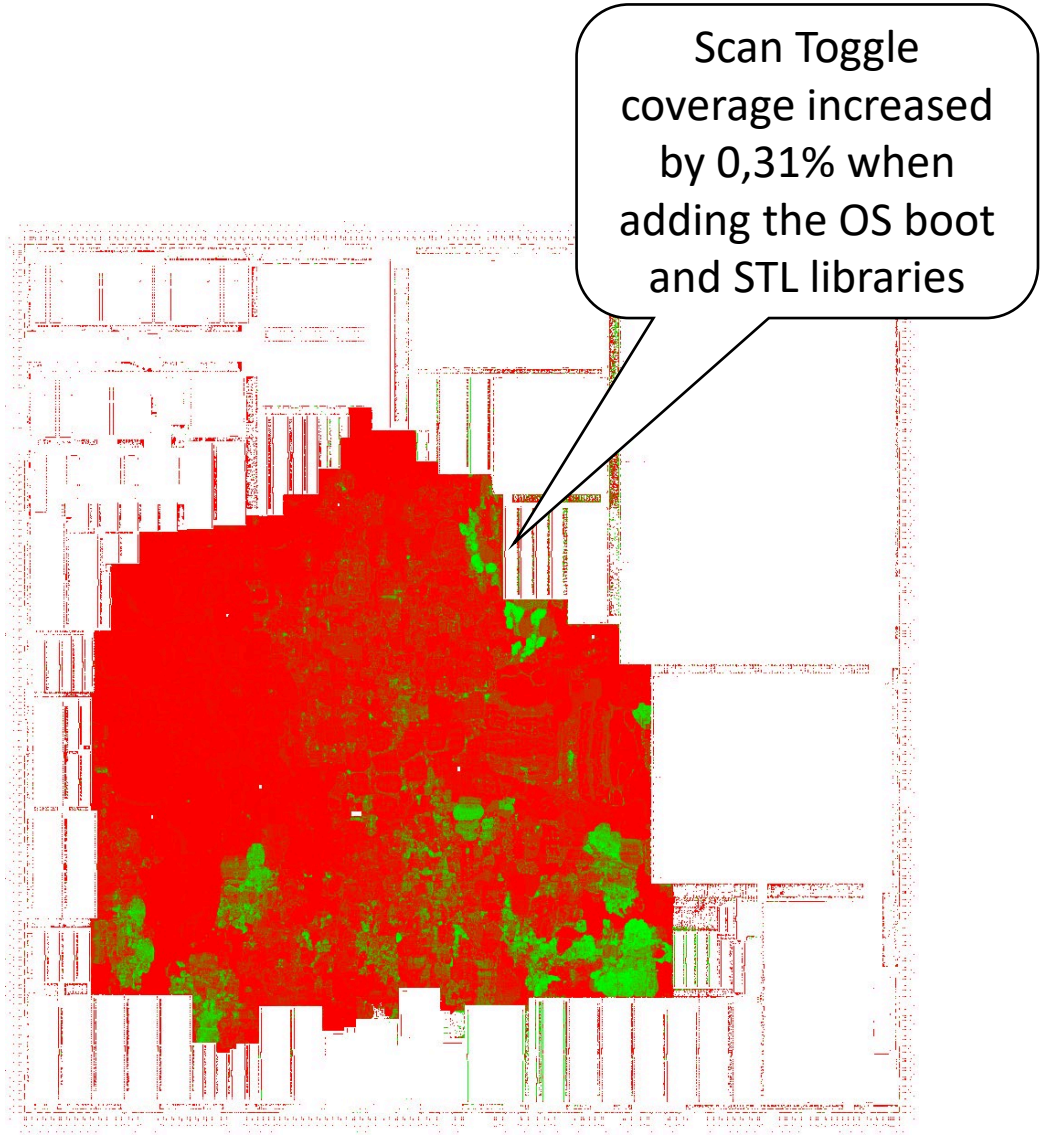
Experimental results reported later are computer on a SPC58 family chip

- 40 nm Automotive SoC manufactured by STMicroelectronics
- About 20 million logic gates
- About 700k flip-flops.
- 6 Mbyte of Flash memory
- 128 Kbytes of general-purpose SRAM



An example (II)

- Integration testing built over RTOS Micrium
  - About 70 key-on and 34 runtime SBST functional procedures launched as tasks of the OS with different priority
  - ECC testing procedures
  - MBIST parallel launch
  - Randomized Timer programmed for triggering interruptions at variable intervals
  - Adding some benchmarks from the **EEMBC** community
    - EDN Embedded Microprocessor Benchmark Consortium



OS Boot toggle evaluation

SLT-unique fails

- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

SLT-unique fails

- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

SLT-unique fails

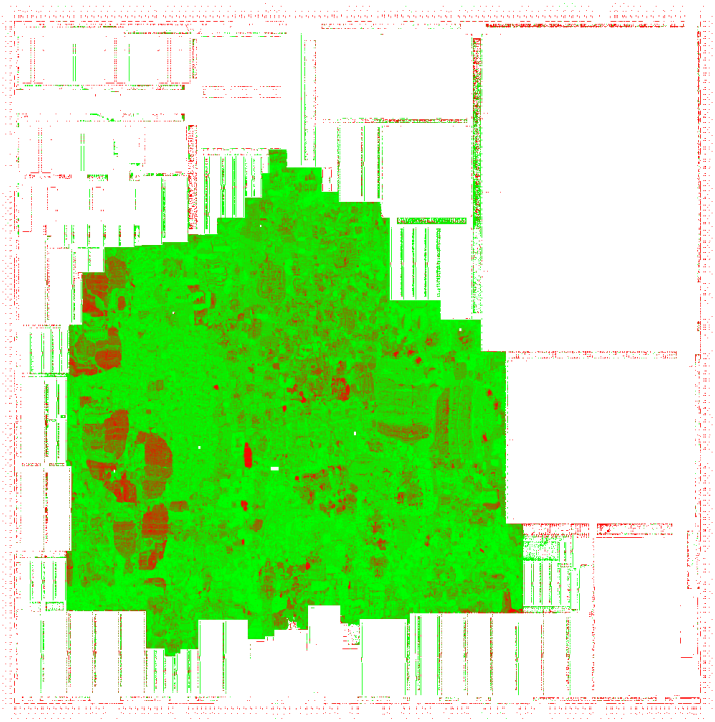
- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

The pattern set is too wide

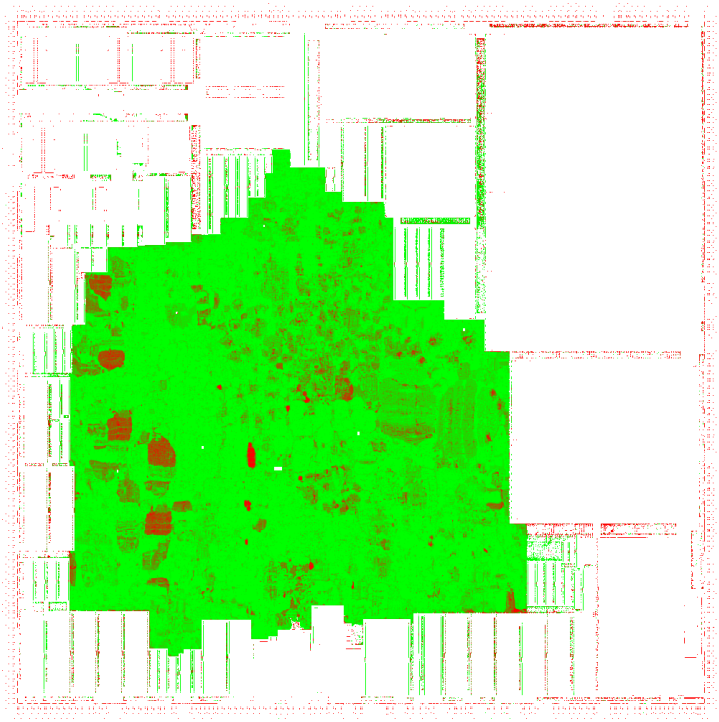
- When the circuit is extensively large, the pattern set may be too wide to be applied or memorized on ATE, or the ATPG cannot reach the full coverage of the circuit
- Therefore, a significant number of faults could be left uncovered.

Fault model	*# Faults*	*Coverage [%]*	*# Patterns*	*# Test Escapes*
Transition Delay	39723688	89.65	(ATPG) 82251	4,111,401
Stuck-at		99.21	(ATPG) 68930	313,817

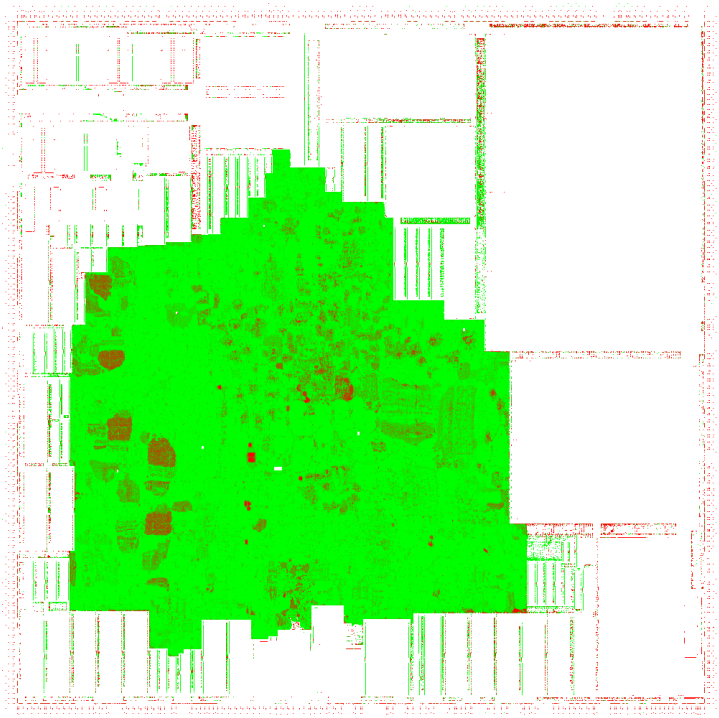
	# nodes	Coverage [%]	# Patterns
Toggles	19651246	90,73	32 ATPG
		92,18	1024 Pseudo
		94,68	+18 LBIST
		95,89	+ 12 ATPG (selective)



32 ATPG patterns through a single chain

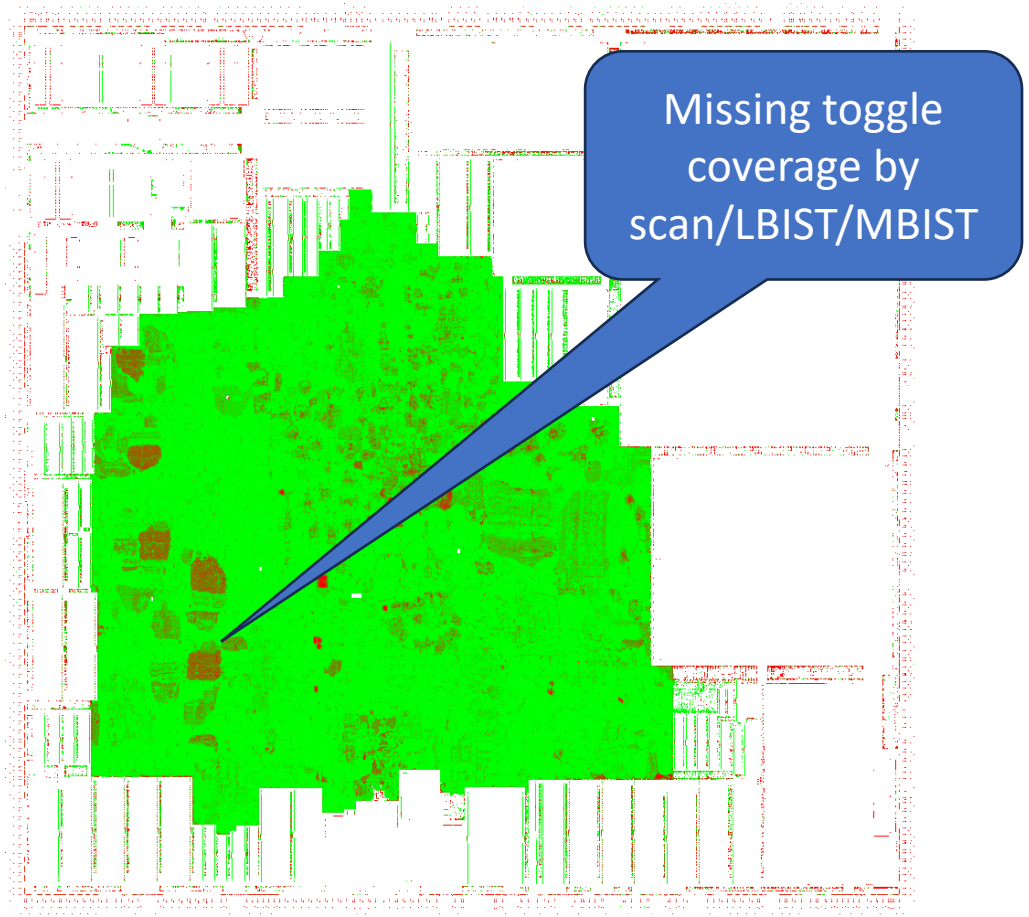


1024 Pseudorandom patterns through a single chain



1024 Pseudorandom patterns + 18 LBIST + 12 ATPG (selective)

Coverage holes analysis



SLT-unique fails

- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

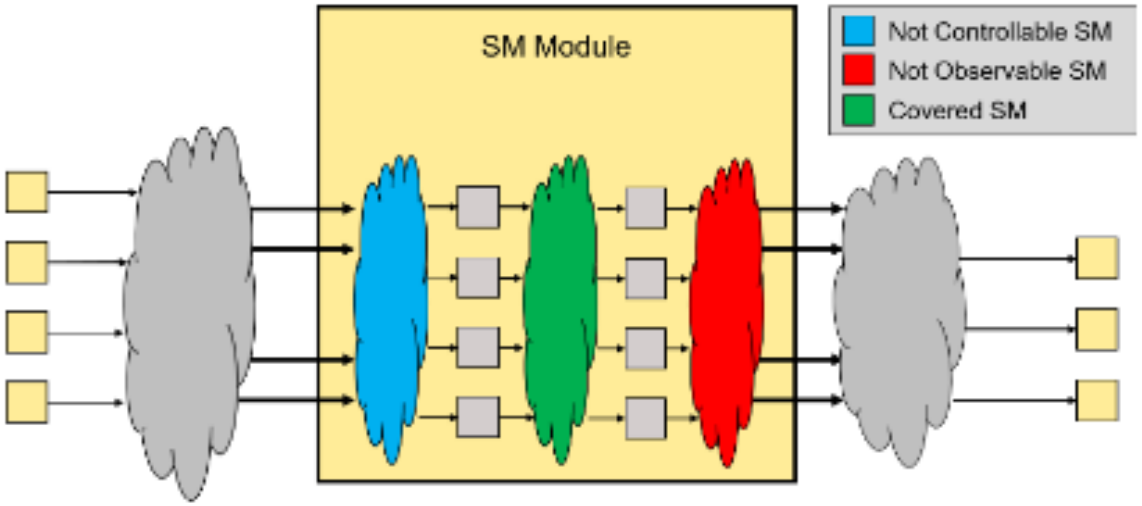
Systematic ATPG coverage holes (II)

- Systematic ATPG coverage holes may show up in different situations
  - When the circuit is extensively large and the pattern set is too wide to be applied or memorized on ATE
  - If there is a partitioning of the test resources into several areas (i.e., safety islands)

[N. Karimi, K. Chakrabarty, P. Gupta, and S. Patil, “Test generation for clock-domain crossing faults in integrated circuits,” in IEEE DATE,2012, pp. 406–411]

[Tille et al. Towards an Automated Flow for Implementation of Dedicated LBIST Scan Chains for Functional Safety, VDE TUZ 2021]

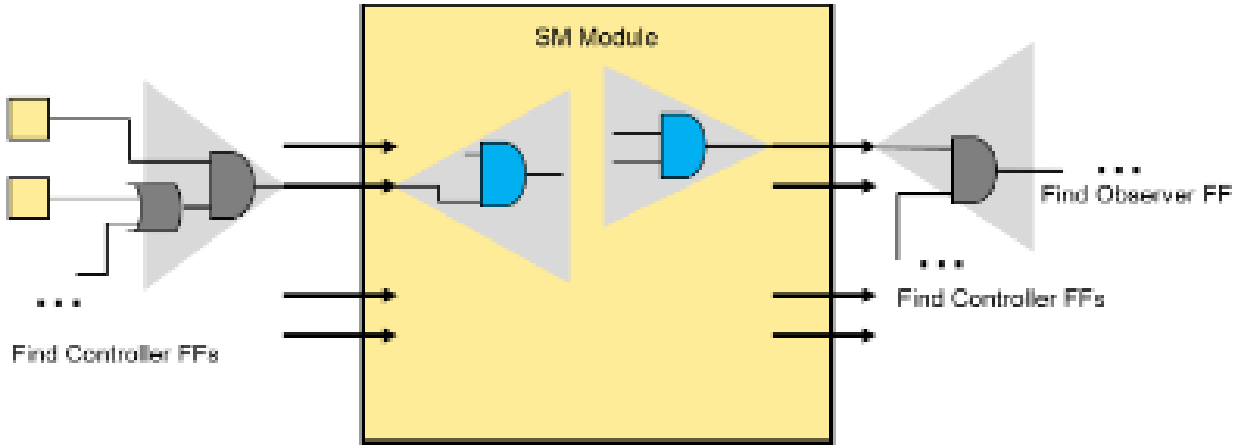
DFT Partitions



- Some safety module gates could be uncovered
- Solution is to extend the set of considered flip-flops to add in the sub-scan chain
- Alternatively, functional programs could be used as SLT.

TABLE I. ANALYSIS RESULTS

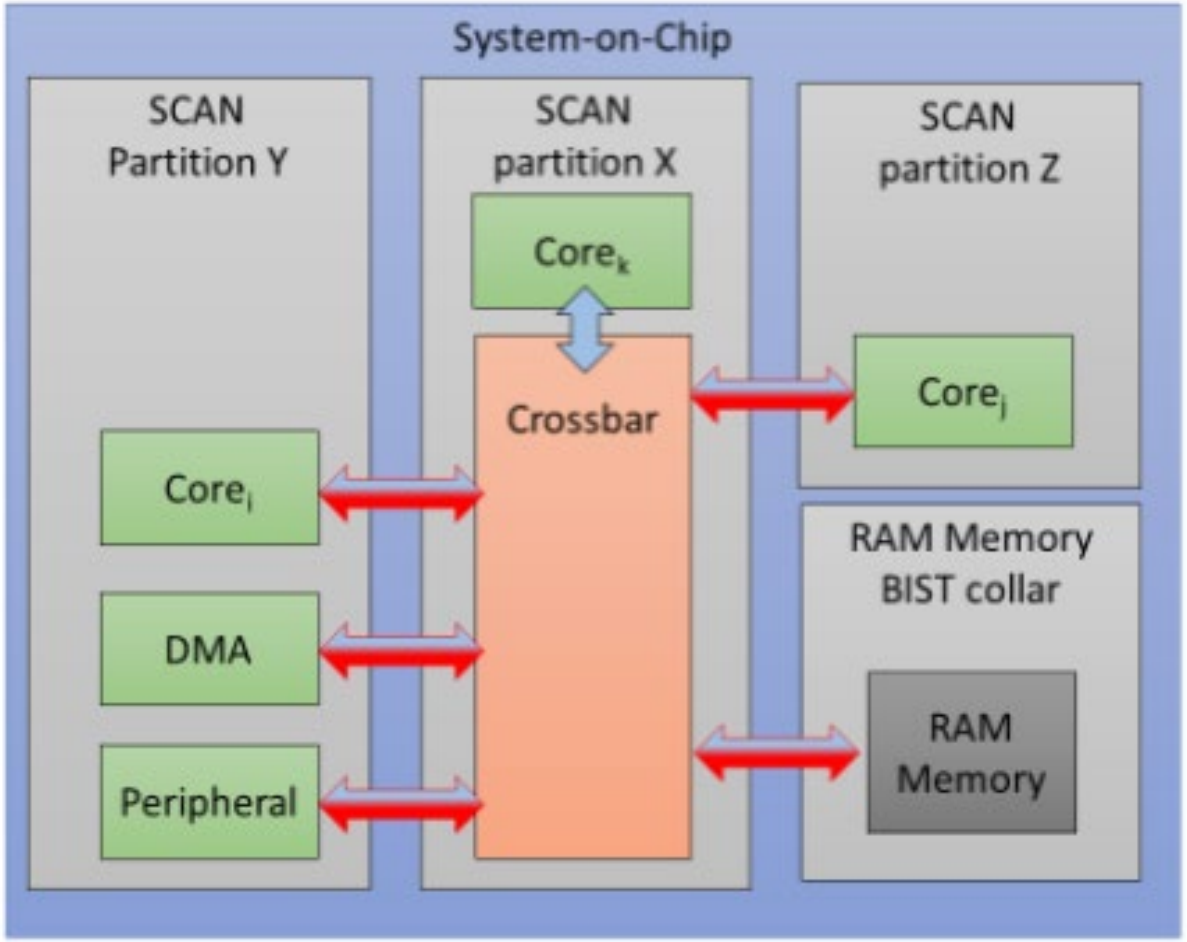
Netlist	Δ TC	SM/all FFs	CPU time
Pure SMs	-1.32%	18.87%	-
Exhaustive	-0.05%	36.50%	+30m
Abort Level (T_h=20)	-0.41%	21.63%	+10m
Abort Level (T_h=30)	-0.30%	23.08%	+15m
Pin Sel. (T_h=100)	-0.40%	27.56%	+30m
Pin Sel. (T_h=10)	-0.34%	21.77%	+35m



Example

Crossbar module is considered

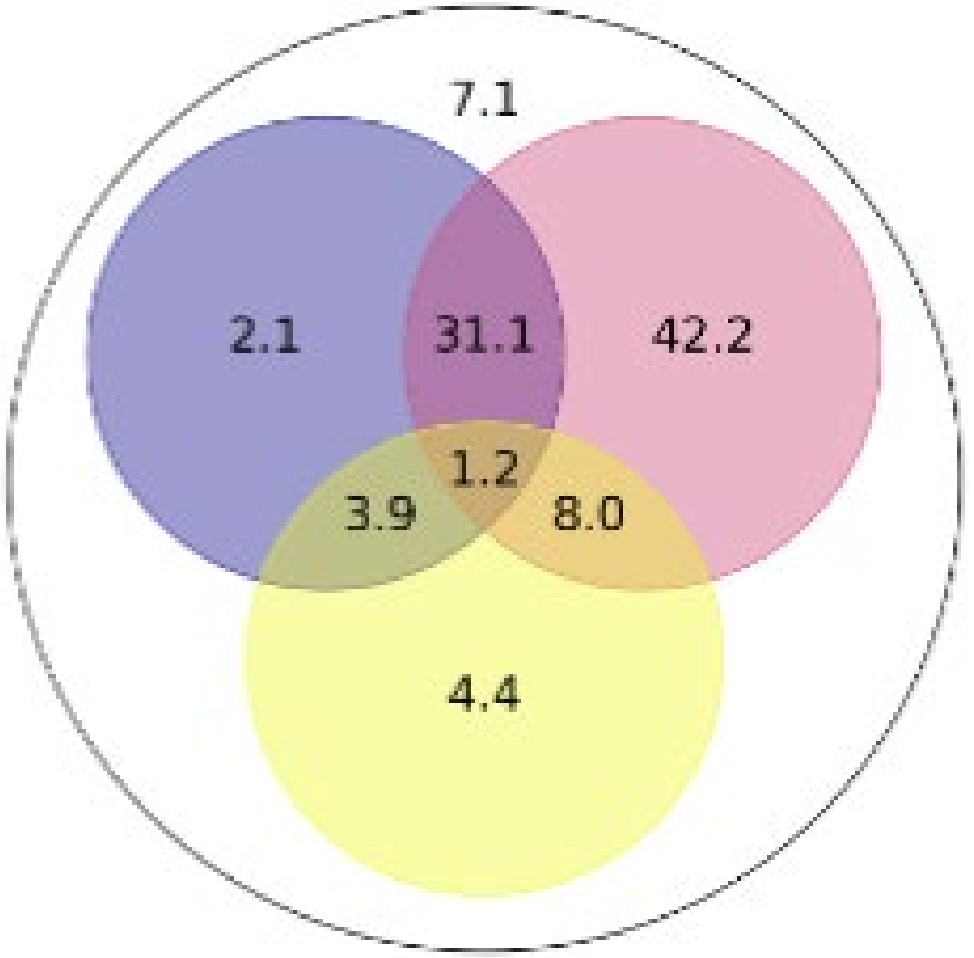
- Undetected faults from structural tests are analyzed
- 155k Transition Delay Fault (TDF)
- 87.49% TDF Coverage by about 155k scan patterns
- About 19k TDF is left uncovered
- A functional procedure is used to complement the scan patterns



Experimental result

Crossbar module is considered

- ❑ Undetected faults from structural tests are analyzed
- ❑ 155k Transition Delay Fault (TDF)
- ❑ 87.49% TDF Coverage by about 155k scan patterns
- ❑ About 19k TDF is left uncovered
- ❑ A functional procedure is used to complement the scan patterns



SLT-unique fails

- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

SLT-unique fails

- Failure mechanisms that are not covered by standard fault models
  - e.g., marginal, or “soft”, timing failures that manifest themselves only under certain operating conditions (voltage, temperature), and power-supply instabilities in conjunction with complex power-management schemes
- Systematic ATPG coverage “holes”
  - incomplete test coverage due to test time and tester memory limitations
  - uncovered faults in the logic structures at the clock domain boundaries, asynchronous or analog interfaces, or clock distribution networks
- Faults exposed only during system-level interactions
  - e.g., complex software-controlled clock- and power-domain interactions or resource contention in a multi-core system that cannot be fully replicated on an ATE
  - complex hardware-implemented protocols or “soft” failures during high-speed memory accesses.

Case of Study – CAN bus SLT

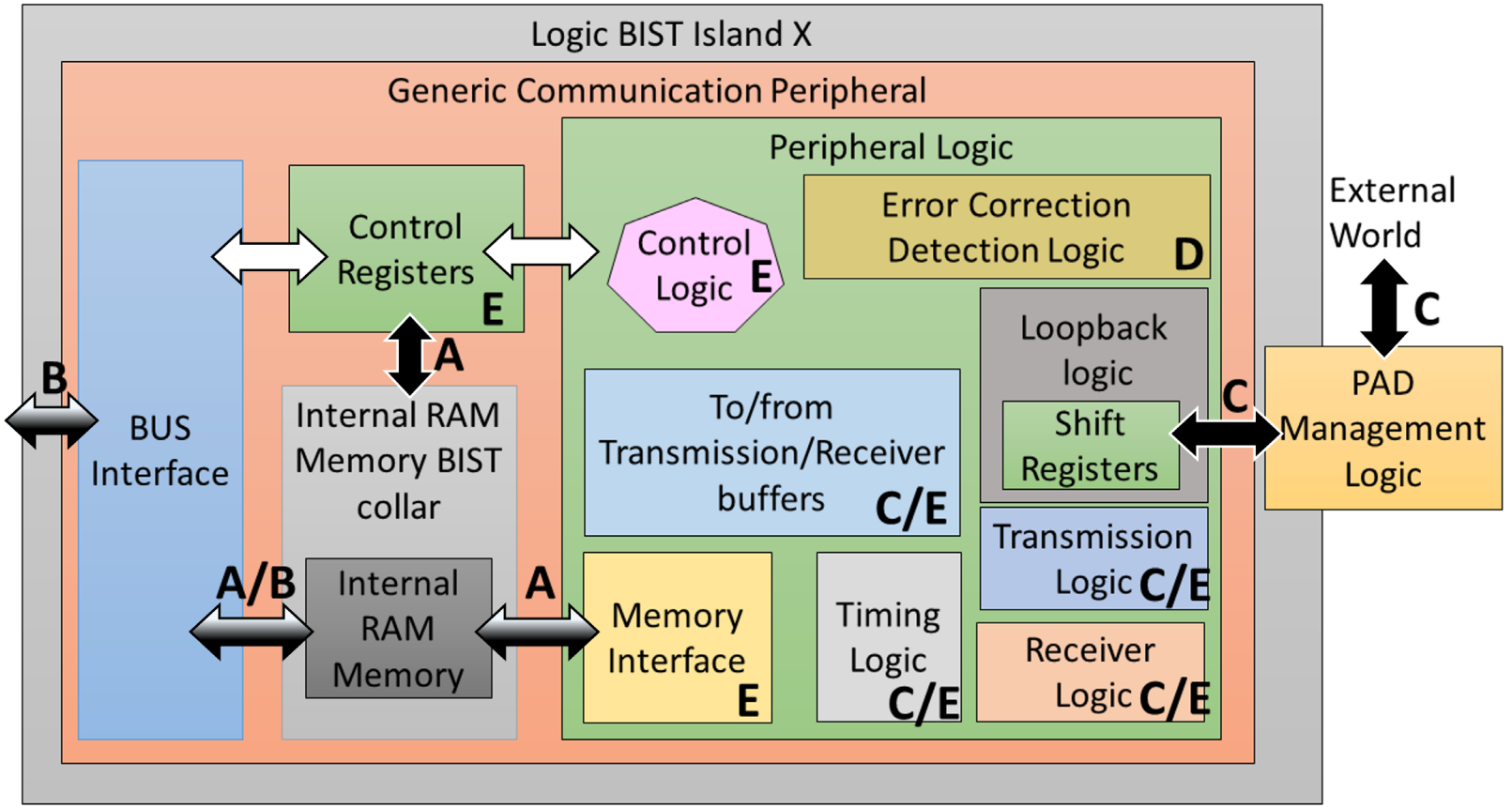
A) Embedded memory access ports

B) Interfaces to other on-chip components

C) Transmission/Reception Interfaces to Chip Top

D) Detection and correction logic circuits

E) Complex hardware-software functions



Case of Study – CAN bus SLT (II)

What are the main weak points?

A) **Embedded memory access ports**: they may not be completely covered along structural tests due to collars and memory DfT circuits like MBIST.

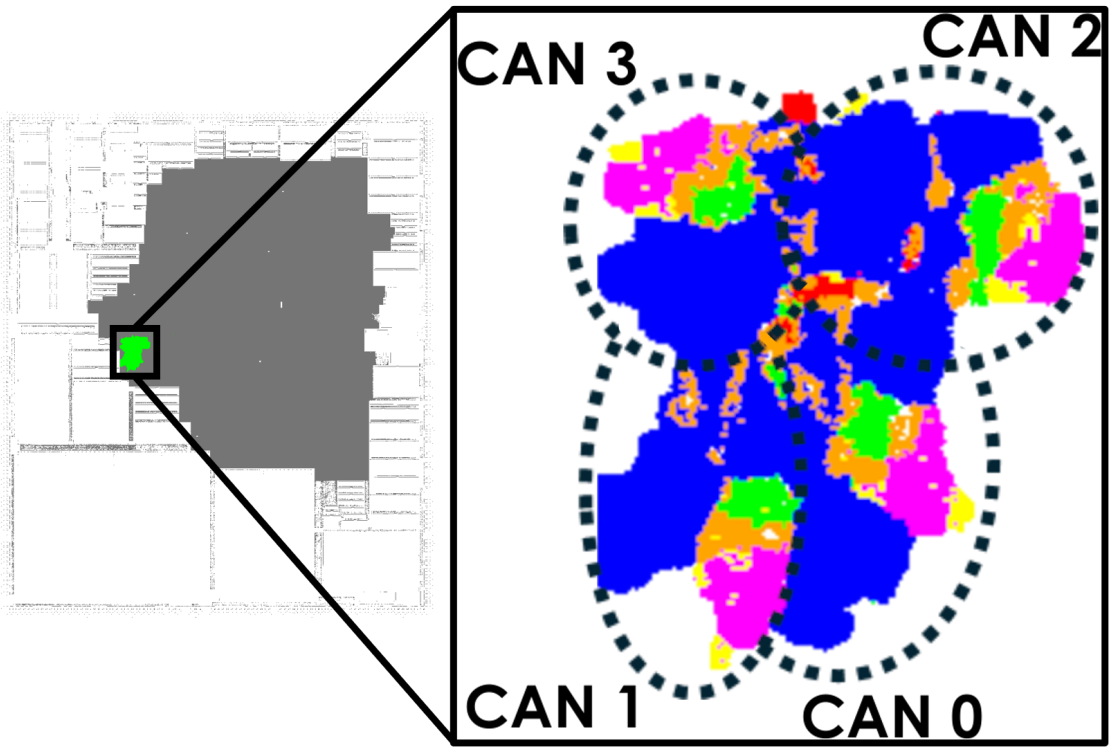
B) **Interfaces to other on-chip components**: they may be included in different LBIST, or Scan Chain islands can introduce testability issues.

C) **Transmission/Reception Interfaces to Chip Top**: Some signals and pins to and from outside the SoC may never be exercised during manufacturing tests.

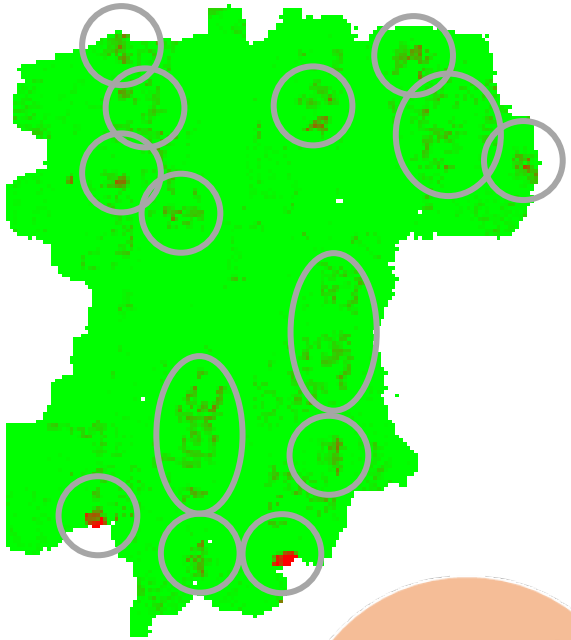
D) **Detection and correction logic circuits**: they usually include large logic functions resulting in deep circuits that are hard to target by structural tests.

E) **Complex hardware-software functions**, like complex protocol functions and synchronization mechanisms, are not exercised.

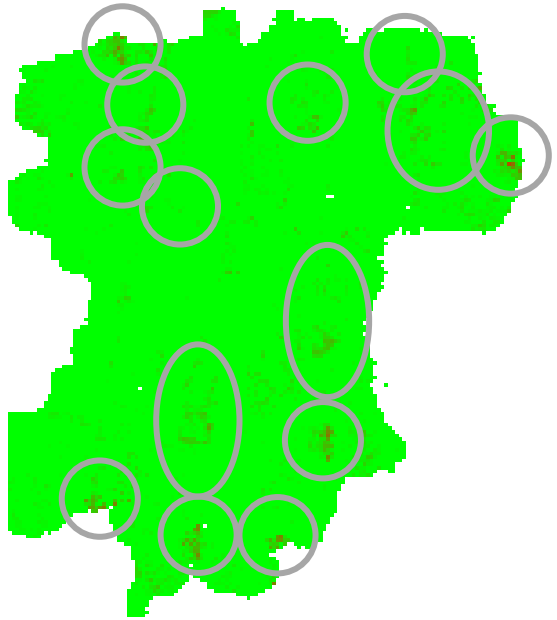
A) Embedded memory access ports
B) Interfaces to other on-chip components
C) Transmission/Reception Interfaces to Chip Top
D) Detection and correction logic circuits
E) Complex hardware-software functions

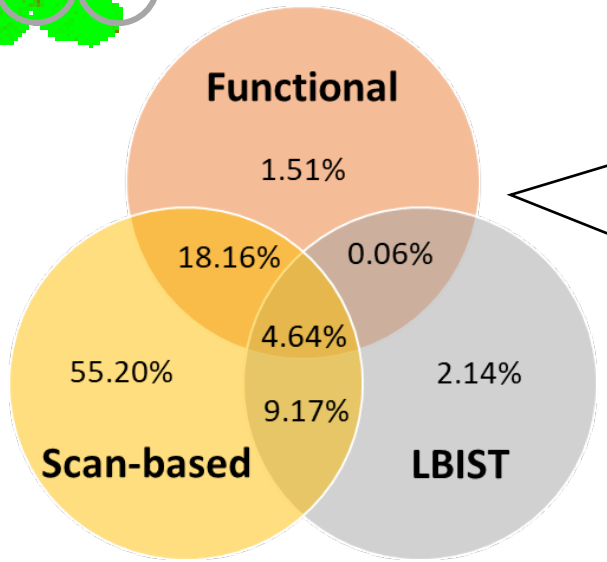


Before SLT



after SLT





35,966 TDF faults

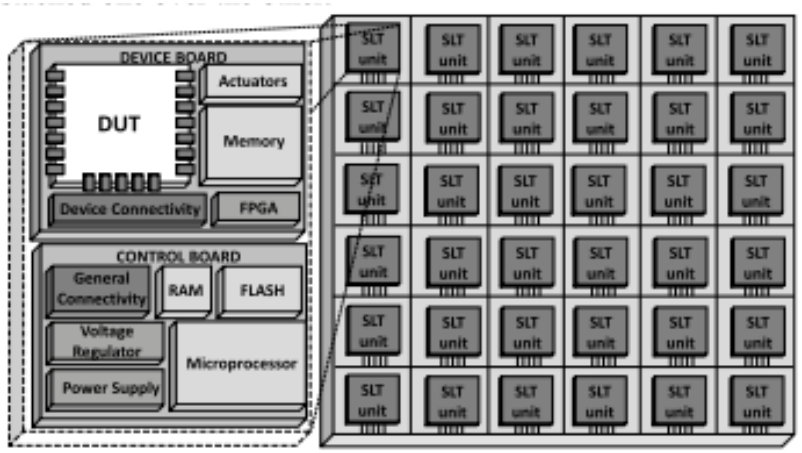
Before SLT – 89.38%
After SLT – 90.89%

More details about SLT

- It is often impossible to perform SLT on standard test equipment that does not include all the required features, but specialized SLT testers are available.
- SLT application time can be very long (several minutes)
- SLT equipment is expensive.

Making SLT less expensive

- Adaptive test
  - SLT is applied only for a subset of circuits determined during earlier test phases
- Combining SLT with Burn-In [Almeida et al., DDECS 2019]
  - Many ICs are tested in parallel
  - During SLT, the DUT must be stimulated, not only powered-up
  - Aging acceleration can be achieved via self-heating, or acting on voltage
- Re-using the setup from one product to another



The future of SLT

- SLT may disappear if
  - New fault models and ATPG/DfT solutions will be introduced, able to catch the new defect types and guarantee sufficiently low PPM
- But
  - Current SoC complexity and new semiconductor technologies variability may prevent achieving this goal
- SLT may play for ICs the role of functional test for PCBs
- Metrics are required!
- Correlating the behavior of single ICs over the whole test process is crucial to
  - Pinpoint the root causes for failures
  - Understanding the fault activation and propagation processes
  - Optimizing the test process.

Conclusions

- Functional testing methods are the most «ancient» testing approach
- Structural method has replaced almost everywhere the use of Functional method
- They are still being used for
  - Refining coverage, especially by addressing performance-related fault models
  - Provide strong electrical stress without running the risk of falling into dangerous thermal overrun effects
  - Last-minute coverage issues solving as a holistic method.

Thank you for your attention

Politecnico
di Torino