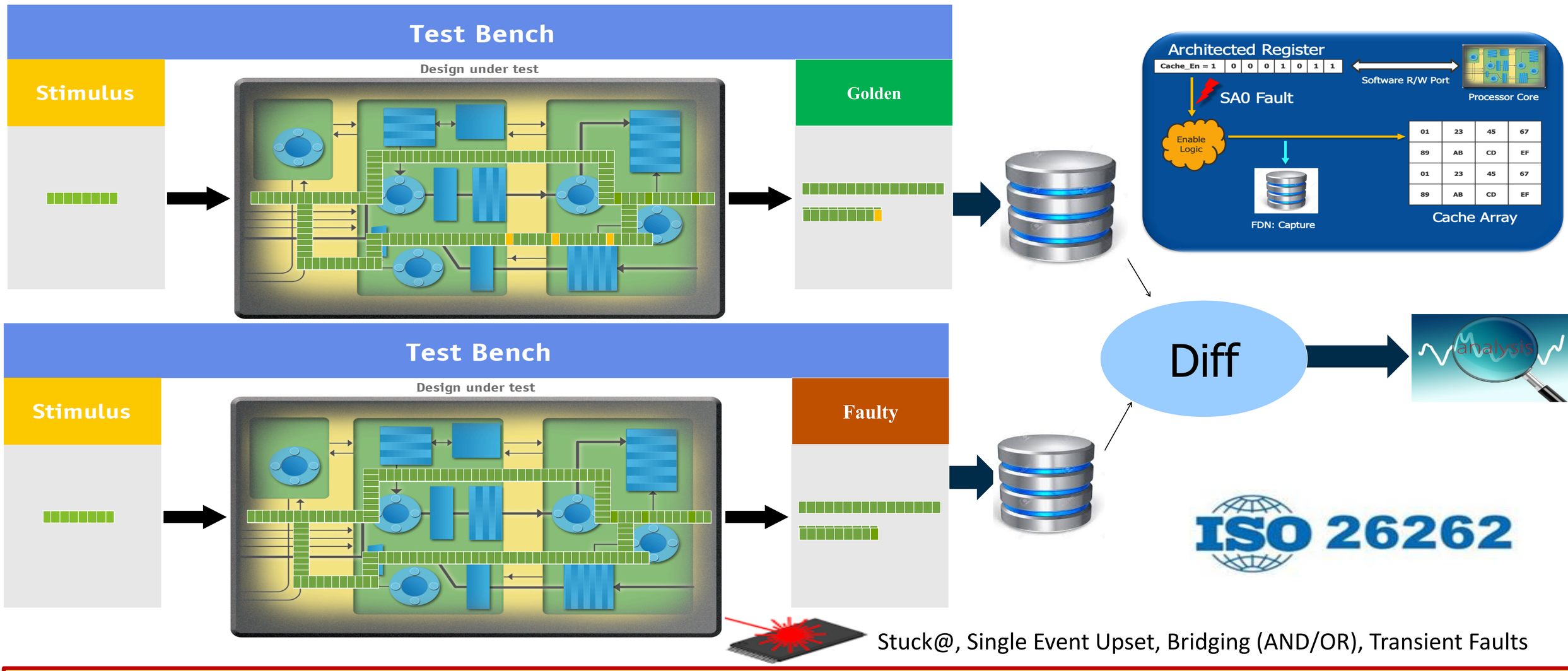
Functional Safety on Arm CPUs

Jitendra Aggarwal

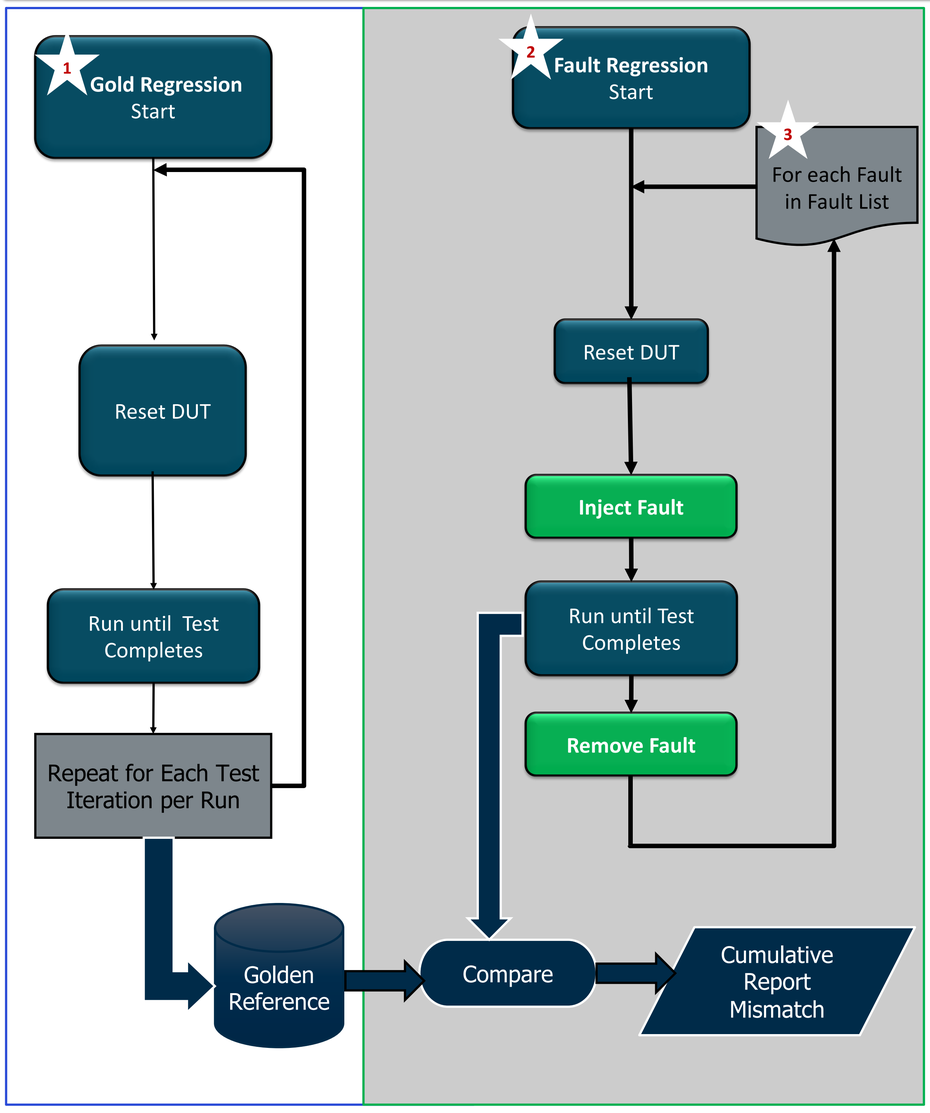
Harish Narayanaswamy

A Siemens Business

Perform System-Level Safety Circuit Verification, Analyze Effectiveness of Safety Mechanisms in a Design



Mimic Effects of Single Event Upsets (SEU) and Hard Faults Targeting Safety Critical Industries (Automotive, Aerospace, Military)

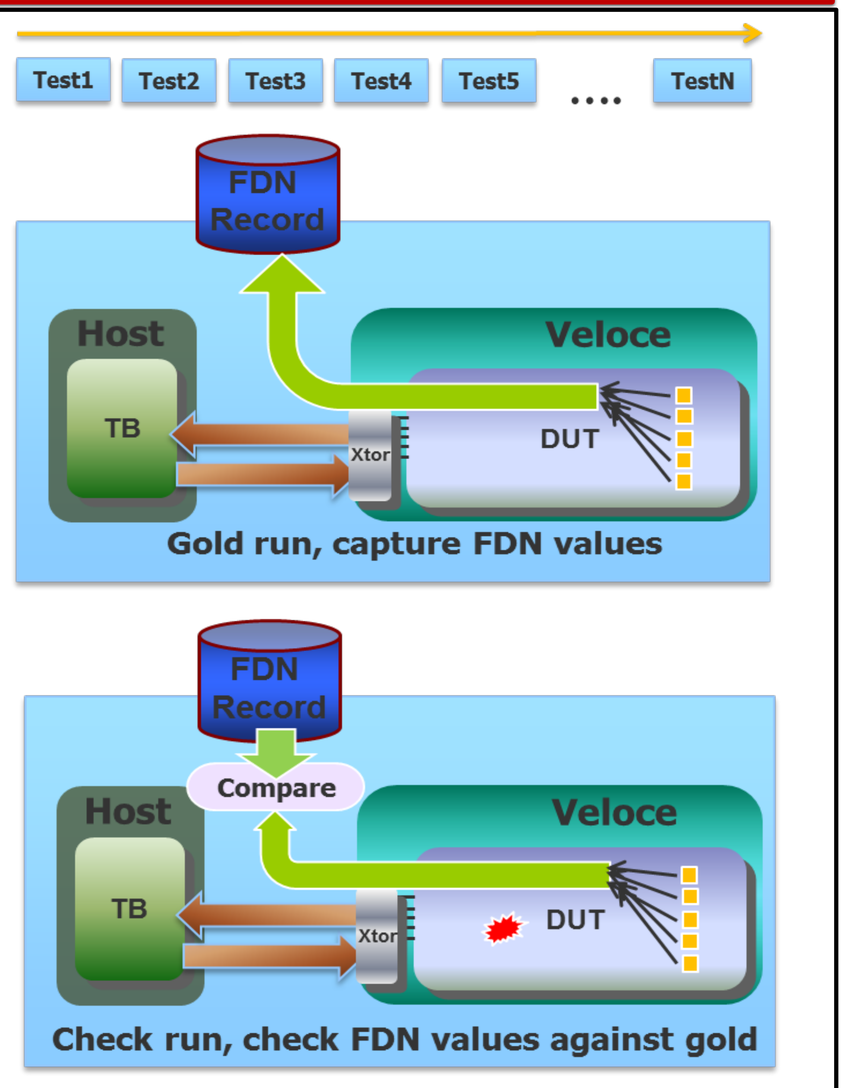


Fault Verification Flow

1. Design test executed with system reset and run until test completes successfully as golden run.
2. Same design is now executed with fault injection by system reset and run until test completes, later remove fault injected.
3. Execution process is repeated for multiple faults given in fault list. Faults are injected sequentially after every system reset:
   -> Inject Fault -> Run -> Remove Fault
   -> Compare Results with golden reference.

Develop Fault Verify Environment

- Generate a golden build without any fault injection for multiple fault comparison and repeat for multiple fault tests.
- Develop transaction-based soft-reset mechanism to efficiently run multiple faults sequentially.
- Implement transaction-based faster fault injection flow to minimize time between faults for best optimized runtime.
- Apply fault pruning methods to speed-up emulation, but include the redundant ones in coverage matrix for better transparency.
- Generate cumulative fault coverage reports from multiple tests.



DESIGN TYPE	M CLASS CPU - 1.2M FAULTS			R CLASS CPU - 370K FAULTS		
Benchmarking parameters	SIMULATION	EMULATION	GAIN	SIMULATION	EMULATION	GAIN
Total time sequentially	1500 days	1 day	1500x	180 days	2 hours	2100x
Number of parallel runs per compute resource	500 runs	1 run	500x	500 runs	1 run	500x
Total time with parallel runs	3 days	1 day	3x	9 hrs	2 hours	4.5x

Conclusion

Veloce Strato FaultApp: a robust solution to decrease compute resources and sim-hours with a significant ROI

- Developed new method for running multiple design copies on a single emulation resource
- Used to speed-up the whole process of fault injection and tolerance; simulation is a less viable solution
- Fault correction and tolerance are the biggest challenges with increasing CPU size and SOC complexity
- Fault coverage completeness is a bare-minimum requirement and a major challenge when targeting safety critical applications
- Emulation is needed to speed up turnaround and optimizing its automation is crucial to achieving expected ROI

Automation efforts were key to making this a generic solution that is easy-to-use in any kind of fault verification