2019 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Simulation Analog Fault Injection Flow for Mixed-Signal Designs

Pablo Cholbi Alenda, Analog Devices Inc., Bengaluru, India (Pablo.Cholbi@analog.com)

Dylan OConnor Desmond, Analog Devices Inc., Limerick, Ireland (Dylan.OConnorDesmond@analog.com)

Raman K, Analog Devices Inc., Bengaluru, India (Raman.K@analog.com)

***Abstract*—This paper proposes a methodology to perform analog fault injection simulations for analog and mixed signal semiconductor designs to fulfil the functional safety requirements in safety critical integrated circuit developments, as required for example by ISO 26262 for automotive parts.**

Keywords— ISO26262; automotive, fault injection; analog; mixed-signal

I. Related Work

Integrated circuits (ICs) for safety critical applications must adhere to safety standards and implement development flows with procedures which are otherwise not required for non-safety critical applications. In IC verification, one of these requirements may be to perform fault injection verification, both in simulation (pre-silicon) and on the physical hardware (post-silicon).

In digital designs, the flow and expectations are considerably well understood, and there are commercial electronic design automation (EDA) tools in the market targeted for digital fault injection simulation verification.

On the other hand, the flow and challenges for analog and mixed-signal (MS) fault injection simulation are not as well understood and widespread. Despite this, standard such as ISO26262 require analog fault injection (AFI) verification for some hardware safety levels.

II. Comparison to Existing Solutions

AFI is not a well-defined process within the industry. Because of this, the available EDA tools for AFI simulation differ in terms of their approach.

Cadence® have the Legato™ Reliability Solution as their AFI platform. Legato is based on the use of Verilog AMS for fault injection and simulation in the analog domain, which is not widely used in Analog Devices Incorporated (ADI). In addition, Legato uses Spectre® for simulation, which requires a costly license.

Mentor® has an AFI solution is named Tessent® DefectSim™. Originally, this tool was geared towards analog manufacturing fault grading, rather than AFI simulation. Defect Sim now supports AFI simulation, but the analog faults are injected at the transistor level, which contrasts with ADI's approach of injecting faults at the block I/O level.

Synopsys® do not have an available AFI tool at the time of writing.

Based on the disparity between the existing solutions and ADI's approach to AFI, the decision was made to develop an AFI tool in-house. An in-house analog fault injection tool was developed to support fault injection at the block I/O level, and uses ADI's internal SPICE-like simulator as the analog simulator. ADI's AFI tool provides a standardized approach for AFI and automates a large portion of the flow.

III. Application

This paper proposes a flow to approach the AFI simulation verification of mid-to-large MS ICs, building upon and extending the flows implemented by digital tool, and adapting them to the challenges of analog and MS co-simulation. This flow was applied to a 77GHz automotive radar sensor design and an automotive power steering angle sensor.

IV. SUMMARY

The purpose of this AFI simulation strategy is supporting the pre-silicon verification of safety measures (SMs) with respect to the safety requirements, including its capability to detect faults and control their effect.

Generally, the most effective SMs are the ones at a higher-level hierarchy, and the safe states are typically defined at the top-level of the hardware element (e.g. IC). It is therefore recommended to perform the simulation at chip top-level.

The functional top-level analog-digital co-simulation environment can be leveraged for the AFI simulation effort, building upon the already existing infrastructure to accelerate the flow bring-up. Figure 1 shows a simplified block diagram of an example top-level simulation environment; in this case for the automotive radar sensor.

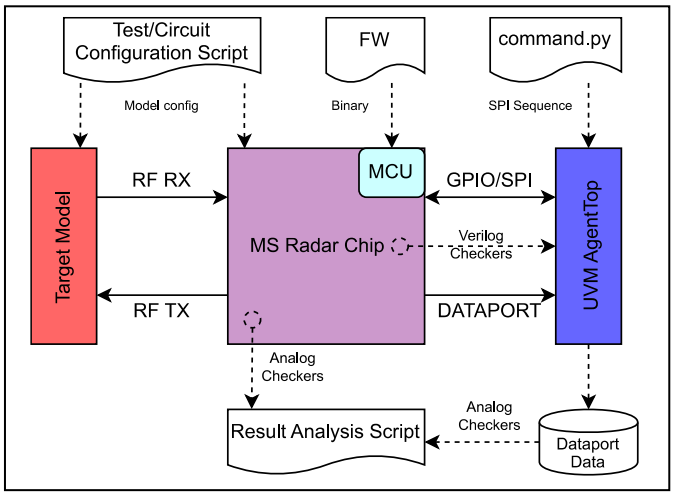

Figure 1. Example top-level mixed-signal co-simulation testbench

V. ANALOG FAULT INJECTION SIMULATION FLOW

The proposed methodology is an iterative flow represented schematically in Figure 2 covered in more detail in in this section.

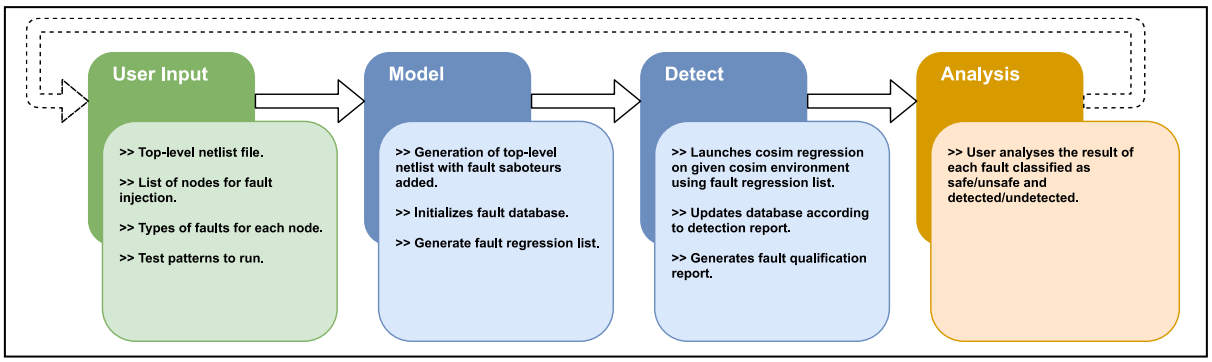

Figure 2. AFI simulation flow

1) User Inputs

As the AFI simulation effort is building on top of top-level co-simulation, several of the inputs are the same for both efforts. Mainly a top-level netlist, a simulation environment and a test or pattern to simulate.

Tests can be re-used from the functional top-level co-simulation regression, or additional tests can be developed for AFI simulations. However, the selected test case(s) should not try to “inject” any fault (ex.: through Verilog force), as this would result in more than one fault to be simulated concurrently, which could lead to inaccurate results.

Another input to the AFI simulation campaign is a list of simulator messages which will be used to flag if the fault was detected/undetected by the IC, or if it was safe/unsafe in the context of the system operation and safety requirements. Identifying the detected/undetected criteria are typically simpler: it is the way the IC flags that a fault

has been detected (ex.: a fault signaling pin). Identifying the safe/unsafe criteria can be a more involved as is it is an absolute check that the testbench will do on the environment and design under test. Testbench checkers must be developed to monitor the environment and DUT and print these messages.

In addition to the above, a fault node and fault type list also need to be provided to run a simulation campaign with those faults injected to observe the outcome.

2) Model Phase

In this phase of the flow, the top-level co-simulation SPICE netlist and the fault node/type list are used to generate an instrumented netlist, through which the different faults can be activated and simulated. This is accomplished by inserting a sub-circuit called a "fault saboteur" between the sub-circuit port on which in inject the fault and the internal circuitry. This procedure evidences one of the limitations of this fault injection flow: faults can only be injected at the port of a sub-circuit, but not an any arbitrary net. However, as the top-level simulation netlist will typically have a deep and detailed hierarchy, this should not suppose a major limitation.

Figure 3 presents an example of fault instrumentation in a netlist, showing the fault saboteur in dormant and activated modes.

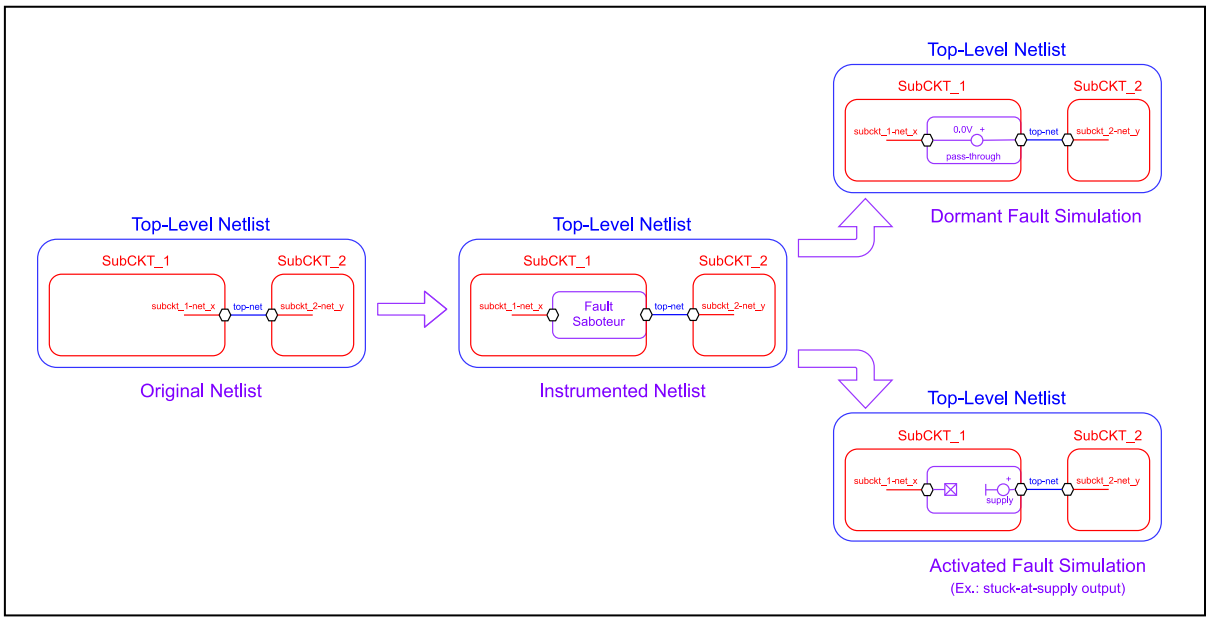

Figure 3. Example of netlist instrumentation for fault injection

By default, all the inserted fault saboteurs are in a dormant state where they behave like a passthrough. In this default behavior, the instrumented netlist is functionally identical to the original netlist. However, when a fault is activated in simulation, one fault saboteur is instead executed as a different sub-circuit which implements the behavior desired to cause the expected fault.

Conceptually, the dormant fault saboteur is a 0V ideal voltage source in series. While intuitively dormant faults do not alter the functionality of the design, a stronger argument needs to be built around the integrity of the netlist's functionality.

To build confidence in the instrumented netlist, the top-level co-sim environment can leverage further. Two co-simulation regressions will be run: one on the original netlist and one on the instrumented netlist with all the fault saboteurs in dormant (default) state. The functional test plan is expected to comprehensively verify the functionality of the design. Therefore, if both regressions with the same tests and checkers finish with the same status, it can be argued that the instrumented netlist is functionally equivalent to the original netlist in the use case(s) of the design under test. Figure 4 represent schematically the functional equivalence check flow.

In digital fault injection (DFI) tools generally consider permanent stuck-at-0/stuck-at-1 faults and, if required, transient faults. It is important to highlight that the faulty node will take one of two values (0 or 1). On the other hand, in AFI the problem is not as well defined. Analog faults are not limited to exhibiting two states and intermediate values.

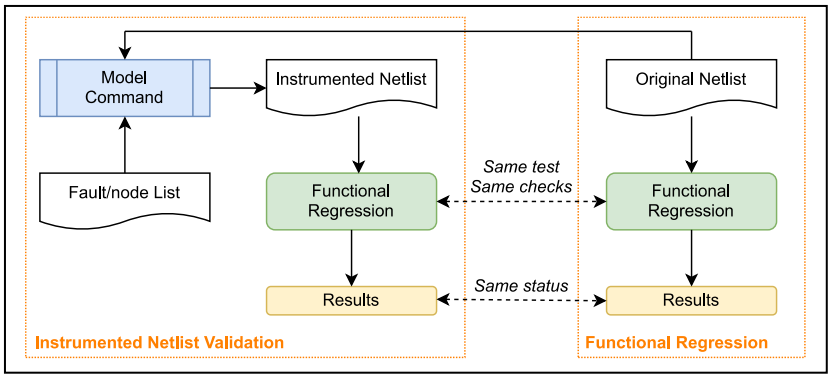

Figure 4. Checking functional equivalence of instrumented netlist

Reference [1] does not define a set faults to consider in an AFI simulation campaign, just an initial list of failure modes which can be extended and tailored as needed, with the relevant justification. Reference [1] requires expert judgement to identify the failure modes to be simulated. Expert judgement is also necessary to break down these failure modes into faults which could cause such failure modes. This will yield the list of AFI faults used in a project. Below are listed a set of *core* fault types which can cover over 90% of faults across projects, as per the experience of the authors:

- Stuck-at-supply
- Stuck-at-ground
- Open-circuit
- Voltage drift
- Voltage oscillation
- Transient

However, the library of fault models can be extended as needed by defining new fault sub-circuits. A fault saboteur is just a sub-circuit with ports A and B. Users can define additional fault models as needed, with this specification in mind. Once a fault model is proven in simulation and validated against silicon results, it can be incorporated into the general fault library, building it up as an internal verification intellectual property (VIP).

There is one last type of fault which does not need instrumentation: a parametric fault. This fault alters the otherwise constant value of a sub-circuit, model or element parameter to cause a deviation in the nominal behavior. For example, the impedance of a resistor or the gain of an amplifier.

3) Detect Phase

In this phase of the flow, a simulation regression is run with the faults injected. If N faults note/type pairs were defined in the input, N+1 simulation are run: 1 “good run” without any fault activated, and N fault simulations each with one fault activated.

The good run can be considered a sanity check. This simulation should pass, and no faults should be detected. This assumption is tested in the good run.

The outcome of each of the N fault simulations is a fault classification. Depending on whether the fault was detected or undetected by the hardware (or hardware + firmware); and whether the fault causes the safety goal to be violated or not. This results in the 2-dimendion classification shown in Table I.

Table I. Fault classification

	Detected	**Undetected**
Safe	Safe Detected	Safe Undetected
Unsafe	Unsafe Detected	Unsafe Undetected

To enable this classification, the test logs are parsed in search of two sets of pre-defined messages:

- Detect message: the hardware signaled the detection of a fault condition and enters the safe state.
- Unsafe message: The safety goal has been violated.

The testbench agents will monitor the relevant signals and issue these messages as needed.

One the regression has completed, each fault should be classified under one of the four possible bins, and this information captured in a report.

Figure 5 represents schematically the model and detect commands of the AFI simulation tool.

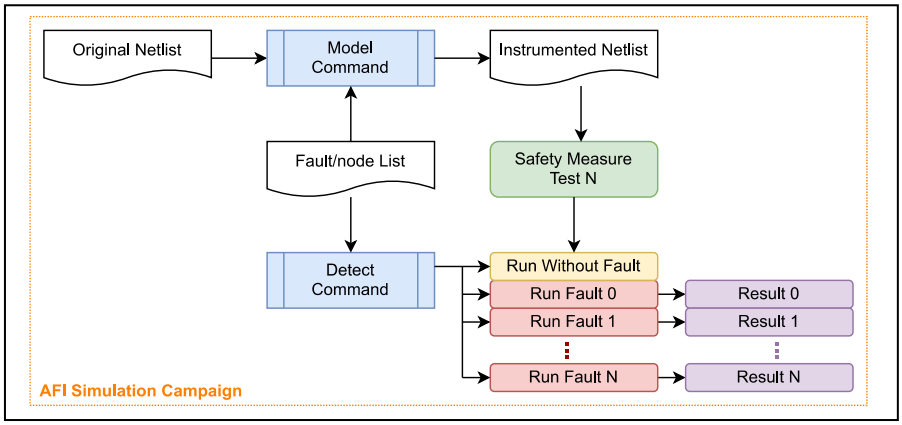

Figure 5. AFI simulation regression flow

4) Analysis

After the detect phase, the output report is analyzed by the functional safety, the design and the verification engineers to review whether the outcome of each fault and the safety metrics are as per assumptions.

An unexpected result should be treated as a verification anomaly and tracked with an issue tracker for further action/closure. Further actions could be repeat test, review simulation parameters and re-run depending upon the anomalies. If the detected anomalies are shown to be valid it could result in modification of the one or many of the following:

1. Safety analysis: modify the assumptions of the safety analysis.
2. Design: modify hardware design, configuration, or firmware to remove anomalies.
3. Environment: anomalies could have occurred due to limitations of the test environment.
4. Refinement of test criteria: the checkers and the pass-fail criteria could not be consistent with the requirements. This could include creating additional tests to improve coverage.

VI. RESULTS

Having a common and capable AFI simulation tool has helped standardize the methodology across automotive projects within the organization. This in turn has shortened and eased the deployment of the tool in the projects and simplified debug as well as reuse. This tool has also helped the organization in aligning the projects with the standards and practices used in industry.

ADI's internal AFI simulation tool has been successfully used in two automotive projects (radar, power steering) to simulate and categorize thousands of faults to support the project's safety analysis.

At the time of writing, the AFI tool presented in this paper is also planned to be used in an automotive battery charge monitor in the immediate future.

VII. FUTURE ENHANCEMENTS

There are several feature enhancements planned for the AFI tool. The tool currently requires several custom switches to be defined with ADI's internal regression management tool's configuration file; these will be integrated

directly into the simulation regression tool. A graphic user interface (GUI) is under development to allow users to easily create the required setup files for the AFI tool. An integration with ANSYS® medini™ analyze is currently being investigated, to see if there is a meaningful flow that can be developed linking the tools. The addition of Spectre™ support in the FI tool is also planned for evaluation. Verilog based co-simulations implementing Verilog AMS models for the fault models is another possible feature to be evaluated. Industry developments are being continuously monitored, with regular updates from vendors on their AFI solutions, so that ADI can remain aligned with the evolving AFI flows as standards.

ACKNOWLEDGMENT

We would like to thank Sriram Madavswamy, Mike Keane and Nilkanth Pathak for sharing the AFI simulation campaign reports. We would also like to thank Yuhong Huang and Alan Whooley for reviewing and collaborating on the AFI simulation strategy and provide valuable feedback in the process.

REFERENCES

[1] International Standard Office (ISO), ISO 26262 Road Vehicles-Functional safety. Geneva, Switzerland: ISO, 2018.