Fault Effect Propagation using Verilog-A for Analog Test Coverage

Aishwarya Prabhakaran, Infineon Technologies, Munich, Germany (Aishwarya.Prabhakaran@infineon.com)

Ahmed Sokar, Infineon Technologies, Munich, Germany (Ahmed.Sokar@infineon.com)

Jaafar Mejri, Infineon Technologies, Munich, Germany (Jaafar.Mejri@infineon.com)

Abstract — In today's small size transistors technologies, transistors become more susceptible to defects that occur either due to process variations or defects like open and short circuits during the manufacturing process. To evaluate the quality of the IC test, Analog test coverage is essential, although it is computationally expensive. Performing system level Analog Defect Simulation (ADS) with several complex analog blocks at the transistor level is time-consuming. One possibility to reduce ADS time is to use behavioral models for circuit blocks where defects are not being injected. Yet, the models should be able to propagate the defects responses to the observation point (a pad). A methodology for augmenting fault effects to the Verilog-A behavior models of analog circuits using Look-Up Tables (LUT) is presented in this paper. System-level ADS is performed where analog blocks are replaced with their behavior models. This speeds up the simulation of defects in analog circuits.

Keywords—Analog Test Coverage; Analog Defect Simulation; Behavioral Modeling; Modeling out of specifications; Fault Propagation; Verilog-A; Analog DFT

I. INTRODUCTION

With transistor size shrunk to few nanometers, and with the integration of millions of transistors into a chip, transistors became more susceptible to defects that occur due to process variations, short circuits and open circuits during the manufacturing process. Integrated Circuits are manufactured in large numbers, and the test procedures adopted to verify their performance must be fast to reduce test cost. At the same time, the testing must be efficient enough to identify all production defects. In the digital domain, Design-For-Test (DFT) techniques like Automatic Test Pattern Generation (ATPG), Scan-based DFT, etc and different fault models like Stuck-at faults, drive faults, path delay faults, etc. have been used to screen defects in the digital blocks. Fault simulation techniques like serial, parallel, deductive and concurrent fault simulation have facilitated digital fault simulation to a great extent. In the analog domain, there is an increasing demand to perform Analog Defect Simulation (ADS) by injecting potential defects to the circuit and checking whether the test programs detect them. On the one hand, this is used to make sure that the test program which we uses on the automatic test equipment (ATE) has high defect detection coverage. On the other hand, the ADS checks whether a chip is being over/under-tested. The test coverage of each test shows how many defects we can detect if we used this test. By comparing the detected defects for each test, we can know whether we have redundant tests. If the overall coverage of all the test is not high enough, it indicates that we are under testing the chip. In the functional safety domain, ADS is used on block level for the safety-relevant blocks which shall comply with the ISO26262 standard [1]. In this paper, the focus will be on the Analog Test Coverage where a fault must be manifested at the system-level [chip pads/pins]. This is because observability of any defect is at the system level after the manufacturing process. Performing system-level ADS with transistor models is impractical due to the long simulation time. This demands the use of abstract models in place of transistor models.

Performing system level ADS at the transistor level is time-consuming. Research work to find speed-up mechanisms in the field of fault simulation has been extensive. The approaches include the use of abstract models to speed-up ADS as proposed in [2], [3] and [4], concurrent simulation and fault collapsing as proposed in [5], layout based fault extraction as proposed in [3] and [6] identify the likelihood of a fault occurrence and simulate the faults that are more likely to occur first.

The motivation for this paper is to inject one defect at a time into a circuit's transistor level netlist and replace all the consecutive blocks in the path leading to the observation point, Fault Propagation Chain (FPC), with abstract models in order to speed up ADS. These abstract models must react the same way as the transistor models would react to stimuli coming from a block that has a defect. There is a difference between the needed models for ADS and the used models for functional verification. We develop models for functional verification to behave in the functioning range of the block and it may through an assertion or an error message if the stimuli is outside the expected range of the correct function of the circuit. Neither an assertion nor an error message is sufficient for ADS, as the models should be able to propagate the effect of a defect coming from the input blocks to the blocks at the output until the error reaches the observation points. The focus of this research work is rapidly building behavioral models that are capable of dealing with stimuli coming from a faulty block and propagating it to the subsequent block.

II. PROPOSED METHODOLOGY AND APPLICATION

As a first step, Verilog-A behavior models for all analog blocks, that have to be modeled, must be developed for the nominal operating conditions. Circuits are modeled in such a way that the performance metrics of an analog circuit are used as parameters and the outputs of the model are a function of these parameters. For example, the output waveform of a Voltage Controlled Oscillator (VCO) depends on the performance metrics of the circuit like Oscillating frequency, Phase, Duty cycle, Output Voltage and so on. The Model is extended after that to cover the effect of receiving faulty inputs out of the developed range of operation. Three steps are done as depicted in Figure 1. In this paper, the block which is needed to be modeled is named "Target block". All the blocks that feed this target block with stimuli are named "Source blocks".

A. Fault Injection

The first stage in the proposed methodology is to inject defects in the source block and simulate the source and the target block (with all the intermediate blocks at the abstract level if any). Open and short circuit across the transistor terminals are considered for ADS. An open circuit is created by connecting a resistor of large resistance in the order of Mega Ohms in series to the device terminal (Source, Drain or Gate) to disconnect the terminal from the rest of the circuit. The short circuit is created by connecting a resistor of small resistance in the order of a few Ohms in parallel to two device terminals, shorting the terminals together. To simulate possible defects, open and short circuit faults are extracted from the layout. A fault model written in Verilog-A injects open and short circuit faults dynamically. This fault model replaces the parasitics capacitors and resistors in the netlist with shorts and opens as the parasitics provide a good indication where a short or an open circuit could occur. An in-house regression tool developed by Infineon, is used to activate one fault per simulation. We used the regression tool to optimize sending the simulation jobs in parallel to the compute farm. Cadence Maestro or any tool that supports parallel simulation of parametric sweep can be used instead.

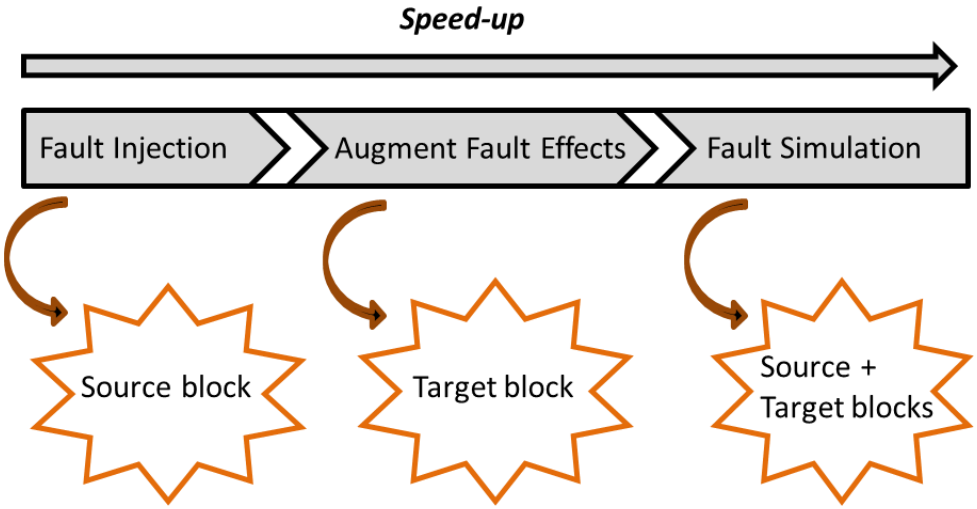

Figure 1. An Overview of the Methodology

B. Fault Effect inclusion into the model

Figure 2 shows the Methodology of fault effects inclusion into the target block. In the first step, defects (open-/short circuits) are injected in the source block as discussed in the section Fault Injection. Meanwhile, measurements of the performance metrics of the target block, which are subject to change under fault conditions, are set up in the simulator. Block level fault simulation is performed, and for each activated defect in the source block, the performance metrics of the target block are captured. These performance metrics of the target block are controlled by the input parameter (s) that is (are) subject to change under fault conditions. For example, in the case of a VCO, the output parameters depend on the input control voltage. Therefore, these output parameters are mapped to their respective input parameters. This one-to-one mapping is done with the help of Look-Up Tables (LUTs). For each output parameter, a LUT is generated. These LUTs are then appended to the behavior model with table function in Verilog-A HDL as shown in Figure 3. The table function chosen allows multi-dimensional mapping; an output parameter can depend on more than one input parameter. The table function that looks for an exact match, like finding a string in a file, will return nothing if the input value is slightly different from the input value in the LUT. This leads to an inaccurate result when compared with the transistor model. To address this problem, the table function in Verilog-A is enabled with interpolation techniques.

For a fixed test pattern and for each activated defect during the block-level fault simulation, the fault response of the source block is recorded. The fault response of the source block for each activated defect is compared with the defect-free case. A defect, that does not cause any deviation in the performance of a circuit at this level, cannot be detected at the point of observation. Therefore, defects that do not have any effects on the source block are marked as 'undetectable' and are dropped from the defect list. Different test patterns must be chosen to detect these defects or theses detects may be generally undetectable defects.

The fault-effect-appended behavior model is compared with its respective transistor model for equivalence. For an activated defect, the accuracy between the parameters of the behavior model and the transistor model is computed. The difference is expected to be within the tolerance limit specified; otherwise, the behavior model must be rewritten in order to meet the requirements.

C. Fault Simulation

The last step in the methodology is the system level fault simulation. Fault simulation is performed with the developed abstract models. These abstract models are now capable of propagating the fault effects that occur due to the presence of a fault in its preceding blocks. The effect of the fault injected into the circuit must be detected at the point of observation. This point of observation can be an output pin of the chip or a test pad. These are the point where any fault can be detected in the real field. For a given test pattern, the response of a circuit to the given input is known. In the case of fault simulation, the test patterns are fixed. For this fixed test pattern, the fault is injected in a circuit which is the source block. All the inputs to this source block must be known for the test pattern applied. This can be known from the nominal case test results. With the fixed input, the presence of a fault alters the output response of a circuit.

For a given test pattern, faults are injected in the source block, the blocks in the FPC are replaced with fault effect appended Verilog-A models. The response of the end block in the FPC for the fault simulation carried out with faults activated in the source block are stored. These results can be used to make a fault dictionary. The response from the fault dictionary can be compared with the response from the test results of the nominal case.

Based on the results of performance metrics comparison, a fault can be marked detected/undetected. The performance metrics under faulty case can be compared to the fault-free case with a tolerance that is user-defined. If the difference between the metrics is within the tolerance, the results of fault simulation is identical to that of the fault-free case. A fault that is injected on a block, leading to a response at the system level that is identical to the fault-free case, is marked as ’Not detected’. To detect these faults, a different test pattern must be chosen and the whole process must be repeated for this new test pattern applied. A fault that is injected on a block, leading to a response at the system level that differs from the fault-free case more than the tolerance limit, is marked as ’Detected’. The faults that are marked as ’Detected’ can be dropped from the fault list.

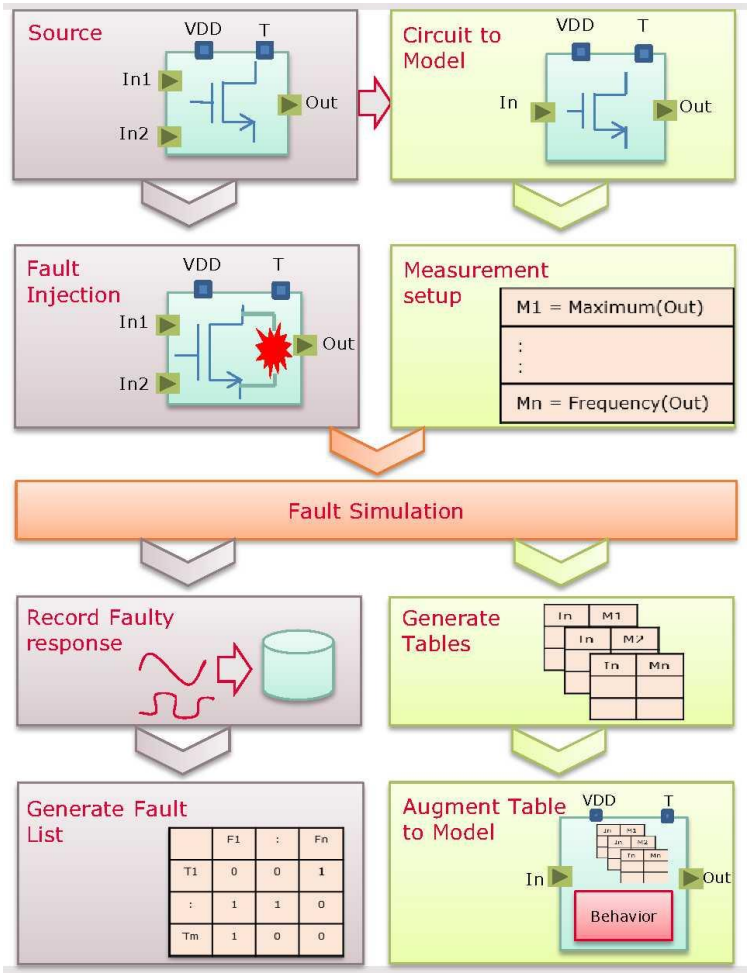

Figure 2. Fault Effect Augmentation

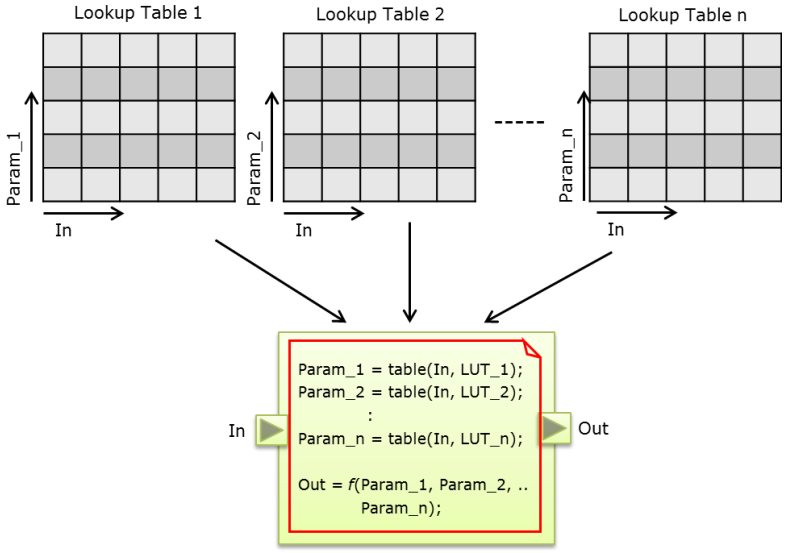

Figure 3. Look-Up Table Mapping

III. EXPERIMENTAL RESULTS

A chain of circuits including a Bandgap Reference circuit, a Voltage to Current converter, an Oscillator, analog switches, a Chargepump and the Load is chosen for experimentation, as shown in Figure 4. The output voltage from the Bandgap Reference circuit is converted to a current signal by the voltage-to-current converter circuit. This reference current controls the Oscillator circuit which produces clock signals. These clock signals pass through switches controlled by the enable signals from the digital blocks. These clock signals from the Oscillator circuit controls the Charge pump circuit. The charge pump circuit produces twice the supply voltage across the load.

For faults injected into the Bandgap Reference circuit, the current input to the oscillator is affected. This faulty input current affects the performance metrics of the Oscillator circuit like frequency, duty cycle, etc. This change in the performance metrics of the oscillator affects the pumping rate of the charge pump. The performance metric of the charge pump that is affected is measured across the load.

A short circuit fault is modeled by connecting a resistor of 10 Ohms resistance in parallel to two device terminals and an open circuit fault is modeled by connecting a resistor of 1G Ohms resistance in series with the device terminals. 375 faults are injected into the post layout extracted netlist of the Bandgap Reference circuit. Behavior models for all the blocks in the chain following the source block are developed and integrated with fault effects. The reference output voltage from the Bandgap Reference circuit for each fault is mapped to the corresponding current using LUT. This input current controls the oscillator performance metrics like frequency, Duty cycle etc. LUT of reference current versus frequency, reference current versus duty cycle, and so on, are generated by sorting the entries in ascending or descending order and by removing the redundant entries. For each output parameter, LUT is used. The table function in Verilog-A takes reference current as an input and returns the corresponding parameter value. These parameters are input to the signal generator which produces the output response of the oscillator. The clock output of the oscillator passes through switches and it controls the rate of pumping of the charge pump.

System-level fault simulation is performed with Bandgap Reference circuit in transistor level and the other circuits in the chain are replaced by their respective fault-included behavioral models. The average output current at the load is measured for each fault activated. The frequency distribution of the average current measured is shown in Figure 5. The average current measured for each fault activated is compared with the fault-free case. The tolerance limit is set to 10%. Figure 6 shows that out of 375 faults injected, 13% of the faults produce the same result as the fault-free case and 51% produce results that differ by less than the tolerance limit. These faults are marked 'Not detected' and a second test pattern must be chosen to detect them. 36% of the faults produce results that deviate by more than the tolerance limit. These faults are marked 'Detected' and these faults are dropped from the fault list.

To measure the average settling current, the transient simulation time used is 20us and the full fault simulation with abstract models completed in 2090 s with 40 simulations run in parallel. The speedup achieved with behavior models is shown in Figure 7.

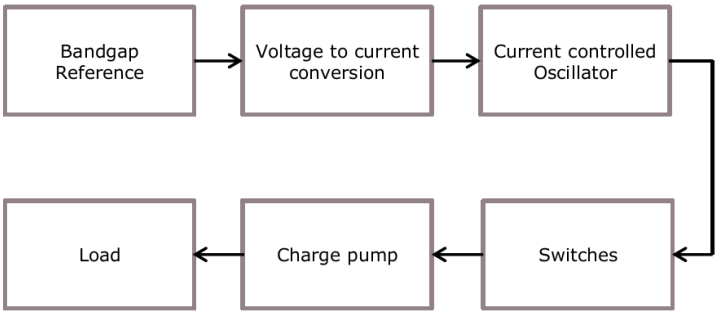

Figure 4. Blocks on the Fault Propagation Chain

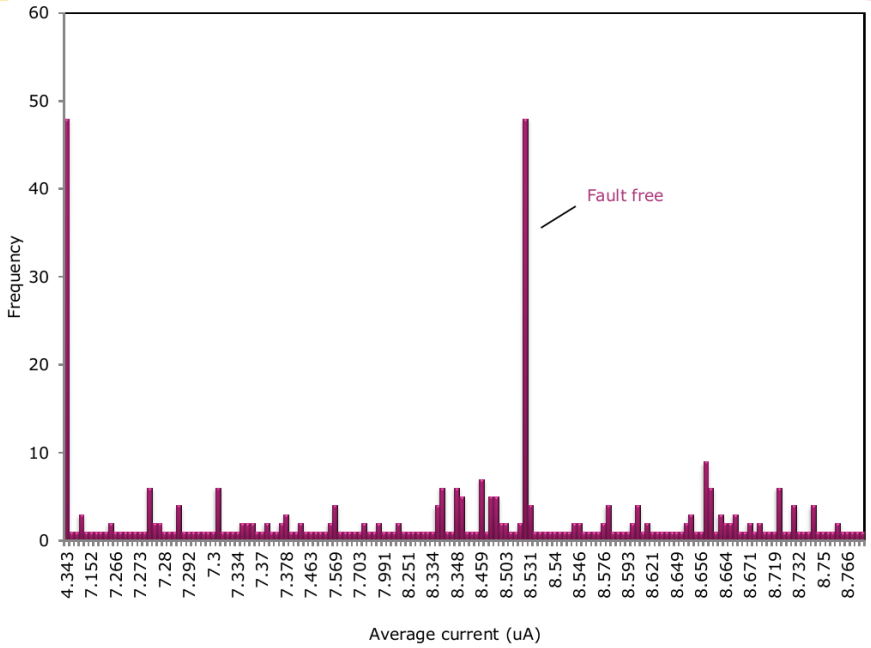

Figure 5. Frequency distribution of Average current measured at the load

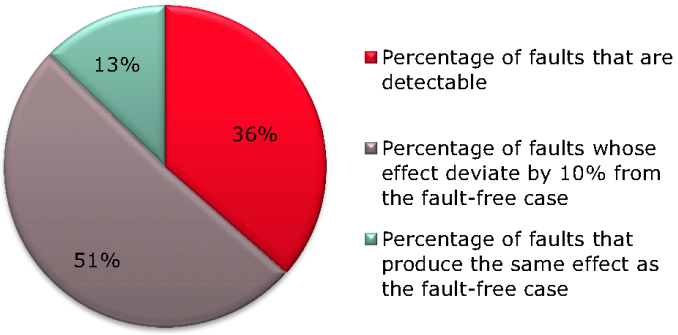

Figure 6. Fault detection results

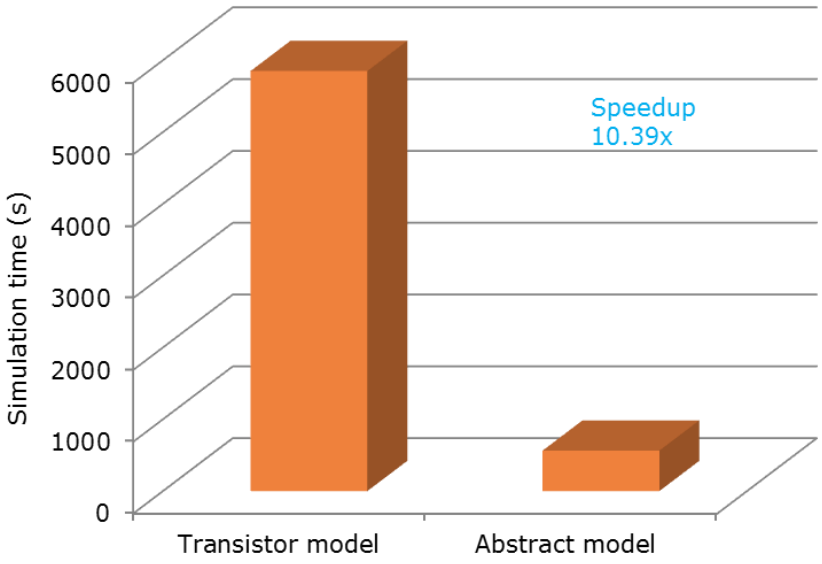

Figure 7. Simulation time comparison

IV. FINDINGS AND FUTURE WORK

A. *Fault Collapsing*

Some of the simulated defects from the Source Block are having a response that is within the limit of the acceptable stimuli range of the Target Block. In other words, some defects in the Source Block are covered by the functional model, which is developed in the first place without the fault-included extension. Some other defects are having a very similar response. These kinds of similarities in the responses are classified and grouped. Each group of these similar classes is presented once in the LUT, which reduces the size of the lookup table and increases the speed of the model.

B. *The lookup Table approach*

The LUTs are fast enough in the simulation but they need a big storage space. For a Target Block that has multiple blocks giving it stimuli, the number of defects will increase and accordingly the number of entries in the LUT. This will require doing Fault Collapsing as discussed in section A.

C. *Automation and speeding up behavior model building*

Extending the behavior model to include the response of stimuli from a block with a defect is automated. The automation of the process can be more enhanced if the LUT of one block is used for developing the subsequent blocks. To model a new Target Block to the same Source Block, there is no need to inject the defects again and save their responses. The already developed LUT can be accessed by an entry which represents the defect number. The benefit of that flow will be obvious in speeding up modeling Target Blocks which have Source Blocks that need long simulation time.

D. *Reusability*

A limiting issue of this methodology is the reusability of the models. These behavior models are extended in an automatic fashion based on the surrounding blocks. If the surrounding blocks to the Target Blocks are changed, the whole flow should run again to regenerate the LUTs. This prevents the reusability of the behavior model of a certain block. A tradeoff between the simplicity of the approach and the reusability has to be made.

V. CONCLUSION

To perform Analog Defect Simulation (ADS) using behavioral models to speed-up the simulation, behavioral models which were developed to cover the nominal behavior of the circuit had to be extended to cover the fault effects. A methodology for generating behavior models that propagate the defects in an analog circuit has been proposed in this work. To identify the impact of a defect on a circuit, each defect has been injected into the post layout extracted netlist of a circuit. The effect of a defect, injected in the source block, on all the other target blocks up to the observation point, has been captured by including one target block at a time. The effect of the defects was included in the behavioral model of the Target Block in LUTs along with the Verilog-a code of the nominal function. Finally, the defect simulation has been performed with behavior models generated to speed up the simulation. The methodology for building the behavior models is easy to be implemented and automated. The cost of the simplicity is a large amount of storage needed for LUTs and the lack of model reusability without rerunning the whole development flow again. The storage issue is tackled by fault collapsing algorithms which reduced the size of the LUTs dramatically. The flow has been developed and tested on a chain with almost 6 blocks. A speed-up of 10x in the defects simulation have been recorded. The acceleration is expected to increase with the increase of the number of modeled blocks. Around 49% of defects were undetectable defects or defects which their response is covered already by the functional behavior model without the fault extension.

VI. REFERENCES

[1] ISO/TC 22/SC 32, E. a. (2011, November). ISO 26262. Retrieved from ISO: https://www.iso.org/standard/43464.html.

[2] Enrico Fraccaroli, Franco Fummi, ”Analog fault testing through abstraction”, Design, Automation & Test in Europe Conference & Exhibition (DATE), May 2017.

[3] A.Meixner and W. Maly, ”Fault Modeling for the testing of mixed integrated circuits”, Proceedings of International Test Conference, 1991.

[4] Jyotsna Sequeira, Suriyaprakash Natarajan, Prashant Goteti and Nitin Chaudhary, ”Fault Simulation for Analog Test Coverage”, Test Conference (ITC), 2016 IEEE International, Nov 2016.

[5] Joonsung Park, Srinadh Madhavapeddi, Alessandro Paglieri, Chris Barr and Jacob A. Abraham, ”Defect-Based Analog Fault Coverage Analysis using Mixed-Mode Fault Simulation”, IMS3TW ’09. IEEE 15th International Mixed-Signals, Sensors, and Systems Test Workshop, June 2009.

[6] Z.R. Yang, M. Zwolinski, ”Fast, robust DC and transient fault simulation for nonlinear analogue circuits”, Design, Automation & Test in Europe Conference & Exhibition (DATE), March 1999.

[7] Ender Yilmaz, Geoff Shofner, LeRoy Winemberg, Sule Ozev, ”Fault analysis and simulation of large scale industrial mixed-signal circuits”, Design, Automation & Test in Europe Conference & Exhibition (DATE), March 2013.