Fast Time-Domain Simulation for Reliable Fault Detection

Bratislav Tasić, Jos J. Dohmen,
Rick Janssen
NXP Semiconductors
5656 AE Eindhoven, the Netherlands
{bratislav.tasic,jos.j.dohmen,
rick.janssen}@nxp.com

E. Jan W. ter Maten
Bergische Universität Wuppertal
42119 Wuppertal, Germany
termaten@math.uni-wuppertal.de

Roland Pulch
Ernst-Moritz-Arndt-Universität Greifswald
17487 Greifswald, Germany
roland.pulch@uni-greifswald.de

Theo G.J. Beelen
Eindhoven University of Technology
5600 MB Eindhoven, the Netherlands
t.g.j.beelen@tue.nl

***Abstract*—Imperfections in manufacturing processes may cause unwanted connections (faults) that are added to the nominal, ”golden”, design of an electronic circuit. By fault simulation we simulate all situations: a huge number of new connections and each with many different values, up to the regime of large deviations, for the newly added element. We also consider ”opens” (broken connections). A strategy is developed to efficiently simulate the faulty solutions until their moment of detection. We fully exploit the hierarchical structure of the circuit. Fast fault simulation is achieved in which the golden solution and all faulty solutions are calculated over the same time step.**

I. Introduction

Designs in nanoelectronics often lead to large-size nonlinear simulation problems. Here DC simulation followed by time domain simulation are the kernel steps in the design process. Industry demands the provisions of variability to guarantee quality and yield. This can be done by carefully setting up Monte Carlo simulations. To guarantee yield, i.e., by estimating the probability of failures, even special Monte Carlo Methods have been designed [1], [20]. One has to simulate in the tail of the probability density functions. When one thinks about variation of parameters, this is in the regime of large deviations. One may also speak of 6-σ variations (assuming a normal density distribution). We have used these techniques to guarantee yield under harsh conditions for Static Random Access Memories (SRAMs). An SRAM is used as a building block for the construction of large Integrated Circuits (ICs). To ensure that a digital bit cell in SRAM does not degrade the yield (fraction of functional devices) of ICs with Megabits of memory, very small failure probabilities $P_{\text{fail}} \leq 10^{-10}$ are necessary. To simulate such probabilities, regular Monte-Carlo simulations require too much computing time, so more advanced techniques are required. Importance Sampling is a technique that provides sufficiently accurate results and is relatively easy to implement [12], [13], [18]. Using this, a speed-up of several orders can be achieved when compared to regular Monte Carlo methods [1], [5]. And, more important, we were able to optimise the SRAM active column, while guaranteeing that the failure probability remained below 10^{-10} by imposing a stochastic constraint [6], [20].
A refinement here is the use of a response surface model technique [21], which can reduce the cpu costs of evaluating many Monte Carlo samples. A new class in this area is provided by recent techniques from Polynomial Chaos Expansions (PCEs), where the expansions effectively provide a response surface facility [14], [29]. The PCEs are most accurate where the involved probability density functions are larger than some treshold [24]. Hence, when applying these techniques to estimating very small tail probabilities a careful mix of new evaluations and results obtained from evaluating the response surface model has to be taken into account [15], [16].

After having applied all these kinds of simulations and making the layout design, the manufacturing process still may introduce a completely new effect. Here we do not mean the effect of geometrical inaccuracies in widths and lengths of several components: these effects can be analysed in advance by proper techniques from sensitivity analyses. Here especially, the PCEs offer great benefit in automatically providing mean and standard variation of each local quantity at any moment of time [14], [29]. The new phenomenon comes from imperfections in manufacturing processes that may cause unwanted connections (faults). They can downgrade the performance of the carefully designed circuit. To analyse their impact we add artificial, faulty elements to the nominal, ”golden”, design of an electronic circuit. By fault simulation we simulate all situations: a huge number of new connections and each with many different (resistance) values, up to the regime of large deviations. We also consider ”opens” (broken connections). A strategy is developed to efficiently simulate the faulty solutions until their moment of detection: i.e., when at the measurement time moment and location the difference between the faulty solution and the golden solution exceeds a certain treshold. We store the results in a database. This database is of help to first externally diagnose a faulty IC and to identify the candidate circuit submodels where the

978-3-9815370-6-2/DATE16/©2016 EDAA

fault may have happened. After that the IC can be studied further internally. This can help to improve next productions. Moreover, the collection of simulations can also be helpful as a priori check before layouting.
Note that essentially we are looking to the weak spots in the circuit. In our approach the manufacturing process is the immediate cause of the problem. However it can also show up later, due to effects of ageing of the design, or by stress effects due to heating. It is also related to other network problems, e.g., in analysing traffic behaviour in a city when suddenly a road is blocked, or when a new connection pops up. Our approach can be extended to energy distribution networks, sewage systems, and even to networks that are not constant of size in time.

In the following sections we briefly summarise the main algorithmic steps in time domain circuit simulation. These will help to understand how Fast Fault Simulation can be done.

II. Time integration of circuit equations

The electronic circuit equations can be written as [8], [9]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{q}(\mathbf{x}) + \mathbf{j}(\mathbf{x}) \quad = \quad \mathbf{s}(\mathbf{x}, t). \qquad (1)$$

Here $\mathbf{s}(\mathbf{x}, t)$ represents the specifications of the sources. The unknown $\mathbf{x} = \mathbf{x}(t)$ consists of nodal voltages and of currents through voltage defined elements. We assume that $\mathbf{q}(\mathbf{0}) = \mathbf{0}$, and $\mathbf{j}(\mathbf{0}) = \mathbf{0}$. At $t = 0$, the initial solution is given by the DC solution, $\mathbf{x}(0) = \mathbf{x}_{\mathrm{DC}}$, which satisfies $\mathbf{j}(\mathbf{x}_{\mathrm{DC}}) = \mathbf{s}(\mathbf{x}_{\mathrm{DC}}, 0)$.
For time integration in circuit simulation we consider the BDF1, or Euler Backward method. Assuming time points $t_{k+1} = t_k + h_k$ $(k \geq 0)$ with stepsizes h_k and approximation $\mathbf{x}^n$ at t_n, BDF1 calculates $\mathbf{x}^{n+1}$ by

$$\frac{\mathbf{q}^{n+1} - \mathbf{q}^n}{h_n} + \mathbf{j}^{n+1} \quad = \quad \mathbf{s}^{n+1}. \qquad (2)$$

Here $\mathbf{q}^k = \mathbf{q}(\mathbf{x}^k)$, $\mathbf{j}^k = \mathbf{j}(\mathbf{x}^k)$, for $k = n, n+1$, and $\mathbf{s}^{n+1} = \mathbf{s}(\mathbf{x}^{n+1}, t_{n+1})$. The system is solved by a Newton-Raphson procedure. For efficient direct methods to solve the intermediate linear systems, see [3]. A fixed Jacobian can reduce the number of LU-decompositions, but, in general, it will increase the number of iterations and thus the number of (costly) evaluations. Also, in the case of circuit simulation, the assembly of the matrices does not need much more effort when compared to the function evaluations. However, a fixed decomposed Jacobian can be efficient within some Picard-iteration [17] in solving a linear system, or, more general, in using it as preconditioner within GMRES. When changing stepsizes during time integration similar remarks apply.
In the case of an hierarchical linear solver one can profit from hierarchical bypassing [8], which we will also exploit in this paper. When applying it also in the time integration, it even supports a first form of multirate time-integration [28].

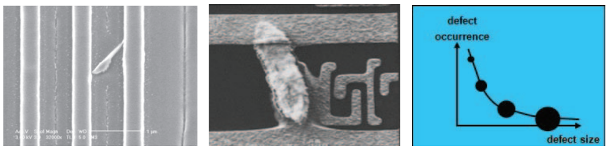


Fig. 1. Examples of disturbing bridges. Connections with small resistors values can lead to large defects to the solution.

III. Fault simulation

We first consider the effect of adding faulty, linear elements to the circuit. E.g., in [4], [26], [27] we added linear bridges (resistors) to the circuit. For each fault, only one element is added to the original, golden circuit. It may mean a new connection, while also different values are considered, see Fig. 1.

In [27] a novel time-integration was introduced where all problems were integrated over the same time step: first the fault-less, golden solution was determined at the next time step. Next, all faulty problems were integrated, see Fig. 2. Hence, effectively, a parameter loop is placed inside the time integration.

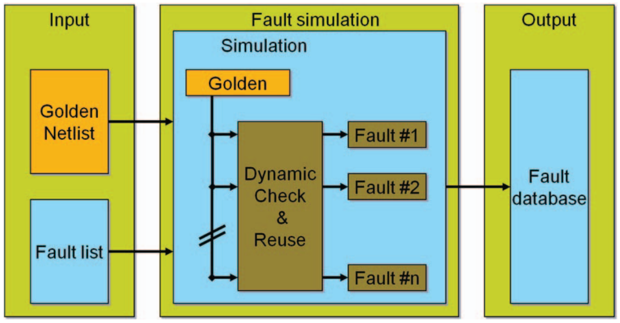


Fig. 2. At each time step, first the golden solution is determined, next all faulty problems are integrated.

The hierarchical structure was enhanced, such that the hierarchical solver could deal with all new elements: note that some new connections may violate the hierarchical structure of the golden circuit. A clever software solution to this was developed and is reported in [27]. By this, also the faulty problems could benefit from an enhanced form of hierarchical bypassing [8].
The golden solution at each new time point provides an estimate for the solution of a faulty problem (in addition to the one using extrapolation by Nordsieck vectors, see [8] and references cited there). Each faulty problem uses the stepsize of the golden solution as a maximum one. When the faulty solution really needs a stepsize that is significantly smaller than used by the golden solution, the traditional time integration is invoked, even without bypassing, until the time moment of synchronization with the golden solution. Time integration of a faulty problem is only needed until the moment when its faulty solution really differs from the golden solution. This check is done only at specfic time moments. Then the

faulty problem is marked as detectable and taken out of the list and its results are stored. Time integration is continued on the next time intervals for both the golden solution and the reduced list of faulty problems, until the next moment for checking deviations.

As in [27] we consider the faulty elements represented by adding linear conductivities to the circuit[1]. In [26] also the effect of additional linear capacitors was considered. However, the main interest is in adding linear conductivities.

$$\mathbf{j}(\mathbf{x}(t,p),p) \;=\; \mathbf{j}_0(\mathbf{x}(t,p)) + p\,\mathbf{u}\mathbf{v}^T\mathbf{x}(t,p), \tag{3}$$

where $p = 1/R$, with resistance R. Fault Analysis consists of simulations for a large number of pairs of vectors $(\mathbf{u}, \mathbf{v})$ and various values of p, and comparing the result $\mathbf{x}(t,p)$ of (1) at specific time points with the "golden" solution $\mathbf{x}(t) = \mathbf{x}(t,0),0)$ of the fault-free circuit, corresponding with $p = 0$. The golden solution $\mathbf{x}(t) = \mathbf{x}(t,0),0)$ uses $\mathbf{j}(\mathbf{x}(t,0),0) = \mathbf{j}_0(\mathbf{x}(t,0))$. If the deviation between $\mathbf{x}(t,p)$ and $\mathbf{x}(t)$ exceeds some threshold, the fault triple $(\mathbf{u},\mathbf{v},p)$, is marked as detectable and is taken out of the list.

Clearly, for each fault, we have a new contribution $p\,\mathbf{u}\mathbf{v}^T\mathbf{x}(t,p)$ as low-rank modification to the system of the golden solution, added to $\mathbf{j}_0$, see (3). Here $p = 1/R \geq 0$ is just a scalar, by which the $\mathbf{p}$-sensitivity 'matrix' $\hat{\mathbf{x}}_p(t,p) = \frac{\partial \mathbf{x}(t,p)}{\partial p}$ reduces to a vector. Let $\mathbf{x}_p^k = \mathbf{x}^k(p) \approx \mathbf{x}(t_k,p)$, $k = n, n+1$, be the numerical approximations of the faulty system and $\hat{\mathbf{x}}_p^k$ be the corresponding sensitivities. Then with $\mathbf{C}_p^k \equiv \frac{\partial \mathbf{q}(\mathbf{x}_p^k)}{\partial \mathbf{x}}$, $\mathbf{G}_p^k \equiv \frac{\partial \mathbf{j}(\mathbf{x}_p^k)}{\partial \mathbf{x}}$ (including the effect of the rank-one term with the factor p) and $\mathbf{S}_p^k \equiv \frac{\partial \mathbf{s}(\mathbf{x}_p^k, t_k)}{\partial \mathbf{x}}$, we obtain by sensitivity analysis [27]

$$\mathbf{A}_p^{n+1}\hat{\mathbf{x}}_p^{n+1} \;=\; -\mathbf{u}\mathbf{v}^T\mathbf{x}_p^{n+1} + \frac{1}{h_n}\mathbf{C}_p^n\hat{\mathbf{x}}_p^n, \;\text{ where} \tag{4}$$

$$\mathbf{A}_p^{n+1} \;=\; \frac{1}{h_n}\mathbf{C}_p^{n+1} + \mathbf{G}_p^{n+1} - \mathbf{S}_p^{n+1}. \tag{5}$$

For $p = 0$, (4) gives the limit sensitivity $\hat{\mathbf{x}}^k = \hat{\mathbf{x}}_0^k$ for the golden, fault-free solution $\mathbf{x}^k = \mathbf{x}_0^k$ $(k = n, n+1)$

$$\mathbf{A}^{n+1}\hat{\mathbf{x}}^{n+1} \;=\; -\mathbf{u}\mathbf{v}^T\mathbf{x}^{n+1} + \frac{1}{h_n}\mathbf{C}^n\hat{\mathbf{x}}^n, \;\text{ where} \tag{6}$$

$$\mathbf{A}^{n+1} \;=\; \frac{1}{h_n}\mathbf{C}^{n+1} + \mathbf{G}^{n+1} - \mathbf{S}^{n+1}, \tag{7}$$

where $\mathbf{C}^k = \mathbf{C}_0^k$ $(k = n, n+1)$, $\mathbf{G}^{n+1} = \mathbf{G}_0^{n+1}$ and $\mathbf{S}^{n+1} = \mathbf{S}_0^{n+1}$ and thus $\mathbf{A}^{n+1} = \mathbf{A}_0^{n+1}$. By Taylor expansion we additionally have

$$\mathbf{x}_p^k \;=\; \mathbf{x}^k + p\,\hat{\mathbf{x}}^k + \mathcal{O}(p^2) \quad (k = n, n+1). \tag{8}$$

The golden solution $\mathbf{x}^k$ satisfies the linearized equations of the fault-free circuit up to a term $\mathbf{R}$ that indicates the deviation from linearity (note that in (1) we did assume that $\mathbf{q}(\mathbf{0}) = \mathbf{0}$ and $\mathbf{j}(\mathbf{0}) = \mathbf{0}$)

$$\mathbf{A}^{n+1}\mathbf{x}^{n+1} \;=\; \mathbf{r}(t_{n+1}, \mathbf{x}^n, \mathbf{x}^{n+1}), \tag{9}$$

where $\mathbf{r}(t_{n+1}, \mathbf{x}^n, \mathbf{x}^{n+1}) = \mathbf{s}^{n+1} + \frac{1}{h_n}\mathbf{C}^n\mathbf{x}^n + \mathbf{R}$, in which $\mathbf{s}^{n+1} = \mathbf{s}(\mathbf{x}^{n+1}, t^{n+1})$. With (8) and (6) this gives [26], [27]

$$\begin{aligned}
&[\mathbf{A}^{n+1} + p\,\mathbf{u}\mathbf{v}^T]\mathbf{x}_p^{n+1} \\
&= \mathbf{r}(t_{n+1}, \mathbf{x}^n, \mathbf{x}^{n+1}) + \frac{1}{h_n}\mathbf{C}^n(\mathbf{x}_p^n - \mathbf{x}^n) + \mathcal{O}(p^2 + \frac{p^2}{h_n}) \quad (10)\\
&= \mathbf{r}(t_{n+1}, \mathbf{x}^n, \mathbf{x}^{n+1}) + \mathcal{O}(p^2 + \frac{p^2}{h_n} + \frac{p}{h_n}). \quad (11)
\end{aligned}$$

This invites for applying the Sherman-Morrison formula [10]. Let $\mathbf{A}\mathbf{w} = p\mathbf{u}$ and $\mathbf{A}\mathbf{y} = \frac{p}{h_n}\mathbf{C}^n\hat{\mathbf{x}}^n$. Then the predictions for $\mathbf{x}_p^{n+1}$ become

$$\mathbf{x}_p^{n+1} \;=\; \mathbf{x}^{n+1} - \frac{\mathbf{v}^T\mathbf{x}^{n+1}}{1 + \mathbf{v}^T\mathbf{w}}\mathbf{w}, \;\text{ or} \tag{12}$$

$$\mathbf{x}_p^{n+1} \;=\; (\mathbf{x}^{n+1} + \mathbf{y}) - \frac{\mathbf{v}^T(\mathbf{x}^{n+1} + \mathbf{y})}{1 + \mathbf{v}^T\mathbf{w}}\mathbf{w}. \tag{13}$$

Here (12) is based on (11), which implicitly uses arguments from sensitivity analysis. The prediction (13) is based on (10), which explicitly determines the effects of the sensitivity quantities by solving for $\mathbf{y}$. Note that (10) may be a more accurate alternative than (11). However, for simplicity, we just used (11), after ignoring the $\mathcal{O}(\cdot)$ terms at the right-hand side. For expressions when dealing with capacitors we refer to [26]. The advantage of the right-hand side in (11) is that it is independent of the solution $\mathbf{x}_p^k$ at the previous time steps. Of course, when followed by further Newton-Raphson iterations, $\mathbf{x}_p^n$ is still needed. To judge the accuracy of the linear sensitivity prediction the nonlinear solver evaluates the circuit at the sensitivity solution and updates the solution. The difference in the initial sensitivity solution and the nonlinear update is a measure for the truncation error.

If we just stick to the prediction, we may calculate the prediction of the fault at a few selected time points, which significantly reduces the work load for the fault sensitivity analysis.

Finally, we shortly describe the modeling of faulty "opens". We consider a faulty resistor, with value R, in series with another, linear resistor, with value r, where r is a correct resistor in the golden circuit. Clearly, this introduces an extra node n_e. If the golden system used $R(n_1, n_2) = r$, the faulty system uses $R(n_1, n_e) = R$, $R(n_e, n_2) = r$. The voltage at this new node n_e can be simply eliminated by noting that $v(n_e) = (r\,v(n_1) + R\,v(n_2))/(r + R)$. Doing this directly, the remaining system can be formulated as in (3) in which $p = R/(r(R + r))$. If $R \to \infty$ we obtain an "open" between the nodes n_1 and n_e and $v(n_e) \to v(n_2)$. In [27] we introduced an extra port to model bridges between models. This extra node can also become functional in providing the extra node.

For modeling a broken joint (or weld) at a node n, it is, mathematically, convenient to first split the node n into two nodes n_1 and n_2, with a simple voltage source in between for the golden circuit: $E(n_1, n_2) = 0$. Clearly this satisfies our assumption $\mathbf{j}(\mathbf{0}) = \mathbf{0}$. The faulty system uses $R(n_1, n_e) = R$,

[1] Note that for voltage-defined resistors we need two rank-one updates to describe the total contribution [8].

$E(n_e, n_2) = 0$. We assume local coordinates that correspond with $v(n_1)$, $v(n_2)$, $i(E)$. We deduce that the faulty system perturbs the golden system with $\mathbf{u}_1\mathbf{v}_1^T(R) + \mathbf{u}_2\mathbf{v}_2^T(R)$, in which, in local coordinates, $\mathbf{u}_1^T = (1, 0, 0)$, $\mathbf{v}_1^T = (-1/R, 1/R, 1)$, $\mathbf{u}_2^T = (0, 0, 1)$, $\mathbf{v}_2^T = (1, -1, R)$. This can be treated in a similar way as before.

IV. Results

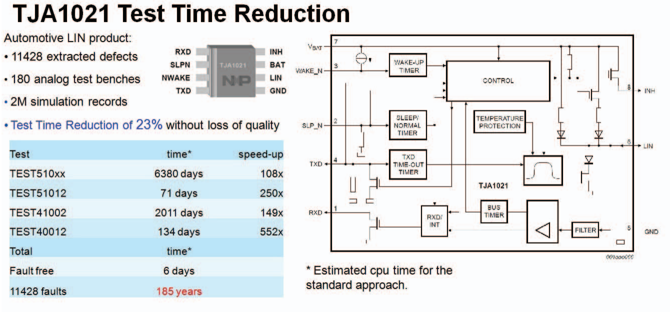
The algorithm for Fast Fault Simulation has been implemented in Pstar[2]. For fault detections in DC-simulations a significant speed-up (> 100) was obtained by exploiting bypassing and abandoning only, but inclusion of sensitivity analysis appeared essential to get significant speed-up for a broad class of problems during transient simulation. Table I shows the speed-up by including sensitivity prediction for a LIN Converter IP Block (first part), as well as for a nonlinear control DAC (second part). Clearly, the linear sensitivity estimate offers an interesting speed-up. The application of nonlinear corrections by Newton iterations reduces this effect. For the LIN Converter IP Block the effect of more iterations remains quite bounded (with 100 iterations still a speed-up of more than 10 was found, see [27]). For the nonlinear control DAC until 5 iterations a speed-up of 10 was obtained.

Analysis	**LIN Converter IP Block**		
	#iterations per step Δt	**CPU Time [sec]**	**Speed-up**
Standard AS/DOTSS	-	100437	1
Linear Sensitivity	0	458	219
Nonlinear Correction	5	2341	43
Analysis	**Control DAC**		
	#iterations per step Δt	**CPU Time [sec]**	**Speed-up**
Standard AS/DOTSS	-	52513	1
Linear Sensitivity	0	916	58
Nonlinear Correction	1	4808	11

TABLE I
SPEED-UP BY INCLUDING SENSITIVITY PREDICTION. TOP: A LIN CONVERTER IP BLOCK, #FAULTS=412. BOTTOM: A CONTROL DAC, #FAULTS=100. SEE ALSO [27].

Currently, NXP Semiconductors' Fast Fault Algorithm is the best in the world in this area [27]. It can identify locations on a chip that are probably affected by very tiny manufacturing inaccuracies and thus causing faulty behaviour at predefined time points for measurements. Fig. 3 gives further results.

Further speed-up scenarios are currently considered by initiating the fault later, see Fig. 4. If one can simply skip the initial integration of the faults until some $t_1 > 0$, for a large collection of faults no initial simulations have to be made. By this, additional orders of speed-up can be achieved. The scenarios differ in how the fault is started: suddenly, or using a smooth start-up, similar as for the source-stepping-by-transient method as described in [27]. Because of the many faults that are possible, a short start-up is a balance between efficiency and robustness. This strategy of initiating a fault in a time window shortly before a moment of measurement indicates an option for a fast pre-scan to reduce the list of faults, after which the approach described in this paper is applied for the remaining list of faults.


TJA1021 Test Time Reduction

Automotive LIN product:
- 11428 extracted defects
- 180 analog test benches
- 2M simulation records
- Test Time Reduction of 23% without loss of quality

Test	time*	speed-up
TEST510xx	6380 days	108x
TEST51012	71 days	250x
TEST41002	2011 days	149x
TEST40012	134 days	552x
Total	time*	
Fault free	6 days	
11428 faults	185 years	

* Estimated cpu time for the standard approach.

Fig. 3. Speed-up in fast fault simulation for the TJA1021 chip.

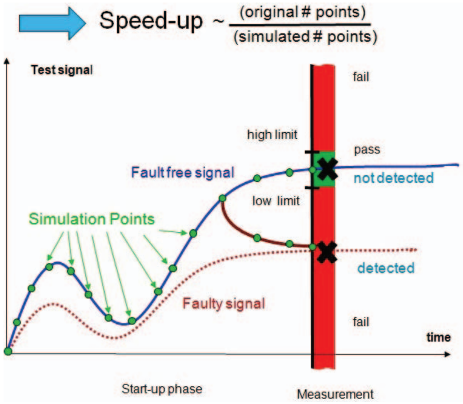


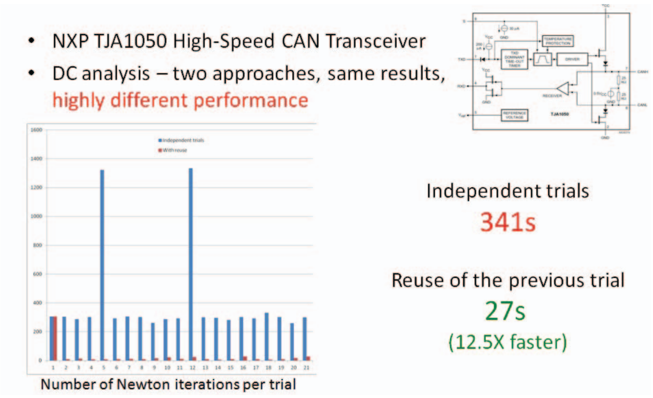
Fig. 4. Speed up potential when initiating the faults shortly before a moment of measurement and predicting the faulty solution from the golden solution.

V. Relation to Uncertainty Quantification

Interchanging the time integration loop with a parameter-sweep loop for a given pair of connection nodes, also has an interesting opportunity for statistics based on Monte Carlo simulations. Recently, at NXP, the new order of loops ("integration of ensemble") was also used for DC statistics. Just being able to exploit reuse gave a speed-up of 12.7 for the TJA1050 High-Speed CAN Transceiver, see Fig. 5.

This indicates also that Uncertainty Quantification (UQ) [14], [19], [29] can benefit from this idea. We explain two possible options. When considering Stochastic Collocation (SC) in which all L_2 inner-products in parameter space are replaced by quadrature, a list of deterministic parameter values p_k, $k = 1, \ldots, K$, is defined for which the solution $\mathbf{x}(t, p_k)$ has to be calculated. Then $\mathbf{x}(t, p) = \sum_{i=0}^{\infty} \mathbf{v}_i(t)\phi_i(p)$. In practice one limits the index i to some upper limit m. The component $\mathbf{v}_i(t)$ along ϕ_i is given by L_2 projection, which involves integration over p, By quadrature we obtain

[2] Pstar: in-house circuit simulator of NXP Semiconductors.



- NXP TJA1050 High-Speed CAN Transceiver
- DC analysis – two approaches, same results, highly different performance

Independent trials
341s

Reuse of the previous trial
27s
(12.5X faster)

Fig. 5. DC Monte Carlo simulations for a TJA1050 High-Speed CAN Transceiver. In blue the original nr of Newton iterations per independent Monte Carlo run is shown. In red we see the reduced nr of iterations due to better predictions obtained with the techniques described before. For both approaches we obtained the same results of the output quantities.

$\mathbf{v}_i(t) \doteq \sum_{k=1}^{K} w_k \mathbf{x}(t, p_k)\phi_i(p_k)$, where the quadrature formula provides an approximation. This expansion is a so-called generalized Polynomial Chaos expansion, using polynomials $\phi_i(p)$ that are orthogonal with respect to some probability density function in the parameter space for p. A library like [2] provides software when using Gauss-Hermite and Laguerre polynomials. For Fast Fault Simulation, where parameter values are positive, one may think about an exponential decay (for an infinite range, see Fig. 1; here one generates Laguerre polynomials), or a beta density distribution function with shape parameters (α, β) (when considering a finite range for p; here one generates Jacobi-polynomials). In the traditional, non-intrusive, implementation for each value p_k a complete time sweep is made. After completion of this the expansion is available and then also additional information like sensitivity. When we change the order of the loops, one can determine at the next time level the K deterministic solutions (in which one can exploit the sensitivity estimate, as described before for the Fast Fault Simulation Algorithm). This variant provides sensitivity by the expansion as soon when it is needed, Clearly, this makes SC intrusive, but this variant is probably faster than the default, non-intrusive, implementation.
An alternative approach that maintains the non-intrusive character as long as possible is to use two SC sweeps: a first sweep is applied that uses a coarse quadratue rule. This is a traditional non-intrusive implementation. It is just intended to have sensivity available for the second sweep in which a higher accurate quadrature formula is used. Now, during the time integration one has sensitivity results available from the expansion resulting as postprocessing after the first sweep and which is provided by just a call to a procedure in the UQ-Library [2]. This minimizes the implementation and make sensitivity available where it is needed. It is related to multifidelity uncertainty quantification [22], [23], [25], [30].

VI. Conclusion

We demonstrated details of a successful algorithm for Fast Fault Simulation. The speed-ups were obtained by a combination of different techniques (hierarchical simulation, bypassing, on-the-fly-reduction of the list of faults, prediction of neighbouring problems by implicit sensititivity) together with elegant enhancement of the hierarchical modeling using extra ports. Currently, NXP Semiconductors' algorithm for Fast Fault Simulation is the best in the world in this area [27]. NXP Semiconductors can identify locations on a chip that are probably affected by very tiny manufacturing inaccuracies and thus causing faulty behaviour at predefined time points for measurements. Options to obtain further speed-ups have been identified (and have been confirmed for first tests).
The algorithm indicated how we can also speed up Uncertainty Quantification by Stochastic Collocation for Polynomial Chaos expansion. This is currently further investigated.

Acknowledgment

This work is part of the FP7 Collaborative Project nanoCOPS, which is supported by the European Union in the FP7-ICT-2013-11 Programme under Grant Agreement Number 619166 (Project nanoCOPS - nanoelectronic COupled Problems Solutions). For further details see http://www.fp7-nanocops.eu/ and [11].

References

[1] A. Di Buccianico, J. ter Maten, R. Pulch, R. Janssen, J. Niehof, M. Hanssen, S. Kapora: *Robust and efficient uncertainty quantification and validation of RFIC isolation*. Radioengineering, Volume 23, Issue 1, pp. 308-318, 2014.

[2] *Dakota 6.2 - Algorithms for design exploration and simulation*, https://dakota.sandia.gov/, Sandia National Laboratories, Albuquerque, NM, USA, 2015.

[3] T.A. Davis, E.P. Natarajan: *Sparse matrix methods for circuit simulation problems*. In: B. Michielsen, J.-R. Poirier (Eds.): *Scientific Computing in Electrical Engineering SCEE 2010*, Series Mathematics in Industry, Vol. 16, Springer, pp. 3–14, 2012.

[4] D. De Jonghe, E. Maricau, G. Gielen, T. McConaghy, B. Tasić, H. Stratigopoulos: *Advances in variation-aware modeling, verification, and testing of analog ICs*. Proc. DATE 2012, pp. 1615–1620, 2012.

[5] T.S. Doorn, E.J.W. ter Maten, J.A. Croon, A. Di Bucchianico, O. Wittich: *Importance Sampling Monte Carlo simulation for accurate estimation of SRAM yield*. In: Proc. IEEE ESSCIRC'08, 34th Eur. Solid-State Circuits Conf., Edinburgh, Scotland, pp. 230–233, 2008.

[6] T.S. Doorn, J.A. Croon, E.J.W. ter Maten, A. Di Bucchianico: *A yield statistical centric design method for optimization of the SRAM active column*. In: Proc. IEEE ESSCIRC'09, 35th Eur. Solid-State Circuits Conf., Athens, Greece, pp. 352–355, 2009.

[7] V. Dubourg: *Adaptive Surrogate Models for Reliability Analysis and Reliability-Based Design Optimization*. PhD Thesis, Univ. Blaise Pascal, Clermont II, Clermont-Ferrand, France, 2011.

[8] J.G. Fijnvandraat, S.H.M.J. Houben, E.J.W. ter Maten, J.M.F. Peters: *Time domain analog circuit simulation*. J. of Comput. and Applied Maths., 185, pp. 441–459, 2006.

[9] M. Günther, U. Feldmann, J. ter Maten: *Modelling and discretization of circuit problems*. In: W.H.A. Schilders, E.J.W. ter Maten (Eds.): *Handbook of Numerical Analysis, Vol. XIII, Special Volume on Numerical methods in electromagnetics*, Elsevier BV, North-Holland, pp. 523–659, 2005.

[10] G.H. Golub, C.F. Van Loan: *Matrix Computations, Third edition*. The Johns Hopkins University Press, Baltimore, MD, 1996.

[11] R. Janssen, J. ter Maten, C. Tischendorf, H.-G. Brachtendorf, K. Bitnner, W. Schoenmaker, P. Benner, L. Feng, R. Pulch, F. Deleu, A. Wieers: *The nanoCOPS project on algorithms for nanoelectronic coupled problems solutions*. In B. Schreffler, E. Oñate, M. Papadrakakis (Eds.): *Coupled Problems in Science and Engineering VI – COUPLED PROBLEMS 2015*. Proceedings of the VI International Conference on Coupled Problems in Science and Engineering, May 18-20, 2015, San Servolo Island, Venice, Italy. Publ.: CIMNE - International Center for Numerical Methods in Engineering, Barcelona, Spain, 2015, ISBN 978-84-943928-3-2, pp. 1029–1036, 2015.

[12] B. Jourdain, J. Lelong: *Robust adaptive importance sampling for normal random vectors*. The Annals of Appl. Prob., 19.5, pp. 1687–1718, 2009.

[13] B. Lapeyre, J. Lelong: *A framework for adaptive Monte-Carlo procedures, Monte Carlo Methods and Applications*, Walter de Gruyter, 17.1, 2011, pp. 77–98.

[14] O.P. Le Maître, O.M. Knio: *Spectral methods for uncertainty quantification, with applications to computational fluid dynamics*. Springer, Science+Business Media B.V., Dordrecht, 2010.

[15] J. Li, J. Li, D. Xiu: *An efficient surrogate-based method for computing rare failure probability*. Journal of Computational Physics, 230.24, pp. 8683–8697, 2011.

[16] J. Li, D. Xiu: *Evaluation of failure probability via surrogate models*. Journal of Computational Physics, 229.23, pp. 8966–8980, 2010.

[17] Z. Li, C.-J.R. Shi: *SILCA: Spice-accurate iterative linear-centric analysis for efficient time-domain simulation of VLSI circuits with strong parasitic couplings*. IEEE Trans. on Comp.-Aided Design of Integr. Circuits and Systems (TCAD), 25-6, pp. 1087–1103, 2006.

[18] E.J.W. ter Maten, T.S. Doorn, J.A. Croon, A. Bargagli, A. Di Bucchianico, O. Wittich: *Importance Sampling for high speed statistical Monte-Carlo simulations – Designing very high yield SRAM for nanometer technologies with high variability*. Report CASA 2009-37, TU Eindhoven, 2009, http://www.win.tue.nl/analysis/reports/rana09-37.pdf.

[19] E.J.W. ter Maten, R. Pulch, W.H.A. Schilders, H.H.J.M. Janssen: *Efficient calculation of Uncertainty Quantification*. In: M. Fontes, M. Günther, N. Marheineke (Eds): *Progress in Industrial Mathematics at ECMI 2012*, Series Mathematics in Industry Vol. 19, Springer, pp. 361–370, 2014.

[20] E.J.W. ter Maten, O. Wittich, A. Di Bucchianico, T.S. Doorn, T.G.J. Beelen: *Importance sampling for determining SRAM yield and optimization with statistical constraint*. In: B. Michielsen, J.-R. Poirier (Eds.): *Scientific Computing in Electrical Engineering SCEE 2010*, Series Mathematics in Industry Vol. 16, Springer, pp. 39–48, 2012.

[21] T. McConaghy, G.G.E. Gielen: *Globally Reliable Variation-Aware Sizing of Analog Integrated Circuits via Response Surfaces and Structural Homotopy*. IEEE Trans. on Design of Integr. Circuits and Systems (TCAD), 28.11, pp. 1627–1640, 2009.

[22] A. Narayan, C. Gittelson, D. Xiu: *A stochastic collocation algorithm with multifidelity models*. SIAM J. Scientific Computing., 36-2, A495–A521, 2014.

[23] L.W.T. Ng, M.S. Eldred: *Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation*. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, 20th AI 23 - 26 April 2012, Honolulu, Hawaii, paper AIAA 2012-1852 (17p).

[24] R. Pulch: *Polynomial chaos for the computation of failure probabilities in periodic problems*. In: J. Roos, L. Costa (eds.): *Scientific Computing in Electrical Engineering SCEE 2008*. Mathematics in Industry Vol. 14, Springer, Berlin 2010, pp. 191–198.

[25] T.P. Scholcz: *Multi-fidelity methods for fluid-structure interaction and uncertainty quantification*. PhD-Thesis, Delft University of Technology, the Netherlands, 2015.

[26] B. Tasić, J.J. Dohmen, E.J.W. ter Maten, T.G.J. Beelen, H.H.J.M. Janssen, W.H.A. Schilders, M. Günther: *Fast Fault Simulation to identify subcircuits involving faulty components*. IMACM-Report 2015-10, Bergische Universitt Wuppertal, 2015. Accepted for: G. Russo, V. Capasso, G. Nicosia, V. Romano (Eds.): *Progress in Industrial Mathematics at ECMI 2014*, Mathematics in Industry, 22, 2016.

[27] B. Tasić, J.J. Dohmen, E.J.W. ter Maten, T.G.J. Beelen, W.H.A. Schilders, A. de Vries, M. van Beurden: *Robust DC and efficient time-domain fast fault simulation*. COMPEL 33:4, pp. 1161–1174, 2014.

[28] A. Verhoeven: *Redundancy reduction of IC models by multirate time integration and model order reduction*. PhD-Thesis Eindhoven University of Technology, http://alexandria.tue.nl/extra2/200712281.pdf, 2008.

[29] D. Xiu: *Numerical methods for stochastic computations - A spectral method approach*. Princeton Univ. Press, Princeton, NJ, USA, 2010.

[30] X. Zhu, A. Narayan, D. Xiu: *Computational Aspects of Stochastic Collocation with Multifidelity Models*. SIAM J. Uncertainty Quantification, 2, pp. 444–463, 2014.