Direct Statistical Simulation of Timing Properties in Sequential Circuits

Javier Rodríguez, Qin Tang, Amir Zjajo,
Michel Berkelaar, and Nick van der Meijs

Circuits and Systems Group, Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
J.RodriguezRodriguezdeGuzman@student.tudelft.nl

Abstract. Accurate timing analysis of digital integrated circuits is becoming harder to achieve with current and future CMOS technologies. The shrinking feature sizes lead to increasingly important local process variations (PV), making existing methods like corner-based static timing analysis (STA) yield overly pessimistic results. In this paper we propose a general purpose statistical circuit simulator for accurate timing analysis. A statistical simplified transistor model (SSTM) is used as the simulator's building block, allowing accurate simulation of sequential circuits while fast statistical analysis is achieved by solving a system of random differential equations (RDE), thus avoiding time-consuming Monte Carlo simulations. The conducted experiments show the accurate calculation of crossing time statistical moments for several sequential cells using 45 nm CMOS technology.

1 Introduction

CMOS technology nodes below 45 nm are currently the state of the art in the semiconductor industry. As the transistor feature sizes are continuously being reduced, process variations (PV, typically random deviations from the intended nominal values) have an increasingly significant impact on chip performance and cannot be ignored by the design and verification tools. This fact is specially critical when dealing with synchronous designs which have to meet strict timing constraints.

Circuit transient simulation is the most accurate way to check a circuit's timing compliance (also known as Dynamic Timing Analysis, DTA). However, DTA's exponential time complexity has made Static Timing Analysis (STA) [1, 2], which is a linear time complexity technique, the standard timing verification tool for the last 20 years. STA is typically used along with corner analysis, which calculates a best/worst case scenario for each parameter that may affect the circuit's performance, thus providing conservative bounds for each circuit delay. In this sense, this method can be seen as an *inter-die* PV set up, i.e. all the transistors on the same die are supposed to be affected in the same way. However, if this assumption is not true, as it is in the case of local *within-die* PV, different correlations between path delays will cause highly pessimistic estimates

or even optimistic estimates, depending on the circuit's topology [3]. Besides, as transistor features shrink with each technology node, the number of parameters and hence corners to take into account are too big to keep these kinds of methods attractive in terms of runtime.

Statistical STA (SSTA) was created to deal with STA's shortcomings by modeling the gate delay as probability distributions instead of deterministic data, which can handle adequately spatial correlations. Accurate timing estimates are expressed now, in terms of timing yield, i.e. the probability of a circuit to meet timing constraints. However, while STA algorithms require deterministic *sum* and *max/min* operations, their statistical counterparts are, in general, not trivial, specially in the presence of spatial correlations [4]. Some approximations to this problem assume normal distributions expressed in a first order canonical form. Then, *sum* operation becomes trivial while *max/min* is mainly approximated by forcing its output to be also a normal distribution in canonical form, whose coefficients are computed using a probabilistic based weighting [5] or a linear time upper bound [6].

Timing analysis tools use mainly gate-level delay models for standard cells characterization since they are the basic building blocks for most circuits and simplify their analysis. Simple LUT-based models, like the Non-Linear Delay Model (NDLM), assume a saturated input voltage ramp and capacitive load. However these assumptions do not hold for modern circuits with increasing cross-talk noise and complex resistive wire interconnections, therefore making NLDMs not an accurate timing model any more. In [7], a noise aware Current Source Model (CSM) is presented for combinational cells using a voltage controlled current source, modeled as a 2D-LUT with input and output voltages as the table indices, and a linear output capacitance. Transient simulation with this model allows accurate timing analysis for arbitrary input signals and loads. This model was extended in [8] to cope with the increasing importance of device parasitics and, later in [9], to provide statistical timing analysis by extended Monte Carlo generated LUTs to characterize the CSM's PV sensitivities.

Up to this point, all the proposed models have been focused on combinational logic cells, but none or very few of them deal with sequential cells, although these elements constitute an essential part of timing analysis, thus requiring a really accurate characterization. Clock to output delay ($T_{\text{CLK-O}}$) is computed by typical STA tools using simple NLDMs, under the assumption of stable input signals before and after the clock edge (e.g. setup and hold times). Some efforts have been done to improve this method's accuracy, like exploiting setup and hold times interdependence [10], but this approach still leads to pessimistic and inaccurate timing estimates, specially if within-die PV are present.

In the presence of arbitrarily shaped input waveforms, the main challenge of sequential cell timing analysis is to determine the conditions for the input signal to change the state of the output. For a typical latch design, this situation will take place when the capturing signal becomes inactive and its internal node has gone beyond the induced feedback loop's meta-stability point [11]. Translating this scenario to gate-level models, such as CSMs, is not trivial. Combinational

cells can be easily modeled as their input fluctuations can be accurately translated to the output node, despite the fact that certain situations, such as multiple input simultaneous switching (MISS), can lead to significant errors [12]. On the contrary, feedback loops make the output node of sequential cells independent of the input nodes at the capturing signal inactive period, losing their characteristic input/output relationship, thus needing additional control mechanisms. Besides, unintended transitions in the stored value can be also considered as a MISS event, which CSMs are not able to model correctly. A CSM for sequential cells can be found in [8] where it is shown how a combinational CSM can be extended for sequential cells using a transmission gate based latch as an example. In particular, the cell is analyzed at its different modes of operation, extracting their respective CSMs and combining them into a quite complex CSM. Finally a D flip-flop CSM is also presented by connecting two complementary latches in series.

In this paper, we propose a general purpose statistical simulation engine for digital circuits which extends the previous work presented in [13, 14] by including the analysis of sequential circuits. By using our statistical simplified transistor model (SSTM), a BSIM4-like transistor model [15], CSM's main limitations are avoided and higher accuracy is achieved at the expense of a slightly longer runtime. Additionally, since our simulation engine works at the transistor level, there is no difference in how sequential and combinational circuits are treated during the simulation. Sequential circuits are only treated in a special way when the initial guess of the DC solution is generated, due to their inherent combinational feedback loops. PV is captured by our SSTM by computing the different sensitivities to physical parameters for the selected process variables. A fast non-Monte Carlo statistical timing analysis method is used to find the statistical output arrival times by solving a system of random differential equations (RDE).

2 Simulation Engine

The typical work flow of any general purpose circuit simulator, like SPICE or SPECTRE, starts with the circuit description, as a text file listing all the discrete components within the circuit. This text file is then analyzed for correctness and translated into a mathematical representation. To perform transient analysis, the simulation engine must solve a system of linear ordinary differential equations (ODE) by successive discretization and linearization of the circuit's mathematical representation. DC analysis provides the circuit's initial operating point, thus ensuring a unique solution for the problem.

The proposed simulator, whose work flow is shown in Fig. 1, is composed of two main parts, a deterministic part and a statistical part. The deterministic part follows the simulation flow described before and, in this sense, is very similar to other SPICE-like circuit level simulators. On the other hand, the statistical part analyzes the circuit's response under PV, for which additional input data is required, similar to what a Monte Carlo loop would need. Our method calculates the sensitivities of the voltage waveforms, which are now characterized as

stochastic processes, with respect to PV by solving a system of RDEs. Finally, the stochastic waveforms are processed to determine the statistical moments of the different timing parameters of interest like the crossing times at different voltage levels.

2.1 Statistical simplified transistor model (SSTM)

The proposed circuit simulator uses a LUT-based statistical simplified transistor model (SSTM), which includes a voltage controlled current source (I_{ds}) and five non-linear parasitic capacitances (C_{gs}, C_{gd}, C_{gb}, C_{db} and C_{sb}). However, C_{db} and C_{sb} are further approximated as linear capacitors in view of their limited voltage dependence and relative small capacitance value. All these values are obtained running several DC SPECTRE simulations for both NMOS and PMOS transistors, using an accurate BSIM-4 transistor model [16].

The generated LUTs are accessed using the transistor voltages V_{gs} and V_{ds} (V_{sb} is also used for I_{ds} to take into account body biasing) as indices, in steps of 100 mV (50 mV for I_{ds} values). This fine grain voltage characterization allows us to use low order methods to compute off-grid values, such as bilinear interpolation (trilinear if body biasing) or 0-order extrapolation for out-of-bounds values.

Additional LUTs are constructed to capture the SSTM's parameters sensitivities to PV by first running the same characterization process for different values of the process parameters, like the transistor length, the oxide thickness or the threshold voltage. Finally, finite differences with respect to the each parameter nominal value are applied to obtain the sensitivities LUTs.

2.2 Circuit description and MNA system generation

A simplified SPICE-like netlist format has been defined for two different levels of abstraction (gate-level and transistor-level). The circuit is usually specified using the gate-level format, so the simulator's first step analyzes the circuit's topology and translates the gate-level format into the transistor-level using a simple parsing program.

Being aware that non-linear devices are present within the circuit, an initial guess for the DC solution is computed. Since the circuit has been described using logic gates, this initial DC guess can be easily found using a breadth-first

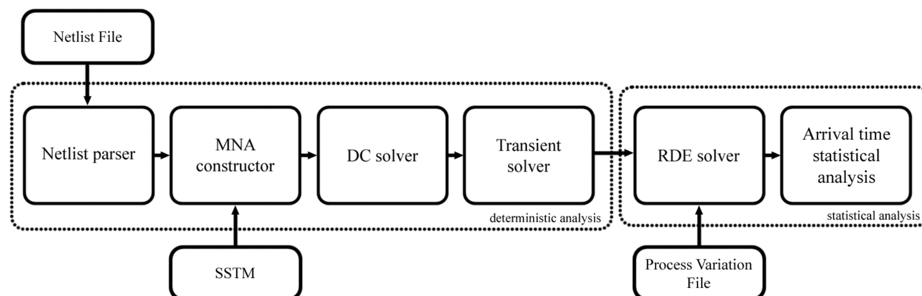

Fig. 1: Statistical simulator detailed overview

traversal algorithm along with basic boolean algebra, leaving only the gate's internal nodes as true unknowns. This algorithm works well if the circuit can be expressed as a directed acyclic graph (DAG) with logic gates and wires as vertices and edges respectively. Any combinational circuit is a clear example of these kinds of circuits. However, sequential circuits introduce combinational feedback loops, thus leading to cyclic graphs. To solve this problem, combinational feedback loops are identified and annotated during the circuit's topology analysis. This information is then used to resolve the loop's nodes initial values before the algorithm starts.

Modified nodal analysis (MNA) [17], the most common technique for systematic circuit analysis, is used by the simulator to set up the circuit's mathematical description. Here, the circuit's transistor-level description and the SSTM are combined to build the set of equations which define the circuit's behaviour, with output node voltages and device controlling branch currents as the system unknowns. Assuming inductor-less interconnection wires and output node voltages as the only system unknowns which need to be computed, the general MNA system is simplified, leading to the following system of ODEs:

$$\mathbf{C}(t)\dot{\mathbf{v}}(t) + \mathbf{G}\mathbf{v}(t) - \mathbf{i}(t) = \mathbf{0} \tag{1}$$

with $\mathbf{C}(t)$ the capacitance matrix, $\mathbf{G}$ the conductance matrix, $\mathbf{v}(t)$ and $\dot{\mathbf{v}}(t)$ the voltages and their time derivatives, and $\mathbf{i}(t)$ the input current sources.

In practice, only systems of linear ODEs can be efficiently solved by a computer program. However, the system in (1) contains non-linear elements introduced by the SSTM in $\mathbf{C}(t)$ and $\mathbf{i}(t)$. The general approach to solve such systems is to approximate every non-linear component by a linear equivalent and iterate, using a Newton-like algorithm, until the approximation error is small enough. To handle this problem, our simulator separates each of these matrices into a constant value part and a time dependent part. While the former is kept in a matrix format, the latter is stored in a transistor indexed data structure. During the DC and transient analysis the linearized system is constructed for each time instant and Newton-like iteration by extracting, from each transistor data structure, the required values of their non-linear parameters. In this sense, Jacobian matrices, needed for Taylor expansion of non-linear elements, are also stored in the same way.

2.3 DC solver and transient analysis

The MNA system in (1) is further simplified by removing the time dependent terms and, along with the initial guess found during the gate-level translation, is now used to find the circuit's exact DC solution, yielding the following system of equations in $\mathbf{V}_0 = \mathbf{v}(t_0)$:

$$\mathbf{G}\mathbf{V}_0 - \mathbf{I}_0 = \mathbf{0} \tag{2}$$

The system in (2) is still non-linear due to the transistors' current source contribution to the output nodes. Therefore, a Newton-like iterative method can be used to solve this system, provided that the initial guess is close enough to

the final solution. Although this is usually true for our computed initial guess, additional convergence strategies are used to ensure the algorithm is able to find a solution. In particular, our simulator adds large resistors to ground at every output node to deal with potentially isolated nodes due to non-linear devices and limits node voltage variations from two consecutive iterations thus solving the problem of non-convergent oscillating solutions (known as *Gmin* and *voltage damping* respectively).

The circuit's time response under nominal process conditions, $\mathbf{p_0}$, can be found by solving the MNA system described in (1) with the initial conditions obtained in (2). The resulting initial value problem can be rewritten as:

$$F(\dot{\mathbf{v}}, \mathbf{v}, t, \mathbf{p_0}) = \mathbf{0}, \text{ with } \mathbf{v}(t_o) = \mathbf{V_0} \quad (3)$$

To find the solution to this system of non-linear ODEs, an implicit linear multistep (LMS) method with variable time step (t_{step}) is used, based on a simple *predictor-corrector* method. Polynomial extrapolation is first used to calculate an initial guess for the solution at the new time instant $t_{k+1} = t_k + t_{step}$ (prediction) while a Newton-like iterative algorithm along with a the desired LMS method (*backward Euler, BE* or *trapezoidal rule, TR*), which can be chosen at the beginning of the simulation, is then used as the correction method. If convergence is achieved in a limited number of iterations, Milne's principal local truncation error (PLTE) estimate is computed for accuracy check and a new time instant and t_{step} are decided upon this. Otherwise the predicted solution is rejected and t_{step} is reduced for a new loop iteration [18].

2.4 Statistical Solver and crossing time statistical analysis

As a consequence of PV affecting the different circuit's elements, the resulting voltage waveforms become stochastic processes. Under these new conditions, Equation (3) becomes a system of non-linear RDEs:

$$F(\dot{\mathbf{v}}, \mathbf{v}, t, \mathbf{p}) = \mathbf{0}, \text{ with } \mathbf{v}(t_o) = \mathbf{V_0} + \boldsymbol{\delta}_{V_0} \quad (4)$$

In (4), $\mathbf{p}$ represents the vector of PV, expressed as $\mathbf{p} = \mathbf{p_0} + \boldsymbol{\xi}$, with $\boldsymbol{\xi}$ the vector of random deviations of the process parameters from the nominal conditions, a vector of random variables with zero mean and σ standard deviation. Finally $\boldsymbol{\delta}_{V_0}$ represents the deviation of the initial conditions due to PV.

Intuitively, the solution of (4) will be close to the deterministic solution found for (3), $\mathbf{v}_n(t)$ [13]. Therefore, if small deviations are assumed, a first order Taylor expansion around $\mathbf{v}_n(t)$ is a valid approximation of (4), resulting in a system of linear RDEs in the new variable $\mathbf{y}(t) = \mathbf{v}(t) - \mathbf{v}_n(t)$, the voltage deviation from the nominal solution:

$$\mathbf{M}(t)\dot{\mathbf{y}}(t) + \mathbf{R}(t)\mathbf{y}(t) + \mathbf{Q}(t)\boldsymbol{\xi} = \mathbf{0} \quad (5)$$

with $\mathbf{M}(t)$, $\mathbf{R}(t)$ and $\mathbf{Q}(t)$ the partial derivatives of F with respect to $\dot{\mathbf{v}}$, $\mathbf{v}$ and $\mathbf{p}$ respectively. A system like this has a unique solution in the mean square [19]:

$$\mathbf{y}(t) = \boldsymbol{\Phi}(t, t_0)\mathbf{y_0} - \int_t^{t_0} \boldsymbol{\Phi}(t, u)\mathbf{Q}(u)\boldsymbol{\xi}\, du = \boldsymbol{\alpha}(t)\boldsymbol{\xi} \quad (6)$$

with $\boldsymbol{\Phi}(t, t_0)$ the solution of the equivalent homogeneous system. As can be seen in (6) the output voltage deviations $\mathbf{y}(t)$ are proportional to $\boldsymbol{\xi}$, with $\boldsymbol{\alpha}(t)$ the sensitivities of the voltage deviation waveforms with respect to $\boldsymbol{\xi}$. Using this relationship, the equivalent system of linear ODEs in $\boldsymbol{\alpha}(t)$ is constructed from (5). The simulator can solve this system by using simplified TR integration method, and the stochastic voltage waveforms can finally be expressed as:

$$\mathbf{v}(t) = \mathbf{v}_n(t) + \mathbf{y}(t) = \mathbf{v}_n(t) + \boldsymbol{\alpha}(t)\boldsymbol{\xi} \tag{7}$$

Finally, to perform timing analysis, the circuit delay is computed as the difference between the voltage crossing times, t_η. These crossing times are defined as the time instant each signal reaches a target voltage value, usually expressed as a percentage of the supply voltage. In the presence of PV, the voltage crossing time of a signal $v(t)$ is also a random variable which can be expressed in a similar way as (7):

$$t_\eta = t_{\eta_{NOM}} + \boldsymbol{\beta}_{t_\eta}\boldsymbol{\xi}, \text{ with } \boldsymbol{\beta}_{t_\eta} = \frac{\partial t_\eta}{\partial \boldsymbol{\xi}} \tag{8}$$

For small deviations, the sensitivity vector $\boldsymbol{\beta}_{t_\eta}$ can be approximated with its value at the nominal crossing time $t_{\eta_{NOM}}$ of the signal $v(t)$ as follows [20]:

$$\frac{dv(t)}{d\boldsymbol{\xi}}\Big|_{t=t_{\eta_{NOM}}} = \Big(\frac{\partial v_n(t)}{\partial t} \times \frac{\partial t}{\partial \boldsymbol{\xi}}\Big)\Big|_{t=t_{\eta_{NOM}}} + \boldsymbol{\alpha}(t_{\eta_{NOM}}) = \mathbf{0} \tag{9a}$$

$$\boldsymbol{\beta}_{t_\eta} \approx \frac{\partial t}{\partial \boldsymbol{\xi}}\Big|_{t=t_{\eta_{NOM}}} = -\frac{\boldsymbol{\alpha}(t_{\eta_{NOM}})}{\frac{\partial v_n(t)}{\partial t}\Big|_{t=t_{\eta_{NOM}}}} \tag{9b}$$

3 Experimental results

As we pointed at the beginning of this paper, our main concern is the accurate timing analysis of sequential circuits under PV using our non-Monte Carlo statistical simulator. In order to test our simulator accuracy, three different sequential circuits, with an increasing level of complexity, were selected: *i*) a high level-active transparent latch (DLH_X1, 16 transistors); *ii*) a positive-edge master-slave D flip-flop (DFF_X1, 28 transistors); and *iii*) a custom sequential circuit (SEQ_X1, 90 transistors). The first two circuits are an obvious choice since they are the most common sequential elements used in synchronous designs. The last one, shown in Figure 2, tries to recreate a more realistic scenario with launching and catching flip-flops, combinational logic and a more elaborated wire model rather than a simple capacitor to ground, which introduces non-zero skew in the clock network. In addition to this, MISS is also present at the circuit's combinational gate since its input signals have similar delays and are captured at the same clock edge. All the circuits have been built using the Nangate 45 nm Open Cell Library as reference [16]. Figure 3 shows the internal structure of the tested D-flip-flop. Details about transistor sizing can be found at the library documentation.

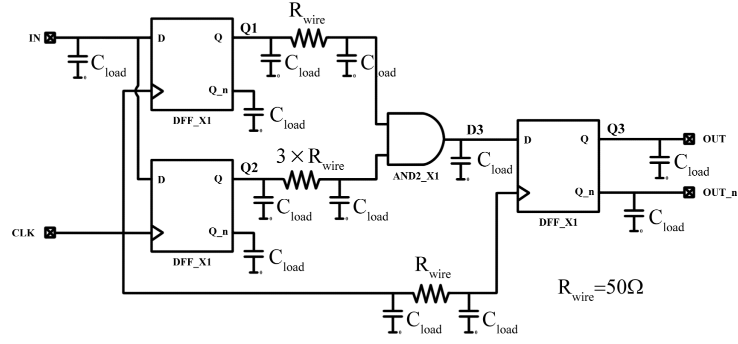

Fig. 2: Sequential test circuit (SEQ_X1)

We analyzed the proposed test circuits using our statistical simulator, implemented in MATLAB, and comparing the results against BSIM4-based SPECTRE 10 K Monte Carlo simulations. This experiment was repeated for different values of load capacitance, ranging from 5 to 25 fF, and 100 ps transition time piecewise linear functions as input signals. Transistor's length (L) and threshold voltage (V_{th}) were modeled as global non-correlated normal distributions to simulate PV with 0.5 nm and 0.04 V standard deviations from their nominal values respectively. Trapezoidal rule (TR) integration method was used to ensure the best possible accuracy.

Table 1 shows the 50% delay mean and standard deviation relative errors at the most important nodes for each test circuit nodes. From these results, we can see that the mean error is, in most of the cases, below 1% and it gets closer to zero as the load capacitor grows. The worst mean error values are found for the transparent latch DLH_X1, being a consequence of the charge non-conserving transistor model for the non-linear parasitic capacitors and it gets reduced as the linear load capacitance grows. Regarding the standard deviation, the results also show a decreasing trend with higher load values for most of the analyzed circuits although the relative error here is significantly larger. Again charge non-conserving capacitive models are the main source of error. Finally, the runtime improvement achieved with our method compared with Monte Carlo simulation, is quite significant, as can be seen in Table 2. However, the current simulation implementation has still some complexity problems dealing with large circuits mainly due to the fact that sparse matrix techniques have not been applied.

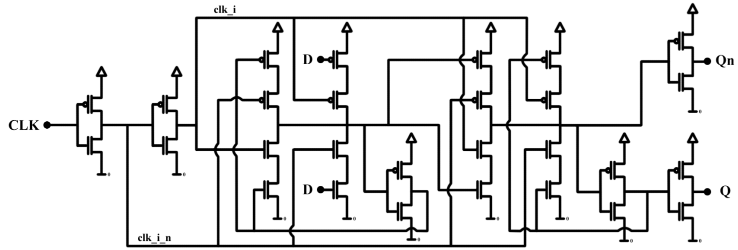

Fig. 3: D flip-flop schematic (DFF_X1)

Table 1: 50% delay statistical analysis under PV (rising input data signal)

PV	Load capacitance (fF)					Load capacitance (fF)				
	5	10	15	20	25	5	10	15	20	25
	Mean (μ) rel. error (%)					Std. deviation (σ) rel. error (%)				
DLH_X1's OUTPUT NODE (Q) $T_{CLK\text{-}O}$										
L	-3.37	-2.29	-1.66	-1.34	-1.10	-9.27	-6.46	-6.31	-4.48	-4.59
V_{th}	-3.86	-2.87	-2.26	-1.97	-1.75	-13.78	-3.16	-1.88	-1.84	-0.83
L & V_{th}	-3.85	-2.86	-2.27	-1.95	-1.75	-12.74	-5.28	-2.28	-2.76	-1.57
DFF_X1's OUTPUT NODE (Q) $T_{CLK\text{-}O}$										
L	-0.54	-0.37	-0.37	-0.25	-0.16	-7.85	-6.99	-6.68	-3.46	-4.37
V_{th}	-1.05	-0.96	-1.00	-0.89	-0.81	5.07	1.88	1.83	2.05	0.07
L & V_{th}	-1.04	-0.95	-0.49	-0.88	-0.79	-1.04	-0.49	-0.53	1.72	-0.28
SEQ_X1's 1st LAUNCHING DFF OUTPUT NODE (Q1) $T_{CLK\text{-}O}$										
L	0.08	0.04	0.03	0.02	0.02	-7.15	-5.34	-4.42	-4.10	-4.24
V_{th}	-0.61	-0.68	-0.71	-0.72	-0.72	1.35	-0.08	-0.42	-0.26	-0.36
L & V_{th}	-0.59	-0.67	-0.69	-0.70	-0.71	-1.04	-0.88	-0.81	-0.49	-0.83
SEQ_X1's CATCHING DFF INPUT NODE (D3) $T_{50\%}$										
L	0.13	0.18	0.20	0.20	0.20	-6.18	-5.86	-6.27	-6.71	-6.31
V_{th}	-0.19	-0.15	-0.13	-0.13	-0.12	-6.80	-8.13	-8.86	-9.11	-10.14
L & V_{th}	-0.18	-0.15	-0.13	-0.12	-0.12	-7.42	-8.08	-8.43	-9.03	-9.45
SEQ_X1's CATCHING DFF OUTPUT NODE (Q3) $T_{CLK\text{-}O}$										
L	-0.51	-0.48	-0.48	-0.48	-0.54	-7.45	-6.33	-5.53	-4.68	-3.40
V_{th}	-1.06	-1.14	-1.18	-1.20	-1.23	5.13	2.68	2.79	1.34	2.41
L & V_{th}	-1.05	-1.12	-1.16	-1.18	-1.20	-0.15	1.07	0.79	0.75	0.62

Table 2: Runtime comparison of the different simulation methods

Simulation method	DLH_X1	DFF_X1	SEQ_X1
MATLAB	25 secs.	45 secs.	12 mins.
SPECTRE 10 K MC	1800 secs.	2400 secs.	120 mins.

4 Conclusion

In this paper we have presented a general purpose statistical circuit simulator for accurate timing analysis, which is mandatory for state of the art integrated circuits verification where random deviations of physical parameters play a relevant role in the circuit's behaviour. A statistical simplified transistor model, instead of gate-level models, along with a fast non-Monte Carlo statistical method allow us to accurately simulate any input circuit, thus overcoming the limitations of gate-level models regarding sequential cells, as can be seen in the conducted experiments.

References

1. R. B. Hitchcock, G. L. Smith, and D. D. Cheng. "Timing Analysis of Computer Hardware". *IBM Journal of Research and Development*, pages 100–105, 1982.
2. R. B. Hitchcock. "Timing Verification and the Timing Analysis program". In *Proc. of the DAC*, pages 594–604, 1982.
3. D. Blaauw, K. Chopra, A. Srivastava, and L. Scheffer. "Statistical Timing Analysis: From Basic Principles to State the Art". *IEEE Trans. on CAD of Integrated Circuits and Systems*, pages 589–607, 2008.
4. M. Berkelaar. "Statistical Delay Calculation, a Linear Time Method". In *Proc. of TAU*, pages 15–24, 1997.
5. C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan. "First-Order Incremental Block-Based Statistical Timing Analysis". In *Proc. of the DAC*, pages 331–336, 2004.
6. A. Agarwal, D. Blaauw, and V. Zolotov. "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations". In *Proc. of the IEEE/ACM ICCAD*, pages 900–907, 2003.
7. J. F. Croix and D. F. Wong. "Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Models". In *Proc. of the DAC*, pages 386–389, 2003.
8. S. Nazarian, H. Fatemi, and M. Pedram. "Accurate Timing and Noise Analysis of Combinational and Sequential Logic Cells Using Current Source Modeling". *IEEE Trans. Very Large Scale Integrated Systems*, pages 92–103, 2011.
9. H. Fatemi, S. Nazarian, and M. Pedram. "Statistical Logic Cell Delay Analysis Using a Current-based Model". In *Proc. of the DAC*, pages 253–256, 2006.
10. E. Salman, A. Dasdan, F. Taraporevala, K. Kucukcakar, and E.G. Friedman. "Exploiting Setup–Hold-Time Interdependence in Static Timing Analysis". *IEEE Trans. on CAD of Integrated Circuits and Systems*, pages 1114–1125, 2007.
11. N. Oh, L. Ding, and A. Kasnavi. "Fast Sequential Cell Noise Immunity Characterization Using Meta-stable Point of Feedback Loop". In *Proc. of the ISQED*, pages 153–159, 2006.
12. C. Amin, C. Kashyap, N. Menezes, K. Killpack, and E. Chiprout. "A Multi-port Current Source Model for Multiple-Input Switching Effects in CMOS Library Cells". In *Proc. of the DAC*, pages 247–252, 2006.
13. Q. Tang, A. Zjajo, M. Berkelaar, and N. van der Meijs. "RDE-Based Transistor-Level Gate Simulation for Statistical Static Timing Analysis". In *Proc. of the DAC*, pages 787–792, 2010.
14. Q. Tang, A. Zjajo, M. Berkelaar, and N. van der Meijs. "Transistor-level gate model based statistical timing analysis considering correlations". In *DATE*, pages 917–922, 2012.
15. UC Berkeley. BSIM4 MOSFET Model, 2003. Available: http://www-device.eecs.berkeley.edu/bsim/?page=BSIM4.
16. Nangate 45nm Open Cell Library, 2009. Available: http://www.nangate.com/?page_id=22.
17. C. W. Ho, A. Ruehli, and P. Brennan. "The Modified Nodal Approach to Network Analysis". *IEEE Transactions on Circuits and Systems*, pages 504–509, 1975.
18. F. N. Najm. *"Circuit Simulation"*. Wiley-IEEE Press, 2010.
19. T. T. Soong. *"Random Differential Equations in Science and Engineering"*. Academic Press, New York, 1973.
20. Q. Tang. Personal communication.