Statistical Blockade Method for Fast Robustness Estimation and Compensation of Nano-CMOS Arithmetic Circuits

Luo Sun[1], Jimson Mathew[2], Dhiraj K. Pradhan[3], and Saraju P. Mohanty[4]
Department of Computer Science, University of Bristol, Bristol, UK.[1,2,3]
NanoSystem Design Laboratory (NSDL), University of North Texas, Denton, TX 76203, USA.[4]
E-mail ID: sun@compsci.bristol.ac.uk[1], jimson@compsci.bristol.ac.uk[2],
pradhan@compsci.bristol.ac.uk[3], and saraju.mohanty@unt.edu[4]

***Abstract*— The challenges for nano-CMOS based design engineers have been aggravated due to the introduction of variability into the design phase. One of the ways to understand the circuit behaviors under process variation is to analyze the rare events that may be originated due to such process variation. A method named Statistical Blockade (SB) has been proposed to estimate the rare events statistics especially for high-replication circuits. It has shown much faster speed than traditional exhaustive Monte Carlo simulation. The full Monte Carlo simulation may estimate the tolerant ability for the designs of different CMOS logic styles by estimating the statistics (e.g. mean, variance, and standard deviation) of the circuit specification. However, it is immensely computationally expensive, can be infeasible for large circuits, and may consume significant man hours in the ever shortening time-to-market. Therefore, the fast robustness comparison for different designs are performed with Intelligent Statistical Blockade (ISB) method. In the ISB method, the tail part of the whole distribution is used in estimation; thereby saving time. In this paper, the ISB method is proposed to compare arithmetic circuits designs. An adder with different logic styles is considered as an example of arithmetic circuit. The novel method with ISB shows much faster than standard Monte Carlo simulation. Furthermore, for the chosen design which is proved to be robust even in worst-case, the optimal body bias voltage is applied to improve the performance and power while reducing the variability with Adaptive Body Bias (ABB) technique.**

I. Introduction

As CMOS technology continues to scale down to achieve higher performance and higher level of integration, the impact of process variations on performance has been increasing with each semiconductor technology generation. The scaling down of technology has resulted in significant deviations from the nominal values of transistor parameters, such as channel length, threshold voltage, and gate-oxide thickness [1]. For example, variation in gate length increases from 35% in 130 nm technology to almost 60% in 65 nm technology, resulting in the large variation in leakage and performance of the designed circuit. Traditional corner model based analysis and design approaches provide guard-bands for parameter variations. Therefore, they are prone to introducing pessimism in the design [2] Electronics Design automation (EDA) tools have traditionally use for handling corner analysis, under the assumption of fixed or deterministic circuit parameters. However, in nano-CMOS circuits small variations due to inaccuracies in the manufacturing process can cause large relative variations in the behavior of the circuit. As an example, 10,000 runs of a Monte Carlo simulation for delay variation in an 8-bit adder for 20% variation in threshold voltage and gate oxide thickness is shown in Fig. 1.

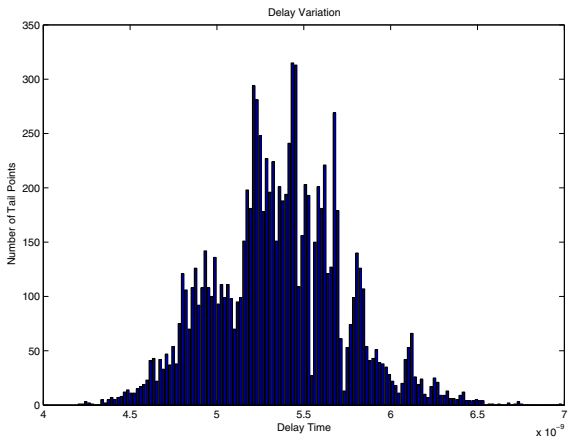

Fig. 1. 10,000 MC Simulation for Delay variation in 8-bit adder for 20% variation in gate oxide thickness and threshold voltage

Process variations which is the single most important issue nano-CMOS technology can be classified into the following two categories: (1) inter-die variations are the variations from die to die; and (2) intra-die variations correspond to variability within a single chip. Inter-die variations affect all the devices on the same chip in the same way, e.g., making the transistor gate lengths of devices on the same chip all larger or all smaller, while the intra-die variations may affect different devices differently on the same chip, e.g., making some devices have smaller transistor gate lengths and others larger transistor gate lengths [3]. Previously, the inter-die variations dominated intra-die variations, so that the latter could be safely neglected. However, in modern technologies, intra-die variations are rapidly and steadily growing and can significantly affect the

variability of performance parameters on a chip. The increase in intra-chip parameter variations is due to the effects such as microloading in the etch, variation in photoresist thickness, optical proximity effects, and steeper within-field aberrations as the manufacturing sizes approach the optical resolution limit. Intra-die variation is spatially correlated. It is locally layout dependent and circuit specific, i.e., devices with similar layout patterns and proximity structures tend to have similar characteristics. It is globally location dependent, i.e., devices located close to each other are more likely to have the similar characteristics than those placed far away [3].

The **novel contributions of this paper** are as follows:

1) A detailed analysis of Statistical Blockade [4] as a Monte Carlo technique is presented.
2) A novel method is proposed for estimating the robust ability of different Nano-CMOS Arithmetic Circuits. This method is combined with an implementation of Statistical Blockade to achieve significant reduction in the computational costs.
3) After picking up the design with worst tolerant ability under process variation, a method is used to compensate the yield loss of the design with Adaptive Body Bias (ABB) technique.

The rest of this paper is organized as follows. Section II describes the related previous work. In Section III, some of the basic definitions and notations of Statistical Blockade method [4] is presented. Section IV explores the full adder designs in different logic styles. In Section V, solutions for the problems is proposed. Section VI and Section VII shows the basic simulation approach and the experimental results then conclusions are drawn in Section VIII.

II. Related Prior Research

Recently developed statistical static timing analysis (SSTA) tools have recognized these unavoidably random aspects of manufacturing variations and have attempted to account for them using simple models that are computationally inexpensive. [5] uses a linear model for the gate delay as a function of varying gate parameters. [3] and [6] recognize the impact of spatial correlation between gates and use simple grid-based models to represent this correlation. Also use Principal Components Analysis (PCA) to extract uncorrelated parameters from this correlation model, and builds the gate timing models using these uncorrelated components. Several analytical and semi-analytical approaches have been suggested to model the behavior of SRAM cells and digital circuits in the presence of process variations. All suffer from approximations necessary to make the problem tractable Monte Carlo analysis (MC) is one of the standard techniques used for statistical modeling. Standard Monte Carlo techniques are, by construction, most efficient at sampling the statistically likely cases. When used for simulating statistically unlikely or rare events, these techniques are extremely slow. In [4] authors proposed Statistical Blockade (SB) as a Monte Carlo technique that allows us to efficiently filter to block unwanted samples insufficiently rare in the tail distributions. The method imposes almost no a priori limitations on the form of the statistics for the process parameters, device models, or performance metrics. The key observation behind Statistical Blockade is that generating each sample is not expensive: we are merely creating the parameters for a circuit.

III. Why Statistical Blockade?

The SB approach is used to efficiently generate samples in the tail of the distribution of the performance metric of a circuit [4]. Standard MC is not suitable as it generates samples that follow the complete distribution. The main idea is to define a region in the parameter space that yields circuit performance values greater than threshold t. Then only those MC samples that lie in the tail region will be simulated, blocking the body samples. For example, if the tail threshold is the 99% point of the distribution then only one out of 100 simulations is useful, which results in an immediate speedup of $100\times$ over standard MC. A small MC sample set (1000 points) is used to train a classifier to identify the tail points. However, it is different to classify all the tail points to build the accurate tail model for high-dimensional data. Thus, a relaxed boundary of the classifier, denote as classification threshold t_c, is applied to block most unwanted points. [7] suggests using t_c as the 97^{th} percentile to extract 1% tail samples (i.e. t=99%), based on empirical analysis of the tradeoff between classifier accuracy, simulation time, and tail model fit. From the extreme value theory [8], an important conclusion can be exploited that the conditional distribution of the events in the tail region trend toward a generalized Pareto distribution (GPD). So the extracted tail points can be fitted GPD to build the tail model by calculating the coefficients of GPD, shape parameter α and scale parameter β. The tail model is the cumulative distribution function (CDF) of the GPD.

SB filtering and GPD model building is then accomplished as follows [9]:

1) Perform initial sampling to generate data to build a classifier. This initial sampling can be standard MC or importance sampling. Also estimate t and $t_c < t$ from this data.
2) Build a classifier using the classification threshold t_c.
3) Generate more samples using MC, following the CDF F, but simulate only those that are classified as tail points. Update the estimate of t.
4) Fit GPD model to the simulated tail points.

However, there are significant practical problems with the original technique. Extensions to make SB practically usable in common scenarios is proposed [10]. The classification threshold for different tail regions and the minimum training points count depending on classification threshold is presented [11]. It offers both fastest speed of simulation and highest accuracy for SB. Statistical Behavioral Circuit Blocks (SBCB) is proposed in [12] which is combined with an implementation of SB to achieve significant reduction in the computational costs on asynchronous circuits.

IV. THE EXAMPLE ARITHMETIC CIRCUIT: FULL ADDER FOR ALTERNATIVE LOGIC STYLES

There are varieties of static CMOS logic styles which have been proposed to implement 1-bit full adder cells [13]. These designs can be mainly divided into two categories: complementary CMOS logic and pass transistor logic circuits.

A complementary CMOS FA (C-CMOS) cell is shown in Fig. 2(a). It is based on the regular CMOS structure with PMOS pull-up and NMOS pull-down. The advantage of C-CMOS FA is its robustness against the supply voltage scaling and transistors sizing. The FA cell in Fig. 2(b) is the complementary pass transistor logic full adder cell (CPL). The difference between pass transistor logic and C-CMOS logic is the source of the pass transistor connects to the input signal instead of connecting to the power lines. A single pass transistor can implement the logic function resulting in smaller count transistors and smaller input load. However, the pass transistor logic meets a threshold voltage drop problem. Thus, the inverts have to be employed to ensure the drivability. Fig. 2(c) shows a transmission-function full adder (TFA) by using 16 transistors. Fig. 2(d) is a 20 transistors transmission gate full adder (TGA). Both of them belong to pass transistor logic circuit. This type design employs PMOS and NMOS in parallel and the signal can pass though the transistors when they are on simultaneously. Therefore, this design has nothing to do with the voltage drop problem. Moreover, less number transistors designs have been proposed as well, for example 10T full adder as shown in Fig. 2(e).

V. THE INTELLIGENT STATISTICAL BLOCKADE APPROACH

In this section, an intelligent method with SB is proposed to estimate tolerant ability of different FA cells based 16-bit ripple-carry adders (RCAs). The algorithm steps are shown in Algorithm 1:

Algorithm 1 Intelligent Statistical Blockage for Robustness Estimation.

$Step.1$ Compute the delay times of 5 RCAs with their nominal values for a single run. Make the delay times to the target value through transistor sizing.

$Step.2$ Use SB method to pick up 1% tail points from $10,000$ generated MC points and simulate the extracted tail points.

$Step.3$ Calculate the coefficients α and β for Generalized Pareto Distribution (GPD) approximation $G_{\alpha,\beta}$ with the extracted true tail points to the Conditional Cumulative Distribution (CDF).

$Step.4$ Compare the CDFs of 5 RCAs for the robust abilities estimation.

In the 1^{st} step of proposed algorithm, it shows how to adjust the delay time using transistor sizing approach:

1) Set all the transistors (NMOS and PMOS) to the minimum size. ($45\ nm$ for channel length and $90\ nm$ for channel width in $45\ nm$ BPTM)
2) Simulate 5 16-bit RCA designs for a single run with nominal values of variables to measure the initial delays. Set a target delay time according to the measured nominal delays.
3) Figure the transition with the highest delay (from 1^{st} bit carry in to 16^{th} bit carry out) and mark the transistors in that are involved.
4) Change the channel width of the transistors which are involved in the critical path to adjust the 5 initial delay times to the target value.

The channel widths of the marked transistors in the critical are listed in the Table I for 5 RCAs.

TABLE I
CHANNEL WIDTH OF TRANSISTORS INVOLVED IN CRITICAL PATH AFTER TRANSISTOR SIZING.

RCA	C-CMOS	CPL	TFA	TGA	10T
Channel Width (nm)	83.2	90	239.3	206.3	178.9

In the 2^{nd} step, the Support Vector Machine (SVM) classifier presented in [11] is considered as the classification tool in SB implementation. To achieve the fastest simulation speed and highest accuracy, the algorithm proposed in [11] is applied to find out the minimal classification threshold t_c and the optimal number of training samples n. The SB method with t_c and n is used to extract 1% tail points from the generalized MC points. Then the true tail points can be selected after the simulation of extracted tail points.

In Step.3, the maximum likelihood estimation (MLE) is used to fit GPD to the 1% tail data to build the CDF tail model. For fitting the GPD model, two GDP coefficients α and β need to be computed by MATLAB function. After building the tail models, the comparison can be done in Step.4 and the detailed analysis will be described in the following section.

VI. EXPERIMENTAL EVALUATION OF THE PROPOSED METHOD

In this section, the details of the experiments and the experimental results are presented. The test platform [14] is shown in Fig. 3 for the delay measurement of 5 RCAs.

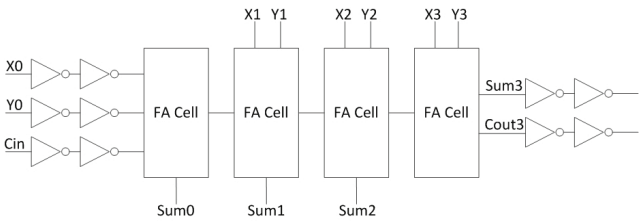

Fig. 3. Test Bench used to perform the experimental evaluations.

The shown 4-bit ripple-carry adder in Fig. 3 is cascaded by 4 FA cells, with the carry output of the current full adder connecting to the next bit full adder input in the chain. It starts from X_0 and Y_0 which represent the least significant bits of the numbers to be added and the output is $sum0$, $sum1$, $sum2$ and $sum3$. The 16-bit RCA is based on the same principle.

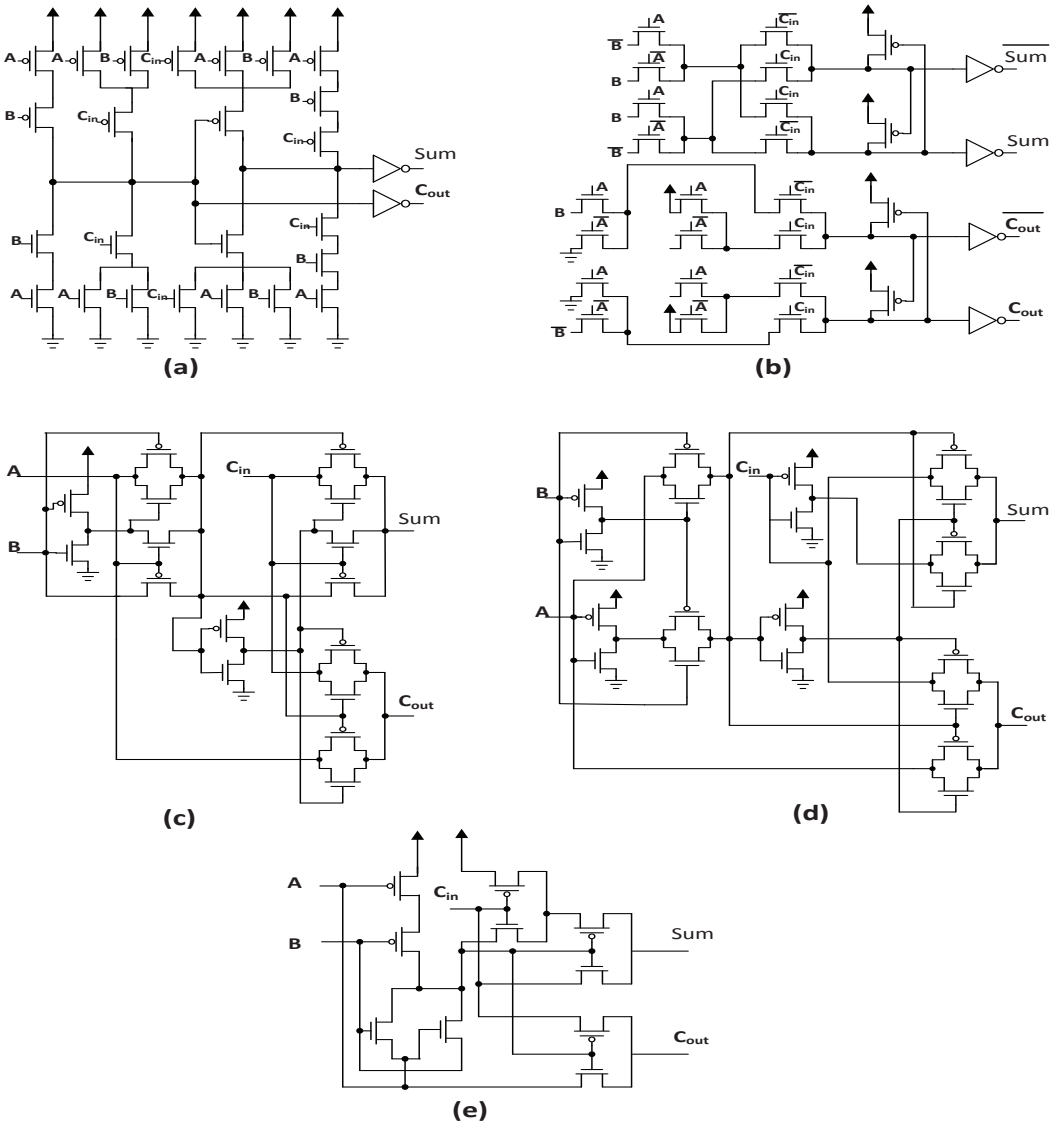

Fig. 2. Full Adder cells of different logic styles. (a) C-CMOS (b) CPL (c) TFA (d) TGA (e) 10T.

The buffers which consist of two cascaded inverters are connected to the input signals and outputs. The inputs are fed from buffers to give more realistic signals. The outputs are loaded with buffers to provide proper loading conditions to ensure the fairness of the comparison. The attached buffers for the outputs can be also used to solve the threshold voltage drop problem caused by the pass transistor logic, thus, to enhance the driven capability.

During the Monte Carlo simulation, the variables to be varied are threshold voltage (Vth) and gate oxide thickness (Tox) for NMOS and PMOS transistors. Thus, there are 4 variables involved in the simulation. The output metric to be measured is the delay time (Td) of the 16-bit RCA. In each run, the delay time is measured from 50% of input voltage swing (carry in of 1^{st} bit) to 50% of output voltage swing (carry out of 16^{th} bit). The variations of the variable parameters are considered to be Gaussian distributed with 10% variation and a deviation of $\pm 3\sigma$ around their nominal values.

In SB based estimation, 1% tail points have been extracted with SB method from 10,000 generated MC points. Fig. 4 is the distribution of the delay time for 1% true tail points of 16-bit C-CMOS RCA. As shown in the distribution, the threshold of delay for 1% is 7.2 ns. i.e., for 1% tail, the threshold of 99% delays is 7.2 ns. It also can be seen clearly that the tail points are no longer Gaussian distributed. The extracted tail data are required to fit of GPD, since GPD is a highly left skewed distribution and especially suitable for the tail data fitting.

After the simulation of extracted tail points, the true ones will be selected and used to compute the coefficients α and

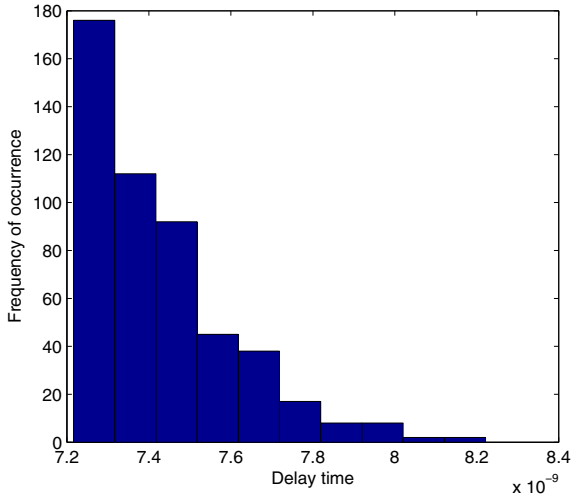

Fig. 4. Delay distribution of 1% tail for 16-bit C-CMOS.

β of GPD to build the CDF tail model. The coefficients are calculated by the MATLAB function. Fig. 5 compares the conditional CDFs of 1% tail points plotted by empirical method and SB method. It shows a good match between two CDFs. The value of horizontal axis is the exceedance of delay, i.e., the values of the delays for 1% tail points. For the tail points which delay time is 7.4 ns, the exceedance is 0.2 ns. The vertical axis is the predicted possibilities for the tail points from the model.

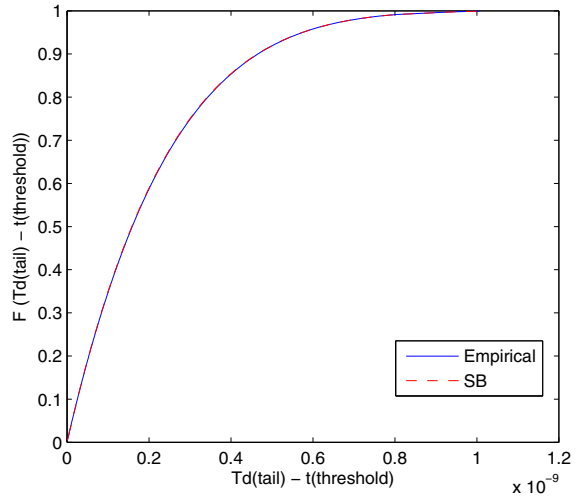

Fig. 5. CDFs for tail points: Empirical Method vs. SB approach.

The comparison of CDFs between 16-bit C-CMOS and 16-bit 10T RCA is shown in Fig. 6. The tail model with full curve is C-CMOS FA based 16-bit RCA and the dashed curve is 10T based. As shown in the diagram, for the tail points which exceedance of delay time is less than or equal to 0.2 ns, the possibilities are 60% for C-CMOS RCA and 43% for 10T. i.e., for the tail points which exceedance is over 0.2 ns, their possibilities are 40% and 57% respectively. Thus, the delay distribution of 10T RCA has a longer tail than C-CMOS and the distribution of C-CMOS is more tighten. Obviously, for a given delay threshold, the 10T RCAs yield loss due to violation of timing requirement is higher than C-CMOS. Then, we can say the C-CMOS based 16-bit RCA is robust better than 10T based RCA.

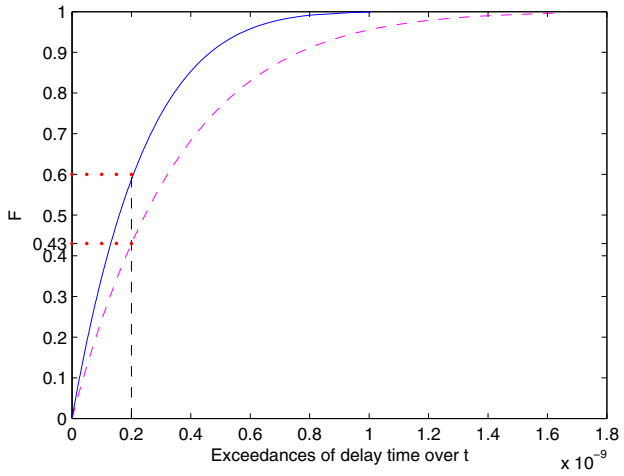

Fig. 6. Comparison of tail model between 16-bit C-CMOS and 10T.

The SB method is applied to pick up 1% tail points to build the CDF tail models for 5 considered RCAs. Fig. 7 shows the comparison of 5 tail models. Through the analysis of the tail models, the sequence of tolerant ability is: C-CMOS $>$ TFA $>$ TGA $>$ 10T $>$ CPL.

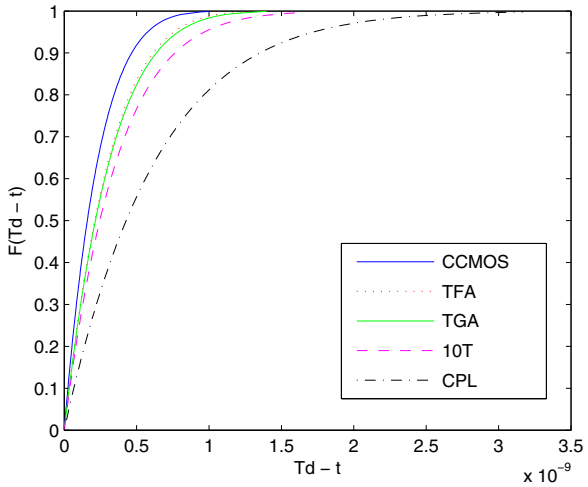

Fig. 7. The tail models (CDFs) for 5 RCAs.

To prove the correctness of SB method based estimation, full MC simulation based evaluations have been done by computing the standard deviation and variance of the simulated 10,000 MC data for each RCA design.

TABLE II

STANDARD DEVIATION AND VARIANCE FOR 5 RCAS

RCA	C-CMOS	CPL	TFA	TGA	10T
Standard Deviation	3.6002 e-010	7.7530 e-010	4.6593 e-010	5.7400 e-010	6.6522 e-010
Variance	1.2962 e-019	6.0109 e-019	2.1709 e-019	3.2947 e-019	4.4251 e-019

Table II lists the standard deviations and variances for the 5 designs. As can be seen from the table, the result is the same with the SB method based estimation. The proposed SB based estimation method shows $10\times$ speedup compared with full MC simulation for the robust ability estimation cause it only requires approximate 1000 simulations (training samples n + 1% extracted tail points) for each RCA design.

VII. ADAPTIVE BODY BIAS (ABB) FOR YIELD COMPENSATION

Through the comparison of robust ability for different designs, the design which is robust worst has been picked up. How to reduce the variability and improve performance subject to a power constraint, and thus, improve the yield loss for a chosen design?

Adaptive Body Bias (ABB) is used to compensate for parameter variation, which reduce variability and improve performance and power consumption. The bias for PMOS transistor trends to be forward biased to improve the circuit performance with the threshold voltage increasing. While the bias for NMOS transistor is reverse biased to reduce the leakage current, thus reduce the power consumption with the threshold voltage decreasing. However, ABB has its own drawback: it requires an additional on-power distribution networks for body voltage, and thus, occupy additional silicon area.

In above experiments, CPL RCA is proved to be the robust worst. Thus, ABB is applied to improve its tolerant ability under process variation and also enhance the circuit performance. All the bulks of PMOS transistors in 16-bit CPL PCA are connected to the body bias voltage V_p and all the NMOS bulks are connected to V_n. In this paper, the optimum combination for V_p and V_n has been selected to achieve the best performance and power consumption for a trade off and improve its robustness. [15] suggests the range of the body bias voltage being $\pm 20\%$ of the nominal supply voltage (1.1V) mainly due to the fact that we would like to provide enough noise margin for body bias voltage signal in order not to allow the possibility of crossing the forward threshold of the transistor junctions by the body bias voltage. The exhaustive combinations for V_p and V_n has been simulated, then the optimum pair has been found out as V_p = 1.3V and V_n = -0.2V. As shown in the Table III, the circuit performance and power consumption has been improved with less variability when optimal body bias voltage has been applied.

TABLE III

COMPARISON BETWEEN WITH AND WITHOUT ABB FOR CIRCUIT PERFORMANCE AND POWER CONSUMPTION

V_p=1.3V V_n=-0.2V	Mean (Delay)	Std (Delay)	Var (Delay)	Mean (Power)	Std (Power)	Var (Power)
Without ABB	6.5909 e-0009	4.6618 e-010	2.1732 e-019	7.1623 e-005	4.6007 e-006	2.1167 e-011
With ABB	6.3544 e-0009	4.6255 e-010	2.1395 e-019	6.9692 e-005	4.3726 e-006	1.9128 e-011

VIII. CONCLUSIONS

Statistical Blockade [4] is an efficient method for rare events analysis. In this paper, a novel method combined with SB has been proposed to fast evaluate the tolerant ability under process variation for different designs. It can make computation time much shorter compared with full MC simulation method. What's more, the circuit delay and power consumption as well as its variability has been reduced for the chosen design using ABB technique by applying optimum body bias voltage. However, it may not significantly improve the yield loss of the products since there is still a distribution tail which may not meet the accepted requirement. Thus, ABB technique can be explored further to compensate the yield loss.

REFERENCES

[1] S. P. Mohanty, "Unified Challenges in Nano-CMOS High-Level Synthesis," in *Proc. 22nd International Conf. VLSI Design*, 2009, pp. 531–531.

[2] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*, New York: Springer, 2005.

[3] H. Chang and S. Sapatnekar, "Statistical timing analysis under spatial correlations," in *Proc. DAC*, Sept. 2005, pp. 1467 – 1482.

[4] A. Singhee and R. A. Rutenbar, "Statistical blockade: a novel method for very fast Monte Carlo simulation of rare circuit events, and its application," in *Proc. DATE*, 2007, pp. 1379–1384.

[5] C. Visweswariah, et al., "First-order incremental block-based statistical timing analysis," in *Proc. DAC*, 2004, pp. 331–336.

[6] V. Khandelwal and A. Srivastava, "A general framework for accurate statistical timing analysis considering correlations," in *Proc. DAC*, 2005, pp. 89–94.

[7] A. Singhee and R. A. Rutenbar, "Statistical blockade: Very fast statistical simulation and modeling of rare circuit events, and its application to memory design," in *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 8, pp. 1176–1189, Aug. 2009.

[8] A. A. Balkema and L. de Haan, "Residual life time at great age," in *Ann. Prob.*, vol. 2, no. 5, pp. 792–804, 1974.

[9] J. Wang, A. Singhee and R. A. Rutenbar, "Two fast methods for estimating the minimum standby supply voltage for large SRAMs," in *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 29, no. 12, pp. 1908–1920, Dec. 2010.

[10] A. Singhee, et al., "Recursive statistical blockade: An enhanced technique for rare event simulation with application to SRAM circuit design," *Proc. International Conf. VLSI Design*, pp. 131–136, 2008.

[11] L. Sun, J. Mathew, D. K. Pradhan and S. P. M. Mohanty, "Algorithms for rare event analysis in nano-CMOS circuits using statistical blockade," in *Proc. International SoC Design Conference*, pp. 162–165, 2010.

[12] Z. Xie and D. Edwards, "Computation reduction for statistical analysis of the effect of nano-CMOS variability on asynchronous circuits," *Design and Diagnostics of Electronic Circuits and Systems*, pp. 161–166, Apr. 2010.

[13] C. Chang, J. Gu and M. Zhang, "A Review of 0.18-μm Full Adder Performances for Tree Structured Arithmetic Circuits," *IEEE Trans. Very Large Scale Integration System*, vol. 13, no. 6, pp. 686–695, June 2005.

[14] A. M. Shams, T. K. Darwish and M. A. Bayoumi, "Performance analysis of low-power 1-bit CMOS full adder cells," *IEEE Trans. Very Large Scale Integration System*, vol. 10, no. 1, pp. 686–695, Feb. 2002.

[15] J. W. Tschanz, J. T. Kao, S. G. Narendra, R. Nair, D. A. Antoniadis, A. P. Chandrakasan, and V. De, "Adaptive Body Bias for Reducing Impacts of Die-to-Die and Within-Die Parameter Variations on Microprocessor Frequency and Leakage," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, pp. 686–695, Nov. 2002.