To appear in *IEEE Trans. on Computer Aided Design*, 2013.

Design and Multi-Corner Optimization of the Energy-Delay Product of CMOS Flip-Flops under the NBTI Effect

Hamed Abrishami, Safar Hatami, and Massoud Pedram, *Fellow, IEEE*

Abstract— **With the CMOS transistors being scaled to 28nm and lower, Negative Bias Temperature Instability (NBTI) has become a major concern due to its impact on PMOS transistor aging process and the corresponding reduction in the long-term reliability of CMOS circuits. This paper investigates the effect of NBTI phenomenon on the setup and hold times of CMOS flip-flops. First, it is shown that the NBTI effect tightens the setup and hold timing constraints imposed on the flip-flops in the design. Second, an efficient algorithm is introduced for characterizing codependent setup and hold time contours of the flip-flops. Third, a multi-corner optimization technique, which relies on mathematical programming to find the best transistor sizes, is presented to minimize the energy-delay product of the flip-flops under the NBTI effect. Finally, the proposed optimization technique is applied to True Single-Phase Clock (TSPC) flip-flops to demonstrate its effectiveness.**

Index Terms—**Circuit reliability, Flip-flop, Multi-corner.**

I. INTRODUCTION

As CMOS transistors are scaled toward ultra deep submicron technologies, circuit reliability cannot be ignored. Device aging processes such as the Negative Bias Temperature Instability (NBTI) can have a huge impact on the circuit performance over time. Indeed the NBTI effect has proven to be an increasing threat to the circuit reliability in nanometer scale technology. Due to this effect, the threshold voltage of the PMOS transistors increases over time, resulting in reduced switching speeds for logic gates, and corresponding degradation in circuit performance, and hence, increased probability of circuit failure due to timing constraint violations [1][2].

The NBTI effect is created by trap generation at the Si/SiO2 interface in PMOS transistors under the negative bias condition (VGS = –VDD) at elevated temperatures and degrades the device driving current. The interaction of inversion layer holes with hydrogen passivated Si atoms can break the Si-H bonds, creating an interface trap and one H atom that can diffuse away from the interface or can anneal an existing trap [1]. However, with time, these Si-H bonds can easily break during operation (i.e., ON-state, negative gate bias for the PMOS.) The broken bonds act as interfacial traps and increase the threshold voltage of the device, thus affecting the performance of the integrated circuit. NBTI impact gets more severe in scaled technology due to higher die temperatures and utilization of ultra thin gate oxide [6]. Even using new technologies like FinFET, NBTI is still really important and there are many studies which have been done [31][32] or in the progress especially in device community.

The effect of NBTI on digital CMOS circuit performance has been methodically studied in [1][3]. Recently, a number of techniques have been proposed to alleviate the NBTI-induced degradation of the CMOS circuit performance with time. These techniques may generally be classified in two categories vis-à-vis design time and runtime techniques.

Design time techniques are focused on addressing the NBTI issues during the design stages. They can be performed during the synthesis or physical design optimization steps as explained next. The authors in [4] pre-characterized the effect of NBTI for each gate in a cell library based on input signal probabilities, and subsequently, exploited this knowledge for technology-dependent mapping considering signal probability of each node when searching for the best matching gate. In [5] pin reordering and logic restructuring techniques were investigated. During pin reordering, the arrival time of each input was considered and an exhaustive search was performed to find the smallest output arrival time taking into account the NBTI effect. In [6], it was shown that the speed degradation of the CMOS circuit may be offset by cell-level up-sizing during the initial design stage to compensate for the NBTI-induced decrease in speed of the PMOS device during its lifetime. The authors of [7] proposed the use of soft-edge flip-flops that in turn allow compensating for the delay increase in the combinational logic by introducing a transparency window for the signal launching and receiving flip-flops. In [8] a method for designing an NBTI-aware SRAM was presented. The key idea was to regularly (e.g., daily) invert the data stored in SRAM cells to maximize the NBTI recovery effect.

Runtime techniques compensate for the NBTI effect

during circuit operation in the field. For example, the authors in [9] introduced an input vector control approach to put the circuit in NBTI recovery phase during the standby mode of circuit operation. This method was similar to minimum leakage vector technique [10]. In [11] an adaptive body biasing technique was used to decrease the threshold voltage during circuit operation so as to alleviate the increase caused by NBTI. In [12] the operating voltage was gradually increased during the circuit lifetime in order to compensate for the performance degradation due to NBTI.

Although these works addressed the NBTI effect on circuit performance, they did not consider the effect of NBTI on the setup/hold time characteristics of the sequential circuit elements (i.e., CMOS latches and flip-flops.) More recently researchers have begun to investigate the effect of NBTI on the timing characteristics of the flip-flops. In [13] it was claimed that the setup and hold time of the flip-flops remain nearly constant with or without the NBTI effect. In [14], the effect of NBTI on different low power and high performance flip-flops was studied; however, no solution was offered to alleviate the problem. The authors of [15] introduced an ad-hoc selective transistor-level sizing to combat the NBTI effect without considering energy consumption as part of the objective.

Operating frequencies of more than 1 GHz are common in modern integrated circuits. As the clock period decreases, inaccuracy in setup/hold times caused by corner-based static timing analysis (STA) tools becomes less acceptable. Optimism in setup/hold time calculation can result in circuit failure, while pessimism leads to inferior performance [16]. Therefore, accurate characterization of the setup and hold times of latches and registers is critically important for timing analysis of digital circuits [17]. Setup and hold times are co-dependent in the sense that there are multiple pairs of setup and hold times that result same clock-to-q [16]. All pairs of setup/hold times that correspond to a constant clock-to-q delay are placed on a contour of clock-to-q delay surface. Salman et al. in [16] presented a methodology to co-dependently characterize the setup and hold times of sequential circuit elements (SCE's) and used the resulting multiple pairs in STA. An Euler-Newton curve tracing procedure was proposed in [17] and [18] to efficiently characterize the setup and hold times codependency. The codependent setup/hold contours are utilized to evaluate setup and hold slacks.

In addition, VLSI circuits may be operated at different voltage corners (i.e., it may be instantiated in different voltage islands in the design or, more notably, it may be subjected to different voltage levels due to employment of *dynamic voltage scaling* techniques in modern low power VLSI designs.) Therefore, there are multiple supply voltage levels at which VLSI circuits are desired to work correctly and power optimally. Moreover, they are required to be tolerant to the temperature variations due to the environment changes and heat generation in the circuit. Hence, there should be a multi-criterion optimization to reach the best solution in terms of different design aspects like sizing, placement, routing and etc. for different corners of operations.

A *multi-criterion optimization* (MCO) problem is the optimization of different objective functions of the same variable vector simultaneously and reaching to a solution that is best in regard to all of the objective functions. A multi-criterion optimization issue can come into sight in different levels of VLSI circuit optimization. During the design, placement and layout, and synthesis circuit designers wish to have a circuit optimized in speed, power consumption, and area in different corners of operation. The optimization of a circuit for speed and power is nearly always challenging considering satisfying multiple of different constraints in various corners. If this optimization is done on the transistor or gate sizing of the circuit, finding the best sizing vector corresponding to both speed and power is challenging. This problem can also emerge in a circuit working under different supply voltages or clock frequencies, or even environmental temperatures. Like the power and speed case, multi-criterion optimization gets its significance when optimization of different objective functions conflicts with each other.

The remainder of the paper is organized as follows. Section II provides some background on the NBTI effect and flip-flop characterization. It also defines the terminology which will be used in subsequent sections. The effect of process variation on NBTI is studied in section III. Section IV introduces Multi-criterion optimization techniques. The algorithm to extract the Codependent Setup/Hold Time (CSHT) contour is proposed in section V. The effect of NBTI on CSHT characterization is described in section VI. The problem formulation and mathematical program are introduced in section VII. Section VIII gives the simulation results and Section IX concludes the paper.

II. BACKGROUND

This section provides the terminology, reviews the manifestation of NBTI on the threshold voltage of a PMOS transistor, defines the CSHT contour for a given clock-to-q delay, and explains how to utilize this contour in a STA tool for timing verification.

A. The NBTI Effect

Aggressive scaling of CMOS technology makes NBTI one of the dominant reliability concerns in nanoscale designs [20]. It is believed that NBTI is caused by broken Si-H bonds, which are induced by positive holes from the channel. Next H, in a neutral form, diffuses away; positive traps are left, which cause the increase of voltage threshold of the PMOS transistors [21].

For a PMOS transistor, there are two phases of NBTI, depending on its bias condition. In phase I, when V_G=0 (i.e., V_{GS}= $-V_{DD}$), positive interface traps accumulate during the stress time with H atoms diffusing towards the gate. This

phase is usually referred to as the "stress" or "static NBTI". In phase II, when $V_G=V_{DD}$ (i.e., $V_{GS}=0$), holes are not present in the channel, and thus, no new interface traps are generated; instead, H atoms diffuse back and anneal the broken Si-H. As a result, the number of interface traps is reduced and some of the NBTI adverse effect is reversed. Phase II is referred to as "recovery" and can have a significant impact on the NBTI effect estimation in VLSI circuits. The stress and recovery phases together are called the "dynamic NBTI".

The NBTI effect on the threshold voltage is highly dependent on the temperature. More precisely, the threshold voltage under NBTI degrades severely in high temperatures. The huge impact of temperature is shown in section VIII through simulations. In addition, the NBTI effect also depends on the oxide thickness (technology node dependency), the duty cycle of stress vs. recovery phases, the supply voltage level, and the voltage value of the signal applied to the gate of PMOS transistor [3].

In this paper, we consider the circuit under the dynamic NBTI model in order to capture realistic circuit operation. There are some analytical models to express the change in V_{th} under the dynamic NBTI model, e.g., [1][3][21]. In this paper in order to predict the threshold voltage degradation due to the NBTI effect at a time t and also considering the duty cycle of stress vs. recovery phases, we adopt the dynamic NBTI model of reference [3].

In recent years, there are several approaches proposed to model the degradation and recovery processes, which can be classified as reaction-diffusion (R-D) theory based models [3] and hole trapping models [33]. There are also combined models, which try to characterize the NBTI effects based on the combination of the above models [33] [34]. The debate is still going on about the right approach to model the NBTI effect while the reality may be a mixture of R-D and T-D. In other words, with extension and appropriate combination with TD, the RD model is expected to be valid for 45nm and beyond technologies.

B. Codependent Setup and Hold Times

Latches and flip-flops are sequential circuit elements used in synchronous designs where a clock edge is used to sample and store a logic value on a data line. The *setup time*, τ_s, is the minimum time before the active edge of the clock that the input data line must be valid for reliable latching. Similarly, the *hold time*, τ_h, represents the minimum time that the data input must be held stable after the active clock edge. The active clock edge is the transition edge (either low-to-high or high-to-low) at which data transfer/latching occurs. The *clock-to-q delay* refers to the propagation delay from the 50% V_{DD} transition point of the active clock edge to the 50% V_{DD} transition point of the output, q, of the latch/register. The *setup skew* refers to the delay from the latest 50% transition edge of the data signal to the 50% active clock transition edge; similarly, the *hold skew* denotes the delay from the 50% active clock transition edge to the earliest 50% transition edge of the data signal. Fig. 1 illustrates the setup and hold skews, which are denoted by τ_{sw} and τ_{hw}, respectively.

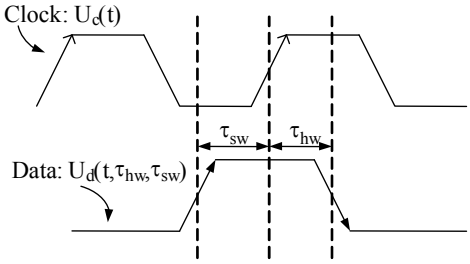

Fig. 1. Setup and hold skews shown on the data and clock waveforms.

A common technique for setup/hold time characterization is to plot the clock-to-q delay for various setup and hold skews via a series of transient simulations. This process in turn produces a *clock-to-q delay surface*. The setup (hold) time is then taken as a particular setup (hold) skew point on the plot, for which the *characteristic clock-to-q*[1], t_{cc2q}, delay increases by say 10%. (We shall denote as t_{c2q} the clock-to-q delay which is 10% higher than t_{cc2q}.) The setup (hold) time is typically made more accurate by identifying an interval around the initial estimate of the setup (hold) time and running transient simulations in that interval according to a binary search method.

As already noted, the setup and hold times are not independent quantities, but depend strongly on one another. Typically, the setup time decreases as the hold skew increases and vice versa. Similarly, the hold time decreases as the setup skew increases and vice versa. The tradeoff between setup and hold skews and the hold and setup times is a strong function of the flip-flop design.

A general method to extract codependent pairs of setup/hold times is to first obtain the clock-to-q surface. This is followed by extraction of a contour in the setup/hold time plane that contains all points that result in a given increase (*e.g.*, 10% is typical) in t_{cc2q}. Fig. 2 (a) and (b) show a typical clock-to-q surface and a CSHT contour plot. Fig. 2 (c) depicts that setup and hold time pairs decrease when clock-to-q increases.

[1] If the setup skew is larger than a certain value, then the clock-to-q delay of a flip-flop will become independent of the setup skew; this constant clock-to-q delay which is achieved for large setup skews is called the "characteristic clock-to-output delay" of the flip-flop.

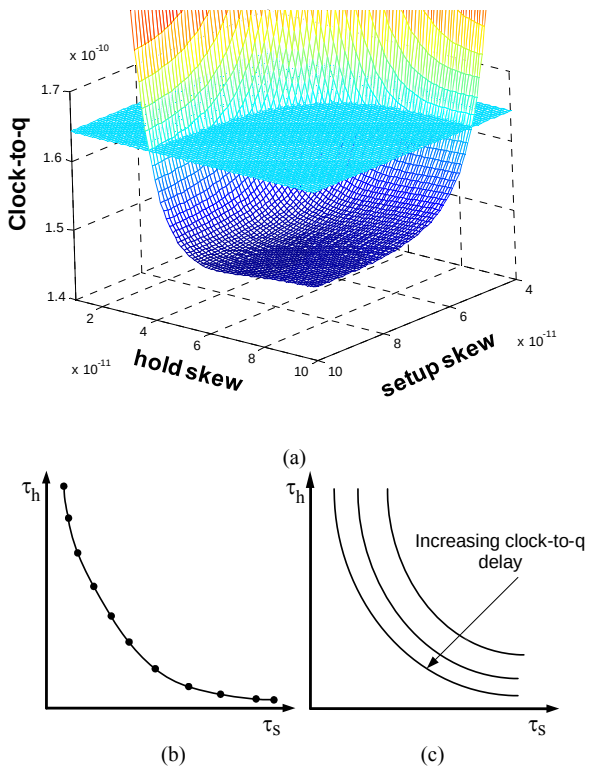

Fig. 2. (a) A clock-to-q surface, (b) A setup/hold time contour, (c) setup/hold time contours with different clock-to-q values.

C. Setup and Hold Slacks and Required Times

In general, a STA tool reads in a circuit netlist, a cell library, and a clock period T [16]. The tool reports whether new data values can be introduced in a (pipelined) circuit every T seconds. This analysis is accomplished by computing the worst setup slack (s_s) and the worst hold slack (s_h) for any flip-flop in the circuit. Referring to Fig. 3, these slacks are computed as follows:

$$s_s \equiv \min(\tau_{sw}) - \tau_s = T + \min(D_{p2}) - t_{c2q} - \max(D_{p1} + D_c) - \tau_s \quad (1)$$

$$s_h \equiv \min(\tau_{hw}) - \tau_h = t_{c2q} + \min(D_{p1} + D_c) - \max(D_{p2}) - \tau_h \quad (2)$$

where D_{p1}, D_{p2}, and D_c stand for the delays of local clock signals compared to the global clock, and delay of the combinational logic encased between the input and output flip-flops, respectively as illustrated in Fig. 3.

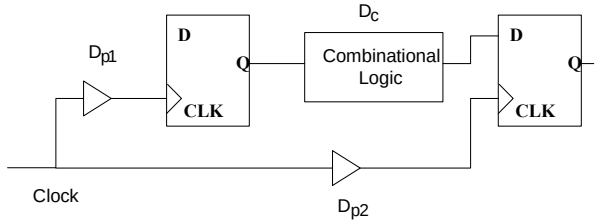

Fig. 3. Definition of s_s and s_h in a synchronous data path.

If a slack is negative, it is said to be "violated". If a setup slack, s_s, is violated, the circuit can operate correctly by increasing T. If a hold slack, s_h, is negative, the circuit will not function correctly unless delay elements are inserted on the short paths in the combinational logic.

The *required setup time* (RST) for a given flip-flop is defined as the minimum value of τ_{sw} for the flip-flop that results in a non-negative setup slack (i.e., the minimum setup skew needed to eliminate setup time violations for the flip-flop.) The *required hold time* (RHT) is defined similarly. On the other hand, the area above the CSHT contour is a pessimistic area where the flip-flop can correctly work in while the area under the CSHT contour is an overly optimistic area. Optimism is not permissible in STA, because it may result in failing chips. Therefore, the feasible working area for the flip-flop is the area above the CSHT contour. In addition, RST and RHT constraints must be satisfied. Hence, the flip-flop should be designed in a way to work in the shaded region in Fig. 4, which is called the *feasible region* (FR.)

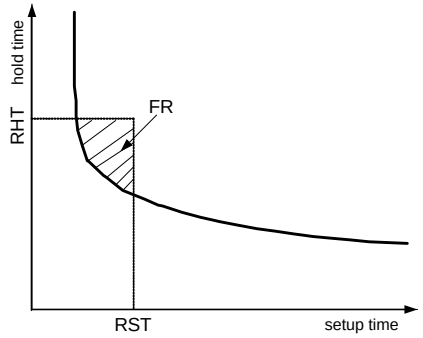

Fig. 4. RST, RHT and FR in CSHT contour.

III. Statistical Variation and NBTI Degradation

One of the key factors in today's VLSI circuits is the manufacturing-induced process variations. There have been numerous studies about process variations, which are mostly about variation in threshold voltage and channel length but there have not been many about the effect of process variations on the NBTI phenomenon. Similar to the random dopant fluctuation (RDF) effect, which causes variation in the threshold voltage of intra die transistors, there is a source of process variation for NBTI in very short channel devices [24].

In very short channel devices, the number of Si-H bonds in the device is rather small, ranging from tens to hundreds of pairs depending on the specific technology [25]. Hence, the breaking and re-passivation of Si-H bonds can be a source of statistical variation; i.e., during the fabrication, the number of Si-H bonds can be different from one transistor to another one in the same die. This variation adds another random variation on top of the common degradation of threshold voltage due to NBTI.

The author of [24] modeled the number of broken bonds N_{it} in the channel as a Poisson random variable and based on the formulation introduced in [25]:

$$\sigma_{N_{it}}^2 = \mu_{N_{it}} = \frac{C_{ox}\mu_{\Delta V_{th,NBTI}}}{q} = \frac{\epsilon_{ox}A_G\mu_{\Delta V_{th,NBTI}}}{qt_{ox}} \quad (3)$$

where σ_{Nit} and μ_{Nit} represent the mean and standard deviation of N_{it}, respectively. A_G is the effective channel area.

Based on (3) and formulations in [25]

$$\sigma_{\Delta V_{th,NBTI}} \propto \frac{t^{1/12}}{A_G} \quad (4)$$

On the other hand the nominal threshold voltage degradation due to NBTI is proportional to $t^{1/6}$.

So, if we consider the threshold voltage variation due to RDF and the NBTI effects to be statistically independent, the total V_{th} variation will be:

$$\sigma_{V_{th}} = \sqrt{\sigma_{RDF}^2 + \sigma_{\Delta V_{th,NBTI}}^2(t)} \quad (5)$$

IV. Multi-criterion Optimization Problem

Multi-Criterion Optimization (MOC) problem can be formulated as follows:

$$\begin{aligned} &minimize\ \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_k(\boldsymbol{x})\} \quad k \geq 2 \\ &subject\ to\ \boldsymbol{x} \in S \end{aligned} \quad (6)$$

where f_i is an objective function and $f_i: R^n \to R$. $\boldsymbol{x} = (x_1, x_2, \dots, x_n)^T$ is called the decision vector that the optimization is done on it. $S \subset R^n$ is the feasible region that is determined by the constraints on the MCO PROBLEM.

The goal is to minimize all objective functions simultaneously. We assume that there is no single solution that is optimal with respect to every objective function, that is, the optimum single-objective solutions are at least partially conflicting with one another. Moreover, the objective functions can also be incommensurable, i.e., they may be expressed in very different units (μw of power dissipation vs. ns of delay.)

A. Multi-criterion Optimization Solution Methods

A MCO problem is usually solved by scalarization methods, that is, various objective functions are combined to produce a single objective function to be optimized [27]. Since there are many such combining functions, each having its optimum solution, solving the MCO problem results in a set of optimum solutions. If the MCO problem is convex, then every locally optimal solution is also global optimal solution.

Moving from a solution (also known as a decision vector) to another solution requires some insight into the problem structure and user preferences. Consequently, a decision maker with good insight to the problem is needed in order to decide which optimal decision vector to use. A function $U: R^n \to R$ that represents the preference of the decision maker among all the objective function is called a "Value Function". In the MCO problem, the value function is (at least implicitly) known [28].

Based on the role of the decision maker during different phases of the MCO problem solving process, solution methods of MCO problem are divided into four classes. In the no-preference methods, the decision maker is not used at all. In posteriori methods, the decision maker will choose the desired solution among a set of derived optimal solutions only in the end. In priori methods, the preference and opinion of the decision maker is considered before solving the MCO problem. In interactive methods, the decision maker is involved in every step of the optimization process, and makes decisions based on the available information throughout this process [28]. There are several methods for solving a MCO problem; here we review two proven methods that are of particular use for our specific MCO problem formulation. We will consider a function minimization problem from here; the case of a maximization problem is similar.

B. Pareto Optimal Set

First we precisely define the notion of a Pareto-Optimal set. If the given objective functions are conflicting, then there will not exist a single solution that simultaneously minimizes all the objective functions. We are thus looking for a non-dominated solution in the sense that if we try to minimize one of the objective functions any further, the other objective function value(s) will go up. This kind of optimality is called Pareto optimality [28].

Definition 1: A decision vector $\boldsymbol{x}^* \in S$ is Pareto optimal if there is no other decision vector $\boldsymbol{x} \in S$ such that $f_i(\boldsymbol{x}) \leq f_i(\boldsymbol{x}^*)$ for all $i = 1, \dots, k$ and $f_j(\boldsymbol{x}) < f_j(\boldsymbol{x}^*)$ for at least one index j.

Mathematically a MCO problem is solved when the Pareto optimal set is reached.

1) Weighted Sum Method

In the Weighted Sum (WS) method the weighted sum of the objective functions is minimized. The problem can be formulated as follows:

$$\begin{aligned} &minimize \sum_{i=1}^{k} w_i f_i(x) \\ &subject\ to\ x \in S \\ &w_i \geq 0\ \ and\ \sum_{i=1}^{k} w_i = 1 \end{aligned} \quad (7)$$

w_i is the weight corresponds to objective function f_i. w_i's are positive real numbers and are normalized. By perturbing the weights in the WS method we can find the Pareto Surface although some solution may be missed in non-convex functions [28]. WS is categorized in posteriori methods.

2) Compromise Programming Method

In this paper we also use a priori method called Compromise Programming (CP) method for our MCO circuit optimization problem. In this method the distance between some reference point and the feasible objective region is minimized. Consider k objective functions $\{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_k(\boldsymbol{x})\}$ to be optimized simultaneously. We assign the design references of $\{f_1^*, f_2^*, \dots, f_k^*\}$ for the set of objective functions. These values can be equal to the

minimum of each objective function. The problem is formulated as follows:

$$minimize \left(\sum_{i=1}^{k} w_i |f_i(x) - f_i^*|^p\right)^{1/p} \quad (8)$$
$$subject\ to\ x \in S$$

The vector of w specifies how close an objective function needs to get to its reference value. In particular, some objective functions may be deemed as more important than others, and hence, the designer wants them to be more optimized compared to the others. Therefore, by assigning a larger w_i coefficient to them they will be forced to get closer to their reference values. Note that in reality the weighting vector specifies the direction of the search toward the optimum solution in the feasible region.

This method is one of the simplest and most straightforward MCO solution methods, yet it is very efficient and quite robust, especially when the weighting vector is adaptively changed. In the circuit optimization problem, the designer can determine the optimum value (reference point) of each function (delay, power, etc.) a priori.

The preference of the decision maker is determined by the weights and the value of the references. If these values are chosen appropriately, the Pareto optimal solution can be obtained from equation (8). However, it is sometimes difficult to determine these weights. Moreover, the solution cannot be better than the aforesaid references, even though they are conservatively set (to be lower bounds on the lowest values of each objective function.) Note that the desirable solution can be obtained by adjusting the weight, and there is no positive correlation between the weight w_i and the corresponding objective function [29].

V. CSHT Characterization

As stated before, the conventional method of extracting the CSHT contour requires a series of transient simulations to generate the t_{c2q} surface, which is not efficient when we must compute many contours. The authors in [18] proposed a method that numerically extracts the contour. In this section another efficient algorithm is proposed to tackle this problem, which is more than two times faster than the algorithm proposed in [23]. We use Fig. 5 to explain the proposed algorithm.

Definition 2: The finite difference slope, α, of contour $\Gamma(\tau_s) = \tau_h$ at point $B = (\tau_s^B, \tau_h^B)$ is defined as: $\alpha^B = \frac{\tau_h^B - \tau_h^A}{\Delta\tau_s}$ where point A is a previously calculated point on Γ such that $\Delta\tau_s = \tau_s^B - \tau_s^A$. The superscript B in α^B denotes the point at which the finite difference slope α is calculated. Note that in Fig. 5 $\Delta\tau_s$ and slope are negative.

In the proposed algorithm (see below), we seek out the setup/hold pairs from two different directions, D_s and D_h as shown in Fig. 5. The search in the right-to-left direction, D_s starts from the largest setup time, τ_s^{large}, i.e., point *X*, and ends at point *M* (Margin point) where $1 \le |\alpha^M| \le 2$. There is a need for the stop point since linear approximation is used to find the hold time of next point but after point M, the linear approximation is not valid anymore (the search point is in the curve of the contour). Note that for a given setup time, we look for its corresponding hold time in a hold time interval whose length is proportional to α calculated at previous setup/hold time pair. When the search gets close to point M, this interval needs to be increased to reach to the right hold time value which makes the search really slow. The slope at a given point *B* is used to guess the next point $C = (\tau_s^C, \tau_h^C)$ on Γ as follows:

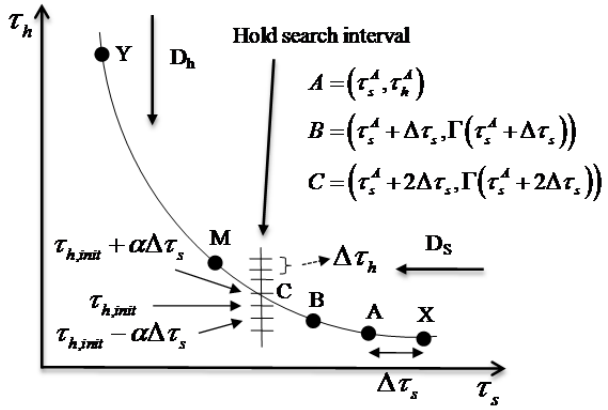

Fig. 5. A setup/ hold time contour for given clock-to-q delay.

$$\tau_s^C = \tau_s^B + \Delta\tau_s\,, \tau_h^C = \tau_h^B + \alpha^B \Delta\tau_s \quad (9)$$

The bounds of the search interval for hold time centered at point *C* is also given by $\tau_h^C \pm \alpha^B \Delta\tau_s$.

The top-down search, D_h, starts from the largest hold time, τ_h^{large}, i.e., point Y, and again ends at point *M*. For the D_h search, for the given hold time, we look for the corresponding setup time in a setup time interval whose length is proportional to the $1/\alpha$ value calculated for the previous setup/hold time pair.

The proposed BES-algorithm is in practice 10 times more efficient than the conventional method (i.e., finding the clock-to-q surface and extracting the contour).

We next describe a backward Euler search (BES) algorithm to efficiently calculate the setup/hold time points for D_s and D_h. Let $\Delta\tau_s$ denote the setup time step resolution that the user intends to have for the CSHT characterization. The BES algorithm for D_s direction is as follows:

BES-Algorithm (D_s, t_{cc2q}, $\Delta\tau_s$, τ_s^{Large})

i. Find t_{cc2q} for the flip-flop by doing a transient simulation with large setup and hold skews. Initialize i=1 and τ_s^i to the largest setup time for which we want to calculate the corresponding hold time. A good guess for the largest value of setup time is half of the clock period. Next sweep the hold skew values and determine the hold time, τ_h^i. To find the second initial point (e.g. point A), we decrease the setup time of the first initial point (e.g. point X) by $\Delta\tau_s$ and again sweep the hold skew values and determine the hold time. These two initial points are the starting points for step ii of the algorithm.

ii. Calculate slope α^i at (τ_s^i, τ_h^i) from *Definition* 2. Notice $\alpha^1 = 0$ because Γ is asymptotic to a constant hold time value when $\tau_s \to \infty$.

iii. Set $\tau_s^{i+1} = \tau_s^i + \Delta\tau_s$ and calculate the first guess for the hold time by using backward Euler (BE) method as follows (see Fig. 5):

$$\tau_{h,init}^i = \tau_h^i + \alpha^i \Delta\tau_s \quad (10)$$

Sweep the hold skew values in the range of $\tau_{h,init}^{i+1} \pm \alpha^i \Delta\tau_s$ with time step $\Delta\tau_h^i = \alpha^i \Delta\tau_s$ (hold time step resolution) and find the hold time τ_h^{i+1} ; i.e., the value of hold skew which results in a clock-to-q delay equal to $1.1\times t_{cc2q}$.

iv. Repeat steps ii-iii for $i \geq 2$ till $|\alpha| \leq 2$ to compute setup/hold pairs on the contour.

To compute all the points of the contour, *BES-Algorithm* (D_s, t_{cc2q}, $\Delta\tau_s$, τ_s^{large}) and *BES-Algorithm*(D_h, t_{cc2q}, $\Delta\tau_h$, τ_h^{large}) are evaluated. For the latter one, D_h, means that all '*s*' subscripts are replaced by '*h*' and vice versa in the body of *BES-Algorithm*. Some setup/hold time points of contour for the interval $1 \leq |\alpha| \leq 2$ are calculated twice (by both D_s and D_h) which can be replaced by their average. For example, two points $P_1 = (\tau_s^{P1}, \tau_h^{P1})$ and $P_2 = (\tau_s^{P2}, \tau_h^{P2})$ can be replaced by $\overline{P} = (0.5(\tau_s^{P1} + \tau_s^{P2}), 0.5(\tau_h^{P1} + \tau_h^{P2}))$.

VI. Effect of the NBTI on the CSHT Contour

Increasing the threshold voltage of PMOS transistors, due to the NBTI effect, results in variation in the CSHT characteristics. This means that for the same t_{c2q}, a new set of setup/hold time pairs should be obtained (cf. Fig. 6 for a pictorial explanation.) On the other hand, due to the NBTI effect, delay of combinational circuits itself increases. Therefore, given a fixed clock frequency, RST and RHT values will change and new STA requirements should be specified to achieve timing closure. By using NBTI-aware design techniques like [6], the delay of combinational logic blocks and clock drivers can be kept relatively unchanged. Notice that it is possible to extend our methodology to handle changes in the RST and RHT values.

With the NBTI effect, a timing failure occurs when the new CSHT contour has no intersection with the FR. This means there is no setup and hold time pairs that result in non-negative setup and hold slacks. Fig. 6 illustrates the effect of NBTI on the CSHT for the timing failure and non-failure cases.

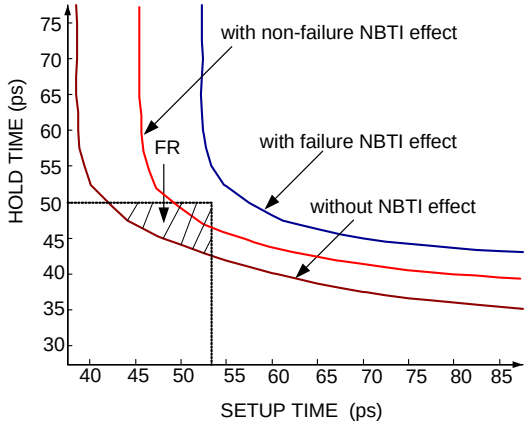

Fig. 6. Setup/hold time codependency change due to the NBTI effect.

VII. NBTI-Aware Flip-Flop Design

The variation in CSHT contour due to the NBTI effect can cause a timing failure in the circuit. To overcome this failure the flip-flop must be designed in a way so as not to violate the timing constraints after aging. Recall that timing characteristics of flip-flops depend on the sizing of the transistors in their circuit realizations. Hence, we present a sizing technique for designing flip-flops to alleviate the aging problem. We also consider minimizing the energy consumption of the circuit. More precisely, the NBTI effect causes increase in the t_{c2q} as well as a right upward shift of the CSHT contour. To compensate for the aging effect, we size transistors in the flip-flop circuit so as to shift the (new) CSHT contour below and to the left of the (original) CSHT contour. The new CSHT contour will gradually move and approach (but not overtake) the original CSHT contour due to the NBTI effect.

A. Deterministic Problem Formulation

In this problem formulation we replace RST and RHT with maximum allowed changes in the setup and hold times of the flip-flop, respectively. These upper bounds should not be violated even after the NBTI-induced aging effect.

The objective of our optimization is to minimize the fresh state (i.e., at the beginning of circuit deployment and use) value of the energy-delay product of a flip-flop by imposing maximum degradation limits on the timing characteristics of the aged flip-flop. Constraints thus include upper bounds for changes in the t_{c2q}, setup, and hold times of the flip-flop due to the NBTI effect over a specific period of time. The solution of the optimization problem determines the transistor sizes in the flip-flop under consideration.

In addition, the target flip-flop may be operated at different voltage corners (i.e., it may be instantiated in different voltage islands in the design or, more notably, it may be subjected to different voltage levels due to engagement of *dynamic voltage scaling* techniques in modern low power VLSI designs.) Therefore, there are multiple supply voltage levels at which the flip-flop is

desired to work correctly and energy-delay optimally. First, we introduce the optimization problem formulation for a single voltage corner and then extend it to multiple voltage corners.

1) Single corner optimization

The mathematical programming problem formulation for single corner operation may be stated as follows:

$$\begin{aligned} &\textit{minimize} \quad Q(\vec{w}) = E^{fr}(\vec{w}).D^{fr}(\vec{w}) = \\ &\qquad E^{fr}(\vec{w}).\left(t_{c2q}^{fr}(\vec{w}) + \tau_s^{fr}(\vec{w})\right) \\ &\textit{subject to} \quad t_{c2q}^{aged}(\vec{w}) \le t_{c2q,max} \\ &\qquad \tau_s^{aged}(\vec{w}) \le \tau_{s,max} \\ &\qquad \tau_h^{aged}(\vec{w}) \le \tau_{h,max} \end{aligned} \tag{11}$$

where $t_{c2q,max}$, $\tau_{s,max}$ and $\tau_{h,max}$ are maximum allowed values of t_{c2q}, setup and hold times, respectively and *fr* means the fresh state and *aged* means after aging effect happened for the specific period of time, e.g., three years. A sizing vector $\vec{w}$ refers to set of transistors' sizes. Notice that the delay contribution of the launching flip-flop and the receiving flip-flop to the worst-case delay of the circuit is equal to t_{c2q} plus τ_s. Also note that instead of minimizing the energy-delay product of a fresh flip-flop circuit, we could have modeled and minimized the energy-delay product of a middle-aged circuit.

We refer to the solution of the optimization problem (11) as $\vec{w}^*$. We point out that for each sizing vector $\vec{w}$, there is one specific contour in the fresh state and one in the aged state since the timing characteristics of flip-flop change when the sizes of the transistors change.

2) Multi-corner Optimization

In the case of multiple voltage corners of operation, it is desired to simultaneously minimize all objective functions, Q_i, where i=1,…,m. Q_i refers to the objective function of the ith corner. Associated with each corner i, there is a weight r_i which indicates the importance of the corner i in the multi-corner optimization, where

$$\sum_{i=1}^{m} r_i = 1 \tag{12}$$

Definition 3: Suppose the optimum solution for corner i is $\vec{w_i}^*$ and $Q_i{}^* = Q_i(\vec{w_i}^*)$ is the *best objective value* at corner i. Clearly the objective function vector $Q^* = \{Q_1^*(\vec{w}), \dots, Q_m^*(\vec{w})\}$ is a lower bound (possibly infeasible) on the Pareto optimal set of solutions to the multi-criteria optimization problem. The *worst objective value* Q_i^{**} is defined as the $Max\{Q_i(\vec{w_j}^*)\}$ where maximization is over j = 1,…,m. Clearly $Q^{**} = \{Q_1^{**}(\vec{w}), \dots, Q_m^{**}(\vec{w})\}$ is an upper bound on the Pareto optimal set of solutions to the multi-criteria optimization problem.

Multi-corner-opt algorithm:

i. Solve the optimization problem (11) for each corner separately to obtain $\vec{w_i}^*$'s, $Q_i{}^*$, and Q_i^{**}.

ii. Solve the following nonlinear optimization problem

$$\begin{aligned} &\textit{minimize} \quad \sum_{i=1}^{m} \frac{r_i}{Q_i^{**} - Q_i^*}(Q_i(\vec{w}) - Q_i{}^*)^2 \\ &\textit{subject to} \quad t_{c2q,i}^{aged}(\vec{w}) \le t_{c2q,i,max} \text{ for i=1,…,m} \\ &\qquad \tau_{s,i}^{aged}(\vec{w}) \le \tau_{s,i,max} \quad \text{for i=1,…,m} \\ &\qquad \tau_{h,i}^{aged}(\vec{w}) \le \tau_{h,i,max} \quad \text{for i=1,…,m} \end{aligned} \tag{13}$$

where $t_{c2q,i,max}$, $\tau_{s,i,max}$ and $\tau_{h,i,max}$ are maximum allowed values of t_{c2q}, setup and hold times in corner i, respectively.

In fact, the optimization strategy in (13) is to minimize an L2-norm criterion. In this criterion, the distance of each function Q_i from its ideal value, $Q_i{}^*$, is weighted proportional to the priority of corner i, i.e., r_i, and normalized by the distance between worst and best objective values at that corner, i.e., $Q_i{}^{**} - Q_i{}^*$. Notice that in the absence of designer feedback about the weight of each voltage corner, we set $r_i = 1/m$.

B. Statistical Problem Formulation

In this work we are concerned about two sources of process variation: effective channel length (L_{eff}) and threshold voltage (V_{th}.) We consider that these two sources of variation are independent. As mentioned in III, the threshold voltage change accounts for variations due to random dopant fluctuation (RDF) and NBTI.

The timing characteristics of flip-flops are affected by process variation. In general, the delay of a gate can be approximated by a first-order Taylor expansion:

$$d_g \cong d_g(L_0, V_{th0}) + \left(\frac{\partial d_g}{\partial L}\right)\Delta L + \left(\frac{\partial d_g}{\partial V_{th}}\right)\Delta V_{th} \tag{14}$$

where L_0 and V_{th0} are nominal channel length and threshold voltage.

Note: In principle, different transistors in a flip-flop can have different process variations which mean different ΔL and ΔV_{th}. However, in practice, considering that the dominant sources of variations tend to be spatially and locally correlated, one can rely on scalar ΔL and ΔV_{th} parameters to characterize the statistical behavior of a logic gate's delay. Most of the prior work papers focusing on delay or power characterization of library cells use the same approximation. For example, the authors of [30] adopt the same first-order Taylor expansion of the gate delay function for statistical characterization of logic gates in a standard cell library i.e., they assume that the logic gate delay behavior can be statistically modeled as linear functions of scalar ΔL and ΔV_{th} parameters, and one need not use a vector of ΔL and ΔV_{th} parameters corresponding to variations in the individual transistors of the logic gate.

On the other hand, delay of a gate can also be represented by the size of the transistors and sources of the variation using regression analysis:

$$d_g = d_g(\vec{w}, V_{th}, L) \tag{15}$$

We use regression analysis to model the timing characteristics of flip-flops.

A precise statistical equivalent for the problem formulation is described next. To handle variability of process parameters mentioned in III and VII.B, the problem is rearranged considering the mean and standard deviation of sources of variation.

Since there are variations in the circuits, timing characteristics of flip-flops are random variables. To set up the statistical problem formulation, means and standard deviations of flip-flops' timing characteristics are calculated. Both channel length and threshold voltage are assumed to be Gaussian random variables, which are approved by empirical data [26].

$$\begin{aligned} t_{c2q}^{aged}(\vec{w}, V_{th}, L) = {} & t_{c2q}^{aged}\left(\vec{w}, V_{th0} + DS_{th,NBTI}, L_0\right) \\ & + \Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,NBTI}\right) \\ & + \Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,RDF}\right) \\ & + \Delta t_{c2q}^{aged}(\vec{w}, \Delta L) \end{aligned} \tag{16}$$

where $\Delta V_{th,NBTI}$, $\Delta V_{th,RDF}$ denote changes in the V_{th} due to NBTI and RDF process variations, respectively. $DS_{th,NBTI}$ is the deterministic shift due to the NBTI effect, i.e., without any process variation. $t_{c2q}^{aged}\left(\vec{w}, V_{th,NBTI} + DS_{th,NBTI}, L_0\right)$ is a deterministic term and the following terms are Gaussian random variables. Hence, $t_{c2q}^{aged}(\vec{w}, V_{th}, L)$ is a random variable.

$$\Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,NBTI}\right) \sim N\left(0, \sigma_{t_{c2q}^{aged}(\vec{w}, \Delta V_{th,NBTI})}\right)$$

$$\Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,RDF}\right) \sim N\left(0, \sigma_{t_{c2q}^{aged}(\vec{w}, \Delta V_{th,RDF})}\right)$$

$$\Delta t_{c2q}^{aged}(\vec{w}, \Delta L) \sim N\left(0, \sigma_{t_{c2q}^{aged}(\vec{w}, \Delta L)}\right)$$

$\Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,NBTI}\right)$, $\Delta t_{c2q}^{aged}\left(\vec{w}, \Delta V_{th,RDF}\right)$ and $\Delta t_{c2q}^{aged}(\vec{w}, \Delta L)$ capture the changes in t_{c2q}^{aged} due to NBTI, RDF and channel length process variations, respectively.

Under (16) model and properties of Gaussian random variables; $t_{c2q}^{aged}(\vec{w}, V_{th}, L)$ is a Gaussian random variable with following mean (μ) and standard deviation (σ):

$$\mu_{t_{c2q}^{aged}} = t_{c2q}^{aged}\left(\vec{w}, V_{th0} + DS_{th,NBTI}, L_0\right) \tag{17}$$

$$\sigma_{t_{c2q}^{aged}} = \sqrt{\sigma^2_{\Delta t_{c2q}^{aged}(\vec{w}, \Delta V_{th,NBTI})} + \sigma^2_{\Delta t_{c2q}^{aged}(\vec{w}, \Delta V_{th,RDF})} + \sigma^2_{\Delta t_{c2q}^{aged}(\vec{w}, \Delta L)}} \tag{18}$$

The same analysis can be done for t_{setup} and t_{hold} for any point on the CSHT contour of the flip-flops (for given t_{c2q}.) In particular, as we will see later in this paper, we shall exploit the concept of minimum setup plus hold time (MSPH) point on each contour. So the sensitivity analysis done above for t_{c2q} can be easily done for t_{setup} and t_{hold} of the MSHP on each contour of interest. These analyses are omitted for brevity. Hence, the statistical problem formulation for single corner optimization is as follows:

$$\begin{aligned} & \textit{minimize} \quad Q(\vec{w}) = E^{fr}(\vec{w}).D^{fr}(\vec{w}) = \\ & \qquad E^{fr}(\vec{w}).\left(t_{c2q}^{fr}(\vec{w}) + \tau_s^{fr}(\vec{w})\right) \\ & \textit{subject to} \quad P\left\{t_{c2q}^{aged}(\vec{w}, V_{th}, L) \le t_{c2q,max}\right\} \ge \eta \\ & \qquad P\left\{\tau_s^{aged}(\vec{w}, V_{th}, L) \le \tau_{s,max}\right\} \ge \eta \\ & \qquad P\left\{\tau_h^{aged}(\vec{w}, V_{th}, L) \le \tau_{h,max}\right\} \ge \eta \end{aligned} \tag{19}$$

where η denotes the parametric yield.

The objective function for statistical optimization is the same as deterministic one which means we considered the deterministic values for fresh state. We can also consider the expected value for the objective function or even $\mu+\sigma$ or $\mu+3\sigma$ to capture the effect of process variation. However, since we just want to minimize the objective function, there is no extra information in considering any other form rather than deterministic one.

The change from (11) to (19) is that the deterministic constraints are transformed into probabilistic constraints. The new constraints capture the uncertainty due to the process variations in the optimization problem. We now analytically transform the probabilistic constraints using Gaussian distribution function characteristics. We just mention the transformation for t_{c2q} constraint. The analysis is the same for other constraints and omitted for brevity.

$$P\left\{t_{c2q}^{aged}(\vec{w}) \le t_{c2q,max}\right\} \ge \eta$$

$$P\left\{\underbrace{\frac{t_{c2q}^{aged} - \mu_{t_{c2q}^{aged}}}{\sigma_{t_{c2q}^{aged}}}}_{u} \le \frac{t_{c2q,max} - \mu_{t_{c2q}^{aged}}}{\sigma_{t_{c2q}^{aged}}}\right\} \ge \eta$$

$$\Phi\left(\frac{t_{c2q,max} - \mu_{t_{c2q}^{aged}}}{\sigma_{t_{c2q}^{aged}}}\right) \ge \eta$$

where $u \sim N(0,1)$ and $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt$.

$$\frac{t_{c2q,max} - \mu_{t_{c2q}^{aged}}}{\sigma_{t_{c2q}^{aged}}} \ge \Phi^{-1}(\eta)$$

Hence, $\mu_{t_{c2q}^{aged}} + \sigma_{t_{c2q}^{aged}} \Phi^{-1}(\eta) \le t_{c2q,max}$

Therefore, the final single corner optimization problem is as follows:

$$\begin{aligned} &\textit{minimize} \quad Q(\vec{w}) = E^{fr}(\vec{w}).D^{fr}(\vec{w}) = \\ &\qquad E^{fr}(\vec{w}).\left(t_{c2q}^{fr}(\vec{w}) + \tau_s^{fr}(\vec{w})\right) \\ &\textit{subject to} \quad \mu_{t_{c2q}^{aged}} + \sigma_{t_{c2q}^{aged}}\Phi^{-1}(\eta) \le t_{c2q,max} \\ &\qquad \mu_{\tau_s^{aged}} + \sigma_{\tau_s^{aged}}\Phi^{-1}(\eta) \le \tau_{s,max} \\ &\qquad \mu_{\tau_h^{aged}} + \sigma_{\tau_h^{aged}}\Phi^{-1}(\eta) \le \tau_{h,max} \end{aligned} \tag{20}$$

Here, μ's are basically the deterministic values of timing characteristics of the flip-flop and σ's are the representations of process variations in the circuit.

C. Critical Pair Definition on the CSHT Contour

It is mentioned that each $(\vec{w}, V_{th}, L)$ results in a different contour. To analyze the variation in the contours based on this tuple, we define few critical points on each contour in the fresh state. These points are the critical points which can be defined by the designer. There can be two or three points as mentioned in [15]; for example, the points with minimum setup or hold times.

Definition 4: The *minimum setup plus hold times* (MSPH) point is defined as the point on a contour which has *minimum* $\tau_s + \tau_h$.

In most of the designs, the desired setup time is the minimum one to increase the clock frequency as much as possible but there is a contrast between setup time and hold time in the sense that if one decreases the other one increases. In the case of minimum setup time, the hold time increases dramatically which causes hold violation in the circuit. Therefore, the desired point of operation for a flip-flop should be a point which minimizes the setup and hold times window which is MSPH point.

Hence, we choose MSPH point as the most critical point and throughout the rest of the paper we use it to do our analysis. This point can be easily found for each contour using *BES-algorithm* which is explained in section V.

D. Polynomial Modeling of Timing and Power Characteristics

After extracting each contour and finding the MSPH point on it by using the *BES-algorithm*, we do *Monte Carlo* simulation for each source of process variation and calculate μ and σ related to it for each specific transistor size vector, $\vec{w}$ based on (17) and (18), respectively.

Now that this statistical information is available for each transistor size vector, $\vec{w}$, we use regression analysis to find the polynomial functions which represent μ and σ of timing characteristics of the flip-flop in terms of transistor size vector, $\vec{w}$. Statistical information is only used in the constraints of the optimization problem; hence, it is just considered for the aged state.

These functions are the second order polynomials as follows

$$\sum_{i=1}^{n}\sum_{j\ge i}^{n}\alpha_{ij}W_i.W_j + \sum_{i=1}^{n}\alpha_i W_i + \alpha_0 \tag{21}$$

where n is the number of the transistors in the flip-flop and W_i's are the transistors' width.

Therefore, $\mu_{\tau_s^{aged}} = f_1{}^{aged}(\vec{w})$, $\sigma_{\tau_s^{aged}} = f_2{}^{aged}(\vec{w})$, $\mu_{\tau_h^{aged}} = f_3{}^{aged}(\vec{w})$ and $\sigma_{\tau_h^{aged}} = f_4{}^{aged}(\vec{w})$.

Objective function of the optimization problem only includes deterministic parameters which are for the fresh state. The same technique is used to find second order polynomial functions for $\tau_s^{fr}(\vec{w})$ using the MSPH points extracted from the fresh state contours. So, we have $\tau_s^{fr}(\vec{w}) = f^{fr}(\vec{w})$.

Energy and t_{c2q} are also modeled with the second order polynomial functions. Hence, the one corner optimization problem (11) considering second order polynomial modeling (21) can be rewritten as:

$$\begin{aligned} &\textit{minimize} \quad \left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\alpha_{ij}W_i.W_j + \sum_{i=1}^{n}\alpha_i W_i + \alpha_0\right).\left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\beta_{ij}W_i.W_j + \sum_{i=1}^{n}\beta_i W_i + \beta_0 + \sum_{i=1}^{n}\sum_{j\ge i}^{n}\gamma_{ij}W_i.W_j + \sum_{i=1}^{n}\gamma_i W_i + \gamma_0\right) \\ &\textit{subject to:} \\ &\left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\theta_{ij}W_i.W_j + \sum_{i=1}^{n}\theta_i W_i + \theta_0\right) + \left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\delta_{ij}W_i.W_j + \sum_{i=1}^{n}\delta_i W_i + \delta_0\right)\Phi^{-1}(\eta) \le t_{c2q,max} \\ &\left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\rho_{ij}W_i.W_j + \sum_{i=1}^{n}\rho_i W_i + \rho_0\right) + \left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\zeta_{ij}W_i.W_j + \sum_{i=1}^{n}\zeta_i W_i + \zeta_0\right)\Phi^{-1}(\eta) \le \tau_{s,max} \\ &\left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\omega_{ij}W_i.W_j + \sum_{i=1}^{n}\omega_i W_i + \omega_0\right) + \left(\sum_{i=1}^{n}\sum_{j\ge i}^{n}\psi_{ij}W_i.W_j + \sum_{i=1}^{n}\psi_i W_i + \psi_0\right)\Phi^{-1}(\eta) \le \tau_{h,max} \end{aligned} \tag{22}$$

The multi-corner optimization formulation (13) can be rewritten by using second order polynomial functions (21), which is omitted for brevity.

E. Complete Algorithm

The complete algorithm used to do the characterization and design optimization is as follows.

i. For some sampled combination of transistor sizes:
   a. Extract the MSPH point on each contour in fresh state using *BES-Algorithm*.
   b. Measure t_{c2q} delay and energy consumption.
ii. Find second order polynomial functions which represent the MSPH points' energy consumption, setup time and t_{c2q} in terms of size vector.
iii. For some sampled combination of transistor sizes:
   a. Extract the MSPH point on each contour in aged state using *BES-Algorithm*.
   b. Do Monte Carlo simulation to find μ and σ.
iv. Find second order polynomial functions which represent the MSPH points' μ and σ in terms of size vector.
v. Call *Multi-corner-opt* algorithm to solve the optimization problem for multi-corner optimization or (22) in the case of single corner optimization.
vi. Make the results discrete in terms of λ.

VIII. Simulation Results

We apply our mathematical program to TSPC flip-flop to determine the best transistor sizes for different corners of operation and also the optimum solution for multi-corner optimization. One of these corners is the corner representing the extreme the NBTI effect, i.e., for high operating temperature (85 degree Celsius.) It must be mentioned that the flip-flop is originally designed to have the minimum energy-delay product in the fresh state and the input signal probability is 0.5. All simulation results in this paper are obtained by HSPICE using a predictive 22nm technology model [22].

A. Polynomial Modeling Results

Timing and power characteristics of the flip-flop are modeled by using the second order polynomials. As an example the error histogram for modeling $\mu_{t_{c2q}}$ is provided in Fig. 7. The reported data is the relative error for data collected from HSPICE and the result of our modeling. The rest of the histograms (for modeling the other parameters) are omitted for brevity. However, TABLE I reports the error statistics of all parameter modeling results for the TSPC flip-flop. We can see that the error is increased for statistical timing characteristics in comparison with the deterministic ones. This level of accuracy is sufficient for our modeling. If there is any need for more precision, the second order polynomial modeling can be easily changed to higher orders like 3rd or 4th degrees to reduce the error.

On the other hand, since setup and hold times are codependent and this work is more focused on the design of flip-flops for achieving higher clock frequencies, we eliminate the hold time constraint in our optimization (22).

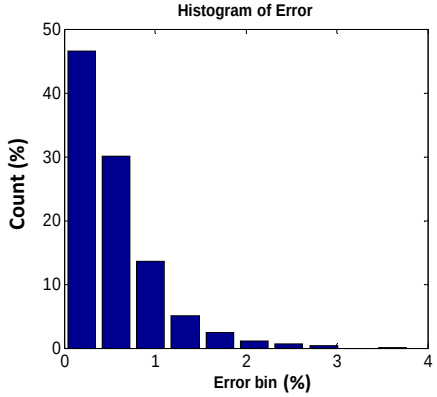

Fig. 7. Histogram of error in $\mu_{t_{c2q}}$ modeling.

B. Optimization Results

In this section we apply the proposed optimization algorithm to TSPC flip-flop to optimally size the transistors in its circuits to overcome the NBTI effect.

TABLE I

Error Statistics of Modeling Flip-flops Characteristics

Error (%)	Max	Mean	Standard deviation
Fresh setup time	1.1	0.22	0.17
Fresh t_{c2q}	3.2	0.4	0.25
Aged setup time	1.5	0.23	0.25
Aged t_{c2q}	2.6	0.36	0.24
Energy consumption	2.0	0.31	0.21
$\mu_{t_{c2q}}$	3.7	0.48	0.29
$\sigma_{t_{c2q}}$	5.5	0.68	0.49
μ_{τ_s}	4.3	0.57	0.44
σ_{τ_s}	4.9	0.64	0.47

The positive edge TSPC flip-flop, whose transistor-level schematics is shown in Fig. 8, features positive setup and hold times. The setup time is equal to the delay of the stage 1 (clocked) inverter whereas the clock-to-q delay is related to the summation of delays of the last three stages of the flip-flop. The hold time is the difference of the falling delays of stage 1 and stage 2 inverters.

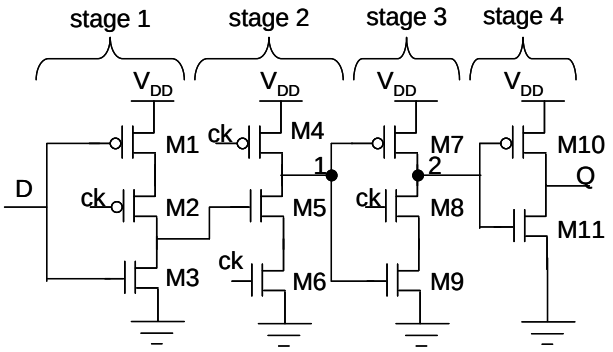

Fig. 8. Positive edge-triggered TSPC flip-flop.

1) Single corner optimization experiments

The first step for the simulation is to find the optimum sizing vector for the flip-flop, which minimizes the energy-delay product in the fresh state for 25˚C and 1.0V supply voltage. We denote this optimal sizing vector as $\vec{w}^{1*}$.

TABLE II shows values of the energy-delay product in the fresh state, area, clock-to-q delay, setup time, and power consumption in the aged state for $\vec{w}^{1*}$.

The aging effect experiment is done by changing the threshold voltage of the PMOS transistors in the TSPC circuit. We considered the effect of aging on the flip-flop after three years of operation. The change in the threshold voltage due to the NBTI effect consists of two parts, first the deterministic shift which is considered as the mean and second, the NBTI statistical variation. The deterministic shift (mean value) is calculated using the model provided in [3]. The statistical variation is determined using the model in [25]. The RDF effect and channel length variation are also considered and the total statistical variation (σ) is calculated using (18).

The characteristic values of TSPC flip-flop for $\vec{w}^{1*}$ are shown in TABLE II. It can be seen that deterministic aged setup time and t_{c2q} are increased by 33% and 15%, respectively. This amount of NBTI-induced increase in the timing characteristics of flip-flops is not acceptable and causes timing failure in the VLSI circuits.

TABLE II

TSPC FF CHARACTERISTICS FOR $\vec{w}^{1*}$

State	E.D. (fresh) ($fJ.ns$)	setup time (ps)	t_{c2q} (ps)	Power (μW)	Area (fm^2)
fresh	0.135	15	39	0.82	27
aged	0.150	20	45	0.77	27
percentage	11	33	15	-6	0

To overcome this undesirable outcome, transistors in the TSPC flip-flop should be sized up. We use the optimization algorithm given in section VII.E to size the transistors. Here the optimization is just for one voltage corner (1.0V.) We consider that up to 10% increase (compared to the corresponding fresh state values) in the setup time and t_{c2q} after a three-year aging process is acceptable. Hence, values of $\tau_{s,max}$ and $t_{c2q,max}$ in the mathematical program (22) are 17ps and 43ps, respectively. We denote the sizing solution of this optimization problem as $\vec{w}^{2*}$. The results of single corner optimization problem for TSPC flip-flop are reported in TABLE III. The degradation percentages are calculated with respect to the fresh state values of sizing vector $\vec{w}^{1*}$, i.e., the original energy-delay optimized TSPC flip-flop.

We use η = 0.975 as the timing-limited parametric yield ($\Phi^{-1}(\eta) = 1.96$). This means 97.5% of the results satisfy the constraints.

Note: We also consider the thermal run-away effect; i.e., by increasing the size of the transistors the power consumption increases which means larger temperature in the substrate. To consider this effect, we use compact thermal model (Electrical thermal analogy equation) which defines the change in the temperature of a cell (or a logical block) is related to the change in the power of that cell times the thermal resistance of that cell to the heat sink.

$$\Delta T = \Delta P\left(R_{Si} + R_{SiO_2} + R_{heat\ sink}\right) \tag{23}$$

where R_{Si}, R_{SiO_2} and $R_{heat\ sink}$ are the thermal resistances of the substrate, silicon and heat sink, respectively.

$$R_{Si} = \frac{R_{t,Si} \cdot H_{Si}}{A_{Si}} \tag{24}$$

where R_{t,S_i} is the *Si* thermal resistivity, A_{Si} is the area and H_{Si} is the height of *Si* layer.

The equations for R_{SiO_2} and $R_{heat\ sink}$ are omitted for brevity.

All the values for the above parameters are taken from Hotspot thermal analysis tool [35] and the area of the chip which this flop will be used as a standard cell is considered to be 2mm×2mm.

Since the increase in the power consumption of the flip-flop due to sizing is less than $1\mu W$, the increase in the temperature due to this phenomenon is negligible.

TABLE III

SINGLE CORNER OPTIMIZATION RESULTS FOR TSPC WITH SIZING VECTOR $\vec{w}^{2*}$

	E.D. (fresh) ($fJ.ns$)	setup time (ps)		t_{c2q} (ps)		Power aged state (μW)		Area (fm^2)
		μ	σ	μ	σ	μ	σ	
Value	0.147	15	1.1	30	6.5	0.78	0.1	36

Notice that the percentage change in the power consumption of the aged state is calculated in comparison by the power consumption of fresh state of sizing vector $\vec{w}^{1*}$. The leakage power decreases because of the increase in the threshold voltage of the PMOS transistors due to the NBTI effect. On the other hand, the increase in the size of the transistors causes the increase in the dynamic power consumption.

2) Multi-corner optimization experiments

There are four different corners (A through D) in our experiment corresponding to two voltage levels (1.0 and 1.2V) and two temperature values (25 and 85˚C.) Corners A, B, C and D are 25 ˚C and 1.0V, 25 ˚C and 1.2V, 85˚C and 1.0V and 85˚C and 1.2V, respectively.

The first step is to optimize each corner individually; therefore, the sizing vector solution for each corner is different from the others. The single corner optimization results for each corner are shown in TABLE IV.

We consider the results for fresh state of the flip-flop with sizing vector $\vec{w}^{1*}$ as the baseline. From now on, all the comparisons are with respect to this baseline.

The values of constraints for 25˚C corners are 10% increase in the fresh values of setup time and t_{c2q} , which are the same as before. However, for 85˚C corners there are no feasible solutions with these constraint values. So, we must relax the constraints. We allow 50% increase in the fresh values of setup time and t_{c2q}.

NOTE: In the case of doing deterministic optimization (not considering process variation effect), we could have a feasible solution by allowing 30% increase in the constraints' values. This shows the severe degradation effect of process variation.

TABLE IV entries for corner B show the importance of supply voltage level. When higher voltage value is used, the timing characteristics improve but power consumption becomes much larger. So, in this case, the sizes of the transistors in the flip-flop are reduced to keep the power consumption as low as possible. We see that decrease in the size of transistors caused an increase in the mean values but σ decreases a lot which underline the effect of supply voltage increase.

TABLE IV

SINGLE CORNER OPTIMIZATION RESULTS FOR TSPC ON FOUR DIFFERENT CORNERS

Corner	E.D. (fresh) (%)	setup time (*ps*)		t_{c2q} (*ps*)		Power aged state (μW)		Area (%)
		μ	σ	μ	σ	μ	σ	
A	9	15	1.1	30	6.5	0.75	0.1	35
B	51	15	1.0	40	1.5	0.96	0.08	-7
C	63	11	6	24	18	1.43	1.1	216
D	131	13	5	30	15	1.52	1.3	22

Results of corners C and D underline the big influence of temperature on the NBTI effect and power consumption. The increase in temperature does not change the mean value of setup time and t_{c2q} by that much (because of the sizing) however σ increases dramatically which makes satisfying the timing constraints almost impossible. To do so, we increase the value of $t_{c2q,max}$ and $\tau_{s,max}$ to 50% increase in the fresh values.

The idea of multi-corner optimization is to find a single solution that produces good results in all corners of interest. So far, we found the optimum solution for each corner independent of the other corners. Now, we use mathematical programming (13) to find the multi-corner optimization solution on corners A, B, C and D. We consider that the corners are all equally important (r_A= r_B= r_C= r_D = 0.25). TABLE V shows the results of this experiment for the statistical optimization. It is clear that the results of multi-corner optimization are worse than each corner's results for its own optimum sizing vector in TABLE IV. Fig. 9 illustrates the comparison between the results of TABLE IV and TABLE V. In Fig. 9 multi-corner A (B) means evaluating the results of multi-corner optimization at corner A (B).

TABLE V

MULTI-CORNER OPTIMIZATION RESULTS FOR TSPC EVALUATED AT DIFFERENT CORNERS

Corner	E.D. (fresh) (%)	setup time (*ps*)		t_{c2q} (*ps*)		Power aged state (μW)		Area (%)
		μ	σ	μ	σ	μ	σ	
A	39	17	1.2	35	7	0.70	0.11	25
B	76	12	0.7	26	1.0	1.36	0.05	25
C	92	20	11	49	20	1.23	1.3	25
D	155	12	4.5	28	10	1.55	1.2	25

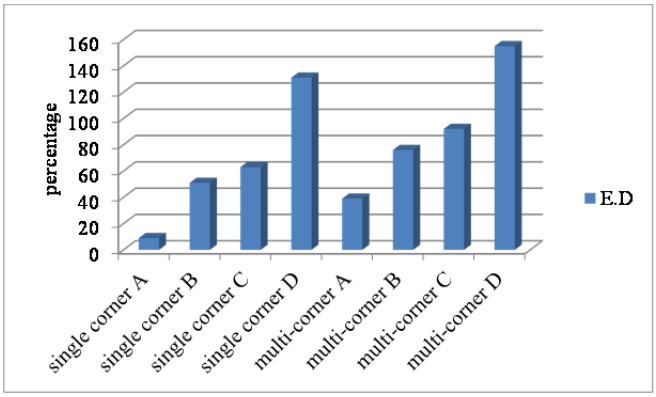

Fig. 9. Comparison between single corner and multi-corner optimizations.

IX. CONCLUSION

In this paper, we studied the NBTI effect on the setup/hold time codependency of flip-flops. We introduced BES-Algorithm as an efficient method to characterize CSHT contour and find MSPH point on it. After that, we explained the concept of statistical variation in NBTI degradation. We introduced our optimization algorithm considering statistical variations in the design including NBTI, RDF and channel length process variations. We used polynomial modeling technique to determine changes in flip flops' timing characteristics based on transistors width and showed that its error is negligible. Consequently, we introduced our multi-

corner optimization algorithms to minimize the energy delay product of flip-flops with aging effects on timing characteristics as constraints. We used nonlinear programming to solve the optimization problem to find the best transistor sizes. Finally, experimental results used to show our modeling accuracy and new transistor sizes for TSPC flip-flop.

REFERENCES

[1] B.C. Paul, K. Kang, H. Kuflouglu, M. A. Alam and K. Roy, "Impact of NBTI on the temporal performance degradation of digital circuits," *Electron Device Letters*, vol. 26, no. 8, pp. 560-562, 2005.

[2] D.K. Schroder and J.A. Babock, "Negative bias temperature instability: Road to Cross in Deep Submicron Silicon Semiconductor Manufacturing," *Journal of Applied Physics*, 2003.

[3] S. Bhardwaj, W. Wang, R. Vattikonda, Y. Cao, and S. Vrudhula, "Predictive modeling of the NBTI effect for reliable design," *Proc. of Custom Integrated Circuits Conference*, 2006.

[4] S.V. Kumar, C.H. Kim, and S.S. Sapatnekar, "NBTI-Aware Synthesis of Digital Circuits," *Proc. of Design Automation Conference*, 2007.

[5] K. Wu and D. Marculescu, "Joint Logic Restructuring and Pin Reordering against NBTI-Induced Performance Degradation," *Proc. of Design, Automation and Test in Europe*, 2009.

[6] B.C. Paul, K. Kang, H. Kufluoglu, M. A. Alam, and K. Roy, "Negative bias temperature instability: estimation and design for improved reliability of nanoscale circuits," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, No. 4, pp. 743-751, Apr. 2007.

[7] K. Duraisami, E. Macii, and M. Poncino, "Using soft-edge flip-flops to compensate NBTI-induced delay degradation," *Proc. of Great Lakes Symposium on VLSI,* 2009.

[8] S.V. Kumar, C.H. Kim, and S.S. Sapatnekar, "Impact of NBTI on SRAM Read Stability and Design for Reliability," *Proc. of Int'l Symposium on Quality Electronic Design*, 2006.

[9] Y. Wang et al., "Temperature-aware NBTI modeling and the impact of input vector control on performance degradation," *Proc. of Design, Automation and Test in Europe*, 2007.

[10] A. Abdollahi, F. Fallah, and M. Pedram, "Leakage current reduction in CMOS VLSI circuits by input vector control." *IEEE Trans. on VLSI Systems*, Vol. 12, No. 2, Feb. 2004, pp.140-154.

[11] Z. Qi and M.R. Stan, "NBTI resilient circuits using adaptive body biasing," *Proc. of Great Lakes Symposium on VLSI,* 2008.

[12] L. Zhang and R.P. Dick, "Scheduled Voltage Scaling for Increasing Lifetime in the Presence of NBTI," *Proc. of Asia and South Pacific Design Automation Conference*, 2009.

[13] W. Wang, S. Yang, S. Bhardwaj, R. Vattikonda, S. Vrudhula, F. Liu, and Y. Cao, "The impact of NBTI on the performance of combinational and sequential circuits," *Proc. of Design Automation Conference*, 2007.

[14] K. Ramakrishnan, X. Wu, N. Vijaykrishnan, and Y. Xie, "Comparative analysis of the NBTI effects on low power and high performance flip-flops," *Proc. of Int'l Conference on Computer Design*, 2008.

[15] H. Abrishami, S. Hatami, B. Amelifard, and M. Pedram, "NBTI-aware flip-flop characterization and design," *Proc. of Great Lakes Symposium on VLSI,* 2008.

[16] E. Salman, A. Dasdan, F. Taraporevala, K. Kucukcakar, and E.G. Friedman, "Exploiting setup–hold-time interdependence in static timing analysis," *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 6, Jun. 2007.

[17] S. Srivastava and J. Roychowdhury, "Rapid and accurate latch characterization via direct Newton solution of setup/hold times," *Proc. of Design, Automation, and Test in Europe Conference*, 2007.

[18] S. Srivastava and J. Roychowdhury, "Interdependent latch setup/hold time characterization via Euler-Newton curve tracing on state-transition equations," *Proc. of Design Automation Conference,* 2007.

[19] H. Abrishami, S. Hatami, and M. Pedram "Multi-corner, energy-delay optimized, NBTI-aware flip-flop design," *Proc. of Int'l Symposium on Quality of Electronic Design,* 2010.

[20] *Int'l technology roadmap for semiconductors*. Semiconductor Industry Association, 2011, http://www.itrs.net/

[21] R. Vattikonda, W. Wang, and Y. Cao, "Modeling and minimization of PMOS the NBTI effect for robust nanometer design," *Proc. of Design Automation Conference*, 2006.

[22] http://www.eas.asu.edu/~ptm/

[23] S. Hatami, H. Abrishami, and M. Pedram, "Statistical timing analysis of flip-flops considering codependent setup and hold times," *Proc. of Great Lakes Symposium on VLSI,* 2008.

[24] S. E. Rausch, "Review and Reexamination of Reliability Effects Related to NBTI-Induced Statistical Variations," *Transactions on Device and Materials Reliability*, Vol. 7, No. 4, Dec. 2007.

[25] K. Kang, S. P. Park, K. Roy, and M. A. Alam, "Estimation of Statistical Variation in Temporal NBTI Degradation and its Impact on Lifetime Circuit Performance," *Proc. of Int'l Conference on Computer Aided Design*, 2007.

[26] C. Visweswariah, "Death, taxes and failing chips," *Proc. of Design Automation Conference*, 2003.

[27] F. Kashfi, S. Hatami, and M. Pedram, "Multi-objective optimization techniques for VLSI circuits," *Proc. of the 12th Int'l Symposium on Quality of Electronic Design,* 2011.

[28] K. Miettinen, 1999: *Nonlinear Multi-criterion Optimization*. Boston: Kluwer Academic Publishers.

[29] H. Nakayama, Y. Yun, M. Yoon. 2009: *Sequential Approximate Multi-criterion Optimization Using Computational Intelligence*. Springer-Verlag Berlin Heidelberg.

[30] M. Mani, A. Devgan, and M. Orshansky "An Efficient Algorithm for Statistical Minimization of Total Power under Timing Yield Constraints," *Design Automation Conference*, 2005.

[31] S.H. Rasouli and K. Banerjee, "Effect of grain orientation on NBTI variation and recovery in emerging Metal-Gate Devices," *Electron Device Letters*, vol. 31, no. 8, pp. 794-796, 2010.

[32] H. Dadgour, K. Endo, V. De, and K. Banerjee, "Modeling and analysis of gain-orientation effects in emerging metal-gate devices and implications for SRAM reliability," *IDEM Tech. Dig.*, 2008.

[33] V. Huard et al., "New characterization and modeling approach for NBTI degradation from transistor to product level," *IEDM*, 2007.

[34] A. Islam et al., "Recent issues in negative-bias temperature instability: Initial degradation, field dependence of interface trap generation, hole trapping effects, and relaxation," *IEEE*

Transactions on Electron Devices, vol. 54, pp. 2143-2154, 2007.

[35] W. Huang et al, "Hotspot: A compact thermal modeling methodology for early-stage VLSI design." *IEEE Trans. VLSI Systems*, 14(5), pp. 501–513, July 2006.