Comprehensive Analysis, Modeling and Design for Hold-Timing Resiliency in Voltage Scalable Design

Huanyu Wang Geng Xie Jie Gu
Department of Electrical Engineering and Computer Science, Northwestern University
2145 Sheridan Road, Evanston, IL 60208
huanyuwang2014@u.northwestern.edu gengxie2014@u.northwestern.edu jgu@northwestern.edu

ABSTRACT

Resiliency to timing violation is a crucial requirement for low power electronics operating across a wide range of supply voltages. Although many existing solutions enhance setup timing tolerance for the higher performance, an accurate modeling and design strategy for hold resiliency dealing with conflicting requirement from both high voltages and low voltages has not been established. This paper proposes a novel voltage-scalable modeling technique that leverages conventional static timing analysis and efficient statistical analysis to achieve accurate stochastic hold timing analysis. Several highly non-linear behaviors of circuit operation are also incorporated into the proposed model to achieve a model accuracy of within 10% of spice Monte-Carlos simulation. Leveraging the proposed modeling technique, a novel hold resilience design technique is proposed to eliminate the excessive hold fixing operation for low voltage operation and its associated performance degradation at high voltage while still being compatible with conventional design closure flow. The proposed design methodology is demonstrated in a 45nm DSP processor design enabling a voltage-scalable operation from 0.35V to 0.9V eliminating more than 20,000 hold buffers as well as 23% performance degradation at high voltages due to hold fixing.

CCS Concepts

• **Hardware→Timing analysis** • **Hardware→Modeling and parameter extraction** • **Hardware→Fault tolerance**
• **Hardware→Process, voltage and temperature variations**
• **Hardware→Circuits power issues**

Keywords

Resilient design, setup and hold violations, ultra-low voltage operation

1. INTRODUCTION

Supporting a wide voltage operation range from nominal high supply voltage to near/sub-threshold region has become a critical requirement for battery operated devices to achieve low power consumption. To incorporate the challenges at low voltage operation especially under Process-voltage-temperature (PVT) variation, error resilient design has drawn significant efforts from industry and academia in the past decades. For example, the "razor" based design technique utilizes additional latch to detect timing error and flush the pipeline when an error is detected [1-3]. Several improved techniques for error resilient system have also been proposed. For example, a bubble razor technique was introduced to stall a clock cycle and intelligently propagate the error message throughout the pipeline [4]. A latch based error detection design along with voltage boosting technique was also introduced to create delay variation resilience at low voltages [5]. However, the razor type of technique sacrifices hold design margin and thus requires significant amount of hold verification and min-delay padding efforts, which make the technique more applicable for high performance design but not for near/sub-threshold operation where functionality and power is more important than performance. More recently, several latch based design with multi-phase clock was proposed to provide a viable migration to the hold timing issues at low voltages [6, 7]. However, latch based design using multi-phase clock generally requires additional retiming efforts and deviates from conventional synthesis design and timing closure flow leading to complexity of design adaptation especially if a high voltage operation is to be supported.

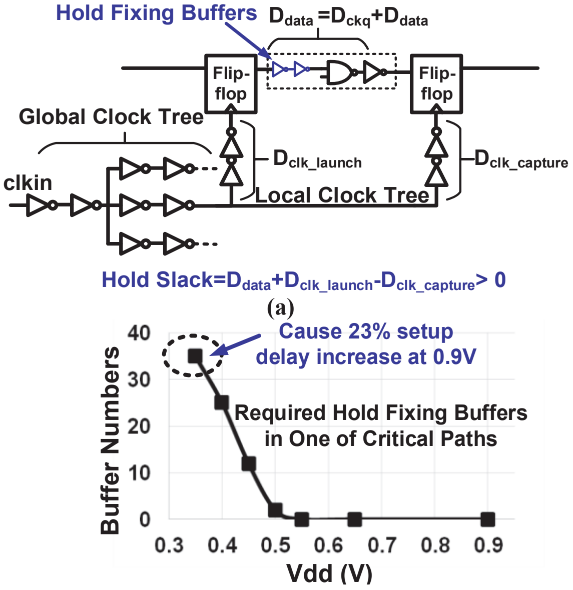

Hold Fixing Buffers
D_{data} =D_{ckq}+D_{data}
Flip-flop
Flip-flop
Global Clock Tree
D_{clk_launch}
$D_{clk\_capture}$
clkin
Local Clock Tree
Hold Slack=D_{data}+D_{clk_launch}-$D_{clk\_capture}$> 0
(a)
Cause 23% setup delay increase at 0.9V
Required Hold Fixing Buffers in One of Critical Paths
Buffer Numbers
40
30
20
10
0
0.3
0.4
0.5
0.6
0.7
0.8
0.9
Vdd (V)
(b)

Figure. 1 (a) Schematic of a typical hold timing paths; (b) Number of hold fixing buffers required in one of critical paths at various low voltages.

Local random process variation holds the largest threat to the chip timing closure as it cannot be captured by conventional corner based static timing analysis. Furthermore, the issue of random variation is significantly elevated for hold timing because the hold critical path is short and suffers from large amount of variation in comparison with setup timing path with deeper logic depth. As a result, either excessive pessimism is built into conventional corner based design or a time-consuming Monte-Carlo simulation or more sophisticated statistical based static timing analysis (SSTA) has to be utilized. For example, fast Monte-Carlos based SSTA has been proposed to estimate the delay variation with relative large computing expense [8-9]. Principle component analysis based SSTA has been proposed to capture the Non-Gaussian parameters in the manufacturing process [10-11]. A canonical model with incremental blocked-based analysis has also been used to allow statistical variables to be propagated down the logic path

ISLPED '16, August 08-10, 2016, San Francisco Airport, CA, USA
© 2016 ACM. ISBN 978-1-4503-4185-1/16/08…$15.00
DOI: http://dx.doi.org/10.1145/2934583.2934584

[12]. However, it is not clear if the above approaches can be extended into ultra-low voltage region where delay is lognormal. For low voltage operation, an operating point based analysis is demonstrated with high accuracy to predict the delay variation of the critical paths [13-14]. However, the iterative search used in the technique and the path-based analysis is rather expensive and requires large computing efforts with several rounds of path searching and analysis to close a design with large number of paths. Hence, there is a lack of efficient way of estimating the hold timing in ultra-low voltage design. Fig. 1(a) shows the schematic drawing of a typical hold timing path illustrating the slack definition for hold analysis. Fig. 1(b) shows simulation result from a processor design example in a 45nm process as will be discussed in section 4. The figure shows the required number of hold fixing buffers on one of the hold critical path across voltage range based on a spice level Monte-Carlo simulation. The required numbers of buffers exponentially increase with lower supply voltages. To allow the design to work down to 0.35V, 35 buffers need to be inserted into the single critical path due to the exponential increase of delay variation at low voltages. This in turn reduces the high voltage performance at 0.9V by 23%. Similar observation has been reported previously where the inserted hold fixing buffers take 60% of clock period leading to 2.2X increase of logic area to allow the design functioning at 0.35V [7]. This observation stresses two issues that we are trying to address in this paper (1) how to efficiently estimate the hold timing slack; (2) how to resolve the conflicting requirement from both low voltage operation and the high voltage operation.

Several innovations in this work are highlighted below: (1) A novel fast statistical timing modeling technique is proposed to efficiently predict the hold margin of the design. While most previous work focuses on SUM and MAX operation [10-12], we specifically modeled the SUBTRACTION operation in subthreshold domain, which is a critical operation for hold analysis. Furthermore, different from conventional separation of Gaussian and Lognormal models, our model features a unified format that can cover both high voltage and low voltage operation, leading to dramatically simplified characterization and modeling effort. The proposed model also leverage conventional STA result and look-up-table based stochastic approach to achieve high computing efficiency. (2) While many previous work use simplified Gaussian or lognormal model to perform stochastic analysis with lack of transistor level timing correlation [10-12], this work modeled several interesting highly non-linear circuit behavior due to non-ideal transistor-level operation at low voltage leading to highly accurate matching of circuit behavior. (3) Leveraging the help from the proposed modeling technique, a novel hold resilient design scheme is proposed to eliminate expensive hold fixing effort and performance impact to high voltage operation.

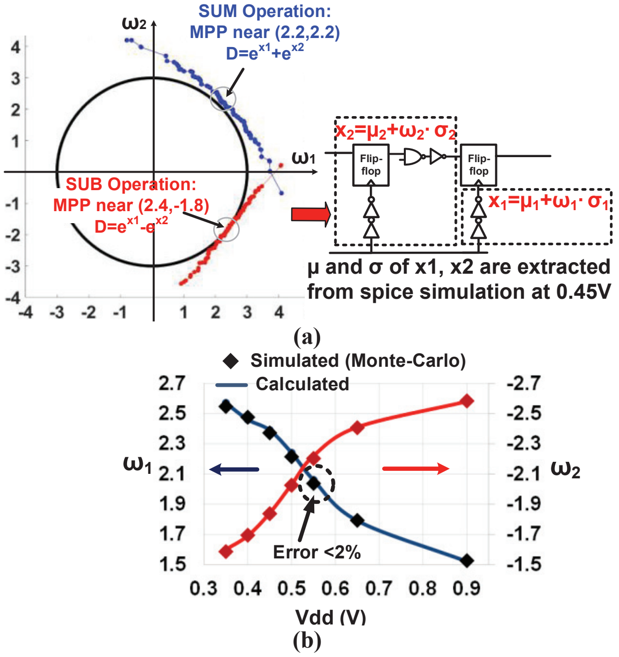
2. STATISTICAL HOLD TIMING MODELING

2.1 Stochastic Subtraction Operation at Low Voltage

In this session, we discuss a modeling methodology for subtraction using a Most Probable Point (MPP) analysis. The hold slack is defined as the difference between data arrival time and data required time as shown in Fig. 1(a) and equation (1):

$$Slack\_neg = D_{clk\_capture} - (D_{clk\_launch} + D_{clkq} + D_{data}) \qquad (1)$$

where $D_{clk\_capture}$ is the delay of capture clock path, D_{data} is the delay of data path, D_{clk_launch} is the delay of launch clock path, D_{clkq} is the delay from clock to output of a flip-flop. Here we calculate the negative slack because we are only interested in finding the maximum negative slack of the design. The goal of our analysis is to predict the stochastic hold slack value of equation (1) at a target percentile, e.g. 3 sigma of 99.7%. Although numerous efforts have been given in predicting a stochastic distribution of SUM and MAX operation of timing path, there is a lack of discussion on the subtraction (SUB) operation, which is critical for hold analysis. In this work, we adapt a Most Probable Points (MPP) Analysis where the vectors representing the cell level contribution to the joint probability of the entire paths are located and computed to find out the entire target numbers of sigma slack of the path [15-16]. The use of MPP method compared with other arithmetic equation based analysis allows us to take advantage of existing static timing analysis (STA) result and lookup table based stochastic analysis to achieve dramatic reduction of characterization and modeling efforts. One of the previous work utilizes an iterative search method to converge on the required MPP of each delay component including data path and clock path [13]. However, it requires extensive iterative search to find the MPP leading to days of computation for design closure of a large design. Instead, this work leverages STA results and proposes a simplified model that can accurately capture the stochastic distribution of the entire timing path without iterative search of MPP.



(a)

(b)

Figure. 2 (a) Monte-Carlo Simulated equal delay contour for SUM and SUB equations with parameters based on spice simulation at 0.45V. (b) Simulated and calculated Most Probable Point (MPP) of ω_1 and ω_2 for SUB operation.

Due to the lognormal delay at near-threshold or subthreshold region, the negative hold slack at low voltage could be formulated as the subtraction of two lognormal items as shown in (2):

$$S_{hold} = e^{x_1} - e^{x_2} \qquad (2)$$

where x_1 and x_2 are two lumped Gaussian variables with different mean and standard deviation, (μ_1, σ_1) and (μ_2, σ_2) respectively, due to random variation, e.g. threshold voltage variation. Here x_1, x_2 represents lumped stochastic delay variables for capture clock path delay and the sum of datapath delay and launch clock delay. The SUM operation will be discussed in 2.3. For simplicity, we normalize the variable x_1 and x_2 by introducing $\omega_1 = \frac{x_1 - \mu_1}{\sigma_1}$ and $\omega_2 = \frac{x_2 - \mu_2}{\sigma_2}$. To determine a stochastic target value of the negative hold slack, e.g. 3-sigma slack, the task is to identify the most possible points of ω_1 and ω_2 that provides the required delay value of negative hold slack at target probability. Note the number of sigma can be chosen arbitrarily and for simplicity, we use 3 sigma as our final target in this work. Equation (3) below elaborates the definition of ω_1 and ω_2 while equation (4) provides a condition for the most probable point.

$$S_{3\sigma} = e^{\sigma_1 \omega_1 + \mu_1} - e^{\sigma_2 \omega_2 + \mu_2} \qquad (3)$$

$$\omega_1^2 + \omega_2^2 = 3^2 \qquad (4)$$

For comparison purpose, we also formulate the SUM operation in a similar constraint as in (5) and (6).

$$S_{3\sigma} = e^{\sigma_1 \omega_1 + \mu_1} + e^{\sigma_2 \omega_2 + \mu_2} \qquad (5)$$

$$\omega_1^2 + \omega_2^2 = 3^2 \qquad (6)$$

Equation (4) and (6) are based on the theoretical expectation that the Most Probable Point (MPP) appears at points that have highest probability density function (PDF) and thus locate nearest to center of the hyper-space formed by ω_1 and ω_2 [15]. The CDF of the target slack function (3) and (5) is directly mapped to the CDF of random variables of ω_1 and ω_2. Theoretically, (4) and (6) is only true for normal distribution. Practically, it can be used to model non-linear process with reasonable accuracy [15]. To illustrate the foundation for equation (3)-(6), Fig. 2(a) shows the location of MPP points for both SUM and SUB operation based on 100,000 Monte-Carlos simulation with the μ and σ of lognormal delay extracted from real circuits using spice simulation on standard cell buffers operating at 0.45V. Each point represents a pair of ω_1 and ω_2 points that provides the same 3σ values for SUM and SUB operation. The group of points form an equal delay contour for SUM and SUB in the hyper-space of ω_1 and ω_2. Two key observations are highlighted here including: (1) the Most Probable Point (MPP) for both SUM and SUB happens near the tangent points of the equal delay contour and the sphere with a radius of the targeted sigma of 3 matching the theoretical expectation. (2) The MPP values of ω_1 and ω_2 represents a "balance" of the two random variables ω_1 and ω_2. For SUM, both values contribute equally and thus ω_1 and ω_2 have similar values. For SUB, the MPP settles toward unequal values, i.e. $\omega_1 = 2.4$ and $\omega_2 = -1.8$ because at the far-out tail of 3-sigma slack, the contribution from ω_1 dominates the contribution from ω_2 due to the lognormal behavior of the delay, i.e. positive tail outruns negative tail.

It is possible to find out an analytical solution for the MPP values for ω_1 and ω_2. Because the MPP values are located at the tangent point between the 3-σ cycle and delay contour, additional constraint equations can be obtained by taking differential operation to equation (3) to find out the tangent of the delay contour:

$$\frac{d\omega_2}{d\omega_1} = \frac{\sigma_1 e^{\sigma_1\omega_1+\mu_1}}{\sigma_2 e^{\sigma_2\omega_2+\mu_2}} \tag{7}$$

Considering the tangency condition to the cycle:

$$\frac{d\omega_2}{d\omega_1}\cdot\frac{\omega_2}{\omega_1} = -1 \tag{8}$$

Finally, we obtain the additional equation for solving ω_1 and ω_2:

$$\frac{\omega_1}{\omega_2} = -\frac{\sigma_1 e^{\sigma_1\omega_1+\mu_1}}{\sigma_2 e^{\sigma_2\omega_2+\mu_2}} \tag{9}$$

Hence, combining equation (4) and (9), the exact value of ω_1 and ω_2 can be calculated. We calculated the Most Probable Point (MPP) of ω_1 and ω_2 from equation (4) and (9) with different (μ_1, σ_1) and (μ_2, σ_2) whose values are extracted from the spice simulation on standard cells across voltages from 0.35V to 0.9V. Fig. 2(b) shows the calculated values of ω_1 and ω_2 using (4) and (9) in comparison with the Monte-Carlo simulation. The calculated values match with the Monte-Carlo simulation value within 2% error. This confirms that we could analytically calculate the 3-sigma value of negative hold slack by finding out the ω_1-sigma value of $D_{clk\_capture}$ minus the ω_2-sigma value of $D_{clk\_launch} + D_{clkq} + D_{data}$ as shown in equation (10). It is interesting to observe that the ω_1 and ω_2 values reverse the trend at high voltage, e.g. 0.9V. This is because the delay distribution becomes Gaussian distribution at high voltage. The sum of clock launch path and data path have a longer delay than the capture clock path (launch and capture clock path are balanced in clock design) and thus starts to dominate the overall hold slack at high voltages.

$$S_{3\sigma} = (D_{clk\_capture})_{\omega 1} - (D_{clk\_launch} + D_{clk\_to\_q} + D_{data})_{\omega 2} \tag{10}$$

Although it is possible to predict the MPP values of ω_1 and ω_2, in reality, the MPP values depend on the circuit configuration, i.e. values of μ and σ. As a result, a large number of circuit characterization still needs to be performed to obtain MPP values. To simply the analysis and characterization, we leverage the following conditions to reduce the analysis space: (1) corner based static timing analysis can be utilized to provide the results for the negative portion of the analysis, i.e. ω_2, leading to elimination of the majority of characterization and modeling efforts. In other words, with the help of STA, there is no need for characterization of the entire standard cell library and the large numbers of data path delay; (2) Since only the capture clock delay needs to be stochastically computed, the design space has been dramatically reduced by only characterizing the limited variety of clock buffers and depths of clock paths. Section 2.2 explains our approach.

2.2 Subtraction using Corner Based Static Timing Analysis

We compare the static timing analysis result, which is based on spice simulation using global corner model, with the Monte-Carlo simulation. Interestingly, the STA corner value of delay is always located at a negative sigma location. Although this deviates from the general expectation of corner location at 0-sigma (50%), it can be well explained from the SUM operation of lognormal variables. To prove the theoretical foundation of this observation, we adapt a widely used Wilkinson model for SUM of lognormal operation in this analysis [17]. A quick summary of the Wilkinson operation is given below. In Wilkinson's method, the sum of lognormal items $\sum_{i=1}^{N}\frac{1}{N}e^{x_i}$ can be approximated as another lognormal e^{y}, where y is a new Gaussian variable with calculable mean and standard deviation. This approximation is completed by matching the first and second moment of both equations. Ignoring the detailed derivation, we list the formula below in (11).

$$u_1 = E(s) = \textstyle\sum_{i=1}^{N}\frac{1}{N}e^{\mu_{xi}+\sigma_{xi}^2/2} = e^{\mu_y+\sigma_y^2/2} \tag{11}$$

$$u_2 = E(s^2) = \frac{1}{N}\left(\sum_{i=1}^{N} e^{2\mu_{xi}+2\sigma_{xi}^2} + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} e^{\mu_{xi}+\mu_{xj}}e^{(\sigma_{xi}^2+\sigma_{xj}^2+2r_{ij}\sigma_{xi}\sigma_{xj})/2}\right)$$

$$\mu_y = 2\ln u_1 - 1/2\ln u_2$$

$$\sigma_y^2 = \ln u_2 - 2\ln u_1$$

$(\mu_{x_i}, \sigma_{x_i})$ and (μ_y, σ_y) are the mean and standard deviation of the original Gaussian variables x_i and the new Gaussian variable y of the lognormal functions, respectively. r_{ij} is the correlation coefficient of each random variable and N represents the number of stages in the data or clock path. Using Wilkinson's law to model this process:

$$Ne^{y} = \textstyle\sum_{i=1}^{N} e^{x_i} \tag{12}$$

Each stage's delay is modeled as one lognormal item (e^{x_i}) and the sum of N stages is also a lognormal item (Ne^{y}). The lumped value Ne^{μ_y} represents the corner delay value reported from static timing analysis or corner spice simulator. Matching the corner location (e^{μ_x}) of the right hand side of (13) with the left hand side gives the difference between μ_y and μ_x :

$$Ne^{\mu_y+\beta\sigma_y} = Ne^{\mu_x} \tag{13}$$

$$\beta = (\mu_x - \mu_y)/\sigma_y$$

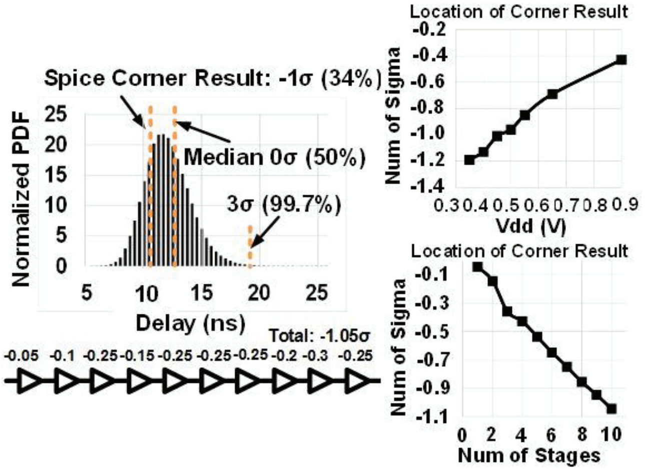


Figure. 3 Monte-Carlo simulation PDF compared with the spice corner delay (left). Corner location versus stages and supply voltages (right).

Due to the shift of the μ_y from μ_x in the SUM operation, the delay sum of a series of gates at mean delay value (Ne^{μ_x}), i.e. the spice corner delay, is no longer located at the mean location of the overall delay (Ne^{μ_y}). Instead, a negative shift at $\beta\sigma_y$ is observed due to the SUM operation of lognormal variables. Fig. 3 shows the histogram of Monte-Carlo spice simulation of a series of 10-stage buffers. The random variables at each buffer stage that contribute to the corner results are also annotated using similar approach as MPP. The overall corner location has been shifted to -1σ despite the fact that the delay at each stage stays at near 0σ. This result matches exactly with our mathematical model from (11) and (13).

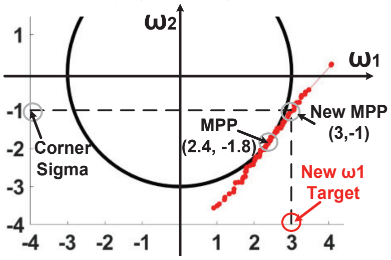


Figure. 4 MPP of SUM operation using STA results.

Because we target to utilize the STA results for our MPP values of $D_{clk\_launch} + D_{clkq} + D_{data}$ in equation (10), we could recalculate the MPP value for ω_1 based on the fact that ω_2 obtained from STA is centered around $-\beta\sigma$. Fig. 4 shows at 0.45V, the MPP becomes (3,-1) by using a STA result. As a result, we only need to obtain a 3σ delay for the capture clock to complete the hold slack analysis. As clock tree has been well balanced and contains less variety of configuration than the data path, the analysis has been significantly simplified. Fig. 3 also shows the simulated variation of corner location versus supply voltages and number of stages. At each supply voltage, the target sigma value for ω_1 is adjusted to account for the impact of supply voltages and depths of clock capture paths. A lookup table based approach is used to calculate the stochastic delay of the capture clock at various sigma targets as will be discussed in Section 2.3. Equation (14) summaries the hold slack calculation in this work where $D_{launch\_data\_STA}$ is the corner based STA result.

$$S_{3\sigma} = e^{\sigma_1 \times \omega 1 + \mu_1} - D_{launch\_data\_STA} \quad (14)$$

2.3 Unified Stochastic SUM Operation across Voltages

To compute the stochastic SUM operation of clock capture paths, we propose a unified model based on equation (11) with μ and σ characterized from spice simulation on delay of standard cell. Different from conventional timing analysis which assumes Gaussian operation for high voltage calculation and lognormal operation for low voltages leading to two separate characterization and analysis across supply voltages, in this work, we propose to only use lognormal distribution to model delay at entire voltage range including high voltages. The reason lies in the fact that are the lognormal distribution converges into Gaussian distribution when the ratio σ/μ becomes very small at a high voltage. Fig. 5 shows the simulated delay differences between Gaussian model and lognormal model across various numbers of stages and the PDF and CDF distribution of the two models at 0.9V. It is clearly seen that lognormal can be used to model the delay at high voltage with high accuracy. As a result, a unified model using lognormal model can be used for SUM operation across the voltage ranges. In this work, we use the Wilkinson equation as presented in (11) to model the SUM with standard cells' delay μ and σ characterized into a look-up-table as will be shown in section 2.5. The unified model leads to significant simplification of standard cell modeling and timing analysis. According to (11), we used an empirical correlation factor r_{ij} (~0.3) based on circuit level simulation to account for slew rate induced delay correlation between stages of the path.

2.4 Hyper-Lognormal Region for Transistor Level Cell Modeling

Most previous work has simplified the circuit delay as pure Gaussian or a lognormal delay based on the current relationship with threshold voltage variation [10-12]. Unfortunately, significant optimism can happen using a simplified lognormal model to characterize a standard cell delay at near-threshold region. The optimism stems from the fact that at near-threshold region, the transistor traverses across subthreshold region and linear/saturation region when the threshold voltage varies. As a result, characterizing the cell delay based on mean and standard deviation of the Monte-Carlo delay of a standard cell is likely to be optimistic because many data points are obtained when transistors operate at weak inversion rather than cut-off region. Fig. 6 shows a NMOS transistor current versus threshold voltage drawn in log scale. Rather than an ideal linear curve, the current flattens as the device moves into linear/saturation region.

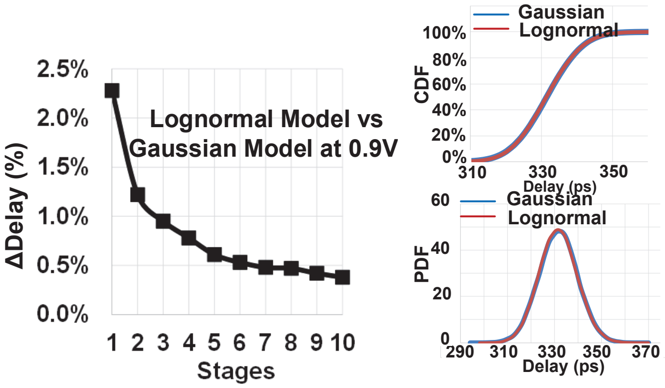


Figure. 5 Delay differences between Lognormal model and Gaussian model at 0.9V.

Fig. 7 shows the delay impact if a standard cell buffer is characterized at 0.45V. A 26% error is observed between ideal lognormal model and real circuit simulation. We refer this condition as "hyper-lognormal" effect because the effect introduces additional nonlinear behavior beyond a conventional lognormal model. To model such effect, we propose to use an additional σ_{hyp} to characterize the standard cell besides a normal σ_{norm}. While σ_{norm} quantify the overall delay distribution of the standard cell, σ_{hyp} captures the super-nonlinear tail of the delay distribution. Fig. 7 also shows the difference between σ_{hyp} and σ_{norm} characterized from a standard cell buffer across supply voltages. As expected, at both linear/saturation region and deep subthreshold region, σ_{hyp} and σ_{norm} converges to be the same while at near-threshold region (~0.5V), the hyper-lognormal effects reach the peak due to the crossing of operation region of transistors. In our methodology, we use a α value to present the impact of σ_{hyp} as shown in equation (15). A α value of 0.4 is used representing a balance of σ_{hyp} and σ_{norm}. Based on our analysis, the values of σ_{hyp} and α only matter for voltages at near-threshold region around 0.5V and does not introduce significant difference at deep subthreshold and high voltages.

$$\sigma_{new} = \alpha\sigma_{norm} + (1-\alpha)\sigma_{hpy} \quad (15)$$

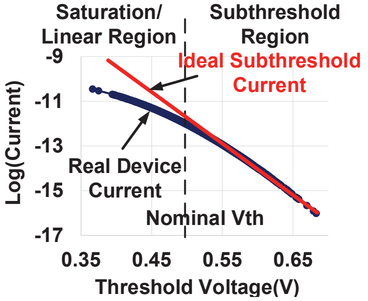


Figure. 6 Current versus threshold voltage in a NMOS transistor current at Vdd of 0.45V (Vds is set at Vdd/2).

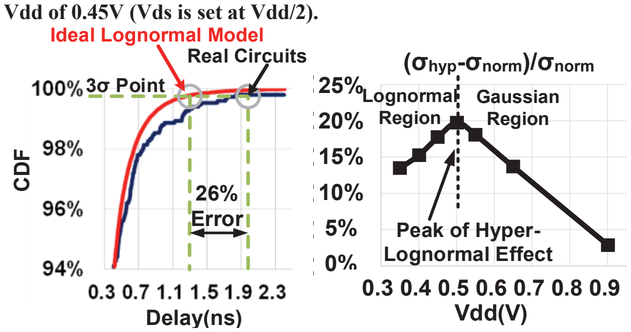


Figure. 7 CDF of buffer delay at 0.45V versus ideal lognormal delay model (left). The deviation of σ_{hyp} from σ_{norm} across Vdd (right).

2.5 Summary of Overall Hold Timing Analysis Flow

Fig. 8 summarizes our stochastic hold timing analysis flow. A 6x6 look-up-table (LUT) of μ and σ with various load and slew condition is generated from spice level Monte-Carlo simulation on standard cells related to clock paths at various supply voltages from 0.35V to 0.9V. Conventional static timing analysis is performed to find out the slew and load condition of the clock path as well as the corner delay for datapath and launch clock path. ω values for stochastic capture clock path delay are pre-characterized from MPP analysis described in section 2.1 and 2.2 depending on supply voltages and circuit configurations. The stochastic summation for capture clock is performed as described in 2.3. Calculation following equation (14) is used to obtain the final hold slack of a particular path. Because the σ value has been characterized in a LUT, any target stochastic location ωσ of the delay can be easily calculated following the proposed methodology. Due to the simplicity of our scheme and compatibility with existing timing analysis flow, the entire stochastic hold analysis can be performed with similar time as conventional static timing analysis, rendering orders of magnitude faster speed than the path based iterative search approach reported in previous work [13-14].

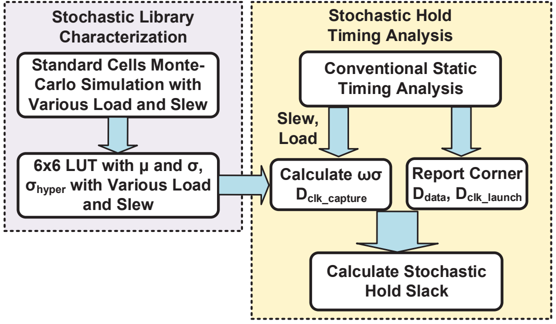

Stochastic Library Characterization
Standard Cells Monte-Carlo Simulation with Various Load and Slew
6x6 LUT with μ and σ, σ_hyper with Various Load and Slew
Stochastic Hold Timing Analysis
Conventional Static Timing Analysis
Slew, Load
Calculate ωσ D_clk_capture
Report Corner D_data, D_clk_launch
Calculate Stochastic Hold Slack

Figure. 8 Flow chart of proposed stochastic hold analysis method.

3. EVALUATION ON A DSP PROCESSOR

To test the proposed timing closure methodology, a 64-point 8-bit highly pipelined FFT processor was implemented using commercial synthesis and backend tools in a 45nm technology. Static timing libraries are generated across supply voltages from 0.35V to 0.9V for static timing analysis. Analysis is at slow corner and low temperature for worst case evaluation. The backend design with routing parasitics from layout was sent to commercial STA engine for both STA and spice netlist extraction. Although the clock tree has been well balanced in the design, due to exponential increase of delay variation at low voltages, significant hold timing issues are observed from 0.55V and below. We extracted selected top 50 paths with minimum hold slack for circuit level Monte-Carlo simulation evaluation. Due to the short data path and regularity of clock trees, the selected paths cover representative variety of clock paths and data paths. Transistor level spice netlist including both clock path and data path with extracted parasitics were simulated using Spice Monte-Carlo simulation. Scripts with the generated Lookup Table were used to perform the proposed timing analysis for comparison with Monte-Carlo simulation results.

Fig. 9 shows histograms of errors on the stochastic capture clock delay and overall hold slack at 0.35V and 0.9V. For the stochastic capture clock delay, the majority paths match within 5% with a maximum error of 8%. For overall hold stack, the maximum error is less than 10% while majority paths still match within 5%. The hold slack error is defined as the difference between the calculated stochastic hold slack and the Spice Monte-Carlo based simulated hold slack over the delay of capture clock path because this would avoid the singularity of divided-by-zero when slack is small and capture clock path delay dominates the worst-case negative slack. Fig. 10 shows the overall accuracy of the capture clock delay and hold slack across the voltages from 0.35V to 0.9V with worst case at 0.35V. This result highlights the accuracy of the proposed unified model where the high voltage is also properly modeled with the lognormal equation. Fig. 10 also shows the accuracy improves with higher voltages due to much tighter stochastic distribution and the improved accuracy of the static timing analysis which also introduces errors compared with spice simulation.

In addition, Fig. 1(b) in section 1 shows the numbers of buffers required for hold fixing from one of the worst-case paths in our design under various supply voltages. A worst-case of 23% performance degradation was observed. Although it is possible to perform more sophisticated backend design improvement to avoid impacting the setup path, it still requires significant design modification and iterations of design verification. As a result, in next section, we propose a novel hold resilient design scheme to remove the high voltage impact leveraging the hold timing analysis approach in this work as shown.

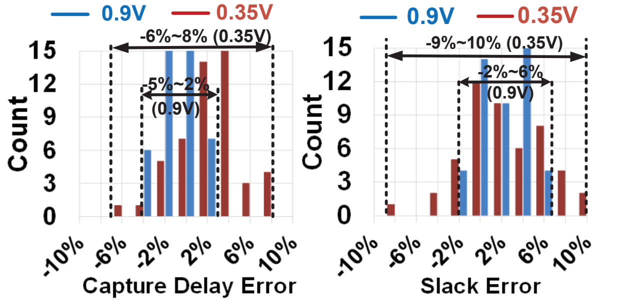


Figure. 9 Histogram of top 50 paths of the errors of the capture clock delay and overall hold slack at 0.35V and 0.9V.

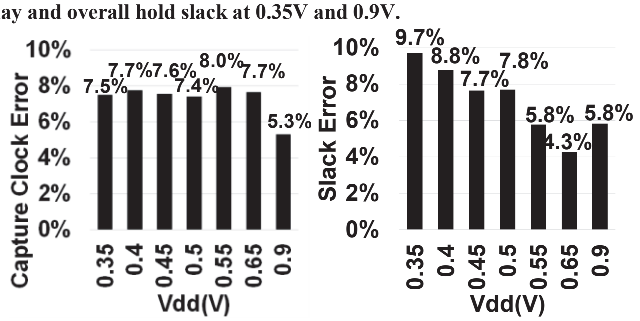
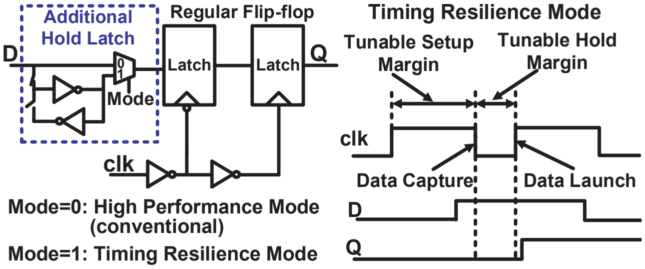


Figure. 10 Errors of proposed methodology across large voltages for capture clock delay (left) and overall hold slack (right).

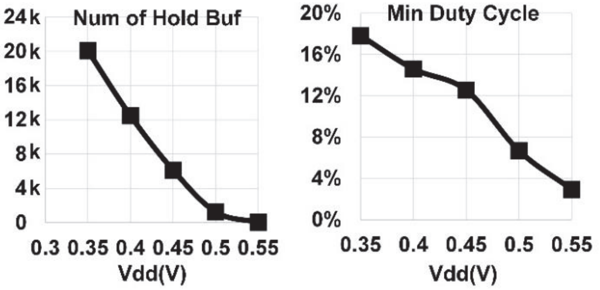
4. HOLD FREE SCHEME WITH CLOCK DUTY-CYCLE MODULATION

To accomplish the target of avoiding excessive hold buffers insertion for high voltage operation, we propose to replace conventional flip-flop with a dual-mode timing resilient flip-flop as shown in Fig. 11(a). An additional hold latch is added in addition to the conventional flip-flop. At high voltage, the additional latch is bypassed and the whole design flow as well as timing closure is identical as the conventional design strategy. At low voltage, the timing resilient mode of the flip-flip is activated as shown in Fig. 11(b). The additional hold latch in timing resilient mode only pass the data when clock is high and gates the input from the main flip-flop when clock is low. As a result, the flip-flop can be considered as only latching the data at falling edge of the clock leaving the entire time of clock-low period as hold timing margin. By modulating the clock duty cycle (defined as clock-low/clock-period), a programmable setup/hold timing margin can be achieved. The downside of this scheme is that the setup time is sacrificed by requiring the data to arrive before the falling edge of the clock although the duty cycle can be kept as minimum to reduce the performance impact. As performance is less of an issue for low voltage operation mode, the proposed scheme provides an optimum tradeoff for the conflicting requirements between high voltage and low voltage. For clock duty cycle control, a digital phase-locked loop or delay-locked-loop with digital controlled oscillator can be used to generate multiple phases for variable duty cycle. In our design, the selection of clock duty cycle and the selected insertion of the timing resilient flip-flop is determined from the proposed timing analysis in previous sections. As a result, we can accurately program the hold timing required across supply voltages without inserting excessive hold fixing buffers. The timing resilient flip-flop has been simulated across voltages with Monte-Carlos simulation for verifying functionality and timing at low voltages. The area overhead of the proposed timing resilient flip-flop is around 25% of the conventional flip-flop. We evaluated the proposed scheme in the DSP processor. Fig. 12(b) shows the required minimum duty cycle for guaranteeing the hold timing without inserting hold fixing buffers. As shown in the figure, no hold violation is observed at above 0.55V and thus the design can be set back into conventional mode. The minimum duty cycle increases at lower voltage as the negative hold slack becomes larger and reaches 18% of the clock period at 0.35V. Note that the minimum duty cycle is only the lower bound of the duty cycle in timing resilient mode. Other clock pulse width constraints required for reliable standard cell operations will likely limit the minimum duty cycle. Increasing duty cycle will increase performance degradation of the scheme at low voltage while gaining more hold margin to the design.



(a) (b)

Figure. 11 Proposed timing resilient flip-flop design (a) schematic; (b) Waveform of operation.



(a) (b)

Figure. 12 (a) Numbers of hold fixing buffers in conventional design; (b) Minimum Duty Cycle in the proposed scheme under different voltages.

We further evaluate all the timing paths (~44,737) in the design using our proposed timing analysis approach in the DSP processor example. Fig. 12(a) shows the total number of hold buffers needed across supply voltages in conventional design. Table 1 summaries the design spec and timing analysis statistics. To allow operation down to 0.35V, total 5,857 flip-flops (37% of all flip-flops) are converted into the timing resilient flip-flops. In conventional design scheme, a total of 20,058 hold buffers would have been inserted for fixing hold timing issues. In our scheme, the hold fixing buffers have been avoided rendering 23% performance improvement at 0.9V. The area overhead of the new flip-flops is compensated by the saving of the hold fixing buffers leading to a total area saving of 5.3%. More importantly, the proposed scheme enables a "hold-free" design strategy that allows the supply voltage to freely operate into subthreshold regime without compromising the high voltage performance.

5. CONCLUSION

This paper provides a comprehensive modeling and design methodology for achieving hold timing resiliency across a large voltage range. A computing efficient stochastic timing analysis approach is developed based on theoretical analysis using most probable point analysis and leverages the help from conventional static timing analysis. The developed timing analysis approach features a unified voltage-scalable timing model and incorporates highly-nonlinear effect of transistor behavior at near-threshold region to achieve high accuracy with computing effort similar with conventional STA. Based on developed timing analysis approach, a novel "hold-free" circuit solution is proposed. Demonstration on a 45nm DSP processor shows that compared with conventional hold fixing strategy, the proposed techniques not only accurately model the stochastic timing margin within 10% of Monte-Carlo simulation but also eliminate expensive hold fixing efforts rendering a "hold-free" operation across large supply range and significant performance saving at high voltages. In addition, proposed techniques are compatible with conventional design closure flow leading to easy adaptation for a voltage-scalable design.

Table 1. Design Spec and Statistics of the DSP processor

Spec	Values	Spec	Value
Technology	45nm	Total Area (w/o hold fixing)	0.135 mm^2
Supply Voltages	0.35V~0.9V	Num of Flip-flops	16,004
Clock Freq	600MHz (0.9V) 2MHz (0.35V)	Total Num of Cells	72,300
Max Negative Slack	-96n (0.35V)	No. of Hold Buffers for 0.35V	20,058
Area of Conv. Design with hold buffers	0.151mm^2	Area of Proposed Design Scheme	0.143 mm^2

6. ACKNOWLEDGMENTS

This work was partially funded by NSF grant CCF-1533656.

7. REFERENCE

[1] Shidhartha Das, et.al., "A Self-Tuning DVS Processor Using Delay-Error Detection and Correction", *IEEE Journal of Solid-State Circuits*, vol. 41, no. 4, pp. 792-804, Apr. 2006.

[2] Shidhartha Das, et. al., "RazorII: In Situ Error Detection and Correction for PVT and SER Tolerance", *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 32-48, Jan. 2009.

[3] Keith A. Bowman, et. al., "Energy-Efficient and Metastability-Immune Resilient Circuits for Dynamic Variation Tolerance", *IEEE Journal of Solid-State Circuits*, vol. 44, no. 1, pp. 49-63, Jan. 2009.

[4] Matthew Fojtik, , "Bubble Razor: An Architecture-Independent Approach to Timing-Error Detection and Correction", *International Solid-State Circuits Conference (ISSCC)*, pp. 488-490. Feb. 2012.

[5] Seongjong Kim, et. al., "R-Processor: 0.4V Resilient Processor with a Voltage-Scalable and Low-Overhead In-Situ Error Detection and Correction Technique in 65nm CMOS", *IEEE Symposium on VLSI Circuits*, 2014.

[6] Yanqing Zhang, Benton H. Calhoun, "Hold Time Closure for Subthreshold Circuits Using a Two-Phase, Latch Based Timing Method*", IEEE SOI-3D-Subthreshold Microelectronics Technology Unified Conf. (IEEE S3S)*, 2013.

[7] Seongjong Kim, Mingoo Seok, "Analysis and Optimization of In-Situ Error Detection Techniques in Ultra-Low-Voltage Pipeline", *International Conference on Low Power Electronics Design*, 2014.

[8] A. Singhee, S. Singhal, and R. A. Rutenbar, "Practical, fastmonte carlo statistical static timing analysis: Why and how," in *Proc. Int. Conf. Comput.-Aided Des.*, 2008, pp. 190–195.

[9] V. Veetil, D. Sylvester, and D. Blaauw, "Efficient monte carlo based incremental statistical timing analysis," in *Proc. Des. Autom. Conf.*, 2008.

[10]J. Singh and S. Sapatnekar, "Statistical timing analysis with correlated non-gaussian parameters using independent component analysis," in *Design Automation. Conf.*, 2006.

[11]Y. Zhan, A. J. Strojwas, X. Li, L. T. Pileggi, D. Newmark, and M. Sharma, "Correlation-aware statistical timing analysis with non-gaussian delay distributions," in *Proc. Des. Autom. Conf.*, 2005, pp. 77–82.

[12]Chandu Visweswariah, et. al, "First-Order Incremental Block-Based Statistical Timing Analysis", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 10, 2006.

[13]R. Rithe., "The Effect of Random Dopant Fluctuations on Logic Timing at Low Voltage," *IEEE Transactions on VLSI Systems*, vol. 20, no. 5, 2012.

[14]Nathan Ickes, "A 28 nm 0.6 V Low Power DSP for Mobile Application", *IEEE Journal of Solid-State Circuits*, vol. 47, no. 1, pp. 35-46, 2012.

[15]X. Du, W. Chen, "A most probable point based method for uncertainty analysis," *J. Des. Manuf. Autom.*, vol. 4, no. 1, pp. 47–66, Oct. 2001.

[16]Y. –T. Wu, H.R. Millwater, and T. A. Cruse, "An Advance Probabilistic Analysis Method for Implicit Performance Function," *Journal of American Institute of Aeronautics and Astronautics* , Vol. 28, pp. 1663-1669, 1990.

[17]A. A. Abu-Dayya and N. C. Beaulieu, "Comparison of Methods of Computing Correlated Lognormal Sum Distributions and Outages for Digital Wireless Applications," *IEEE Veh. Tech. Conf.*, vol. 1, pp. 175-179, 1994.