A General Framework for Accurate Statistical Timing Analysis Considering Correlations

Vishal Khandelwal
Department of ECE
University of Maryland-College Park
vishalk@glue.umd.edu

Ankur Srivastava
Department of ECE
University of Maryland-College Park
ankurs@glue.umd.edu

1. ABSTRACT

The impact of parameter variations on timing due to process and environmental variations has become significant in recent years. With each new technology node this variability is becoming more prominent. In this work, we present a general Statistical Timing Analysis (STA) framework that captures spatial correlations between gate delays. Our technique does not make any assumption about the distributions of the parameter variations, gate delay and arrival times. We propose a Taylor-series expansion based polynomial representation of gate delays and arrival times which is able to effectively capture the non-linear dependencies that arise due to increasing parameter variations. In order to reduce the computational complexity introduced due to polynomial modeling during STA, we propose an efficient linear-modeling driven polynomial STA scheme. On an average the degree-2 polynomial scheme had a 7.3x speedup as compared to Monte Carlo with 0.049 *units of rms error w.r.t Monte Carlo. Our technique is generic and can be applied to arbitrary variations in the underlying parameters.*

Categories and Subject Descriptors: B.8.2 [Hardware]: Performance Analysis and Design Aids

General Terms: Algorithms, performance, verification

Keywords: Statistical timing, variability, correlation

2. INTRODUCTION AND MOTIVATION

Statistical Timing Analysis has become a widely researched area with increasing impact of process and environmental variations on deep-submicron designs. The growing sources of variations along with the delay correlations they introduce in the design make it increasingly hard to perform fast and accurate timing analysis. Traditional design-corner based static timing analysis has become inaccurate due to pessimistic timing yield estimates. Monte-Carlo based statistical timing approaches become expensive in the presence of such large number of sources of variability.

The central idea in STA is to capture the variability by modeling delays as distributions and performing timing analysis statistically on these distributions while capturing possible correlations that could exist between gate delays.

A lot of recent work in statistical timing analysis tries to consider the process and environmental variabilities in performance analysis. Some approaches propose bounds on the statistical timing information [3, 2, 11] which can be computed efficiently for quick statistical timing estimation. Other approaches explicitly compute the timing statistically, making approximations at every step for curtailing the data explosion and improving the runtime. The authors in [6] propose a first order approximate delay model that takes into account both the correlated and independent randomness from different sources of variation. A similar strategy is presented in [8], where the authors present an efficient PERT-like traversal based statistical timing algorithm which considers the effects of the correlations of intra-die parameter variations by imposing an approximation similar to [6]. A moment based approach for capturing correlations is presented in [9].

In this paper we present a novel, general framework for accurate STA. In our approach we model each gate delay and arrival time distribution as a polynomial using Taylor-series expansion on the underlying parameters. The degree of the polynomial depends on the magnitude of the variations and the desired level of accuracy. Our technique also calculates the arrival time at each gate as a polynomial in the underlying parameters. As compared with running Monte-Carlo simulations to generate such timing information at each gate, we have significantly lower memory requirement as well as lower runtime. We do not make any assumptions about the distribution of the gate delays or arrival times in the circuit. Any arbitrary distribution will work in our general framework. In this paper we also present a strategy for computing the MAX of arrival time signals represented as polynomials. Using regression, we approximate the result of MAX back to a polynomial with minimum impact on error. Since all timing variables are approximated as polynomials in global parameters, the correlations are inherently considered. As the degree of the polynomial approximation is increased, the computation complexity of STA can become high. We also propose a novel linear-regression driven polynomial-modeling STA scheme. The computational complexity of this scheme is similar to that of the linear STA scheme. Hence, efficient STA using higher order polynomials can be done through our proposed approach. There are several ways in which our approach is superior to existing approaches.

1. The approaches in [6, 1, 8] model the dependence of gate and arrival time delay at each node in the circuit as a linear combination of global variations which are taken to be gaussian in nature. It is quite clear that this linear approximation can inject large amount of error in statistical timing estimate especially when the parametric variations become more significant. In our approach we represent all delay and timing signals as polynomials and therefore do not pay the penalty in accuracy.
2. We approximate each timing signal to be a polynomial in global variables. The approach of [6, 8] also assume each timing signal to be a linear combination of global variables. Their approach although will be valid only if these variables have a gaussian distribution. Our approach on the other hand is trivially generalizable to any distribution of the underlying variables.
3. We explicitly evaluate the arrival time at the output of each gate as a polynomial expression. This information can be used to perform optimization on the benchmark. As opposed to Monte-Carlo simulations, we do not have a memory overhead to generate this information at each gate.

Our experimental results have shown that the proposed polynomial gate delay and arrival time modeling scheme has on an average an *rms* error of 0.049 in the output CDF as compared to 0.158 from linear gate delay and arrival time

DAC 2005, June 13–17, 2005, Anaheim, California, USA.

modeling (which is done by most existing STA schemes) when compared with accurate Monte Carlo CDFs. This clearly brings out the effectiveness of polynomial modeling (assuming quadratic polynomials) of gate delays and arrival times to better capture the variability in timing due to parameter variations. The average runtime speedups for the polynomial scheme over Monte-Carlo was $7.3x$, while that from linear scheme was $7.5x$.

The rest of the paper is organized as follows: Section 3 describes modeling scheme used for parameter variations, correlation handling and gate delay computation. Section 4 contains the proposed STA framework along with the error management strategy used in this work. Section 5 presents our novel linear modeling based STA driven polynomial modeling STA scheme. Section 6 presents our experimental results and section 7 presents the conclusions drawn from this work.

3. MODELING PARAMETER VARIATIONS AND SPATIAL CORRELATIONS

In this section we will discuss the methodology that we impose for modeling the statistical correlations between the gate delay variables. We assume that the gate delay is dependent on a number of location-dependent parameters which are assumed to be mutually independent random variables. Let P_i, Q_i and R_i denote three such parameters (although our approach is very general and can be trivially extended to having more sources of variations also). Therefore, the delay of a gate i can be modeled as a function of these independent parameters as given by equation 1:

$$D_i = F(P_i, Q_i, R_i) \quad (1)$$

We note here that F can be a non-linear function of the parameters. Even if the underlying variables P_i, Q_i, R_i are gaussian distributions, the distribution in delay will not be gaussian. Most state of the art techniques for statistical timing assume the node delays to be gaussian either directly or indirectly. In our formulation we do not need to make any such approximation on either the delay distribution or on the distributions of P_i, Q_i and R_i. As it has been indicated in several other statistical timing techniques, spatial correlation would exist between delay variables of different gates due to spatial proximity. This occurs predominantly due to the fact that the underlying variables P_i, Q_i and R_i for two gates in close spatial proximity would show correlated behavior.

3.1 Spatial Correlation Modeling

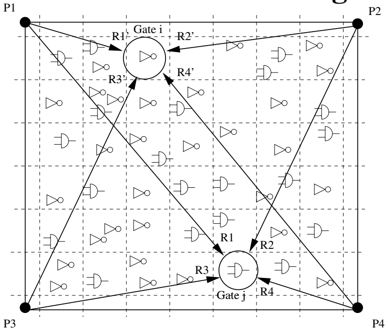

Figure 1: Grid-Based Spatial Correlation Model

We now present a modeling strategy to capture the spatial correlations between the parameter variables P_i, Q_i and R_i for each gate. Therefore, this technique captures the delay correlation between gates. Let us suppose that we are given a placed netlist as shown in figure 1. We impose a uniform grid on the placement to partition the gates into spatial regions. Let us now consider the parameter P and assume that its variation can be represented as a linear combination of four independent random components namely $P1$, $P2$, $P3$ and $P4$ that are zero mean and finite variance. These four random variables correspond to the four corners of the chip (as illustrated in figure 1). For any gate j, we model its corresponding parameter P_j as given by equation 2:

$$P_j = a_1P1 + a_2P2 + a_3P3 + a_4P4 + a_0 \quad (2)$$

where a_0 is the nominal value of parameter P_j. For any gate j in the netlist, we can compute the grid-based radial distance for the gate from the corners of the placement. This is represented by $R1$, $R2$, $R3$ and $R4$ for gate j as shown in the figure. The coefficients a_1, a_2, a_3 and a_4 can be computed by using these radial distances. Depending on the nature of the underlying variability parameter P_j (which can be obtained by analyzing the actual variability data), we can use an appropriate function $H(R)$ to compute these coefficients as follows:

$$a_1 = H(R1);\ a_2 = H(R2);\ a_3 = H(R3);\ a_4 = H(R4) \quad (3)$$

The underlying random variables $P1, P2, P3, P4$ can have any arbitrary distribution depending on the distribution of the parameter P_j. Therefore, we can see that if two gates i and j are far apart, they will get different contributions from each of the four components $P1$, $P2$, $P3$ and $P4$ and will have a weak correlation. If they are placed close together, then their coefficients will be similar and strong correlation will exist between them. In this way, we model spatial correlations for each of the remaining parameters in the system (Y and Z in this case).

Note that a similar strategy was proposed by [1] but the number of underlying random variables that capture the correlations was significantly higher. In our case we need only four variables per parameter ($P_i, Q_i etc.$) to capture the spatial closeness of two gates for capturing their correlations.

3.2 Gate Delay Modeling

We will now illustrate a gate delay model that incorporates the spatial correlation model described in the previous sub-section. We have represented our gate delay as a function of the independent parameters as given by equation 1. Each of P_i, Q_i and R_i can be represented as a linear combination of their underlying random components as given by equation 2. Hence, we can represent our gate delay as a function of these variables as:

$$D_i = G_i(P1, P2, P3, P4, Q1, Q2, Q3, Q4, R1, R2, R3, R4) \quad (4)$$

For simplification in representation, let us represent these variables as $Y1$, $Y2$, $Y3$, $Y4$, $Y5$, $Y6$, $Y7$, $Y8$, $Y9$, $Y10$, $Y11$ and $Y12$ respectively. We can use Taylor-series expansion about the mean values on this relation and obtain gate delay D_i as a sum of a series of multiple-order components as given by equation 5. The nominal values for the gate delay happens when all Y_i variables are zero (essentially no variance). Therefore $D_i(nominal) = G_i(0)$.

$$G_i = G(0) + \sum_{k=1}^{12}(Yk)G'(0) + 1/2!(\sum_{k=1}^{12}(Yk))^2G''(0)..... \quad (5)$$

The approach in [8] presents a similar strategy in which the delay for each gate is simplified according to Taylor series. Their approach however arbitrarily ignores the higher order polynomial terms and simply represents each gate delay as a a linear combination of the random variables (the Y_i terms in our case). Such a simplification is shown below

$$D_i = c_1Y1 + c_2Y2 + c_3Y3 + c_4Y4 + c_5Y5 + c_6Y6 + c_7Y7$$
$$+ c_8Y8 + c_9Y9 + c_{10}Y10 + c_{11}Y11 + c_{12}Y12 + G_i(0) \quad (6)$$

Typical gate delay models have terms which illustrate a high degree of non linear sensitivity. Such a linear approximation can inject a large amount of error in gate delay modeling (and therefore the statistical timing estimate) itself. In this work we choose not to ignore the higher order terms in the expanded Taylor series. Therefore, we model the gate delays as a general polynomial in the global variables Y_i. The order of this polynomial decides the degree of accuracy in the delay estimate. Note that this polynomial also has cross terms of the form Y_iY_j etc. A general second degree polynomial representing the gate delay would have the following structure

$$D_i = c_1Y1 + c_2Y2 + ... + c_{12}Y12 + c_{13}Y1^2 + ... + c_{24}Y12^2$$
$$+\ 66\ degree - 2\ cross - terms\ +\ G_i(0) \quad (7)$$

It can be seen that as we increase the degree of the approximating polynomial, the number of terms increase and the error in approximation reduces. Therefore it could be expected that there would be a tradeoff between runtime of statistical timing analysis and its accuracy. This tradeoff could be generated by controlling the degree of the polynomial used in representing the timing variables. Moreover, all delay variables in the circuit would share the same global variables Y_i. This would enable effective capturing of the correlations between them.

4. STATISTICAL TIMING ANALYSIS FRAMEWORK

We will now describe our general STA framework. We use a block-based STA approach that traverses the circuit topologically from the primary inputs to the primary outputs. There are two basic operations that are performed at each gate during this traversal. We first perform a SUM operation on the arrival time at a fanin and the corresponding gate delay. This SUM operation is repeated for each fanin of the gate. We then perform the MAX operation on the result of the already computed SUM operations. This gives us the arrival time at the output of the gate. As described in section 3, each gate delay is represented as a polynomial in the independent/global parameters. Following a similar strategy we would like to approximate each arrival time signal as a polynomial too. The approach in [6] proposes a similar strategy for representing all arrival time signals as linear combinations of global variables. At the end of the topological traversal of the circuit, the STA data has been generated. Let us now try to understand the two basic operations that are performed repeatedly in STA. Figure 2 shows a typical gate in the circuit that has K fanins and a polynomial gate delay representation D. The arrival time at fanin i of the gate is denoted by A_i, which is also a polynomial representation similar to D.

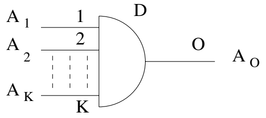

Figure 2: SUM and MAX Computation

$$D = poly(Y1, Y2, ..., Y12) \tag{8}$$
$$A_1 = poly(Y1, Y2, ..., Y12) \tag{9}$$
$$A_2 = poly(Y1, Y2, ..., Y12) \tag{10}$$
$$... \quad ... \quad ... \quad ... \tag{11}$$
$$A_K = poly(Y1, Y2, ..., Y12) \tag{12}$$

4.1 SUM Operation

It is very simple to compute the result of the SUM operation. Since arrival time and gate delay are both polynomials in the same independent parameters, the result of the SUM operation is also a polynomial. The coefficient of each term in the resulting polynomial is the sum of the coefficients of the corresponding terms in A_i and D. For each fanin i, we denote the result of the SUM operation by A_{io}:

$$A_{1o} = A_1 + D \tag{13}$$
$$A_{2o} = A_2 + D \tag{14}$$
$$... \quad ... \tag{15}$$
$$A_{Ko} = A_K + D \tag{16}$$

We note that this is an accurate computation and no approximation has been made at this step.

4.2 MAX Operation

We now compute the MAX operation in our proposed framework. We perform a MAX of K polynomials to get the arrival time signal A_o at the output of the gate. We would like to represent A_o as a polynomial too. Since all timing variables are represented as a polynomial in global variables, the correlations are effectively captured.

$$\begin{aligned} A_o &= MAX(A_{1o}, A_{2o}, ..., A_{Ko}) && (17)\\ &= poly(Y1, Y2, ..., Y12) && (18)\end{aligned}$$

It is known that the MAX operation introduces the complexity in STA. It is very hard to efficiently generate an accurate result of the MAX operation. We propose a regression based strategy to compute the resulting polynomial A_o by performing least square fitting. Assuming we know the degree of the polynomial that we want A_o to be approximated in, least square fitting will try to find the best polynomial of that degree that has the smallest error with the actual data of the MAX operation. Let us suppose that we are trying to approximate A_o with a degree two polynomial as indicated in equation 19. We need to evaluate all coefficients such that the resulting polynomial has smallest error when compared with the actual MAX data.

$$A_o = c_1Y1 + c_2Y2 + ... + c_{12}Y12 + c_{13}Y1^2 + ... + c_{24}Y12^2 + \ 66\ degree\_2\ crossterms\ + c_{91} \tag{19}$$

Now we will formalize the regression strategy that is used to compute these coefficients. Let us assume that we are given a set of n sampling vectors for the parameters (Y1,....,Y12) (these n samples will not be a very large set). We can evaluate the exact value of the MAX result at these n sampling vectors. This could be done by evaluating all the polynomials A_{io} and calculating their MAX. Let the ith value be represented by z_i. We can define a residual R for least square fitting as

$$R^2 = \sum_{i=1}^{n} \Big[z_i - (c_1Y1_i + ... + c_{12}Y12_i + c_{13}Y1_i^2 + ... + c_{24}Y12_i^2 + \ 66\ degree\_2\ cross\_terms\ + \ c_{91})\Big]^2 \tag{20}$$

This residue essentially is the root mean square error between the actual data of MAX z_i and the one predicted by the polynomial. In order of minimize the residual, we evaluate the partial derivative wrt. each coefficient in the polynomial and equate the result to zero. This can be represented as :

$$\frac{\partial(R^2)}{\partial c_1} = -2\sum_{i=1}^{n}\Big[z_i - (c_1Y1_i + ...)\Big]Y1 = 0 \tag{21}$$
$$\frac{\partial(R^2)}{\partial c_2} = -2\sum_{i=1}^{n}\Big[z_i - (c_1Y1_i + ...)\Big]Y2 = 0 \tag{22}$$
$$.... = ... \tag{23}$$
$$\frac{\partial(R^2)}{\partial c_{13}} = -2\sum_{i=1}^{n}\Big[z_i - (c_1Y1_i + ...)\Big]Y1^2 = 0 \tag{24}$$
$$.... = ... \tag{25}$$
$$\frac{\partial(R^2)}{\partial c_{91}} = -2\sum_{i=1}^{n}\Big[z_i - (c_1Y1_i + ...)\Big]1 = 0 \tag{26}$$

We can re-organize these to get equations :

$$c_1\sum_{i=1}^{n} Y1_iY1_i + c_2\sum_{i=1}^{n} Y2_iY1_i + ... + c_{13}\sum_{i=1}^{n} Y1_i^2Y1_i + ... + c_{91}\sum_{i=1}^{n} Y1_i = \sum_{i=1}^{n} z_iY1_i \tag{27}$$

$$c_1\sum_{i=1}^{n} Y1_iY2_i + c_2\sum_{i=1}^{n} Y2_iY2_i + ... + c_{13}\sum_{i=1}^{n} Y1_i^2Y2_i + ... + c_{91}\sum_{i=1}^{n} Y2_i = \sum_{i=1}^{n} z_iY2_i \tag{28}$$

....

$$c_1\sum_{i=1}^{n} Y1_iY1_i^2 + c_2\sum_{i=1}^{n} Y2_iY1_i^2 + ... + c_{13}\sum_{i=1}^{n} Y1_i^2Y1_i^2 + ... + c_{91}\sum_{i=1}^{n} Y1_i^2 = \sum_{i=1}^{n} z_iY1_i^2 \tag{29}$$

....

$$c_1\sum_{i=1}^{n} Y1_i + c_2\sum_{i=1}^{n} Y2_i + ... + c_{13}\sum_{i=1}^{n} Y1_i^2 + ... + c_{91}n = \sum_{i=1}^{n} z_i \tag{30}$$

We can combine these equations to give a more compact matrix representation as:

$$\begin{pmatrix} \sum_{i=1}^{n} Y1_i Y1_i & \dots & \sum_{i=1}^{n} Y1_i^2 Y1_i & \dots & \sum_{i=1}^{n} Y1_i \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} Y1_i Y1_i^2 & \dots & \sum_{i=1}^{n} Y1_i^2 Y1_i^2 & \dots & \sum_{i=1}^{n} Y1_i^2 \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \sum_{i=1}^{n} Y1_i & \dots & \sum_{i=1}^{n} Y1_i^2 & \dots & n \end{pmatrix}$$

$$\times \begin{pmatrix} c_1 \\ \dots \\ c_{13} \\ \dots \\ c_{91} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{n} z_i Y1_i \\ \dots \\ \sum_{i=1}^{n} z_i Y1_i^2 \\ \dots \\ \sum_{i=1}^{n} z_i \end{pmatrix}$$

Essentially, we have represented the polynomial regression as the system $YC = Z$ where we need to solve for the C matrix. There are several well known techniques for solving such a system of matrix, any of which could be used. This approach essentially selects the coefficients in such a way that the polynomial approximation of A_o has minimum error with the real data set z_i. This polynomial re-approximation is performed every time a MAX operation is computed.

The regression strategy used in MAX operation has two sources of complexity. The first one is the size n of the sampling values. Increasing the number of samples at each MAX operation increases the computational cost of this operation but improves the accuracy of the polynomial fit. Also, we note that as we increase the degree of polynomial approximation, the dimensions of matrix Y also increases. For first order linear regression, this matrix is a 13×13 matrix while for degree two approximation, this is a 91×91 matrix. Thus, we can clearly see a trade-off between accuracy and computation runtime through the order of polynomial approximation used.

We also point out here that the generality of our STA approach to handle all kinds of parameter variation distributions, gate delay distributions and arrival time distributions is made possible by not making any distribution based approximation in the MAX operation. Polynomial regression can be applied to any arbitrary distribution of the parameter variables Y_i and the accuracy controlled through the degree of the polynomial and the number of sampling vectors used.

After the topological traversal of the circuit, the arrival time at the primary output is represented as a polynomial in global variables. It can be seen that we have presented a generic statistical timing methodology that is not constrained by any assumptions on underlying distribution.

5. REDUCING COMPLEXITY IN POLYNOMIAL REGRESSION

We note that the computational complexity in polynomial STA comes primarily from the MAX operation as described in section 4.2. The size of the polynomial regression matrix formed in this step is grows exponentially with the degree of the polynomial approximation used. Hence, this step becomes the run-time determining step of the STA scheme. Ideally, we would like to maintain the accuracy obtained from using a higher degree polynomial (chosen to be degree 2 in this paper) while keeping a runtime that is comparable to an STA scheme with linear delay/arrival time models. The advantage of using regression is the generality in the scheme to handle timing distributions of any nature (not gaussian only) and the mathematical accuracy inherent in regression. In order to achieve the desired level of accuracy as well as runtime behavior, we propose a scheme that uses linear-modeling based STA to drive the polynomial STA.

5.1 Linear Regression Driven Polynomial STA

Polynomial modeling based STA is more accurate in generating the PDF/CDF of arrival time distribution because of two primary reasons: firstly, because polynomial gate delay modeling is better able to capture the nature of distribution due to the underlying parameter variation and secondly, because polynomial arrival time modeling is able to represent the PDF/CDF more accurately than linear modeling. However, the mean and variance of the arrival time distributions are captured with reasonable accuracy in the linear modeling based STA. We will now propose a polynomial modeling based STA technique that is driven by linear modeling based STA (which has lower runtime).

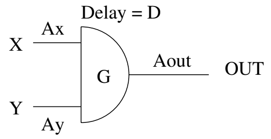

Figure 3: STA technique at Gate G

We traverse the circuit topologically and at each gate, we run linear STA and then use linear STA results to drive polynomial STA. Linear STA corresponds to performing linear regression assuming a linear model for arrival time and gate delay. Thus, we generate and store both linearly and polynomially modeled timing values at each gate. Let us suppose we are evaluating the arrival time at output of gate G with two fanins (X and Y) as shown in figure 3. For each input X and Y, we are given both linear and polynomial modeling values for the signal arrival times A_x and A_y respectively. Let us denote the linear arrival times as A_x^l and A_y^l respectively and the polynomial arrival times as A_x^p and A_y^p respectively. The linear and polynomial models for gate delays are given as D^l and D^p respectively.

The linear arrival time A_{out}^l at the output of gate G is given by:

$$A_{out}^l = MAX(A_x^l + D^l, A_y^l + D^l) \tag{31}$$

During linear STA we perform regression based MAX operation based on linear gate delay and arrival time models as given by equation 31. In section 4, we have discussed the details of the proposed regression based STA scheme. This enables the time consuming regression in the MAX step (section 4.2) to be much faster than the polynomial case (degree 2 or higher). The linear regression output gives us the arrival time (A_{out}^l) at the output of gate G as a linear combination of parameters:

$$A_{out}^l = c_0 + c_1 Y1 + c_2 Y2 + ... + c_{12} Y12 \tag{32}$$

where Y1, Y2,......., Y12 are the independent parameter variables as discussed in section 3. We know the distribution of these random variables and hence can calculate the mean and the variance of the arrival time A_{out}^l as:

$$Mean(A_{out}^l) = c_0 + c_1 * Mean(Y1) + ... + c_{12} * Mean(Y12) \tag{33}$$

$$Var(A_{out}^l) = c_1^2 * Var(Y1) + c_2^2 * Var(Y2) + ... + c_{12}^2 * Var(Y12) \tag{34}$$

We will now assume that the mean and variance of the output arrival time after linear regression is accurate. We will run polynomial STA by matching the mean and variance (first two moments) of the polynomial arrival time with the linear regression output. Let us now understand the scheme in more detail.

The polynomial arrival time at the output of gate G (say A_{out}^p) is given by:

$$A_{out}^p = MAX(A_x^p + D^p, A_y^p + D^p) \tag{35}$$

where A_x^p and A_y^p are the signal arrival times at the input-pins X and Y respectively and D^p is the polynomial gate delay. Now let us suppose that we know the probability p such that arrival time $(A_x^p + D^p) \geq$ arrival time $(A_y^p + D^p)$. We can calculate the probability $p = Prob(A_x^p + D^p \geq A_y^p + D^p)$ during the linear STA run at gate G.

We can run polynomial STA on gate G by utilizing this probability p to generate an output polynomial A_{out}^p, which will then be scaled to match its first two moments to the values evaluated from linear regression based STA as given by equations 33 and 34. Let the output arrival time polynomial A_{out}^p be generated as follows:

$$A^p_{out} = p * (A^p_x + D^p) + (1-p) * (A^p_y + D^p) \tag{36}$$

where A^p_x and A^p_y are the polynomial arrival times of the signal at the fanin pins X and Y (which have already been calculated previously). After this step, we need to match the variance of A^p_{out} to the variance of A^l_{out} from linear regression. For simplicity of understanding, we will keep this discussion limited to A^p_{out} being a polynomial of degree 2 as given by:

$$\begin{aligned} A^p_{out} = c_0{+}c_1Y1{+}c_2Y2{+}.....{+}c_{12}Y12{+}c_{13}Y1^2{+}.....{+}c_{24}Y12^2 \\ +\ \ 66\ degree-2\ cross-terms \end{aligned} \tag{37}$$

But the analysis that follows can trivially be extended to higher order polynomial modeling as well. Since we know the distribution of each underlying parameter variation ($Y1$ to $Y12$), we know their mean and variance values. We can evaluate the mean and variance of A^p_{out} as follows:

$$\begin{aligned} Mean(A^p_{out}) = c_0{+}c_1{*}Mean(Y1){+}.....{+}c_{12}{*}Mean(Y12) \\ + c_{13} * Mean(Y1^2) + ...\ other\ terms \end{aligned} \tag{38}$$

$$\begin{aligned} Var(A^p_{out}) = c_1^2 * Var(Y1) + ... + c_{12}^2 * Var(Y12) \\ + c_{13}^2 * Var(Y1^2) + ... + c_{24}^2 * Var(Y12^2) \\ + c_{25}^2 * Var(Y1 * Y2) + ...\ other\ cross\ terms \\ + 2c_1c_2 * Cov(Y1, Y2) + 2c_1c_3 * Cov(Y1, Y3) \\ + ...\ all\ other\ covariance\ terms \end{aligned} \tag{39}$$

We will first match the variance of A^p_{out} (from equation 39) with that of A^l_{out} (from equation 34) by scaling A^p_{out} with a factor α such that:

$$\alpha^2 = Var(A^l_{out})/Var(A^p_{out}) \tag{40}$$

$$A^{p'}_{out} = \alpha * A^p_{out} \tag{41}$$

The mean of the new scaled polynomial will be:

$$Mean(A^{p'}_{out}) = \alpha * Mean(A^p_{out}) \tag{42}$$

Hence, to match the mean of the polynomial arrival time expression with that obtained from linear regression (equation 33), we can add a constant factor β to the constant term c_0 of $A^{p'}_{out}$ such that:

$$\beta = Mean(A^l_{out}) - Mean(A^{p'}_{out}) \tag{43}$$

$$c_0' = c_0 + \beta \tag{44}$$

Hence the final polynomial arrival time at the output of gate G can be given by A^{poly}_{out}:

$$A^{poly}_{out} = \alpha * A^p_{out} + \beta \tag{45}$$

This completes our linear regression driven polynomial STA technique. We have avoided the complexity of solving a large polynomial regression problem at each gate (during the MAX operation) by solving a smaller linear regression problem and then performing moment matching (first two moments) as explained in this section. The runtime complexity of this scheme will be of the order of the runtime for linear regression.

5.2 Gaussian Approximation

The linear regression based technique discussed earlier in this section is applicable to any given gate delay distribution. If the gate delay distributions are known to be gaussian, then we could avoid the generic regression based linear STA, and use faster techniques (under the gaussian approximation) to drive the polynomial STA. In [6], the authors have proposed a first order approximate delay model based STA under the assumption that all underlying parameters have a gaussian distribution. Their scheme approximates the arrival time distribution after the MAX operation to be gaussian as well. For brevity, we do not go into the details of their scheme. As explained in the previous subsection, to drive polynomial STA using linear STA, we need to evaluate three quantities for each MAX operation: the probability p that one arrival time is larger than the other, the mean of the output arrival time distribution and the variance of the output arrival time distribution. In our generic scheme we use linear regression to get these quantities. Under the gaussian assumption, the authors in [6] use the results from [5, 10] to perform the MAX operation. They represent each arrival time as a linear combination of gaussian random variables (representing the underlying parameter variations). The distribution of the timing signal after the MAX operation is reapproximated back as a linear combination of the underlying gaussian variables (to maintain the gaussian form). Analytical expression proposed in [5, 10] are used to evaluate the mean and variance of the result of performing the MAX operation on two jointly gaussian arrival time distributions. They use the probability p of one arrival time being larger than the other to generate an expression for the resulting linear arrival time. Using the variance value obtained from the analytical expression, they match the variance of the resulting output arrival time to get the final output arrival time expression as a linear combination of the underlying parameter variations.

We can utilize the scheme presented in [6] to drive our polynomial STA under the assumption that underlying parameters are gaussian and by imposing a first order approximation on gate delay. This is a faster technique than performing our more generic regression based STA to drive polynomial STA and can be used when the underlying parameters are given to be gaussian in nature.

6. EXPERIMENTAL RESULTS

The proposed STA framework was implemented in sis [7] assuming three underlying parameters. For the gate delay model in equation 46, we assumed the parameters supply voltage (V_{dd}), threshold voltage (V_{th}) and the velocity saturation index for short channel effects (α) as the underlying sources of variability. We used an academic placement tool (CAPO [4]) to get a valid placement for each benchmark. This placement information was used to generate the V_{dd}, V_{th}, α variations at each gate as indicated in equation 2. This automatically captures correlations due to spatial proximity. We imposed 10%, 20% and 7% variability on corresponding mean values of $1.8V$, $0.5V$ and 1.3 respectively. Furthermore, each of these parameters were assumed to have a uniform distribution to see the effects of relaxing the gaussian distribution assumption to consider a more general approach.

$$D_i \propto \frac{C_L V_{dd}}{(V_{dd} - V_{th})^\alpha} \tag{46}$$

In section 4, we have presented the general regression based STA scheme. As the degree of the polynomial approximation is increased, the computation complexity of regression makes this scheme impractical. In section 5, we have proposed a novel linear regression driven polynomial STA scheme. The computational complexity of this scheme is similar to that of linear regression. Hence, efficient STA using higher order polynomials can be done through this scheme. We assumed the gate delays were a second order polynomial of the parameters. This second order polynomial for each gate delay was generated using best fit regression with Monte Carlo data for gate delay. The Monte Carlo data for the gate delay was calculated using the delay model indicated in equation 46 with different parameter instances. We generated accurate timing CDFs for each benchmark using equation 46 for gate delays through Monte Carlo simulations. All runtimes and error comparisons are are made with Monte Carlo.

We experimented with the following cases:

1. Using linear gate delay and arrival time models, we performed regression based STA (as described in section 4). This approach is similar to the one proposed by state of the art STA techniques like [6, 8] where all delay and arrival time variables are assumed to be linear approximations of underlying global parameters.

Benchmark	Monte Carlo	Linear-Modeling STA			Linear-Driven Polynomial Modeling		
	Runtime	Runtime	Speedup	*rms* Error	Runtime	Speedup	*rms* Error
C432	3152	1461	2.2X	0.147	1495	2.1X	0.053
C499	10421	3114	3.4X	0.164	3157	3.3X	0.057
C880	8687	2566	3.4X	0.136	2586	3.4X	0.035
C1355	10381	3081	3.4X	0.175	3099	3.4X	0.033
C1908	13649	3220	4.2X	0.155	3237	4.2X	0.042
C3540	68204	7834	8.7X	0.153	7916	8.6X	0.047
C5315	148053	10414	14.2X	0.170	11735	12.6X	0.093
C6288	381245	18601	20.5X	0.165	18635	20.5X	0.035
Average			7.5X	0.158		7.3X	0.049

Table 1: Runtime and *rms* Error Comparison

2. We performed polynomial STA using our proposed linear regression driven polynomial STA scheme (as described in section 5). All gate delays as well as arrival times in the STA were represented as degree two polynomials.

Table 1 presents the experimental results. All runtime and error comparisons are made wrt. Monte Carlo results. Columns 2, 3 and 6 present the runtime values for Monte Carlo, linear regression based STA and linear regression driven - polynomial STA respectively. The corresponding speedups wrt. Monte Carlo are given in columns 4 and 7 respectively. On an average, we get $7.5x$ and $7.3x$ speedup compared with Monte Carlo runtime from the two schemes respectively. On an average, there is 0.158 and 0.049 units of *rms* error in the output CDFs from the two schemes respectively as compared with the accurate CDFs from Monte Carlo. These results point out the superiority of polynomial STA as compared to linear STA. Polynomial gate delay and arrival time models are better able to capture the distribution as compared to linear models. We also note that the runtime from the linear regression driven polynomial STA are comparable to that of pure linear regression based STA. Our proposed scheme is a fast technique to perform higher order polynomial approximations during STA. With increasing variability in underlying parameters, such a scheme would be very useful.

From the runtime speedups reported in table 1, we can see that as the benchmark size is increasing (listed in order of increasing number of total gates) the speedup as compared with Monte-Carlo also increases. We note here that we perform regression and Monte-Carlo simulations at the same number of samples to make a fair comparison. Additionally, as pointed out earlier in the paper, we generate a polynomial expression for arrival time at each gate which can be used for performing optimization. In order to generate this information using Monte-Carlo simulations, we would require a very high memory overhead to save the results at each gate. These are the advantages of using our regression based scheme over Monte-Carlo based simulations.

Even though the *rms* error numbers are small in magnitude, they can make a significant impact on the CDF. For example, the average *rms* error in linear regression scheme is 0.158 units, so if we are looking at the 50 percentile point on the accurate CDF, the predicted CDF potentially be showing a value of either 0.342 or 0.658, which is a very significant difference from the actual value of 0.5. The impact of this inaccuracy on decisions made on the design using these CDFs could be very drastic.

Figure 4 depicts the CDF at the output of benchmark C880. We can see that the linear regression driven polynomial STA gives us a more accurate CDF as compared to the linear regression based STA scheme. This clearly brings out the superiority of polynomial STA over linear STA.

7. CONCLUSION AND FUTURE WORK

In this work we have proposed a general framework for accurate STA. Our scheme is independent of the distributions of the variations, gate delay and arrival times. We consider the impact of intra-die parameter variations on gate delays and also consider the spatial correlations that can exist between them. We have proposed a polynomial gate delay modeling scheme where the order of the polynomial can be decided by the desired accuracy as well as the magnitude of the underlying variations. We have presented a regression based MAX computation technique that can be used to represent each arrival time as a polynomial in the underlying parameters. However, since the computational complexity of this regression increases significantly with the degree of the polynomial, we have proposed a novel linear regression driven polynomial STA scheme. Our results show the advantage of using polynomial modeling over linear modeling as done in the existing literature.

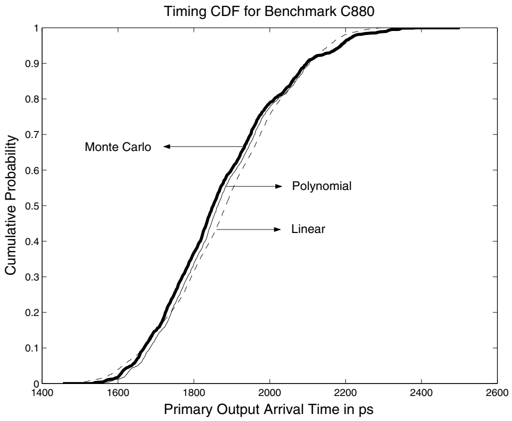

Figure 4: CDF Result for C880

Future work would be to develop fast techniques for polynomial MAX operation in STA. As the impact of the parameter variations increases, non-linearity creeps in and we need to develop higher order approximation schemes which are accurate but at the same time efficient in computation.

8. REFERENCES

[1] A. Agarwal, D. Blaauw and V. Zolotov. "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations". In *Procs of ICCAD*, 2003.

[2] A. Agarwal et al. "Computation and Refinement of Statistical Bounds on Circuit Delay". In *Procs of DAC*, 2003.

[3] A. Agarwal, V. Zolotov and D. Blaauw. "Statistical Timing Analysis Using Bounds and Selective Enumeration". In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.22, Sept. 2003.

[4] A. Caldwell et al. "Can Recursive Bisection Alone Produce Routable Placements?". In *Proc. of DAC*, 2000.

[5] C. E. Clark. "The Greates of a Finite Set of Random Variables". In *Operations Research*, pages 145–162, 1961.

[6] C. Visweswariah et al. "First-Order Incremental Block-Based Statistical Timing Analysis". In *Procs of DAC*, 2004.

[7] E.M. Sentovich, K.J. Singh, L. Lavagno, C. Moon, R. Murgai, A. Saldanha, H. Savoj, P.R. Stephan, R.K. Brayton, A.L. Sangiovanni-Vincentelli. *SIS: A System for Sequential Circuit Synthesis.* Memorandum No. UCB/ERL M92/41, Department of EECS. UC Berkeley, May 1992.

[8] H. Chang and S. Sapatnekar. "Statistical Timing Analysis Considering Spatial Correlations Using a Single Pert-Like Traversal". In *Procs of ICCAD*, 2003.

[9] J. Le, X. Li and L. Pileggi. "STAC: Statistical Timing Analysis with Correlation". In *Procs of DAC*, 2004.

[10] M. Cain. "The Moment Generating Function of the Minimum of Bivariate Normal Random Variables". In *The American Statistician*, pages 124–125, May 1994.

[11] M. Orshansky et al. "Fast Statistical Timing Analysis Handling Arbitrary Delay Correlations". In *Procs of DAC*, 2004.