Statistical Timing Analysis with Extended Pseudo-Canonical Timing Model

Lizheng Zhang*, Weijen Chen§, Yuhen Hu†, Charlie Chung-Ping Chen‡
*§† ‡ ECE Department, University of Wisconsin, Madison, WI53706-1691, USA
Email:*§{lizhengz,weijen}@cae.wisc.edu, †‡{hu,chen}@engr.wisc.edu

***Abstract*— State of the art statistical timing analysis (STA) tools often yield less accurate results when timing variables become correlated due to global source of variations and path reconvergence. To the best of our knowledge, no good solution is available dealing *both* types of correlations *simultaneously*.**

In this paper, we present a novel extended pseudo-canonical timing model to retain and evaluate both type of correlation during statistical timing analysis with minimum computation cost. Also, an intelligent pruning method is introduced to enable trade-off runtime with accuracy.

Tested with ISCAS benchmark suites, our method shows both high accuracy and high performance. For example, on the circuit c6288, our distribution estimation error shows $15\times$ accuracy improvement compared with previous approaches.

I. Introduction

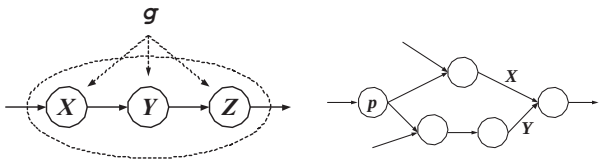
It is well-known that the timing performance of future generations of deep-submicron micro-architecture will be dominated by several factors. IC manufacturing process parameter variations will cause device and circuit parameters to deviate from their designed value. Low supply voltage for low-power applications will reduce noise margin, causing increased timing delay variations. Due to dense integration and non-ideal on-chip power dissipation, rising temperature of substrate may lead to hot spot, causing excessive timing variations.

Classical worst case timing analysis produces timing predictions that are often too pessimistic and grossly conservative. On the other hand, statistical timing analysis (STA) that characterizes timing delays as statistical random variables offers a better approach for more accurate and realistic timing prediction.

In literatures, there are two distinct approach for STA: **path based STA** and **block based STA**. The fundamental challenge of the path based STA [1]–[4] is its requirement to select a proper subset of paths whose time constraints are statistically critical. This task has a computation complexity that grows exponentially with respect to the circuit size, and hence can not be easily scaled to handle realistic circuits.

This potential difficulty has motivated the development of block base STA [5]–[10] that champions the notion of *progressive computation*. Specifically, statistical timing analysis is performed block by block in the forward direction in the circuit timing graph without looking back to the path history. As such, the computation complexity of block based STA will grow linearly with respect to the circuit size. To even further speed up the computation, *Gaussian assumption* has been widely adopted([6], [9], [10]) with small accuracy penalty, and all internal timing random variables in a circuits are forced to follow the Gaussian distribution.

However, to realize the full benefit of block based STA, one must solve a difficult problem that timing variables in a circuit could be correlated due to either *global variation* ([6], [7], [10]) or *path reconvergence*([5], [9]). As illustrated in the left hand side of Figure 1, *global correlation* refers to the statistical correlation among timing variables in the circuit due to *global variations* such as inter- or intra-die spatial correlations, same gate type correlations, temperature or supply voltage fluctuations, etc. *Path correlation*, illustrated in the right hand side of Figure 1, refers to the correlation resulting from the phenomenon of *path reconvergence*, that is, timing variables may share a common subset of gate or interconnect along their path histories.



(a) X, Y and Z depend on g (b) X and Y depend on p

Fig. 1: Global Correlations (left) and Path Correlation(right)

The importance of the path correlation comes from the fact that each gate in the circuit will have some local variations which are independent to other gates in the circuit. These local variations will propagate towards the circuit output and cause additional correlations. However, the correlations caused by sharing some of these local variations because of path reconvergence, cannot be correctly captured by any algorithm that deals with global variations only.

Several preliminary solutions have been proposed to deal with these correlations. In [6], [7], [10] the dependence on global variations is explicitly represented using a *canonical timing model*. However, none of these approaches has taken into account the path correlations. In [9], a method based on common node detection is introduced to deal with the path correlations. However, this method does not address the issue of dependence on global variations.

In this paper, we present a systematic STA solution that takes into account correlations caused by *both* global variations and path reconvergence. Specifically,

- We *extend* the commonly used canonical timing model to

represent all timing variables in the circuit as a weighted linear combination of a set of Gaussian random variables. A *variation vector*, consisting of all the weights, is then used to explicitly represent *both* global and path correlation information.

- We develop a *pseudo*-canonical timing model which relax the requirement of the independence between the global variation sources involved in the original canonical timing model. With such, the expensive principle component analysis is avoided and total computation time is saved.
- We further explore the sparse structure of the variation vector and develop a *flexible vector format* so that the non-significant entries of the variation vector are dynamically dropped during computation. According to simulations on ISCAS circuits, this technique significantly curtails the amount of storage and computation required for our method.

Since $min(X,Y) = -max(-X,-Y)$, in the interests of brevity, in the rest of this paper, we will only discuss the MAX operator, with the understanding that the same results can be easily adapted to the MIN operator.

The rest of the paper is organized as following: In section II, previous block based STA methods are reviewed briefly; Section III describes the vectorized timing format and a theorem used for correlation decompose; Section IV is the detailed algorithm and technique to reduce computation complexity. Section V presents a real implementation of our method in C/C++ and the testing result with ISCAS85 benchmark suites; Section VI gives the conclusions.

II. Canonical Timing Model

In timing analysis field, the circuit is modeled as a *timing graph*, where nodes are used to represent the gate/wire in the circuit and there will be some delay, called *node delay*, associated with them. Signals propagate through these nodes will add their delays into its *arrival time*. Node delay and arrival time are usually called *timing variables* of the circuit.

Different from classical timing analysis, the statistical timing analysis models timing variables as *random variables*, which are characterized by its *probability density function(p.d.f.)* or *cumulative distribution function(c.d.f.)*. The purpose of statistical timing analysis is then to estimate the arrival time distribution at the primary output of the circuits knowing input arrival time distributions and all internal node delay distributions. This is accomplished through two operators [5]:

- ADD: When an input arrival time X propagates through a node delay Y, the output arrival time will be $Z = X+Y$
- MAX: When two arrival times X and Y merge in a node, a new arrival time $Z = max(X,Y)$ will be computed before the node delay is added.

[6], [7], [10] proposed a *canonical delay model* to address the node delay correlations through sharing global variations. In particular, they model each of the node delay as a summation of three terms:

$$n_i = \mu_i + \alpha_i R_i + \sum_j \beta_{i,j} G_j \quad (1)$$

where $n_i (i = 1, 2, ...)$ are random variables corresponding to the the i^{th} node delay in the timing graph; μ_i is the expected value of n_i; R_i, (named *node variation*), is a zero-mean, unity variance Gaussian random variable representing the localized statistical uncertainties of n_i; G_j represents the j^{th} *global variation*, and is also modeled as a zero-mean, unity variance Gaussian random variable; $\{R_i\}$ and $\{G_j\}$ are additionally assumed to be mutually independent; the weight parameters α_i (named *node sensitivity*) and $\beta_{i,j}$(named *global sensitivities*) are deterministic constants, *explicitly* expressing the amount of dependence of n_i on each of the corresponding independent random variables.

With this canonical representation, the correlation (covariance) between any two node delays, n_i and n_k, can be easily evaluated.

$$cov(n_i, n_k) = E\{(n_i - \mu_i)(n_k - \mu_k)\} = \sum_j \beta_{i,j}\beta_{k,j} \quad (2)$$

Note that random variables $\{R_i, R_k, G_j (j = 1, 2, ...)\}$ are mutually independent.

This canonical timing model is very good for node delay(gate delay or interconnect delay) modeling. But it is not sufficient to model the arrival time. When an arrival time is somehow mapped into this canonical timing model as done in [10], some important path information will be dropped and the accuracy will be compromised.

For example, referring back to Figure 1(b), arrival time X and Y share the common history of node p. So between them, there will be some extend of correlation caused by the node p's local variation R_p. But X and Y have only one term of local variation of R_X and R_Y respectively in their canonical representations. And according to the assumption from the canonical timing model, R_X and R_Y are assumed to be independent to each other, so the correlation caused by R_p in node p is incorrectively ignored.

Another significant disadvantage of the original format of the canonical model is that it requires the independence between the global variation sources G_i. This requirement, as stated in [10], will be satisfied by *Principle Component Analysis(PCA)* before the timing analysis. But we think it is not reasonable to exclude the computation complexity of PCA from the overall performance evaluation of the statistical timing analysis because the complexity of the PCA grows as quickly as $O[M^3]$ where M is the total number of global variation sources considered. Due to the important spatial correlation between global variation sources, the number of global variation sources have to be included in the PCA will easily go up to thousands and so that PCA itself will be a very timing consuming step.

For example, to model the spatial correlation, the chip is usually partitioned into grids and each grid will be associated with m global variations. So the total number of global variation sources will be proportional to the number of grids used to represent the chip area. If the chip is partitioned into 10×10 grid, the total number of global variations will be $M = 100m$. This number maybe tolerable but the coarse grid may

not be sufficient to accurately model the spatial correlation of global variations between different grid locations. If we partition the chip with 100×100 to get more accurate spatial correlation model, the total number of global variations will now be $M = 10000m$ which is beyond the reasonable range for cases where a fast PCA is demanded. To be a fact, for a real chip with dimension of $10mm \times 10mm$, 100×100 grids will only have grid cell with size of $100\mu m \times 100\mu m$ which is not yet small enough for high quality spatial correlation modeling.

III. Extended Pseudo-Canonical Timing Model

The canonical timing model [6], [7], [10] is a powerful tool to represent the numerous timing variables for a given circuit. However, as pointed out in the previous section, in its original format, it can only handle node delay correlations caused by global variations. Furthermore, the original format of canonical timing model requires a computationally expensive PCA which may prohibit it from using in cases where large number of global variations are correlated with each other. In this work, we propose an *extended pseudo-canonical timing model(EPCT)* that is capable of capture *all* the correlation between any pair of timing variables in the circuit be it a node delay or an arrival time. And also it removes the requirement of a PCA procedure for its validation.

A. Extended Pseudo-Canonical Timing Model

We notice an important fact that it is not necessary to have the canonical format to be able to evaluate the correlations caused by global variation sources. So we propose a *Pseudo-Canonical Timing Model* for node delay as following:

$$n_i = \mu_i + \alpha_i R_i + \sum_j \beta_{i,j} G_j \tag{3}$$

which, compared with the commonly used canonical model in equation (1), relaxes the requirement of the independence of the global variations and so that *the global variations G_j may be CORRELATED*.

With such pseudo-canonical timing model, the correlation between two node delays can be evaluated as:

$$cov(n_1, n_2) = \sum_i \sum_j \beta_{1,i}\beta_{2,j} cov(G_i, G_j) \tag{4}$$

Assume that there are N nodes and M global variations in the timing graph, if every node delay can be modeled by the pseudo-canonical format of Equation (3), then every timing variable, including all the node delays and arrival times will then have a *extended pseudo-canonical timing(EPCT)* model as:

$$X = \mu_X + \sum_{i=1}^{N} \alpha_{X,i} R_i + \sum_{j=1}^{M} \beta_{X,j} G_j \tag{5}$$

where R_i and G_j are independent to each other although there will possibly correlations between G_js.

With this equation, both global and path correlations can be handled elegantly. More specifically, global variations are represented by the set of global sensitivity terms $\{\beta_{X,j}\}$, and dependence on path history are represented by non-zero node sensitivity terms $\alpha_{X,k}$.

B. Variation Vector

Equation (5) can be rewritten in a compacted vector format as

$$X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X) = \mu_X + \boldsymbol{\alpha}_X^* \boldsymbol{r} + \boldsymbol{\beta}_X^* \boldsymbol{g} \tag{6}$$

where "*" means transpose and

$$\begin{aligned} \boldsymbol{r} &\equiv [R_1, \cdots, R_N]^* \sim N(\mathbf{0}, \boldsymbol{I}) \\ \boldsymbol{g} &\equiv [G_1, \cdots, G_M]^* \sim N(\mathbf{0}, \boldsymbol{\Sigma}_g) \end{aligned} \tag{7}$$

are mutually independent *local variation vector* and *global variation vector* respectively. $\mathbf{0}$ is a zero vector and $\boldsymbol{\Sigma}_g = E\{\boldsymbol{g}\boldsymbol{g}^*\}$ is the covariance matrix of global variations and generally not equal to the unit matrix $\boldsymbol{I}$ due to the potential correlations existing between global variations. $\boldsymbol{\alpha}_X = [\alpha_{X,1}, \alpha_{X,2}, ..., \alpha_{X,N}]^*$ and $\boldsymbol{\beta}_X = [\beta_{X,1}, \beta_{X,2}, ..., \beta_{X,M}]^*$ are deterministic vectors called *Variation Vector(v.v.)* of X.

Authors in [10] proves the correlation evaluation formula between timing variables represented by the canonical timing model of equation (1). We here prove a similar formula for correlation evaluation between time variables expressed with the EPCT model as equation (5) or (6).

Theorem 1: *Given timing variables $X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X)$ and $Y \sim L(\mu_Y, \boldsymbol{\alpha}_Y, \boldsymbol{\beta}_Y)$, the correlation between them can be evaluated as:*

$$cov(X, Y) = \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_Y + \boldsymbol{\beta}_X^* \boldsymbol{\Sigma}_g \boldsymbol{\beta}_Y \tag{8}$$

Proof: By definition:

$$\begin{aligned} & cov(X, Y) = E\{(X - \mu_X)(Y - \mu_Y)\} \\ = \; & cov(\boldsymbol{\alpha}_X^* \boldsymbol{r}, \boldsymbol{\alpha}_Y^* \boldsymbol{r}) + cov(\boldsymbol{\alpha}_X^* \boldsymbol{r}, \boldsymbol{\beta}_Y^* \boldsymbol{g}) \\ & + cov(\boldsymbol{\alpha}_Y^* \boldsymbol{r}, \boldsymbol{\beta}_X^* \boldsymbol{g}) + cov(\boldsymbol{\beta}_X^* \boldsymbol{g}, \boldsymbol{\beta}_Y^* \boldsymbol{g}) \\ = \; & E\{\boldsymbol{\alpha}_X^* \boldsymbol{r}\boldsymbol{r}^* \boldsymbol{\alpha}_X\} + E\{\boldsymbol{\beta}_X^* \boldsymbol{g}\boldsymbol{g}^* \boldsymbol{\beta}_X\} \\ = \; & \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_Y + \boldsymbol{\beta}_X^* \boldsymbol{\Sigma}_g \boldsymbol{\beta}_Y \end{aligned}$$

where the independence between $\boldsymbol{r}$ and $\boldsymbol{g}$ is applied. ■

For the variance of a time variable, it is easy to get:

Corollary 1: *Given timing variable $X \sim L(\mu_X, \boldsymbol{\alpha}_X, \boldsymbol{\beta}_X)$, its variance is:*

$$\sigma_X^2 = \boldsymbol{\alpha}_X^* \boldsymbol{\alpha}_X + \boldsymbol{\beta}_X^* \boldsymbol{\Sigma}_g \boldsymbol{\beta}_X \tag{9}$$

This corollary is actually the special case when $X = Y$ of theorem 8.

IV. Propagating Mean and Variation Vector

In a timing graph, the mean and variation vectors of a node delay is obtained from technology extraction. A STA algorithm, instead, will take those node's means and variation vectors as its input and calculate the mean and variation vector for all arrival times in the entire circuit.

A. Exploration of Sparsity

Since there are N nodes and M global variations in a timing graph, the length of the local variation $\boldsymbol{\alpha}$ will be N and the length of global variation vector $\boldsymbol{\beta}$ will be M which is the same as the rank of the correlation matrix $\mathbf{\Sigma}_g$. So for every step of arrival time propagation in the timing analysis, the computation complexity will be $O(N + M^2)$. And the overall timing analysis computation will have complexity of $O(N(N + M^2)) \approx O(N^2 + NM^2)$ since there will be N steps of arrival time propagations.

Although the correlation matrix $\mathbf{\Sigma}_g$ will be dense and have rank of M which is the same as the total number of global variation sources. However, while working with benchmark circuits, we noticed that many components in the *variation vectors* have very small values, indicating that their contributions to the overall correlation evaluation is insignificant. By setting these small coefficient to zero, the variation vector will become a sparse vector that contains many zero components.

Motivated by this observation, we developed a novel technique called the *flexible vector format* to exploit the sparsity of the variation vector. For this purpose, a *drop threshold* is selected so that if $\alpha_{X,i}$ or $\beta_{X,j}$ is smaller than this threshold, it is deemed to have small value and will be placed into a drop candidate pool to be pruned from the variation vector representation.

However, dropping $\alpha_{X,i}$ or $\beta_{X,j}$ with small magnitude is the same as applying truncation to the variation vector. In subsequent computations, the quantization error may accumulate, causing non-negligible error. This is a problem that can not be overlooked for large circuits. Our solution to this problem is to lump those components in the drop candidate pool into a single correction term

$$x_{pool} \quad = \quad \sqrt{\sum x^2_{\textit{dropped Components}}} \tag{10}$$

B. Complexity and Path Correlation Length

Using this drop and pool mechanism, the actual number of non-zero terms in $\boldsymbol{\beta}$ will be much smaller than M. Assuming there are m global variation sources for a single node, the average number of non-zero terms in $\boldsymbol{\beta}$ will be either (1) m if the time variable is a node delay; or (2) $M_C = m \times q << M$ if the time variable is an arrival time and q is proportional to the logic depth of the circuit. So, using equation (8) to evaluate the correlation between time variables, the complexity for a single evaluation will have complexity of $O[M_C^2 + \Gamma]$ where the average number of non-zero terms in local variation vector $\boldsymbol{\alpha}$ is Γ.

Using this drop and pool mechanism, what is really dropped in the local variation vector $\boldsymbol{\alpha}$ during computation is then the path correlation. So the length of the local variation vector actually gives a good indication to the extent of path correlation in the circuit. The *path correlation length*(Γ) of the circuit is then defined as the average length of the pruned local variation vectors for a given drop threshold.

Theoretically speaking, the path correlation length will be the number of critical nodes which are in the statistically critical paths. Specifically, If a node is not in any statistically critical paths, its variation will be automatically dropped. On the other hand, if the node is in one critical path but is not statistically important, it will be dropped too. So by studying the variation vector in the circuit output, it is easy to know which path is critical and which node is critical for circuit performance and this is very important information to help designer improve the circuit design.

In real circuits, usually only a few paths are statistically critical and so that only a few nodes in the circuit will survive the variation vector propagation. So the computation complexity of our method will be $O[(\Gamma + M_C^2) \cdot N]$.

Since $\Gamma << N$ and $M_C << M$, the total complexity will then be $O[N]$ and a significant reduction of computation and storage is achieved with the drop and pool mechanism. Compared with the PCA-based approach, the complexity saving is even larger since there the complexity will be $O[M^3 + (M_C + \Gamma)N] \approx O[M^3 + N]$ even the drop and pool mechanism is used. So the *extended pseudo-canonical timing model* actually results in a big saving in computation complexity when the total number of global variations is large.

V. Simulation Results and Discussions

The above described algorithm has already been implemented in C/C++ and tested by ISCAS85 benchmark circuits.

Before testing, however, all benchmark circuits are remapped into a library which has gates of *not, nand2, nand3, nor2, nor3 and xor/xnor*. All library gates are implemented in $0.18\mu m$ technology and their delays are characterized by Monte Carlo simulation with Cadence tools assuming all variation sources, either process variations or operational variations, follow Gaussian distribution.

For illustration purpose, only three parameter variation are considered global: channel length(L), supply voltage(Vdd) and temperature(T). All other variation sources, specified in the $0.18\mu m$ technology file, are assumed to be localized in the considered gate only.

A. Accuracy and Performance

Monte Carlo simulation results with 10,000 repetitions are used as "Golden Value" for each benchmark circuit. Each repetition is a process of static timing analysis by fixing global and node variation into a set of randomly sampled values. The global variations are sampled once for each repetition while node variation for each gate is newly sampled every time when the gate is computed.

To elaborate the importance of including the path reconvergence correlation, A STA methods called *CTM* is implemented with canonical timing model where no path correlations are considered.

Table I summarizes the arrival time distribution parameters at the primary output of each testing circuit from Monte Carlo(M.C.), *CTM* and our method of *EPCTM* which is implemented with extended pseudo-canonical timing model. μ and σ are mean and standard variation of the distribution. $\tau_{97} = \mu + 2\sigma$ is the delay estimation at confidence level of

97%. The accuracy of STA methods compared with Monte Carlo method, is evaluated in Table II.

Circuit	STA Method	CPU Time[s]	Delay Distribution[ps]		
			μ	σ	τ_{97}
c432	M.C.	6.449	1288.8	219.3	1727.5
	CTM	0.010	1348.6	216.0	1780.7
	EPCTM	0.030	1299.0	220.4	1739.9
c499	M.C.	8.182	1073.6	178.9	1431.4
	CTM	0.010	1125.4	178.3	1482.0
	EPCTM	0.030	1084.8	180.5	1445.8
c880	M.C.	14.831	1445.4	266.3	1977.9
	CTM	0.010	1463.1	264.2	1911.5
	EPCTM	0.050	1447.6	264.9	1977.3
c1355	M.C.	19.007	1445.4	251.4	1948.3
	CTM	0.010	1529.4	249.0	2027.4
	EPCTM	0.071	1460.9	250.7	1962.3
c1908	M.C.	35.801	1828.2	327.3	2482.8
	CTM	0.030	1881.7	326.5	2534.7
	EPCTM	0.150	1841.9	328.4	2498.6
c2670	M.C.	72.163	2097.0	382.9	2862.8
	CTM	0.050	2161.8	379.7	2921.2
	EPCTM	0.181	2104.4	382.9	2870.1
c3540	M.C.	84.020	2747.2	498.8	3744.8
	CTM	0.050	2850.3	500.6	3851.5
	EPCTM	0.240	2752.3	502.1	3756.5
c5315	M.C.	140.832	2399.3	441.7	3282.6
	CTM	0.080	2474.1	441.3	3356.7
	EPCTM	0.641	2404.8	442.2	3289.2
c6288	M.C.	114.235	6740.1	1286.8	9313.6
	CTM	0.070	7290.8	1273.1	9836.9
	EPCTM	5.198	6775.9	1275.1	9326.2
c7552	M.C.	202.972	1911.7	348.7	2609.0
	CTM	0.110	1974.3	352.5	2679.3
	EPCTM	0.571	1916.6	353.8	2624.2

TABLE I: Testing Results for ISCAS Benchmarks

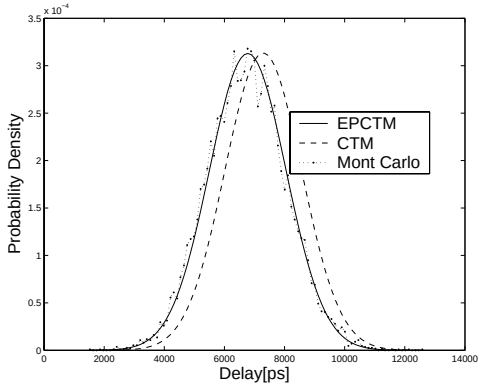
Table II shows that *CTM* have significantly larger error in mean estimation than *EPCTM*. This is reasonable because *CTM* will overestimate the mean at every MAX operation due to smaller correlation considered and this over estimation is accumulated through distribution propagation. It is also interesting to notice that *CTM* and *EPCTM* give similar accuracy in variance estimation. This is possibly because of the fact that the variance is dominated by global variation in the tested cases.

Of course, Monte Carlo simulation gives the best STA results but with big runtime penalty. *EPCTM* runs order-of-magnitude faster but can provide both mean and variance estimation almost as accurate as Monte Carlo does if most of the path correlations are considered as the cases of *EPCTM* shown in Table I and II.

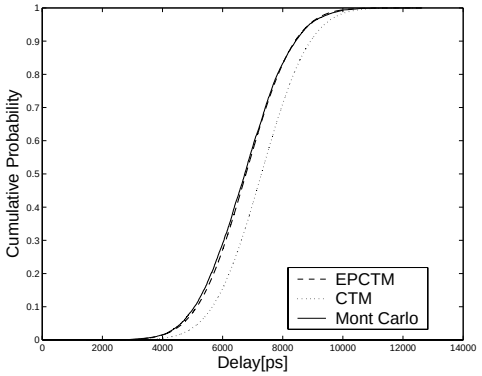
To further elaborate the accuracy of *EPCTM*, Figure 2 shows the $p.d.f.$ and $c.d.f.$ for circuit c6288 from three methods: Mont Carlo and two methods of *EPCTM* and *CTM*. Apparently enough, *EPCTM* shows excellent accuracy since most path correlation is considered.

B. Performance and Path Correlation Length

It has been mentioned in Section IV-B that path correlation length (Γ) is an interesting macro property of the simulated circuit and gives a good indication of the extent of the path correlation existing in that circuit. For the above ISCAS



(a) $p.d.f.$ from Three Methods



(b) $c.d.f.$ from Three Methods

Fig. 2: $p.d.f.$ and $c.d.f.$ comparison for c6288 from three methods

Circuit	Mean Error($\delta\mu$)			Variance Error($\delta\sigma$)	
	CTM	EPCTM	Improve	CTM	EPCTM
c432	4.64%	0.79%	5.9x	1.50%	0.50%
c499	4.82%	1.04%	4.6x	0.34%	0.89%
c880	1.22%	0.15%	8.1x	0.79%	0.53%
c1355	5.81%	1.07%	5.4x	0.95%	0.28%
c1908	2.93%	0.75%	3.9x	0.24%	0.27%
c2670	3.09%	0.35%	8.8x	0.84%	0.00%
c3540	3.75%	0.19%	19.7x	0.36%	0.66%
c5315	3.12%	0.23%	13.6x	0.09%	0.11%
c6288	8.17%	0.53%	15.4x	1.06%	0.65%
c7552	3.27%	0.25%	13.1x	1.09%	1.46%

TABLE II: Distribution Errors

circuits, the path correlation length(Γ) at drop threshold of 1% is summarized in Table III where the run time improvement is also shown when EPCTM is compared with Monte Carlo.

From Table III, we can firstly conclude that the correlation length Γ is much smaller than the circuit size and basically independent on the circuit size since it remains about $10-20$ when circuit size changes dramatically. This observation helps the conclusion we made before about the complexity reduction of our method by using the technique of flexible vector format.

Secondly, the only exceptional high path correlation length among the tested circuits happens with the circuit c6288 which is known as a 16-bit array multiplier. Since there are large amount of equal delay paths in the circuit, large

Name	c432	c499	c880	c1335	c1908
Gate Counts	280	373	641	717	1188
Γ	22.0	11.1	14.2	19.3	27.0
CPU Improve	217x	273x	297x	268x	239x
Name	c2670	c3540	c5315	c6288	c7552
Gate Counts	2004	2485	3865	2704	5355
Γ	15.4	21.2	14.4	80.9	16.0
CPU Improve	399x	350x	220x	22x	355x

TABLE III: Path Correlation Length and Runtime Improvement over Monte Carlo

path correlation length is natural: Few node variation can be dropped due to the equal importance.

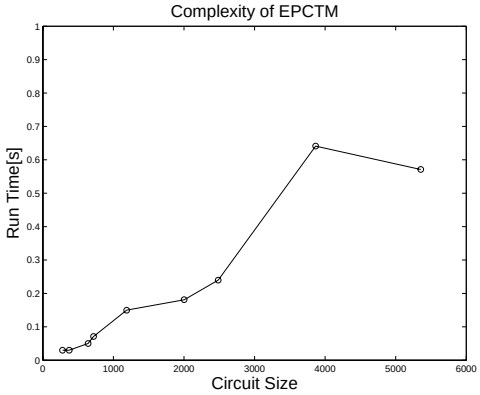


Fig. 3: Run time Complexity of EPCTM

Shown in the figure 3 is the run time complexity of the proposed timing algorithm EPCTM where the run time and circuit size of all circuits except c6288 are shown. From the figure, it is clear that the run time is almost linear with respecting to the circuit size even when the circuit size changes dramatically. This result clearly demonstrates our complexity discussion in section IV-B.

To study the relationship between path correlation length and the accuracy of the STA method, An experiment is conducted for circuit c6288 and results are shown in figure 4 where the error in τ_{97} and path correlation length are both plotted against the drop threshold. It is clear that the path correlation length drops sharply when the drop threshold changes slightly from zero and maintain almost constant after that. But the error changes readily when drop threshold changes. This phenomenon proves the efficiency of the drop mechanism introduced in this work since it means we can sacrifice very little accuracy to gain very significant reduction in the path correlation length and so that save significant amount of CPU time since the run time of the proposed STA method is proportional to the path correlation length.

VI. Conclusions

This paper presents a novel method for block-based statistical timing analysis. Applying the generally accepted Gaussian assumption, we firstly disclose that the MAX operation can be approximated by linear supposition of its inputs. Secondly we extend the commonly used canonical timing model into a

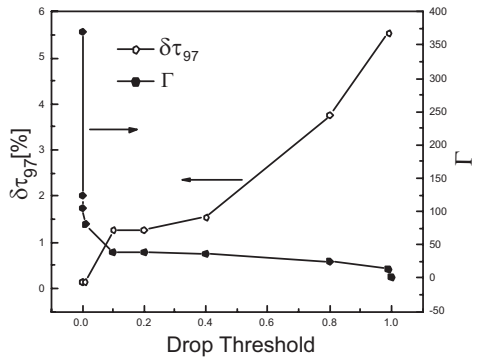


Fig. 4: Path correlation length(Γ) and Error in 97% delay ($\delta\tau_{97}$) when drop threshold changes

vectorized format, variation vector. We also disclose a novel method to decompose correlated timing variables into independent ones to simplify computation. With these theoretical progress, we are able to evaluate and propagate the global and path correlation systematically in the circuit timing graph.

We also design a novel algorithm which treat both global and path correlation simultaneously and systematically. This algorithm, with the help with a new flexible vector format achieves high accuracy and high performance at the same time as tested by ISCAS circuits and compared with Monte Carlo results.

References

[1] J.-J. Liou, A. Krstic, L.-C. Wang, and K.-T. Cheng, "False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation," *Design Automation Conference, 2002. Proceedings. 39th*, pp. 566 – 569, June 2002.

[2] M. Orshansky, "Fast computation of circuit delay probability distribution for timing graphs with arbitary node correlation," *TAU'04*, Feb 2004.

[3] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," *Design Automation Conference, 2002. Proceedings. 39th*, pp. 556 – 561, June 2002.

[4] A. Agarwal, D. Blaauw, V. Zolotov, S. Sundareswaran, M. Zhao, K. Gala, and R. Panda, "Statistical delay computation considering spatial correlations," *Design Automation Conference, 2003. Proceedings of the ASP-DAC 2003. Asia and South Pacific*, pp. 271 – 276, Jan 2003.

[5] A. Agarwal, V. Zolotov, and D. Blaauw, "Statistical timing analysis using bounds and selective enumeration," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 9, pp. 1243 –1260, Sept 2003.

[6] C. Visweswariah, K. Ravindran, and K. Kalafala, "First-order parameterized block-based statistical timing analysis," *TAU'04*, Feb 2004.

[7] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," *Computer Aided Design, 2003 International Conference on. ICCAD-2003*, pp. 900 – 907, Nov 2003.

[8] S. Bhardwaj, S. B. Vrudhula, and D. Blaauw, "τau: Timing analysis under uncertainty," *ICCAD'03*, pp. 615–620, Nov 2003.

[9] A. Devgan and C. Kashyap, "Block-based static timing analysis with uncertainty," *ICCAD'03*, pp. 607–614, Nov 2003.

[10] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single pert-like traversal," *ICCAD'03*, pp. 621–625, Nov 2003.