Block Based Statistical Timing Analysis with Extended Canonical Timing Model

***Abstract*— Block based 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 handling both types of correlations simultaneously.**

In this paper, we present a novel statistical timing algorithm, $AMECT$ (Asymptotic MAX/MIN approximation & Extended Canonical Timing model), that produces accurate timing estimation by handling *both* types of correlations simultaneously. An extended canonical timing model is developed to evaluate and decompose correlations between arbitrary timing variables. And an intelligent pruning method is designed enabling trade-off runtime with accuracy.

Tested with ISCAS benchmark suites, AMECT shows both high accuracy and high performance compared with Monte Carlo simulation results: with distribution estimation error $< 1.5\%$ while with around 350X speed up on a circuit with 5355 gates.

I. Introduction

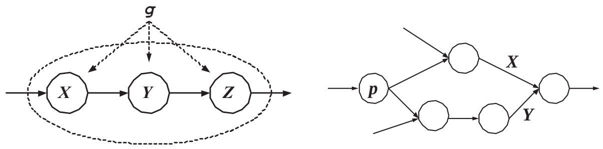
It is well-known that the timing performance of of deep-submicron micro-architecture will be dominated by several factors. IC manufacturing process parameter variations will cause device and circuit parameters to deviate from their designed value. Low supply voltage for low-power applications will reduce 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 approaches 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 worst-case computation complexity that grows exponentially with respect to the circuit size, and hence is difficult to be 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]) where all internal timing 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 variations* ([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)

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, named **AMECT**, 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 independent random variables. A *variation vector*, consisting of all these weights, is then used to explicitly represent *both* global and path correlation information.
- We further explore the sparse structure of the variation vector and develop a *flexible vector format* to dynamically drop non-significant entries of the variation vector and so that significantly curtail the amount of storage and computation required to implement **AMECT**.

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;Section IV is the detailed algorithm and technique to reduce computation complexity. Section V presents a real implementation of **AMECT** in C/C++ and the testing result with ISCAS85 benchmark suites; Section VI gives the conclusions.

II. A Brief Review of Previous STA Algorithms

In timing analysis field, the circuit is modeled as a *timing graph*, which is a directed acyclic graph(DAG) where each delay source, either logic gates or interconnect segments, is represented as a *node*. Each node connects to other nodes through some input and output *edges*. Nodes and edges are called *delay elements*. Each node is assigned with a *node delay* representing the delay incurred in the corresponding logic gates or interconnect segments. The *edge delay*, a short term of signal arrival time at the edge, represents the cumulative timing delays upto and including the node that feeds into the edge. The *history* or *path history* of the edge delay is then defined as the set of node delays through which the signal arrives at this edge ever passes.

Different from classical timing analysis, the statistical timing analysis models delay elements 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 edge delay distribution at the primary output of the circuits knowing all internal node delay distributions. This is accomplished through two operators [5]:

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

In the ADD operation, if both input delay elements X and Y are Gaussian random variables, then $Z = X + Y$ will also be a Gaussian random variable whose mean and variance are:

$$\begin{aligned} \mu_Z &= \mu_X + \mu_Y && (1) \\ \sigma_Z^2 &= \sigma_X^2 + \sigma_Y^2 + 2cov(X,Y) && (2) \end{aligned}$$

where $cov(X,Y) = E\{(X-\mu_X)(Y-\mu_Y)\}$ is the covariance between X and Y.

Due to the nonlinearity, the output delay element of the MAX operator, $Z = max(X,Y)$, will not have Gaussian distribution even if both inputs are Gaussian random variables. For this case, Clark [11] in 1961 derived the first and second moments of the distribution of $max(X,Y)$ assuming X and Y are Gaussian,

$$\begin{aligned} \mu_Z &= \mu_X \cdot Q + \mu_Y(1-Q) + \theta P && (3) \\ \sigma_Z^2 &= (\mu_X^2 + \sigma_X^2)Q + (\mu_Y^2 + \sigma_Y^2)(1-Q) \\ &\quad + (\mu_X + \mu_Y)\theta P - \mu_Z^2 && (4) \end{aligned}$$

where $\theta^2 = \sigma_X^2 + \sigma_Y^2 - 2cov(X,Y)$. P and Q are $p.d.f.$ and $c.d.f.$ of standard Gaussian distribution at $\lambda = (\mu_X - \mu_Y)/\theta$:

$$P(\lambda) = \frac{1}{\sqrt{2\pi}} exp(-\frac{\lambda^2}{2}) \quad ; \quad Q(\lambda) = \int_{-\infty}^{\lambda} P(x)dx$$

An intuitive solution to the non-linear problem of MAX operator is to use a Gaussian *p.d.f.* to approximate the MAX output such that the first two moments of the Gaussian *p.d.f.* match those derived by Clark. This approach has been adopted in [6], [10]. Nonetheless, they fail to address the issue of path correlations among delay elements.

A. Canonical Timing Model

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

$$n_i = \mu_i + \alpha_i R_i + \sum_{j=1} \beta_{i,j} G_j \qquad (5)$$

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 between any two node delays, n_i and n_k, can be easily evaluated as:

$$cov(n_i, n_k) = E\{(n_i - \mu_i)(n_k - \mu_k)\} = \sum_j \beta_{i,j}\beta_{k,j} \qquad (6)$$

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

B. Existing Method for Handling Correlations

Delay elements in a timing graph, including node delays and edge delays, may become correlated due to sharing global variations and/or common path histories. Multiple methods handling these correlations have been proposed to get more accurate STA estimation.

In [6], [7], [10], the canonical timing model of Equation (5) is directly applied into the edge delays in a timing graph. So it is implicitly assumed that edge delay only depends on global variations and no path correlation occurs in the timing graph. This method will work well apparently only when global variation dominates the correlations in the timing graph but will have severe problem where path correlation is important.

In [9], a common node detection procedure is introduce to deal with the path correlation. This method assumes that if two edge delays, X and Y, ever pass a common node whose output edge delay is W, then $X = X' + W$ and $Y = Y' + W$. Operation $max(X, Y)$ is then done as $W + max(X' + Y')$. This is not a good approximation since X and Y usually don't have such a strong dependence on W. A counter example is illustrated in Figure 2 where both X and Y are theoretically dependent on W. But practically speaking, X will be independent on W if $U >> W$ and similarly Y will be independent on W if $V >> W$.

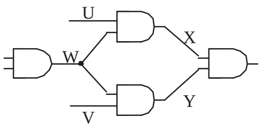


Fig. 2: Example to Fail Common Node Approach

To the best of our knowledge, existing STA methods have yet to offer a solution to deal with the correlation problem caused by *both* global variations and path reconvergence.

III. Variation Vector

The authors in [6] propose the use of *tightness* to retain global correlation information through the nonlinear MAX operation. The output global sensitivities from a MAX operation is treated as a tightness-based supposition of the inputs global sensitivities. This method is valuable since it hints to use linear supposition as the approximation of nonlinear MAX operation:

If $Z = max(X, Y)$ and X and Y are Gaussian random variable, then $Z = t_X X + t_Y Y$ and the $tightness$ of X and Y are defined as

$$t_X = 1 - t_Y = \Psi(\frac{\mu_X - \mu_Y}{\theta})$$

where $\theta^2 = \sigma_X^2 + \sigma_Y^2 - 2cov(X, Y)$.

The canonical timing model of Equation (5) is a powerful tool to represent the numerous delay elements for a given circuit. However, in its original format, it can only handle node delay correlations caused by global variations. In this work, we propose an *extended canonical timing model* that is capable of capture *all* the correlation, either global or path correlation, between any pair of delay elements in the circuit be it a node delay or an edge delay.

Assume that there are N nodes and M global variations in the timing graph, if every node delay can be modeled by the canonical format of Equation (5), then every delay element, including all the node delays and edge delays will then have the extended canonical timing expression as:

$$X = \mu_X + \sum_{i=1}^{N} \alpha_{X,i} R_i + \sum_{j=1}^{M} \beta_{X,j} G_j \tag{7}$$

If X is a node delay, then it will automatically have the extended canonical timing format because the original canonical timing Equation (5) is a subset of Equation (7) in that for k^{th} node delay, only one $\alpha_{X,k}$ has non-zero value while all other $\alpha_{X,i \neq k}$ are set to zero.

If $X = A + B$ and delay elements A, B fit Equation (7) then X must have extended canonical timing format.

If $X = max(A, B)$ given that delay elements A, B fit Equation (7), using the method proposed in [6], $X = t_A A + t_B B$ where t_A and t_B are $tightness$ of A and B. So X will still have canonical timing format of Equation (7).

Any delay element, if it is not a node delay, can ultimately be expressed as the result of one or multiple steps of ADD and/or MAX operations from node delays. So based on the above three assertions, the mathematical induction principle guarantees that all delay elements will have the extended canonical format of Equation (7).

The extended canonical format of Equation (7) can be rewritten in a compacted vector format as

$$X = \mu_X + \boldsymbol{x}^T \boldsymbol{b} \tag{8}$$

where

$$\boldsymbol{b} \equiv [R_1, \cdots, R_N, G_1, \cdots, G_M]^T$$

is a random vector consisting of zero-mean, unity variance independent Gaussian random variables and

$$\boldsymbol{x} \equiv [\alpha_{X,1}, \cdots, \alpha_{X,N}, \beta_{X,1}, \cdots, \beta_{X,M}]^T$$

is a deterministic vector and is the *Variation Vector(v.v.)* of X.

So Each delay element(X) in a circuit will be uniquely represented by its mean(μ_X) and variation vector($\boldsymbol{x}$), noted as

$$X = X(\mu_X, \boldsymbol{x}) \tag{9}$$

With equation (7), both global and path correlations can be handled uniformly. 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 $\alpha_{X,k}$.

From definition, it is easy to verify the following properties for variation vector: Assuming k and c are constants and $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{z}$ are variation vectors of delay elements X, Y, Z.

(1) X and $X + c$ have the same variation vector of $\boldsymbol{x}$;
(2) If $Z = X + Y$ then $\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y}$;
(3) If $Z = kX$, then $\boldsymbol{z} = k\boldsymbol{x}$.
(4) $\sigma_X^2 = \boldsymbol{x}^T \cdot \boldsymbol{x} = ||\boldsymbol{x}||$
(5) $cov(X, Y) = \boldsymbol{x}^T \cdot \boldsymbol{y} = \boldsymbol{y}^T \cdot \boldsymbol{x}$

Property (1) indicates that variation vector remains unchanged if a constant is added to the corresponding delay element since variation vector contains only the variance information of the delay element while the added constant only affects the mean of the delay element. Properties (2) and (3) are the basis of variation vector propagation discussed later. Properties (4) and (5) make variation vector an convenient and systematic way to evaluate the variances and correlations for any delay elements.

IV. Propagating Mean and Variation Vector

In a timing graph, the mean and variation vector of node delays are obtained from technology extraction. To get orthogonality required by the canonical timing representation of delay elements, Principle Component Analysis may be conducted after extraction.([10]) But this is done only once for a specific technology and so that is not considered as a part of STA. Our STA algorithm, **AMECT**, on the other hand, will take those node's means and variation vectors as its input and calculate edge's mean and variation vector for the entire circuit.

A. Algorithm for ADD and MAX Operations

Through an ADD operation

$$Z(\mu_Z, \boldsymbol{z}) = X(\mu_X, \boldsymbol{x}) + Y(\mu_Y, \boldsymbol{y})$$

the mean and variation propagation is straightforward:

$$\mu_Z = \mu_X + \mu_Y \quad (10)$$

$$\boldsymbol{z} = \boldsymbol{x} + \boldsymbol{y} \quad (11)$$

It is very easy to verify the consistency between this variation vector approach and Equation (2).

The mean and variation vector propagation through MAX operation,

$$Z(\mu_Z, \boldsymbol{z}) = max\{X(\mu_X, \boldsymbol{x}), Y(\mu_Y, \boldsymbol{y})\}$$

is similar if we apply the method proposed in [6]. From Clark's Equations (3 and 4), it is easy to calculate μ_Z and σ_Z for $Z = max(X, Y)$. And the tightness, as defined in [6], $t_X = 1 - t_Y = \Psi(\lambda)$. So the final result of variation vector for $Z = max(X, Y)$ will be

$$\boldsymbol{z} = t_X \boldsymbol{x} + (1 - t_X)\boldsymbol{y} \quad (12)$$

If there are more than two delay elements involved in the MAX operation, then MAX is done iteratively by MAX two delay elements at each iteration.

B. Exploration of Sparsity

Since there are N nodes in a timing graph, and each node delay as well as corresponding edge delay will have a $N+M$ dimensional variation vector, the total computation and storage required will be $O(N(N+M)) \approx O(N^2)$. However, while working with benchmark circuits, we noticed that many components in the variation vector have very small values, indicating that their contributions to the overall variance is insignificant. By setting these small coefficients 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. In particular, we focus on curtailing node sensitivity part of the variation vector, $\alpha_{X,i}$, that has small values. For this purpose, a *drop threshold* is given so that if $\alpha_{X,i}$ 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 $\alpha_{X,i}$ 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

$$x_{pool} = \sqrt{\sum x^2_{\textit{dropped Components}}} \quad (13)$$

When two variation vectors merge through either ADD or MAX operation, their pooling components are assumed to be independent. Hence,

$$z_{pool}(ADD) = \sqrt{x^2_{pool} + y^2_{pool}} \quad (14)$$

$$z_{pool}(MAX) = \sqrt{\rho^2 x^2_{pool} + (1-\rho)^2 y^2_{pool}} \quad (15)$$

C. Complexity and Path Correlation Length

Using this drop and pool mechanism, what is really dropped during computation is then the path correlation. So the length of the 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 to be the average length of node part of the pruned variation vectors for a given drop threshold.

With this notation, the computation complexity can be reduced from $O(N^2)$ down to $O[(\Gamma + M) \cdot N]$. Simulation results indicated that $\Gamma << N$ and is not a function of N. Hence it represents a significant reduction of computation and storage.

V. Simulation Results and Discussions

Our STA algorithm, **AMECT**, 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*. Table I summarizes the gate count for each test circuit after gate re-mapping.

Name	c432	c499	c880	c1335	c1908
Gate Counts	280	373	641	717	1188
Name	c2670	c3540	c5315	c6288	c7552
Gate Counts	2004	2485	3865	2704	5355

TABLE I: Gate Count for ISCAS85 Benchmark Circuits

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. Furthermore we don't address the spatial dependency of the gate delays just for demonstration purpose. In real life, gate delay parameters are position dependent but our method is still applicable.

A. *Accuracy and Performance*

Extensive Monte Carlo simulations 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.

Table II summarizes the edge delay distribution parameters at the primary output of each testing circuit from Monte Carlo(M.C.) and two flavors of **AMECT** STA methods. For comparison purpose, a fourth STA method *NoCorr* with no correlation considered is also implemented and simulated. 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 summarized in Table III.

The drop threshold in **AMECT** will determine the level at which the path correlation is considered. Method *HighAccu* is the high accuracy version of **AMECT** when drop threshold is set to as small as 1% and most path correlations are considered while method *HighPerf* is the high performance version of **AMECT** with drop threshold of 100% and only global correlation is considered.

From Table III, it is very clear that method *NoCorr* fails to give reasonable variance estimation because no correlation is considered. This is a good example highlighting the importance of correlations in STA. It is also notable that *NoCorr* can still have fairy reasonable mean estimation which tells that the mean delay is not so sensitive to the correlation. This interesting phenomenon may come from the ADD operation whose variance is very sensitive to the input correlation.

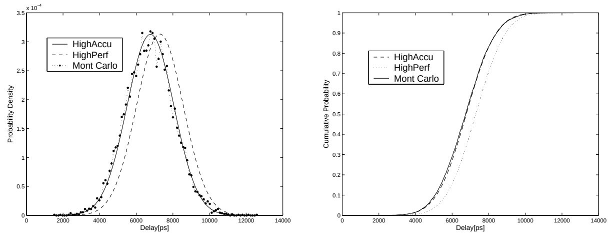
Table III also shows that *HighPerf* have significantly larger error in mean estimation than *HighAccu*. This is reasonable because *HighAccu* will overestimate the mean at every MAX operation due to smaller considered correlation and this mean overestimation is accumulated through all steps of distribution propagation. It is also interesting to notice that *HighPerf* and *HighAccu* 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. **AMECT** 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 *HighAccu* shown in Table II and III. If the accuracy on mean estimation can be relaxed, then the drop threshold can be higher and **AMECT** will give some mean overestimation but with better performance in runtime. In another word, **AMECT** is parameterized by the drop threshold and can be used to trade-off accuracy with performance in some circumstances.

Circuit	Mean Error($\delta\mu$)			Variance Error($\delta\sigma$)		
	HighAccu	HighPerf	NoCorr	HighAccu	HighPerf	NoCorr
c432	0.79%	4.64%	8.05%	0.50%	1.50%	89.8%
c499	1.04%	4.82%	6.99%	0.89%	0.34%	88.5%
c880	0.15%	1.22%	1.81%	0.53%	0.79%	93.8%
c1355	1.07%	5.81%	6.32%	0.28%	0.95%	95.0%
c1908	0.75%	2.93%	3.66%	0.27%	0.24%	91.8%
c2670	0.35%	3.09%	4.58%	0.00%	0.84%	94.3%
c3540	0.19%	3.75%	4.10%	0.66%	0.36%	95.3%
c5315	0.23%	3.12%	6.06%	0.11%	0.09%	92.8%
c6288	0.53%	8.17%	8.68%	0.65%	1.06%	98.8%
c7552	0.25%	3.27%	6.05%	1.46%	1.09%	92.5%

TABLE III: Distribution Error Respecting to Monte Carlo Results

To further elaborate the accuracy of **AMECT**, Figure 3 shows the $p.d.f.$ and $c.d.f.$ for circuit c6288 from three methods: Mont Carlo and two methods of **AMECT**(*HighAccu* and *HighPerf*). Apparently enough, **AMECT** shows excellent accuracy if most path correlation is considered as in method *HighAccu*.



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

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

Fig. 3: $p.d.f.$ and $c.d.f.$ comparison for c6288 from three methods

Name	c432	c499	c880	c1335	c1908
Γ	22.0	11.1	14.2	19.3	27.0
Name	c2670	c3540	c5315	c6288	c7552
Γ	15.4	21.2	14.4	80.9	16.0

TABLE IV: Path Correlation Length(Γ) at 1% of Drop Threshold

B. *Path Correlation Length*

It has been mentioned in Section IV-C that path correlation length (Γ) provided by **AMECT** 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 circuits, the path correlation length(Γ) at drop threshold of 1% is summarized in Table IV.

From Tale IV, 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 **AMECT** 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

Circuit	STA Method	CPU Time[s]	Delay Distribution[ps]			Circuit	STA Method	CPU Time[s]	Delay Distribution[ps]		
			μ	σ	τ_{97}				μ	σ	τ_{97}
c432	M.C.	6.449	1288.8	219.3	1727.5	c2670	M.C.	72.16	2097.0	382.9	2862.8
	HighAccu	0.030	1299.0	220.4	1739.9		HighAccu	0.181	2104.4	382.9	2870.1
	HighPerf	0.010	1348.6	216.0	1780.7		HighPerf	0.050	2161.8	379.7	2921.2
	NoCorr	0.000	1392.6	22.3	1437.1		NoCorr	0.020	2193.1	22.0	2237.1
c499	M.C.	8.182	1073.6	178.9	1431.4	c3540	M.C.	84.02	2747.2	498.8	3744.8
	HighAccu	0.030	1084.8	180.5	1445.8		HighAccu	0.240	2752.3	502.1	3756.5
	HighPerf	0.010	1125.4	178.3	1482.0		HighPerf	0.050	2850.3	500.6	3851.5
	NoCorr	0.000	1148.6	20.5	1189.5		NoCorr	0.020	2859.7	23.4	2906.5
c880	M.C.	14.83	1445.4	266.3	1977.9	c5315	M.C.	140.8	2399.3	441.7	3282.6
	HighAccu	0.050	1447.6	264.9	1977.3		HighAccu	0.641	2404.8	442.2	3289.2
	HighPerf	0.010	1463.1	264.2	1911.5		HighPerf	0.080	2474.1	441.3	3356.7
	NoCorr	0.010	1471.6	16.5	1504.5		NoCorr	0.040	2544.8	31.7	2608.2
c1355	M.C.	19.01	1445.4	251.4	1948.3	c6288	M.C.	114.2	6740.1	1286.8	9313.6
	HighAccu	0.071	1460.9	250.7	1962.3		HighAccu	5.198	6775.9	1275.1	9326.2
	HighPerf	0.010	1529.4	249.0	2027.4		HighPerf	0.070	7290.8	1273.1	9836.9
	NoCorr	0.000	1536.8	12.5	1561.8		NoCorr	0.030	7325.1	14.9	7355.0
c1908	M.C.	35.80	1828.2	327.3	2482.8	c7552	M.C.	203.0	1911.7	348.7	2609.0
	HighAccu	0.150	1841.9	328.4	2498.6		HighAccu	0.571	1916.6	353.8	2624.2
	HighPerf	0.030	1881.7	326.5	2534.7		HighPerf	0.110	1974.3	352.5	2679.3
	NoCorr	0.010	1895.1	27.0	1949.2		NoCorr	0.050	2027.4	26.3	2080.1

TABLE II: Testing Results for ISCAS Benchmarks

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

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 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 *both* global and path correlation in the circuit timing graph.

We also design a novel algorithm, **AMECT** 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 "golden value".

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.

[11] C. Clark, "The greatest of a finite set of random variables," *Operations Research*, pp. 145–162, March 1961.