Generation and Use of Statistical Timing Macro-models considering Slew and Load Variability

Debjit Sinha, Vladimir Zolotov[1], Jin Hu, Sheshashayee K. Raghunathan[2],
Adil Bhanji and Christine M. Casey
IBM Systems, Poughkeepsie, USA [Bangalore, India[2]]
[1]IBM T J Watson Research, Yorktown Heights, USA

ABSTRACT

Timing macro-modeling captures the timing characteristics of a circuit in a compact form for use in a hierarchical timing environment. At the same time, statistical timing provides coverage of the impact from variability sources with the goal of enabling higher chip yield. This paper presents an efficient and accurate method for generation and use of statistical timing macro-models. Results in a commercial timing analysis framework with non-separable statistical timing models demonstrate average performance improvements of 10× when using the model with less than 0.3 picosecond average and 5.5 picosecond maximum accuracy loss, respectively.

Categories and Subject Descriptors

B.8.2 [**Hardware**]: Performance and Reliability—*Performance Analysis and Design Aids*

General Terms

Algorithms, Performance

Keywords

Statistical timing, timing macro-modeling, variability

1. INTRODUCTION

Advances in semiconductor technologies and a continued expansion of desired features in modern sub-45 nanometer Very Large Scale Integrated (VLSI) circuit chip designs have resulted in significantly larger and more complex designs in comparison to prior technology generations. In recent years, modern Application Specific Integrated Circuit (ASIC) designs contain several to tens of millions of gates, and the transistor count in microprocessor chips have surpassed the two billion mark. This design scale has forced a paradigm shift from the traditional *flattened* to a *hierarchical* static timing analysis (STA) and optimization approach for these chips throughout their design life-cycle.

ICCAD'16, November 07-10, 2016, Austin, TX, USA

DOI: http://dx.doi.org/10.1145/2966986.2967043

Given partitions of a large design, a hierarchical timing flow performs STA and optimization on each partition (component) separately. Once a component is *timing-closed*, a timing macro-model is generated for the component [1–4]. The macro-model captures the timing characteristics for that component in a compact form, and is used at the parent or next higher level of hierarchy. This flow has three advantages: (i) it facilitates parallel chip design, wherein several components are designed and optimized at the same time, (ii) it enables faster timing at the parent level(s) of hierarchy by employing the simplified timing macro-models, and (iii) it facilitates usage of a single macro-model for multiple instantiations of the same component at the parent level(s) of hierarchy.

The ability to perform timing closure on a component and generate a macro-model that could be used seamlessly across multiple instances is attractive as it eliminates the overheads of storing a model for each instance of a component. An instance specific model also involves the risk of an instance being "out-of-sync" or obsolete. Furthermore, using a macro-model reduces the overall turnaround time and memory footprint.

With technology scaling, STA tools must account for a multitude of different voltage, temperature, and process (front- and back-end of line) variability effects on timing while being computationally efficient. Statistical static timing analysis (SSTA) captures the impact of variability on timing in a single or few run(s) by encapsulating timing values (e.g., delays, slews, slacks) in a parameterized form, where the impact of each source of variation is quantified as a sensitivity [5–9]. There is naturally a need for statistical timing macro-models in a hierarchical statistical static timing analysis (SSTA) flow. Prior work on statistical macro-modeling [3, 4] extends deterministic macro-modeling, but does not describe the use of these models in the presence of statistical input slews and loads. Goyal *et al.* present a chain rule-based approach to consider the variability in slew and load for only linear statistical models in [10].

This paper presents a method for the generation and usage of statistical timing macro-models. Specifically, three methods of incorporating the variability in the input slew and output load when using a statistical macro-model are presented and compared, where the model is characterized as a function of deterministic input slew and load. The preferred method employs efficient finite differencing that is $O(N^2)$ faster than traditional finite differencing, for N sources of variation. The methods also apply to the usage of *variation aware* extensions of industry standard timing models (e.g. CCS, ECSM). Experimental results in a commercial SSTA framework demonstrate average performance improvements of 10× with less than 0.3 picosecond average and 5.5 picosecond maximum accuracy loss, respectively, and significant accuracy improvements over prior art methods.

2. STATISTICAL TIMING

2.1 Statistical modeling

A set of global *parameters* or sources of variation are denoted as: $X_1, X_2, \ldots, X_N$. A non-separable statistical timing model is used in this work that is a generalization of the model proposed by Zolotov *et al.* [11]. In this model, a statistical timing value T (e.g. delay, slew, arrival-time, slack) is represented as:

$$T = a_0 + \sum_{i=1}^{N} a_i X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij} X_i X_j, \tag{1}$$

where, a_0 is the *mean* of the timing value T and denotes the nominal value of T in the absence of variability, a_i denotes the *sensitivity* of T to a parameter X_i in the absence of any other parametric variability, and a_{ij} denotes a cross-term sensitivity of T to a set of parameters X_i and X_j. Conceptually, a_{ij} denotes the sensitivity of a_i to X_j (or vice-versa). Without loss of generality, it is assumed that all parameters have 0 mean, and a subset of the parameters are modeled as Gaussian (Normal) variables with unit variance (although some parameters like voltage are treated as non-statistical random variables since they do not have a distribution and may be set adaptively during chip operation).

During SSTA, statistical timing modeling of each gate or wire is typically performed via finite differencing [12]. In the presence of multi-corner gate and interconnect models, multiple (corner-based) deterministic gate or wire timing computations are performed, and the results fitted to a statistical model. A corner denotes a vector of deterministic (known) conditions for each parameter. For illustration, given extreme values (base and opposite) for a parameter X_k as (B_k, O_k), corners are mathematically represented as follows:

$$C_* \triangleq [X_1 \rightarrow B_1, \ldots, X_N \rightarrow B_N] \tag{2}$$
(all parameters at base values);

$$C_k \triangleq [X_1 \rightarrow B_1, \ldots, X_k \rightarrow O_k, \ldots, X_N \rightarrow B_N] \tag{3}$$
(parameter X_k at opposite value, all other parameters at base values); and

$$C_{kl} \triangleq [X_1 \rightarrow B_1, \ldots, X_k \rightarrow O_k, \ldots, X_l \rightarrow O_l, \ldots] \tag{4}$$
(parameters X_k and X_l at opposite values, all other parameters at base values).

In the simplest form, the cross-term sensitivity a_{ij} of a statistical timing value T (e.g. delay) to parameters X_i and X_j $(i+1 \leq j \leq N)$ is obtained by computing a linear sensitivity to one of the parameters at different values of the other parameter, and subsequently, calculating a sensitivity of the computed linear sensitivity to the latter parameter. Mathematically:

$$a_{i,C_*} \triangleq \frac{t_i - t_*}{O_i - B_i}, \tag{5}$$

$$a_{i,C_j} \triangleq \frac{t_{ij} - t_j}{O_i - B_i}, \tag{6}$$

$$a_{ij} = \frac{a_{i,C_j} - a_{i,C_*}}{O_j - B_j}, \tag{7}$$

where, t_k denotes a deterministic timing value for T evaluated at corner C_k; and a_{i,C_*} and a_{i,C_j} denote linear sensitivity of T to X_i, computed at values B_j and O_j of X_j, respectively. The sensitivity a_{i,C_*} does not change for different $j \in [i+1, N]$, and is computed once, while a_{i,C_j} is computed for each j.

Since a_i denotes the sensitivity of T to X_i at the nominal or 0 value of all other parameters, it is computed as:

$$a_i = a_{i,C_*} - \sum_{j=i+1}^{N} a_{ij} B_j. \tag{8}$$

The mean value of T is next computed as:

$$a_0 = t_* - \sum_{i=1}^{N} a_i B_i - \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij} B_i B_j, \tag{9}$$

where, t_* denotes a deterministic timing value for T evaluated at corner C_*.

2.2 Statistical timing projection

A statistical timing value T may be *projected* to a given corner C_k by setting each parameter to a deterministic value corresponding to that corner. This yields a deterministic (projected) value t_k. Ideally, this value is identical to the deterministic timing value at corner C_k used during finite differencing to obtain T. For clarity of notation, statistical and deterministic values are denoted in upper case (e.g., T) and lower case (e.g., t_k) notation, respectively. Mathematically, the projected values are computed as:

$$t_* = a_0 + \sum_{i=1}^{N} a_i B_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij} B_i B_j, \tag{10}$$

$$t_k = a_0 + a_k O_k + \sum_{j=k+1}^{N} a_{kj} O_k B_j + \sum_{i=1, i\neq k}^{N} a_i B_i + \sum_{i=1, i\neq k}^{N} \sum_{j=i+1}^{N} a_{ij} B_i B_j, \tag{11}$$

$$t_{kl} = a_0 + a_k O_k + a_l O_l + \sum_{i=1, i\neq\{k,l\}}^{N} a_i B_i + \sum_{j=k+1}^{N} a_{kj} O_k B_j + \sum_{j=l+1}^{N} a_{lj} O_l B_j + \sum_{i=1, i\neq\{k,l\}}^{N} \sum_{j=i+1}^{N} a_{ij} B_i B_j. \tag{12}$$

2.3 Statistical-slew and -load

Circuit (gate or wire) delays and output-slews are typically computed as a function of input slew and output load. During statistical modeling, each (deterministic) corner specific timing calculation t is performed using the input slew (or waveform) and output load at that corner. Specifically, an input statistical slew and output statistical load are *projected* or evaluated at the given corner to deterministic values, and used for the corner specific deterministic timing of the gate or wire. Loads have traditionally not been considered statistical, but back end of line (metal) variation makes the load statistical. In addition, the gate input pin capacitances at the sink(s) of an interconnect are susceptible to process, voltage, and temperature variability. Consequently, each electrical parasitic (resistance and capacitance) of the interconnect may be considered statistical.

From Eqn. (5)-(9), a_0, a_i, and a_{ij} are functions of multiple projected values t, which in turn are functions of the statistical

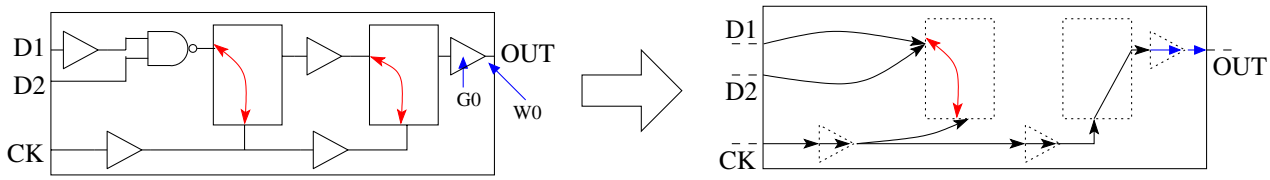

Figure 1: Timing macro-modeling illustrating pruning and compression.

input slew and output load. It follows that accurate statistical modeling must incorporate the impact of variability in the slew and load. The idea also applies to the statistical modeling of timing test guard-times (margins) which are functions of the input slews at the clock and data ends of the test.

3. TIMING MACRO-MODELING

This paper considers a *gate* style of timing macro-modeling, wherein a component's timing characteristic is captured as a gate's timing model (for example, as an industry standard Liberty or ECSM format). In this approach, each gate or wire segment is characterized as a function of input slew s and/or output load l since the potential slew and load at the primary-inputs (PIs) and -outputs (POs), respectively, during use of the model, are unknown at the time of macro-modeling. Alternatively, paths may be characterized instead of individual gate or wire segments. In deterministic macro-modeling, the timing value t_M for each segment or path (e.g. delay and output slew) is stored in the model as $t_M(s,l)$ akin to a industry standard gate timing model.

Timing macro-modeling may employ techniques to reduce the size of the model by performing timing segment pruning and compression. However, multiple internal pins are retained for higher accuracy. Internal register to register paths are often pruned since the slack for these paths are invariants of the non-clock PI arrival times (with clock slew and skew restrictions). Segment compression implies that the model does not retain a segment for each gate or wire in the component, and may instead have segments denoting a path in the original component [1–3]. Figure 1 illustrates timing macro-modeling for a component wherein the internal register to register path is pruned. Timing tests like setup and hold for a register (flip-flop) are represented by a *test* segment ($\leftrightarrow$) as shown in the figure. Selective segments are compressed: for example, paths from the non-clock inputs to the register are compressed, while the clock path is left uncompressed for illustration.

Additional techniques may be employed to reduce the model size and characterization overheads. It is observed that the timing of only the segments for the gate G0 and wire W0 in Figure 1 are dependent on the load at the port OUT. Subsequently, all other segments and paths do not need to be characterized as a function of load. The presented style of macro-modeling thereby differs from industry standard Extracted Timing Models (ETMs) [13], wherein all segments are compressed and the final model contains no internal pins.

3.1 Statistical timing macro-modeling

Statistical timing macro-modeling is an extension of deterministic macro-modeling, wherein the captured timing characteristic of each segment or path of the component additionally contains the impact of variability. In this model, each statistical timing value T_M may be characterized as a function of either "statistical-" slew S and load L as $T_M(S,L)$, or as a function of "deterministic-" slew and load as $T_M(s,l)$. While the former approach enables using the model directly given statistical slews and loads, the model size may grow exponentially since S and L are each a function of the parameters. Consequently, statistical timing macro-models capture statistical timing values as a function of deterministic slew and load. Mathematically, each statistical timing value in the model $T_M(s,l)$ is represented as:

$$\alpha_0(s,l) + \sum_{i=1}^{N} \alpha_i(s,l) X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \alpha_{ij}(s,l) X_i X_j, \qquad (13)$$

where, the mean value $\alpha_0(s,l)$, and each sensitivity and cross-term sensitivity $\alpha_i(s,l)$ and $\alpha_{ij}(s,l)$, respectively, are functions of deterministic input slew s and deterministic output load l. In practice, each of these terms is often stored as a look-up table.

Two cases are highlighted that necessitate the computation of the statistical timing value $T(S,L)$ as a function of statistical slew and load: (i) during use of the macro-model in a parent level hierarchical SSTA flow with statistical slews and loads, and (ii) during segment compression as part of statistical timing macro-modeling. The second case is illustrated for a pair of segments in series that need to be compressed. Each segment's statistical delay $(T1_M^D, T2_M^D)$ and output slew $(T1_M^S, T2_M^S)$ are captured as functions of deterministic input slew as an example. The goal is to compute the statistical delay $T_M^D(s)$ and output slew $T_M^S(s)$ of the compressed segment as:

$$T_M^D(s) = T1_M^D(s) + T2_M^D\left(T1_M^S(s)\right), \qquad (14)$$

$$T_M^S(s) = T2_M^S\left(T1_M^S(s)\right). \qquad (15)$$

Since $T1_M^S(s)$ is a statistical value, the above illustrates the problem of computing $T2_M^D$ and $T2_M^S$ as a function of a statistical value, while the model only contains the information as a function of deterministic value (in this case $T2_M^D(s)$, $T2_M^S(s)$).

Problem definition: Given a statistical model of a segment in the macro-model as $T_M(s,l)$, a statistical input slew S, and a statistical output load L as:

$$S = a_0^S + \sum_{i=1}^{N} a_i^S X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij}^S X_i X_j, \qquad (16)$$

$$L = a_0^L + \sum_{i=1}^{N} a_i^L X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij}^L X_i X_j, \qquad (17)$$

compute the statistical timing value $T(S,L) = f\big(T_M(s,l), S, L\big)$ as:

$$a_0 + \sum_{i=1}^{N} a_i X_i + \sum_{i=1}^{N} \sum_{j=i+1}^{N} a_{ij} X_i X_j, \qquad (18)$$

that is, compute the mean a_0 and each sensitivity and cross-term sensitivity a_i and a_{ij}, respectively.

4. STATISTICAL VALUE COMPUTATION WITH STATISTICAL SLEW AND LOAD

Three approaches to solving the computation problem are presented in this section. The first is based on chain ruling, and is an extension of the work in [10] (cross-terms were not considered). This approach is presented for completeness and for quantitative comparisons. The next two approaches are based on a finite-differenced method and are more accurate, where the latter, more efficient approach is our preferred technique.

4.1 Chain rule-based computation

Chain ruling may be applied to include the variability in the slew and load as part of timing value computation T. Mathematically:

$$\begin{aligned} a_0 &= \alpha_0(a_0^S, a_0^L), \qquad (19)\\ a_i &= \frac{d}{dX_i}\left\{T\big(S(X_i), L(X_i), X_i\big)\right\} \\ &= \frac{\partial T}{\partial S}\frac{dS}{dX_i} + \frac{\partial T}{\partial L}\frac{dL}{dX_i} + \frac{\partial T}{\partial X_i} \\ &= \frac{\partial T}{\partial S}a_i^S + \frac{\partial T}{\partial L}a_i^L + \alpha_i(s_k, l_k), \qquad (20)\end{aligned}$$

where, $\frac{\partial T}{\partial S}$ and $\frac{\partial T}{\partial L}$ are the partial derivatives of the timing value T with respect to input slew and load, respectively; and $\alpha_i(s_k, l_k)$ is the sensitivity obtained from the macro-model. While the partial derivative calculations may be performed by finite differencing, non-linearity effects introduce inaccuracies. In addition, it is unclear at which deterministic values of slew s_k and load l_k should α_i be looked-up from the macro-model. Nominal values are chosen arbitrarily for empirical validation.

Extension to cross-terms involves a second order derivative of the above sensitivity with respect to X_j. Mathematically:

$$\begin{aligned} a_{ij} &= \frac{d}{dX_j}\left\{\frac{d}{dX_i}\left\{T\big(S(X_i, X_j), L(X_i, X_j), X_i, X_j\big)\right\}\right\} \\ &= \frac{d}{dX_j}\left\{\frac{\partial T\big(S(X_j), X_j\big)}{\partial S}\frac{dS}{dX_i} + \frac{\partial T\big(L(X_j), X_j\big)}{\partial L}\frac{dL}{dX_i} + \frac{\partial T}{\partial X_i}\right\} \\ &= \frac{\partial^2 T}{\partial S^2}\frac{dS}{dX_i}\frac{dS}{dX_j} + \frac{\partial^2 T}{\partial S \partial X_j}\frac{dS}{dX_i} + \frac{\partial T}{\partial S}\frac{\partial^2 S}{dX_i dX_j} + \\ &\quad \frac{\partial^2 T}{\partial L^2}\frac{dL}{dX_i}\frac{dL}{dX_j} + \frac{\partial^2 T}{\partial L \partial X_j}\frac{dL}{dX_i} + \frac{\partial T}{\partial L}\frac{\partial^2 L}{dX_i dX_j} + \\ &\quad \frac{\partial^2 T}{\partial X_i \partial X_j} \\ &= \frac{\partial^2 T}{\partial S^2}a_i^S a_j^S + \frac{\partial^2 T}{\partial S \partial X_j}a_i^S + \frac{\partial T}{\partial S}a_{ij}^S + \\ &\quad \frac{\partial^2 T}{\partial L^2}a_i^L a_j^L + \frac{\partial^2 T}{\partial L \partial X_j}a_i^L + \frac{\partial T}{\partial L}a_{ij}^L + \alpha_{ij}(s_k, l_k), \qquad (21)\end{aligned}$$

where, multiple partial derivatives of the timing value to slew and load need to be computed (e.g. $\frac{\partial^2 T}{\partial S^2}$ $\frac{\partial^2 T}{\partial S^2}$, $\frac{\partial^2 T}{\partial S \partial X_j}$, $\frac{\partial T}{\partial S}$); and $\alpha_{ij}(s_k, l_k)$ is the cross-sensitivity obtained from the macro-model. In practice, the second order derivative of T to slew and load may be ignored, but computing the derivative of T with respect to slew (or load) and X_j is error-prone. Nominal values of slew and load are used for s_k and l_k during empirical validation, since the choice is arbitrary in the equation.

Algorithm: Mean and sensitivity computation
Input: $T_M(s, l), S, L$
Output: $T(S, L)$

1: $s_*, l_* \leftarrow S, L$ projected to C_*, respectively
2: $t_* \leftarrow T_M(s_*, l_*)$ projected to C_*
3: foreach $i \in [1, N]$
4: $s_i, l_i \leftarrow S, L$ projected to C_i, respectively
5: $t_i \leftarrow T_M(s_i, l_i)$ projected to C_i
6: $a_{i,C_*} \leftarrow \frac{t_i - t_*}{O_i - B_i}$
7: foreach $i \in [1, N]$
8: foreach $j \in [i+1, N]$
9: $s_{ij}, l_{ij} \leftarrow S, L$ projected to C_{ij}, respectively
10: $t_{ij} \leftarrow T_M(s_{ij}, l_{ij})$ projected to C_{ij}
11: $a_{i,C_j} \leftarrow \frac{t_{ij} - t_j}{O_i - B_i}$
12: $a_{ij} \leftarrow \frac{a_{i,C_j} - a_{i,C_*}}{O_j - B_j}$
13: $a_i \leftarrow a_{i,C_*} - \sum_{j=i+1}^{N} a_{ij} B_j$
14: $a_0 \leftarrow t_* - \sum_{i=1}^{N} a_i B_i - \sum_{i=1}^{N}\sum_{j=i+1}^{N} a_{ij} B_i B_j$
15: $T(S, L) \leftarrow a_0 + \sum_{i=1}^{N} a_i X_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} a_{ij} X_i X_j$

Figure 2: Finite difference-based mean and sensitivity calculation in the presence of statistical slew and load.

4.2 Finite difference (FD)-based computation

Using a traditional finite differencing approach, $T(S, L)$ may be obtained via computations described in Eqns. (5)-(7) for each a_{ij}, Eqn. (8) for each a_i, and Eqn. (9) for the mean a_0. This requires corner specific timing values t_i, t_*, t_j, and t_{ij}. However, corner specific timing models are not natively available in the statistical timing macro-model T_M for the segment. Projections may be used to obtain these values by first projecting S and L to each desired corner, and subsequently, projecting T_M to the same corner using the projected slew and load. The steps for computing $T(S, L)$ using this approach are outlined in Figure 2.

The primary overheads in this approach are the computations of T_M and its projections, since multiple values of T_M need to be evaluated (function of slew and load) as illustrated in lines 2, 5, and 10 of Figure 2. It is observed that the total number of T_M computations is $O(N^2)$. Each component of T_M, specifically α_0, α_i, and α_{ij}, is a function of slew and load, and is often stored as a two dimensional look-up table. From Eqn. (13), for every unique slew and load, the total number of table look-ups (involves interpolation) for a T_M computation is $[1 + N + \{(N-1) + (N-2) + \ldots + 1\}] \approx O(N^2)$. This implies that the total number of table look-ups involving interpolations for computing $T(S, L)$ is $O(N^4)$. In practice, it is observed that this large overhead minimizes the performance advantages of a macro-model over the original design, and motivates the need for an efficient, yet accurate method to compute $T(S, L)$.

4.3 Efficient FD-based computation

An efficient alternative method is presented wherein base values for parameters are considered to be 0 (sigma units) instead of B_i. Conceptually, it implies that the base values are assumed to be the nominal values. The choice of B_i traditionally stems from the availability of corner specific deterministic timing models used during finite differencing. However, given a statistical macro-model, the choice of corners is conceptually arbitrary and should yield the same sensitivity across any choice of base and opposite values for all parameters. Based on Eqns. (10)-(12), the following is obtained:

Algorithm: Efficient mean and sensitivity computation
Input: $T_M(s,l), S, L$
Output: $T(S,L)$

1: $s_*, l_* \leftarrow a_0^S, a_0^L$, respectively
2: $a_0 = t_* \leftarrow \alpha_0(s_*, l_*)$
3: foreach $i \in [1,N]$
4: $s_i, l_i \leftarrow (a_0^S + a_i^S O_i), (a_0^L + a_i^L O_i)$, respectively
5: $a_i \leftarrow \frac{\alpha_0(s_i,l_i)-\alpha_0(s_*,l_*)}{O_i} + \alpha_i(s_i,l_i)$
6: foreach $i \in [1,N]$
7: foreach $j \in [i+1,N]$
8: $s_{ij} \leftarrow a_0^S + a_i^S O_i + a_j^S O_j + a_{ij}^S O_i O_j$
9: $l_{ij} \leftarrow a_0^L + a_i^L O_i + a_j^L O_j + a_{ij}^L O_i O_j$
10: $a_{ij} \leftarrow \frac{\alpha_0(s_{ij},l_{ij})-\alpha_0(s_*,l_*)}{O_i O_j} + \alpha_{ij}(s_{ij},l_{ij}) + \frac{\alpha_i(s_{ij},l_{ij})-a_i}{O_j} + \frac{\alpha_j(s_{ij},l_{ij})-a_j}{O_i}$
11: $T(S,L) \leftarrow a_0 + \sum_{i=1}^{N} a_i X_i + \sum_{i=1}^{N}\sum_{j=i+1}^{N} a_{ij} X_i X_j$

Figure 3: Efficient mean and sensitivity calculation in the presence of statistical slew and load.

$$s_* = a_0^S, \text{(similar for } l_*) \quad (22)$$

$$s_i = a_0^S + a_i^S O_i, \text{(similar for } l_i) \quad (23)$$

$$s_{ij} = a_0^S + a_i^S O_i + a_j^S O_j + a_{ij}^S O_i O_j, \text{ (similar for } l_{ij}) \quad (24)$$

$$t_* = \alpha_0(s_*, l_*), \quad (25)$$

$$t_i - t_* = \{\alpha_0(s_i,l_i) - \alpha_0(s_*,l_*)\} + \alpha_i(s,l)O_i, \quad (26)$$

$$t_{ij} - t_* = \{\alpha_0(s_{ij},l_{ij}) - \alpha_0(s_*,l_*)\} + \alpha_{ij}(s_{ij},l_{ij})O_iO_j + \alpha_i(s_{ij},l_{ij})O_i + \alpha_j(s_{ij},l_{ij})O_j. \quad (27)$$

Substituting these computed values in Eqn. (9), the mean is obtained as:

$$a_0 = t_* = \alpha_0(s_*,l_*) = \alpha(a_0^S, a_0^L). \quad (28)$$

Similarly, substituting the above in Eqns. (5) and (8), each sensitivity value is obtained as:

$$a_i = \frac{t_i - t_*}{O_i} = \frac{\alpha_0(s_i,l_i) - \alpha_0(s_*,l_*)}{O_i} + \alpha_i(s_i,l_i). \quad (29)$$

Finally, each cross-term sensitivity is obtained by substituting the computed values in Eqns. (5)-(7) as:

$$\begin{aligned} a_{ij} &= \frac{(t_{ij}-t_j)-(t_i-t_*)}{O_iO_j} \\ &= \frac{(t_{ij}-t_*)-(t_j-t_*)-(t_i-t_*)}{O_iO_j} \\ &= \frac{t_{ij}-t_*}{O_iO_j} - \frac{a_j}{O_i} - \frac{a_i}{O_j} \\ &= \frac{\alpha_0(s_{ij},l_{ij})-\alpha_0(s_*,l_*)}{O_iO_j} + \alpha_{ij}(s_{ij},l_{ij}) + \\ &\quad \frac{\alpha_i(s_{ij},l_{ij})-a_i}{O_j} + \frac{\alpha_j(s_{ij},l_{ij})-a_j}{O_i}. \end{aligned} \quad (30)$$

Figure 3 summarizes the steps for the efficient mean and sensitivity calculation method for $T(S,L)$. It is observed that no traditional complete projections are required. More importantly, only a single table look-up is required for a_0, and two new look-ups are required for the computation of each a_i. The computation of each a_{ij} requires only four additional table look-ups. The overall complexity is thus reduced by a factor of N^2 from $O(N^4)$ to $O(N^2)$. This significantly improves the performance of SSTA using statistical timing macro-models.

4.4 Test guard-time computation

Timing test (e.g. setup, hold) margins or *guard-times* are modeled as functions of the input slews at the data and clock ends of the test. Given statistical data- and clock-end slews S_D and S_C, respectively, and a statistical timing macro-model $T_M(s_D, s_C)$ wherein a test's guard-time is modeled as a function of deterministic data and clock end slews, the problem is to compute the statistical guard-time considering the statistical slews, that is, $T(S_D, S_C)$. This problem is conceptually identical to the one solved in Sections 4.1-4.3. As an example, substituting $\{S, L\}$ with $\{S_D, S_C\}$ in the steps shown in Figure 3 provides an efficient method for obtaining $T(S_D, S_C)$.

4.5 Handling resistive primary output loads

The approaches presented in this section are illustrated with a statistical representation of the load L. For a gate segment driving a capacitive interconnect (e.g. short wire), the load may denote the sum of interconnect- and downstream gate input pin-capacitances. Representing the load for a resistive interconnect as a lumped capacitance, however, is well known to be inaccurate. Theoretically, the load may be represented as a set of load variables $l_1, \ldots, l_k$, wherein each variable denotes a resistance or capacitance value, and the statistical timing macro-model would be a function of each of these deterministic load variables: $T_M(s, l_1, \ldots, l_k)$. Given variability in the load values, it is straightforward to extend any of three presented approaches to compute $T(S, L_1, \ldots, L_k)$.

In practice, loads are converted to an "effective capacitance" value for gate delay calculation [14]. Considering L a statistical representation of the effective capacitance is another alternative to modeling the load with multiple values.

The output-slew calculation for a gate driving a resistive wire may not use a single effective capacitance representation of the load, however. An iterative approach is used for the output-slew calculation [14]. If statistical load representations are not available for the inner loop of these iterations, using the presented methods require special handling. Additionally, interconnect timing calculations are not based on an effective capacitance. Fortunately, in a macro-model, only a small subset of segments need to be modeled as a function of load as illustrated in Section 3. An approach to avoiding any special handling of these segments is to defer all calculations to the usage of the macro-model. In this approach, the timing of the gate driving a primary output and the primary output interconnect are not included in the macro-modeled. Instead, the gate and wire are preserved in their detailed form. During the use of the macro-model, finite differencing using corner specific gate models and corner specific interconnect reduced order models are used for statistical timing modeling of these gates and wires, respectively.

5. RESULTS

Statistical timing macro-models are generated for a set of designs mapped to a commercial 14 nanometer technology library. Statistical timing macro-modeling of each design is performed without any pruning or compression. Each macro-model thus contains the same number of segments as the original design, and provides a consistent basis for performance comparisons. Statistical timing and macro-modeling are performed in an industrial timing analysis framework on the original designs consisting of standard-cells. Sources of variability include voltage, temperature, process, metal-layer

Table 1: Design sizes and performance comparison

Design	$\|E\|$	Normalized run-time		
		R_O	R_M	R_G
D1	3K	-	-	-
D2	6K	-	-	-
D3	28K	100	11	8
D4	52K	100	13	32
D5	73K	100	12	18
D6	97K	100	10	4
D7	80K	100	13	9
D8	421K	100	11	23
Average		100	11	18

and random variations (> 10 parameters) with cross-terms.

Table 1 presents the set of designs and the number of segments/edges ($|E|$) in the original design being macro-modeled. Using normalized values, the table presents the SSTA run-time of the original design (R_O), the SSTA run-time of an instance of the macro-model (R_M), and the run-time needed to generate the statistical macro-model (R_G) for the larger designs (with $>$ 10K segments). It is observed that the macro-models deliver an order of magnitude of performance improvement for SSTA ($\frac{R_M}{R_O} \approx 11\%$). In practice, commercial macro-models have further reduced sizes due to pruning and compression, and provide additional performance gains. The run-time for the statistical macro-model generation varies between 4% to 32% of the SSTA run-time for the original design. The intent of presenting this data is to provide a qualitative estimate of the model generation overhead ($\frac{R_G}{R_O} \approx 18\%$ on the average). In practice, this value depends on factors like pruning, compression, design topology (presence of more or less latch-to-latch paths), and model accuracy requirements.

Table 2 presents accuracy comparisons of a common statistical timing macro-model when evaluated using three statistical value computation approaches: (i) *No* slew and load variability is considered and all computations are done using the mean value of the slew and load, (ii) Chain ruling (*CR*) is performed to include the variability in the slew and load, and (iii) the presented efficient finite difference (*FD*) approach is used. Absolute mean (E_μ), standard deviation (E_σ), and maximum (E_{Max}) errors (picoseconds) in the slacks across all primary inputs, outputs and timing tests are shown for each design. It is observed that ignoring slew variability causes up to 41.4 picoseconds error ($\approx$ 16% error with respect to a representative 250 picosecond cycle time). The chain ruling-based approach produces smaller errors on the average, but causes up to 18.7 picoseconds error in the worst case. The presented efficient finite difference-based approach is observed to be most accurate with maximum error of 5.5 picoseconds ($\approx$ 2.2%).

Table 2: Macro-model usage accuracy comparison

Design	$E_\mu(ps)$			$E_\sigma(ps)$			$E_{\text{Max}}(ps)$		
	No	CR	FD	No	CR	FD	No	CR	FD
D1	1.4	0.6	0.1	0.7	0.2	0.0	4.3	1.7	0.3
D2	5.2	0.0	0.2	1.1	0.1	0.1	7.3	0.5	0.4
D3	1.5	0.9	0.2	2.5	1.0	0.3	21.3	8.4	3.5
D4	1.6	0.7	0.2	2.1	0.7	0.2	20.5	6.5	2.1
D5	1.7	0.8	0.2	2.8	0.9	0.2	20.4	7.4	1.3
D6	1.5	0.8	0.3	1.8	0.6	0.3	36.6	15.7	4.3
D7	1.0	0.7	0.4	1.3	0.5	0.2	41.4	18.7	5.5
D8	1.2	0.9	0.4	1.1	0.7	0.6	15.5	6.4	4.1
Avg	1.9	0.7	**0.3**	1.7	0.6	**0.2**			
Max							41.4	18.7	**5.5**

6. CONCLUSIONS

This paper presents a method for generation and use of statistical timing macro-models. Specifically, it addresses the challenge of maintaining accuracy during macro-model usage by considering the variability in the input slew as well as the output load, while minimizing computational overheads. Experimental results in a commercial timing analysis framework with non-separable statistical timing models demonstrate average performance improvements of 10× when using the model with less than 0.3 picosecond average and 5.5 picosecond (2.2%) maximum accuracy loss, respectively. Comparisons are performed with alternate approaches which validate the higher accuracy of the presented efficient finite difference-based approach.

7. REFERENCES

[1] S. V. Venkatesh, R. Palermo, M. Mortazavi, and K. A. Sakallah, "Timing abstraction of intellectual property blocks," in *CICC*, 1997, pp. 99–102.

[2] C. W. Moon, H. Kriplani, and K. P. Belkhale, "Timing model extraction of hierarchical blocks by graph reduction," in *DAC*, 2002, pp. 152–157.

[3] A. Bhanji, C. Visweswariah, D. Sinha, G. Ditlow, K. Kalafala, N. Venkateswaran, and S. Gupta, "A hierarchical transistor and gate level statistical timing flow for microprocessor designs," in *TAU*, 2009, pp. 1–4.

[4] B. Li, N. Chen, Y. Xu, and U. Schlichtmann, "On timing model extraction and hierarchical statistical timing analysis," in *IEEE Transactions on Computer-Aided Design, 32(3) March 2013*, pp. 367–380.

[5] A. Agarwal, D. Blaauw, and V. Zolotov, "Statistical timing analysis for intra-die process variations with spatial correlations," in *ICCAD*, 2003, pp. 900–907.

[6] H. Chang and S. S. Sapatnekar, "Statistical timing analysis considering spatial correlations using a single PERT-like traversal," in *ICCAD*, 2003, pp. 621–625.

[7] C. Visweswariah, K. Ravindran, K. Kalafala, S. G. Walker, and S. Narayan, "First-order incremental block-based statistical timing analysis," in *DAC*, 2004, pp. 331–336.

[8] L. Zhang, W. Chen, Y. Hu, and C. C. Chen, "Statistical static timing analysis with conditional linear MAX/MIN approximation and extended canonical timing model," in *IEEE Transactions on Computer-Aided Design, 25(6), pp. 1183 - 1191*, June 2006.

[9] L. Chen, J. Xiong, and L. He, "Non-Gaussian statistical timing analysis using second-order polynomial fitting," in *IEEE Transactions on Computer-Aided Design, vol. 28, no. 1*, 2009, pp. 130–140.

[10] R. Goyal, S. Shrivastava, H. Parameswaran, and P. Khurana, "Improved first-order parameterized statistical timing analysis for handling slew and capacitance variation," in *Proc. Intl. Conf. on VLSI*, 2007, pp. 278–282.

[11] V. Zolotov, D. Sinha, J. Hemmett, E. Foreman, C. Visweswariah, J. Xiong, J. Leitzen, and N. Venkateswaran, "Timing analysis with nonseparable statistical and deterministic variations," in *DAC*, 2012, pp. 1061–1066.

[12] D. Sinha, L. Silva, J. Wang, S. Raghunathan, D. Netrabile, and A. Shebaita, "TAU 2013 variation aware timing analysis contest," in *ISPD*, 2013, pp. 171–178.

[13] *VLSI concepts - ETM basics.* http://www.vlsi-expert.com/2011/02/etm-extracted-timing-models-basics.html.

[14] J. Qian, S. Pullela, and L. T. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gates," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 13, no. 12*, 1994, pp. 1526–1535.