

Accurate Waveform Modeling using Singular Value Decomposition with Applications to Timing Analysis

Anand Ramalingam

Ashish Kumar Singh

Sani R. Nassif

Michael Orshansky

David Z. Pan

Department of Electrical and Computer Engineering, The University of Texas, Austin, TX 78712
Austin Research Laboratory, IBM, Austin, TX 78758

{anandram,asingh,dpan,orshansky}@cerc.utexas.edu, and nassif@us.ibm.com,

ABSTRACT

It is known that ramp-based models are not sufficient for accurate timing modeling. In this paper, we develop a technique that accurately models the waveforms, and also allows a flexible trade-off of accuracy vs. computational and representational cost. The technique is based on Singular Value Decomposition (SVD) and it naturally leads to a more general gate delay model which can be applied in any timing analysis engine with minor modifications. We demonstrate its application in timing analysis by propagating a waveform along a path. When compared with Spice, the proposed model shows good accuracy.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance

Keywords

Waveform modeling, SVD, Timing Analysis

1. INTRODUCTION

For accurate timing analysis one needs to accurately model gate delay. Traditionally, gate delay modeling is performed using a simple ramp approximation of the input and output waveforms. As technology scales into the nanometer regime, a ramp approximation is no longer sufficient [1]. There are several studies in the literature which improve upon ramp approximation [2–8].

The existing approaches can be broadly classified into one of the following three approaches:

- *Improved heuristic model* where authors proposed models other than ramp, e.g. Equivalent waveform model [2], Weibull distribution [4, 5]
- *Data based approaches* where authors have proposed using statistical techniques such as Principal Component Analysis (PCA) [3] or a heuristic approach [6].
- *Change of basis models* where authors have proposed modeling current rather than voltage, e.g. CSM [7–10].

An equivalent waveform modeling at the input has been proposed where weighted least squares fit is employed [2]. The heuristic equivalent waveform has a ramp up to a certain point and an exponential after that. A heuristic for weight

is given by $\frac{\partial v_{out}}{\partial v_{in}}$ which weighs that part of input waveform which affects the output waveform. This is an improvement over the ramp model since the proposed equivalent waveform can model the non-linearity of real waveforms to some extent. An analytical technique to model the waveforms is to approximate the waveform with a cumulative probability distribution function (CDF). The Weibull distribution, a two parameter model, is used to model the waveforms [4]. The two parameters approximately model the slope and shape of the waveform, in contrast to a simple ramp model where there is only one parameter, slope. Thus, Weibull model gives better results since it also models the shape of the waveform. The Weibull model has been extended to include crosstalk noise waveforms [5].

An alternative to analytical approach is the data-based approach which needs a set of all possible waveforms that one will encounter as a starting point. The set of all possible *input* waveforms is generated by collecting waveforms at the output of different interconnect structures one encounters. The set of all possible *output* waveforms is generated by collecting waveforms at the output of all gates in the library under process variations and different environmental conditions [6]. Thus generating the collection of all possible waveforms is a time-consuming pre-processing step but it gets amortized over many runs of the timer. After all possible waveforms have been generated, a set of basis waveforms can be extracted which approximates all possible waveforms using an affine transformation [6]. The basis waveform selection from a set of all possible waveforms is done efficiently using unate covering heuristic. It has been shown that a few basis waveforms are sufficient for accurate waveform modeling.

Current source-models (CSM) based on transistor physics have been proposed to model waveforms [7, 8]. In CSM, the most important feature is the introduction of current source to model the output drive. Thus it can capture the non-linearity of the driver accurately.

Another data-based approach is the principal component analysis (PCA) based waveform modeling [3]. PCA based waveform modeling is a data-based approach in a sense that one generates all possible waveforms that one will encounter. The waveforms are discretized at n equal voltage intervals and recording the time at which certain voltage thresholds are crossed. If the crossing times are treated as random variables then if they are highly correlated, a dimension reduction technique like PCA can be applied to represent a waveform accurately in a reduced $r < n$ uncorrelated space [3].

Our approach builds on the PCA based waveform modeling [3]. The PCA method was described for a single gate and we generalize the method to a library of gates. We use Singular Value Decomposition (SVD) instead of PCA in this paper. The reason is that we are interested in finding the orthogonal basis of the waveforms and SVD provides a simple and direct way to achieve it. Each of these orthogonal basis

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2007, June 4–8, 2007, San Diego, California, USA

Copyright 2007 ACM 978-1-59593-627-1/07/0006 ...\$5.00.

provide a linear combination of the n time points. Interestingly as we will see later on, the first two of these orthogonal basis (linear combinations) can be interpreted as 50% point and slope of the waveform respectively. This allows us to link this approach to current methodologies, and to have it gracefully degrade to a simple ramp approximation when needed.

Our goal in this paper is to show that we do not need n time points to model the waveform since only a few orthogonal basis (linear combinations) of these time points are sufficient to model the waveform accurately.

The contributions of this paper are:

- We provide a rigorous mathematical analysis of waveforms using SVD leading to a generalized gate delay model.
- We link the proposed approach to ramp-based models and show it is a logical extension to current modeling and simulation methods.
- We show how the approach can provide a systematic method for trading off complexity vs. accuracy in the waveform models.
- We generalize and extend PCA approach [3] to a library of gates.

2. DATA BASED MODEL FOR A GATE

In this section we illustrate how to generate a SVD based timing model for a single gate. Later on, we extend our approach to the entire library. Since proposed timing model is based on data, assume that we have a diverse set of waveforms to work with to obtain our timing model. The method for generating diverse set of waveforms is described later in § 6.

A waveform is *discretized* by recording the time points t_k , when the voltages cross $\frac{k-1}{n-1}$, $1 \leq k \leq n$ as shown in Figure 1. In the case of falling waveforms the notation changes, the time points t_k denote the voltages crossing $(1 - \frac{k-1}{n-1})$, $1 \leq k \leq n$. We also assume that the voltages are normalized ($V_{DD} = 1$).

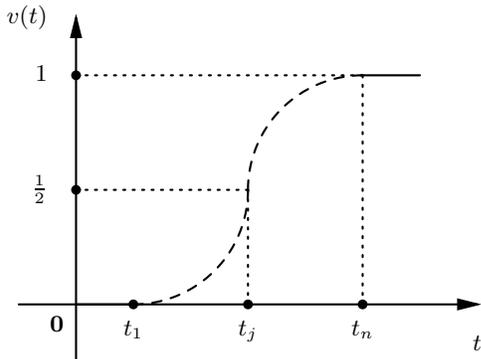


Figure 1: A waveform is discretized and time points are recorded when a signal crosses a certain voltage threshold. Here time points are recorded when the waveform crosses the threshold of $\frac{1}{2}$ and 1.

The collection of output waveforms are discretized and the times at which the voltage thresholds are crossed are recorded. The discretized input and output waveforms can be collected in one single matrix \mathbf{T} :

$$\mathbf{T} = \begin{pmatrix} t_{1,1} & t_{1,2} & \dots & t_{1,n} \\ \dots & \dots & \dots & \dots \\ t_{m,1} & t_{m,2} & \dots & t_{m,n} \end{pmatrix} \quad (1)$$

Note that each row in the matrix \mathbf{T} contains one discretized waveform. This is akin to current procedures except that instead of taking each waveform and approximating it as a ramp, we take *all* the waveforms and develop a new model via the SVD process which we outline next.

3. ANALYZING WAVEFORMS USING SVD

The Singular Value Decomposition (SVD) is the fundamental theorem of Linear Algebra [11]. SVD is defined as [12]:

Definition 3.1 (Singular Value Decomposition). Let $m, n \in \mathbb{N}$ be arbitrary; we do not require $m \geq n$. Given $\mathbf{T} \in \mathbb{R}^{m \times n}$, not necessarily of full rank, a *singular value decomposition* (SVD) of \mathbf{T} is a factorization

$$\mathbf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \quad (2)$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$ is orthonormal, $\mathbf{V} \in \mathbb{R}^{n \times n}$ is orthonormal, and $\mathbf{\Sigma} \in \mathbb{R}^{m \times n}$ is diagonal. In addition, $\mathbf{\Sigma}$ is assumed to have its diagonal entries σ_j nonnegative and in nonincreasing order; that is, $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ where $p = \min(m, n)$.

It can be proved that for any matrix SVD exists [12]. The matrix \mathbf{U} is called the left singular matrix and its columns provide the *orthogonal basis for the columns* of \mathbf{T} . The matrix \mathbf{V} is called the right singular matrix and its columns provide the *orthogonal basis for the rows* of \mathbf{T} . The diagonal elements of $\mathbf{\Sigma}$ are called the *singular values*.

As noted above, the columns of \mathbf{V} provide the orthogonal basis for the *rows* of \mathbf{T} and since each row contains a discretized waveform, the columns of \mathbf{V} turn out to be the orthogonal basis for the waveforms in matrix \mathbf{T} .

Now Eq. (2) can be rewritten by post-multiplying both sides by \mathbf{V} . Since \mathbf{V} is orthonormal ($\mathbf{V}^\top \mathbf{V} = \mathbf{I}$) we get $\mathbf{TV} = \mathbf{U}\mathbf{\Sigma}$. We denote the resultant product matrix as \mathbf{M} called the *moments* matrix because this is another way to represent time points just like an equivalent representation of any function by its moments:

$$\mathbf{M} = \mathbf{TV} = \mathbf{U}\mathbf{\Sigma} \quad (3)$$

The *moments* matrix defined here is a linear combination of time points weighed by the right singular vectors (rsv) $\mathbf{V}_{\cdot j}$:

$$m_{ij} = \mathbf{T}_{i \cdot} \mathbf{V}_{\cdot j} = \sum_{k=1}^n t_{ik} v_{kj} \quad (4)$$

The right singular vectors transform a waveform from time domain $\mathbf{t} = (t_1, t_2, \dots, t_n)$ to moments domain $\mathbf{m} = (m_1, m_2, \dots, m_n)$ through $\mathbf{m} = \mathbf{tV}$ and vice-versa through $\mathbf{t} = \mathbf{mV}^\top$.

This equivalent representation leads to an interesting possibility in the context of timing analysis. If a waveform can be represented accurately using a few moments then by propagating these moments, one can do an accurate waveform analysis instead of propagating all the n time points. Suppose we represent a waveform by r moments, where $r < n$ then the last $n - r$ moments are set to zero. The process of setting the last $n - r$ moments to zero is equivalent to setting the last $n - r$ singular values to zero since zeroing singular value will force the corresponding moment to zero. But zeroing out singular values is equivalent to approximating a matrix \mathbf{T} with another matrix $\tilde{\mathbf{T}}$ having a smaller *rank* and this is proved in Theorem 2. To measure the goodness of approximation, Frobenius norm is used. This norm measures the goodness of fit in a root mean square fashion.

The above discussion can be summarized by saying that the following statements are equivalent.

- Approximating a waveform \mathbf{T}_i using the first r moments.

- Approximating a matrix considering the first r -singular values of matrix \mathbf{T}
- A rank- r approximation of matrix \mathbf{T} in Frobenius norm.

Definition 3.2 (Frobenius Norm). Given $\mathbf{T} \in \mathbb{R}^{m \times n}$, the *Frobenius norm* of \mathbf{T} is defined as

$$\|\mathbf{T}\|_F = \sqrt{\left(\sum_{i=1}^m \sum_{j=1}^n t_{ij}^2\right)} \quad (5)$$

There is an equivalent way to compute the Frobenius norm of a matrix by using the singular values of a matrix which is stated next.

Theorem 1. Given $\mathbf{T} \in \mathbb{R}^{m \times n}$, and its singular values after SVD is given by $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$ where $p = \min(m, n)$. Then the Frobenius norm of \mathbf{T} can be computed by

$$\|\mathbf{T}\|_F = \sqrt{\left(\sum_{i=1}^p \sigma_i^2\right)} \quad (6)$$

Now we are ready to state what is the best approximation to a matrix in the context of Frobenius norm.

Theorem 2. Given $\mathbf{T} \in \mathbb{R}^{m \times n}$, with a singular value decomposition $\mathbf{T} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the best approximation in Frobenius norm to \mathbf{T} by a matrix of rank $k \leq \min(m, n)$ is given by

$$\tilde{\mathbf{T}}_k = \mathbf{U} \mathbf{diag}(\sigma_1, \dots, \sigma_k, 0, \dots, 0) \mathbf{V}^T$$

Thus the idea of having r singular values is equivalent to approximating a matrix \mathbf{T} with another matrix $\tilde{\mathbf{T}}$ having a rank r . As stated earlier, having r singular values is equivalent to having only the first r moments to represent the waveform while discarding the rest of $n - r$ moments. Thus we can define a metric in Frobenius norm to see how well r moments approximate a waveform.

Now define f_r as relative error incurred in Frobenius norm by approximating matrix $\mathbf{T} \in \mathbb{R}^{m \times n}$ by rank- r matrix $\tilde{\mathbf{T}}_r \in \mathbb{R}^{m \times r}$.

$$f_r = \frac{\|\mathbf{T} - \tilde{\mathbf{T}}_r\|}{\|\mathbf{T}\|} = \frac{\sqrt{\left(\sum_{i=k+1}^p \sigma_i^2\right)}}{\sqrt{\left(\sum_{i=1}^p \sigma_i^2\right)}} \quad (7)$$

where $p = \min(m, n)$. We plot the relative error in Frobenius norm f_r in Figure 2. In the figure, one can observe that the error measured by Frobenius norm quickly reduces to a very small quantity when we approximate using the first few moments. A more interesting comparison would be in terms of the absolute values involved. The error in approximating by r moments is given by:

$$\Delta \mathbf{T}_r = \mathbf{T} - \tilde{\mathbf{T}}_r \quad (8)$$

The entry $\Delta t_{r_{ij}}$, represents the error in approximating j th time point in the i th waveform with r moments. For a rank-3 approximation, we found that nearly 90% of the entries in $\Delta \mathbf{T}_r$ had an absolute error which was less than 1 ps. Thus we need a few moments to represent a waveform with a high degree of accuracy.

Next we interpret what the moments mean. We will see that the moments are related to the more familiar notions of 50% time point and slope. Recall from Eq. (4) that moments of a waveform are obtained by the right singular vectors (\mathbf{V}_j) weighing the linear combination of time points of a waveform.

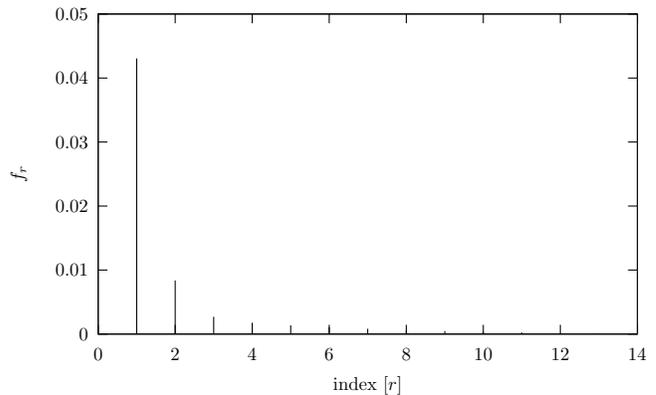


Figure 2: Relative error in Frobenius norm when approximating using the first k moments. Note that using the first 2 moments the approximation is accurate within 1% in Frobenius norm. The data is shown for an inverter and the number of time points is $n = 14$.

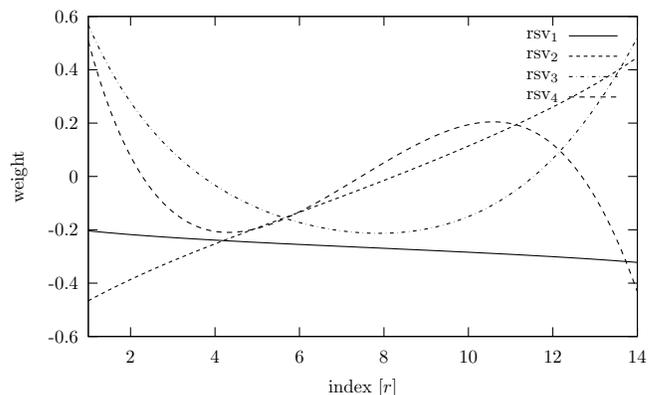


Figure 3: Plot of the first 4 right singular vectors (rsv) obtained on applying SVD to \mathbf{T} in Eq. (1). The number of time points is $n = 14$.

The first right singular vector $\mathbf{V}_{.1}$ (rsv₁), weighs the time points \mathbf{t} nearly *equally* to produce the first moment m_1 :

$$m_1 = \sum_{k=1}^n t_k v_{k1} \approx \frac{1}{n} \sum_{k=1}^n t_k$$

The first moment m_1 can be interpreted as the *average* of all the sample time points and in that sense represents the 50% *time point* of the waveform.

The second right singular vector $\mathbf{V}_{.2}$ (rsv₂), weighs the time points \mathbf{t} nearly *linearly* to produce the second moment m_2 . This can be interpreted as a quantity proportional to the average slope of a linear ramp approximation to the waveform. Note that the first and second moments are only *interpreted* to be the 50% time point and slope respectively and they are not the same in terms of numerical value. The third right singular vector weighs the time points *quadratically* to produce m_3 . While the fourth right singular vector weighs the time points *cubically* to produce m_4 .

4. EXAMPLE OF MOMENTS CALCULATION

We illustrate the calculation of moments by means of an example. After sampling a waveform at $n = 14$ time points,

we get a time vector \mathbf{w}_{t_1} and its corresponding voltage vector \mathbf{w}_v :

$$\mathbf{w}_{t_1}^\top = \begin{bmatrix} 599 & 632 & 656 & 673 & 690 & 705 & 716 \\ 728 & 740 & 752 & 764 & 776 & 788 & 802 \end{bmatrix} \text{ps} \quad (9)$$

$$\mathbf{w}_v^\top = \left[0 \quad \frac{1}{13} \quad \frac{2}{13} \quad \dots \quad \frac{11}{13} \quad \frac{12}{13} \quad 1 \right] \text{V} \quad (10)$$

Now collect m such waveforms, $(\mathbf{w}_{t_i}, i = 1, \dots, m)$ in a matrix \mathbf{T} . On applying SVD Eq. (2) to \mathbf{T} , we obtain the right singular vectors \mathbf{V} . Note that once the SVD analysis is done, the right singular vectors are fixed. For example, the second right singular vector $\mathbf{V}_{.2}$ after the SVD analysis:

$$\mathbf{V}_{.2} = \begin{bmatrix} -0.52 & -0.41 & -0.32 & -0.25 & -0.18 \\ -0.12 & -0.06 & -0.01 & 0.05 & 0.12 \\ 0.18 & 0.24 & 0.31 & 0.39 & \uparrow \end{bmatrix} \quad (11)$$

The second moment (m_2) is given by the dot product of the waveform (\mathbf{w}_{t_1}) and the second right singular vector $(\mathbf{V}_{.2})$:

$$m_2 = \mathbf{w}_{t_1}^\top \mathbf{V}_{.2} = -202.31 \quad (12)$$

The second moment (m_2) was interpreted as quantity whose absolute value approximates the slope. If we define slope as the difference in time at which voltage crosses 0V and 1V then the slope of \mathbf{w}_{t_1} in Eq. (9) turns out to be 203ps which is approximately the same value as $|m_2| = 202.31$ ps.

It is important to note that the *right singular vectors are constant vectors*, for example Eq. (11) shows the second right singular vector. The right singular vectors just weigh the linear combination of time points or in other words transform waveform from time domain to moments domain. Thus only the moments vary depending on the waveform time points and this is shown in Eq. (12), depending on \mathbf{w}_{t_1} we will get different values of moments.

5. EXAMPLE OF SVD BASED TIMING MODEL

To illustrate the proposed timing model we calculate the delay of the simple inverter chain in Figure 4 and compare it with Weibull-based timing model [4]. The Weibull model has two parameters: slope, and shape in addition to the arrival time.

In our characterization, we follow the same strategy as proposed in the Weibull model. We normalize the arrival time of all input waveforms to 0 during characterization and keep the arrival time information separate from the waveform information. In other words, instead of doing SVD on (t_1, t_2, \dots, t_n) we do it on $(t_2 - t_1, \dots, t_n - t_1)$, where t_1 is the arrival time. Due to this normalization we are now left with $n - 1$ time points. This translates to the fact that we have $n - 1$ moments for a given waveform instead of n moments. We found that the new first moment \tilde{m}_1 is equivalent to the old m_2 described in Eq. (12).

Another way to think about it is that we have forced the first moment to be the arrival time instead of the $t_{50\%}$ time point while keeping the rest of the moments. To ensure fairness in comparison, we use two moments in our proposed model in addition to the arrival time.

Now we intuitively introduce the idea of using *moments* to model the waveform propagation across the gates. In § 3, we saw that a few moments are sufficient to characterize a waveform. The first and second moments are interpreted as quantities very similar to 50% time point and slope. Thus if we are using only the first two moments, then the gate delay modeling is equivalent to the ramp based delay modeling. A simple ramp based delay equation is given by:

$$\text{delay} = a_0 + a_1 S_{in} + b_1 S_{in} C_{out} + c_1 C_{out} \quad (13)$$

The moments-based equation can be thought of as an generalization of Eq. (13):

$$t_{\text{arrival}} = a_0 + \sum_{i=1}^2 a_i \tilde{m}_i^{\text{in}} + \sum_{i=1}^2 b_i \tilde{m}_i^{\text{in}} C_{out} + c_0 C_{out} \quad (14)$$

$$\tilde{m}_j^{\text{out}} = a_{0j} + \sum_{i=1}^2 a_{ij} \tilde{m}_i^{\text{in}} + \sum_{i=1}^2 b_{ij} \tilde{m}_i^{\text{in}} C_{out} + c_0 C_{out} \quad (15)$$

where $j = 1, 2$. Note that we have *not* modeled the non-linear capacitance of the fanout gates. The equations were fitted using linear regression [13]. A more detailed discussion about the moment modeling with complex load modeling and the error involved in fitting is described in § 7.

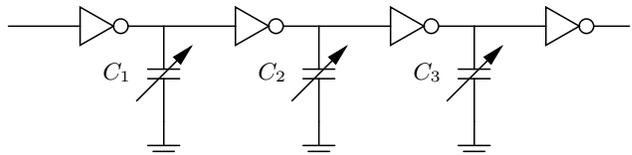


Figure 4: Three stage inverter chain with a single capacitor modeling the load.

We calculated delay for 1000 different values of the (C_1, C_2, C_3) tuple, where $C_i, i = 1, 2, 3$ was randomly sampled from 100 fF to 200 fF. The minimum error in delay using SVD model was 1.2% and the maximum error was 8.7% when compared to Spice. The corresponding statistics for Weibull based model was 1.6% and 9.7%. and for a simple slope based model was 4.9% and 17.6%. It is clear that both Weibull and SVD based timing models are superior to simple slope model and demonstrate the need for more complex delay modeling.

In the next section, we extend our SVD-based timing model to the entire library.

6. DATA BASED MODEL FOR A LIBRARY

In a timing analyzer we generally represent the circuit by a directed graph consisting of gates and the wires that connect them. In forming this representation, we make use of two main abstractions:

- An abstraction of the switching waveform at the input and output of each gate in the circuit, and
- An abstraction for the model describing how these waveforms are changed when they go through gates or wires.

In current methodologies, the first abstraction is the ramp waveform model, and the second is the delay model for gates and wires. Since the waveform for a ramp model is represented by a tuple of delay and slope, we can represent a typical delay models as:

$$(D_{out}, S_{out}) = f((D_{in}, S_{in}), C_{load}, \dots) \quad (16)$$

Note that while f is different for each gate and wire, the representation of the waveform is the *same* for all.

Thus the first step we must perform is the generation of a new *uniform* waveform model that would be valid for all components in a circuit. We do this by taking *all* the gates, which are collected in a gate *library*, and generating a large number of diverse waveforms from them. Mathematically this means that the matrix \mathbf{T} in Eq. (1) contains waveforms of *all* gates in the library instead of a single gate as described in § 2. Both falling and rising waveforms are considered.

The SVD analysis described in § 3 is now performed on the waveforms generated from all gates in the library. By building the model based on all gates in the library, we insure that the representation of the waveform that we produce will

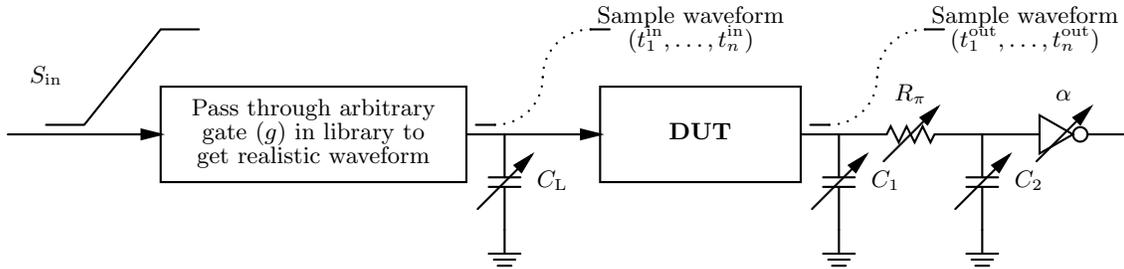


Figure 5: Gate characterization setup

be valid for all gates in the library, and thus can be used in an equation similar to Eq. (16).

The most important thing to note is about the *right singular vectors* \mathbf{V} . The right singular vectors \mathbf{V} obtained after SVD analysis of all the waveforms obtained from the library is the *same* for every gate in the library. Consider for the sake of argument that \mathbf{V}_2 in Eq. (11) was obtained after doing SVD analysis on all the waveforms generated from the library. Then irrespective of whether the waveform is generated by an INV (inverter) or NAND, we use \mathbf{V}_2 to generate the second moment for that waveform. This helps ensure a uniform model for all gates in the library.

In our delay models, we use a more complex load model, namely the π -model. Note that this load modeling is completely *orthogonal* to the waveform modeling which is the primary focus of the paper. When the π -model is used to model the load in timing analysis, the accuracy of the simulation improves [14, 15]. Also we model the non-linear capacitance of the driven gate marked as α in Figure 5.

The experimental setup to generate these waveforms for one gate is shown in Figure 5. By varying S_{in} , and C_L we vary the *input* waveform to the gate, and by varying C_1 , R_π , C_2 and α we vary the *loading* on the gate. While it may appear that having a complex π -model will drastically increase the characterization time, we balance this addition by the use of experiment planning techniques such as Latin Hypercube Sampling (LHS) [16] in order to reduce the number of simulations required.

7. WAVEFORM PROPAGATION ACROSS GATES IN STA

In this section, we present the equations for propagating waveform through a gate with π -model load. The SVD-based timing model is a straightforward extension of Eq. (14) and Eq. (15):

$$\tilde{m}_j^{\text{out}} = a_0 + \sum_{i=1}^r a_i \tilde{m}_i^{\text{in}} + \sum_{j=1}^2 b_j C_j + \sum_{i=1}^r \sum_{j=1}^2 c_{ij} \tilde{m}_i^{\text{in}} C_j + \sum_{j=1}^2 d_j R_\pi C_j + e_0 R_\pi + f_0 \alpha, j = 1, \dots, r \quad (17)$$

where

- r is the number of moments used for characterization, usually set to 2 or 3.
- C_1, R_π, C_2 are the parameters of π -model interconnect. The interconnect values are assumed to be deterministic in this paper.
- α is the width of the gate load being driven as shown in Figure 5.
- \tilde{m}_i^{in} are the moments of the waveform whose arrival time

has been normalized to 0 as described in § 5.

The equations for characterizing arrival time t_{arrival} have the same form as in Eq. (17).

In the context of STA, the independent variables in Eq. (17) namely, $m_1^{\text{in}}, \dots, m_r^{\text{in}}, C_1, R_\pi, C_2, \alpha$ are just numbers. Thus the Eq. (17), could have been any *arbitrary* function of the independent variables. But we have opted for a function which is linear in input moments (\tilde{m}_i^{in}). The linearity restriction becomes crucial in the context of Statistical STA (SSTA) [17]. We note that the extension to SSTA is a part of our future work.

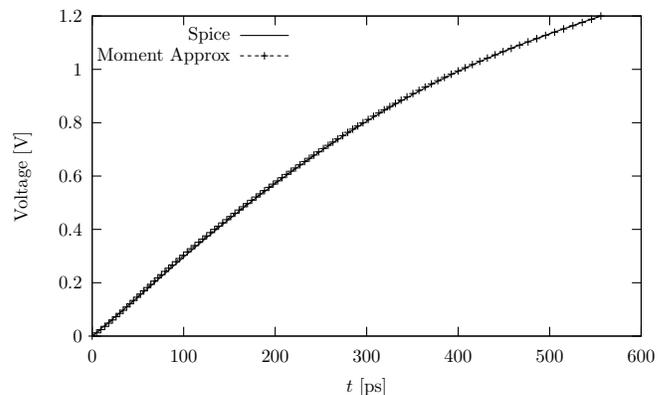


Figure 6: Waveform comparison at the output of an inverter. We use a 2 moment approximation in addition to arrival time and it is clear from the figure that the waveforms are indistinguishable.

The linearity restriction does not affect the accuracy of our modeling. To illustrate, consider Figure 6, where we plot the waveform obtained at the output of an inverter. An input waveform is fed to an inverter and simulated in Spice and the output is plotted with the legend ‘Spice’. In the case of the legend ‘Moment Approx’ the output waveform was obtained as follows. The input waveform is transformed from timepoint representation to moment representation using $\tilde{\mathbf{m}} = \mathbf{t}\mathbf{V}$. We keep the first two moments alongside the arrival time and then using the fitting equations in Eq. (17), to find the moments of the output waveform. Then using the transformation from moments to timepoints $\mathbf{t} = \tilde{\mathbf{m}}\mathbf{V}^\top$ we obtain the timepoints of the output waveform which is plotted with the legend ‘Moment Approx’. It is clear from the figure that the waveforms are indistinguishable thus illustrating the accuracy of the moments-based model.

Another way of comparing the output waveforms was to find the maximum relative difference between the time points predicted by the SVD-based model and Spice. In the case of INV, fitting over 1000 waveforms produced a maximum

error around 5%. The other gates in the library NOR and NAND had similar error statistics in the predicted output waveform.

Now we can generalize and propagate moments across a path and recover the waveform at the end of the path.

8. PATH DELAY EVALUATION IN STA

In this section we demonstrate waveform propagation along a path by considering a simple stage consisting of an INV, NOR and NAND gate. with π -model for load as shown in Figure 7. The gate delay models are generated using 90 nm

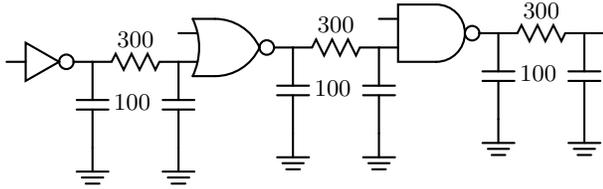


Figure 7: Test case for evaluating path delay. The units for resistance is Ohms (Ω) and the capacitors is femtoFarads (fF). Both the capacitors in the π -model have the same value.

Berkeley Predictive Technology Model [18] and the interconnect parameters are obtained from ITRS roadmap [19]. We assume single-input switching while propagating waveform.

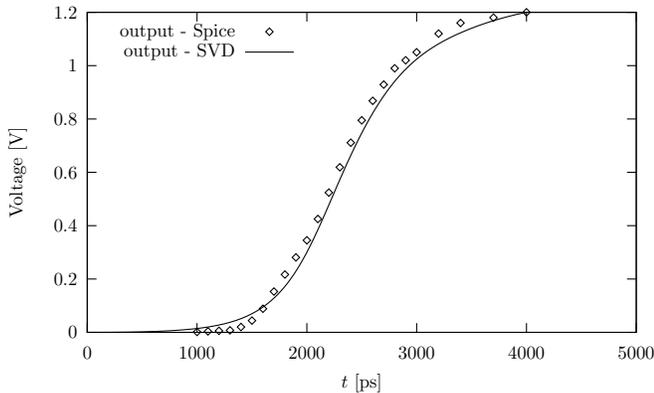


Figure 8: Waveform at the output of the NAND gate in Figure 7. The proposed SVD method and Spice output have a very close match.

We used the first $r = 2$ moments along with arrival time to propagate waveform. We compare our results with Spice and the waveforms at the output of NAND gate in Figure 7 is shown in Figure 8. The waveform predicted by the proposed method closely matches with the Spice result and the error in delay was less than 1%. The maximum error at any given time point was around 8%. and it is around the point at which the output waveform begins to rise ($t \approx 1600$ ps) in Figure 8.

9. CONCLUSION AND FUTURE WORK

This paper presented a rigorous mathematical analysis of waveforms which led to a logical extension of present waveform modeling methods such as saturated-ramp model. The waveform models described in the paper would find application in any situation where having a more detailed description of digital switching waveforms is useful. We demonstrated the application of waveform modeling to path-based

STA demonstrating near-Spice like accuracy. We plan to extend our waveform model to include interconnects and process variations. In addition, we plan to implement a full-fledged path-based timer using the proposed waveform models.

Acknowledgment

This work is partially supported by SRC, IBM Faculty Award, Fujitsu, Sun, and Intel equipment donation.

10. REFERENCES

- [1] Larry McMurchie and Carl Sechen. WTA: waveform-based timing analysis for deep submicron circuits. In *ICCAD '02: Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 625–631, 2002.
- [2] Masanori Hashimoto, Yuji Yamada, and Hidetoshi Onodera. Equivalent waveform propagation for static timing analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(4):498–508, 2004.
- [3] Sani R. Nassif and Emrah Acar. Advanced waveform models for the nano-meter regime. In *ACM/IEEE International Workshop on Timing Issues*, 2004.
- [4] Chirayu S. Amin, Florentin Dartu, and Yehea I. Ismail. Weibull-based analytical waveform model. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(8):1156–1168, 2005.
- [5] Alireza Kasnavi, Joddy W. Wang, Mahmoud Shahram, and Jindrich Zejda. Analytical modeling of crosstalk noise waveforms using weibull function. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 141–146, 2004.
- [6] Amit Jain, David Blaauw, and Vladimir Zolotov. Accurate delay computation for noisy waveform shapes. In *ICCAD '05: Proceedings of the 2005 IEEE/ACM international conference on Computer-aided design*, pages 946–952, 2005.
- [7] John F. Croix and D. F. Wong. Blade and razor: cell and interconnect delay analysis using current-based models. In *DAC '03: Proceedings of the 40th conference on Design automation*, pages 386–389, 2003.
- [8] Igor Keller, Ken Tseng, and Nisath Verghese. A robust cell-level crosstalk delay change analysis. In *ICCAD '04: Proceedings of the 2004 IEEE/ACM International conference on Computer-aided design*, pages 147–154, 2004.
- [9] Chirayu Amin, Chandramouli Kashyap, Noel Menezes, Kip Killpack, and Eli Chiprout. A multi-port current source model for multiple-input switching effects in cmos library cells. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 247–252, 2006.
- [10] Hanif Fatemi, Shahin Nazarian, and Massoud Pedram. Statistical logic cell delay analysis using a current-based model. In *DAC '06: Proceedings of the 43rd annual conference on Design automation*, pages 253–256, 2006.
- [11] Gilbert Strang. The fundamental theorem of linear algebra. *American Mathematical Monthly*, 100(9):848–855, November 1993.
- [12] Lloyd N. Trefethen and David Bau, III. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.
- [13] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2006.
- [14] Peter R. O'Brien and Thomas L. Savarino. Modeling the driving point characteristic of resistive interconnect for accurate delay estimation. In *ICCAD '89: Proceedings of the 1989 IEEE/ACM international conference on Computer-aided design*, pages 512–515, 1989.
- [15] Dmitry Messerman, Alex Gershtein, Sergey Goldenberg, and Vladi Tsipenyuk. Advanced modeling techniques for accurate transistor-level timing analysis. In *ACM/IEEE International Workshop on Timing Issues*, 2007.
- [16] M. D. McKay, R. J. Beckman, and W. J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2):239–245, May 1979.
- [17] Anand Ramalingam, Ashish Kumar Singh, Sani R. Nassif, Gi-Joon Nam, Michael Orshansky, and David Z. Pan. An accurate sparse matrix based framework for statistical static timing analysis. In *ICCAD '06: Proceedings of the 2006 IEEE/ACM International conference on Computer-aided design*, pages 231–236, 2006.
- [18] Yu Cao, Takashi Sato, Michael Orshansky, Dennis Sylvester, and Chenming Hu. New paradigm of predictive MOSFET and interconnect modeling for early circuit simulation. In *Proceedings of Custom Integrated Circuits Conference*, pages 201–204, 2000.
- [19] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors*, 2005.