Transistor-Level Static Timing Analysis by Piecewise Quadratic Waveform Matching and SC Method

Zhong Wang, Jianwen Zhu
Electrical and Computer Engineering
University of Toronto, Ontario M5S 3G4, Canada
{zwang, jzhu}@eecg.toronto.edu

Abstract

While fast timing analysis methods based on model order reduction have been well established for linear circuits, the timing analysis for non-linear circuits, which are dominant in digital circuits, is usually performed by a SPICE-like, numerical integration-based approach solving differential equations. In this paper, we propose a new technique that leads to the transient solution of charge / discharge paths with a complexity equivalent to only K DC operating point calculations, where K is the number of transistors along the path. This is accomplished by approximating each nodal voltage as a piecewise quadratic waveform, whose characteristics can be determined by matching the charge / discharge currents calculated by the capacitive components and the resistive components. Successive chord method is then applied to reduce the matrix construction and inversion overhead. Experiments on a wide range of circuits show that an average of 20 times speed-up over HSPICE simulation (transient time only) with 10 picosecond step size can be achieved, while maintaining an average accuracy of 98.03%.

1 Introduction

Timing analysis is the process of verifying the timing properties, such as propagation delay, setup/hold time violations etc., of a digital VLSI circuit. Since timing properties are inherently associated with the transient response of a circuit, circuit simulators, such as SPICE, have been the fundamental tools to obtain such characteristics. Circuit simulation involves the solution of differential equations whose size is proportional to the size of the circuit. In addition, the equations have to be solved as many times as the number of input combinations. Therefore, many techniques have been devised to reduce the exponential circuit simulation time.

Circuit partitioning is used so that differential equation solving is confined within small circuit partitions. Traditionally, *Gate abstraction* is used so that each partition corresponds to a gate, whose timing property can be pre-characterized. With *Static timing analysis*, only the best and the worst case scenarios of each gate need to be simulated and only the timing of the gates along the longest paths needs to be considered.

Example 1 *In Figure 1, the NAND gate, pass transistor M1 and wire segment W1 form a logic stage.*

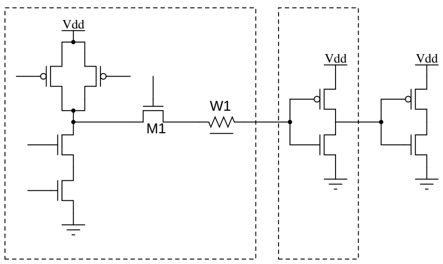

Figure 1. Logic stage.

However, gate abstraction is not always practical in high performance designs. Instead, timing analyzer has to partition a circuit into *logic stages*, each of which is a set of channel connected transistors and wire segments. First of all, not every cell created by designers maps naturally to a gate, in other words, the output of a cell is not always connected to the gate input of another cell. Therefore, the design cell cannot be pre-characterized using the gate abstraction. Instead, a logic stage has to be constructed dynamically, depending on how it is connected to the rest of the circuit. Second, transistors are coupled with interconnect, whose electrical properties cannot be ignored in deep submicron design. What makes interconnects particularly challenging is that their geometric shape cannot be pre-determined until routing is completed. This makes it extremely hard even for the pre-characterization of gates, since the output load can no longer be modeled as a lumped capacitor. Furthermore, many common layout structures in high-performance designs contain channel-connected transistors through long wires.

Example 2 *Consider a Manchester carry chain in Figure 2. Note that the output of each bit-sliced cell, e.g., C1, is channel-connected to other cells. Therefore, the cell does not correspond to a logic gate.*

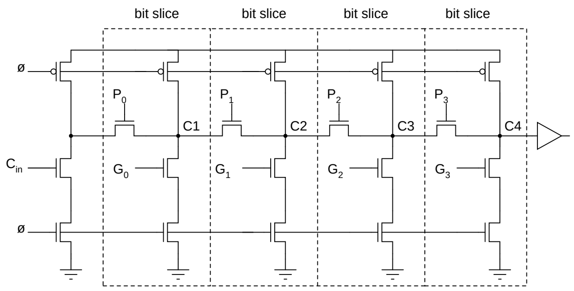

Figure 2. Manchester carry chain.

Example 3 *Consider the memory decoder tree in Figure 3, which is drawn deliberately to mimic the layout structure. Note here the lengths of the bold wire segments (wire0, wire1, wire2), which connect transistors' diffusions, grow exponentially with the tree level.*

Therefore, fast, on-the-fly worst case analysis of a logic stage, which boils down to the transient simulation of transistor chains, becomes an absolute necessity. On the other hand, while circuit partitioning offers order-of-magnitude speed-up over SPICE for full-chip timing analysis by exploiting the spatial and temporal latency of the circuit, it lends no help in speeding up the timing analysis of an individual logic stage. The simulation speed and accuracy for each logic stage will

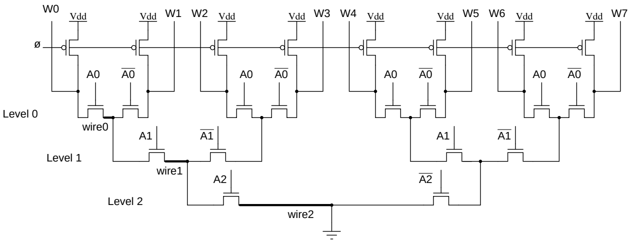

Figure 3. Memory decoder.

be a critical problem for any timing analyzer facing designs with the complexity nowadays.

Two methodologies have been pursued in the past for the fast simulation of logic stages. The first methodology exploits a simplified transistor device model, for example, a linear or piecewise linear model. This approach enables the modeling of non-linear circuits as linear systems. Efficient frequency-domain analysis techniques, such as asymptotic waveform evaluation (AWE), can then be used. While extremely fast, this approach introduces significant error during the device linearization process. The second methodology continues to use the time-domain numerical integration based approach, however, Newton-Raphson (NR) iteration, as the engine of the solver, is replaced by Successive Chord (SC) iteration, which is much faster due to the constant nature of the resultant admittance matrix.

In this paper, we introduce a third methodology, called quadratic waveform matching (QWM), which has not been attempted before. We achieve fast simulation speed by avoiding the brute-force solution of differential equations, while maintaining the accuracy of device models. In fact, the circuit only needs to be solved as a system of algebraic equations at K critical points, where K is the number of transistors. This approach is inherently much faster than SPICE-like simulators in that nonlinear iterations only need to be performed at large time steps. To achieve this, the transient process is divided into regions separated by the K critical points. Nodal voltages in each region are then approximated by quadratic waveforms, each of which is characterized by one parameter. These parameters are determined subsequently by matching the charge/discharge currents at the critical points with those predicted by the device I/V relationship. Our approach is accelerated further by the use of SC iteration in place of NR iteration. While the advantage of SC has been demonstrated for solving nonlinear equations in previous work, we show how the same idea can be applied in the context of quadratic waveform matching by extra manipulations.

The rest of the paper is organized as follows. A brief review of the previous work is given in Section 2. We will then the state problem formulation in next section. The proposed QWM method is described in detail in Section 4 and the SC method is introduced in Section 5. Finally, we present our experimental results.

2 Related Work

Efficient extraction of timing metrics for linear circuits, typically modeled as RC or RLC networks, has been well established for years. Elmore delay [1] is used extensively for its closed form and ease of evaluation on RC trees. Since Elmore delay is inherently linked to the first moment of the system transfer function, a straightforward improvement uses higher order moments and retains a larger number of dominant poles to obtain a better approximation of the transfer function. Pileggi and Rohrer pioneered in this area with their asymptotic waveform evaluation (AWE) method [2]. Alpert *et al.* [3] showed that empirical delay metrics can be directly obtained from moments without further computation of dominant poles. Derivatives of AWE [4] [5] solve the numerical problems such as stability and passivity associated with AWE.

No transfer function can be defined for the nonlinear circuits. Nevertheless, one can simplify the device model in such a way so that analysis methods developed for linear circuits can be employed. The switch-level simulators, such as Crystal [6] and IRSIM [7], model the transistors as switched resistors. A logic stage can then be reduced into an RC network, for which Elmore delay is computed. MOM [8] and ACES [9] improve this approach by the piecewise linearization of transistor model, while using AWE to further improve the evaluation accuracy of each linear region.

The piecewise simplification of the device model introduces events, when a device is at the boundary of its piecewise region. The circuit activity, or the number of events generated, will increase rapidly when more accurate device model is used. This leads to the rapid degradation of simulation speed for highly nonlinear deep submicron devices. To avoid that, TETA [10] keeps an accurate, nonlinear device model and remains to use the time-domain integration based approach to solve differential equations. However, it uses tabular device models to avoid the dominant model building time in SPICE. In addition, it replaces Newton-Raphson iteration with successive chord iteration [11]. While with a theoretically inferior convergence rate, SC can evaluate each iteration much faster because the admittance matrix of the linearized circuit stays constant. The authors also showed the efficiency of TETA approach for multi-port logic stages coupled by interconnects. While the use of SC iteration in QWM is inspired by TETA, our approach is fundamentally different in that numerical integration is not needed.

3 Waveform Evaluation

Since the partitioning of circuits into logic stages as well as path-based timing analysis of logic stage networks have been well established, we focus only on the static timing analysis of individual logic stage. We formulate it as a *waveform evaluation* problem.

3.1 Circuit Model

A CMOS logic stage is modeled as a polar directed graph, ,as shown in Definition 1whose vertices represent the set of circuit *nodes* and edges represent the set of *circuit elements*. The source of the graph represents the power supply and the sink of the graph represents the ground. There are three types of circuit elements: NMOS transistor, PMOS transistor and wire segment. Each circuit element is characterized by its geometric parameters, including its width, length, and additionally for the transistor, the area and perimeter of its junctions. The electrical properties of circuit elements can be derived from such geometric information. A logic stage contains a set of *inputs*, each of which is associated with the gate of a transistor, and a set of *outputs*, which are circuit nodes that are intended to be connected to the inputs of other stages.

Definition 1 *A CMOS* **logic stage** *is a polar directed graph* $\langle N, E, s, t, I, O \rangle$*, with the set of nodes* $N \subseteq Node$*, the set of edges* $E \subseteq Edge$*, the source node* $s \in N$*, the sink node* $t \in N$*, the inputs* $I \subseteq E$ *and the outputs* $O \subseteq N$.

Node = **tuple** {
  incoming : $\langle\rangle^{Edge}$;
  outgoing : $\langle\rangle^{Edge}$;
  }
Edge = **tuple** {
  kind : *Device;*
  src, snk : *Node;*
  w, l : $\mathcal{R}$;
  }
Device = $\{nmos, pmos, wire\}$;

Example 4 *The logic stage in Figure 4 (a) can be reduced into the graph model in Figure 4 (b), which can be defined as* $V = \{Vdd, N1, N2, N3, N4, GND\}$*,* $E = \{M1, M2, M3, M4, M5, W1\}$*,* $I = \{M1, M2, M3, M5, M5\}$*,* $O = \{N4\}$. $s = Vdd$*,* $t = GND$.

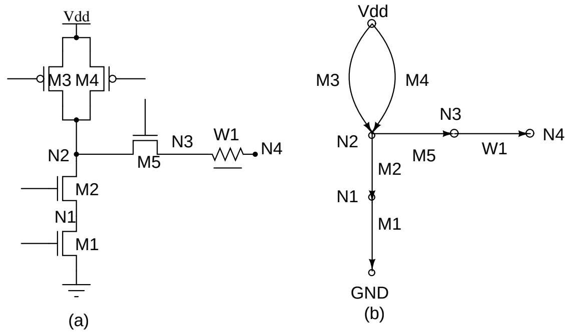

Figure 4. Circuit model.

3.2 Device Model

Each circuit element type is associated with a device model, as shown in Definition 2,. The model defines the device I-V relationship *iv()* as a mapping from its geometric parameters and terminal voltage configuration (*TermVoltage*)to the corresponding current flowing from the source node to the sink node. The device model also defines how threshold voltage *threshold()* is related to the terminal voltages in order to factor in the body effect. The model also includes the definition of the parasitic capacitance contributions to the source and sink, which are functions of voltages as well. As illustrated later in Section 7, we can use a tabular approach to accurately pre-characterize the deep submicron devices.

Definition 2 *A* **device model** *m is a member of DeviceModel.*

```
DeviceModel = tuple {
   iv          : ℛ × ℛ × TermVoltage ↦ ℛ;
   threshold   : TermVoltage ↦ ℛ;
   srcCap      : ℛ × ℛ ↦ ℛ;
   snkCap      : ℛ × ℛ ↦ ℛ;
   inputCap    : ℛ × ℛ ↦ ℛ;
   }
TermVoltage = tuple {
   input       : ℛ;
   src         : ℛ;
   snk         : ℛ;
   }
```

Example 5 *Figure 5 plots the projection of the NMOS device model: it demonstrates how current changes (I_{ds}) with respect to the change of source voltage (V_d) and sink voltage (V_s).*

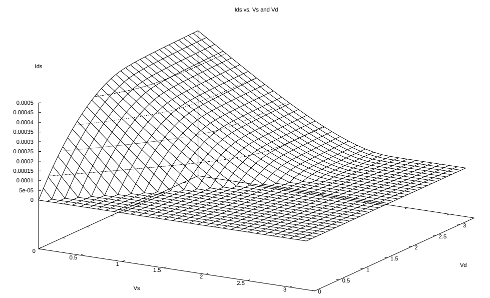

Figure 5. I-V relationship in device model.

3.3 Waveform Evaluation Problem

The waveform evaluation process computes the output waveforms given the input waveforms and load capacitances, as shown in Definition 3. Waveform evaluation computes richer information than traditional timing analysis where only the delay/slope pair is computed. The importance of waveform evaluation is confirmed by a recent paper [12] that in deep submicron circuits, the traditional delay metric can lead to up to 30% error.

Definition 3 *A* **waveform evaluation** *of a logic stage $\langle N, E, s, t, I, O \rangle$ under the set of device models model : Device $\mapsto$ DeviceModel computes a set of output voltage waveforms $V : O \times \mathcal{T} \mapsto \mathcal{R}$, given a set of input voltage waveforms $G : I \times \mathcal{T} \mapsto \mathcal{R}$ and a set of load capacitances $C_L : N \mapsto \mathcal{R}$.*

Since we are performing the static timing analysis, only the worst case, in other words, charging/discharging along the longest paths, needs to be considered. For a logic stage, the worst case scenario usually happens when the only switching input is at the the gate of the bottom transistor along the stage. Without loss of generality, we consider the discharge case of a stack of K NMOS transistors. With the same methodology, PMOS transistors can be easily incorporated. Each transistor M^k connects circuit node $k+1$ and k, and has a size of w^k and l^k, as shown in Figure 6. In addition, the input waveform is assumed to be G^k. The capacitance of each node to ground is C^k, which equals to the sum of all capacitances contributed by the incidental circuit elements and the load capacitance. To further simplify the presentation, we assume all parasitic capacitances are constant. Our implementation, as Equation (10) demonstrates, does not make this assumption.

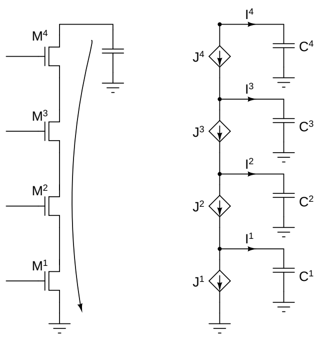

Figure 6. Discharge along the longest path.

4 Quadratic Waveform Matching

Assume that the discharge current flowing through the capacitance I^k associated with node k can be approximated by a waveform with a simple *analytical form*, for example, a polynomial with respect to time. While the analytical form is predefined, its characterization parameters are to be determined. In contrast to the numerical integration used by the SPICE, we can symbolically integrate I^k to obtain the voltage waveform. Assume the initial condition at time τ is known. We can compute the voltage at τ'

$$V^k_{\tau'} - V^k_{\tau'} = \int_{\tau}^{\tau'} \frac{1}{C^k} I^k(t)dt, \forall k \quad (1)$$

We then look at a particular time point τ'. With the analytical voltage waveform, every nodal voltage at τ' can be easily evaluated. By examining the I-V relationship defined in the device model, the current flowing through each circuit element J^k can be determined.

$$J^k_{\tau'} = m.iv(w^k, l^k, G^k_{\tau'}, V^k_{\tau'} - V^{k-1}_{\tau'}) \quad (2)$$

The discharging current at time τ', $I_{\tau'}$, should be matched with the difference between currents flowing through its neighboring devices:

$$\begin{aligned} I^k_{\tau'} &= J^{k+1}_{\tau'} - J^k_{\tau'}, \forall k < K & (3) \\ I^K_{\tau'} &= J^K_{\tau'} & (4) \end{aligned}$$

Now we obtain an algebraic equation for each circuit node. These equations can help solve the parameters to be determined. If there are r parameters chosen to characterize each output waveform, then $r \cdot K$ equations need to be generated, in other words, r time points need to be chosen to perform waveform matching. Given that, the transient solution of the circuit is then reduced to the solution of a system of algebraic equations!

The art part of the waveform matching methodology is the choice of the analytical waveform model. The discharging currents of all circuit nodes of a stack of 6 NMOS transistors are shown in Figure 7. An interesting and important observation is that each current waveform has a single peak, which coincides with the time when the transistor above turns on, in other words, when its gate drive is equal to its threshold voltage. An intuitive explanation of this phenomenon is that for any transistor M^k, when its upper transistor M^{k+1} turns on and its channel current J^{k+1} increases, the absolute value of the discharge current I^k, which is the difference between channel currents J^k and J^{k+1}, should start to decrease.

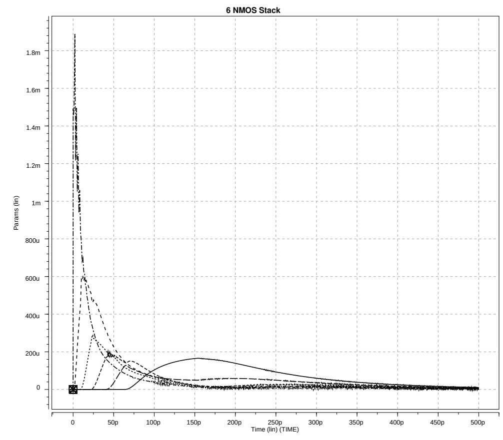

Figure 7. Discharge current of of 6 NMOS transistor stack.

Based on this observation, we choose to approximate the current waveform between two critical points $[\tau, \tau']$ by a linear model:

$$I^k_t = I^k_\tau + \alpha^k(t - \tau) \quad (5)$$

Substituting Equation (5) into Equation (1) and performing integration, we can obtain the *quadratic approximation* of the voltage waveform, which is characterized by a single parameter α^k:

$$V^k_t = V^k_\tau + [I^k_\tau(t-\tau) + 0.5\alpha^k(t-\tau)^2]/C^k, \quad t \in [\tau, \tau'] \quad (6)$$

A *piecewise quadratic waveform matching* strategy is used here: divide the transient process into K regions according to the critical points. The voltage and current waveform of each region is approximated by Equation (6) and Equation (5). We then solve for the parameters α^k of each region by matching currents at the corresponding critical point. More specifically, given the initial voltage value V^k_τ and current value I^k_τ, the parameters α^k are solved by the algebraic equations at the next critical point τ', i.e., when the transistor M^L turns on. All equations are collected in Equation (7). With the existence of nonlinear function $iv()$, Equation (7) ends up to be a nonlinear equation set. It is obvious that the approach proposed can also handle non-equilibrium condition, which is usually not considered in static timing analysis though.

$$\begin{cases} I_{\tau'}^{k} & = & I_{\tau}^{k}+\alpha^{k}(\tau'-\tau), \forall k \\ V_{\tau'}^{k} & = & V_{\tau}^{k}+[I_{\tau}^{k}(\tau'-\tau)+0.5\alpha^{k}(\tau'-\tau)^{2}]/C^{k}, \forall k \\ J_{\tau'}^{k} & = & m.iv(w^{k}, l^{k}, G_{\tau'}^{k}, V_{\tau'}^{k}, V_{\tau'}^{k-1}), \forall k \\ I_{\tau'}^{k} & = & J_{\tau'}^{k+1}-J_{\tau'}^{k}, \forall k<L \\ I_{\tau'}^{L} & = & -J_{\tau'}^{L} \\ G_{\tau'}^{L+1} & = & V_{\tau'}^{L}+m.threshold(V_{\tau'}^{L}) \end{cases} \tag{7}$$

5 Successive Chord Method

Let $\mathbf{F}(x)=0$ be a set of nonlinear algebraic equations over the variable set $\mathbf{x}$ and $\mathbf{A}=\partial\mathbf{F}/\partial\mathbf{x}$ be the Jacobian matrix of $\mathbf{F}(\mathbf{x})$. One can apply the iterative approach to solve the equations numerically by computing a better approximation of the solution $\mathbf{x}_{k+1}$ based on the current approximation $\mathbf{x}_k$. The convergence criterion can be simply whether the norm of $\mathbf{F}(\mathbf{x_k})$ is less than certain threshold value. Many iterative schemes have been devised, among which the most effective one seems to be the Newton-Raphson (NR) method. NR method calculates the next approximation $\mathbf{x}_{k+1}$ by computing the derivative at $\mathbf{x}_{k+1}$.

$$\mathbf{x}_{k+1}=\mathbf{x}_k-\mathbf{A}(x_k)^{-1}\cdot\mathbf{F}(x_k)$$

Since the Jacobian matrix is dependent on $\mathbf{x}$, each iteration of the NR method usually involves the following computations: (1) error evaluation and convergence checking; (2) Jacobian matrix construction; (3) Jacobian matrix inversion; (4) update calculation. Figure 8 shows the breakdown of the runtime cost of each task when NR method is used to evaluate a 6 transistor stack. It is obvious that Jacobian matrix construction and inversion dominate the runtime. This is due to the cubic complexity of matrix inversion algorithm, be it done explicitly or implicitly (by performing LU decomposition).

A very useful observation of the iteration procedure is that while the Jacobian matrix $\mathbf{A}$ gives both the direction and magnitude of the update vector, only the direction is needed to ensure convergence. In other words, one can use a substitute $\hat{\mathbf{A}}$ for $\mathbf{A}$ as long as $\hat{\mathbf{A}}$ always the gives the correct update direction. The essence of the SC method is to use a matrix $\hat{\mathbf{A}}$ that is

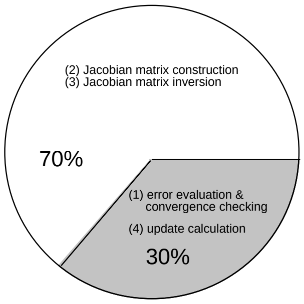

Figure 8. NR method profiling.

independent of $\mathbf{x}$. Since $\hat{\mathbf{A}}$ is constant, the computational overhead of (2) and (3) can be eliminated from the deepest iteration loop. Therefore, the constant Jacobian matrix is LU decomposed only once and the decomposition result can be reused for all later iterations. The tradeoff here is that SC method has a theoretical linear convergence rate, which is slower than the quadratic convergence rate of NR method.

What remains is the art of selecting the appropriate $\hat{\mathbf{A}}$. Rearranging Equation (7) in the form of $\mathbf{F}(x)$, we obtain Equation (8), where the variables are the set of nodal voltages $\mathbf{V}_{\tau'}=\{V_{\tau'}^{k}|\forall k<L\}$ as well as the timestep $T=\tau'-\tau$.

$$\begin{cases} \dfrac{I_{\tau}^{k}+J_{\tau'}^{k+1}(\mathbf{V}_{\tau'})-J_{\tau'}^{k}(\mathbf{V}_{\tau'})}{2}\cdot\dfrac{T}{C^{k}}+V_{\tau}^{k}-V_{\tau'}^{k} & = & 0, \forall k<L \\ G_{\tau'}^{L+1}-V_{\tau'}^{L}- & & \\ \quad [V_{T0}+\gamma(\sqrt{-2\phi_F+V_{\tau'}^{L}}-\sqrt{|-2\phi_F|})] & = & 0 \\ \dfrac{2\cdot C^{L}\cdot(V_{\tau'}^{L}-V_{\tau}^{L})}{-J_{\tau'}^{L}(\mathbf{V}_{\tau'})+I_{\tau}^{L}}-T & = & 0 \end{cases} \tag{8}$$

Most of the elements in the Jacobian matrix are the combinations of partial derivatives of channel current J versus the drain voltage V_d or the source voltage V_s, which can be obtained from device I-V relationship. Let ρ be the derivative $\partial J/\partial V_d$ and let ϕ be the derivative $\partial J/\partial V_s$. Then Equation (9) defines the non-zero elements of matrix $\hat{\mathbf{A}}$.

, as shown in Figure 9.

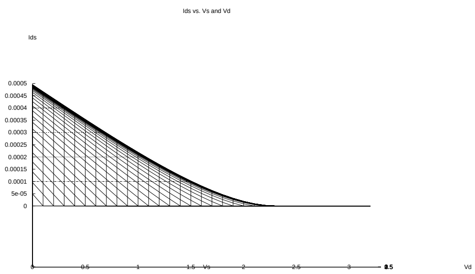

Figure 9. I_{ds} **vs** V_s**.**

$$
\begin{cases}
\hat{\mathbf{A}}_{k,k-1} &= -\dfrac{T}{2C^k}\cdot\phi(M^k), \qquad \forall k<L\\
\hat{\mathbf{A}}_{k,k} &= \dfrac{T}{2C^k}[\phi(M^{k+1})-\rho(M^k)]-1, \quad \forall k<L\\
\hat{\mathbf{A}}_{k,k+1} &= \dfrac{T}{2C^k}\cdot\rho(M^{k+1}), \qquad \forall k<L\\
\hat{\mathbf{A}}_{k,L+1} &= \dfrac{I_\tau^k+J_{\tau'}^{k+1}(\mathbf{V}_{\tau'})-J_{\tau'}^k(\mathbf{V}_{\tau'})}{2\cdot C^k}, \qquad \forall k<L\\
\hat{\mathbf{A}}_{L,L} &= -1-\dfrac{\gamma}{\sqrt{-2\phi_F+V_{\tau'}^L}}\\
\hat{\mathbf{A}}_{L,L+1} &= dG_{\tau'}^{L+1}/dt\\
\hat{\mathbf{A}}_{L+1,L} &= \dfrac{2\cdot C^L\cdot(V_{\tau'}^L-V_\tau^L)}{(-J_{\tau'}^L+I_\tau^L)^2}\cdot\phi(M_L)\\
\hat{\mathbf{A}}_{L+1,L+1} &= -1
\end{cases}
\tag{9}
$$

Jacobian $\hat{\mathbf{A}}$is close to a tridiagonal matrix except the last column.

$$
\hat{\mathbf{A}}=\begin{bmatrix}
\hat{\mathbf{A}}_{1,1} & \hat{\mathbf{A}}_{1,2} & 0 & \dots & 0 & \dots & \hat{\mathbf{A}}_{1,L}\\
\hat{\mathbf{A}}_{2,1} & \hat{\mathbf{A}}_{2,2} & \hat{\mathbf{A}}_{2,3} & \dots & 0 & \dots & \hat{\mathbf{A}}_{2,L}\\
0 & \hat{\mathbf{A}}_{3,2} & \hat{\mathbf{A}}_{3,3} & \hat{\mathbf{A}}_{3,4} & 0 & \dots & \hat{\mathbf{A}}_{3,L}\\
 & & & \vdots & & & \\
0 & & \vdots & & 0 & \hat{\mathbf{A}}_{L-1,L-1} & \hat{\mathbf{A}}_{L,L}
\end{bmatrix}
$$

6 Device Modeling

The device model contains two parts: I-V relationship and C-V relationship. For I-V relationship, a direct, tabular implementation can ensure no loss of accuracy as long as the grid size is fine enough. However, such approach can lead to unacceptable amount of memory usage. Therefore, we use a combination of curve-fitting and interpolation technique to compress the device model data. To characterize transistor I-V relation, we sweep V_s and V_g from 0 volt to 3.3 volt with a step size of 0.1 volt. For each V_s/V_g pair, we then generate polynomial functions to capture the the dependence of channel current on drain voltage V_d using curve fitting technique. We use a linear function for the saturation region and a quadratic function for the triode region, as shown in Figure 10. Thus, together with the threshold voltage and saturation voltage, we store 7 parameters for each pair. If an I-V query is performed with terminal voltages not captured by the grid of the table, the current value will be interpolated from neighboring points. One benefit of this characterization and fitting method is that $\partial I_{ds}/\partial V_d$ and $\partial I_{ds}/\partial V_s$ can be computed very fast.

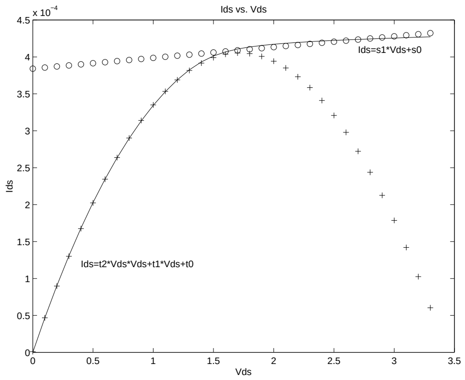

Figure 10. I-V curve fitting.

A large error on capacitance model will result in a large error in the final delay value. Therefore, it is important to have an accurate capacitance model in QWM. For example, the junction capacitance has a variety of sources, all of which which depend on the junction voltage and the working mode of the transistor. One of the sources, the reversed biased diode junction capacitance, is $C_j=C_{j0}/(1-V_D/\phi_0)^m$. We integrate it on voltage range $[V_{\tau'},V_\tau]$ and get the effective value on that range as in Equation (10).

$$
C_{eff}=C_{j0}\cdot\frac{-\phi_0^m\cdot[(\phi_0-V_\tau)^{1-m}-(\phi_0-V_{\tau'})^{1-m}]}{(V_\tau-V_{\tau'})(1-m)}
\tag{10}
$$

With the analytical form of the nodal voltage, for each transistor, dV_g/dV_d is readily available and the

Miller effect can be easily calculated.

7 Experiments

To verify the QWM method, we first characterize the device models using the CMOSP35 technology with $\lambda = 0.25\mu$. The sample data used for characterization are obtained by HSPICE simulation using BSIM3 V3.1 model. We then analyze a set of standard CMOS logic gates. To further measure how QWM method scales with the transistor stack size, we also analyze transistor stacks of lengths ranging from 5 to 10, with randomly chosen transistor widths. The result is then compared against the HSPICE simulation time. Since the simulation time of HSPICE for small circuits is dominated by the model building time, which is minimal in QWM approach due to its tabular device model, we compare only with the **transient time** reported by HSPICE to ensure fairness. Since the user-specified step size has an impact on the HSPICE simulation time, we perform HSPICE simulation with step size of 1ps and 10ps.

All experiments are carried on a SUN Blade 100 system running at 500 MHZ.

Table 1. QWM vs HSPICE for logic gates.

Circuit	HSPICE(1ps)		HSPICE(10ps)		QWM	
	Run Time	Speed-up	Run Time	Speed-up	Run Time	Error
inv	0.06	600	0.02	200	0.0001	0.77%
nand2	0.14	156	0.03	33.3	0.0009	1.32%
nand3	0.25	179	0.05	35.7	0.0014	1.16%
nand4	0.41	216	0.06	31.6	0.0019	1.33%

We observe an impressive speed-up of QWM over HSPICE. Table 7 shows our simulation result (in seconds) on minimum sized logic gates. An average speed-up over 180 for 1ps step size and 33 for 10ps step size with an average error around 1.3% is observed. The 600 speed-up for an inverter case comes from a close enough initial guess, which dramatically cuts down the number of iterations in QWM. In Table 2, for each stack length, we show results for three circuit configurations, each of which has different transistor sizes. For timestep of 1 ps, the average speed up is over 150; for timestep of 10 ps, the number is over 20. Note that this speed-up is for transient time only. We observe over 200 times speed-up if total HSPICE runtime is compared. In the mean time, the delay metric obtained contains a worst-case error of 4.00% error and average error of 1.97%.

Table 2. QWM vs HSPICE for randomly generated logic stages.

Size		HSPICE(1ps)		HSPICE(10ps)		QWM	
		Run Time	Speed-up	Run Time	Speed-up	Run Time	Error
5	ckt1	0.3	107	0.05	17.9	0.0028	0.50%
	ckt2	0.43	134	0.07	21.9	0.0032	2.33%
	ckt3	0.81	159	0.12	23.5	0.0051	0.49%
6	ckt1	0.8	242	0.11	33.3	0.0033	0.61%
	ckt2	0.93	194	0.12	25	0.0048	3.33%
	ckt3	0.65	171	0.09	23.7	0.0038	2.01%
7	ckt1	1	175	0.13	22.8	0.0057	0.44%
	ckt2	1.11	188	0.15	25.4	0.0059	0.09%
	ckt3	0.83	136	0.11	18	0.0061	1.26%
8	ckt1	1.11	137	0.14	17.3	0.0081	0.62%
	ckt2	1.52	214	0.19	26.8	0.0071	3.50%
	ckt3	1.49	154	0.19	19.6	0.0097	2.74%
9	ckt1	2.15	102	0.27	12.8	0.0211	4.61%
	ckt2	1.78	151	0.22	18.6	0.0118	3.11%
	ckt3	1.94	110	0.24	13.6	0.0176	2.96%
10	ckt1	1.8	173	0.22	21.2	0.0104	0.63%
	ckt2	2.09	145	0.26	18.1	0.0144	2.15%
	ckt3	2.04	179	0.25	21.9	0.0114	4.00%

The simulation result of a 6 NMOS logic stage is illustrated in Figure 11. The transient result produced by QWM is simply plotted as straight solid lines connecting the critical points calculated by QWM. The result produced by HSPICE is plotted in dashed line. One can observe that QWM result follows quite closely with the HSPICE result. Indeed, the propagation delay calculated for this case is 98.34% accurate, and it is produced 33.3 times faster than HSPICE with 10ps timestep.

Figure 12 demonstrates the simulation result for a logic stage coupled with long wire. We first used AWE approach to build a macro π model for the wire. This can be evidenced by closely spaced waveform pairs in Figure 12, which correspond to the two terminals of wire segments. A speed-up of 26 over HSPICE for 10 ps timestep and accuracy of 96.44% is achieved.

8 Conclusion

In this paper, we propose a new methodology, called quadratic waveform matching, for the fast timing anal-

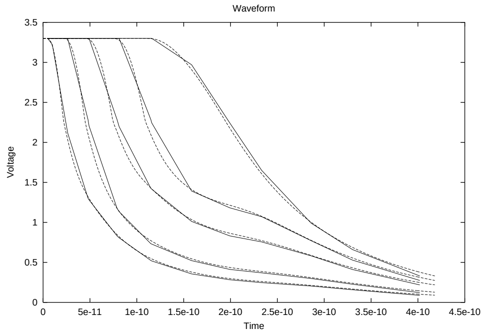

Figure 11. A 6 NMOS stack simulation result.

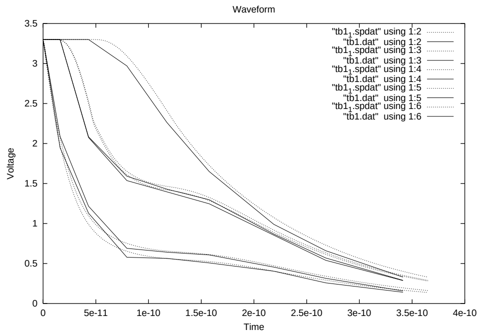

Figure 12. Decoder tree simulation result.

ysis of logic stages. This approach replaces the solution of a system of differential equations by the solution of a few systems of algebraic equations. One instance of this methodology, called piecewise quadratic waveform matching, produces on-average 98.03% accurate delay metric with order-of-magnitude speedup over SPICE.

In the future, we will study the suitability of other waveforms for the timing analysis problem. More sophisticated waveform model and critical point model may help further improve speed and accuracy.

References

[1] W. C. Elmore, "The transient analysis of damped linear networks with particular regard to wideband amplifiers," *Journal of Applied Physics*, vol. 19, 1948.

[2] Lawrence T. Pillage and Ronald A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 9, no. 4, pp. 352–366, April 1990.

[3] Charles J. Alpert, Anirudh Devgan, and Chandramouli Kashyap, "A two moment RC delay metric for performance optimizaion," in *International Symposium on Physical Design*, 2000, pp. 73–78.

[4] P. Feldmann and F. W. Freund, "Efficient linear circuit analysis by padè approximation via the lanczos process," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 14, no. 5, pp. 639–649, May 1995.

[5] A. Odabasioglu, M. Celik, and Lawrence T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 8, pp. 645–654, August 1998.

[6] J. Ousterhout, "Crystal: A timing analyzer for nMOS VLSI circuits," in *Third CalTech Conference on VLSI*, 1983, pp. 57–89.

[7] Arturo Salz and Mark Horowitz, "IRSIM: An incremental MOS switch-level simulator," in *Proceeding of the 26th Design Automation Conference*, 1989, pp. 173–178.

[8] Russel Kao, *Piecewise Linear Models for Switch-Level Simulation*, Ph.D. thesis, Stanford University, 1992.

[9] Anirudh Devgan and Ronald A. Rohrer, "Adaptively controlled explicit simulation," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 13, no. 6, pp. 746–762, June 1994.

[10] F. Dartu and L. Pileggi, "TETA: Transistor-level engine for timing analysis," in *Proceeding of the 35th Design Automation Conference*, 1998, pp. 595–598.

[11] J. M. Ortega and W. R. Rheinbolt, *Iterative Solution of Non-Linear Equations in Several Variables*, New York: Acadmic, 1970.

[12] Larry McMurchie and Carl Sechen, "WTA-waveform-based timing analysis for deep sub-micron circuits," in *Proceedings of the International Conference on Computer-Aided Design*, 2002.