

Blade and Razor: Cell and Interconnect Delay Analysis Using Current-Based Models

John F. Croix
Silicon Metrics Corporation
12710 Research Blvd.
Suite 300
Austin, Texas 78759
John.Croix@siliconmetrics.com

D. F. Wong
University of Illinois at Urbana-Champaign
Department of Electrical and Computer
Engineering
Urbana, Illinois 61801
mdfwong@uiuc.edu

ABSTRACT

In order to adequately account for nanometer effects during timing analysis, archaic standard cell models must be replaced. Simplifying assumptions used during characterization, such as nearly linear voltage inputs or lumped-capacitance loads, are no longer valid. Signal integrity analysis further complicates the characterization process because the typical voltage waveform used during characterization does not contain a noise component. This paper introduces two new technologies for standard cell and interconnect timing analysis: Blade and Razor. Blade is a novel cell model and runtime engine based on current flow. Razor is the accompanying interconnect model. Both Blade and Razor produce and consume arbitrary voltage waveforms with near-SPICE accuracy at speeds tens of thousands of times faster than SPICE.

Categories and Subject Descriptors

B.7.2 [Design Aids]: Simulation

General Terms

Algorithms, Measurement, Performance, Design

Keywords

Razor, Blade, cell model, interconnect model, recursive convolution, timing analysis, current-based model

1. INTRODUCTION

Stage delay (the delay between a point of the voltage waveform used to drive the input of a cell to a point of the voltage waveform at the next cell being driven) has long been dominated by interconnect delay. As a result, research has focused on faster and more accurate algorithms to analyze interconnect while largely ignoring cell delay. The majority of the cell models used in today's IC design flows consist of lookup tables or characteristic equations that rely on linear (ramp) voltage inputs and simplified loads and that create linear (ramp) output voltage waveform approximations.

Interconnect models consume the linear voltage waveform descriptions produced by these cell models and produce voltage waveforms for consumption by downstream cell models. These interconnect models are well-suited for the simplified approximations

created by existing cell models but do not scale well when used in conjunction with more complex input voltage waveform representations. Furthermore, the arbitrary waveforms produced by these interconnect models are simplified (linearized) for consumption by these cell models.

This paper introduces Blade and Razor as solutions to these challenges. Blade is the first high-speed cell model and accompanying runtime engine that can both produce and consume arbitrary voltage waveforms, including noisy waveforms, at near-SPICE accuracy. The Blade model is also the first cell model to operate on arbitrary loads including lumped-C, π -models, trees, and meshes. Razor adds a SPICE-accurate interconnect model to create a stage-delay model. This interconnect model consumes the arbitrarily complex voltage waveform created by the Blade model and accurately calculates the resulting waveform used to drive the next Blade model, all at speeds 5-6 orders of magnitude faster than SPICE.

2. BACKGROUND

Accurate electrical performance of a cell under all operating conditions for any arbitrary input waveform and output loading is only possible with a CPU-intensive circuit simulation program such as SPICE [10]. However, this CPU overhead precludes its use as the primary timing or power analysis application in the IC design flow. Thus, simplified cell models have been created for the rapid estimation of a cell's behavior within other analysis applications.

Cell models have traditionally been created by placing a capacitive load on the output pin of the cell, supplying the input pin with a rising or falling voltage waveform, and measuring the response [3]. Other characterization variables include temperature, voltage level, and process corner. By altering the environment under which these measurements are taken, a cell model can be created.

Many different forms of cell models exist. Two of the most popular forms, *characteristic equation* and *lookup table*, are used extensively by existing commercial applications. More accurate but less commercially successful forms have been proposed, such as the linear time-varying voltage source and associated resistance proposed in [5] and refined in [1]. In [4] the authors extend the Thevenin equivalent model of [5] to more accurately model nonlinear tails exhibited when driving highly resistive interconnect. While many cell models have been derived by measuring voltages, other models have focused on measuring current flow into or out of a cell. Examples of these models include [6] and [7].

Once the voltage waveform produced by a cell has been determined, it must be processed by an interconnect model to determine the voltage waveform to pass to the next cell model in the path. Moment-matching techniques have become the de facto standard in the derivation of interconnect models. Asymptotic Waveform Evaluation, or AWE [11], uses the moments of the step response to approximate the dominant poles of a linear interconnect circuit.

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 2003, June 2-6, 2003, Anaheim, California, USA.
Copyright 2003 ACM 1-58113-688-9/03/0006 ...\$5.00.

Techniques pioneered by RICE [12] have been incorporated into many different interconnect analysis applications in which the driving voltage can be approximated by a linear ramp. Other techniques, like recursive convolution [2] and FTD [8] were developed to accommodate more complex voltage waveforms, though their performance rapidly degrades as the number of segments describing the waveform increases.

Unfortunately, existing voltage-based cell models are unable to cope with effects such as noise or highly resistive interconnect that can significantly alter the shape of the input voltage waveform from the waveform used during characterization, leading to inaccurate timing analysis. Existing current-based models, while more dynamic in nature, can still exhibit large variations from SPICE when presented with complex interconnect models or non-monotonic input voltage waveforms. This paper introduces a new model, Blade, that meets these specifications.

In order to take advantage of the accuracy associated with a new cell model, arbitrary voltage waveforms must be accurately propagated through an interconnect model. Both FTD and recursive convolution can propagate arbitrary waveforms, but these approaches are only efficient when the volume of data is small. Razor offers a new, more effective model.

3. Blade

There are two components of the Blade runtime environment: the model and the runtime engine. The runtime engine uses transient analysis to evaluate the Blade current model.

3.1 The Blade Model

The Blade model, as shown in Fig. 1, consists of a voltage-controlled current source, an internal capacitance ($C_{internal}$), and a time shift of the output waveform. This model represents the electrical performance from an input pin to an output pin under the conditions from which the current source was derived.

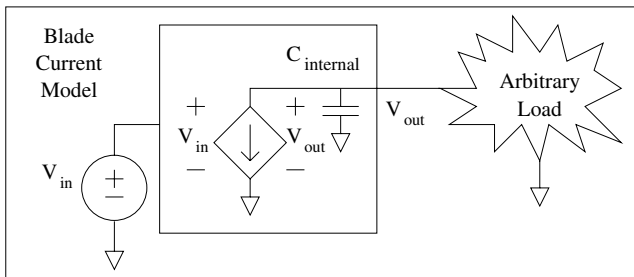


Figure 1: The Blade model consists of a voltage-controlled current source with a constant internal capacitance and input waveform time shift driving an arbitrary load.

Derivation of a Blade model is accomplished in two steps. The first step is to determine the amount of current sourced by a cell in response to voltage levels on the input and output pins of interest, $i_{out}(V_{in}, V_{out})$. Specifically, for a given process corner, voltage, and temperature, a DC voltage supply is attached to the input pin of interest and another to the output pin of interest. The two voltage sources are then swept from V_{ss} to V_{dd} and the current sourced by the cell's output pin is measured to create an I-V table.

The Blade runtime engine uses the I-V table to determine a cell's transient response to input voltage waveforms and output pin loading. However, a response derived exclusively from the DC-based I-V table results in an overly optimistic timing analysis as the DC sweep of the input and output ignores the effects of parasitic elements within the cell. The effects of these parasitics must be in-

cluded within the final Blade model. The calibration program determines these parameters.

Calibration of the Blade model involves the determination of an internal capacitive load ($C_{internal}$) which, when applied to the Blade model, results in a transient waveform that matches a SPICE-generated waveform for the cell under identical conditions. Once the waveform shapes have been matched, a time shift is calculated by examining the time difference between the 50% points of the SPICE output and the calibrated Blade output. A single SPICE transient analysis is sufficient to calibrate that input/output pin pair for any arbitrary input waveform and output load.

Compound cells (cells that consist of two or more logic functions like the OR gate or the AND gate) require this process to be performed twice: once for the logic function at the input and once for the logic driving the output. Thus, an OR gate model would consist of two current sources and require two calibration runs.

3.2 The Blade Runtime Engine

The Blade model is evaluated by the Blade runtime engine. The runtime engine accepts as input an arbitrary voltage waveform and output load and produces the voltage response at the output pin. This engine consists of a small nonlinear solver based on secant iteration. Unlike Newton-Raphson iteration, secant iteration can solve systems of equations when lookup tables are employed. Furthermore, secant converges upon an answer only slightly slower than Newton-Raphson [9].

Like SPICE, Blade can drive any arbitrary load including simple capacitive loads, π -models, or complex interconnect networks. Techniques employed to optimize SPICE for speed can also be applied to Blade for similar performance enhancements including variable time-step sizes, error tolerance specification and matrix pivoting and reduction techniques. Additionally, interconnect reduction techniques such as those pioneered in AWE can also be applied.

3.3 Noise Immunity

Cells differ from one another in the way that they respond to voltage waveforms that include distortions due to noise. Some cells are relatively noise tolerant while others exhibit dramatic variations in the shape of the output voltage waveforms. The response of a cell to noise imposed on its input signal is largely dependent upon the intrinsic noise immunity of the cell.

The standard formulation of the Blade model, consisting of a voltage-controlled current source and an internal capacitor, unintentionally disregards the noise immunity characteristics of a cell. To alleviate the problem of incorrect noise modeling, an enhancement to the Blade model, called the noise immunity filter, has been developed. This noise rejection filter conditions the input signal to correct for the lost noise immunity in the model. Specifically, a single-pole low-pass filter is used, thereby reducing the undesirable high-frequency transfer characteristic of the original Blade model. This filter can be thought of as a simple RC segment applied to the input signal prior to use by the Blade model.

4. Razor

The output of a Blade transient analysis is a time-indexed voltage array. While the Blade engine can operate using either variable or fixed time steps, time steps within the current implementation are uniform. Thus, for a 1 nanosecond transient analysis period with a 1 picosecond time-step, a 1000-segment piecewise linear (PWL) description is computed. This PWL represents the voltage waveform that must propagate through the interconnect accurately and efficiently to determine the voltage waveform at each cell being driven. By taking advantage of the fixed step-size, a simple and efficient calculation of the voltage at a node in the interconnect can

be done. Note that a time-indexed array in which the time steps are not uniform can easily be converted into an array with a constant time step.

Using a moment-matching technique such as AWE, n^{th} -order reduced interconnect models can be constructed for each interconnect sink (the point at which the interconnect and cell being driven are connected) driven by $v_{in}(t)$. Given n poles and residues of a sink node (p and k , respectively) driven by a saturated ramp input voltage of slope A , the voltage at the sink at time t , $v(t)$, can be described by a reduced-order model of the form given in (1) where τ is the point in time after which the voltage remains constant.

$$v(t) = \begin{cases} A[t - \sum_{i=1}^n \frac{k_i}{p_i} (1 - e^{p_i t})] & \text{if } t < \tau \\ A[t - \sum_{i=1}^n \frac{k_i}{p_i} (1 - e^{p_i t})] - \\ A[(t - \tau) - \sum_{i=1}^n \frac{k_i}{p_i} (1 - e^{p_i (t-\tau)})] & \text{if } t \geq \tau \end{cases} \quad (1)$$

Recursive convolution extends this concept to accommodate a PWL description consisting of many individual segments. Razor is a novel implementation of recursive convolution that takes advantage of the fixed time-step size used in Blade to calculate efficiently and accurately the voltage waveforms at each interconnect sink node. Razor is an $O(sp)$ algorithm where s is the number of time-steps and p is the number of poles.

Razor achieves its speed through the calculation of partial sums of the fully-factored saturated ramp equation given in (1). Consider a two-pole representation of a RC interconnect sink with an infinite ramp of slope A driving it. For a fixed time-step size of Δt , $v(t) = v(i\Delta t)$. The contribution of PWL segment j after time $i\Delta t$ (where $i \geq j$) is given by (2).

$$A_j[(i-j)\Delta t - (\frac{k_1}{p_1} - \frac{k_1}{p_1} e^{p_1(i-j)\Delta t} + \frac{k_2}{p_2} - \frac{k_2}{p_2} e^{p_2(i-j)\Delta t})] \quad (2)$$

Rearranging terms yields (3).

$$A_j[(i-j)\Delta t - (\frac{k_1}{p_1} + \frac{k_2}{p_2}) + \frac{k_1}{p_1} e^{p_1(i-j)\Delta t} + \frac{k_2}{p_2} e^{p_2(i-j)\Delta t}] \quad (3)$$

The Razor interconnect model independently calculates each of these components and adds the partial sums.

$$v(t) = v(i\Delta t) = \sum_{m=1}^4 S_m(i\Delta t) \quad (4)$$

$$S_1(i\Delta t) = \sum_{j=1}^i A_j \Delta t = A_i \Delta t + S_1((i-1)\Delta t) \quad (5)$$

$$S_2(i\Delta t) = -A_i (\frac{k_1}{p_1} + \frac{k_2}{p_2}) \quad (6)$$

$$S_3(i\Delta t) = (A_i - A_{i-1}) \frac{k_1}{p_1} e^{p_1 \Delta t} + e^{p_1 \Delta t} S_3((i-1)\Delta t) \quad (7)$$

$$S_4(i\Delta t) = (A_i - A_{i-1}) \frac{k_2}{p_2} e^{p_2 \Delta t} + e^{p_2 \Delta t} S_4((i-1)\Delta t) \quad (8)$$

A similar derivation can be made for RLC circuits (with imaginary pole and residue components) as well as for higher order representations.

5. EXPERIMENTAL RESULTS

The Blade modeling system and runtime engine were written in C++ and compiled under RedHat Linux 7.1 using G++. Results are compared against runs under HSPICE 2001.4 for RedHat Linux

7.1. All runs, Blade and HSPICE, were made on a 1800+ AMD Athlon (1.533GHz clock speed) system using identical transient time-step sizes and limits. Only the transient analysis portion of the HSPICE run is presented for comparison. The CPU time associated with the other aspects of the HSPICE run are not considered in order to present an ‘‘apples to apples’’ comparison.

5.1 Cell Modeling

Blade models were created for cells in a 0.13 μ m, 1.5V production cell library using parasitically extracted netlists. The input and output voltages were swept from 0V to 1.5V in 0.05V increments, yielding a 31 \times 31 I-V table. HSPICE was used to perform both the DC simulation from which the I-V table data was measured and the calibration run.

An automated test system was constructed to validate the Blade model and runtime engine. Several hundred Blade models were constructed for a large variety of cell types and input/output pin combinations. These models were evaluated using a variety of different input voltage waveforms (from simple linear ramps to noisy voltage waveforms) and output loads (from lumped-C loads to 2-segment π -models), and their results were compared to HSPICE runs under identical conditions. In all cases the waveforms produced by the Blade models matched their HSPICE counterparts to within 1-2%. Figures 2 and 3 graphically depict typical HSPICE and Blade results.

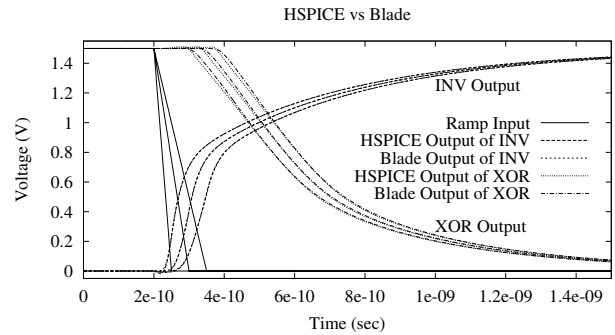


Figure 2: Three falling ramps drive an inverter loaded with π -model to create distinct tail. The output of the inverter drives a 2-input XOR with a π -model load.

Unlike SPICE, the Blade model runtime is independent of the number of transistors or parasitics in the cell netlist. The experimental results shown in Table 1 are for a 1 picosecond step size and a 1 nanosecond transient analysis period (1001 nonlinear iterations). On the average, 1 nonlinear iteration took 125 nanoseconds to complete. The OR gate took twice as long since it consists of 2 Blade models.

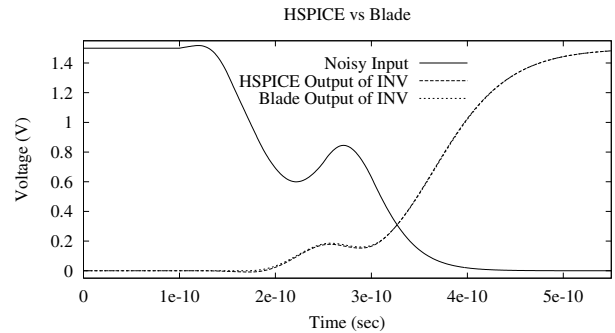


Figure 3: A noisy input waveform drives an inverter to produce an output waveform exhibiting noise.

Some cells, such as the INV, were highly sensitive to the presence of noise in the input waveform. Others, such as the largest AOI in the library, exhibited a high degree of intrinsic noise immunity. In the case of the AOI, SPICE-accurate results were achieved using a noise filter with RC time constant of 0.026 nanoseconds and a time offset of 0.113 nanoseconds.

| Cell | Number of Elements | | | | HSPICE Time | Blade Time | Speedup |
|------|--------------------|-----|---|-----|-------------|---------------|---------|
| | X | C | D | R | | | |
| AOI | 18 | 118 | 4 | 141 | 0.67s | 129 μ sec | 5,200 |
| INV | 8 | 59 | 1 | 72 | 0.35s | 125 μ sec | 2,800 |
| OR | 14 | 95 | 2 | 127 | 0.53s | 250 μ sec | 2,120 |
| XOR | 35 | 316 | 3 | 345 | 2.23s | 122 μ sec | 18,300 |

Table 1: Blade and HSPICE runtimes are shown for various cells. The number of transistors (X), capacitors (C), diodes (D), and resistors (R) within the parasitically extracted netlist for each cell is also shown.

5.2 Stage Modeling

Razor is written in C++ and can run as part of the Blade runtime engine to produce stage delay values or in a stand-alone mode to simply compute interconnect delay. When used in conjunction with Blade, the fixed-time-step PWL created by Blade during model evaluation is transparently passed to Razor to compute the interconnect delay. When used in stand-alone mode, the input can be any arbitrary PWL representing a voltage waveform. If the PWL does not use a fixed time-step, a new fixed-time-step PWL is created to approximate the input.

For a given number of segments to describe a voltage waveform and a given number of poles to represent the interconnect, Razor executes in a fixed time period. Whereas Blade uses a nonlinear solver to determine current flow from the Blade model to its load, Razor uses a purely deterministic solution for interconnect analysis.

Table 2 summarizes some of the performance results exhibited by Razor. Results are shown for a 1 nanosecond transient analysis using a 1 picosecond step size.

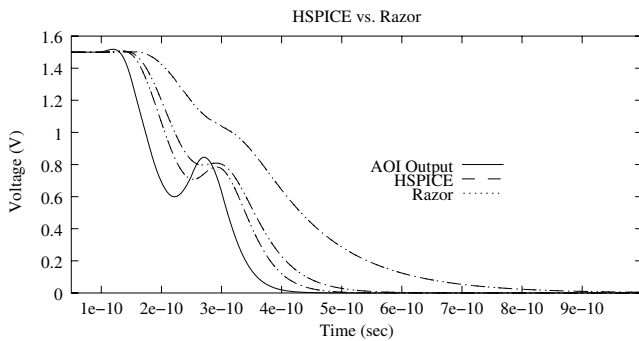


Figure 4: A noisy signal output by an AOI cell drives an interconnect load. Razor results at 3 terminal points of the interconnect are nearly identical to those produced when HSPICE modeled the cell and interconnect simultaneously.

As shown in Fig. 4, Razor is not limited to monotonic waveforms. The noisy waveform created by Blade in the previous section can be consumed by the Razor interconnect model just as easily and accurately.

| PWL Segments | Poles | Razor Time | HSPICE Time | Speedup |
|--------------|-------|---------------|-------------|---------|
| 1000 | 2 | 80 μ sec | 1.2s | 15,000 |
| 500 | 2 | 44 μ sec | 1.2s | 27,300 |
| 1000 | 4 | 115 μ sec | 1.2s | 10,500 |
| 1000 | 2 | 82 μ sec | 2.7s | 33,000 |
| 800 | 2 | 63 μ sec | 2.7s | 43,000 |

Table 2: Razor and HSPICE results are shown for sample RC trees.

6. CONCLUSIONS

Modeling requirements for accurate analysis of nanometer designs are growing faster than the ability to create models using the traditional characterization process. A new model that properly handles dynamic effects of noisy and elongated input waveforms and that evaluates rapidly and accurately is required to meet these demands. Blade is a high performance, accurate runtime model and evaluation engine that meets these challenges. Razor adds an interconnect model to create a comprehensive solution to the problem of analyzing stage delay. Together these innovative models set a new standard for cell and interconnect timing analysis.

7. REFERENCES

- [1] R. Arunachalam, F. Dartu, and L. T. Pileggi. CMOS gate delay models for general RLC loading. In *Proceedings 1197 IEEE International Conference on Computer Design: VLSI in Computers and Processors*, pages 224–229, October 1997.
- [2] J. Bracken, V. Raghavan, and R. Rohrer. Interconnect simulation with asymptotic waveform evaluation. In *IEEE Transactions on Circuits and Systems*, volume 39, pages 869–878, 1992.
- [3] J. F. Croix and D. F. Wong. A fast and accurate technique to optimize characterization tables for logic synthesis. In *34th IEEE/ACM Design Automation Conference Proceedings*, pages 337–340, 1997.
- [4] F. Dartu, N. Menezes, and L. T. Pileggi. Performance computation for precharacterized CMOS gates with RC loads. *IEEE Transactions on CAD*, pages 544–553, May 1996.
- [5] F. Dartu, N. Menezes, J. Qian, and L. T. Pillage. A gate-delay model for high speed CMOS circuits. In *31st IEEE/ACM Design Automation Conference Proceedings*, pages 576–580, 1994.
- [6] M. Hafeed, M. Oulmane, and N. C. Rumin. Delay and current estimation in a CMOS inverter with a RC load. In *IEEE Transactions on CAD*, volume 20, pages 80–89, January 2001.
- [7] A. Korshak and J. Lee. An effective current source cell model for VDSM delay calculation. In *IEEE International Symposium on Quality Electronic Design*, pages 296–300, 2001.
- [8] Y. Liu, L. T. Pileggi, and A. J. Strojwas. *ftd*: An exact frequency to time domain conversion for reduced order RLC interconnect models. In *35th IEEE/ACM Design Automation Conference Proceedings*, pages 469–472, 1998.
- [9] M. J. Maron. *Numerical Analysis: A Practical Approach*. Macmillan Publishing Co., Inc., 1982.
- [10] L. W. Nagel. *A Computer Program to Simulate Semiconductor Circuits*. PhD thesis, University of California, Berkeley, May 1975.
- [11] L. T. Pillage and R. A. Rohrer. Asymptotic waveform evaluation for timing analysis. In *IEEE Transactions on CAD*, volume 9, pages 352–366, 1990.
- [12] C. L. Ratzlaff and L. T. Pillage. RICE: Rapid interconnect circuit evaluation using AWE. In *IEEE Transactions on CAD*, pages 763–776, 1994.