SYMBOLIC ANALYSIS OF POWER/GROUND NETWORKS USING MOMENT-MATCHING METHODS*

Jatan C. Shah[1] Ahmed A. Younis[1] Sachin S. Sapatnekar[2] Marwan M. Hassoun[1]

[1]*Department of Electrical and Computer Engineering, Iowa State University, Ames, IA 50011, USA.*
[2]*Department of Electrical Engineering, University of Minnesota, Minneapolis, MN 55455, USA.*

Abstract — **This paper presents an event-driven algorithm for the symbolic analysis of power and ground bus networks using moment matching techniques to estimate the transfer function at each node in the P/G network. The P/G network is modeled by a hierarchical combination of mesh and tree structures that are composed of a collection of RC-π segments and pulldown (or pullup) switches. The switches are symbolically represented by Boolean variables and a compiled symbolic code is generated only once for each P/G network. The transient waveforms are then produced by repetitive evaluation of the symbolic output. The results show that the symbolic implementation is an order of magnitude faster, with reasonably good accuracy, than using a traditional analog circuit simulator like SPICE.**

I. INTRODUCTION

Switching activity in VLSI circuits introduces transient currents in the power and ground (P/G) buses. These currents produce two major effects: voltage drops across the buses and high current densities leading to electromigration of metal atoms. For the diagnosis of these problems, it is essential to find fast but accurate techniques to simulate the network. Moreover, the procedure of optimizing P/G buses to control voltage drops and electromigration problems requires repeated simulations under different sets of parameters.

Any procedure that requires repeated simulations is a good candidate for the application of symbolic analysis. This paper applies symbolic analysis using moment-matching approaches based on Padé approximants to speed up the computation. Our work produces a compiled code for a given circuit structure and enables a designer to run this compiled code with various values of the circuit parameters (namely, any list of switching events and any set of widths of the wire segments).

Traditionally, symbolic circuit analysis methods have focused on finding transfer function solutions for general topology circuits. However, the P/G network, as modeled herein, is a circuit with a specific mesh/tree topology and a specific transfer function construct. An application specific symbolic analyzer is developed in order to provide an efficient solution mechanism for this problem. Some of the general purpose programs use approximation algorithms to produce "visually" interpretable transfer functions [1, 2, 3] that are centered around specific component values and are not necessary accurate over a large range of component values. The goal of the symbolic analysis in this paper is to produce an efficient and accurate approximation that is valid over a large range of component values. The "visual" nature of the final transfer function is of no value here due to the large sizes of P/G networks. Hence, a sequence of expressions approach was used to generate the symbolic code in order to maintain the accuracy of the functions with a linear growth in the number of arithmetic evaluations in the generated code [4], [5].

The moment-matching techniques for circuit analysis based on Padé approximants, which are used in this work, gained popularity after the publication of the Asymptotic Waveform Evaluation (AWE) approach [6]. These methods approximate the exact transfer function of the circuit by a reduced order transfer function that captures the effects of these dominant poles. While no theoretical guarantees of accuracy or stability exist, it has been seen practically that moment-matching approaches have been hugely successful in approximating large interconnect systems.

Various algorithms and simplified device models for P/G buses that offer faster but less accurate results have been explored in the past, such as [7, 8]. The proposed approach presents a fast yet accurate simulation technique.

II. MODELING P/G BUSES

The design of power buses and ground buses that convey the V_{dd} and GND signals, respectively, across the chip have essentially symmetric procedures. Therefore,

*This work is supported in part by NSF through grant MIPS-9502556 and by a grant from Texas Instruments, Inc.

we generically refer to the problem as the P/G bus problem. In our descriptions, we will chiefly focus on the design of ground bus, realizing that power buses may be analyzed in like fashion.

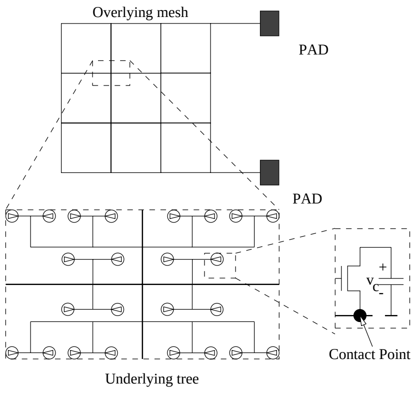

Figure 1: Typical ground bus topology

In large integrated circuits, the P/G buses may support a number of switching elements, each of which draws current from the power bus, or injects current into the ground bus during transitions. We assume an overlying mesh over the area of the chip to distribute the P/G signals, and to feed these signals locally using underlying local tree structures, as shown in Figure 1.

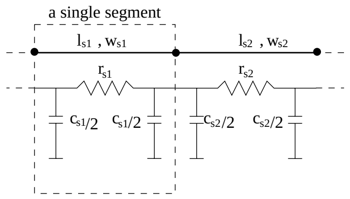

Figure 2: Wire segment model for P/G buses

Each wire on the mesh or tree structure in a P/G bus can be modeled as a set of connected π-segments as shown in Figure 2, where $r_s = \rho l_s / w_s$, $c_s = (\beta w_s + \alpha) l_s$. Here l_s and w_s are the length and the width of the segment, and the parameters ρ, β and α are the sheet resistance per square, sheet capacitance per unit area and the fringe capacitance per unit length of the metal layer that is being used for routing the P/G buses, respectively. The segments are connected at contact points (nodes), which may be located at the ends of segments.

To model the current drawn by gates, we first reduce each gate to an equivalent inverter whose transistors are modeled as a switch with a given on-resistance and infinite off-resistance. When turned on, this switch connects a charged capacitor to the ground net, thereby injecting a discharge current through it. The switches are controlled by a Boolean control variable that determines whether the switch is on or off, and the time points at which switches are turned on or off are assumed to be described by a user-specified event list.

III. OVERVIEW OF AWE

The Asymptotic Waveform Evaluation method approximates a high order system by a low order transfer function that captures the dominant poles of the system and their residues. The basic AWE algorithm [6] involves the computation of first few terms of the Maclaurin expansion of the transfer function of the system, and then matches these terms with the coefficients of a strictly proper reduced order n-pole rational function.

The remainder of this section will summarize the AWE procedure; for further details, the reader is referred to [9]. We begin with a modified nodal equation description for a circuit of the type:

$$C\dot{\mathbf{x}}(t) + G\mathbf{x}(t) = \mathbf{e}(t) \tag{1}$$

where $C, G \in \mathbf{R}^{n \times n}$, $\mathbf{x}, \mathbf{e} \in \mathbf{R}^n$, and n is the number of modified nodal variables in the circuit. For a circuit with zero initial conditions and a single excitation at node 1, the Laplace transform of $x(t)$ can be approximated by a Maclaurin series as

$$\mathbf{X}(s) = \mathbf{m}_0 + \mathbf{m}_1 s + \mathbf{m}_2 s^2 + \cdots = \sum_{i=0}^{\infty} \mathbf{m}_i s^i \tag{2}$$

It can be shown that this results in the following set of linear equations for each moment vector $\mathbf{m}_i$:

$$\begin{aligned} G\mathbf{m}_0 &= \mathbf{E} \\ G\mathbf{m}_i &= -C\mathbf{m}_{i-1} \text{ for } i \geq 1 \end{aligned} \tag{3}$$

It should be noted that G can be LU-decomposed once to solve for $\mathbf{m}_0$ and the higher order moment vectors can then be generated by forward and backward substitutions.

Further, it can be shown that by using a small number of moments generated, a low-order M-zero, N-pole ($M \leq N$) approximation to the actual system transfer function can be constructed in the form

$$\begin{aligned} \tilde{H}(s) &= \frac{A_0 + A_1 s + A_2 s^2 + \cdots + A_M s^M}{1 + B_1 s + B_2 s^2 + B_3 s^3 + \cdots + B_N s^N} \\ &= \sum_{j=1}^{N} \frac{\hat{k}_j}{s - \hat{p}_j} \end{aligned} \tag{4}$$

where the poles $\hat{p}_j$ approximate the set of dominant poles and $\hat{k}_j$ their residues. This is referred to as an $[M/N]$ Padé approximant. The coefficients of this reduced order approximation can be computed by matching them with the moments of the original transfer function. The values of the B_i's can be calculated by solving the following system of linear equations [6]:

$$\begin{bmatrix} m_{M-N+1} & m_{M-N+2} & \cdots & m_M \\ m_{M-N+2} & m_{M-N+3} & \cdots & m_{M+1} \\ \vdots & \vdots & \cdots & \vdots \\ m_N & m_{N+1} & \cdots & m_{M+N-1} \end{bmatrix} \begin{bmatrix} B_N \\ \vdots \\ B_1 \end{bmatrix} = \begin{bmatrix} -m_{M+1} \\ -m_{M+2} \\ \vdots \\ -m_{M+N} \end{bmatrix} \quad (5)$$

Once the B_i's have been calculated, the denominator polynomial of Equation (4) are determined, and the roots of this polynomial are the poles, $\hat{p}_j, j = 1 \cdots N$ in Equation (4). To find the residues corresponding to these poles, we expand the right hand side of Equation (4) to obtain [6]

$$m_0 = c - \sum_{j=1}^{N} \frac{\hat{k}_j}{\hat{p}_j} \qquad m_i = - \sum_{j=1}^{N} \frac{\hat{k}_j}{\hat{p}_j^{\,i+1}}, i \geq 1$$

where $c = A_N / B_N$ if $M = N$, and zero otherwise.

Using the relation in Equation (6), the following linear equations in the residues $k_1, \cdots, k_N$ can be derived and solved to completely define the approximating transfer function:

$$\begin{bmatrix} \hat{p_1}^{-1} & \hat{p_2}^{-1} & \cdots & \hat{p_N}^{-1} \\ \hat{p_1}^{-2} & \hat{p_2}^{-2} & \cdots & \hat{p_N}^{-2} \\ \vdots & \vdots & \cdots & \vdots \\ \hat{p_1}^{-N} & \hat{p_2}^{-N} & \cdots & \hat{p_N}^{-N} \end{bmatrix} \begin{bmatrix} \hat{k}_1 \\ \hat{k}_2 \\ \vdots \\ \hat{k}_N \end{bmatrix} = - \begin{bmatrix} m_0 - c \\ m_1 \\ \vdots \\ m_{N-1} \end{bmatrix} \quad \mathbf{(6)}$$

IV. OUTLINE OF THE SOLUTION

The input to the problem is a given P/G network topology and various technology parameters. The objective is to compute the voltage waveform at each node. The procedure can be divided into two steps that are executed every time one or more switch turns on or off:

(1) An efficient path tracing algorithm that exploits the structure of the P/G network is used to obtain V(s), a Maclaurin series approximation in the frequency domain to the actual voltage.

(2) Given V(s) at every node, the time domain response is then computed by approximating V(s) with a low order transfer function.

We also incorporate the presence of nonzero initial conditions in the P/G network.

We produce a compiled code containing the symbolic expressions that describe the circuit topology. The symbolic expressions are generated in sequence of expressions format in order to minimize the number of arithmetic operations needed for the code evaluation. It has been shown that for linear circuits, which is the case here, the number of symbolic variables grow linearly with the size of the circuit if a hierarchical sequence of expressions format is used in contrast to an exponential growth if a single expression transfer function is generated [4, 5]. Once this code is generated, the user may specify a set of parameters (any event list and any set of wire widths) and execute this code for those specified parameters. Due to the fact that the actual analysis is only performed once on the network and the repetitive numerical evaluations are performed using the symbolic code, this method is faster than repetitive full SPICE simulations on the same network.

The algorithm is event-driven, with events specified by the user at the time points at which one or more switch changes state. An interval spans the period between the specified time point of the current set of transitioning switches to the next switching activity in the circuit. The simulation now proceeds from the first interval to the last. Let the time points $t_1, \cdots, t_{10}$ be the points at which one or more nodes in the P/G net change states. The procedure begins at time 0, with all the initial conditions set to 0. The interval $(0, t_1)$ is then processed to compute the transient voltages at all nodes during that interval. At t_1, the final conditions due to the transient response in the previous interval are computed and are used as the initial conditions for the next interval. This procedure is repeated until all the intervals are processed.

V. SIMULATION OF ONE INTERVAL

The algorithm proceeds in three stages:
(1) *Reduction of the trees*: The underlying trees in the P/G net are first reduced to Padé-approximated Norton equivalents.
(2)*Solving the mesh*: The modified nodal formulation is then applied to solve for the Padé approximation for the voltage at each node in the mesh.
(3)*Propagating waveforms down the trees*: The computed voltages at the mesh nodes are propagated down the tree to compute all voltages within the tree.

Reduction of the trees

This section discusses in detail the computation of moments of the voltage at any given node by recursively reducing the subtrees rooted at the given node n to an admittance $Y_{T_i}(n)$ and a current source $J_{T_i}(n)$, where $Y_{T_i}(n)$ is the equivalent admittance of the subtree T_i, as seen from node n, to the ground, and $J_{T_i}(n)$ is the combined effect of all the switch elements in the given subtree T_i. The equivalent current sources and admittances are computed for every node in linear time using a path tracing algorithm. Note that the Y_{T_i}'s and J_{T_i}'s are symbolic expressions in terms of the Boolean variables that control each switch (whose values are set at run-time by the user-specified event list) and the wire widths.

The algorithm proceeds from the leaf-nodes, which are the contact points at the terminating segments of the ground net. A terminating segment e_t of a tree T is a segment that connects exactly one contact point (leaf node) to the rest of the ground net. The admittance $Y(n)$, which is the equivalent downstream admittance at node n as seen from edge e_t, is the sum total admittance of all the switches connected to n, and the current source $J(n)$ represents the equivalent current source corresponding to the total effect of all the switches connected to node n, as shown in Figure 3.

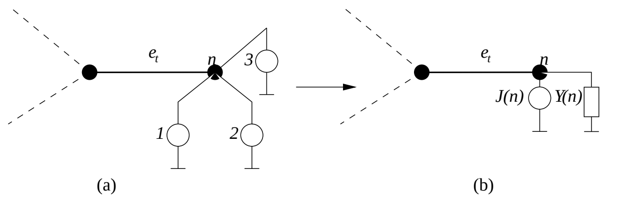

Figure 3: Computing equivalent devices at terminating segments

The equivalent devices $J(n)$ and $Y(n)$ can be computed using Norton's theorem. The equivalent current source of a single switch i with resistance R_d, connecting a load capacitance C_L to the network, is given by:

$$J^i = b \cdot \frac{V_c}{s} \frac{sC_L}{1 + sR_dC_L} \quad \mathbf{(7)}$$

where V_c is the initial voltage on the capacitor C_L at the start of the given interval, and b is a Boolean variable whose value is 1 if the switch is closed and 0 otherwise. The pole at the origin in Equation (7) indicates that the capacitance C_L with an initial voltage V_c is modeled as a voltage source $V_c \cdot u(t)$ in series with a capacitance C_L having zero initial conditions, where $u(t)$ is a unit step input, implying that the initial condition appears in the current expression only when the switch closes. Generating moments for $J(n)$ using the Maclaurin expansion of $\frac{1}{1+sR_dC_L}$, we get

$$J^i(n) = b \cdot V_c(C_L - R_dC^2{}_Ls + R^2{}_dC^3{}_Ls^2 + \cdots) \quad \mathbf{(8)}$$

The admittance of a closed switch is equal to the combined admittance of the driving resistance R_d and the load capacitance C_L:

$$Y^i = b \cdot \frac{C_Ls}{1 + R_dC_Ls} \quad \mathbf{(9)}$$

The moments of Y^i can be computed as

$$Y^i = b \cdot (C_Ls - R_dC^2{}_Ls^2 + R^2{}_dC^3{}_Ls^3 + \cdots) \quad \mathbf{(10)}$$

The total downstream admittance at node n as seen from segment e_t is given by $Y(n) = \sum Y^i \ \forall\ i \in$ Switches at n and the equivalent current source at n is $J(n) = \sum J^i \ \forall\ i \in$ Switches at n.

In the next step, the effect of the equivalent current sources is propagated to every node in the network. To illustrate the recursive step of moving up the tree, consider the situation in Figure 4. Given that the equivalent admittance and current source of all the switches connected to the terminating side of e_t are given by the formulæ for $Y(n)$ and $J(n)$ above, respectively, as applied at node u, the admittance and the equivalent current source, as seen into the edge e_t from node v is computed as follows. The equivalent admittance at v can be computed based on the method used in REX [10], an algorithm for handling trees with zero initial conditions.

$$\begin{aligned} Y(v) &= \frac{1}{R(e) + \frac{1}{(Y_0(u) + (Y_1(u) + C(e)/2)s + \cdots + Y_{2n-1}(u)s^{2n-1})} + \frac{C(e)}{2}} \\ &= \frac{Y_0(u) + (Y_1(u) + C(e)/2)s + \cdots + Y_{2n-1}s^{2n-1}}{(1 + R(e)Y_0(u)) + R(e)(Y_1(u) + C(e)/2)s + \cdots + R(e)Y_{2n}s^{2n}} + \frac{C(e)}{2} \\ &= Y_0(v) + Y_1(v)s + Y_2(v)s^2 + \cdots + Y_k(v)s^k + \cdots \quad \mathbf{(11)} \end{aligned}$$

where $Y_k(v)$ is given by

$$Y_k(v) = \frac{Y_k(u) - \sum_{i=1}^{k} R(e)Y_{k-i}(v)Y_i(u)}{1 + R(e)Y_0(u)} \quad \mathbf{(12)}$$

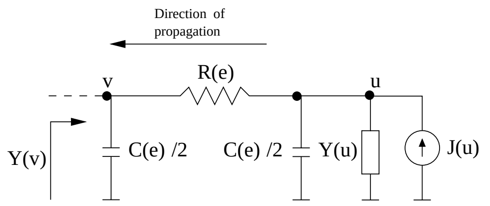

Figure 4: Reduction to equivalent current source and admittance

Unlike REX, we must deal with downstream current sources due to the presence of initial conditions. It can be shown that $J(v)$, which is equal to the short circuit current, is given by

$$J_k(v) = J_0(v) + J_1(v)s + J_2(v)s^2 + \cdots + J_k(v)s^k + \cdots$$

where

$$J_k(v) = \frac{J_k(u) - \sum_{i=1}^{k} R(e)J_{k-i}(v)Y_i(u)}{1 + R(e)Y_0(u)} \quad \mathbf{(13)}$$

Thus, we say that the equivalent admittance and current source have been *propagated* from node u to v. Each admittance and current source can be propagated when all downstream edges have been processed.

Solving the mesh

The overlying mesh M comprises p segments $me_1, \cdots, me_p$, and q mesh nodes $mn_1, \cdots, mn_q$, with a local tree T_{mn_i} rooted at each node mn_i, where the prefix m on each of these indicates that the node or edge lies on the mesh. The problem is to compute all the mesh node voltages $V_{mn_i}, i = 1, \cdots, q$, and then use the solution to compute the voltages at all the internal nodes within the subtrees. Using the technique presented above, the trees T_{mn_i} can be reduced to an equivalent admittance $Y(mn_i)$ and currents source $J(mn_i)$.

The expression in Equation (1) is modified to include higher order moments of $Y(mn_i)$ and $J(mn_i)$:

$$(G + Y)\mathbf{V} = \mathbf{J} \tag{14}$$

where G is the conductance matrix of the mesh edges (leaving out the load admittance equivalents of the i^{th} local subtree, $Y(mn_i)$), Y is a diagonal matrix with its i^{th} diagonal elements set to $Y(mn_i)$, and $\mathbf{J}$ is the vector of current sources which represent the equivalent current sources of the local subtrees at each mesh node. Expanding both sides of Equation (14) as a series in s, equating the coefficients of powers of s, and rearranging yields the following set of equations:

$$\begin{aligned} G^{'}\mathbf{V}_0 &= \mathbf{J}_0 \\ G^{'}\mathbf{V}_k &= \mathbf{J}_k - \sum_{j=1}^{k} Y_j \mathbf{V}_{k-j} \text{ for } k \geq 1. \end{aligned} \tag{15}$$

where

$$G^{'} = G + Y_0 \tag{16}$$

is the combined conductance matrix of the mesh segments and the resistive components of subtree admittances. The above sets of linear equations are used to compute the voltages at the mesh nodes.

The matrix $G^{'}$ can be LU-factorized while solving for V_0 and the subsequent moments can be computed by computing the RHS and performing one forward and backward substitution. Since the order of $G^{'}$ is typically much smaller than the total number of nodes in the circuit, solving for mesh voltages can be done cheaply.

Propagating waveforms down the trees

The solution obtained from Equation (15) is used to set the node voltages to the computed value. These values are used to recursively compute the voltage at each of the internal node in the local trees.

Consider a subtree T_{mn_k} rooted at the mesh node mn_k. Let the calculated voltage at mn_k be $V(mn_k)$. Then the current in $Y(mn_k)$ is given by,

$$I(mn_k) = V(mn_k) \times Y(mn_k) \tag{17}$$

$I(mn_k)$ is the total current flowing into the subtree T_{mn_k}. T_{mn_k} is progressively decomposed into multiple subtrees, and the current flowing into each of the subtrees is calculated as a fraction of $I(mn_k)$. For example, if the node mn_k was connected to r edges in the tree T_{mn_k} then the tree T_{mn_k} can be decomposed into r subtrees $\{T_1, \cdots, T_r\}$ rooted at mn_k. The current flowing in the subtree T_i is

$$I_{T_i}(mn_k) = V(mn_k) \cdot Y_{T_i}(mn_k) \tag{18}$$

where $Y_{T_i}(mn_k)$ is the admittance of the subtree T_i, as seen from node mn_k.

Given a segment e_j connecting the nodes mn_k and n_j, and given that the current through the subtree rooted at node mn_k and includes e_j is computed as $I_{T_j}(mn_k)$ from Equation (18), the voltage at n_j is given by

$$V(n_j) = V(mn_k) - I_{T_j} R(e_j) \tag{19}$$

where $R(e_j)$ is the resistance of the edge e_j.

This process is continued until the voltages at all the nodes in the tree T_{mn_k} have been computed. In this situation where the node of interest has a nonzero initial condition, a simple application of the Final Value Theorem [11] shows that an $[N/N]$ approximant is essential to capture the initial condition. The moments are used to generate these $[N/N]$ Padé approximations at each node (using the methods described in Section III B), which is then used to generate the time domain approximations for the given interval.

VI. EXPERIMENTAL RESULTS

The symbolic algorithm was implemented in C++, and the results on several interconnect meshes were tested. We used the commercial simulator, HSPICE [12], to analyze the speed and accuracy of our approach. In the experiments shown here, the compiled code first attempts a fourth order Padé approximant at each node. It has been observed in the literature that for interconnect circuits, reducing the order of Padé approximants usually improves the stability of the solution (Chapter 5 of [9]). If the fourth order approximation is unstable, then successive lower order approximations (third order, second order, etc.) are attempted until a stable approximant is found; the stability is tested by applying the Routh-Hurwitz criterion [11]. Our experimental results showed that the program was never required to go below a second order approximant. The advantage of using approximants of at most order 4 is that closed-form formulæ can be used to calculate the roots

Circuit	# Nodes	$T_{COMPILE}$	T_{SIM}	T_{HSPICE}
Circuit 1	20	4.5s	0.33s	14.24s
Circuit 2	52	9.0s	0.81s	22.15s
Circuit 3	104	17.5s	1.58s	35.75s
Circuit 4	204	40.4s	3.29s	63.39s
Circuit 5	404	142.2s	7.78s	124.48s

Table 1: Execution time comparisons with HSPICE

of the transfer function denominator. We tested the use of higher order approximants, but observed that these did not improve the accuracy significantly and increased the execution time substantially. This is attributable to the slower iterative techniques that must be used to find the roots of equations of order ≥ 4.

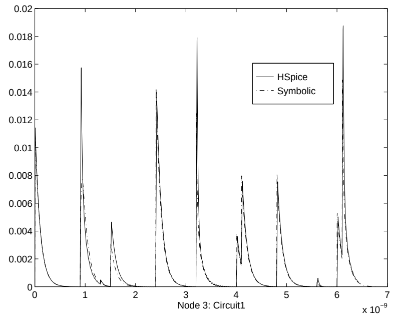

Figure 5: Waveform at node 3, Circuit 1

The CPU time of applying this method to five circuits, Circuit 1 through Circuit 5, of various sizes, are shown in Table 1. The quantity $T_{COMPILE}$ refers to the CPU compile time for the generated code, T_{SIM} shows the time required for one simulation of our compiled code, and T_{SPICE} represents the simulation time required by HSPICE. For a fair comparison, the same switch models were used in HSPICE as for our program, i.e., each switch was represented as a resistor and a capacitor, rather than by a transistor.

To illustrate the accuracy of this method, Figures 5 and 6 show the waveform at specific nodes for circuits Circuit 1 and Circuit 5, respectively.

VII. CONCLUSION

A new algorithm for the analysis of P/G network topologies using Padé approximations, handling nonzero initial conditions, has been presented. The application was found to be very amenable to symbolic analysis since the P/G network of each chip must be repeatedly analyzed by designers under different parameters and event sets before an optimal design is achieved.

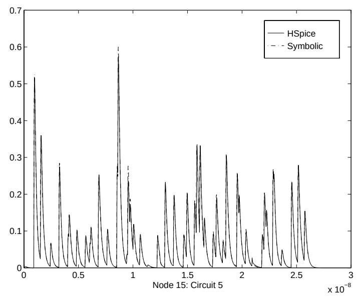

Figure 6: Waveform at node 15, Circuit 5

Experimental results show that the algorithm is accurate and that it achieves execution time improvements of more than an order of magnitude over SPICE.

References

[1] J. J. Hsu and C. Sechen, "Accurate extraction of simplified symbolic pole/zero expressions for large analog ICs," *Proc. ISCAS*, pp. 2083–2087, 1995.

[2] F. V. Fernández *et al.*, "Approximating nested format symbolic expressions," *Alta Frequenza Rivista Di Elettronica*, vol. 5, Nov. 1993.

[3] P. Wambacq, F. Fernández, G. Gielen, W. Sansen, and A. Rodríguez-Vázquez, "A family of matroid intersection algorithms for the computation of approximated symbolic network functions," *Proc. ISCAS*, pp. IV–806–IV–809, 1996.

[4] M. M. Hassoun and P.-M. Lin, "A hierarchical network approach to symbolic analysis of large-scale networks," *IEEE Trans. Circuits Sys. I*, vol. 42, pp. 201–211, Apr. 1995.

[5] B. S. Rodanski and M. Pierzchala, "Symbolic analysis of large electronic networks - an algebraic approach," *Proc. ISCAS*, pp. 2075–2078, 1995.

[6] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, Apr. 1990.

[7] S. Chowdhry and J. S. Barkatullah, "Estimation of maximum currents in MOS IC logic cirtuits," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 642–654, June 1990.

[8] U. Jagau, "SIMCURRENT- an efficient program for the estimation of current flow of complex CMOS circuits," *Proc. ICCAD*, pp. 396–399, 1990.

[9] E.-Chiprout and M.-S.-Nakhla, *Asymptotic Waveform Evaluation.* Boston, MA: Kluwer Academic Publishers, 1994.

[10] M. Sriram and S.-M. Kang, *Physical Design for Multichip Modules.* Boston, MA: Kluwer Academic Publishers, 1994.

[11] B. C. Kuo, *Automatic Control Systems.* Englewood Cliffs, NJ: Prentice-Hall, 6th ed., 1993.

[12] Meta-Software, Inc., *HSPICE user's manual, Version 96.1 for Release 96.1*, 1996.