

A Frequency Relaxation Approach for Analog/RF System-Level Simulation

Xin Li, Yang Xu, Peng Li, Padmini Gopalakrishnan and Lawrence T. Pileggi

Department of ECE, Carnegie Mellon University

5000 Forbes Avenue, Pittsburgh, PA 15213, USA

{xinli, yangx, pli, pgopalak, pileggi}@ece.cmu.edu

ABSTRACT

The increasing complexity of today's mixed-signal integrated circuits necessitates both top-down and bottom-up system-level verification. Time-domain state-space modeling and simulation approaches have been successfully applied for such purposes (e.g. Simulink); however, analog circuits are often best analyzed in the frequency domain. Circuit-level analyses, such as harmonic balance, have been successfully extended to the frequency domain [2], but these algorithms are impractical for simulating large systems with wide-band input and noise signals. In this paper we proposed a frequency-domain approach for analog/RF system-level simulation that is capable of capturing various second order effects (e.g. nonlinearity, noise, etc.) for both time-invariant and time-varying systems with wide-band inputs. The simulator directly evaluates the frequency domain response at each node via a relaxation scheme that is proven to be convergent under typical circuit conditions. Our experimental results demonstrate the accuracy and efficiency of the proposed simulator under various wide-band input and noise excitations.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids—Verification.

General Terms:

Algorithms

Keywords

System-Level Simulation, Analog/RF Circuits

1. INTRODUCTION

The topic of analog circuit simulation has been extensively studied ever since the advent of integrated circuits over three decades ago. SPICE [1] was developed and applied to analyze individual circuit blocks at the transistor level, both in the time and frequency domains. Recently, the remarkable evolution of the wireless/personal electronics market has introduced numerous new analog/RF products, as well as new challenges for the simulation of such systems. In order to conquer the increasing difficulties encountered in IC simulation, many advanced techniques, including steady-state analysis [2], envelope following [3], etc., have been developed. At the same time, the

advance of VLSI technologies has made it possible to integrate an entire mixed-signal system onto a single chip or within a single electronic package. It is, therefore, important to evaluate the performance of the full mixed-signal system during both top-down design and bottom-up verification. Various system-level simulation tools, such as FAST [4], SPW (Cadence), and COSSAP (Synopsys), have been developed to address this need for analog/digital co-simulation.

As IC technologies scale to finer feature sizes and circuit applications move to higher frequency bands, the behavior of analog/RF circuits becomes more complicated and more difficult to understand. Although only a small section of the entire mixed-signal system operates with truly analog signals, the design and verification of the analog components is generally the most challenging. Furthermore, design specifications are not only defined for individual analog/RF circuit blocks, but detailed high-level specifications are described for the entire analog/RF subsystem. For example, an analog front-end in the wireless transceiver is evaluated by several system-level specifications such as ACPR (adjacent channel power ratio). Such specifications require that the analog/RF subsystem is verified independently, as an intermediate stage between circuit-level analysis and mixed-signal system-level simulation. This hierarchical structure for mixed-signal system verification is illustrated in Fig. 1.

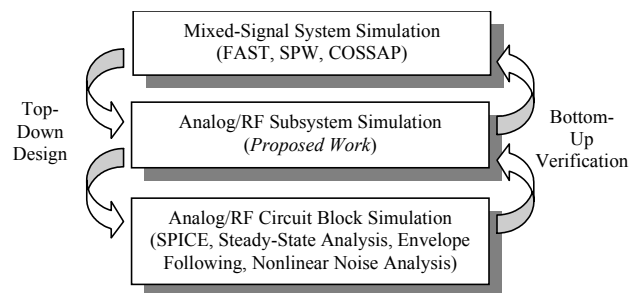


Fig. 1. Diagram for mixed-signal system verification

Unfortunately, directly applying or extending existing simulation techniques [1]-[4] to analog/RF system-level analysis suffers from several limitations. First, a complete analog/RF system consists of a large number of individual analog circuit blocks. As the system size increases, the traditional algorithms for circuit-level simulation do not accommodate the system-level simulation requirements. Secondly, but perhaps most importantly, the time-domain transient analysis used in FAST [4], SPW and COSSAP is effective for analog/digital co-simulation. However, for an analog/RF system, the wide-band input/output signals (e.g. the power spectral density for random noise) are best described by frequency-domain representations. Analyzing an analog/RF system in the time-domain over wide frequency bands can quickly become infeasible.

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, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2004, June 7–11, 2004, San Diego, California, USA
Copyright 2004 ACM 1-58113-828-8/04/0006...\$5.00.

In this paper we propose a frequency relaxation approach for analog/RF system-level simulation that accommodates both large system size and complex signal space. The proposed simulator is developed to capture various second order effects (e.g. nonlinearity, noise, etc.) for both time-invariant and time-varying (e.g. switching mixer) systems. Our simulator operates in the frequency domain and supports wide-band analog input (deterministic) signals (e.g. multi-tone sinusoidal signals) as well as wide-band noise (stochastic) signals. Our frequency-domain representation is also compatible with most system-level specifications, such as ACPR, noise figure, etc., which are defined by frequency-domain responses.

We prove that this proposed frequency relaxation approach is convergent under typical circuit conditions. While relaxation algorithms [5]-[7] have been studied for several decades, the novelty of our approach lies in our formulation of the system-level problem and our manipulation of the system components to achieve simulation *efficiency*. By exploiting the latency offered by the corresponding analog macromodels, we demonstrate that we can partition an entire analog/RF system into small unit-size blocks to facilitate the efficient and accurate analysis of complex response signals over wide frequency ranges.

2. LATENCY IN ANALOG/RF SYSTEMS

Circuit blocks/components, including those that are analog, are generally defined as part of a top-down design methodology. The blocks are designed to be weakly coupled to provide for their independent specification and creation. For analog/RF components there is also a dominant signal flow or propagation direction, which, along with the weak coupling, allows system designers to analyze them using a state-flow type of model [4].

For example, in a receiver front-end, the RF signal propagates through the LNA, mixer, IF amplifier, etc. By proper design, the parasitic coupling between these components is restricted to ensure that each component operates correctly. It follows that any backward signal propagation due to second order effects (e.g. nonideal coupling) is much weaker than the forward propagation. For such simulation models that are characterized by dominant unidirectional signal flow and blocks with high latency, the well-known relaxation methods [5]-[7] can be applied to exploit these properties. Namely, if the circuit blocks are solved individually in a proper order, a good approximate solution to the entire system is quickly produced after several iterations.

However, it is not sufficient to explore the latency only among circuit blocks. For numerical simulation, computation cost is determined by the circuit size, as well as the complexity of the signal space for representing the circuit response. An important difference between circuit-level analysis and system-level simulation is that, in system-level simulation, the response signal space is much larger. For example, a wireless transceiver front-end is tested with digitally modulated signals that contain a large number of frequency components and spreads over various (RF, IF and base) frequency bands when passing through the entire transceiver. Such a large signal space has to be completely considered during the simulation of each circuit block. Applying a relaxation approach facilitates the partitioning of large system into small blocks but, unfortunately, it cannot decompose the signal-response space simultaneously. Therefore, in system-level analysis, it is inefficient, if not impossible, to simulate each circuit block by traditional circuit-level techniques [2]-[3]. Circuit-level simulation algorithms such as harmonic balance

have been extended to the system level for frequency domain analysis of weakly nonlinear systems, but they cannot efficiently manipulate large signal spaces over wide frequency bands.

Various methods have been proposed recently to fill the analog/RF modeling void [8]-[10]. The primary purpose of macromodeling is to extract simple high-level abstractions that facilitate fast evaluation of nonideal effects in analog/RF circuits. However, from the relaxation point of view, the macromodeling process can also help to break the strong feedback loops inside a circuit that preclude us from decomposing the circuit block into smaller units. In analog/RF circuit design, feedback techniques are widely used in order to improve the circuit performance. These strong feedback loops are solved during the macromodeling process and the final macromodel incorporates the closed-loop input-output relation in an explicit form [8]-[10]. After macromodeling, a circuit block is further partitioned into much smaller units (e.g. static nonlinear functions and linear transfer functions in macromodels [8]-[9]), which can facilitate efficient system level simulation.

3. RELAXATION SIMULATION

Our proposed frequency-domain system-level simulation methodology is facilitated by a combination of macromodeling, partitioning, and frequency relaxation. In this section we will define the key features of this simulator and highlight the critical interdependencies between them. We start with the macromodeling strategy which makes identification of signal flow and feedback possible. We then describe our representations for handling the propagations of both deterministic input signals and stochastic noise signals. The core of our frequency relaxation approach is then described in Section 3.4, followed by an overview of our proof and conditions for convergence in Section 3.5.

3.1 Macromodeling of Circuit Blocks

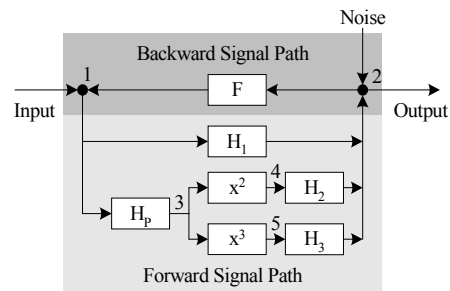


Fig. 2. Macromodel for a circuit block

Shown in Fig. 2 is a typical circuit block macromodel proposed in [8]-[10]. The macromodeling algorithms [8]-[10] approximate the circuit input-output relation by a number of static nonlinear functions (e.g. x^2 , and x^3) and linear transfer functions (e.g. F , H_p , H_1 , H_2 and H_3), and, therefore, decompose the entire circuit into much smaller units. The static nonlinear functions and linear transfer functions in Fig. 2 are determined by the circuit design of the current stage, the output impedance of the previous stage and the input impedance of the next stage. Due to a variety of nonidealities, signals in a circuit block might not propagate from input to output in an exactly unique direction. In other words, the reverse gain is nonzero and should be modeled by the backward signal path, as shown in Fig. 2. In addition, a noise

source is injected at the circuit output, which represents the total contribution of all noise sources (white noise, shot noise, etc.) inside the circuit. Such a noise source can be extracted by the macromodeling algorithm in [10].

It should be noted that today's analog/RF systems probably include time-varying components such as switching-mixers. These time-varying components bring about several different features here. First, the linear transfer functions in the macromodel are not restricted to traditional time-invariant ones. Linear periodically time-varying (LPTV) transfer functions should be applied to describe the input-output behaviors for time-varying circuits [9]:

$$H(t, j\omega) = \sum_{n=-\infty}^{\infty} H_n(j\omega) \cdot e^{jn\omega_0 t} \quad (1)$$

where $\omega_0 = 2\pi/T$ and T is the period of the LPTV transfer function. Secondly, due to the time-varying effects, the noise source in Fig. 2 is a cyclostationary process and, therefore, should be characterized by a set of harmonic power spectral densities [10]-[11].

3.2 Propagation of Deterministic Signal

The signals in an analog/RF system can be classified into two categories: deterministic signal (e.g. multi-tone sinusoidal signal) and stochastic signal (e.g. random noise). In this subsection, we describe the propagation for deterministic signals.

Taking the simple circuit block in Fig. 2 as an example, the relaxation simulator first injects the input signal at node 1. After that, the input signal propagates through three different paths $\{H_1\}$, $\{H_p \rightarrow x^2 \rightarrow H_2\}$ and $\{H_p \rightarrow x^3 \rightarrow H_3\}$, and all signals from these three paths add together at node 2. Next, the signal at node 2 returns node 1 through the backward path $\{F\}$ and further changes the signal value at node 1. The relaxation simulator propagates these signals back and forth inside the system. Eventually, if the reverse gain F is sufficiently small, all signals in the system become stable after several iterations. In other words, the relaxation simulation converges to the solution of the actual system response.

During the above signal propagation, the key operations are to compute the output signals when the input signals pass through basic macromodeling units, i.e. static nonlinear functions and linear transfer functions. Next, we describe the propagation properties for multi-tone sinusoidal signals and continuous spectra signals respectively.

3.2.1 Multi-tone Sinusoidal Signal

A multi-tone signal is the sum of a number of sinusoidal terms:

$$x(t) = \frac{1}{2} \cdot \sum_{i=1}^M (A_i e^{j\omega_i t} + A_i^* e^{-j\omega_i t}) \quad (2)$$

where $\omega_i > 0$ and A_i^* denotes the conjugate of A_i . When the multi-tone signal (2) passes through the static nonlinear block x^K , its output response contains various terms of the form [4]:

$$\prod_{i=1}^M C_{ai,bi} \cdot (A_i)^{ai} \cdot (A_i^*)^{bi} \cdot e^{j(ai-bi)\omega_i t} \quad (3)$$

where

$$\sum_{i=1}^M (ai + bi) = K \quad (ai \geq 0, bi \geq 0) \quad (4)$$

The coefficient $C_{ai,bi}$ can be computed using combinatorics [12].

On the other hand, given a multi-tone input (2), the output response $y(t)$ of the LPTV transfer function (1) is the sum of the individual responses to all sinusoidal tones, i.e.

$$y(t) = \frac{1}{2} \sum_{i=1}^M \sum_{n=-\infty}^{\infty} \left[A_i H_n(j\omega_i) e^{j(\omega_i + n\omega_0)t} + A_i^* H_n(-j\omega_i) e^{j(-\omega_i + n\omega_0)t} \right] \quad (5)$$

3.2.2 Continuous Spectra Signal

A digitally modulated signal has continuous frequency spectra $X(j\omega)$. When $X(j\omega)$ passes through the static nonlinear block x^K , its output response $Y(j\omega)$ equals the convolution of $X(j\omega)$ in frequency domain.

$$Y(j\omega) = \left(\frac{1}{2\pi} \right)^{K-1} \cdot \underbrace{X(j\omega) \otimes X(j\omega) \otimes \dots \otimes X(j\omega)}_K \quad (6)$$

Otherwise, if $X(j\omega)$ passes through the LPTV transfer function (1), the output response $Y(j\omega)$ can be expressed as:

$$Y(j\omega) = \sum_{n=-\infty}^{\infty} [H_n(j\omega - jn\omega_0) X(j\omega - jn\omega_0)] \quad (7)$$

3.3 Propagation of Stochastic Signal

Due to the time-varying effects in today's analog/RF systems, random noise is described as a cyclostationary stochastic process whose power spectral density is a time-varying function [10]-[11]:

$$X(t, j\omega) = \sum_{n=-\infty}^{\infty} [X_n(j\omega) \cdot e^{jn\omega_0 t}] \quad (8)$$

where $X_n(j\omega)$ is the n th harmonic power spectral density of X .

There are several key differences between deterministic signals and stochastic signals. First, since the amplitude of the physical noise signal is very small, nonlinearities can be ignored for noise analysis [10]-[11]. We only consider linear transfer functions when studying the propagation of random noise. Secondly, but more importantly, when two stochastic processes A and B are added, their power spectral densities can be added if and only if A and B are uncorrelated. Taking the simple circuit block in Fig. 2 as an example, the stochastic noise at node 2 propagates to node 1 through the path $\{F\}$ and then return node 2 through $\{H_1\}$. But, at node 2, the original noise signal and the returning noise signal cannot be added directly, since they come from the same source, thereby making them correlated. This observation motivates us to propose a new simulation scheme for noise analysis, as shown in Fig. 3.

Instead of propagating the noise signal directly, Step 2 in Fig. 3 "propagates" the transfer function from each noise source to the system output. When several transfer functions are parallel-connected at one node, the overall transfer function is equal to the sum of all individual ones. This implies the fact that transfer functions can be handled as response signals and propagated throughout the system during relaxation iteration. Taking the circuit in Fig. 2 as an example, we first initialize the transfer function from noise source to node 2 as $H_{Noise \rightarrow 2} = 1$. Then, $H_{Noise \rightarrow 2}$ cascades with F and propagates to node 1 which yields

$H_{Noise \rightarrow 1} = F$. Next, $H_{Noise \rightarrow 1}$ cascades with H_1 , returns node 2 and further changes the value of $H_{Noise \rightarrow 2}$. Note that we ignore the nonlinear signal paths $\{H_P \rightarrow x^2 \rightarrow H_2\}$ and $\{H_P \rightarrow x^3 \rightarrow H_3\}$ here. The relaxation simulator repeatedly applies these propagations until we obtain the closed-loop transfer functions from input to each node of the system.

1. For a system with N independent noise sources $X_i(t, j\omega)$ ($i=1, 2, \dots, N$), set $i=1$.
2. Apply relaxation iteration to compute the closed-loop transfer function $H_i(t, j\omega)$ from i th noise source $X_i(t, j\omega)$ to output.
3. Propagate the noise signal $X_i(t, j\omega)$ through the linear transfer function $H_i(t, j\omega)$ and compute the output power spectral density $Y_i(t, j\omega)$.
4. If $i < N$ then $i=i+1$ and return Step 2. Otherwise, go to Step 5.
5. The total noise power spectral density at the system output is the sum of the individual responses to all noise sources.

$$Y(t, j\omega) = Y_1(t, j\omega) + Y_2(t, j\omega) + \dots + Y_N(t, j\omega)$$

Fig. 3. Noise analysis algorithm

The above discussion shows that cascading different transfer functions is the key operation involved in Step 2 of the noise simulation. The following equation gives the overall transfer function $H(t, j\omega)$ when two LPTV transfer functions $F(t, j\omega)$ and $G(t, j\omega)$ are cascaded.

$$H(t, j\omega) = \sum_{n=-\infty}^{\infty} \left\{ \sum_{k=-\infty}^{\infty} [G_{n-k}(j\omega + jk\omega_0) F_k(j\omega)] \cdot e^{jn\omega_0 t} \right\} \quad (9)$$

After the transfer function $H(t, j\omega)$ from noise source $X(t, j\omega)$ to the output is obtained, the n th harmonic power spectral density $Y_n(j\omega)$ at the output can be expressed as a function of the input harmonic power spectral densities $X_k(j\omega)$ [11].

$$Y_n(j\omega) = \sum_{m=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \left\{ H_m(-j\omega - jm\omega_0) \cdot X_k(j\omega + jm\omega_0) \cdot X_{n-m-k}(j\omega + jm\omega_0 + jk\omega_0) \right\} \quad (10)$$

Studying (10), one would find that the input noise components at various frequencies $\{\omega + m\omega_0; m = \dots, -1, 0, 1, \dots\}$ will mix to the output at frequency ω . In addition, the k th input harmonic component $X_k(j\omega)$ will translate to the n th output harmonic component $Y_n(j\omega)$. These two features are the key differences between time-varying and time-invariant systems. Compared with the traditional noise simulation approach for LTI systems, our relaxation simulator is capable of accommodating such a noise folding effect involved in today's analog/RF systems.

3.4 Algorithm for Scheduling and Execution

After parsing the input netlist, our simulator converts the given analog/RF system to a signal flow graph; i.e. a weighted directed graph $G=(V, E)$ without multi-edges and self-loops*.

* In our simulator, multi-edges between two vertices are merged into a

Each vertex $V_i \in V$ denotes an input, output or internal node, each edge $E_i = \langle V_m, V_n \rangle \in E$ denotes a signal path from vertex V_m to V_n , and the weight of E_i stands for the static nonlinear function, linear transfer function or their combinations associated with the signal path.

It is well known that scheduling the signal flow graph in a proper order can speed up the convergence of the relaxation (e.g. Gauss-Seidel) iteration [5]-[7]. Remember that signal flows in an analog/RF system almost propagate in a unique direction. If the system is solved along the same direction in which signals propagate, a good approximate solution can be reached quickly. This is the essence of the scheduling algorithm given in Fig. 4.

After scheduling, we obtain an ordered sequence of vertices $\{V_1, V_2, \dots, V_N\}$. Either the Gauss-Jacobi or Gauss-Seidel iteration can be applied for relaxation simulation. In our simulator, the Gauss-Seidel algorithm is implemented, since the convergence analysis in Section 3.5 shows that Gauss-Seidel iteration is more efficient (converges more quickly) than the Gauss-Jacobi approach in our application. Fig. 5 shows the detailed algorithm for Gauss-Seidel iteration.

1. Partition the edge-set E into two subsets E_F and E_B such that $E_F \cap E_B = \emptyset$, $E_F \cup E_B = E$, and both $G_F = (V, E_F)$ and $G_B = (V, E_B)$ are directed acyclic graphs (DAGs). E_F contains all forward signal paths through which analog/RF signals propagate from input to output, and E_B contains all backward signal paths that feed signals back to previous stages.
2. Apply the One-Way Ordering algorithm (Algorithm 8.1.1 in [7]) to order graph $G_F = (V, E_F)$. This step results in an ordered sequence of vertices $\{V_1, V_2, \dots, V_N\}$ for further relaxation iterations.

Fig. 4. Algorithm for scheduling a signal flow graph

1. Set the initial response X_i^0 at each vertex V_i equal to 0.
2. Set iteration index $k=1$.
3. For each vertex V_i in the ordered sequence $\{V_1, V_2, \dots, V_N\}$, compute the signal value X_i^k at vertex V_i based on the input excitation at V_i , the signal $X_j^k (j < i)$ and $X_j^{k-1} (j > i)$ at each of its predecessors V_j and the weight associated with the edge $\langle V_j, V_i \rangle$.
4. Compute the difference $\Delta X_i = |X_i^k - X_i^{k-1}|$ between two successive iterations for each vertex V_i .
5. If $\Delta X_i < \epsilon$ for all $\{i=1, 2, \dots, N\}$, then stop iteration. Otherwise, set $k=k+1$ and go to Step 3.

Fig. 5. Algorithm for Gauss-Seidel iteration

3.5 Convergence Analysis

Without loss of generality, the system equation of a signal flow graph can be written as

single edge and a self-loop is broken by inserting a new vertex.

$$X = HX + W \quad (11)$$

where X_i is the response signal at vertex V_i , W_i is the input signal to V_i , and H_{ij} is the operator associated with edge $\langle V_j, V_i \rangle$. Note that the diagonal elements in matrix H are 0, i.e. $H_{ii} = 0$, since we assume that there are no self-loops in the signal flow graph.

In order to study the convergence property of the Gauss-Seidel iteration, we partition the operator matrix H into

$$H = L + U \quad (12)$$

where L is a strictly lower triangular operator matrix corresponding to the backward signal paths and U is a strictly upper triangular operator matrix associated with the forward signal paths.

Given a system described by (11) and operators L and U defined in (12), [13] proves that the Gauss-Jacobi iteration error is bounded by

$$|X^{k+1} - X^*| \leq |H| \cdot |X^k - X^*| \quad (13)$$

and the Gauss-Seidel iteration error is bounded by

$$|X^{k+1} - X^*| \leq (I - |L|)^{-1} \cdot |U| \cdot |X^k - X^*| \quad (14)$$

In (13) and (14), I is the identity matrix and X^* is the exact solution of the system response. The notation $|A|$ denotes a vector/matrix whose elements correspond to the norms of the elements in A , i.e. $|A|_{ij} = |A_{ij}|$. It is easy to verify that $|H| = |L| + |U|$ and $|H|$ is a nonnegative matrix, i.e. all elements in $|H|$ are nonnegative.

Based on (13) and (14), the convergence conditions for the Gauss-Jacobi and the Gauss-Seidel iterations are $\rho\{|H|\} < 1$ and $\rho\{(I - |L|)^{-1} \cdot |U|\} < 1$ respectively, where $\rho\{A\}$ stands for the spectral radius of matrix A . In order to further compare the convergence for these two iteration schemes, we need one additional theorem from matrix analysis.

Stein-Rosenberg Theorem [5]: Let $|H| = |L| + |U|$ be a nonnegative matrix with zero diagonal entries. If the spectral radius $\rho\{|H|\} < 1$, then $\rho\{(I - |L|)^{-1} \cdot |U|\} < \rho\{|H|\} < 1$.

The Stein-Rosenberg theorem tells us an important fact that, since there are no self-loops in the signal flow graph, i.e. $H_{ii} = 0$, the Gauss-Seidel iteration is more efficient (converges more quickly) than the Gauss-Jacobi approach in our application. Furthermore, the Stein-Rosenberg theorem also provides us with some physical intuition about the convergence. *Roughly speaking, Gauss-Seidel iteration converges as long as the signal flow graph for the original analog system does not contain a closed-loop gain larger than 1.* It is again worth mentioning that, because of both top-down design methodology and macromodeling technique, signal flows in an analog/RF system almost propagate in a unique direction and feedback signal paths are very weak. The above convergence analysis provides the theoretical background to explain why the relaxation approach works well for such cases.

4. NUMERICAL EXAMPLES

Shown in Fig. 6 is a GSM receiver system in 0.25 μ m TSMC CMOS process. We utilize this simplified GSM receiver as an example. Since traditional algorithms cannot efficiently simulate large systems, using a full GSM receiver will preclude us from comparing our simulator with traditional circuit simulation techniques. However, since the relaxation approach partitions a large problem into a number of small ones, we expect that the efficiency of our relaxation simulator will be more pronounced as the system size increases.

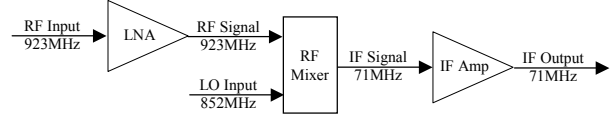


Fig. 6. Block diagram for GSM receiver

We first build the high-level macromodel for each circuit block in Fig. 6, by using the algorithm in [9]-[10]. The macromodel for each circuit block contains forward signal paths, as well as backward signal paths due to the nonideal couplings. In this example, the reverse gain for a circuit block is around 20dB-40dB less than the forward gain at the center frequency. Next, based on these pre-characterized macromodels, the proposed frequency relaxation simulator is used to simulate the entire system with various input excitations. All the simulations are performed on a Sun Sparc - 450MHz server.

4.1 Multi-tone Sinusoidal Simulation

We simulate the GSM receiver system with sinusoidal inputs. The input excitation is a single-tone sinusoidal signal in the simulation. However, due to the nonlinearities, many harmonic components are generated when such a single-tone signal passes through the receiver. For comparison, we describe the same macromodel for the GSM receiver by Verilog-A and run periodic steady-state (PSS) analysis in SpectreRF with the same error tolerance. TABLE I summarizes the simulation results for both approaches.

From the results in TABLE I, one finds that the number of iterations required by the relaxation method is more than that by the Newton method in SpectreRF. However, since each relaxation iteration partitions the large system into much smaller units and thereby reduces the computation cost significantly, our relaxation simulator eventually achieves about 1-2 orders of magnitude of runtime improvement.

TABLE I. Simulation results for multi-tone sinusoidal signals

Circuit	Harmonic #	Iteration #		CPU Time (Sec.)	
		Spectre	Relax	Spectre	Relax
LNA	5	3	8	0.44	0.015
Mixer	100	3	5	360.06	0.25
IF Amp	5	3	4	0.25	0.015
Receive r	100	4	8	626.33	1.08

4.2 Continuous Spectra Simulation

A QAM-16 modulated signal is applied to the input port of the GSM receiver. The carrier frequency is 923MHz. First, we describe the GSM receiver by means of a signal flow graph in Matlab Simulink and simulate the system in the time domain. An FFT is applied to the time-domain output waveform, resulting in

the frequency-domain spectrum shown in Fig. 7. The overall computation time is 73.38 seconds for such a time-domain simulation approach.

Next, we run frequency-domain simulation directly with our relaxation simulator. The QAM-16 modulated signal is represented by its continuous frequency spectrum with 400 sampling points in frequency domain. After running continuous frequency spectra simulation for 6.09 seconds, the output frequency spectrum of the GSM receiver system is obtained in Fig. 7. Note that the simulation results from both approaches are nearly identical, while a runtime improvement of more than 11X is achieved by the proposed relaxation simulator.

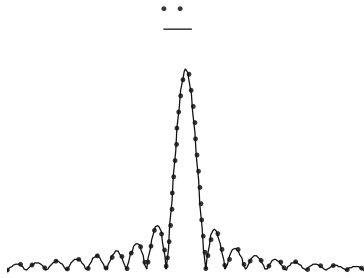


Fig. 7. Frequency spectrum for GSM receiver output

4.3 Noise Simulation

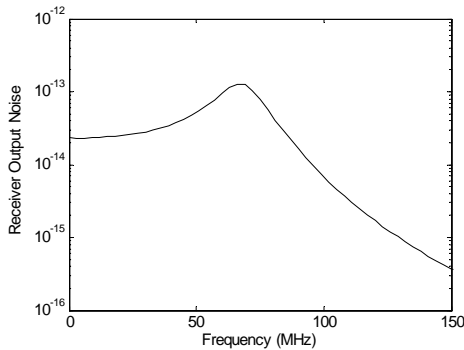


Fig. 8. Stationary noise PSD at GSM receiver output

Finally, we run noise analysis for the entire GSM receiver system. The noise macromodels for each circuit are extracted by the algorithm in [10]. After that, cyclostationary noise sources are added at the output port of each circuit block.

During noise simulation, the relaxation simulator first computes time-varying transfer functions from each input noise source to system output. In this example, a time-varying transfer function $H(t, j\omega)$ is represented by a set of harmonic transfer functions up to 7th order, i.e. $\{H_n(j\omega), n=0, \pm 1, \dots, \pm 7\}$ in (1). For relaxation iteration, each $H_n(j\omega)$ is approximated by a piecewise-linear representation with 1000 sampling points in frequency domain. Next, the output noise value is evaluated based on the harmonic power spectral densities of input noise sources and the computed time-varying transfer functions from each input to output. Fig. 8 shows the power spectral density for the stationary (time-average) noise component at the GSM receiver output. The overall computation time for noise analysis is 247.78 seconds. Note that, such cyclostationary noise analyses based on macromodels are impractical for existing system-level simulation

environments. Therefore, there is no comparison with other methods.

5. CONCLUSIONS

As analog/RF circuits become increasingly complex and have a crucial impact on the overall mixed-signal system performance, it is important to simulate and verify analog/RF systems independently at the system level. In this paper, we propose a frequency relaxation approach for analog/RF system-level simulation that is capable of capturing various second order effects (e.g. nonlinearity, noise, etc.) for both time-invariant and time-varying systems. Compared with traditional simulation methods, the proposed approach reduces the computation cost significantly by applying relaxation techniques to explore the latency from both top-down design methodology and macromodeling process. The efficiency of the proposed simulator is demonstrated by a GSM receiver system example in 0.25 μ m TSMC CMOS process. In summary, our relaxation simulator can be applied to verify analog/RF system performance and noise analysis effectively, as an intermediate stage between circuit-level and mixed-signal system-level verifications.

6. ACKNOWLEDGMENTS

This work has been supported by the MARCO Focus Center for Circuit & System Solution (under MARCO contract 2003-CT-888) and the Semiconductor Research Corporation (under contract 2003-TJ-1112).

7. REFERENCES

- [1] L. Nagel, "SPICE2: A computer program to simulate semiconductor circuits," Technical Report ERL-M520, UC-Berkeley, May, 1975.
- [2] K. Kundert, J. White and A. Sangiovanni-Vincentelli, *Steady-State Methods for Simulating Analog and Microwave Circuits*, Kluwer Academic Publishers, 1990.
- [3] A. Brambilla and P. Maffezzoni, "Envelope following method for the transient analysis of electrical circuits," *IEEE Trans. CAS-I*, vol. 47, pp. 999-1008, Jul. 2000.
- [4] P. Wambacq, G. Vandersteen, Y. Rolain, P. Dobrovolny, M. Goffioul and S. Donnay, "Dataflow simulation of mixed-signal communication circuits using a local multirate, multicarrier signal representation," *IEEE Trans. CAS-I*, vol. 49, pp. 1554-1562, Nov. 2002.
- [5] R. Varga, *Matrix Iterative Analysis*, Prentice-Hall Inc., 1962.
- [6] J. White, A. Sangiovanni-Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publishers, 1987.
- [7] A. Ruehli, *Circuit Analysis, Simulation and Design*, Elsevier Science, 1987.
- [8] P. Wambacq, P. Dobrovolny, S. Donnay, M. Engels and I. Bolsens, "Compact modeling of nonlinear distortion in analog communication circuits," *IEEE DATE*, pp. 350-354, 2000.
- [9] X. Li, P. Li, Y. Xu and L. Pileggi, "Analog and RF circuit macromodels for system-level analysis," *IEEE DAC*, pp. 478-483, 2003.
- [10] Y. Xu, X. Li, P. Li and L. Pileggi, "Noise macromodel for radio frequency integrated circuits," *IEEE DATE*, pp. 150-155, 2003.
- [11] J. Roychowdhury, D. Long and P. Feldmann, "Cyclostationary noise analysis of large RF circuits with multi-tone excitations," *IEEE JSSC*, vol. 33, pp. 324-336, 1998.
- [12] R. Sea, "An algebraic formula for amplitudes of intermodulation products involving an arbitrary number of frequencies," *Proc. IEEE*, vol. 56, pp. 1388-1389, Aug. 1968.
- [13] X. Li, Y. Xu, P. Li, P. Gopalakrishnan and L. Pileggi, "A frequency relaxation approach for analog/RF system-level simulation," Technical Report, No. CSSI 03-15, Carnegie Mellon Univ., Dec. 2003.