Modeling Digital Substrate Noise Injection in Mixed-Signal IC's

Edoardo Charbon, *Member, IEEE*, Paolo Miliozzi, Luca P. Carloni, *Senior Member, IEEE*, Alberto Ferrari, and Alberto Sangiovanni-Vincentelli, *Fellow, IEEE*

***Abstract*— Techniques are presented to compactly represent substrate noise currents injected by digital networks. Using device-level simulation, every gate in a given library is modeled by means of the signal waveform it injects into the substrate, depending on its input transition scheme. For a given sequence of input vectors, the switching activity of every node in the Boolean network is computed. Assuming that technology mapping has been performed, each node corresponds to a gate in the library, hence, to a specific injection waveform. The noise contribution of each node is computed by convolving its switching activity with the associated injection waveforms. The total injected noise for the digital block is then obtained by summing all the noise contributions in the circuit. The resulting injected noise can be viewed as a random process, whose power spectrum is computed using standard signal processing techniques. A study was performed on a number of standard benchmark circuits to verify the validity of the assumptions and to measure the accuracy of the obtained power spectra.**

***Index Terms*— Floorplanning, high-performance, mismatch, noise, parasitics, placement.**

I. Introduction

In the design of today's very large scale integration (VLSI) integrated circuits (IC's) noise immunity is becoming a major concern at all production levels. Increased chip complexity and speed in general tend to make the circuit more sensitive to both internal and external noise.

Feature miniaturization has been mainly responsible for dramatically reducing the distance between high-frequency noise sources and sensitive devices, the substrate being a major carrier of this type of spurious signals. The problem is particularly acute in mixed-signal circuits, where signals of different nature and strength interfere, thus affecting overall performance. Heavily over-designed structures are generally used to alleviate the problem, thus limiting the positive impact of advanced technologies. Furthermore, with the emergence of submicron technologies, the problem of high-speed substrate noise interference may compromise performance at the nominal frequencies for which the circuits were designed. These are the main reasons why substrate modeling has recently received renewed attention from designers attempting to integrate radio-frequency analog and baseband digital circuitry on a single chip.

A signal transition occurring in a typical logic gate causes a spike of current to be absorbed from the supply and to charge a load. A similar spike traveling toward ground is generated when the load is discharged. A significant portion of transitional current is discharged to ground through direct feedthrough. Spurious currents can also be injected directly into the substrate through various mechanisms [1]. The cumulative effect of spurious microcurrents absorbed/discharged by switching gates, is referred to as *switching noise*. Switching noise can quickly travel through interconnect coupling, power/ground busses and substrate, to be picked up by sensitive devices through capacitive coupling and body effect.

Any given circuit injects a unique current waveform into the substrate as a direct consequence of switching noise. Such waveform, known as *substrate noise signature*, is dependent on the circuit implementation, technology and input vector set. Evaluating accurate substrate noise signatures is useful to speed up the estimation of the impact of physical design on performance. Such estimates are often critical during architecture definition, floorplan and placement phases, where they are used to drive a number of design optimizers. Unfortunately, the level of complexity reached by today's digital circuits has made exact waveform characterizations impractical. To overcome the complexity problem, substrate noise signatures are often approximated by a single Gaussian white or pink noise source. The underlying assumption is that the global switching activity of the circuit is uniformly distributed over a large section of the spectrum. Substrate noise signatures have also been modeled in the literature [2], [3] as a capacitively and/or resistively coupled current or voltage generator whose waveform is derived from the circuit's global clock. The accuracy of these models is often the main limiting factor in the circuit performance. Simple approximations for injected noise often capture only a relatively small portion of the entire noise energy spectrum. Thus, potentially detrimental noise components may be underestimated.

In this paper a methodology is proposed to accurately evaluate substrate noise signatures in arbitrary circuits. The restriction to substrate noise is by no means binding and a similar scheme can be applied to compute supply and ground

Manuscript received January 30, 1998; revised June 8, 1998. This paper was recommended by Associate Editor D. Ling.

E. Charbon is with Cadence Design Systems, Inc., San Jose, CA 95134 USA (e-mail: charbon@cadence.com).

P. Miliozzi is with Conexant Systems, Newport Beach, CA 92660 USA.

L. P. Carloni and A. Sangiovanni-Vincentelli are with the University of California at Berkeley, Berkeley, CA 94720 USA.

A. Ferrari is with PARADES Labs, 00186 Rome, Italy.

Publisher Item Identifier S 0278-0070(99)01595-X.

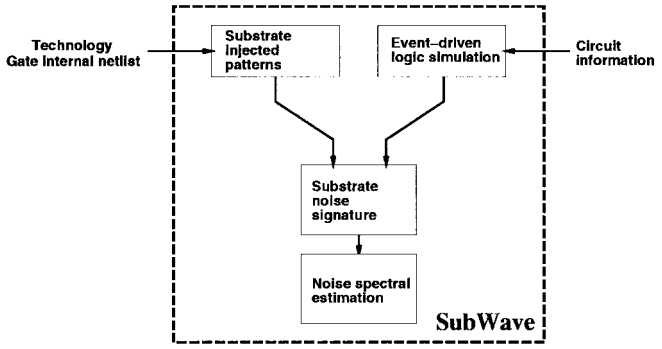


Fig. 1. Flow diagram of SUBWAVE.

ripple noise. The methodology exploits the fact that any given logic gate injects a particular signal into the substrate through capacitive coupling and impact ionization. Such signal, known as *substrate injection pattern*, is a unique fingerprint of gate, input transition and technology. It can be accurately calculated using standard device modeling and circuit simulation. The substrate noise signature of the entire circuit is then evaluated using the substrate injection patterns and a precise analysis of the switching activity in the circuit's internal nodes. Switching activities are computed from user-specified input vector sequences. No restriction has been imposed on the selection of input vectors. Hence, the user can simulate a realistic load or perform a worst-/best-case analysis, exploring alternative scenarios of operation.

The methodology, called SUBWAVE [4], consists of the following phases. First, the substrate injection patterns associated with all the gates in the library are accurately simulated using detailed extraction. Second, using event-driven simulation techniques the switching activity of the entire Boolean network is computed, thus producing a transition trace for each node of the network. Third, assuming that technology mapping has been performed, every node corresponds to the output of a gate, hence, the convolution of the nodal trace with the substrate injection pattern of the gate yields its substrate noise contribution. The substrate noise signature is computed as the sum of the contributions of all the nodes in the network. Fourth, the energy spectrum of the substrate noise signature is estimated using high-order autoregressive process modeling. Fig. 1 shows the flow of SUBWAVE.

The advantages of this procedure are threefold. First, the model of substrate noise is compact and can be translated into a single voltage source for any given digital circuit, hence, it can be reused in future redesign cycles. Second, glitch energy is completely captured by the power spectrum and, hence, even high-frequency power spikes can be accurately evaluated for performance degradation analysis. Third, the model computation is inherently fast due to the efficiency of event-driven simulation.

The obtained models are useful in several design and optimization processes. During floorplanning, specific well-isolated areas can be allocated to noisy circuits. Minimum distance requirements can be computed based on the overall spectral energy produced by such circuits and the maximum levels of spurious energy tolerated by sensitive circuits [5]. When space is not available, specific guard rings can be designed to block those frequencies in the spectrum which could interfere with the operation of surrounding circuits. The design of guard rings and other blockage devices can be tuned to work optimally for problematic noise spectra. Rapid characterization of injected noise can be used to test whether redesigned logic blocks are compatible with existing circuitry or if special measures—including further redesign—must be taken. Similarly, spectral characterization of substrate noise could be provided as part of intellectual property interface description, along with the block basic functionality, to reduce the risk of system failure due to unexpected second-order effects.

The efficient generation of substrate noise signature models can be used to drive logic synthesis in circuits which have a limitation in the amount of noise they can produce. A model of the performance degradation due to the effects of noise at specific frequencies can be embedded in the synthesis tool or used to assist a designer. Electromagnetic compatibility requirements for block and systems can be tested at or before actual integration, or *a posteriori* to verify existing problems and causes. Finally, substrate noise signatures can be used as fingerprints for fault analysis and diagnosis.

The paper is organized as follows. In Section II techniques are described for characterizing substrate injection patterns associated with the gates used in the circuit. Section III outlines the evaluation of the switching activity generated by the digital circuit. In Section IV the substrate noise signature is evaluated in time-domain and its power spectrum is estimated using autoregressive process modeling. Finally, in Section V a number of experiments on industrial benchmarks is presented and discussed.

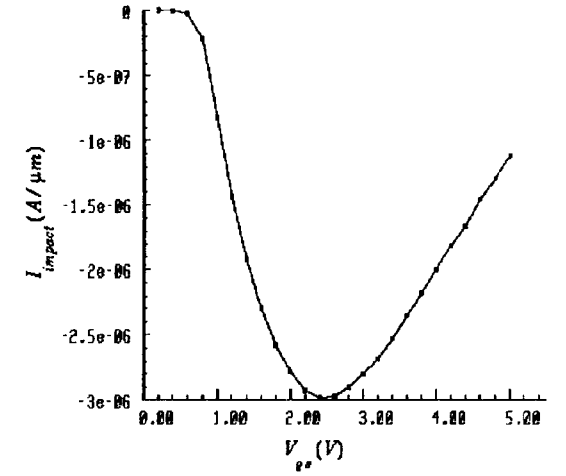
II. SUBSTRATE INJECTION PATTERN EVALUATION

Noise injection is caused at the device level by either impact ionization or drain/source-substrate junctions or both. Reverse-biased n/p junctions form nonlinear capacitances through which noise can easily propagate to substrate. Impact ionization is a highly localized phenomenon. A current is generated in the drain-channel-substrate interface due to the high electric fields present in that region. In submicron technologies, short channels and reduced oxide thicknesses cause electric fields to exceed the electric field's critical value, thus resulting in large electron-hole pair generation.

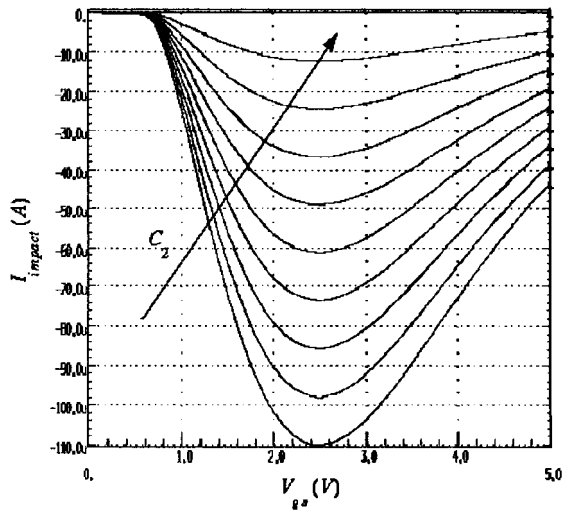
Over 99% of all spurious substrate currents are injected during CMOS gate transitions. Generally, at least one type of MOSFET can be isolated effectively using individual or group wells, thus providing an adequate reduction of substrate coupling. As a consequence, for a particular technology, only one type of MOSFET is responsible for most of the noise injected into the substrate. For this reason, in what follows, only models associated with n-type MOSFET's were considered. Similar considerations can be extended to p-type devices.

A. Impact Ionization

Electron-hole pairs are generated in the pinch-off region, when the electric field exceeds a given threshold. The excess



(a)



(b)

Fig. 2. (a) Impact ionization current density versus $V_{gs}@V_{ds} = 5$ V simulated using PISCES; (b) family of curves obtained from HSPICE simulations while fitting parameter C_2.

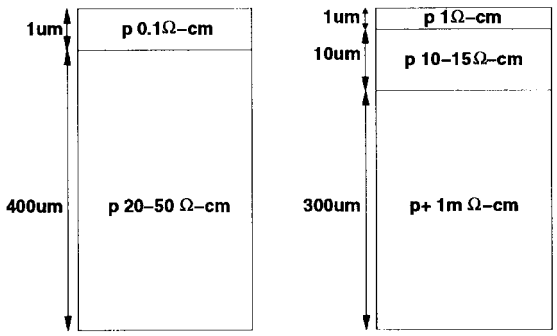
holes are collected in the region of substrate under the device and from there they are transported throughout the chip. The total current produced by impact ionization is evaluated as

$$I_{\text{impact}} = \int_{E_s}^{E_m} I_d A e^{-B/E(x)}\, dx \tag{1}$$

where E_s, E_m, $E(x)$, and I_d are source electric field, maximum electric field, local electric field and drain current, respectively. Constants A and B are material related coefficients. Formulae relating these parameters to measurable quantities and the derivation of (1) can be found in [6]. Since $E_m \gg E_s$, integral (1) can be approximated to

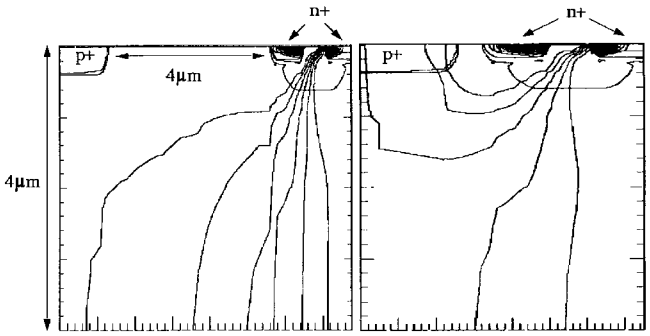
$$\begin{aligned} I_{\text{impact}} &\simeq \frac{A}{B}\, lE_m I_d e^{-B/E_m} \\ &= C_1(V_{ds} - V_{d\text{sat}}) I_d e^{-(C_2/V_{ds} - V_{d\text{sat}})} \end{aligned} \tag{2}$$

where l, V_{ds}, and $V_{d\text{sat}}$ are effective channel length, drain-source voltage and saturation voltage, respectively. C_1 and C_2 are material related coefficients [6].



(a) (b)

Fig. 3. Typical substrate doping profiles: (a) high resistivity; (b) low resistivity.



(a) (b)

Fig. 4. Current flow lines (low-resistivity substrate): (a) with a distant substrate contact; (b) with a near substrate contact.

Equation (2) is used by most MOSFET models to represent impact ionization currents [7]. Simulations of industrial device structures were performed using the 2-D drift-diffusion device simulator PISCES [8]. Using HSPICE and standard curve-fitting techniques, it was possible to fit (2) to the measured data by adjusting parameters C_1 and C_2. Fig. 2(a) depicts the impact ionization current density as it results from a PISCES simulation. Fig. 2(b) shows a family of curves obtained from HSPICE simulations by varying fitting parameter C_2.

B. Noise Source Analysis

There exist two main substrate types: one referred to as *high-resistivity* and the other as *low-resistivity* substrate. Fig. 3 shows examples of such types. In general, the first substrate type is composed of a uniformly doped layer with a resistivity coefficient of 20–50 Ω cm. The second type consists of a thick, high-resistivity epitaxial layer ($d \simeq 10\,\mu$ m, $\rho \simeq$ 10–15 Ω cm) and a low-resistivity bulk ($\rho \simeq 1$ m Ω cm). Low-resistivity substrates have been widely adopted for desirable latch-up suppression properties [6]. In general, it has been found that at low and medium frequencies, typically less than 5 GHz, substrates show a resistive behavior. At higher frequencies, the transport patterns are too complex to be accurately modeled using resistive or resistive-capacitive meshes [1].

In high-resistivity substrates, distance and guard rings are effective attenuation techniques to reduce signal interaction. In low-resistivity substrates, the current tends to flow through low-impedance paths located deep in the chip's lower layers, as shown in Fig. 4(a). As a result, guard rings are generally

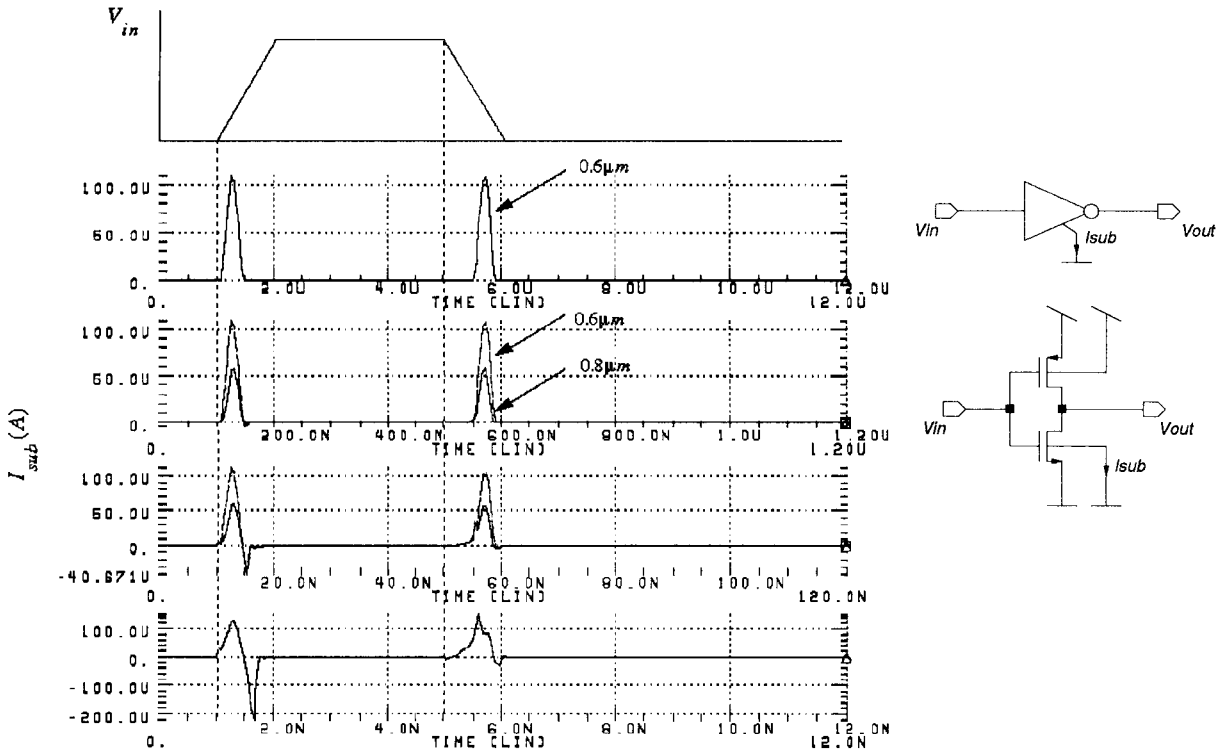


Fig. 5. SPICE simulations of low-to-high and high-to-low transitions in an inverter for different rising and falling times of the input signal for 0.6 μm and 0.8 μm technologies.

ineffective in blocking substrate currents. Fig. 4(b) on the contrary shows a substrate section with a more even current distribution across the top layers. In low-resistivity substrates the use of a very low impedance backplate contact is often preferred. In particular, the inductance of the backplate conductive glue and of the bond wires must be controlled very accurately.

To analyze the amount of substrate current generated by switching gates, consider a simple inverter. Impact ionization induced currents are always positive since they are generated by hole injection. As a result, during both transitions (high-to-low and low-to-high) a positive pulse of current is generated in the substrate (see Fig. 5). The power spectrum associated with this type of signals accounts for low-frequency and DC components. Injection due to capacitive coupling provides positive and negative contributions, thus resulting in a reciprocal cancellation, on average, when the two transitions are equally probable. The injection waveform due to impact ionization has the following features:

- time t_0 corresponding to its maximum depends on the rising and falling time of the input signal;
- duration in time Δt of the pulse depends on the rising and falling time of the input signal;
- the maximum shown from DC simulations (see Fig. 2), which corresponds to the value obtained by (2), is an upper bound for the maximum during a transient.

These characteristics are shown in Fig. 5 where different rising and falling times for the input signal have been chosen. The positive pulse due to impact ionization has approximately the same shape in all cases and only its duration is different (see different time scales). Capacitive coupling due to drain-substrate and source-substrate reverse-biased junctions begins to dominate in rising/falling times of less than 10 ns.

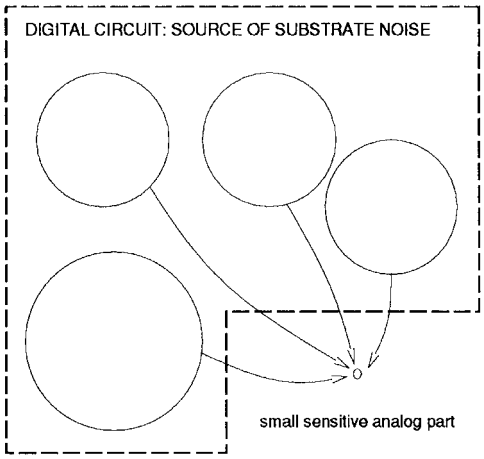


Fig. 6. Distribution of analog and digital sections throughout a mixed-signal chip.

C. Noise Source Partitioning

The gate count of realistic digital circuits is typically in the millions. A complete extraction and detailed simulation of each individual substrate noise injector is impractical. If the substrate underlying the circuit can be approximated to be equipotential, then simultaneous substrate currents injected in different locations contribute in a cumulative fashion to the spurious potential sensed remotely.

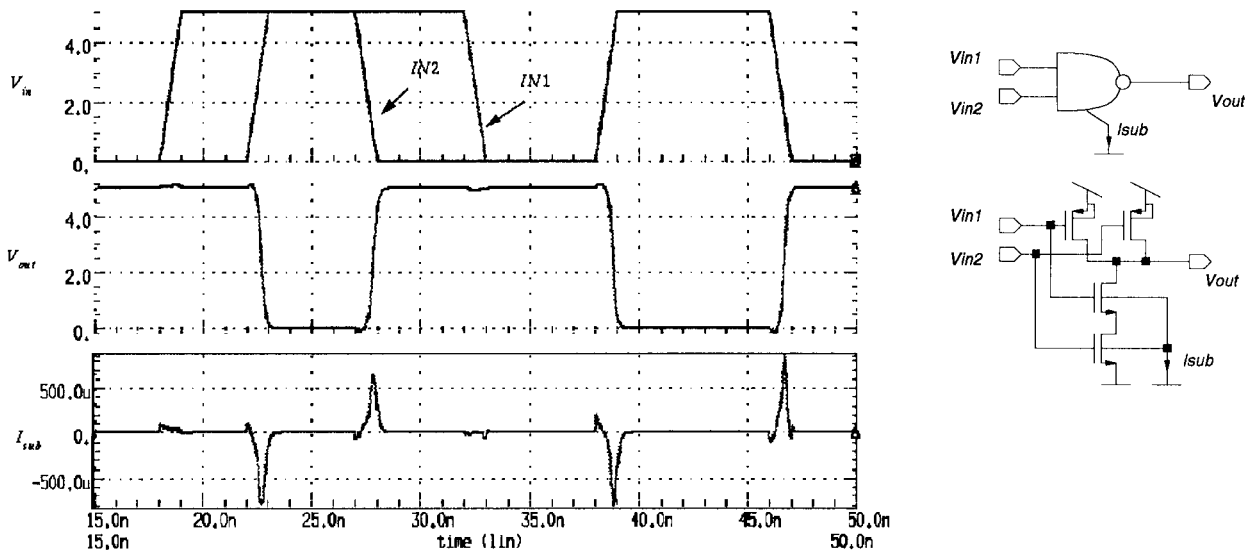


Fig. 7. Substrate injection patterns of the nand gate for all possible input transitions.

Definition 1: Let c be a set of all the gates producing an identical substrate injection pattern for a given input transition scheme. Call c *injection class* and let w_c be the substrate injection pattern of c. Moreover, let $\mathcal{C}$ be the collection of all such classes.

Not only do some of the injection patterns depend on the input transition scheme but also on the load configuration. This problem can be overcome by parametrizing each injection pattern with respect to the load value.

Definition 2: Let p_s be the physical location of the critical analog block where the substrate noise signature of the digital block is sensed.

In [1] it was observed that, in the case of low-resistivity substrates, the relative location of different sections of the circuit is not critical within certain spatial limits. Hence, the following assumption seems reasonable.

Assumption 1: Let p_s be far enough from the logic block. Then, the propagation time of substrate injection patterns between each source and the sensing point can be assumed to be approximately the same.

As a direct consequence of Assumption 1, a logic circuit can be thought as being a collection of clusters of nodes of the Boolean network sharing the same injection characteristics, namely identical injection class and simultaneous injection event. Due to this fact, a partitioning based on time rather than space is conceivable. If complete spatial independence cannot be assumed (as in high-resistivity substrates), a partitioning based on both space and time is needed. In this case, injection classes will encompass exclusively those simultaneously switching gates which are placed within well-defined boundaries. The size and location of such boundaries are the result of tradeoffs between computation time and accuracy. In the remainder of this paper, we will suppose that Assumption 1 is satisfied.

Assumption 2: Let us model a cluster of simultaneously switching gates associated with injection class c as a single *equivalent noise source* applied through a single contact of appropriate dimensions.

In most synchronous circuits, a large portion of signals switch at or near clock edges. The remaining signals, switching at later times in combinatorial blocks, are spread throughout the clock cycle. In this way the injection activity of the chip can be represented by a few hundred equivalent noise sources which can be easily simulated with SPICE.

As an illustration, assume that a circuit consists of N_G types of gates. If each gate produces N_I substrate injection patterns, depending on its input, then, under Assumption 2, the total number of classes for that circuit will be $|\mathcal{C}| = N_G \times N_I$. Suppose now that all the gates associated with a given class are clustered in a specific chip location (bubble in Fig. 6). Then, only $|\mathcal{C}|$ equivalent noise sources need be used to properly model the circuit's substrate injection signature.

To clarify how substrate injection patterns are derived, let us consider the nand gate shown in Fig. 7. The substrate terminals of the nand's NMOS transistors are connected to ground as shown in Fig. 7. The total substrate current I_{sub} is measured on that net. The resulting waveform is shown in the figure for raising, falling and mixed inputs. In this case, only two classes are necessary to capture the behavior of each gate with respect to substrate injection, as only two types of pulses are injected for several input transitions. In fact, it can be shown that injections only depend on output transitions (rise and fall).

III. COMPUTING SWITCHING ACTIVITIES

Several tools have been proposed for fast evaluation of switching activity (see [9] for a review). Event-driven simulation was chosen for its efficiency and its ability of detecting switching signals originated from logic glitches. The core algorithm of the simulator is described in Fig. 8, the notation is similar to that used in [10].

Let us assume that the Boolean network associated with the logic circuit is mapped onto a specific technology, call `Mapped_Network` the resulting network. Let us define sequence `Input_Vectors`, as the input to the simulator. The simulator produces table `Gate_Table` which associates each

```
SIMULATE( Mapped_Network, Input_Vectors )
 foreach vector k ∈ Input_Vectors
  foreach node p ∈ PI
   SetNode(p, t_0^k, input_value(p))
   foreach node n ∈ FO(p)
    Mark(n, (t_0^k + wire_delay(p, n)))
  foreach instant t ∈ T^k
   foreach node n ∈ (DFO − PI)
    if IsMarked(n, t)
     SimulateNode(n, t)
     if (Event(n, t) = Rise or Event(n, t) = Fall)
      g_n ← gate_of(n)
      UpdateGateTable(g_n, Event(n, t), t)
      foreach fanout n' ∈ FO(n)
       Mark(n', t + wire_delay(n, n'))
     else
      SetNode(n, t, value at time(t − 1))
    else
     SetNode(n, t, value_at_time(t − 1))
 return Gate_Table
```

Fig. 8. Event-driven logic simulator.

injection class with the corresponding switching activity. Let $\mathcal{N}$ be the set of nodes of the Boolean network, f_n the logic function associated with node n, and y_n the logic variable associated with the output of n. Let $\mathcal{PI}$ denote the primary input set, $\mathcal{FO}(n)$ the set of fan-out nodes of n, and $\mathcal{TFI}(n)$ the set of transitive fan-in of n, i.e., $\mathcal{TFI}(n) = \{n' \in \mathcal{N} | \exists$ *directed path from* n' *to* $n\}$. Finally, let the ordered set $\mathcal{DFO}$ contain the elements of $\mathcal{N}$ ordered in such a way that every node appears somewhere after all of its transitive fan-in nodes (*Depth-First Order from the outputs*).

For simplicity, but without loss of generality, assume that the `Mapped_Network` consists only of one- or two-input logic gates. Furthermore, assume that for each gate g_n, only two injection classes, one for the rise and one for the fall transition are necessary to fully describe the injection patterns. SIMULATE performs internally an event-driven gate level simulation based on a *pure bounded wire delay model* [11]. All the input vectors are processed sequentially as if they were read from an external register file controlled by a clock having a clock period longer than the longest delay in the circuit (*critical path*).

Suppose node m changes its output at t, each fan-out n of m is marked by **Mark**(n, t'), with $t' =$ *wire_delay*$(m, n))$, so that when SIMULATE processes the instant t', **IsMarked**(n, t') returns a true value and function f_n is evaluated by **SimulateNode**(n, t'). **SimulateNode**(n, t') computes the output for node n at instant t', detecting if an event, i.e., a logic transition, has occurred, while **Event**(n, t') returns its type. When an event is detected for a node n at the instant t', SIMULATE calls **UpdateGateTable** to update the data associated to g_n within the Table `Gate_Table`. At the end of the simulation `Gate_Table` contains an entry for each injection class with a complete trace of its rate of occurrence. Furthermore SIMULATE allows the user to scale its time granularity by a parameter α, such that $K' = \alpha \cdot K$, where K is the default minimum time step of the simulator. This feature is necessary to synchronize the output to the time steps of a SPICE simulation.

In an event-driven simulation each change in the output of a gate g in the circuit produces as many events as the *fanout* of g. Each event e refers to a fanout gate g_e and it is stored in the *Event Queue* together with the time t_e at which it is scheduled. At time t_e the event e is processed forcing a logic evaluation of g_e. If g_e changes, new events are generated and inserted in the queue. All the scheduled events are processed in sequence until the queue becomes empty, then a new input vector is read. The time t_e associated with an event is determined by the delay model which is adopted. The pure bounded wire delay model implies that there is exactly one delay element per gate input. Hence, since a gate has generally more than one input, we may have more than one evaluation per gate during a clock cycle: in general, the gate output may oscillate between the two logic values before settling to the correct value for the current cycle. This model allows us to consider implicitly and with good accuracy every possible spurious transition (*glitch*) in the circuit. The level of accuracy is bounded by the precision of the characterization of the cell library and, for example, can be improved by accounting for interconnect delay.

For each vector in the input sequence SIMULATE determines the output vector together with the traces of values associated to the output of each gate g_n. Such data are obtained by evaluating the Boolean equation f_n for each node n after considering the delay elements associated with the particular instance of g_n. Two injection classes are associated to a particular gate and the transition activities of all the nodes being mapped with each gate are cumulated to produce the trace for the corresponding injection class. This operation is automatically performed by SIMULATE and results in the loss of the relative location of the transition occurrence. This fact is of no concern under the assumptions of Section II.

SIMULATE introduces a discretization of time based on the delay information available from the gate library and the network topology. Let $t_0 = 0$ be the instant at which the simulation begins, $t_{\max}$ the delay of the longest path in the network, and K the number of input vectors. Then, the simulation interval is $\mathcal{T} = [t_0, t_{\text{end}}]$, with $t_{\text{end}} = K \times t_{\max}$. Moreover, while processing the kth vector, only subinterval $\mathcal{T}^k = [t_0^k, t_{\text{end}}^k]$, where $t_0^k = (k-1) \times t_{\max}$ and $t_{\text{end}}^k = k \times t_{\max}$, has to be considered.

Fig. 9 illustrates the trace computation with an example consisting of two injection classes. The corresponding Boolean network has four nodes: three are instances of the AND-gate and one of the OR-gate. For simplicity, we assume a unit wire delay model and we observe the following transition at the primary inputs: $\{x_1, x_2, x_3, x_4, x_5\} = \{0,1,1,1,0\} \rightarrow \{1,1,0,0,1\}$.

It can be proven that, as the example shows, the delay of the critical path of the circuit is 4 time units, therefore all the nodes at the output of the gates settle at time t_4. The traces tr_{and} and tr_{or} are evaluated by cumulating the transition activity at the output of nodes y_1, y_2, y_4, for the AND gate, and y_3, for the OR gate. Notice that at time t_2 in trace tr_{and} a double falling transition occurs. This is due to the fact that the waveforms of both y_2 and y_4 have a falling edge. On the contrary, at time t_1 a single raising transition occurs, since y_2 and y_4 have a raising edge while y_1 has a falling edge.

`Gate_Table` contains a complete trace of the switching activity of each injection class c within the digital circuit, for the overall simulation time interval $\mathcal{T}$. This trace is denominated as $tr_c(t)$. Each injection class is also associated

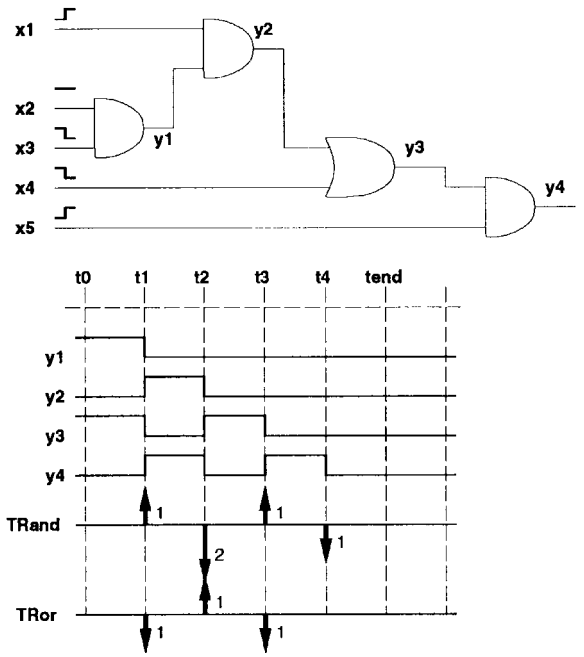


Fig. 9. Example of trace computation.

with a unique substrate injection pattern $w_c(t)$ stored in `Wave_Table`. Both signals are used for the computation of the substrate noise signature of the logic.

IV. Noise Spectral Estimation

To make our observation of the injected noise more realistic at the macroscopic level, we may select one or more sequences derived from either digital simulations or measurements coming from a digital analyzer, thus allowing the user to consider scenarios of interest for a particular application.

Note that in general the digital input bit sequences are not known *a priori*, in fact they can be video or audio data streams, for this reason they are normally modeled as random processes, called *input processes*. The digital input bit sequences, given by simulations or measurements, are realizations of the input processes and are used to compute the digital activity present at every node of the circuit in a well-defined time window and with a predefined resolution, as outlined in Section III.

It is reasonable to assume that input processes be ergodic. It is, however, not required—hence not assumed—that the processes be wide-sense stationary, cyclo-stationary, nor quasi-stationary [12], [13]. Assuming that the digital circuit be fully specified and deterministic, from the statistical characteristics of the input processes one could analytically compute the statistics of the substrate noise. However this is obviously a tedious process, in fact, due to the complexity of the circuit, the final expression could be unmanageable. Moreover, any simplification should not exploit the statistical properties of the input process.

A valid alternative is the conversion of the substrate noise signature in the frequency domain. Consider injection class $c \in \mathcal{C}$. Using the manipulations of Section II one obtains the substrate injection pattern $w_c(t)$ associated with c. By Assumption 2, the injected noise $i_c(t)$ associated with c is computed as

$$i_c(t) = \sum_{r=t_o}^{t_{\text{end}}} \text{tr}_c(t-\tau) w_c(\tau)$$

where $tr_c(t)$ is the trace of the cumulative switching activity of all the nodes in the network associated with c. Injected noise $i_c(t)$ can be viewed as the response of a linear shift-invariant system with impulse response $w_c(t)$. The trace can be interpreted as its input. Due to the time-invariance of the circuit, note that the statement

$$i_c(t)\text{: ergodic random process} \quad \forall c$$

is necessarily true. Summing all $i_c(t)$ over set $\mathcal{C}$ one obtains substrate noise signature $i_{\text{sub}}(t)$ as

$$i_{\text{sub}}(t) = \sum_{\forall c \in \mathcal{C}} i_c(t).$$

The frequency domain estimation of the substrate noise signature $I_{\text{sub}}(\omega)$ can be computed using a variety of techniques.

In order to capture the characteristics of substrate noise, spectral estimation methods based on direct application of the discrete Fourier transform (DFT) should be avoided, while non parametric methods, like periodograms, do not characterize the noise spectrum in a compact form. For these reasons, we decided to adopt a parametric method. In this case, a model for $i_{\text{sub}}(t)$ is given and, based on the observed sequence, identified. Suppose the input processes are stationary and zero-mean (the extension to the quasi-stationary case is straightforward), since the circuit is time invariant we can apply Wold's theorem [14]. The substrate noise signature can be represented as the output of a stable, causal, shift-invariant linear filter with a white noise input. For computational efficiency, we adopted an M-pole transfer function for the linear filter, giving an Mth-order autoregressive (AR) model of the noise process. The AR model of the filter is

$$\text{ARN}(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_M z^{-M}}$$

where $a_i, i = 1, \cdots, M$ are the filter coefficients. Hence, the spectrum of the output zero-mean noise sequence takes the following form

$$S(\omega) = \frac{\sigma_\omega^2}{|1 - a_1 e^{-j\omega} - \cdots - a_M e^{-jM\omega}|^2}$$

where σ_w^2 is the variance of the white noise. The estimation process consists of selecting the correct order of the AR model (i.e., M) and estimating the filter coefficients a_i and the power of the white noise σ_w^2.

The AR model condenses the characteristic of the injection patterns $w_c(t)$ and the statistic of the input process $tr_c(t)$ for all injection classes. Since the former is modeled as a response of a linear shift-invariant system, if the input process is well modeled by an AR process (i.e., audio data stream), this approach to the noise estimation can be very accurate.

Unlike periodogram based methods, the AR spectral estimation is *consistent*, i.e., an increase in the number of data samples results in the decrease of the variance of the estimator

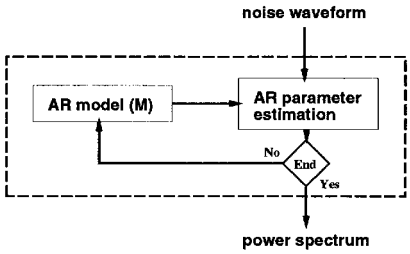


Fig. 10. Noise spectral estimation scheme.

at any given frequency. Hence, in our case the accuracy of the noise estimation monotonically grows with the length of the simulation. Furthermore, the estimator does not suffer from resolution limitations since no signal windowing is used, and long simulations give very good noise spectral estimation. The Yule-Walker method [15] has been used for estimating the parameters a_i in the AR model. This method computes the AR parameters from the autocorrelation matrix using the Levinson–Durbin recursion [16]. The selection of M, i.e., the order of the model, has been done using subjective judgement since the standard criteria proposed did not give satisfactory results, as shown in [16]. Nevertheless, this problem is not crucial, since it is normally driven by the noise information that the designer wishes to consider. We view this process as one based on successive refinements. At first, few parameters are computed (low M) in order to have a rough noise spectral estimation, then, based on the incremental details of the noise spectrum, additional coefficients (high M) are estimated from the same simulation data. The method is illustrated in Fig. 10.

V. Results

The methodology proposed in this paper is supported by several tools which implement the various functions described in Fig. 1. The tools are implemented in C/C++ running under the UNIX operating system. Fig. 11 shows the flow adopted to test the methodology. A Boolean network was mapped onto the selected technology using the logic synthesis tool SIS [17]. SUBWAVE was then run on the mapped network using the substrate injection patterns stored in `Wave_Table` to obtain the substrate noise signature as outlined in Sections III and IV. `Wave_Table` had been previously computed for the given technology. A single sensing node n was established at a reasonable distance from the logic circuit, to ensure that Assumption 2 hold. The substrate noise signature was evaluated in terms of potential $V_{\rm tot}$ sensed in p_s. Equivalent noise sources $V_1, V_2, \cdots, V_N$, computed using the techniques of Section III, were connected to the sensing node via a lumped resistive model extracted by SUBRES [1], a substrate extraction tool based on the Boundary Element Method. Fig. 12 shows the approach. In our experiments we assumed a backplate contact, although this assumption is not necessary in SUBWAVE.

In order to verify the accuracy of the resulting substrate noise signature, we proceeded as follows. First, a layout was generated in the OCTTOOLS environment from the mapped network using TIMBERWOLF [18]. Then, the entire layout contact resistance matrix was extracted using ESTPAR and SUBRES.

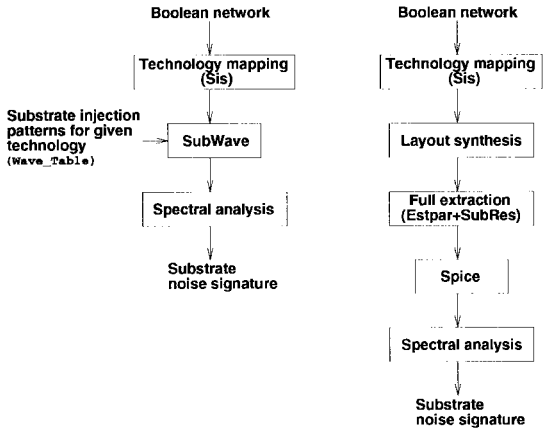


Fig. 11. Comparing substrate noise signatures obtained using SUBWAVE and full extraction and simulation.

TABLE I
RESULTS ON MCN91 BENCHMARKS

Mapped_Network				SUBWAVE	SUBRES	SPICE
Circuit	function	I/O	Gates	*sec*	*sec*	
misex1	pla	8/7	70	75	257	395
z4ml	2b add	5/16	75	54	69	509
mux	mux	21/1	115	75	129	9139
my_adder	16b add	33/17	242	134	3383	-
alu2_cl	alu	10/6	506	430	10105	-
C1355	ecc	41/32	1215	202	-	-
C6288	16b mlt	32/32	2731	1426	-	-

Finally, when possible,[1] the resulting contact resistance matrix was simulated by SPICE using current injectors obtained from the models of Section II. Noise spectral analysis was carried out using the PTOLEMY [19] simulation environment. Then, estimated noise power spectra were computed for different orders M from SPICE (when available) and SUBWAVE data. Values between 16 and 512 have been used for M with the benchmark circuits, while all the presented results have been carried out with a value for M of 32.

Several circuits from the MCNC91 benchmark suite were tested using one thousand randomly generated input vectors. It was assumed that the benchmarks would be implemented in a low-resistivity substrate technology of the type depicted in Fig. 3(b). In Table I the CPU times for the generation of the models and for the verification step are reported for a DEC AlphaServer 2100 5/250. Typically, substrate noise signatures were computed two orders of magnitude faster when SUBWAVE, rather than full simulation, was used. A direct comparison of the waveforms obtained with the two approaches shows a contained error estimated to be less than 10% for all the benchmarks. The largest circuits, such as the 16-b multiplier, could not be simulated using SPICE due to their extreme complexity. Fig. 13 shows the spectrum generated for `misex1` using SUBWAVE and full extraction/SPICE. One observes that most of the spectrum is centered around 1 GHz. An identical behavior was observed in other benchmarks, such as `C6288`, which had been mapped onto the same library

[1] Matrices corresponding to more than 1000 contacts could be extracted by SUBRES but the underlying network could not be simulated in a reasonable time.

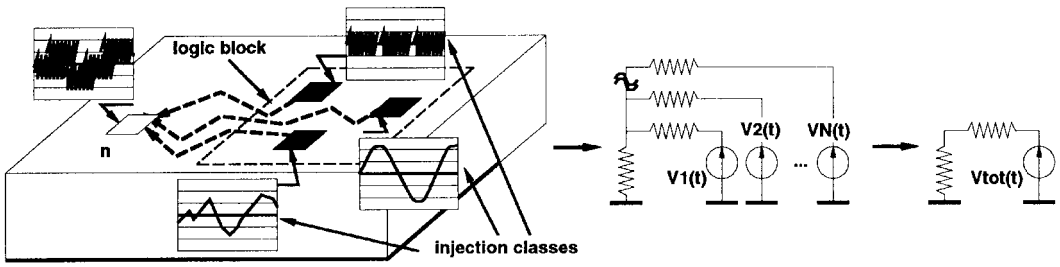


Fig. 12. Observation point and substrate noise sources on logic block.

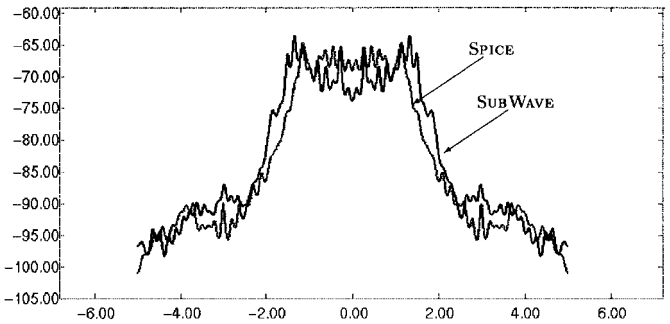


Fig. 13. Normalized substrate noise spectrum of `misex1` in decibels versus gigahertz.

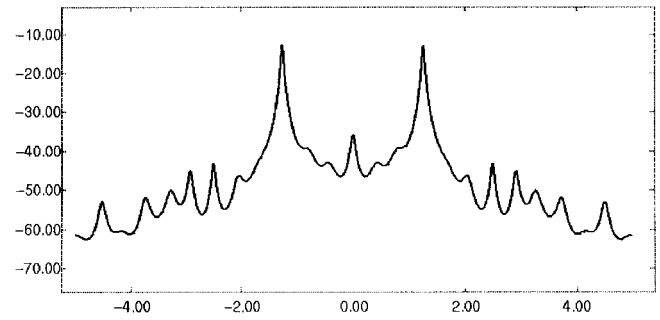


Fig. 14. Normalized substrate noise spectrum of `C6288` in decibels versus gigahertz.

based on 1-ns delay gates (see Figs. 13 and 14). This behavior suggests that the substrate noise spectrum is mainly dependent on technology rather than other design parameters (clock, topology). Benchmark `C6288`, shown in Fig. 14, consists of 2731 gates. In this case the SPICE simulation result could not be obtained, hence, a direct comparison could not be drawn. However, the example shows that similar complexities can be addressed efficiently by SUBWAVE.

VI. CONCLUSIONS

A comprehensive methodology has been presented for the automated generation of compact models for spurious substrate noise currents in realistically complex logic blocks. Every gate in the library is characterized based on its input-dependent substrate injection patterns. The switching activity of the circuit is efficiently computed using event-driven simulation, which produces a transition trace for each node in the circuit. Such trace is convolved with the appropriate gate substrate injection patterns, thus allowing to compute the substrate noise signature for the entire circuit. Standard benchmarks have been used to test the assumptions and the resulting accuracy of the computed models.

ACKNOWLEDGMENT

The authors would like to thank R. Gharpurey and A. Saldanha for their support and useful discussions.

REFERENCES

[1] R. Gharpurey and R. G. Meyer, "Modeling and analysis of substrate coupling in IC's," in *Proc. IEEE Custom-Integrated Circuit Conf.*, May 1995, pp. 125–128.
[2] B. R. Stanisic, N. K. Verghese, D. J. Allstot, R. A. Rutenbar, and L. R. Carley, "Addressing substrate coupling in mixed-mode ICs: Simulation and power distribution synthesis," *IEEE J. Solid-State Circuits*, vol. 29, pp. 226–237, Mar. 1994.
[3] S. Mitra, R. A. Rutenbar, L. R. Carley, and D. J. Allstot, "Substrate-aware mixed-signal macro-cell placement in WRIGHT," in *Proc. IEEE Custom-Integrated Circuit Conf.*, May 1994, pp. 529–532.
[4] P. Miliozzi, L. Carloni, E. Charbon, and A. L. Sangiovanni-Vincentelli, "SUBWAVE: A methodology for modeling digital substrate noise injection in mixed-signal IC's," in *Proc. IEEE Custom-Integrated Circuit Conf.*, May 1996, pp. 385–388.
[5] P. Miliozzi, I. Vassiliou, E. Charbon, E. Malavasi, and A. L. Sangiovanni-Vincentelli, "Use of sensitivities and generalized substrate models in mixed-signal IC design," in *Proc. IEEE/ACM Design Automation Conf.*, June 1996, pp. 227–232.
[6] C. Hu, *VLSI Electronics: Microstructure Science*, vol. 18. New York: Academic, 1981.
[7] Y. Cheng, M. Chan, K. Hui, M. Jeng, Z. Liu, J. Huang, K. Chen, J. Chen, R. Tu, P. K. Ko, and C. Hu, *BSIM3v3 Manual*, Univ. California at Berkeley, Berkeley, CA, 1996.
[8] M. R. Pinto, C. S. Rafferty, H. R. Yeager, and R. W. Dutton, *PISCES-IIB*, Stanford University, Stanford, CA, 1985.
[9] F. Najm, "Power estimation techniques for integrated circuits," in *Proc. IEEE Int. Conf. on Computer Aided Design*, Nov. 1995, pp. 492–499.
[10] R. K. Brayton, G. D. Hachtel, and A. L. Sangiovanni-Vincentelli, "Multilevel logic synthesis," *Proc. IEEE*, vol. 78, pp. 264–300, Feb. 1990.
[11] L. Lavagno and A. Sangiovanni-Vincentelli, *Algorithms for Synthesis and Testing of Asynchronous Circuits*. Amsterdam, The Netherlands: Kluwer, 1993.
[12] C. W. Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry, and the Natural Sciences*, Springer Series in Synergetics, vol. 13. Berlin, Germany: Springer-Verlag, 1983.
[13] W. A. Gardner, *Introduction to Random Processes with Applications to Signals & Systems*, 2nd ed. New York: McGraw Hill, 1990.
[14] H. O. A. Wold, *A Study in the Analysis of Stationary Time-Series*. <Publisher's location: Almquist and Wiksell, 1938.
[15] S. M. Kay, *Modern Spectral Estimation: Theory and Application*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
[16] P. M. Clarkson, *Optimal and Adaptive Signal Processing*. Boca Raton, FL: CRC, 1993.
[17] E. M. Sentovich, K. J. Singh, C. Moon, H. Savoj, R. K. Brayton, and A. L. Sangiovanni-Vincentelli, "Sequential circuit design using synthesis and optimization," in *Proc. IEEE Int. Conf. on Computer Design*, Oct. 1992, pp. 328–333.
[18] C. Sechen, *VLSI Placement and Routing Using Simulated Annealing*. Boston, MA: Kluwer, 1988.

[19] J. Buck, S. Ha, E. A. Lee, and D. G. Messerschmitt, "Ptolemy: A framework for simulating and prototyping heterogeneous systems," *Int. J. Comput. Simulation*, vol. 4, pp. 155–182, Apr. 1994.

Edoardo Charbon (S'90–M'92), for a photograph and biography, see p. 190 of the February 1999 issue of this TRANSACTIONS.

Paolo Miliozzi received the Laurea degree in electrical engineering, *summa cum laude*, from the University of Bologna, Italy, in 1992 and the Ph.D. degree in electrical engineering and telecommunications from the University of Padova, Italy, in 1996.

Between 1994 and 1996 he visited the Department of Electrical Engineering and Computer Science of the University of California at Berkeley, where he was involved in substrate noise characterization within top-down constraint-driven design methodologies of mixed-signal IC's. In 1996 he joined Rockwell Semiconductor Systems, Newport Beach, CA. His research interests include development of CAD tools and methodologies for radio-frequency, analog, and mixed-signal IC's; parasitic analysis and substrate noise characterization for automatic layout generation; device modeling for circuit simulation.

Luca P. Carloni (M'95–SM'95) received the Laurea degree in electrical engineering, *summa cum laude*, from the University of Bologna, Italy, in July 1995 and the M.S. degree in electrical engineering and computer sciences from the University of California, Berkeley, in December 1997. He is currently a Ph.D. degree candidate in the Department of Electrical Engineering and Computer Sciences of the University of California, Berkeley.

His research interests are in the area of computer-aided design of electronic systems and include logic synthesis, high-level synthesis, and combinatorial optimization.

Alberto Ferrari received the Dr.Eng. degree *summa cum laude* and the Ph.D. degree in electrical engineering at the University of Bologna, Italy, focusing on low-power low-voltage digital signal processing (DSP) architecture.

He visited the Electrical Engineering Department of University of California at Berkeley, where he worked on low-power digital design for DSP's, and the SGS-Thomson Berkeley Lab, working on subvolt low-power speech-recognition systems. He recently joined the PARADES, Rome, Italy, working on design of embedded systems.

Alberto Sangiovanni-Vincentelli (M'74–SM81–F'83), for a photograph and biography, see p. 190 of the February 1999 issue of this TRANSACTIONS.