Defect-based Fault Simulation Model for i_{DDT} Testing

Abhishek Singh, Jim Plusquellic, Dhananjay Phatak and Chintan Patel
Department of CSEE, University of Maryland, Baltimore County

Paper No:

Contact Author:

Abhishek Singh

University of Maryland Baltimore County

ITE 371, CSEE Dept.,

1000 Hilltop Circle,

Baltimore, MD 21250.

Phone: 410-455-6771.

E-mail: abhishek@cs.umbc.edu.

Fax: 410-455-3969.

Defect Simulation Model for i_{DDT} Testing

Authors:
Abhishek Singh, abhishek@cs.umbc.edu
Jim Plusquellic, plusquel@cs.umbc.edu
Chintan Patel, cpatel2@cs.umbc.edu

Address:
Department of CSEE, University of Maryland, Baltimore County
1000 Hilltop Circle, Baltimore, MD - 21250.

Abstract

The International Technology Roadmap for Semiconductors (ITRS) identifies two main challenges associated with the testing of manufactured ICs. First, the increase in complexity of semiconductor manufacturing process, physical properties of new materials, and the constraints imposed by resolution of lithography techniques etc., give rise to more complex failure mechanisms and hard-to-model defects that can no longer be abstracted using traditional fault models. Majority of defects, in today's technology, include resistive bridging and open defects with diverse electrical characteristics. Consequently, conventional fault models, and tools based on these models are becoming inadequate in addressing defects resulting from new failure mechanisms. Second, the defect detection resolution of main-stream I_{DDQ} testing is challenged by significant elevation in off-state quiescent current and process variability in newer technologies. Overcoming these challenges demands innovative test solutions that are based on realistic fault models capable of targeting real defects and thus, providing high defect coverage. In prior works power supply transient current or i_{DDT} testing has been shown to detect resistive bridging and open defects. The ability of transient currents to detect resistive opens and their insensitivity (virtually) to increase in static leakage current make i_{DDT} testing all the more attractive over I_{DDQ} testing. However, in order to integrate i_{DDT} based methods into production test flows, it is necessary to develop a fault simulation strategy to assess the defect detection capability of test patterns and facilitate the ATPG process. The analog nature of the test observable, i.e. i_{DDT} signals, entail compute intensive transient simulations that are prohibitive. In this work, we propose a practical fault simulation model that partitions the task of simulating the DUT (device under test) into linear and non-linear components, comprising of *power/ground-grid and core-logic respectively. Using divide-and-conquer strategy, this model replaces the transient simulations of power/ground-grid with simple convolution operations utilizing its impulse response characteristics. We propose a path isolation strategy for core-logic as a means of reducing the computational complexity involved in deriving i_{DDT} signals in the non-linear portion. The methodology based on impulse response functions and isolated path simulation, can enable i_{DDT} fault simulation without having to simulate the entire DUT. To our knowledge, no practical technique exists to perform fault simulation for i_{DDT} based methods. The proposed fault simulation model offers two main advantages, first, it allows fault induction at geometric or layout level, thus providing a realistic representation of physical defects, and second, the current/voltage profile of power/ground-grid, derived for i_{DDT} fault simulation, can be used to perform accurate timing verification of logic circuit, thus facilitating design verification. In summary, the proposed fault simulation framework not only enables the assessment of defect detection capabilities of i_{DDT} test methodologies, but also establishes a platform for performing defect-based testing on practical designs.*

1.0 Introduction

The unrelenting pursuit of device scaling brings the MOS technology into nanometric domain quickly surpassing the sub-micron era. Motivated by higher performance and greater density, technology scaling has enabled reduction in transistor and feature size from 1mm in 1970s to 65nm in the present time. On one end, the technology scaling promises lower cost by providing faster and smaller systems, at the other end, it imposes several intricate challenges involved in manufacturing and testing of these systems. The International Technology Roadmap for Semiconductors (ITRS) [1] identifies two main challenges imposed by aggressive trends in technology scaling on manufacturing-imperfection-related testing. These challenges mainly arise from (a) increase in the failure probability of devices manufactured in advanced technologies, and (b) diminishing effectiveness of existing test techniques due to adverse effects of device scaling.

Transition to nanometric technologies unveils diverse failure mechanisms and defect populations attributed to changing manufacturing process technology, physical properties of new materials, circuit sensitivities, and functional characteristics of transistors. For example, smaller or higher-aspect ratio vias are more susceptible to incomplete etch, which may lead to greater prevalence of resistive vias. Similarly, subtractive-aluminum interconnect based technologies were more susceptible to bridge defects, whereas, additive-copper dual damascene processes are more likely to cause opens or resistive/spongy/porous connects. The decrease in gate oxide thickness below 20Å (12Å for 65nm technology) increases the probability of complex gate-oxide related shorts. Introduction of low-κ dielectrics may lead to an increase in possibly latent, resistive bridges. Furthermore, the decrease in transistor size may amplify the effect of degradation mechanisms.

I_{DDQ} testing has been used as the main stream defect and reliability screen for several technology generations. Reduced ATPG complexity, ability to target real defects with few test patterns and sensitivity to weak defects, have been the main advantages of I_{DDQ} testing. However, miniaturization of MOSFET devices, governed by the laws of constant field scaling, imposes significant challenge on its effectiveness. Constant field scaling dictates lower supply voltages for successive technology nodes to prevent the electric field across MOSFET transistor gate oxide from surpassing reliability limits. However, lowering the supply voltage also degrades the circuit performance. To compensate for this degradation in performance, transistor threshold voltage is reduced to acceptable levels. Nevertheless, reduction in transistor threshold voltage impacts transistor off-state or leakage current, which has an exponential dependence on this parameter. Increase in defect-free I_{DDQ} due to elevation in leakage currents significantly affects the defect detection resolution of single-threshold I_{DDQ} testing. Several techniques, based on deriving statistical correlations, have been proposed to extend the use I_{DDQ} for defect-based testing. However, the effectiveness of these techniques is uncertain due to (a) significant increase in defect-free I_{DDQ} level, washing out the effect of defect, and (b) exponential dependence of I_{DDQ} on process parameter variations (V_{th}, V_t, L_{eff}, etc.). In addition, the spread of parameter variation is likely to increase as process control below 90nm becomes more difficult. Recently, the test industry has turned its focus on delay based test methods to supplant the perceived increase in DPM level caused by uncertainty associated with I_{DDQ} testing. However, the effectiveness of existing delay test methods is limited by delay measurement accuracy, which becomes a greater challenge with decreasing clock periods and adverse effects of process variability.

These technology trends suggests the need for new screening mechanisms that are based on realistic defect-based fault models that are capable of targeting real manufacturing imperfections. Defect-based testing (DBT) propounds a detailed understanding of defects' electrical behavior and their classification based on these behavioral properties. The goal is to then, identify test methods that can target these defect populations. The fundamental requirements for implementation of a defect-based test methodology are (a) understanding the occurrence of manufacturing imperfections, and (b) realistic modeling of these imperfections. Understanding the occurrence of defects facilitates apriori enumeration of most likely defects using techniques such as, inductive fault analysis (IFA) and/or statistical methods. Similarly, realistic defect-based fault modeling must preserve those characteristics of defects, that can enable and enhance their detectability. Layout or circuit level design abstractions are the most appealing representations for defect-based fault modeling.

Prior research on i_{DDT} test, such as [2-8], [11-12], shows that changes in the circuit configuration caused by a defect, manifests itself as anomalies in the transient signals measured at the power supply ports. Power supply transient current signals reflect the charge transfer phenomena that take place when a CMOS device switches in response to an input transition. The charge drawn by the MOS transistors as they transition through various states (cut-off, linear, and saturation) reflect their functional and structural integrity. Manufacturing imperfections that cause deviation in the charge transfer characteristics of the circuit, manifest as aberrations in shape characteristics of transient current pulse. In addition, the width of transient current pulse preserves information on propagation delay of the switching circuit. The ability of transient current signals to preserve vital functional and delay information of the circuit may allow the test industry to exploit the combined benefits of I_{DDQ} and delay testing.

The fault simulation framework presented in this work is applicable to any i_{DDT} test method. However, we believe that the method described in [17], called *Transient Signal Analysis (TSA)*, exploits the full potential of i_{DDT} testing using distributed transient current/voltage measurement across individual power supply ports. Therefore, we use TSA as a vehicle for screening defects in the proposed fault simulation flow. TSA employs a two step procedure to identify the outliers. First, a set of unique signatures of a DUT are derived using time, frequency- or phase-spectrum information of i_{DDT} signals measured at various power supply ports. The area under these signature waveforms are then cross-correlated with that of other power supply ports in the DUT. The first step captures the difference between the DUT and a reference device, induced by defect or process variations. The second step helps to screen devices that exhibit lack of correlation between the signatures of two different supply ports. The lack of correlation results from the localized effect of defect-induced anomaly in the i_{DDT} signal. Regression analysis is used to derive prediction bounds that probabilistically separate the defective devices from the defective-free devices with process variations.

The integration of TSA or other i_{DDT} test techniques into existing production test flows requires tools for performing ATPG and fault simulations. Due to their analog nature, the generation of i_{DDT} signals entails transient simulations of the entire DUT. The memory and time requirements of such simulations are prohibitive. In this work, we propose a model that can be used to implement a practical fault simulator for i_{DDT} testing without performing transient simulation on the entire DUT. The main idea behind the proposed model is to decompose the DUT into two systems, such that the DUT can be represented as a cascade of these two systems. Methods are proposed to reduce the simulation complexities of these systems based on their electrical properties. The two systems involve, 1) a linear constituent, namely, the power grid, and 2) a non-linear core logic circuit. A unique method based on impulse response (IR) and convolution is proposed for the linear power grid component. This method allows simulation-less computation of the power grid response to transient inputs produced from the core logic (non-linear) component. Furthermore, a path isolation scheme is proposed to address the simulation complexity of the non-linear component.

2.0 Related Work

Testing methods based on the analysis of power supply transient signals are described in [2-6], [11-12] for digital circuits and in [7-8] for analog circuits. However, we have not uncovered any prior work that proposes defect-based models that can enable fault simulation of i_{DDT} test vectors. In addition to providing a practical methodology for i_{DDT} fault simulation, we believe that this work can be leveraged for power verification and signal integrity analysis. This is true because these tasks share the requirement of transient simulations. The former analyzes these signals to identify anomalies caused by defects whereas, the latter processes dynamic IR and Ldi/dt drop, and package/on-chip resonance.

Due to signal integrity problems caused by aggressively increasing device densities, simulation of transient power distribution in a chip has become an essential step in power verification. To meet this growing requirement several static and transient simulation techniques have been proposed in the past decade. Mathematical tools that speed-up the power grid simulations, such as [15], [18-20] and methods based on Transient Current Simulation of logic circuits [9], [13] and [16] have been proposed. These allow derivation of transient currents drawn by logic circuits without running SPICE level transient simulations.

3.0 System Overview

Any digital CMOS chip can be modeled as a combination of two complementary electrical systems, a linear RC system formed by the power (V_{DD} and GND) grid and a non-linear RC-transistor system formed by the underlying CMOS logic circuit. For example, Figure 1 shows a portion of a typical row based standard-cell design. The upper half of the figure depicts the power grid, laid out as a stack of uniformly spaced metal runners at alternating layers connected using stacked contacts. (For simplicity only a two layer structure is shown) The parasitic resistance and capacitance of the power grid along with the on-chip decoupling capacitors form a linear RC system. This linear system is referred to as *Power Grid Circuit (PGC)*. In reality the physical structure of metal rails comprising a power grid also contributes a significant amount of inductance(L), thereby rendering it a three-dimensional RLC ladder circuit. In this work, we ignore the inductive component of the power grid for simplicity. It must be realized that this does not have any bearing on the proposed concept of system partitioning. This is true because L, like R and C, is a linear component and does not affect the linear nature of the PGC.

Figure 1. Linear and non-linear system representation of a chip.

The lower half of the figure shows the core logic comprising the standard-cells (MOS-transistors) and the signal routing of the chip. Since, MOS-transistors are inherently non-linear devices, this portion forms a non-linear electrical system, referred to as *Core Logic Circuit (CLC)*. The local V_{DD} and GND runners on the standard-cells are connected to the global power grid at various points through special nets/vias called *follow-pins*, as indicated in the figure.

A transition sequence applied at the primary inputs (or outputs of scan latches) causes the gates along a sensitized path to switch in a temporal sequence. Each switching gate draws transient current (i_{DS}) from the power grid, sourced by the external power supply pads or C4s. The PGC transforms these i_{DS} signals into composite current transients (i_{DDT}) that are measured at the C4s. Therefore, the DUT can be modeled as a cascade of two electrical systems wherein, the outputs the CLC feeds the inputs of PGC. The transient current drawn by a CMOS gate can be modeled as a PWL current source connected between the V_{DD} and GND grid. The metal layer at which the DUT is partitioned into the above two systems, determines the sparseness of the input and output ports in the PGC and CLC respectively.

4.0 Fault Simulation Flow for i_{DDT} Testing

The fault simulation procedure proposed for i_{DDT} testing can be partitioned into two separate flows, a preprocessing flow and a post-processing flow. The preprocessing flow is performed only once for a given design (or circuit) while, the post-processing flow is performed for every vector (or fault). These two flows are depicted in figure 2(a) and 2(b) respectively. The subsequent sections of this paper elaborate on each part of the flow diagram.

Figure 2. Flow diagram of the fault simulation procedure for i_{DDT} testing.

4.1 Preprocessing Flow

The preprocessing flow involves three basic steps described below.

- ***Step1: System Partitioning:*** The most important step in the flow is the partitioning of the DUT into a linear (PGC) and a non-linear (CLC) electrical system.
- ***Step2: Power Grid Characterization:*** Power grid characterization involves the derivation of IR functions between each input-output port of the power grid. The IR functions, by definition, completely characterize the power grid and can be used to derive its response to any arbitrary input using convolution.
- ***Step3: Generation of Iso-IR Bands:*** An iso-IR band defines a physical region in the grid layout consisting of input locations that are characterized by a similar IR. Such a categorization helps in significantly reducing the number of convolution operations required to generate the power grid response.

4.2 Post-processing Flow

The post processing flow starts with a vector sequence (pair of test vectors) that can cause a transition along a path in the design. The aim of the fault simulation (post-processing) procedure is to determine the fault coverage of this vector sequence. The steps involved in the flow are as follows

Step1: Fault Injection: A fault is injected between two nodes in the layout of the DUT by introducing resistive connection modeling resistive shorts/bridges or opens. The advantages of using layout information are 1) the fault model closely represents the physical defects and 2) only physically adjacent nets are considered as short/bridge candidates, thereby reducing the number of fault candidates. We believe that this fault injection scheme closely follows the defect based test paradigm. The ability to insert fault at the layout (mask) level makes the proposed technique generic to any (or most) fault model. For example, the fault can be 1) any physical deformity that can be represented at layout level such as shorting/open or bridge between two (or more) nets, 2) variation in electrical characteristic of a net such as increase/decrease in resistance of a net and/or 3) variation in process parameter caused by extreme process variations. Furthermore, the proposed setup allows fault simulation of the test vector under more realistic conditions such as, noise, process variations, cross-talk, leakage, etc. thus giving a more meaningful coverage metric.

Step2: Isolated-Path Transient Simulation: An isolated path corresponds to a sensitized path that is physically separated from the chip layout by breaking connections with the unsensitized logic and the power grid, while, preserving the fanout loads and the signal connections of the gates within the sensitized path. The path (or paths) sensitized by the input vector sequence are identified and physically isolated from the layout. These isolated paths can be further segmented into fault-dependent and fault-independent paths that can be independently simulated. A detailed RC-transistor exaction of the segmented paths is performed. The transient current waveform generated by the sensitized path (paths), at each output port of the CLC, can be derived through SPICE or time-domain Current Waveform Simulations of these isolated paths.

Step3: Iso-IR Band Selection: Based on the physical location of the input current source, the iso-IR band corresponding to that input is identified.

Step4: Convolution: The i_{DDT} or i_{DS} waveforms measured at specific nodes during the isolated path simulations are convolved with the IR functions of the iso-IR band selected in the previous step.

Step5: Linear Superposition: By the virtue of superposition property of linear time invariant (LTI) systems the power grid response to individual current input source can be combined through linear superposition to derive the overall response of the grid at a given C4. This property allows us to simulate all the sensitized paths independently. Another major advantage of this property is to limit the maximum number of convolution operations required to derive the overall response of the PGC. This is detailed further in the paper.

Step6: i_{DDT} analysis for fault detection: The i_{DDT}s thus obtained at the C4s, includes the current drawn by the defect and the defect-free logic. These i_{DDT}s can be used as the input for TSA or as input to a similar technique using the RLC model of the probecard as a means of obtaining the overall response. In either case, the output of this fault simulation procedure is the basis for a) determining the i_{DDT}-test coverage of a given set of test patterns, b) guiding the ATPG algorithm to generate vectors that can enhance the detectability of the defect and c) determining the range of resistance values for which a resistive short, open or bridging defect is detectable.

5.0 System Partitioning

The DUT is decomposed into a linear and non-linear system by breaking the physical connections between the power grid and the core logic cells. The metal layer at which the connections are broken defines the partitioning scheme. For example, Figure 3 shows two possible partitioning schemes. In one case the partitioning is done at the follow-pins whereas, in the second case the partitioning is done at the nodes where MOS transistors in each standard-cell connects to the local V_{DD}/GND rails. The "cross" symbol indicates the nodes at which the connections are broken in each scheme. The same is true for GND grid not shown in the figure. In the first case, referred to as *FP-scheme* (for follow-pins), the linear system (PGC) consist of the global power-grid and the non-linear system (CLC) consists of the standard-cells, including their local V_{DD}/GND rails and the signal routing. In the second case, referred to as the *TR-scheme* (for transistors), the PGC includes the global power grid as well as the local V_{DD}/GND rails in each row of the core logic and the CLC includes the standard-cells (without V_{DD}/GND rails) and the signal routing. The locations (or nodes) at which the systems are partitioned represent the input-ports of the PGC and output-ports of the CLC. For the TR-scheme, when the source nodes of more than one MOS-transistor are connected directly to the local V_{DD}/GND rails, which may be true for complex standard cells, any one node can be considered as an output. This is a reasonable assumption given that the impedance between any two points on the local V_{DD} (or GND) rail in a standard-cell is sufficiently small. The FP-scheme offers fewer number of input-output ports, reducing the complexity involved in the PGC characterization but may increase the complexity of core-logic (non-linear) simulation due to inclusion of local V_{DD}/GND metal rails in the CLC. In contrast, the TR-scheme offers larger number of input-output ports which increases the complexity of PGC characterization but simplifies the core-logic simulations. These trade-offs will be revisited in the subsequent sections.

Figure 3. Partitioning schemes of a chip into linear and non-linear systems.

6.0 Power Grid Circuit (PGC) Characterization

In current technologies the V_{DD}/GND routing can occupy more than 25% of the total routing area on a chip and thus consist of millions of linear elements. SPICE simulations on these grids can be very expensive and therefore, several techniques, such as [15], [18], [19] and [20], have been proposed for fast power grid simulation. Even these tools may prove infeasible as the base simulation engine for an i_{DDT} fault simulator, due to the large number of faults and test vectors. We demonstrate a convolution procedure based on the linearity property of the PGC that enables us to compute the power-grid response to the input switching transients from the core logic without running simulations.

Any linear time invariant (LTI) system can be completely characterized by its Impulse Response (IR) function denoted as *h(t)*. The impulse response *h(t)*, is the output of the system to a unit impulse function, δ*(t)*. Once the IR of a linear system is known, we may construct the response of a the system to an arbitrary input signal as a sum of suitably delayed and scaled impulse responses. This process is called convolution and is mathematically described using Eq. 1. Here, *f(t)* is the input signal, *g(t)* is the output signal and *h(t)* is the IR function. The response of a linear system

$$\begin{aligned} g(t) &= h(t) \otimes f(t) = \int_{-\infty}^{\infty} h(u) f(t-u) du \\ &= \int_{-\infty}^{\infty} h(t-u) f(u) du \end{aligned} \qquad \textbf{(1)}$$

to an arbitrary input signal can thus be computed by convolution using the IR function in time domain.

The PGC represents a multi-input and multi-output linear system. Each input on the grid sees a different RC network to each of the outputs (C4s) therefore, there exists a unique IR function for each input-output pair, denoted as $h_{ij}(t)$, where *i* and *j* represents the input and output port of the PGC respectively. A set of such IR functions, $h_{ij}(t)$, can be used to characterize the PGC. Once the grid is characterized, its response to transient inputs can be determined by convolving the transients with the corresponding IR functions. Superposition and shift-invariance properties of a LTI system are used to determine the outputs due to 1) multiple switching gates within a sensitized path and 2) multiple sensitized paths under the same input sequence. This can eliminate the need to simulate millions of RC elements with a fixed number of multiplication and addition operations during the fault simulation.

7.0 Demonstration of Characterization on a Commercial Power-grid

Figure 4(a) shows a portion of the commercial power grid subsequently referred to as the Quad. The Quad occupies a 10,000 by 10,000 unit area and interfaces to a set of external power supplies through an area array of V_{DD} and GND C4 pads. As indicated in the figure, there are 4 V_{DD} C4s and 6 GND C4s in this portion of the grid. Figure 4(b) shows that the grid is constructed over 4 layers of metal, with metal 1 (M1) and metal 3 (M3) running vertically and M2 and M4 running horizontally. The C4s are connected to wide runners of vertical M5, shown in Figure 4(a), that are in turn connected to the M1-M4 grid. In each layer of metal, the V_{DD} and GND rails alternate. Stacked contacts are placed at the appropriate crossings of the horizontal and vertical rails.

Figure 4. The "Quad": A portion of the commercial power grid used in the simulation experiments.

We derived an RC model of the Quad using an extraction script that preserves the physical structure of the metal interconnect in the topology of the RC network, i.e. no network reduction heuristics are applied. The resistance per square and the overlap capacitances per unit area of TSMC's 0.25µm 5 metal process used in the extraction process were obtained from published parameters by MOSIS [21].

In this experiment we consider the Quad as the PGC and a custom designed 16-bit logarithmic adder as the CLC. We inserted ~3000 labels at the M1-M2 crossovers in the grid layout to represent possible input locations on the PGC.

The impulse response from each of the input ports to the C4s are obtained by first stimulating the grid at each of the input ports with a unit step input. This gives the step response, $s_{ij}(t)$, of the grid with respect to input location *i* and output *j*. The impulse response, $h_{ij}(t)$, can be derived from the step response, $s_{ij}(t)$, using differentiation in time domain. This is again due to the linearity property of the system and can be described using Eq. 2, where $\delta(t)$ and $u(t)$ represent unit impulse and step input functions, respectively and $h(t)$ and $s(t)$ represent the unit impulse and step response of the linear system, respectively.

$$\delta(t) = \frac{d}{dt}u(t) \Rightarrow h(t) = \frac{d}{dt}s(t) \tag{2}$$

Figure 5(a) shows a subset of the possible input locations in the quad. It also indicates the IR functions that exists between a source location, *i*, to each of the four outputs. The step response curves, $s_{ij}(t)$, obtained using SPICE simulations are shown in Figure 5(b) and the IR functions obtained after the differentiation operation on $s_{ij}(t)$ are shown in Figure 5(c).

Figure 5. (a) IRs from a sample current source location, i, to all the C4s. (b) Step response. (c) Impulse Response.

To verify the convolution based procedure, the PGC was stimulated with a triangle current source shown in Figure 6(a). The results obtained using the SPICE simulations are superimposed with the response obtained by convolving the triangle input with the IR function corresponding to that current source location in Figure 6(b). As shown in the figure the two curves are almost identical.

Figure 6. a) Triangle input current to the PGC (b) Superimposed SPICE and Convolution responses.

7.1 Iso-IR Contours

Eq. 3 gives the expression for convolution sum of discrete signals where *g[i]*, *h[i]* and *f[i]* represents the output, IR and input of the linear system respectively. If *f[i]* is a N point signal, *h[i]* is a M point signal then *g[i]* is a N+M-1

$$\begin{aligned} g[i] &= h[i] \otimes f[i] = \sum_{j=0}^{M-1} h[j]f[i-j] \\ &= \sum_{j=0}^{N-1} f[j]h[i-j] \end{aligned} \qquad \textbf{(3)}$$

point signal as given by the above equation. Eq. 3 shows that the convolution based method requires N*M additions and multiplications. Thus, the complexity involved in the computation of grid response is significantly reduced as compared to a SPICE simulation based approach that involves solving several partial differential equations.

Comparison of IR function curves from adjacent input locations suggests that their amplitude and shape characteristics vary slowly as a function of distance. Thus IR functions from adjacent input locations, within a user defined threshold, can be grouped into regions or bands. The threshold is chosen based on tolerable difference in the output waveforms obtained using Eq. 3. However, selection of a suitable threshold requires a means of quantifying the similarity between two waveforms in terms of their shape characteristics. This similarity analysis is performed using cross-correlation and auto-correlation operations.

Cross-correlation of two waveforms results into a third waveform, the amplitude of which indicates the degree of similarity between the two waveforms. Also, the location of its peak indicates the time-shift required in the second signal to obtain the maximum match with the first waveform. This is mathematically expressed by Eq. 4, where $r_{xy}[i]$ represents cross-correlation of two waveforms *x[i]* and *y[i]*. When *x[i]* and *y[i]* are identical, the operation is termed

$$\begin{aligned} r_{xy}[i] &= x[i] \oplus y[i] = \sum_{j=0}^{M-1} x[j]y[j-i] \\ a_x[i] &= x[i] \oplus x[i] = \sum_{j=0}^{M-1} x[j]x[j-i] \end{aligned} \qquad \textbf{(4)}$$

as auto-correlation, $a_x(t)$. The peak value of an auto-correlation function provides us the maximum expected value for the degree of similarity in the IR functions.

First, the input locations are sorted in an ascending order of their euclidian distance with respect to a reference location. The creation of a new iso-IR band begins with the selection of an representative location, referred to as *focus* of the iso-IR band, which is the first input location in the distance-sorted list and is not a part of any previously identified iso-IR band. The IR function of the focus is auto-correlated to obtain the maximum expected degree of similarity, max($a_{fi}(t)$). The algorithm then searches the entire grid space to find all input locations that have a peak cross-correlation value not exceeding 5% (selected difference tolerance) of the peak auto-correlation value of the focus. This condition is given by Eq. 5.

$$max\left\{r_{f_i h_j}(t)\right\} \le 0.05 \cdot max\left\{a_{f_i}(t)\right\} \qquad \textbf{(5)}$$

Where the quantity on the left represents the cross-correlation of the IR function of focus with the IR function of any other location $h_j(t)$. The quantity on the right represents the auto-correlation of focus's IR function.

Figure 7 shows the Quad with the iso-IR band limit contours. A band is enclosed within two iso-IR limit contours. For the purpose of clarity only every other contour is shown in the figure. Using the difference tolerance of 5%, 28 iso-IR contours were obtained. This shows a factor of 100 reduction in the number of IR functions (~3000 to ~30) required to generate the response of the PGC within the given difference tolerance.

The categorization of IR functions into iso-IR bands reduces the maximum number of convolution operations to the total number of identified iso-IR bands. This is due to the superposition property of a linear system explained using Eq. 6. Where B represents the total number of iso-IR bands on the grid and S represents the total number of input locations inside a given iso-IR band. As the equation suggests the output $y_{C40}[n]$ is a linear superposition of responses due to each iso-IR band, $y_j[n]$. Ideally, computation of each $y_j[n]$ requires S convolutions, however, due to creation of iso-IR bands the complexity involved in the computation of each $y_j[n]$ reduces to a single convolution operation. This helps to reduce the total number of convolutions to B.

$$y_{C40}[n] = \sum_{j=0}^{B} y_j[n] \qquad \textbf{(6)}$$

$$y_j[n] = \sum_{i=0}^{S} h_i[n] \otimes f_i[n] = h[n] \otimes \sum_{i=0}^{S} f_i[n]$$

Where $y_{C40}[n]$ is the overall response (i_{DDT}) measured at $C4_0$,
$y_j[n]$ is response due the inputs in an iso-IR band (j) at $C4_0$,
$h_i[n]$ is the IR function from each input in an iso-IR band to $C4_0$,
$f_i[n]$ is the input signal inside an iso-IR band and h[n] represents the IR function of the focus.

Figure 7. Iso-IR contours depicting the regions with similar impulse response at C4 V_{DD0}.

8.0 Isolated Path (CLC) Simulations

Although the aforementioned technique greatly simplifies the generation of power-grid response, generation of transient current signals in the CLC plays a significant role in determining the overall complexity of the fault simulation process. We present a technique, referred to as *path-isolation scheme*, based on divide and conquer strategy, for reducing the complexity of CLC transient simulation.

The path-isolation scheme exploits the fact that, under an i_{DDT} test pattern only finite number of logic paths are sensitized that give rise to a transient current pulse with the switching of every gate along that path. Therefore, only the sensitized paths need to be considered to compute the transient currents in the CLC. The sensitized logic paths for each test pattern are identified and are physically isolated from the design layout. These paths are then extracted to obtain their detailed R, L, C, and MOS-transistor circuit representation. In case of fanout paths branches that do not propagate the logic transition are not considered in the isolated path (therefore only sensitized portions of logic paths are isolated). In such cases an equivalent discrete capacitance is placed at the fanout node to represent the capacitive loading of the ignored branch (or branches).

In cases where more than one path is sensitized by the input pattern, the simulation complexity can be reduced extracting each independent (non-overlapping) path into a separate circuit that can be simulated in parallel. The circuit corrensponding to each independent isolated path is simulated using SPICE and the transient current drawn by them at various CLC output ports are measured. The current signal measured at each CLC output port is convolved with the current-to-current IR of the corresponding PGC port to obtain the transient current signal at the C4. The transient current signals thus obtained at the C4, are linearly added to obtain the composite i_{DDT} signal. Thus, the task of simulating the entire CLC can be replaced with a finite number of simultaneous transient simulations performed on much smaller circuits.

. We can analyze the complexity involved in such a scheme by assuming the average number of gates sensitized under a given test sequence. If we assume the maximum fanout (F_O) of 4, logic depth (D) of 5 and maximum fanin (F_I) of 3, the maximum number of sensitized gates can be computed using geometric progression given by Eq. 7. Assuming that the minimum number of gates sensitized under a test sequence is 5 (30 transistors), the average num-

$$Max\ \ Numer\ \ of\ \ gates = \frac{F_O^{\ D} - 1}{F_O - 1} = 341 \qquad \textbf{(7)}$$

$$Max\ \ Numer\ \ of\ \ transistors = 2F_I \cdot \frac{F_O^{\ D} - 1}{F_O - 1} = 2046$$

ber of transistors per test sequence is 1038. This shows a significant reduction in the number of transistors that need to be simulated under a given test sequence compared to the entire CLC of the DUT. It must be realized that under any given test sequence several paths may get sensitized independently. All such independently sensitized paths can be isolated and simulated in parallel and their results can be combined during the PGC convolution process. Instead of using SPICE to simulate the isolated paths we may also make use of tools that perform Transient Current Simulations at the switch level verilog netlist. We are currently investigating the accuracy and complexity of Transient Current Simulations. Also results for paths already simulated when processing a previous input test vector can be reused for other test sequences.

9.0 Computation of i_{DDT} using Convolution and Superposition

The accuracy of the convolution and superposition based computation of i_{DDT} is demonstrated using a full-custom designed 16-bit logarithmic adder as a representative of the sensitized portion in the CLC. Figure 8(a) shows the approximate location of the adder in the lower left corner of the PGC. The layout for the 16-bit logarithmic adder is shown in Figure 8(b). The gates in the layout consist of transistors with W/L ratios ranging from 2 to 5 for NMOS and 3 to 7 for PMOS (however, most are minimum size). The power rails of the adder are connected to SPICE voltage sources at the six labeled points, V_{a0} through V_{a5}. These points are determined by locating the intersection of each label placed in the PGC with the local V_{DD} and GND rails of the adder. Therefore, it may be considered a case of partitioning using the FP-scheme as the local rails are considered part of the CLC. The GND connection points (not shown) are adjacent to the V_{DD} connection points.

Figure 8. (a) A 16-bit logarithmic adder connected to the PGC. (b) Layout of the adder.

The six V_{DD} input locations in the PGC (V_{a0} through V_{a5}) were found to traverse 4 different iso-IR bands. Therefore, the computation of power grid response for a given input sequence entails six convolution operations using four IR functions. Figure 9(a) shows the current transients measured at the six input locations using SPICE simulations.

The response of the PGC at V_{DD0} obtained by convolving each of the current waveforms with their corresponding IR functions is shown in Figure 9(b). The overall response of PGC to the sensitized adder logic measured at V_{DD0} is obtained by linear superposition of its response to individual current sources. Figure 10(a) shows the overall response of the PGC to the sensitized adder logic at V_{DD0} obtained using convolution based method overlaid with the response obtained using SPICE simulations.

Figure 9. (a) Adder i_{DDT} waveforms that form inputs the PGC. (b) Response of the PGC to the individual i_{DDT}s in figure 9 (a).

The complexity involved in the computation of power grid response based on convolution operations can be further reduced by creating piece-wise-linear (PWL) abstractions of the input current waveforms. An algorithm based on detection of the change in polarity of slope in a waveform is used to derive it's PWL abstraction. This significantly reduces the number of points in each input signal and thus the number of multiplication and addition operations. Figure 10(b) shows the overall response of the grid obtained using convolution based on PWL abstracted inputs. Overlaid with this curve is the SPICE generated response obtained using original SPICE waveforms as inputs. The peak amplitude and width values obtained using SPICE and the percentage difference error as compared to the convolution results on original SPICE and PWL inputs are listed in Table 1. The width of i_{DDT} is measured as the time-interval between the points at which the waveform attains 5% of its peak value.

Original i_{DDT}s	**PWL fitted i_{DDT}s**
5% width (spice) = 3.26ns	5% width (spice) = 3.26ns
peak (spice) = 0.594 mA	peak (spice) = 0.594 mA
% width error = 0.4%	% width error = 3%
% peak error = 1.8%	% peak error = 10%

Table 1: Amplitude and Peak Error.

Figure 10. a) Overall response using SPICE superimposed with convolution derived results. (b) Overall response using SPICE superimposed with convolution derived results of PWL fits to i_{DDT}s

10.0 Automation and Implementation of Fault Simulation Flow

The key to fault simulation of a design (at layout level) based on i_{DDT} signals, is to automate the computation of i_{DDT} signals as measured on the power supply pads (C4s) of the chip. Since the proposed fault simulation flow works at the layout level representation of the design, the layout processing steps involved in path-isolation and power-grid separation must also be automated. Figure 11 delineates the components of the tool implemented to enable automated computation of i_{DDT} signatures without requiring circuit (SPICE) simulation of the entire design. We refer to this tool as *i_{DDT} Computation Engine* or *ICE*. The ICE is implemented under the Cadence design infrastructure and follows standard digital design flow.

Figure 11. Automated Flow for i_{DDT} Generation (ICE)

The i_{DDT} computation flow requires a standard cell based layout of the design. The design layout consists of placed-and-routed standard cells as well the power-grid. This layout is processed by a tool referred to as *Path Isolation Engine*. The path isolation engine takes a transition pattern as input and creates layout of the logic path sensitized by the input pattern. It also separates the power-grid of the design layout into a separate layout. For a given design layout, the power-grid is isolated only once whereas isolated path layouts are created on a per test-pattern basis. The power-grid and the isolated path layouts are extracted using standard layout (and parasitic) extraction tool. The power-grid (PGC) then undergoes the characterization process, detailed in earlier sections, resulting in iso-IR bands. The R, C, MOS-transistor circuit representation of the isolated path layout is simulated using SPICE and transient current waveforms at pre-selected output ports are measured. These transient current waveforms, denoted as $i_{DD}(t)$ in the figure, are convolved with the impulse response of the corresponding iso-IR region of the power-grid. The result of each convolution operation is linearly added to compute the $i_{DDT}(t)$ signature corresponding to each power supply port. This flow allows fast computation of i_{DDT} signals corresponding to each selected path. Defect(s) inserted at layout (and/or circuit) level representation of the isolated path can thus be quickly simulated using this flow.

The most important part of ICE is the path isolation engine detailed next. Figure 12 depicts the various components involved in the path isolation engine. The path isolation engine comprises of three main components (a) Test-bench-Generator, (b) Path-Tracer and (c) Layout-Processor. It also requires the following as inputs:

1. synthesized gate netlist of the design
2. placed-and-routed layout and
3. transition test pattern obtained from the ATPG tool

Figure 12. Automated Flow for Path Layout Isolation

The testbench-generator processes the gate netlist and creates a verilog test-bench for the design. The transition pattern obtained from the ATPG tool is inserted into the testbench and verilog simulation is performed. The simulation dump along with the gate-netlist form the inputs to the Path-Tracer, which traces the gate netlist for components involved in the sensitized path (path undergoing logic transition from a primary input to a primary output). This path information is used by the Layout-Processor which creates the layout of sensitized path from the full-chip layout. The layout corresponding to the sensitized path is referred to as *isolated path layout*. The isolated path layout consists of only those components (i.e. standard-cells, interconnect wires, and primary input and outputs) that constitute the sensitized path. The path isolation engine also creates SKILL command files that are used to insert input/output pins in the power-grid and isolated path layout. These input/output pins define the interface between the PGC and CLC. The tool supports both types of system partitioning schemes namely, TR-scheme and FP-scheme. In case of TR-scheme, the isolated path layout consists of only the standard-cells and their signal routing. While in the FP-scheme, the isolated path layout also contains the metal-1 follow-pin rails, in addition to the standard-cells and their signal routing.

The entire ICE flow is implemented using Perl, SKILL, expect (TCL), and signal processing software. The path isolation engine works under the cadence design infrastructure and does not require any specialized tool outside the standard digital design flow.

10.1 Experimental Design

The accuracy and performance of the fault simulation tool is demonstrated on a ISCAS85[23] benchmark circuit, C499. The C499 is single-error-correcting circuit with 41 inputs and 32 outputs. The Verilog design for C499 benchmark was synthesized using Synopsys *design_vision* tool with Cadence's crete GSCLib.3.0 [24] as the target library. The synthesized gate netlist consists of 262 complex logic gates. The netlist was functionally verified using tests provided in [23]. The design layout was implemented in 1.8µm technology using Cadence's *SOC-Encounter* tool based on GSCLib.3.0 standard-cell library and gpdk_MIET_2.0 pcell library [24].

The power-grid of the chip consists of V_{DD}/GND power-ring implemented in metal-1 and metal-2 and the uniformly distributed V_{DD}/GND power-stripes implemented in metal-3 through metal-6. The power-grid contains highest density of metal-3 stripes, with lower width and pitch, and lowest density of metal-6 stripes, with higher width and pitch. The grid consists of equal number of V_{DD} and GND stripes. The chip occupies 160µm by 160µm area. Figure 13 shows the layout of the entire C499 design.

Figure 13. Layout of C499 Benchmark Design

Using the path isolation engine, the C499 layout is processed to isolate the power-grid and the individual sensitized paths into separate layouts. Figure 14 shows the power-grid layout isolated from the full-chip layout based on TR-scheme of system partitioning. The SKILL command files generated by the path isolation engine are used to insert power supply ports at user specified co-ordinates. Four power supply ports (C4s), namely V_{DD0} through V_{DD3}, are inserted on metal-6 stripes at locations indicated in the figure. Input ports are inserted at the center of V_{DD}/GND pins on the standard cells, indicating the locations where the standard cells would tie into the metal-1 (follow-pin) rails. The Cadence's *Diva* extraction tool is used to extract the full-chip and power-grid layouts. The resistance and capacitance extraction parameters in divaEXT.rul (of gpdk_MIET_2.0) are replaced with more realistic values for TSMC 1.8µm technology obtained from MOSIS[21]. The extracted full-chip layout consists 1860 MOS-transistors, 76711 resistors and 118824 capacitors. The extracted PGC consists of 70,406 resistors and 17,632 capacitors. Decoupling capacitors are inserted at each input port location in the PGC to account for the inserted-decoupling capacitor and the diffusion capacitance contributed by each standard-cell. The PGC is characterized using the flow shown in figure 11.

Figure 14. Power Grid Layout of C499 obtained using TR-scheme

The path transition patterns are derived manually to sensitize logic paths between primary-inputs and primary-outputs. The patterns are derived such that only one primary input changes its logic state similar to the traditional transition-fault test patterns. For each path transition pattern, the corresponding isolated path layout is created using the path isolation engine. The isolated path consists of only a small fraction of gates of the full-chip layout. Furthermore, the isolated path circuit is simulated without power-grid thus significantly reducing the path simulation time.

10.1.1 Fanout Load Consideration for Path-isolation

The fanout load of each gate in the sensitized path plays an important role in determining the characteristics of the

i_{DDT} waveform created during switching of these gates. Since, any logic gate on the sensitized path can fanout to one or more non-switching gates, it is important to maintain the same fanout load on each gate in the isolated path. The path isolation engine provides two different modes to account for the fanout loads. In one mode, referred to as *fanout-cap mode*, the tool extracts all the fanout-nets of every gate in the sensitized path and inserts an output pin at the end of each fanout-net. After the isolated path layout is extracted, a discrete capacitor is connected at each of the inserted output pins. This capacitor represents the total gate-capacitance of MOS transistors in the fanout gate and its value is derived from the LEF (Library Exchange Format) database of the standard-cell library. This mode allows us to account for the RC load of each gate in the sensitized path without increasing the device count. Figure 15 shows the layout of an isolated path created by the path isolation engine in the fanout-cap mode.

Figure 15. Isolated Path Layout Created using fanout-cap Mode under TR-scheme

The other mode, referred to as *fanout-cell mode*, not only extracts fanout-nets of every gate in the sensitized path, but also extracts the fanout gates (standard-cells) connected to these nets into the isolated path layout. This mode is likely to provide more accurate i_{DDT} signals at the cost of additional simulation time caused by the inclusion of fanout gates. Figure 16 shows the layout of the isolated path created by the path isolation engine in the fanout-cell mode.

Figure 16. Isolated Path Layout Created using fanout-cell Mode under TR-scheme

It is observed that any given interconnect in the extracted isolated path layout, consists of fewer capacitors than it contains in the extracted full-chip layout. This is due to the fact that in the full-chip extracted layout the interconnect is capacitively coupled to several other nets that are not present in the isolated path layout. These include, V_{DD}/GND stripes in power-grid and interconnects on other paths that are not sensitized by the given test pattern. The absence of these capacitive elements can cause the isolated path circuit to appear faster than the full-chip circuit under the same test pattern. In order to account for these capacitive elements the path isolation engine can scale the capacitance extraction coefficients corresponding to each metal layer, by a given percentage during the extraction process. These include *Plate*, *Sidewall*, and *Fringe* capacitance coefficients. The percentage increase in these coefficients is determined experimentally. In this experiment the capacitance extraction coefficients of the isolated path layouts were increased by ~35% of the values used for full-chip layout extraction. However, some layout extraction tools like

Assura (Cadence) allows extraction of these coupling capacitors and may be used to avoid the need to scale the capacitance coefficients.

10.2 Experimental Results

We manually derived 10 path transition patterns to demonstrate the accuracy of ICE. The R, C, and MOS-transistor netlist of the isolated paths are simulated using Berkeley SPICE. The full-chip netlist of C499 is also simulated using SPICE for all the 10 patterns. The i_{DDT} signal measured at each C4 (VDD0 in this case) during full-chip simulation is compared with the i_{DDT} computed using ICE, for all the 10 patterns. For instance figure 17(a) shows the i_{DDT} signal measured using the two methods for one of the transition paths. In this case, the two i_{DDT} signals appear almost identical with small differences in the amplitudes at certain time values. Figure 17(b) shows the similar comparison for another transition path where slightly higher error is obtained in peak amplitude of the two i_{DDT} signals. In order to quantify the distinction between the i_{DDT} signals (obtained using SPICE simulation versus ICE) we measure three different parameters. These parameters include (a) width measured at 5% of the peak i_{DDT} value, (b) peak amplitude and (c) area under the i_{DDT} signal (also represents the charge content of the i_{DDT} signal).

Figure 17. Comparison of i_{DDT} signals for two different paths (a) and (b) full-chip SPICE simulation vs. ICE generated

Table 2 shows the results obtained with full-chip SPICE simulation of C499 (R, C and MOS transistor netlist) for 10 transition patterns. The average full-chip simulation time for the 10 transition patterns is 20 hours. Table 3 shows the results obtained with SPICE simulation of isolated paths using ICE. In contrast to full-chip netlist, the isolated paths contain 25 standard cells on an average. The average simulation time of the isolated path is 94 seconds. Table 4 gives the percentage error in width, peak and area values of i_{DDT} signals obtained using the two methods (given in table 2 and 3). The average percentage error in i_{DDT} width, area and peak amplitude is 2.52, 2.83 and 4.21 respectively. We believe that the above error can be further reduced with tighter control on the layout extraction procedure and distribution of decoupling capacitance in PGC.

The power-grid characterization takes 16 hours using SPICE simulations. It must be realized that the PGC characterization time can be significantly reduced (factor of ~10) using linear solvers or power-grid simulators. A major

Path #	Standard Cell Count	Simulation Time(Hr)	5% Width (ns)	i_{DDT} Area (E-13)	Peak Amplitude (E-04)
1	262	17	2.13	7.25	6.89
2	262	19	2.51	2.70	1.88
3	262	17	2.18	2.51	2.50
4	262	20	2.90	3.65	3.84
5	262	18	2.58	3.14	3.14
6	262	19	2.61	3.32	3.37
7	262	23	2.40	2.91	2.56
8	262	20	1.89	2.84	2.79
9	262	25	3.28	8.94	6.23
10	262	22	3.37	9.84	6.19
Ave	262	20Hr			

Table 2: Results for full-chip SPICE simulation of 10 transition paths in C499

Path #	Standard Cell Count	Simulation Time(Hr)	5% Width (ns)	i_{DDT} Area (E-13)	Peak Amplitude (E-04)
1	36	93	2.05	7.39	7.27
2	16	30	2.59	2.74	1.94
3	15	28	2.20	2.56	2.55
4	20	48	2.81	2.53	3.81
5	18	56	2.67	3.29	3.13
6	19	62	2.63	3.40	3.36
7	16	38	2.31	2.89	2.56
8	18	33	1.83	2.88	3.18
9	44	252	3.34	8.62	5.75
10	47	300	3.35	9.15	5.70
Ave	25	94sec			

Table 3: Results for isolated path simulation of 10 transition paths in C499

advantage offered by the proposed method comes from the fact that PGC characterization involves multiple simulations of the same R, L, C network. Hence, this task can be easily split amongst several machines computing in parallel. For instance, the PGC characterization took less than 3 hours when the 262 simulations were split amongst 5 computers. It is re-emphasized that PGC characterization is done only once and can be started at an early stage in physical design phase, once the design has been frozen. Once characterized, the i_{DDT} signals corresponding to any number of paths can be quickly generated. It must also be realized that the PGC can also be characterized using fast power-grid solvers (linear solvers) instead of SPICE based tools. This can further decrease the PGC characterization time by a factor ~10 [15].

Path #	%Width Error	%Area Error	%Peak Amplitude Error
1	3.79	1.88	5.54
2	3.23	1.54	3.36
3	1.01	1.68	1.97
4	3.15	3.26	0.89
5	3.64	4.86	0.31
6	0.65	2.27	0.26
7	3.74	0.77	0.06
8	3.55	1.63	13.9
9	1.96	3.48	7.80
10	0.50	6.93	7.97
Ave	2.52	2.83	4.21

Table 4: Percentage error between results obtained using SPICE and ICE

In order to make a fair comparison of the performance of ICE in computing the i_{DDT} signals with the full-chip SPICE based simulations, we must factor-in the time taken to characterize the PGC. Equation 8 gives a generic expression that represents the total time taken to compute an i_{DDT} signal using ICE in terms of the PGC characterization time. In this equation, T_i represents the total time required to compute an i_{DDT} signal for a sensitized path

$$T_i = T_{IPSi} \div \frac{T_{PGC}}{N} \qquad \textbf{(8)}$$

denoted by i. T_{IPSi} denotes the time taken in simulating the extracted isolated-path layout for path i using SPICE, T_{PGC} denotes the time required to characterize the PGC and N denotes the total number of paths (or test-patterns) for which $\mathrm{i_{DDT}}$ needs to be computed. Therefore, T_{PGC} is equally divided amongst all the N paths. It is interesting to note that as the number of paths increase the T_i asymptotically approaches T_{IPSi}. In contrast, the time required to derive $\mathrm{i_{DDT}}$ using full-chip simulations remains almost constant. It must be realized that for designs of practical dimensions it is impossible to perform full-chip SPICE simulations and hence, not a feasible option. Thus, the ICE based approach not only makes it possible to derive $\mathrm{i_{DDT}}$ signals (without full-chip SPICE simulations), it provides a fast means of analytically computing them with SPICE -like accuracy.

11.0 Related Applications

In addition to enabling the fault simulation of $\mathrm{i_{DDT}}$ test methods, the proposed simulation model has several other applications.

The impulse response based characterization of linear time-invariant systems, proposed in this work, also enables on-chip impulse response signature generation in analog and mixed-signal circuits. These signatures implicitly represent the overall functional characteristics of a circuit. Analysis of these signatures based on the auto- and cross-correlation approach, also proposed in this work, exhibits better defect detection capability compared to a typical specification-based test method [22].

Timing analysis is a key component in the design verification flow that verifies the compliance of modeled design's timing to the timing specifications of the intended product. Voltage drop across the resistive power-grid degrades the performance of the circuit. To account for this degradation designers generally over-design the circuit by considering the peak voltage drop across the power grid. This can lead to (a) wasted chip resources, (b) increased susceptibility to manufacturing defects, (c) inaccuracy in timing analysis and (d) significant time and effort on the designer's part in optimizing the design. The proposed fault simulation procedure involves derivation of spatial current and voltage distribution profile of power grid, required for system characterization. This spatial voltage profile can also be used to perform more accurate and realistic timing analysis.

12.0 Conclusions

This paper investigates the practical issues concerning the implementation of fault simulation methodology for i_{DDT} testing. A new model based on convolution based approach is proposed that can be used to compute the power grid response using the precomputed impulse response (IR) functions. This circumvents the need for running time and memory intensive transient simulations on the entire DUT. The categorization of IR functions into iso-IR bands is shown to further reduce the number of convolutions required to compute the transient response of the grid to the maximum number of iso-IR bands. An approach based on isolation of sensitized paths from the layout is proposed as a means of enabling the simulation of core logic circuit. The accuracy and complexity of these methods are evaluated on part of a commercial power grid using a 16-bit logarithmic adder as the core-logic.

13.0 References

[1]ITRS (website: http://public.itrs.net/).

[2]J. F. Frenzel and P. N. Marinos, “Power supply current signature (PSCS) analysis: A new approach to system testing”, IEEE proceedings of International Test Conference, pp. 125–135, 1987.

[3]M. Hashizume, et al., “Fault detection of combinatorial circuits based on supply current”, IEEE proceedings of International Test Conference, pp. 374–380, 1988.

[4]A. P. Dorey, et al., “Reliability testing by precise electrical measurement”, IEEE proceedings of International Test Conference, pp. 369–373, 1988.

[5]F. Franco and E. J. McCluskey, “Delay testing of digital circuits by output waveform analysis”, IEEE proceedings of International Test Conference, pp. 798–807, 1991.

[6]J. S. Beasley, et al., “IDD pulse response testing of analog and digital CMOS circuits”, IEEE proceedings of International Test Conference, pp. 626–634, 1993.

[7]D. K. Papakostas and A. A. Hatzopoulos, “Analogue Fault Identification Based on Power Supply Current Spectrum”, Electronics Letters, 7th January Vol. 2, No. 1, pp. 118-119, 1993.

[8]G. Gielen, et al., “Fault Detection and Input Stimulus Determination for the Testing of Analog Integrated Circuits Based on Power-Supply Current Monitoring”, IEEE/ACM ICCAD, pp. 495-498. 1994.

[9]J. H. Wang, et al., “An Accurate Time-Domain Current Waveform Simulator for VLSI Circuits”, Proceedings of EDAC, pp. 562 - 566, 1994.

[10]A. Barber, et al., “A Bare-chip Probe for High I/O, High Speed Testing”, HP Labs 1994 Technical Report-18, Hewlett Packard,1994. (http://www.hpl.hp.com/techreports/94/HPL-94-18.pdf)

[11]R. Z. Makki, et al., “Transient power supply current testing of digital CMOS circuits”, IEEE proceedings of International Test Conference, pp. 892–901, 1995.

[12]B.Vinnakota, “Monitoring Power Dissipation for Fault Detection”, IEEE proceedings of VLSI Test Symposium, pp. 483-488,

1996.

[13]A. Bogliolo, et al., “Gate-Level Power and Current Simulation of CMOS Integrated Circuits”, IEEE Transactions on VLSI systems, Vol. 5, No. 4, 1997.

[14]L. Smith, et al., “Power Distribution System Design Methodology and Capacitor Selection for Modern CMOS Technology”, IEEE Transactions on Advanced Packaging, Vol. 22, pp 284-291, 1999.

[15]S. R. Nassif and J. N. Kozhaya, “Fast Power Grid Simulation”, Proceedings of the 37th conference on Design Automation, pp 156 - 161, 2000.

[16]J. H. Wang, et al., “Current Waveform Simulation for CMOS VLSI Considering Even-Overlapping”, IEICE Trans. Fundamentals, Vol. E83-A, No.1, 2000.

[17]A. Singh, C. Patel, S. Liao, J. F. Plusquellic and A. Gattiker, “Detecting Delay Faults using Power Supply Transient Signal Analysis”, IEEE Proceedings of International Test Conference, pp.704-712, 2001.

[18]Y. Lee and C. C. Cheng, “Power Grid Transient Simulation in Linear Time Based on Transmission-Line-Modeling Alternating-Direction-Implicit Method”, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp 1343-1352, Vol. 21, NO. 11, Nov. 2002.

[19]Y. Lee and C. Chen “The Power Grid Transient Simulation in Linear Time Based on 3D Alternating-Direction-Implicit Method”, Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, 2003.

[20]T. Wang, et al., “3D Thermal-ADI: An Efficient Chip-Level Transient Thermal Simulator”, ACM International Symposium on Physical Design (ISPD), ACM 1581136501/ 03/0004 2003.

[21]MOSIS (website: http://www.mosis.edu/Technical/Testdata/tsmc-025-prm.html)

[22]A. Singh, C. Patel, and J. F. Plusquellic, “On-Chip Impulse Response Generation for Analog and Mixed-Signal Testing“, IEEE Proceedings of International Test Conference (ITC), pp.262-270, 2004.

[23]ISCAS85 Benchmark Circuits at http://www.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html.

[24]Cadence Crete Digital Design Library at http://crete.cadence.com.

Figure Captions

Figure 1: Linear and non-linear system representation of a chip.

Figure 2: Flow diagram of the fault simulation procedure for i_{DDT} testing.

Figure 3: Partitioning schemes of a chip into linear and non-linear systems.

Figure 4: The "Quad": A portion of the commercial power grid used in the simulation experiments.

Figure 5: (a) IRs from a sample current source location, i, to all the C4s. (b) Step response. (c) Impulse Response.

Figure 6: (a) Triangle input current to the PGC (b) Superimposed SPICE and Convolution responses.

Figure 7: Iso-IR contours depicting the regions with similar impulse response at C4 V_{DD0}.

Figure 8: (a) A 16-bit logarithmic adder connected to the PGC. (b) Layout of the adder.

Figure 9: (a) Adder i_{DDT} waveforms that form inputs the PGC. (b) Response of the PGC to the individual i_{DDT}s in figure 9 (a).

Figure10: (a) Overall response using SPICE superimposed with convolution derived results. (b) Overall response using SPICE superimposed with convolution derived results of PWL fits to i_{DDT}s.

Figure11: Automated Flow for i_{DDT} Generation (ICE).

Figure12: Automated Flow for Path Layout Isolation.

Figure13: Layout of C499 Benchmark Design.

Figure14: Power Grid Layout of C499 obtained using TR-scheme.

Figure15: Isolated Path Layout Created using fanout-cap Mode under TR-scheme.

Figure16: Isolated Path Layout Created using fanout-cell Mode under TR-scheme.

Figure17: Comparison of i_{DDT} signals for two different paths (a) and (b) - full-chip SPICE simulation vs. ICE generated.

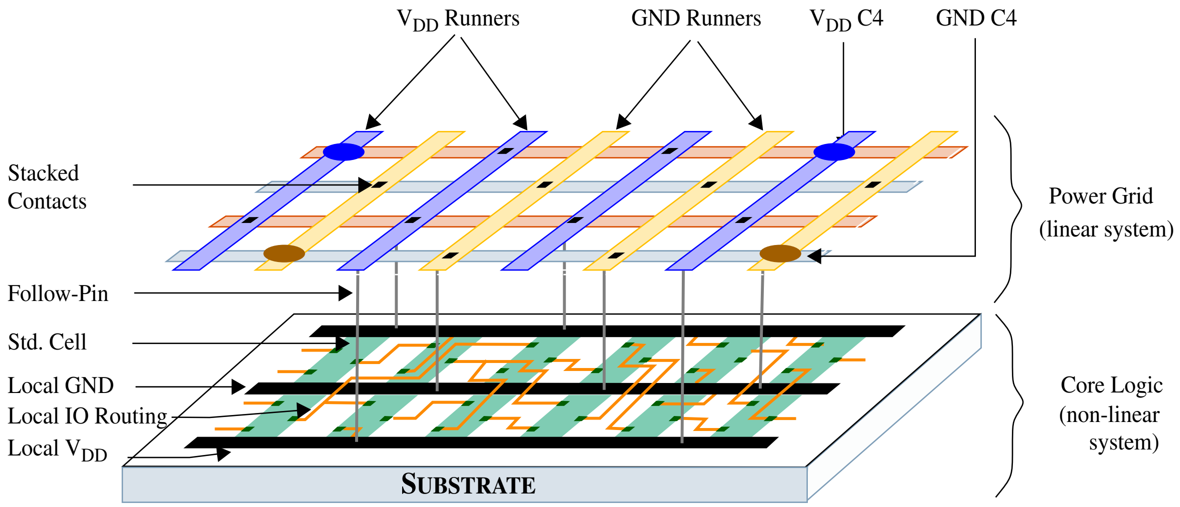
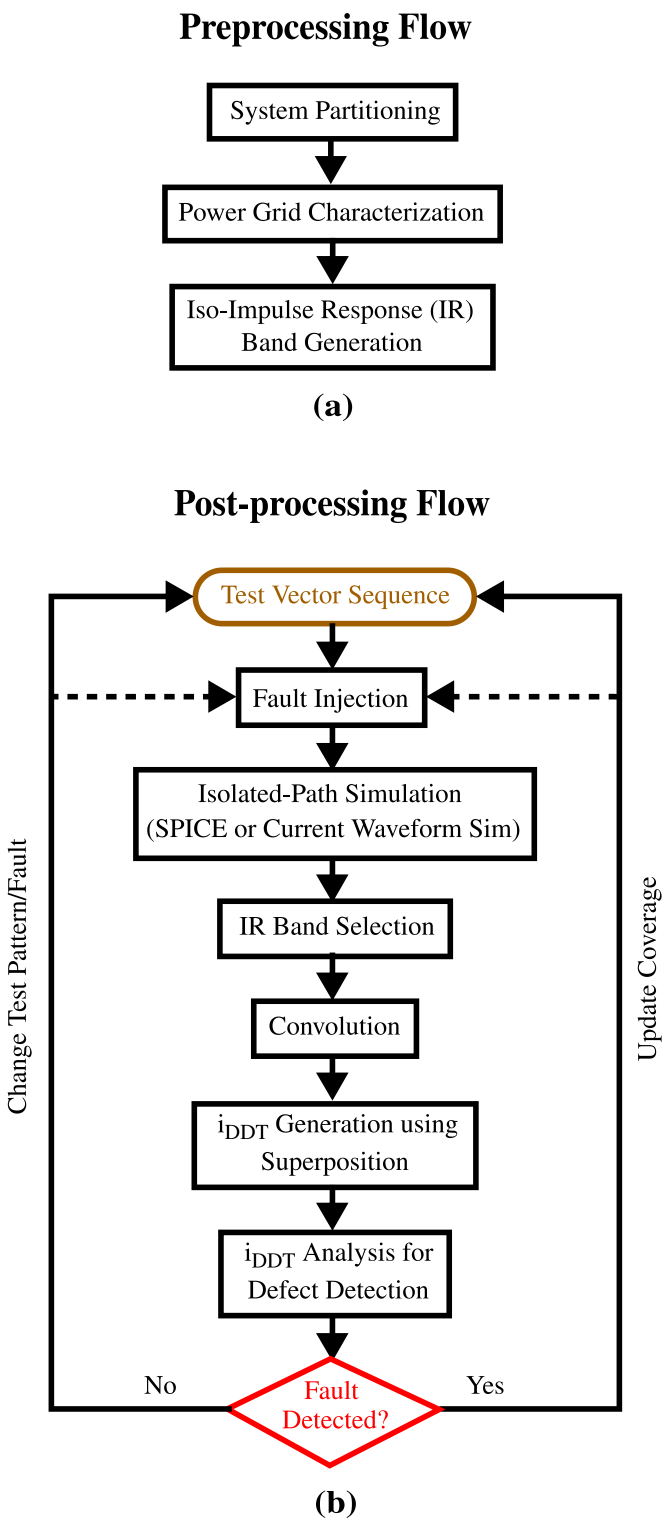

V_{DD} Runners
GND Runners
V_{DD} C4
GND C4
Stacked Contacts
Follow-Pin
Std. Cell
Local GND
Local IO Routing
Local V_{DD}
SUBSTRATE
Power Grid (linear system)
Core Logic (non-linear system)

Figure 1



Preprocessing Flow

System Partitioning
Power Grid Characterization
Iso-Impulse Response (IR) Band Generation

(a)

Post-processing Flow

Test Vector Sequence
Fault Injection
Isolated-Path Simulation (SPICE or Current Waveform Sim)
IR Band Selection
Convolution
iDDT Generation using Superposition
iDDT Analysis for Defect Detection
Fault Detected?
No
Yes
Change Test Pattern/Fault
Update Coverage

(b)

Figure 2

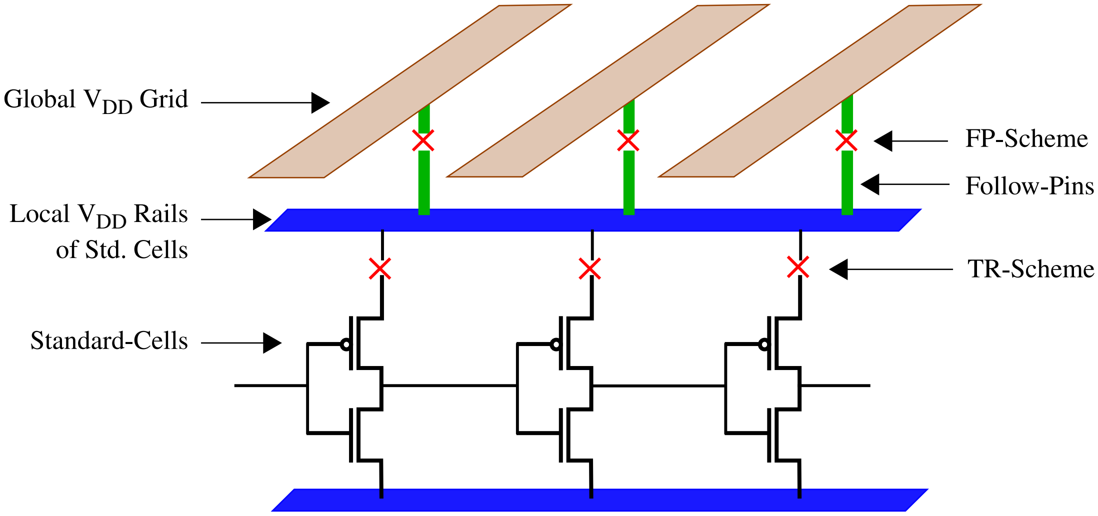

Global V_{DD} Grid
FP-Scheme
Follow-Pins
Local V_{DD} Rails of Std. Cells
TR-Scheme
Standard-Cells

Figure 3

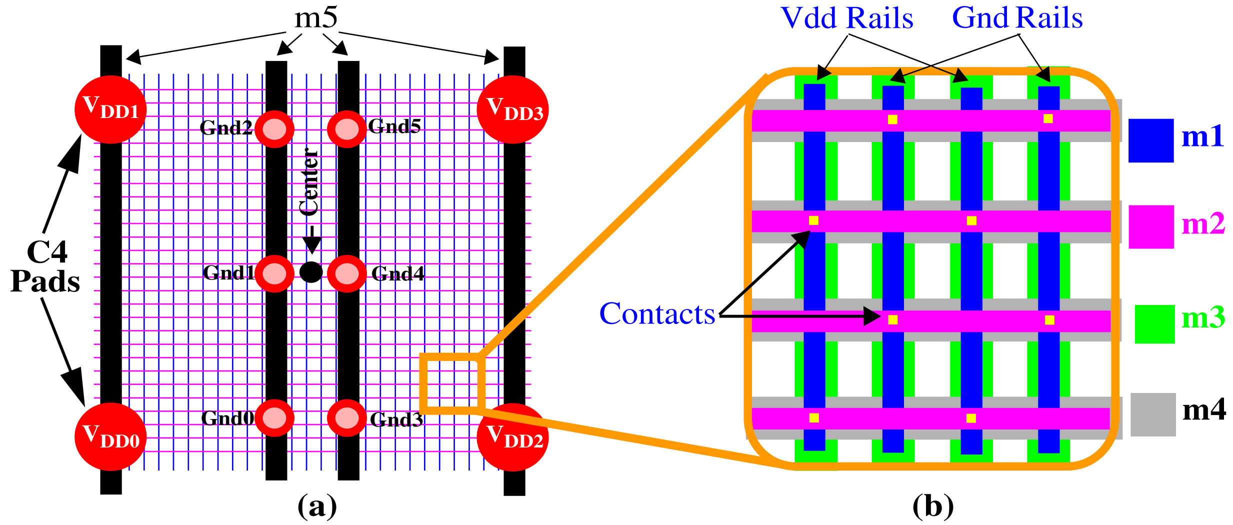
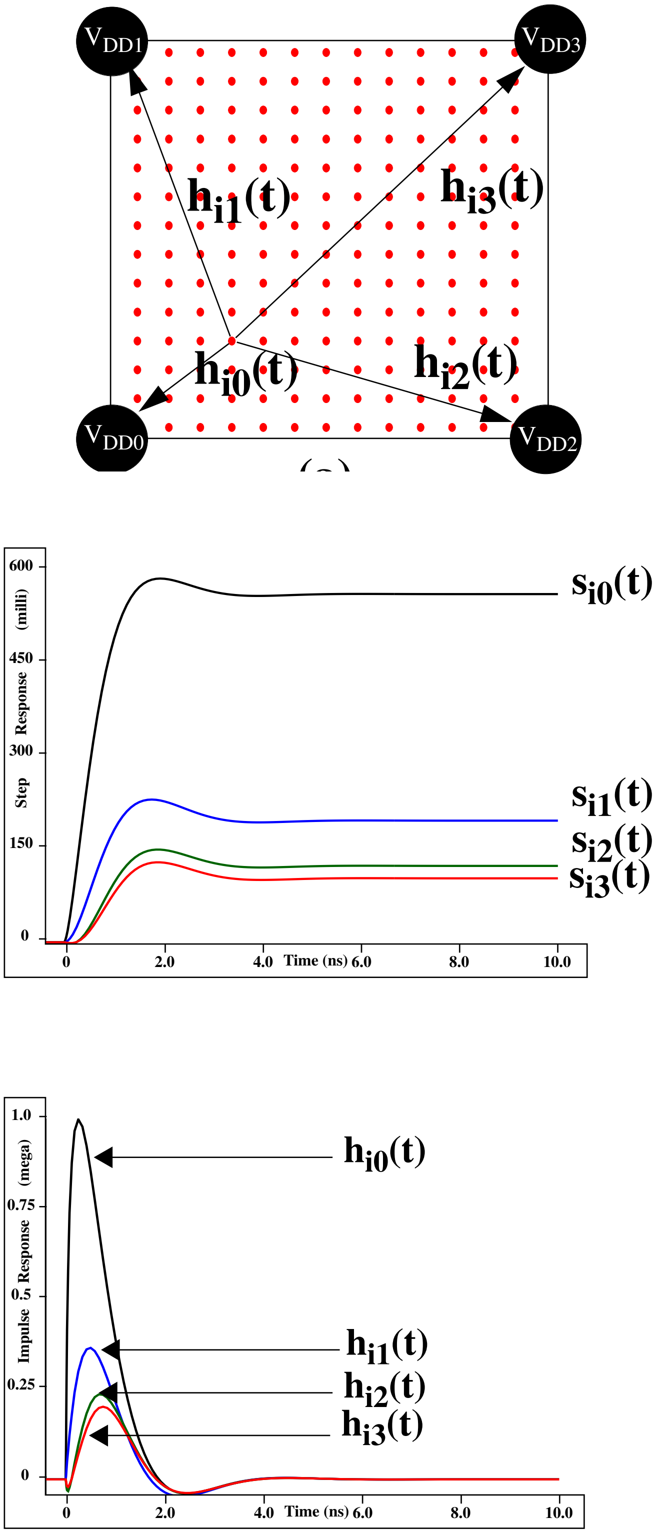
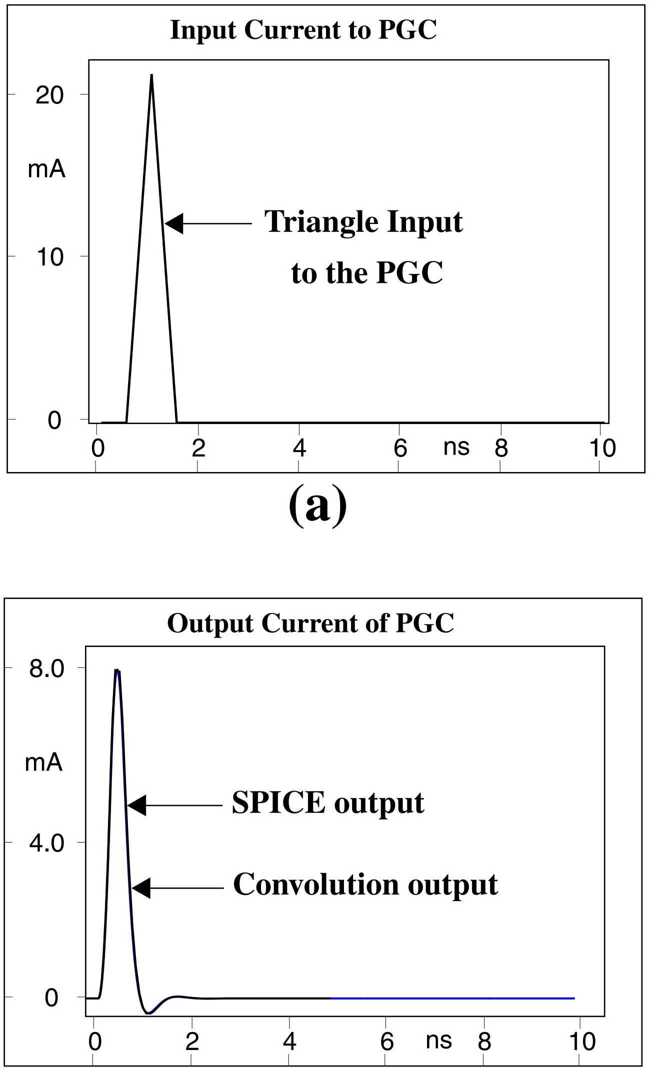

m5
VDD1
VDD3
Gnd2
Gnd5
Center
C4
Pads
Gnd1
Gnd4
Gnd0
Gnd3
VDD0
VDD2
(a)
Vdd Rails
Gnd Rails
m1
m2
Contacts
m3
m4
(b)

Figure 4



V_{DD1}
V_{DD3}
$h_{i1}(t)$
$h_{i3}(t)$
$h_{i0}(t)$
$h_{i2}(t)$
V_{DD0}
V_{DD2}
Step Response (milli)
600
450
300
150
0
0
2.0
4.0
Time (ns)
6.0
8.0
10.0
$s_{i0}(t)$
$s_{i1}(t)$
$s_{i2}(t)$
$s_{i3}(t)$
Impulse Response (mega)
1.0
0.75
0.5
0.25
0
0
2.0
4.0
Time (ns)
6.0
8.0
10.0
$h_{i0}(t)$
$h_{i1}(t)$
$h_{i2}(t)$
$h_{i3}(t)$

Figure 5


Input Current to PGC
20
mA
10
0
0
2
4
6
ns
8
10
Triangle Input
to the PGC
Output Current of PGC
8.0
mA
4.0
0
0
2
4
6
ns
8
10
SPICE output
Convolution output

(a)

(b)

Figure 6

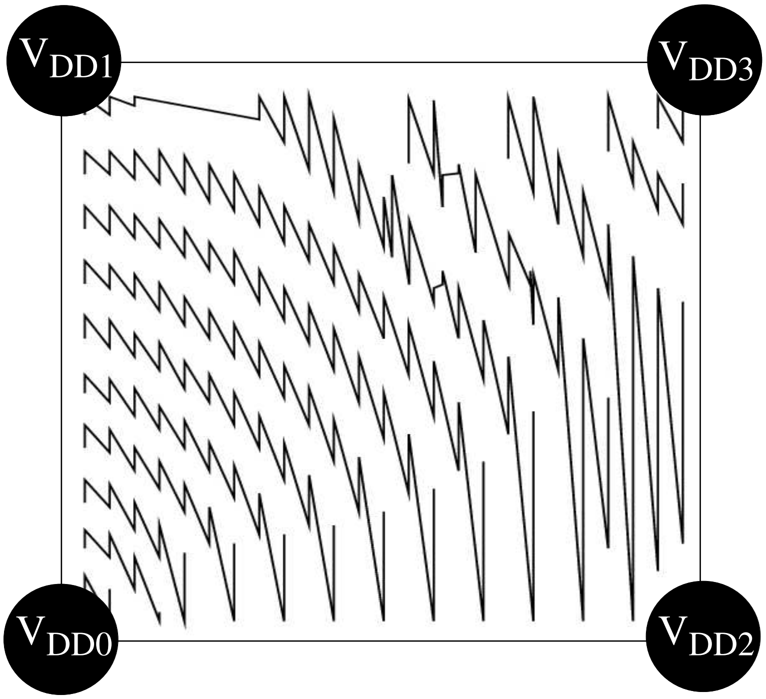


Figure 7

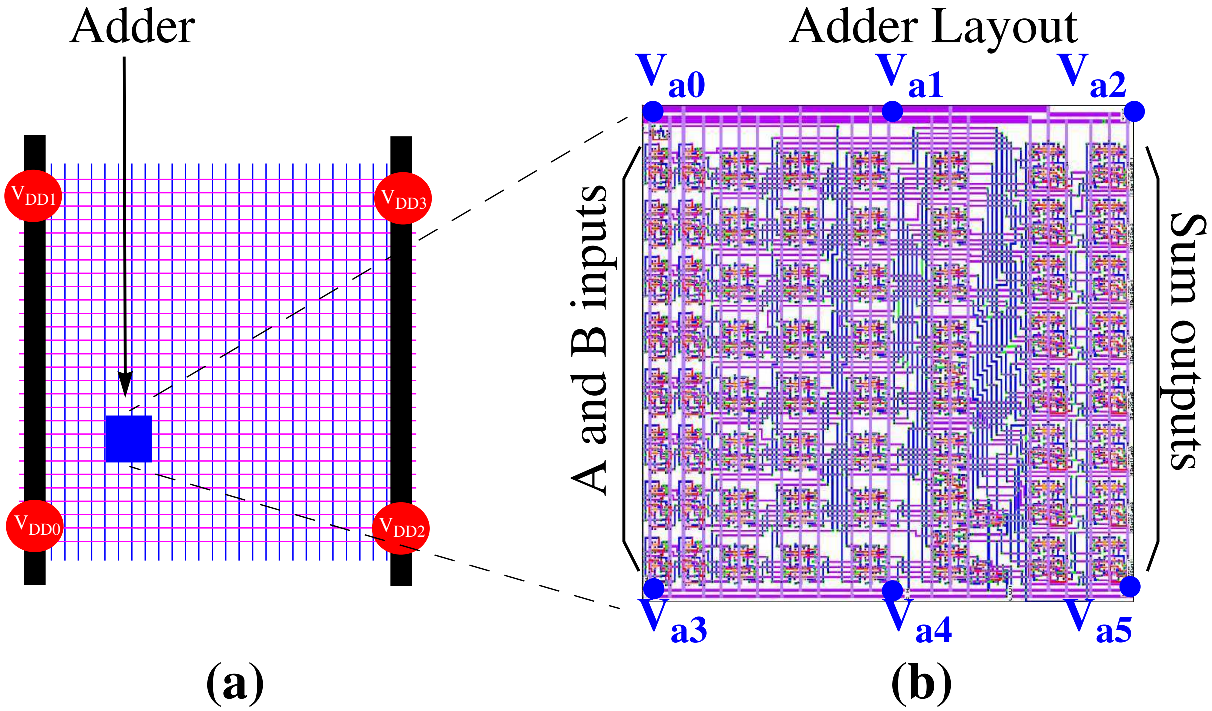
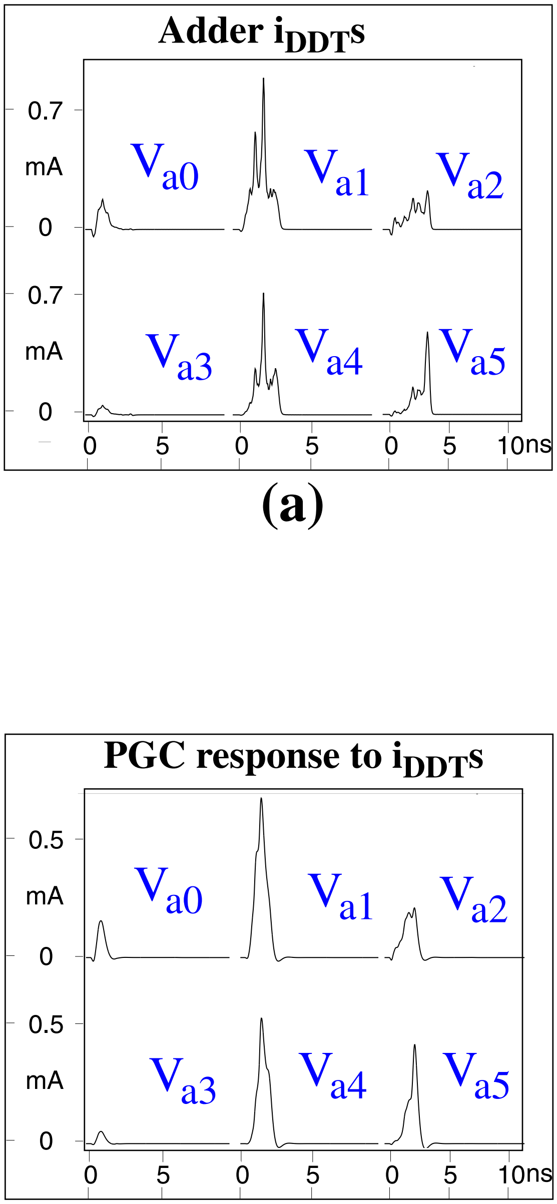

Adder
V_{DD1}
V_{DD3}
V_{DD0}
V_{DD2}
(a)
Adder Layout
V_{a0}
V_{a1}
V_{a2}
A and B inputs
Sum outputs
V_{a3}
V_{a4}
V_{a5}
(b)

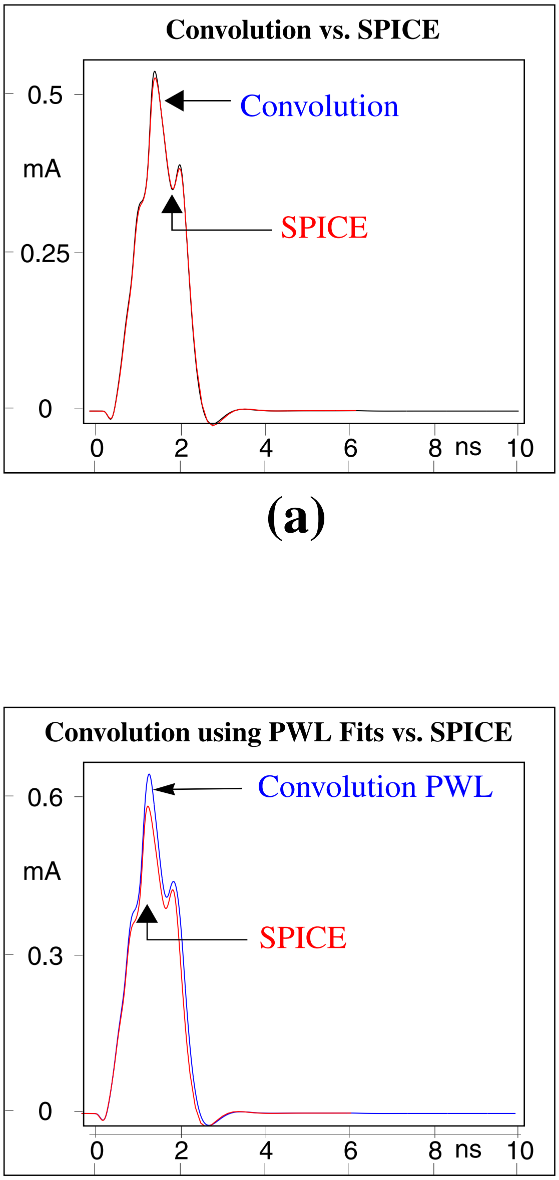
Figure 8


Adder i_{DDT}s
0.7
mA
0
0.7
mA
0
V_{a0}
V_{a1}
V_{a2}
V_{a3}
V_{a4}
V_{a5}
0
5
0
5
0
5
10ns
PGC response to i_{DDT}s
0.5
mA
0
0.5
mA
0
V_{a0}
V_{a1}
V_{a2}
V_{a3}
V_{a4}
V_{a5}
0
5
0
5
0
5
10ns

(a)

(b)

Figure 9


Convolution vs. SPICE
0.5
mA
0.25
0
0
2
4
6
8
ns
10
Convolution
SPICE
Convolution using PWL Fits vs. SPICE
0.6
mA
0.3
0
0
2
4
6
8
ns
10
Convolution PWL
SPICE

(a)

(b)

Figure 10

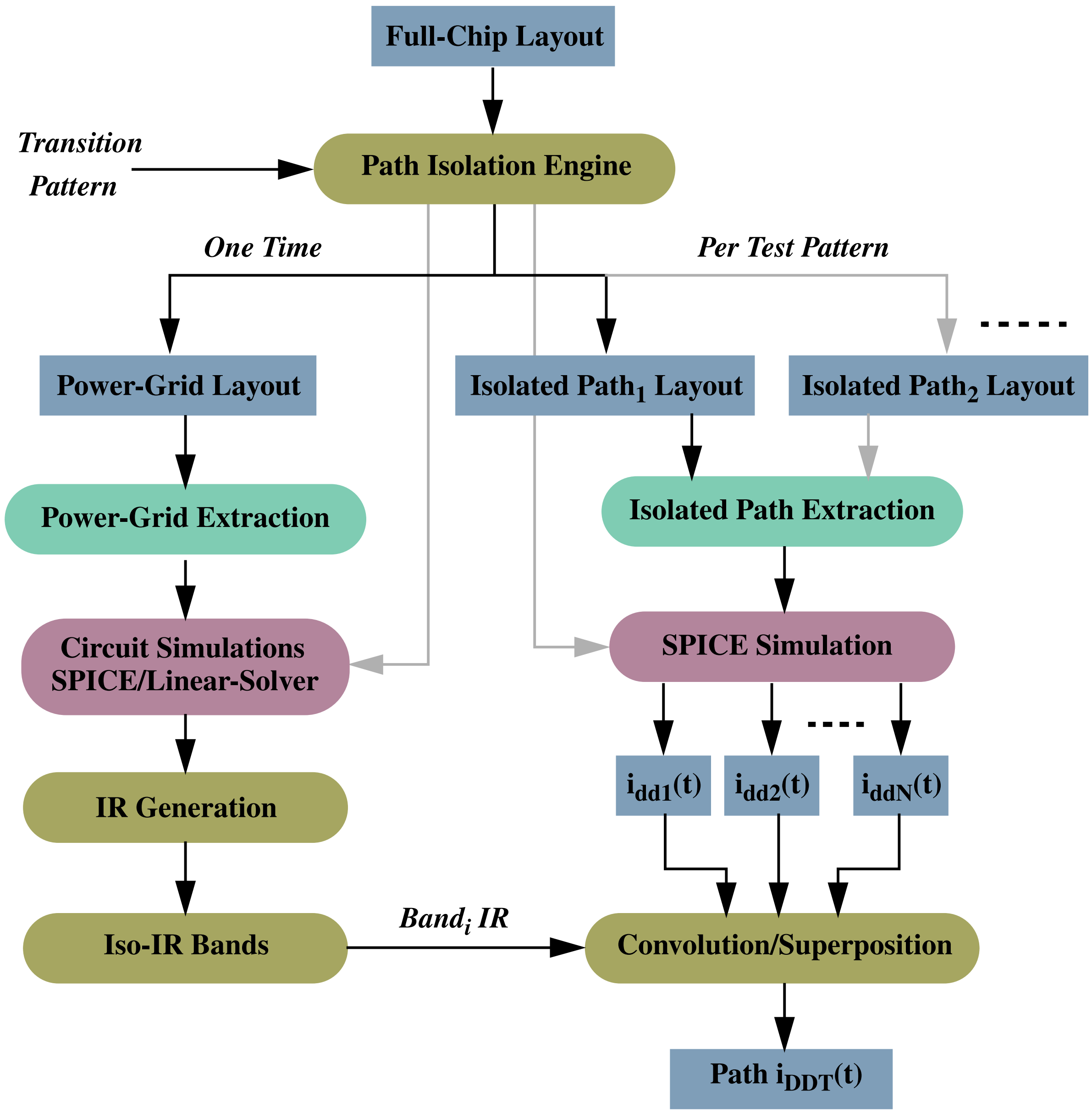

Full-Chip Layout
Transition Pattern
Path Isolation Engine
One Time
Per Test Pattern
Power-Grid Layout
Isolated Path$_1$ Layout
Isolated Path$_2$ Layout
Power-Grid Extraction
Isolated Path Extraction
Circuit Simulations SPICE/Linear-Solver
SPICE Simulation
IR Generation
$i_{dd1}(t)$
$i_{dd2}(t)$
$i_{ddN}(t)$
Iso-IR Bands
Band$_i$ IR
Convolution/Superposition
Path $i_{DDT}(t)$

Figure 11

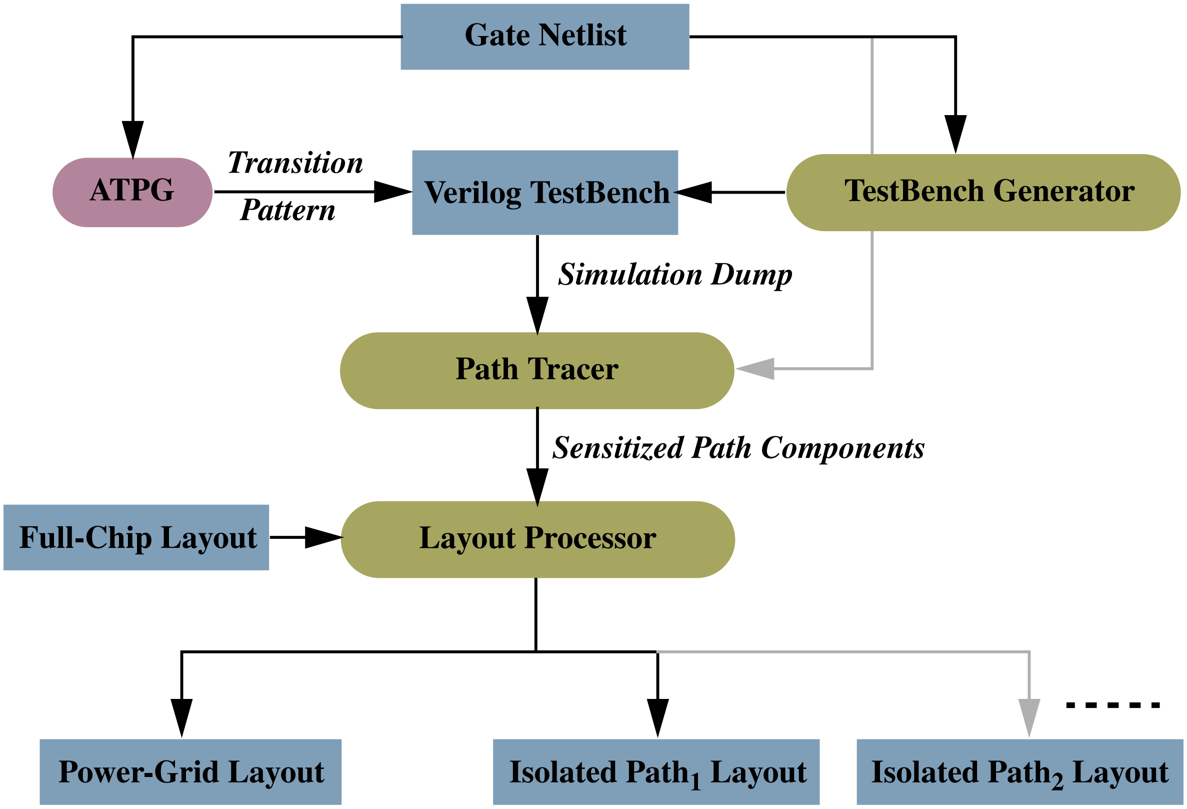

Gate Netlist
ATPG
Transition
Pattern
Verilog TestBench
TestBench Generator
Simulation Dump
Path Tracer
Sensitized Path Components
Full-Chip Layout
Layout Processor
Power-Grid Layout
Isolated Path$_1$ Layout
Isolated Path$_2$ Layout

Figure 12

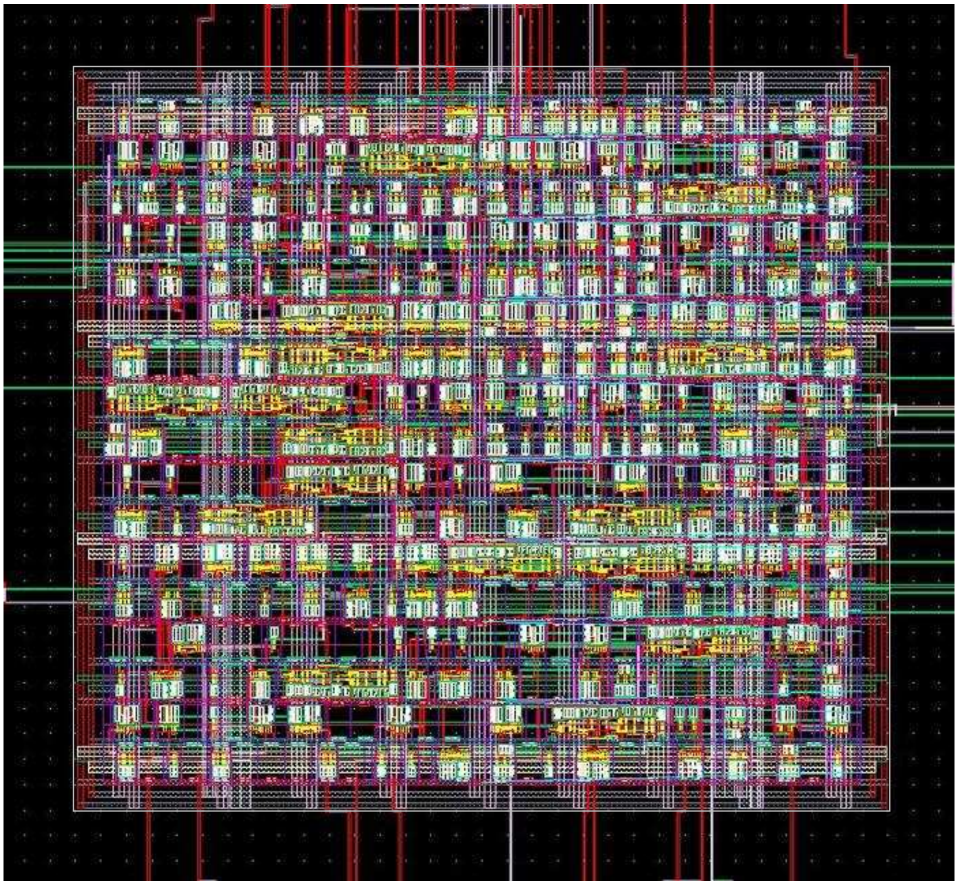


Figure 13

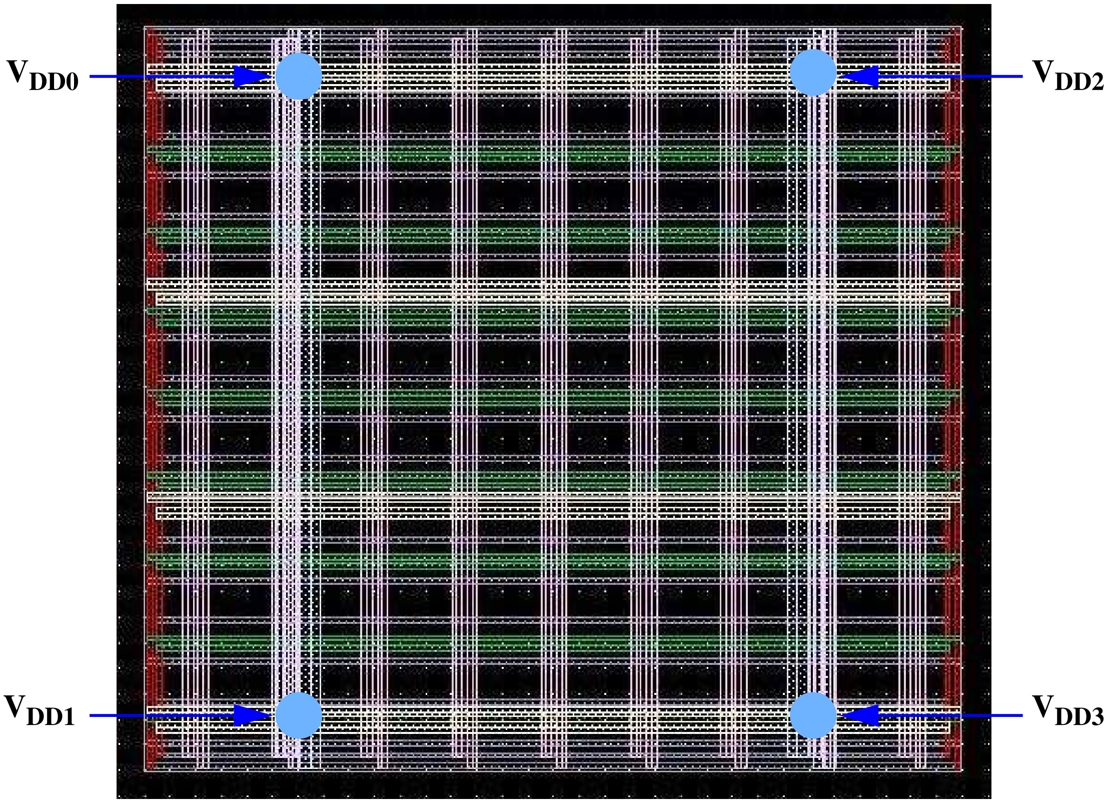


Figure 14

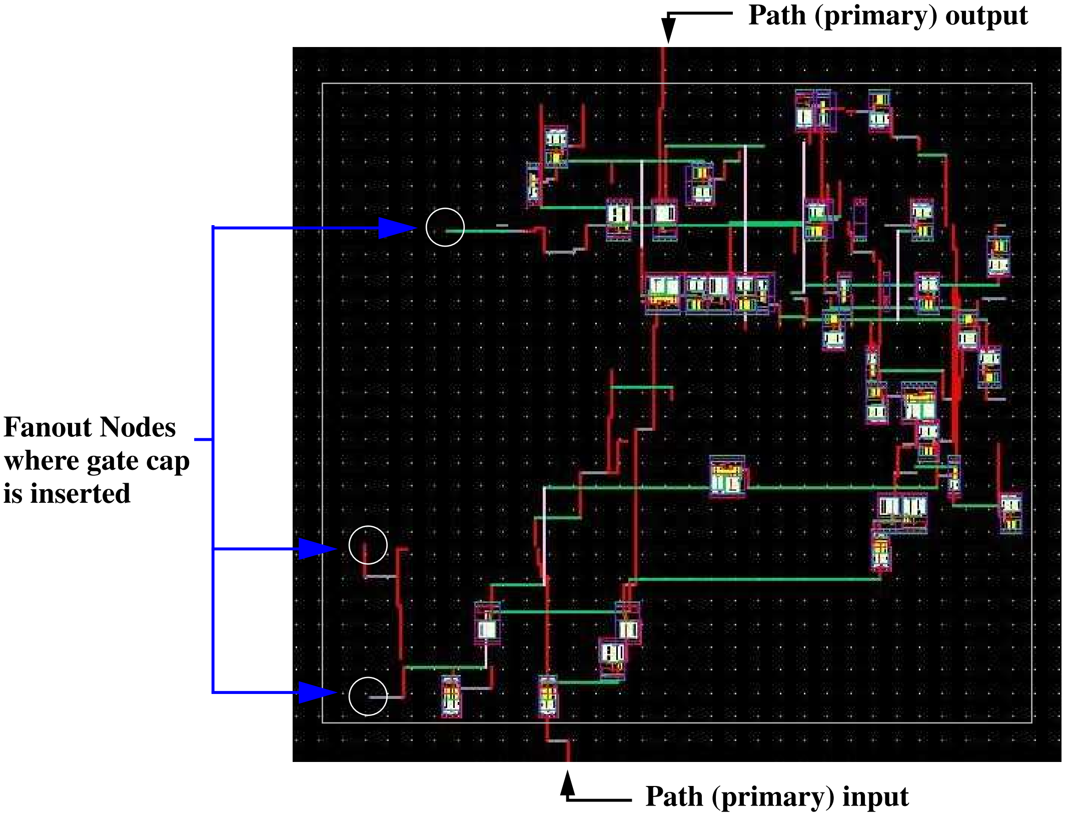

Path (primary) output
Fanout Nodes where gate cap is inserted
Path (primary) input

Figure 15

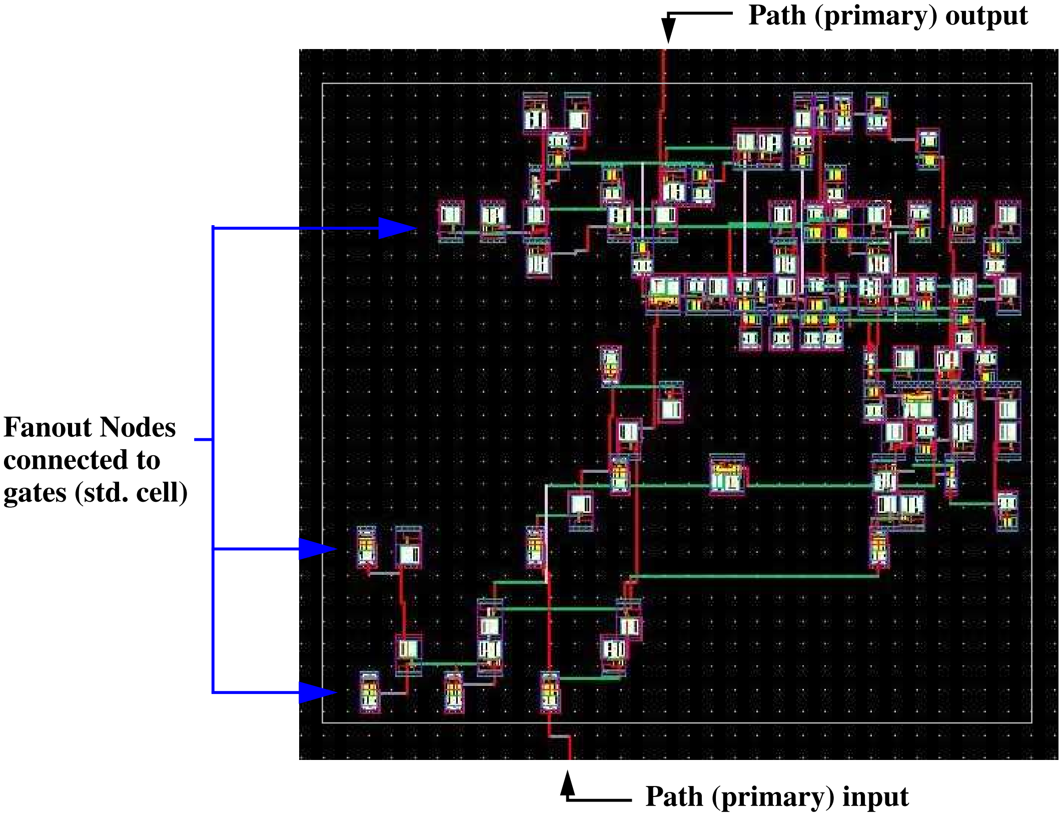
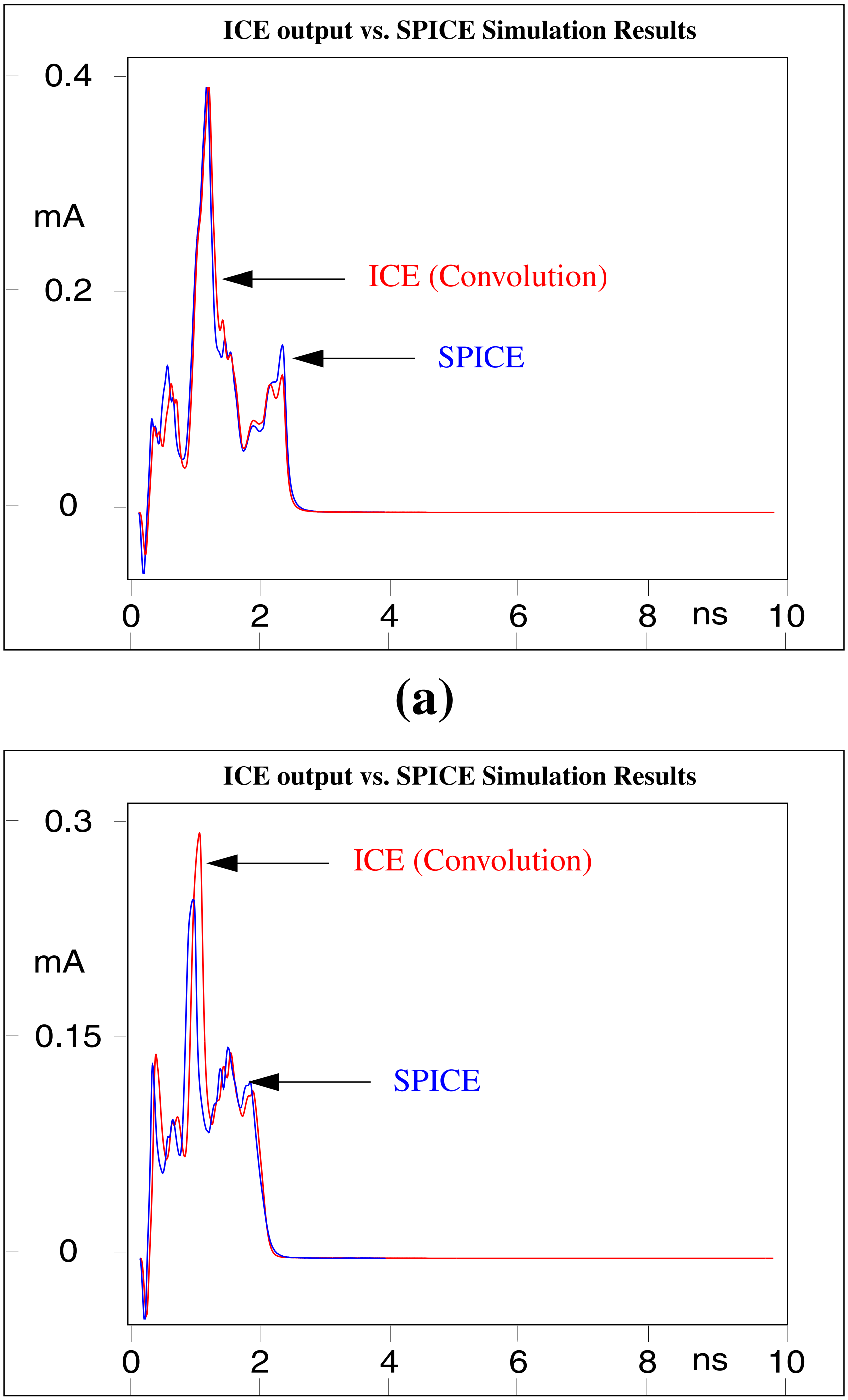

Path (primary) output
Fanout Nodes connected to gates (std. cell)
Path (primary) input

Figure 16


ICE output vs. SPICE Simulation Results
0.4
mA
0.2
0
ICE (Convolution)
SPICE
0
2
4
6
8
ns
10
ICE output vs. SPICE Simulation Results
0.3
mA
0.15
0
ICE (Convolution)
SPICE
0
2
4
6
8
ns
10

(a)

(b)

Figure 17