MIRID: **M**ixed-Mode **IR**-Drop **I**nduced **D**elay Simulator

J. Jiang
University of Passau
Faculty of Comp. Science & Mathematics
Innstr. 43 94032
Passau Germany
jie.jiang@uni-passau.de

M. Aparicio, M. Comte, F. Azaïs and M. Renovell
University of Montpellier
LIRMM
161 rue Ada 34095
Montpellier France
{name}@lirmm.fr

I. Polian
University of Passau
Faculty of Comp. Science & Mathematics
Innstr. 43 94032
Passau Germany
ilia.polian@uni-passau.de

***Abstract*—IR-drop effects are increasingly relevant in context of both design and test. We introduce the event-driven simulator MIRID that calculates the impact of IR-drop to the circuit timing. MIRID performs the simulation on two abstraction levels: timing effects in the gate-level net-list; current and voltage waveform propagation in the electrical model of the power-distribution network (PDN). Switching events at the logic gates are forwarded to the electrical model, where induced currents and their impact on the neighboring PDN nodes are computed. From this information, values of voltages at the Vdd and ground terminals of logic gates are determined, which in turn are used to calculate accurate switching delays of the gates. MIRID supports a generic interface to electrical models, allowing for a seamless integration of arbitrary models of PDN and gate timing. We report experiments based on a simple PDN model that was introduced previously and incorporates a pre-characterized library. The simulation accuracy is validated by matching the results from MIRID and SPICE.**

***Keywords*—Digital CMOS IC; Test; Power Noise; IR-drop; Simulation.**

I. Introduction

Power supply noise is becoming an important concern for VLSI system design in deep submicron regime [1]. It reduces the actual voltage levels supplied to the gates of the device, which has a detrimental impact on signal integrity. Excessive power supply noise can degrade circuit performance by inducing additional signal delay, or even lead to functional failure of logic gates [2]. Moreover, as technology scales, designs are becoming more sensitive to power supply noise due to reduced power supply voltage and increased power density [3].

Two major components are referred by power supply noise: the inductive voltage noise (Ldi/dt) that depends on the rate of change of the instantaneous currents flowing through parasitic inductive elements of the network; and the resistive voltage noise (IR) that depends on the instantaneous currents flowing through the resistive elements of the network [4]. Compared to package interconnect, PDN is predominantly resistive. Voltage fluctuations of power supply are mainly caused by IR noise. In this paper, we focus on the IR-drop effects in PDNs.

IR-drop has been studied with the purpose of reducing overall IR-drop effect as much as possible at chip level. A number of commercial tools (such as Redhawk [5], PrimeRail [6] and HyperLynx [7]) can do IR-drop analysis taking certain inputs in the design stage and targeting power network verification throughout physical implementation. Various supply network and circuit models are proposed for estimating power supply noise due to IR-drop [8]. Most of these works are based on a vectorless approach with the primary target of identifying critical areas to help designers by power network design. In other words, these commercial tools predict IR-Drop (using statistical models) but not the induced delay using a driven-event simulation. This is a classic design approach to the IR-Drop phenomenon.

PDN-related impact on delay is also of interest in context of testing. Testing of small-delay defects (SDD) in circuits requires an adequate fault coverage, which can be heavily affected by the process variation [9]. The nominal delay of gates are considered by ATPG for SDD testing while extra delay can be induced by noises in practice. Estimation of IR-drop induced delay is important by evaluation of test patterns for SDD, and even improve test compaction technology [10]. Power analysis is also essential for scan testing. Peak power consumption during test cycles of scan testing has to be controlled to avoid noise phenomena [11]. Accurate and yet efficient simulation of IR-drop-induced delays is required because electrical-level simulation is prohibitively expensive for large circuits and pattern counts.

We propose a mixed-mode simulation algorithm for IR-drop induced delays in the circuit. The gate-level event-driven simulator MIRID (Mixed-Mode IR-Drop Induced Delay Simulator) provides a compromise between the accuracy of SPICE and the efficiency of an IR-drop-unaware gate-level delay simulator and is applicable to academic and industrial benchmark circuits. The algorithm is applicable to any power network design which is encapsulated by a generic interface. We report experiments based on a pre-characterized library created for a resistive PDN design. We investigate both the accuracy of the IR-drop induced delay calculation and the runtime.

This paper is organized as follows. An overview of the simulation is given in Section II. The electrical pre-characterization is introduced in Section III. Section IV presents the simulation algorithm. Experimental results are given in section V. Finally section VI concludes the work.

II. Simulation Overview

As illustrated in Figure 1, the simulation engine *MIRID* performs an event-driven logic simulation for applied test vector *Patterns*. The IR-drop induced delays *Delays* are given as outputs by the engine. The simulator provides generic interfaces to a PDN model *PM* and a timing model *TM* for calculation of gate delays. Mapping from gates to power supply nodes is defined in advance according to the PDN topology and other electrical parameters in the PDN model.

Switching signals during the logic simulation are reported via the interface as events $S\text{-}event(s, t, v)$ where v is the new value assigned to s at time t. The voltage level of a power supply node n in VDD/GND network can be acquired by the simulator through the interface. These voltages are computed by the PDN model, which uses a different set of events, $D\text{-}events$, to capture the distribution of currents over the PDN.

Both S- and $D\text{-}events$ are sorted by the time t, and processed in order by the simulation engine. If the output s_{out} of gate g driven by s switches to a new value v', new event $S\text{-}event(s_{out}, t + \delta, v')$ will be generated where δ is the delay of g. The gate delay δ may differ from its nominal value due to the IR-drop effect. It is obtained via the interface to a timing model, which takes parameter values from the active simulation environment, such as voltage swings of the upstream gate and the switching gate, the load capacitance of the switching gate or other parameters according to the implementation of the timing model. In our experiment the interfaces between the models are built based on a pre-characterization library [12] containing pre-computed tables for gate delay and current draw induced by the switching activity, and the distribution of the current through the PDN structure as well.

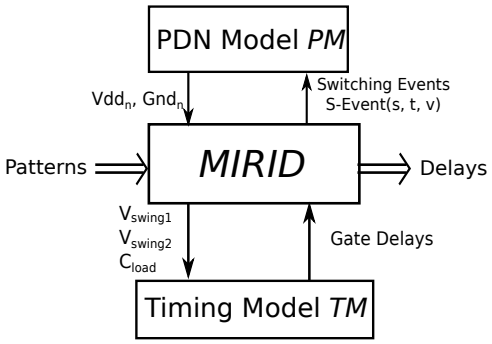


Fig. 1: Schematic illustration of the simulator

III. Electrical Models

As indicated above, the generic interfaces provided to the simulator contain functions of mapping the logic gates to the power supply nodes, determining the voltage levels on power supply nodes, and the gate delay induced by voltage drop on the gate. The implementation of these functions highly depends on the PDN configuration. In our experiments we consider the block under test (BUT) on the chip. Power and ground supply networks are modeled by grids with resistors in horizontal and vertical lines. Each gate is mapped to a grid node in the VDD/GND network.

The current drawn due to switching of the gates and its distribution through the resistors of the PDN are calculated and then used to compute the voltage drop at each node, using the Ohm's law. The current draw and the delay of a switching gate depend on the power supply levels, which are likely to be affected by IR-drop induced by neighboring gates, and by the load capacitance of the gate. Thus the computation of current draw and delay takes the gate environment into account, which is presented by three variable parameters as follows:

- **Edge**: The rising or falling input transition of the gate
- **Supply and input swings**: The voltage swing of the upstream gate (V_{swing1}) and the considered gate (V_{swing2}), defined by Equations (1) and (2) where $Vdd_1(t)$ and $Gnd_1(t)$ (resp. $Vdd_2(t)$ and $Gnd_2(t)$) are the power supply and ground supply levels of the upstream gate (resp. considered gate), as illustrated in Figure 2

$$V_{swing1}(t) = Vdd_1(t) - Gnd_1(t) \quad (1)$$

$$V_{swing2}(t) = Vdd_2(t) - Gnd_2(t) \quad (2)$$

- **Load capacitance**: Equivalent capacitance C_{load} of the downstream gates connected to the considered gate

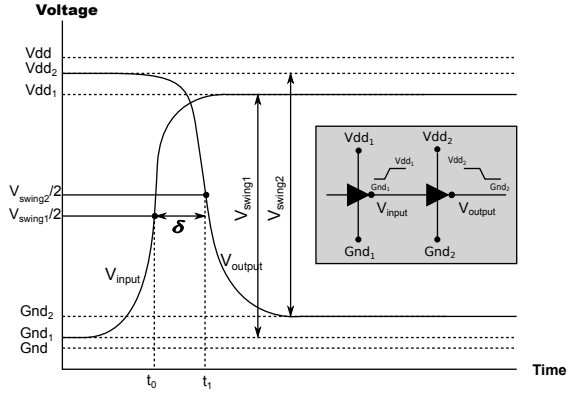


Fig. 2: Supply and input voltage swings and gate delay

Moreover, the implementation of the interface functions depends on the CMOS technology used by the circuits for the simulator. A pre-characterization of standard gates in terms of the current draw and the delay, as well as a pre-characterization of the current distribution in the PDN are needed.

SPICE simulations were performed for 45nm CMOS technology to pre-characterize current draw and delay of gates under all possible conditions which are likely to occur in a realistic environment. That is, all combinations of voltage swings between 100% and 80% of the nominal power supply swings for V_{swing1} and V_{swing2}, load capacitance values from one to five times the elementary equivalent capacitance and input transition edges (rising and falling) for all standard gates.

Section III-A and III-B detail the computation of gate delay and current draw, respectively. Section III-C gives the distribution model of current through PDN grids.

A. Gate Delay

The propagation delay δ is defined as the duration between the time t_0 at which the input signal of the gate crosses half of its excursion ($V_{swing1}/2$) and the time t_1 at which the output signal of the gate crosses half of its excursion ($V_{swing2}/2$), as illustrated in Figure 2 for an example of 2 inverters.

The variation of the delay in function of a single variable parameter (V_{swing1}, V_{swing2} or C_{load}) can be approximated by a linear function, when the other two parameters hold to their nominal values. The three variable parameters are independent, thus the delay function can be given by a polynomial of the first order for each variable parameter and for all combinations

of the variables. That is, a polynomial function of V_{swing1}, V_{swing2} and C_{load} with coefficients depending on the input edge (rising or falling) and the switching input of a gate. Sets of polynomial coefficients are provided in the library for all possible conditions.

B. Current Draw

The current drawn by a switching gate from the power and ground supplies is a time variant function. Rather than deriving an analytical expression for the current, we employ pre-calculated current waveforms, where each waveform is represented by an array of discrete values with picoseconds resolution. Different waveforms are required for different values of V_{swing1}, V_{swing2} and C_{load}. We pre-calculate the waveforms for a number of load capacitance values C_{load}, between one and five times the elementary equivalent capacitance, assuming $V_{swing1} = V_{swing2} = 100\%$. The currents for different voltage swing values are obtained by a linear transformation. The details can be found in our prior work [12].

Sets of reference current waveforms and corresponding shift/multiplying factors are saved for different gates, switching inputs and transition edges in the pre-characterization library. The current drawn by a switching gate from the VDD and GND supply networks can be obtained by modifying the appropriate reference current according to the environmental conditions.

C. Current Distribution in PDN Grid

The instantaneous current induced by the switching activity will be distributed in PDN resistors. The PDN grid is modeled by a resistive 100×100 power grid, each grid node serves as a power supply node for gates. Horizontal and vertical grid resistances can be predefined arbitrarily. For each node on the grid edge, different power supply voltages can be predefined as well. The current distribution model is established by simulating such a grid with a current source in the center, at the corner and in the band area of the grid.

In the first case, a current draw is modeled by a unitary current source in the center of the grid. A SPICE simulation is performed and the current values in the horizontal and vertical resistors of the grid are computed. As the current source is unitary, these values directly correspond to the desired distribution factor. The result shows that the current distribution is highly localized. Although the current spreads through all the resistors in the grid, its value is significant only through the resistors neighboring the current source. We define a sub-grid of $m \times n$ resistors outside which all currents are inferior to 1% and can thus be neglected. Although the current distribution depends on the position of the current draw in the grid, the same sub-grid can be used as distribution factor for all the nodes located in the central area of the PDN with an error inferior to 1%.

If the current is drawn from a node at the corner and in the band area of the PDN, which are defined as 10×10 nodes at each corner and bands with 10-nodes width between corners, respectively, dedicated distribution factors are computed. The error between the current distributions simulated with SPICE and computed using the pre-characterization factor is guaranteed to be less than 1%.

Obviously, the current draw distribution in the PDN grid also depends on the effective power supply at the extreme nodes at the border of the grid, which differ from the nominal value due to the activity of the neighboring blocks. We consider the globally static effect resulting from the average activity of the neighboring blocks. Static currents can be assumed to flow through the grid resistors in addition to the current draw distribution, which in turn presents the influence of the non-ideal power supply values at the extreme nodes on the border. More details of the library pre-characterization are given in [12].

IV. Simulation Algorithm

As given in Section II the simulation run is driven by switching events $S\text{-}event(s, t, v)$. We define another event $D\text{-}event(n, t, curr)$ with the object of a PDN node n to handle the current waveform $curr$ induced by the switching activity at time t through the PDN grids. Both events are defined as follows.

- $S\text{-}event(s, t, v)$ triggered when signal s changes its value to v at time t
- $D\text{-}event(n, t, curr)$ for a PDN node n with current $curr$ to be distributed in neighboring nodes at time t

As shown in Algorithm 1, the simulator takes the circuit netlist *C* and a set of test pattern pairs *P* as inputs, and provides interfaces to the electrical model of gate delay *TM* and the PDN model *PM*. In preprocessing phase each gate is mapped to a power supply node in VDD/GND network according to the PDN design. The simulation is run for each pattern pair (p_1, p_2) in *patterns* in two consecutive time frames. The first pattern is applied to get the initial value of each signal, i. e. the conventional logic simulation is run by calculating the logic value of each gate function in order of gate level. In the second time frame, the simulation is run for the second pattern taking the IR-drop effect into account.

The event queue EQ is initialized by a set of $S\text{-}event(s, 0, v)$ where s is a primary input and v is the newly assigned value by p_2 at time 0, v differs from its previous value assigned by p_1. Events in EQ are processed in increasing order of time. The variable t_{sim} refers to the time of event processed currently. It is initially assigned with zero. If the event in processing is a signal assignment event $S\text{-}event(s, t, v)$, s is assigned with the new value v, and marked as a *active signal*. The handling of all *active signals* is given in Algorithm 3. Otherwise e is the event for current distribution $D\text{-}event(n, t, curr)$, which is processed by distributing the current $curr$ to neighbors of n by a function implemented in the interface to the PDN model. In our model for experiments, the function $DistributeCurrent$ is implemented as shown in Algorithm 2.

In function $DistributeCurrent$, a corresponding matrix of dispersion factors $DF_{sx \times sy}$ is obtained from the library with respect to the position of the node in PDN: in the center, at the corner or in the band area as defined in Section III-C. The neighboring area of n is defined as a rectangular area of size $sx \times sy$ with n in the center. Each neighbor n' (including

Algorithm 1: IR-drop Simulation

Data: Circuit Netlist ***C***, Test Patterns ***P***, Timing Model ***TM***, PDN Model ***PM***
Result: Gate delay of each switching gate
Preprocessing:
begin
  Assign each gate to a VDD and a GND power supply node
begin
  foreach *pattern pair* $(p_1, p_2) \in P$ **do**
    Run good simulation for first pattern p_1
  /* Initialize event queue EQ with events $S\text{-}event(s, 0, v)$ for each input s changing its value to v at time 0 */
  InitializeEventQueue()
  $t_{sim} \longleftarrow 0$
  while EQ *not empty* **do**
    Event $e \longleftarrow$ first event in EQ
    $t_{sim} \longleftarrow$ time of e
    if e *is* $S\text{-}event(s, t, v)$ **then**
      $s \longleftarrow v$
      Mark s as *active signal*
    else
      /* e is $D\text{-}event(n, t, curr)$ */
      PM.DistributeCurrent(n, t, curr)
    Remove e from EQ
    $e_{next} \longleftarrow$ next event in EQ
    /* Process each *active signal*, if the time of e_{next} differs from t_{sim} */
    if *time of* $e_{next} \neq t_{sim}$ **then**
      foreach *active signal* s **do**
        ProcessActiveSignal(s)
        Unmark s as *active signal*

Algorithm 2: DistributeCurrent

input : PDN Model ***PM***, n, t, $curr$
begin
  $DF_{sx \times sy} \longleftarrow$ corresponding $sx \times sy$ dispersion factor matrix for n
  for $i = 1 \rightarrow sx$ **do**
    for $j = 1 \rightarrow sy$ **do**
      Neighbor $n' \longleftarrow$ node at position (i, j) in the $sx \times sy$ neighboring area
      $curr_{n'} \longleftarrow$ current waveform saved for node n';
      $fac_{i,j} \longleftarrow$ dispersion factor at position (i, j) in the matrix
      Update $curr_{n'}$ by adding current values in $curr$ multiplied with $fac_{i,j}$ to $curr_{n'}$ for time points $t, t+1, \ldots$

n itself) refers to a dispersion factor $fac_{i,j}$ in the matrix. The current waveform of n' is updated by adding the waveform of $curr$ scaled by $fac_{i,j}$, i.e. each discrete current value $curr_k$ in $curr$ is multiplied by $fac_{i,j}$ and added to the corresponding value in $curr_{n'}$ saved for time point $t+k$, where t is the event time and k is the index of $curr_k$ in $curr$ ($k = 0, 1, \ldots$).

After an event e is processed, it can be removed from the event queue EQ. If the time of next event in the queue differs from t_{sim}, all *active signals* are handled as given in Algorithm

Algorithm 3: ProcessActiveSignal(s)

input : Timing Model ***TM***, PDN Model ***PM***, s
begin
  upstream gate g_{pred}
  /* get power swing V_{swing1} for the upstream gate g_{pred} */
  $V_{swing1} \longleftarrow$ PM.GetPowerSwing(g_{pred}, t_{sim})
  foreach *gate* g *fed by* s **do**
    /* get power swing V_{swing2} for commuting gate g */
    $V_{swing2} \longleftarrow$ PM.GetPowerSwing(g, t_{sim})
    /* get induced current drawn from VDD and GND power supply nodes for g */
    $curr_{vdd} \longleftarrow$ PM.CalculateCurrent(n_{vdd}, V_{swing1}, V_{swing2}, C_{load})
    $curr_{gnd} \longleftarrow$ PM.CalculateCurrent(n_{gnd}, V_{swing1}, V_{swing2}, C_{load})
    /* create drop events */
    new $D\text{-}event(n_{vdd}, t_{sim}, curr_{vdd})$
    new $D\text{-}event(n_{gnd}, t_{sim}, curr_{gnd})$
    calculate gate output of g
    if *gate output* s_{out} *switches to* v' **then**
      /* calculate gate delay */
      $\delta \longleftarrow$ TM.CalculateGateDelay(V_{swing1}, V_{swing2}, C_{load})
      /* create signal assignment event for gate output) */
      new $S\text{-}event(s_{out}, t_{sim} + \delta, v')$
    insert new events into event queue
    sort event queue by the event time

3. For each *active signal* s the voltage swing V_{swing1} on the upstream gate g_{pred} is first calculated by $GetPowerSwing$ as defined in Equation 1 using the voltage values at corresponding power supply nodes. For the resistive PDN model used in the experiments, the voltage value at a power supply node can be obtained by applying the Ohm's law with given grid resistance, saved current in resistors and the voltage supply level at each extreme node on the PDN border.

For each gate g fed by s, the voltage swing V_{swing2} is then computed in the same way. The current induced by the switching input s is calculated by function $CalculateCurrent$ using V_{swing1}, V_{swing2} and the load capacitance C_{load} as inputs. Both functions $CalculateCurrent$ and $GetPowerSwing$ are defined in the interface to PDN model PM. Events $D\text{-}event(n_{vdd}, t_{sim}, curr_{vdd})$ and $D\text{-}event(n_{gnd}, t_{sim}, curr_{gnd})$ are then generated for current distribution to neighbors of power supply nodes n_{vdd} and n_{gnd} of g at time t_{sim}.

If the output s_{out} of gate g differs from its previous value, a new switching event $S\text{-}event(s_{out}, t_{sim} + \delta, v')$ is generated where the event time is given by the active simulation t_{sim} added by the gate delay δ and v' is the new value. The function of delay is provided in the interface to the timing model TM. In our model it is given by a polynomial with coefficients obtained from the pre-characterization library according to the simulation environment, as given in Section III-A. The parameters of the function are the voltage swings V_{swing1} and V_{swing2}, and the load capacitance of g as well.

All new events are inserted into EQ. After the insertion EQ is sorted by the event time again.

As given in Algorithm 1, after all *active signals* are handled as given above, they are unmarked. And the next event in EQ is to be processed in the same way. This iterative process continues until EQ is empty, i.e. all events in queue are processed and no new events are generated during the processing. The simulation flow ends then.

V. Experiments

The first experiment aims at validating the accuracy of the IR-drop induced delay calculated by MIRID by matching it with SPICE simulations of identical circuits. The simulation was performed to a chain of 14 inverters. As shown in Figure 3, each inverter is connected to a power supply node in 100×100 resistive PDN grid. Only the connected nodes with neighboring resistors in one network are drawn. Test pattern pair $(1, 0)$ is applied to the single primary input, the current and voltage curves in each PDN resistor from both simulation flows are compared to each other. It turns out that the results from both simulations match well to each other. The slight difference in current and voltage curves can be neglected in consideration of their impact on gate delay.

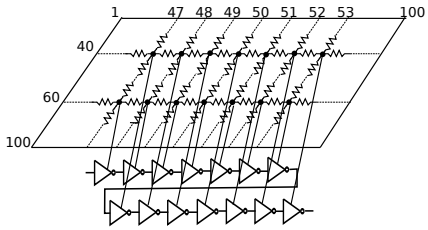
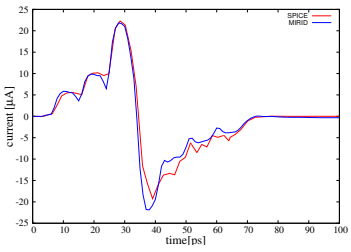
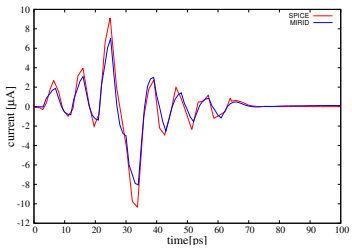


Fig. 3: 14-inverters model for validation

The current curves in resistor$(40, 50)$ in GND and VDD network from both simulations are given in Figure 4, while Figure 5 gives the comparisons of voltage curves.



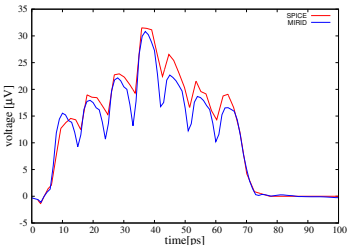
(a) Current curve in GND resistor$(40, 50)$



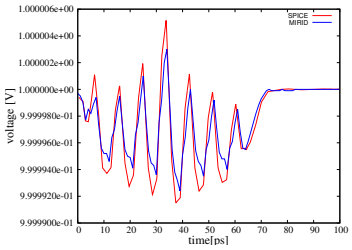
(b) Current curve in VDD resistor$(40, 50)$

Fig. 4: Current validation for 14-inverters chain

Another validation is done to the ISCAS circuit c17, which contains 2 INV, 3 NOR2 and 4 NAND2 gates. All gates are connected to a power supply node $(50, 50)$ in the PDN networks. For an applied test pattern the current and voltage curves at the node given by MIRID and SPICE are compared in Figure 6 and Figure 7, respectively. Difference in amplitude can be observed due to the fact that SPICE takes more electrical parameters into account while our simulator uses approximated electrical models of PDN and gate delay. The impact on the delay is reflected by the time at which the

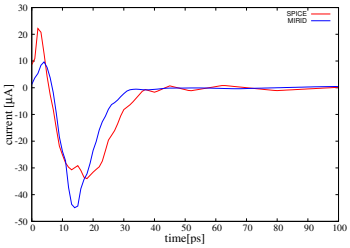


(a) Voltage curve in GND resistor$(40, 50)$

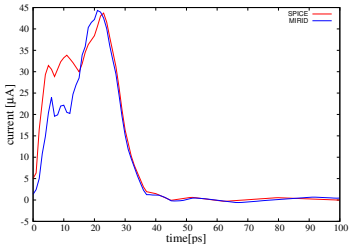


(b) Voltage curve in VDD resistor$(40, 50)$

Fig. 5: Voltage validation for 14-inverters chain

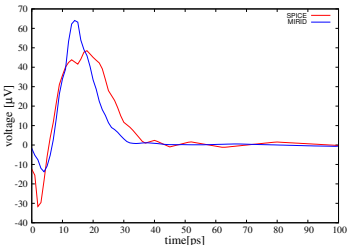


(a) Current curve in GND resistor$(50, 50)$

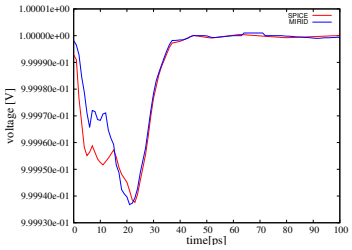


(b) Current curve in VDD resistor$(50, 50)$

Fig. 6: Current validation for c17



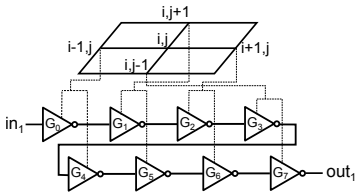
(a) Voltage curve in GND resistor$(50, 50)$



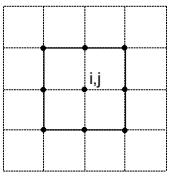
(b) Voltage curve in VDD resistor$(50, 50)$

Fig. 7: Voltage validation for c17

voltage curve gets back to its nominal value. It can be observed that the difference in delay from both simulation is very slight and thus acceptable.



(a) An 8-inverter chain placed in a 3×3 grid



(b) Placement of 9 inverter chains

Fig. 8: Model of 9 inverter chains

To investigate the correlation of the IR-drop induced delay

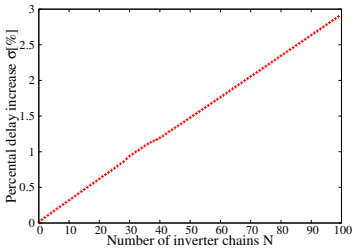


Fig. 9: Correlation of the IR-drop induced delay and N

and the switching activity, another experiment was performed to a dedicated design with maximal 100×9 inverter chains. Each chain comprises of 8 inverters $G_0, \ldots, G_7$ connected as follows: $in_1 \rightarrow G_0 \rightarrow G_1 \ldots \rightarrow G_7 \rightarrow out_1$ where in_1 and out_1 are primary input and output. The connection of the gates in a 3×3 PDN is illustrated in Figure 8a. G_0 and G_4 are connected to node $(i-1, j)$, G_1 and G_5 are connected to node $(i, j+1)$, G_2 and G_6 are connected to node $(i+1, j)$, G_3 and G_7 are connected to node $(i, j-1)$. Such a placement in a 3×3 grid is repeated 9 times in a 5×5 grid for 9 inverter chains. As shown in Figure 8b, a inverter chain is placed in the 3×3 area with the node (i, j) at the central position. All 9 chains are placed in areas with corresponding nodes in the center, which are marked by solid dots.

Simulations were performed to N copies of 9 inverter chains. (For $N = 2$, another set of 9 inverter chains is placed at the same position as the first one.) Figure 9 illustrates the ratio of the IR-drop induced delay in simulation runs for $N = 1 \ldots 100$. It can be seen that delay induced by IR-drop increases linearly with the value of N, which is proportional to the number of switching inverters.

The simulator was run for several ISCAS and NXP circuits with test pattern pairs for small delay faults generated by our in-house ATPG tool [9]. Rather than using full-custom layouts of the circuits, we created, for each simulated circuit, a PDN consisting of 100×100 nodes and connected each gate to the center of the PDN. The IR-drop induced delay is estimated by comparing the transition propagation time with and without IR-drop effect. The transition propagation time corresponds to the total delay of gates on the longest sensitized path.

As given in Table I, column 2 contains the number of gates, the number of test pattern pairs is given in column 3, column 4 presents the average transition propagation time under impact of IR-drop effect. The ratio of additional induced delay compared to the delay calculated using the nominal values is given in column 5. Column 6 gives the average number of switching activity during the simulation. The last column presents the average CPU runtime per test. As shown in the table, the IR-drop induced delay has a positive correlation to the number of switching activities, and so does the runtime. Though p35 has less gates than p45, it has a more complicated structure and thus more switching events are activated, which in turn induce longer propagation time. The IR-drop induced delay increase observed for the larger circuits matches well with values around 3% that have been previously reported in the literature [10].

TABLE I: Simulation for ISCAS and NXP circuits

circuit	#gates	#patterns	ave. trans. prop. time[ps]	σ [%]	ave. #switching	ave. CPU time [s]
c432	334	322	151.19	0.086	290	1
c1908	871	1000	197.78	0.88	1004	2
p35	27902	1000	573.35	5.33	19488	83
p45	30584	1000	383.87	3.75	3034	13

VI. Conclusion

This paper presents a simulation flow running across the logical and the electrical domains to simulate IR-drop induced delay. The logic simulation engine performs timing simulation and is driven by switching events as the input transitions are propagated through sensitized paths in the circuit. A PDN model and an electrical model for gate delays are used for simulating the IR-drop effect in electrical domain. Interfaces between the simulation engine and the models are provided for calculating the gate delay and updating electrical parameters in the PDN. The interfaces can be adapted to any PDN design and electrical models without influences to the simulation engine basically. Experiments conducted on a resistive PDN design show good match to SPICE results and scalability to industrial circuits within reasonable runtime. In the future we will concentrate on the integration of more complex PDN models into the simulator and the development of algorithmic speed-up techniques suited for these models.

Acknowledgment

This work was partially supported by the DFG (German National Science Foundation) grant Po 1220/1-2 and by the BFHZ (Bavarian-French University Center) grant FK-26-11.

References

[1] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," *Proceedings of IEEE ICCAD*, pp. 524–531, 1996.

[2] H. Chen and D. Ling, "Power supply noise analysis methodology for deep submicron VLSI design," *Proceedings of ACM/IEEE Design Automation Conf.*, pp. 638–643, 1997.

[3] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser, "Clock skew verification in the presence of IR-drop in the power distribution network," *IEEE Trans. on Computer-Aided Design, vol. 19, No. 6*, pp. 635–644, 2000.

[4] N. H. E. Weste and D. M. Harris, *CMOS VLSI Design*. Addison-Wesley, 2011.

[5] http://www.apache-da.com/products/redhawk.

[6] http://www.synopsys.com/Tools/Implementation/SignOff/Pages/PrimeRail.aspx.

[7] http://www.mentor.com/pcb/hyperlynx/.

[8] S. R. Nassif and O. Fakhouri, "Technology trends in power-grid-induced noise," *Proceedings of International Workshop on System-level Interconnect Prediction*, pp. 55–59, 2002.

[9] J. Jiang, M. Sauer, A. Czutro, B. Becker, and I. Polian, "On the optimality of K longest path generation algorithm under memory constraints," *Proceeding of IEEE Conf. on Design, Automation and Test in Europe*, pp. 418–423, 2012.

[10] Z. Jiang, Z. Wang, J. Wang, and D. M. H. Walker, "Levelized low cost delay test compaction considering IR-drop induced power supply noise," *IEEE VLSI Test Symposium (VTS)*, pp. 52–57, 2011.

[11] N. Badereddine, P. Girard, S. Pravossoudovitch, and H.-J. Wunderlich, "Structural-based power-aware assignment of don't cares for peak power reduction during scan testing," *Proceeding of IEEE IFIP International Conf.*, pp. 403–408, 2006.

[12] M. Aparicio, M. Comte, F. Azaïs, M. Renovell, J. Jiang, I. Polian, and B. Becker, "Pre-characterization procedure for a mixed mode simulation of IR-drop induced delays," *Proceeding of IEEE LATW*, 2013.