Application of a Stable Latency Insertion Method for Simulations of Power Distribution Networks

Kin Hang Tan, Patrick Goh, Mohd Fadzil Ain
School of Electrical and Electronic Engineering, Universiti Sains Malaysia,
14300 Nibong Tebal, Pulau Pinang, Malaysia.
eepatrick@usm.my

***Abstract*—This paper presents an application of a stable implementation of the latency insertion method for simulations of power distribution networks (PDN). Traditionally, simulations of PDNs poses a considerable challenge due to their large circuit sizes. While the latency insertion method can be applied to simulate these networks, the existence of low latency elements results in a more stringent stability criterion which reduces the efficiency of the method. Using the improved formulation, a latency insertion method that is free from the stability criteria is obtained, which results in no limitation on the size of the time step.**

***Index Terms*—Latency Insertion Method (LIM); On-Chip, Power Distribution Network.**

I. Introduction

Early design planning is crucial in any integrated circuit design as the noise margin becomes smaller and the operating frequency increases. It is important to perform power integrity simulation in the early design stages in order to prevent chip failures. With the trend towards deep submicron technology, state of the art interconnection feature size has also been reduced in tandem. The smaller wire spacing, together with longer wire length, higher operating speed, and smaller power supply voltage, have led to significant noise problems in on-chip power distribution networks (PDN). In particular, the power supply noise (PSN) which is caused by the parasitic inductances and capacitances, which traditionally only occurred on the packaging level, can no longer be ignored even at the chip level [1]. Thus it is important to be able to quickly and accurately simulate power distribution networks in order to design a PDN with a guaranteed PSN level within a specified margin.

The PDN of an integrated circuit is a distributed system that can be approximately modelled as a big RLGC mesh. Conventional simulation method such as the modified nodal analysis (MNA) based SPICE simulator, struggles to simulate the on-chip power grid consisting of a large number of elements as it requires a matrix inversion with huge memory requirements and excessive computation time, even by utilizing sparse matrix techniques. On the other hand, the latency insertion method (LIM), first presented in [2], is capable of solving the circuit equations in a leapfrog manner, resulting in a reduction in memory and computation time requirement. Because of its computational efficiency, LIM has been applied to solve a number of problems including power distribution networks [3-4]. However, due to its explicit formulation, LIM suffers from a conditional stability limitation [5]. This results in an inefficient simulation, especially when applied to on-chip simulations, with very small inductances and capacitances. In this paper, we present an application of a stable formulation of the latency insertion method for the simulations of power distribution networks. The stabilized formulation alleviates the stability limitation of LIM and is able to produce stable and accurate results even when the original LIM becomes unstable.

II. Review of Basic LIM Formulation

The formulation of the basic LIM is presented in this section. LIM can be applied to any arbitrary network whereby through the use of Thevenin and Norton transformations, the branches and nodes of the circuit are described by a general topology. Each node is denoted by a parallel combination of a current source, a conductance, and a capacitor to ground. A branch is formed by the connection between two different nodes where it is represented by a series combination of a voltage source, a resistor and an inductor. A node i with k branches connected to it is shown in Figure 1, while a branch connecting nodes i and j is shown in Figure 2. V_i represents the voltage at node i while I_{ij} represents the current flowing from node i to j. LIM discretizes the time variable whereby the voltages and currents are collated in half time steps. Then, the voltages are solved at half time steps whereas the currents are solved at full time steps. The algorithm starts from the Kirchhoff's current law (KCL) at node i:

$$C_i\left(\frac{V_i^{n+1/2}-V_i^{n-1/2}}{\Delta t}\right)+G_iV_i^{n-1/2}-H_i^n=-\sum_{k=1}^{M_i}I_{ik}^n \qquad (1)$$

where the superscript n is the index of the current time step, Δt is the time step and M_i is the number of branches connected to node i. Solving for the unknown voltage yields:

$$V_i^{n+1/2}=V_i^{n-1/2}+\frac{\Delta t}{C_i}\left(-\sum_{k=1}^{M_i}I_{ik}^n-G_iV_i^{n-1/2}+H_i^n\right) \qquad (2)$$

for $i = 1, 2, \ldots, N_n$, where N_n is the number of nodes in the circuit.

Then, writing Kirchhoff's voltage law (KVL) at branch ij yields:

$$V_i^{n+1/2}-V_j^{n+1/2}=L_{ij}\left(\frac{I_{ij}^{n+1}-I_{ij}^n}{\Delta t}\right)+R_{ij}I_{ij}^n-E_{ij}^{n+1/2} \qquad (3)$$

and solving for the unknown current yields:

$$I_{ij}^{n+1} = I_{ij}^{n} + \frac{\Delta t}{L_{ij}}\left(V_i^{n+1/2} - V_j^{n+1/2} - R_{ij} I_{ij}^{n} + E_{ij}^{n+1/2}\right). \quad (4)$$

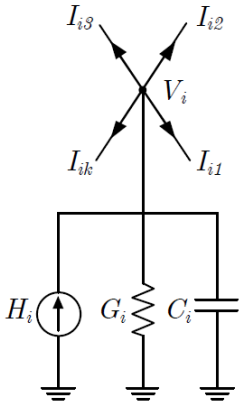

Figure 1: LIM node equivalent circuit

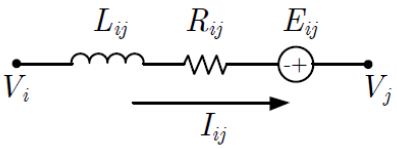

Figure 2: LIM branch equivalent circuit

As time progresses, the calculation of the node voltages and the branch currents are alternated in a leapfrog manner. LIM is similar to Yee's algorithm for the solution of Maxwell's equations in the finite-difference time-domain (FDTD) method in this context [6]. It is well known that the LIM algorithm depends on the latencies in the network in order to perform the leapfrog time stepping formulation. Hence, a capacitor to ground has to be present at every node. If not, a small fictitious capacitor is inserted to enable the method. Likewise, small fictitious inductors are introduced into branches without latencies. Similar to the traditional FDTD method, LIM is only conditionally stable which means that there is an upper bound on the time step that will result in a numerically stable solution to equations (2) and (4). To be precise, the maximum time step size of LIM, Δt_{max}, has the following upper bound [5]:

$$\Delta t_{max} < \sqrt{2} \min_{i=1}^{N_n}\left(\sqrt{\frac{C_i}{M_i}} \min_{p=1}^{M_i}\left(L_{i,p}\right)\right) \quad (5)$$

where $L_{i,p}$ denotes the value of the pth inductor connected to node i. Further analysis on the stability of LIM can be found in [7], [8].

III. Stable LIM Formulation

In order to circumvent the limitation of the stability condition in LIM, a stable reformulation of LIM is used [9]. In the basic LIM formulation, each node and branch was updated purely based on its previous time step value. The $(n+1)$th voltage will be first calculated using the (n)th step element and the $(n+1)$th current will then be calculated using the just calculated $(n+1)$th voltage. In this method, calculating the $(n+1)$th step for a "voltage-current" set will be considered as a complete cycle and the same process will be repeated for the next cycle. This is illustrated in Figure 3 for the case of a single line of elements.

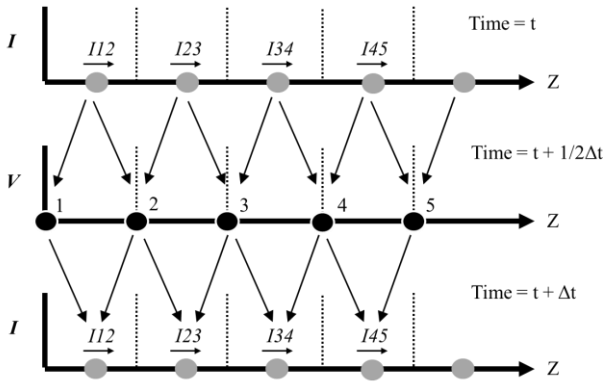

Figure 3: Leapfrog concept for LIM

In order to overcome the stability limitation, an implicit formulation is first considered. However, updating both the $(n+1)$th step voltages and currents using $(n+1)$th step elements involve heavy substitutions within the algorithm loop, and the complexity increases non-linearly with the size of the circuit.

Hence, a reformulated LIM has been proposed using a mixture of explicit and implicit formulation. First a choice is made to either solve all the currents first by substituting in the voltages, or to solve all the voltages first by substituting in the currents. Calculating the current first simplifies the process to some extent as each branch is always formed by the connection of two nodes, but each node could have a varying number of branches connected to it.

Consider the implicit formulation of the voltages and currents in Figures 1 and 2 given below:

$$V_i^{n+1} = \left(\frac{C_i}{\Delta t} + G_i\right)^{-1}\left(\frac{C_i}{\Delta t} V_i^{n} - \sum_{k=1}^{M_i} I_{ik}^{n+1} + H_i^{n+1}\right) \quad (6)$$

$$I_{ij}^{n+1} = \left(\frac{L_{ij}}{\Delta t} + R_{ij}\right)^{-1}\left(\frac{L_{ij}}{\Delta t} I_{ij}^{n} + V_i^{n+1} - V_j^{n+1} + E_{ij}^{n+1}\right). \quad (7)$$

Substituting all the voltages V_i^{n+1} and V_j^{n+1} from equation (6) into equation (7) then gives:

$$I_{ij}^{n+1} = \frac{\frac{L_{ij}}{\Delta t} I_{ij}^{n} + \frac{\frac{C_i}{\Delta t} V_i^{n} - \sum_{k=1}^{M_i} I_{ik}^{n+1} + H_i^{n+1}}{\frac{C_i}{\Delta t} + G_i} - \frac{\frac{C_j}{\Delta t} V_j^{n} - \sum_{k=1}^{M_j} I_{jk}^{n+1} + H_j^{n+1}}{\frac{C_j}{\Delta t} + G_j} + E_{ij}^{n+1}}{\frac{L_{ij}}{\Delta t} + R_{ij}}. \quad (8)$$

The substituted equation (8) merges all the node and branch calculation into a single formulation. Next, the I_{ij}^{n+1} that is exactly the same with the calculated I_{ij}^{n+1} that exists in both the V_i^{n+1} and V_j^{n+1} substitutions, are pulled out from the summation:

$$I_{ij}^{n+1} = \frac{\frac{L_{ij}}{\Delta t} I_{ij}^{n} + \frac{\frac{C_i}{\Delta t} V_i^{n} - I_{ij}^{n+1} - \sum_{k=1,k\neq j}^{M_i} I_{ik}^{n+1} + H_i^{n+1}}{\frac{C_i}{\Delta t} + G_i} - \frac{\frac{C_j}{\Delta t} V_j^{n} + I_{ij}^{n+1} - \sum_{k=1,k\neq i}^{M_j} I_{jk}^{n+1} + H_j^{n+1}}{\frac{C_j}{\Delta t} + G_j} + E_{ij}^{n+1}}{\frac{L_{ij}}{\Delta t} + R_{ij}} \quad (9)$$

All the I_{ij}^{n+1} are then grouped together on the left hand side and solved to create the final formula for the current:

$$I_{ij}^{n+1} = \frac{\frac{L_{ij}}{\Delta t} I_{ij}^{n} + \frac{\frac{C_i}{\Delta t} V_i^n - \sum_{k=1,k\neq j}^{M_i} I_{ik}^{n+1} + H_i^{n+1}}{\frac{C_i}{\Delta t} + G_i} - \frac{\frac{C_j}{\Delta t} V_j^n - \sum_{k=1,k\neq i}^{M_j} I_{jk}^{n+1} + H_j^{n+1}}{\frac{C_j}{\Delta t} + G_j} + E_{ij}^{n+1}}{\frac{L_{ij}}{\Delta t} + R_{ij} + \left(\frac{C_i}{\Delta t} + G_i\right)^{-1} + \left(\frac{C_j}{\Delta t} + G_j\right)^{-1}} \quad (10)$$

In this case, the next time step must actually be calculated simultaneously for all the currents in the circuit. In order to avoid the complexity of an implicit solution, a combination of explicit and implicit substitution is presented here.

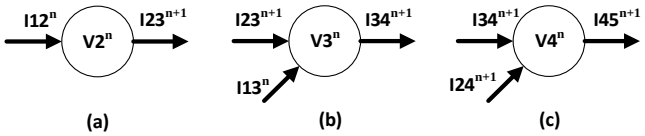

Figure 4: Node updating process in sequence (a) fully explicit updating (b) explicit and implicit updating and (c) fully implicit updating.

Consider Figure 4, where each circle indicates a node and the arrows on the left represent the currents entering the node while the arrows on the right represent the currents exiting the node. Figure 4(a) shows an explicit integration because $I23^{n+1}$ is calculated using $I12^{n}$. However, within the same cycle of the branch updating process, once $I23^{n+1}$ has been obtained, it can actually be directly used for the next calculation. Figure 4(b) shows that $I34^{n+1}$ is taking $I23^{n+1}$ and $I12^{n}$ as input values for its calculation, combining explicit and implicit integration. Lastly, Figure 4(c) shows a fully implicit integration since $I45^{n+1}$ is entirely dependent on $(n+1)$th input values.

Similar to other FDTD or heat transfer algorithms, one needs to start the calculation from a predefined starting point. All the calculated points will then either be updated from left to right, top to bottom, inner to outer or vice versa. In our method, the branch calculation starts with a purely explicit integration at the first branch as in Figure 4(a) and slowly progresses to a mixed explicit and implicit integration as in Figure 4(b) in subsequent branches. Finally, the end branches will involve only implicit integrations as in Figure 4(c).

Once the branch currents have been calculated, the node voltages can then be updated by using the newly solved values, I_{ij}^{n+1} through equation (6). This completes the solution.

The mixed implicit-explicit solution for equations (10) and (6) is summarized in the pseudo-code below.

Pseudo-code for the mixed implicit-explicit solution of LIM:

```
Begin transient solution:
For time = n+1
Update branch ij according to (10)
        if (all other branches connected to ij have
        values at t=n+1)
                solve (10) using values at n+1;

        else if (some branches connected to ij have
        values at t=n+1)
                solve (10) using a mixture of values at n
                and n+1;

        else (no branch connected to ij have values at
        t=n+1)
                solve (10) using values at n;

        end if
Next branch;
Update node i according to (6);
Next node;
Next time;
end transient solution;
```

IV. Numerical Results

In this section, two numerical examples are presented by applying the basic LIM and stable LIM formulations for the simulation of PDNs.

A. Basic PDN Block

In this case, a simple PDN block circuit is simulated. The circuit consists of three nodes on the top layer and three nodes on the bottom layer which are connected by a single coupling branch as shown in Figure 5. This circuit represents a building block that can be used to construct larger PDN models. Nodes 1 to 3 represent a power line while nodes 4 to 6 represent a ground line. R_m and L_m are fictitious elements while C_m is the coupling capacitance that exists between the two lines. We note that this circuit contains a branch capacitor which can be handled using the companion model similar to that in [10]. We note also that the values of the elements across branch 2-5 are much smaller than the rest of the circuit which would cause a limitation on the time step size used in LIM. A simulation using the basic formulation of LIM with a time step size smaller than the stability limit will serve as a benchmarking result in this comparison.

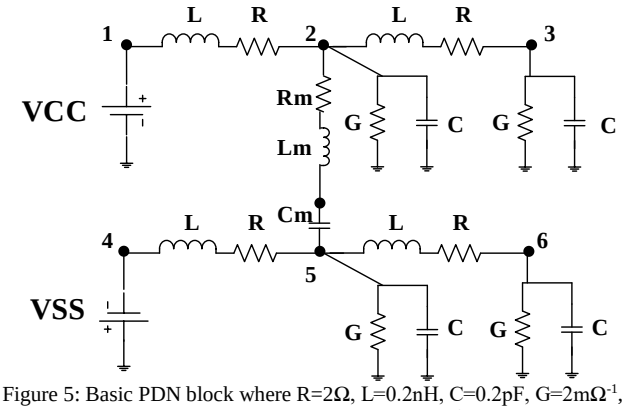

Figure 5: Basic PDN block where R=2Ω, L=0.2nH, C=0.2pF, G=2mΩ^{-1}, R_m=0.2mΩ, L_m=0.2pH and C_m=0.2fF

The total simulation time is 1 ns while VCC carries a voltage of 1 V and VSS carries a voltage of 0 V. Voltage waveforms at node 6 will be used for comparison. Figure 6 shows the output from both methods as the voltage at node 1is switched from 0V to 1V. We see that both methods are able to predict the switching noise induced in this circuit. However, in order to obtain a stable simulation, LIM required a time step of 0.01 ps, and a total of 100,000 simulation points.

Next, the simulation is repeated with a time step size of 0.1ps, which is 10x larger compared to the previous value. Figure 7 shows the result from both methods. We see that the basic LIM is no longer stable while the reformulated LIM is still able to retain a stable and accurate solution as compared to the results in Figure 6. We remark that results tested using larger time steps indicate that the reformulated LIM is unconditionally stable, where stable result is obtained regardless of the choice of the time step.

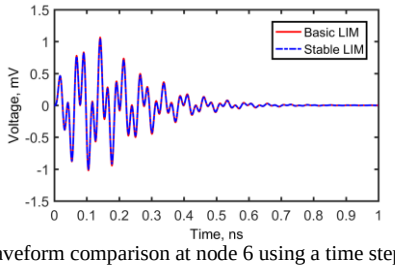

Figure 6: Waveform comparison at node 6 using a time step size of 0.01ps

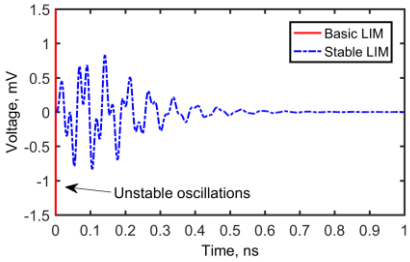

Figure 7: Waveform comparison at node 6 using a time step size of 0.1ps

B. PDN Model Circuit

In this case, a larger PDN circuit model as shown in Figure 8 is simulated. All the nodes in the circuit are connected to form a distributed circuit with uniform RLGC parameters where R = 20 Ω, L = 0.2 nH, G = 0.02 Ω^{-1}, and C = 0.2 pF. The top layer will serve as the power source layer with one side supply voltage, VCC while the bottom layer is supplied by the VSS voltage source. The red colored lines denote VCC branches while the blue colored lines denote VSS branches. The yellow box denotes the related elements for the coupling between the two layers, where R = 20 Ω, L = 0.02 nH, and C = 0.02 pF. The current sources represent the switching elements which are connected to the PDN and are modelled by a triangular current waveform with a maximum value of 1 mA and a minimum constant value of 0.1 mA to model the leakage current through the device. The VCC sources are 1 V while the VSS sources are 0 V. The simulations will include an initial switching-on of the power supplies from 0 V to their VCC and VSS values at 0 ns, and also a switching current at 0.2 ns.

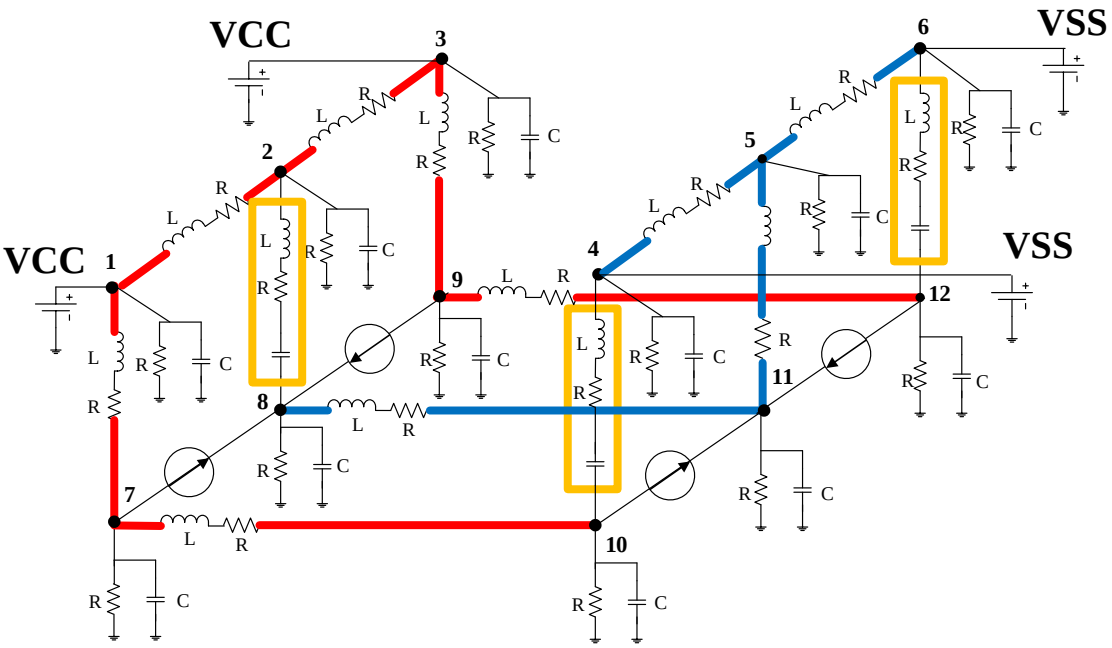

Figure 8: PDN model circuit

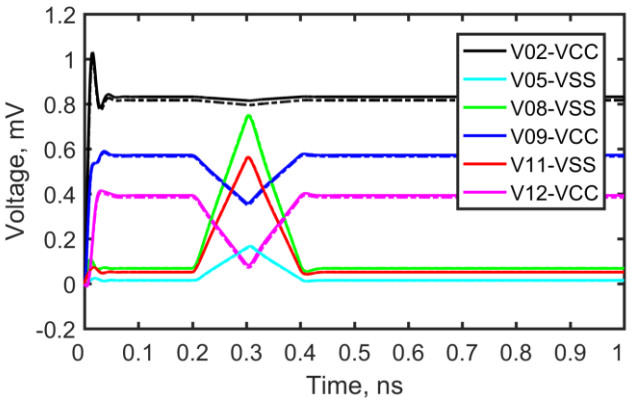

Figure 9: Waveform result of PDN model circuit with Δt = 0.1 ps. (Solid line is Basic LIM; dotted line is Stable LIM.)

Figure 9 shows the simulation result for the voltages at a few selected nodes. The solid lines indicate the results from the basic LIM method while the dotted lines indicate the results from the reformulated stable LIM. The time step used in this case is 0.1 ps which is within the stability limit of LIM. We see that both methods produce similar results. Next the simulation is repeated with a time step size of 1 ps. This is shown in Figure 10. In this case, the time step size is larger than the stability limit of LIM and the basic LIM becomes unstable and is unable to produce any result. On the other hand, the reformulated LIM remains stable and the result can be seen in the figure. Some discrepancy are observed in this case due to the accuracy degradation of the finite difference formulation when the step size is too large. The solution of this will be a focus of future work by using a higher order integration method.

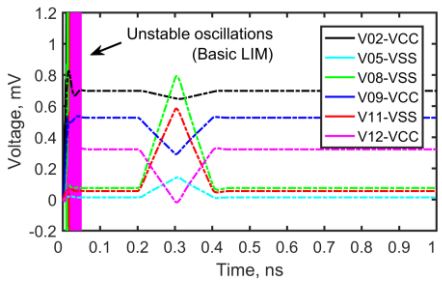

Figure 10: Waveform result of PDN model circuit with Δt = 1 ps. (Solid line is Basic LIM; dotted line is Stable LIM.)

V. CONCLUSION

In this work, an improved stable formulation of LIM has been applied to the simulation of PDNs. This method is able to overcome the stability limitation, in which the time step size is normally determined by the smallest capacitor and inductor in the circuit, thus allowing the use of larger time steps compared to the normal LIM. By using larger time steps, the overall computational time can be reduced. Future work will focus on improving the accuracy of the method by using a higher order integration method.

ACKNOWLEDGMENT

This work is supported by the Malaysian Ministry of Higher Education Fundamental Research Grant Scheme (FRGS) grant no. 203/PELECT/6071246.

REFERENCES

[1] E. P. Li. et al., "Progress review of electromagnetic compatibility analysis technologies for packages, printed circuit boards, and novel interconnects," *IEEE Trans. Electromagn. Compat.*, vol. 52, no. 2, pp. 248–265, May 2010.

[2] J. E. Schutt-Aine, "Latency insertion method (LIM) for the fast transient simulation of large networks," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 48, no. 1, pp. 81–89, Jan. 2001.

[3] S. N. Lalgudi, M. Swaminathan, and Y. Kretchmer, "On-chip power-grid simulation using latency insertion method," *IEEE Trans. Circuits Syst. I*, vol. 55, no. 3, pp. 914–931, Apr. 2008.

[4] T. Sekine and H. Asai, "Block latency insertion method (block-LIM) for fast transient simulation of tightly coupled transmission lines," *IEEE Trans. Electromagn. Compat.*, vol. 53, no. 2, pp. 193–201, Feb. 2011.

[5] S. N. Lalgudi and M. Swaminathan, "Analytical stability condition of the latency insertion method for nonuniform GLC circuits," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 9, pp. 937–941, Sep. 2008.

[6] K. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," *IEEE Trans. Antennas Propagat.*, vol. 14, no. 3, pp. 302–307, May 1966.

[7] P. Goh, J. E. Schutt-Aine, D. Klokotov, J. Tan, P. Liu, W. Dai, and F. Al-Hawari, "Partitioned latency insertion method with a generalized stability criteria," *IEEE Trans. Compon., Packag., Manuf. Technol.*, vol. 1, no. 9, pp. 1447–1455, Sep. 2011.

[8] J. E. Schutt-Aine, "Stability analysis of the latency insertion method using a block matrix formulation," in Proc. *IEEE Electr. Des. Adv. Packag. Syst. Symp.*, Seoul, Korea, Dec. 2008, pp. 155–158.

[9] K. H. Tan, P. Goh, and M. F. Ain, "A voltage-in-current formulation for the latency insertion method for improved stability," *Electron. Lett.*, in press.

[10] J. Choi, M. Swaminathan, N. Do and R. Master, "Modeling of power supply noise in large chips using the circuit-based finite-difference time-domain method," *IEEE Trans. Electromagn. Compat.*, vol. 47, no. 3, pp. 424–439, Aug. 2005.