A Multicore Chip Load Model for PDN Analysis Considering Voltage–Current-Timing Interdependency and Operation Mode Transitions

Jun Chen, Hajime Kando, Toshiki Kanamoto, *Member, IEEE*, Cheng Zhuo, *Senior Member, IEEE*, and Masanori Hashimoto, *Senior Member, IEEE*

***Abstract*— Power consumption and current fluctuation are continuously increasing in modern multicore systems. Such current flow may cause severe supply noise via off-chip and on-chip power delivery networks (PDN). Unexpected noise impacts the chip delay performance or even causes malfunction. In traditional practice, PDN designers assume a simple current source-based chip load, but it is often oversimplified, where the load current is modeled only for one or a few operation modes, and it is constant irrelevant to supply noise. In this paper, we propose a new chip load model that enables even off-chip PDN designers to assess the noise impact on circuit performance and use realistic current profile under supply voltage noise. We also integrate a control signal interface so that the model can switch the processor operation modes for finding unexpected noise behavior in design time and pursuing robust PDN design. Experimental results show that the proposed model mostly described by Verilog-A reproduces the current profile, current peak, and timing data well even while it achieves over 300× run-time reduction compared to a transistor-level model. We also experimentally demonstrate that a land-side capacitor is helpful to improve processor timing performance in our test case.**

***Index Terms*— Chip load model, multicore system, operation mode transition, power delivery network, power supply noise, voltage–current-timing interdependency.**

I. Introduction

MULTICORE systems have become popular for pursuing higher performance and energy efficient computing. In a multicore system, various noise sources exist at both off-chip and on-chip sides, such as power delivery network (PDN) resonance, voltage regulator at the off-chip side, individual core activities variation, and interference among adjacent cores of the on-chip side [1]. These noise sources require a PDN to be carefully designed in consideration of both PDN parameters and multicore operation status so that the chip functionality and performance are assured.

Traditionally, an allowable maximum voltage drop, e.g., 10% of nominal voltage, is given to PDN designers as a design guard bound, and designers believe the chip functionality and timing will be ensured as long as the worst supply noise stays within the given guard bound. One of the reasons for this situation is that there is no way for PDN designers to assess the noise impact on timing and PDN-timing interaction. However, with the scaling down of the technology node, timing sensitivity to noise becomes more and more severe. Saint-Laurent and Swaminathan [2] reported over 8% timing impact under supply voltage noise after the 90-nm technology node. The supply noise is thought to become more severe under the even smaller node. At the 55-nm node, the peak supply noise can reach 20%–30% of nominal voltage [3]. Thus, PDN design relying on voltage guard bound becomes very difficult.

A multicore system makes the noise problem even more challenging and complex. Taking the worst voltage droop as an example, a dual-core system may experience 50% larger droop than single-core system [1]. On the other hand, in a multicore system, the worst case voltage droop supposed in the design time tends to be pessimistic, and consequently, voltage guard bound-based methodology is inefficient for PDN design. Meanwhile, the on-chip timing impact needs to be evaluated over various intercore activation scenarios since the timing is the primary metric in digital chip design. Suppose simultaneous activity variation arises on several cores, for example, power-on or wake-up, then a significant voltage droop is induced, and it propagates to adjacent cores. Such a droop may reach 150 mV and may exceed the voltage guard bound [1], [4]. If a signal is propagating on a critical path in the victim core, the voltage droop causes extra path delay, which may result in malfunction [5]. The intercore noise-timing impact is even more severe if the core is located far from power supply ports, such as at the center of shared power and ground (PG) mesh [6]. In another scenario, if adjacent cores stay in retention mode or idle mode, the parasitic capacitance in those cores can be used to mitigate the noise and consequential timing impact.

To help PDN designers to assess the noise-timing impact on the multicore system, this paper proposes a new chip load

Manuscript received April 5, 2019; accepted April 19, 2019. Date of publication April 25, 2019; date of current version September 26, 2019. Recommended for publication by Associate Editor M. G. Telescu upon evaluation of reviewers' comments. *(Corresponding author: Jun Chen.)*

J. Chen and M. Hashimoto are with the Department of Information Systems Engineering, Osaka University, Osaka 565-0871, Japan (e-mail: j-chen@ist.osaka-u.ac.jp; hasimoto@ist.osaka-u.ac.jp).

H. Kando is with Murata Manufacturing Company, Ltd., Kyoto 617-8555, Japan.

T. Kanamoto is with the Department of Electronics and Information Technology, Hirosaki University, Aomori 036-8561, Japan (e-mail: kanamoto@eit.hirosaki-u.ac.jp).

C. Zhuo is with the College of Information Science and Electronics Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: czhuo@zju.edu.cn).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCPMT.2019.2913202

model that can provide the on-chip timing information, replay detailed voltage-dependent current profile, and extensively explore the intercore operation mode variation with a short runtime.

A. Related Work on Chip Load Model

Revealing the chip timing and detailed current–voltage profile can help PDN designers to find potential design issues and avoid overdesign/underdesign. The on-chip measurement module is widely applied in the postsilicon validation stage. Modules such as on-chip sensors and critical path replica are developed to measure chip internal timing information [1], [3], [7]–[11]. The inherent limitation of the postsilicon methodology is the silicon resource cost and the difficulty in design modification due to the late feedback.

On the other hand, the presilicon simulation requires no silicon resource and provides feedback in design time. For performing the simulation, a chip load model that represents the chip behavior from the point of view from load current is necessary. The chip load model that consists of on-chip PDN model and full transistor-level switching circuit model can replay the on-chip behavior with high accuracy. However, even a very short period run takes days or even months to finish. Extensive PDN design exploration is infeasible.

For reducing the computational cost for a chip load model, the switching circuit is often modeled by a current source [12], [13] or equivalent *RC* circuit models [14]–[16]. The current source model is usually described with a current profile in a piecewise linear format. Once a current profile is obtained under a given supply voltage, these piecewise linear current values are irrelevant to supply voltage variation. Hence, a large simulation error is introduced when the actual supply voltage has a significant dynamic supply noise. The current source can be also modeled by voltage-controlled-current-source (VCCS) to take into account the dependence of current on voltage. However, VCCS relies on instant voltage–current scaling, which is not suitable for replaying temporal behavior.

On the other hand, the *RC* circuit model can roughly model the voltage–current interdependency. This modeling method uses variant resistors, typically implemented by VCCS, to mimic the equivalent resistance of ON- and OFF-state transistors. Then, parasitic capacitors are characterized to mimic cell transition delay. However, even with careful characterizing effort on *RC* parameters, the oversimplified *RC* model is difficult to replay for a detailed current profile for large-scale circuit operation.

B. Challenge and Contribution

The main challenge to the traditional chip load model is to consider voltage–current-timing interdependency. Such interdependency can be demonstrated in Fig. 1. In actual circuits, the supply noise affects chip timing performance such as clock latency and path delay [17], [18]. When supply voltage drops, signal propagation is delayed, clock latency gets longer, and transistor switching current becomes smoother and smaller. When the load current becomes smaller due to the supply voltage drop, the dynamic noise becomes smaller, and its impact is naturally mitigated. Then, the simplified models such as the current source model in the piecewise linear format are irrelevant to voltage variance, and hence, the supply noise is likely to be overestimated.

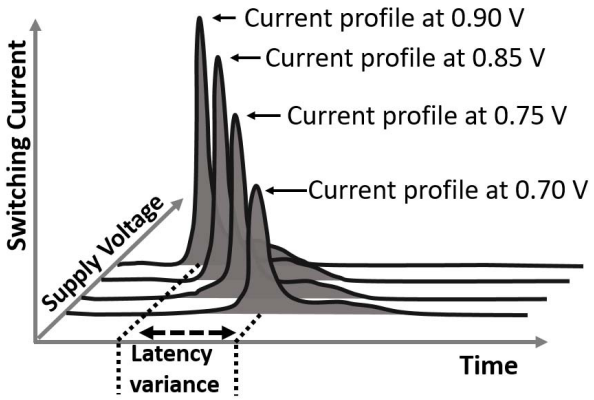

Fig. 1. Voltage–current-timing interdependency for a switching circuit.

Another main challenge to the traditional chip load model is exploring the noise impact over operation modes and mode transitions. Cui *et al.* [12] prepare multiple current profiles and manually switch the profile for different operation modes. On the other hand, in multicore designs, there are many combinations of mode transitions. Also, their transition timings could affect the noise magnitude and timing performance. For efficiently exploring the impacts of modes and their transitions, the chip load model should have an interface that can easily and flexibly manipulate the operation modes of individual cores, which contributes to finding unexpected noise and consequential timing behaviors.

The work in this paper is an extension of our preliminary work [19]. In this paper, we extend the preliminary work to the large-scale multicore system for replaying the on-chip timing information such as critical path delay, timing slack, and global clock skew. We introduce a control logic interface and critical path replica in the load model so that PDN designers can assess the on-chip timing information and explore the noise impact on different multicore operation modes and their transitions. In terms of the simulation quality, compared with the transistor-level model, we achieved over 300× run-time reduction in a test case. Compared with the current source model, the correlation of the current profile, current peak, and timing data is significantly improved. Furthermore, we reveal the critical path slack variation caused by the mode transition process and land-side capacitor (LSC) configurations. We also experimentally demonstrate the LSC boosts processor clock frequency.

II. Multicore Chip Load Modeling

This section describes the details of the proposed multicore chip load model. The overview of modeling flow is explained in Section II-A. Target multicore system and usage model are explained in II-B. Individual core load model is constructed in Section II-C. Detailed model characterization and simulation procedure are covered in Sections II-D and II-E, respectively.

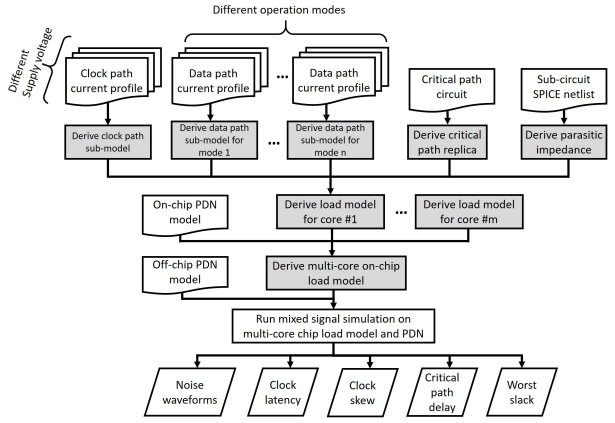

Fig. 2. Flow of multicore chip load modeling.

A. Overview of Chip Load Modeling Flow

Fig. 2 shows the overall modeling flow for a multicore chip load. A chip load model consists of an on-chip PDN model and switching circuit model. The on-chip PDN model consisting of various *RLC* components is used to deliver power to switching circuits, where this on-chip PDN model is supposed to be given to the flow. The procedure of switching circuit modeling is shown as gray blocks in Fig. 2. To complete the switching circuit modeling, three kinds of input materials are necessary, and they are explained in the following.

The first input material is the current profiles that are necessary to construct submodels for clock path and data path, respectively. The current profiles for the clock path are prepared over different voltage levels, and the current profiles for the data path are prepared over different supply voltage levels and operation modes (for example, shut down, clock-gated, full function, and reset), where the mode selection is design dependent and the designers need to choose the modes that consume large and small power. Here, the current profiles are generated by transistor-level SPICE simulation in this paper, but there are speed-up solutions provided by commercial electronic design automation tools, which claim a reasonable time for current profile preparation. The second input material is transistor-level SPICE netlist to extract parasitic impedance. The third one is a set of critical path subcircuits to generate submodels replaying the worst cycle-by-cycle slack. With these three inputs, we derive the voltage–current-timing-dependent load model for individual core circuit.

The same process is performed on other cores. By combing the multicore circuit model, which consists of multiple individual core models in parallel, with other on-chip PDN components such as bumps and PG meshes, we build up the multicore chip load model. Finally, the on-chip load model is connected with off-chip PDN to form a PDN system. Mixed-signal simulation is executed for the PDN system to generate on-chip and off-chip noise waveforms and on-chip timing information of clock latency, clock skew, critical path delay, and worst slack.

In this flow, the switching circuit model and on-chip PDN components can be constructed from subcircuit level to individual core circuit level depending on the granularity of the provided circuit current profile and on-chip PDN model. Note that the granularity of the current profiles and on-chip PDN model affects simulation runtime and model construction time. Appropriate granularity should be selected such that large power operation and mode transitions inducing large current variation can be reproduced. Without losing the generality, in the remainder of this paper, we build up the model from the individual core circuit level.

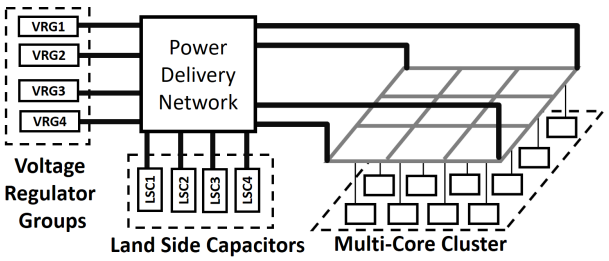

Fig. 3. Example of block diagram of power delivery network for multicore system.

B. Target Multicore PDN System and Usage Model

Let us illustrate a usage model with an example of a multicore system. The system block diagram is exemplified in Fig. 3. Suppose this example system is powered by multiple-phase voltage regulators separated into several voltage regulator groups (VRGs). The supply voltage is delivered across the board and package, which are represented by the multiport PDN in the diagram. The output of PDN is connected to on-chip power-ground mesh that supplies power to each core. Decoupling capacitors are attached to PDN at various locations. In Fig. 3, LSC, which is gaining its importance in modern high-performance chips, is depicted. Tasks for PDN designers may include determining LSCs.

The multicore cluster has many operation modes and their transitions. Individual cores may be activated or deactivated by clock and power gating according to environment and application requirements, and their workloads are scheduled and distributed by, for example, an operating system. Also, supply voltage and clock frequency may be controlled for each core or a group of cores. Such variations in PDN configuration and operation mode transitions can affect power supply noise and, consequently, impact chip timing. The proposed chip load model aims to provide timing information, such as clock skew, clock latency, path delay, and worst slack, to off-chip PDN designers so that, for example, various configurations of LSCs can be explored from a chip performance point of view.

The proposed load model is composed of multiple individual core load models. A high-level structure of individual core load model is depicted in Fig. 4, where the detail will be explained in Section II-C. We use a time–voltage-variant resistor to reproduce voltage-dependent load current taking into account voltage-dependent switching delay for a given operation mode. There are multiple time–voltage-variant resistors, and they are enabled or disabled by control logic interface so that mode transition is triggered. Critical paths

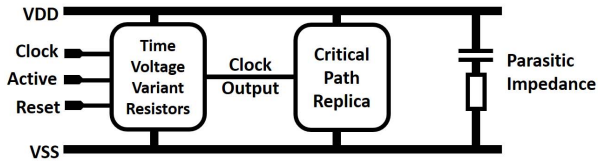

Fig. 4. Overall structure of individual core load model.

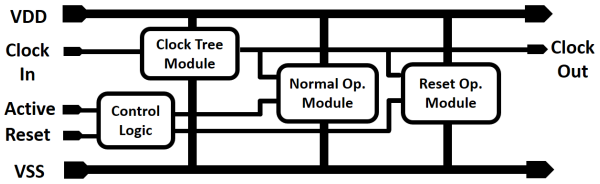

Fig. 5. Time–voltage-variant resistors model structure.

are represented by the critical path replica module to replay critical path delay. Also, parasitic and intrinsic decoupling capacitances are modeled in Fig. 4.

Instantiating multiple individual core models, a multiple core load model is organized as a core cluster with a global clock distribution network, which is also modeled as time–voltage-variant resistors. Hence, the global and local clock latency, skew, and path delays can be computed with simulation. In Section II-C, we will describe the details of the individual core load model.

C. Individual Core Load Model

This section discusses the details of the individual core load model. As discussed in Section I-B, the main challenge for a single core load model is to replay the interdependency between voltage, current, and timing. Here, we divide the interdependency modeling challenge into subtasks. First, for the current profile and supply voltage interdependency, we need to model the voltage-dependent equivalent resistance of switching transistors. With this voltage-dependent resistance, the interdependency between the current profile and the supply voltage is naturally considered in the circuit simulation. Second, for the voltage-timing interdependency, we need to develop clock path model and critical data path model that take into account supply voltage. Combining the clock latency and data path delay, we can provide the on-chip timing information. Finally, the current profile and timing should be aligned. Especially the switching peak current, which dominates the current profile, should be aligned with the clock latency. This task is achieved by the resistance profile (RP) method. The individual core load model is composed of three submodels as explained by Fig. 4. The time–voltage-variant resistor model is responsible for reproducing the switching current in time domain. Changing the active and inactive time–voltage-variant resistor models corresponds to operation mode transition, which is triggered by control signals such as set or reset. The critical path replica model takes the output clock with latency and reproduces the propagation delay in a set of the representative critical paths. The parasitic impedance is responsible for reproducing the voltage–current response in high-frequency domain.

Among the three components, developing the time–voltage-variant resistors is the key challenge to replaying the interdependency between clock latency, current profile, and supply voltage. This challenge is addressed by proposing a scaled RP method, which will be explained in Sections II-C1–II-C4. Parasitic impedance is described in Section II-C5 followed by the critical path replica in Section II-C6.

1) Time–Voltage-Variant Resistor Modeling: This section proposes a scaled profile method to model the time–voltage-variant resistor. The inside structure is shown in Fig. 5. The submodules of chip clock tree and data path are modeled separately.

In this diagram, only two modes of normal and reset operation are offered for simplifying the explanation. A reset signal is inputted to enable or disenable the submodel for different operation modes. Active signal is used to turn on or shut down the model. This structure is expandable for additional modes and submodules.

First, we define the RP element by a pair of $\big(t_n(V_{\rm DD})\ r_n(V_{\rm DD})\big)$, where t_n is the time in simulation and r_n is the equivalent load resistance. t_n and r_n are the functions of supply voltage $V_{\rm DD}$. The simulator updates the resistance r_n at t_n according to $V_{\rm DD}$, and naturally deduces current by Ohm's law. Supposing a core load operation is composed of N RP elements, we define **RP** as a vector pair

$$\mathbf{RP} = (\mathbf{T}_N\ \mathbf{R}_N) \tag{1}$$

where $\mathbf{T}_N$ and $\mathbf{R}_N$ are the time and resistance vectors, respectively. Each RP element pair consists of $t_n \in \mathbf{T}_N$ and $r_n \in \mathbf{R}_N$. Sections II-C2 and II-C3 explain the resistance vector modeling and time vector modeling, respectively.

2) Resistance Vector Modeling: Given a submodule switching circuit, N_{tr} transistors are conductive. Suppose V_{DS} over a conductive transistor is small, and supply voltage $V_{\rm DD} \approx V_{GS}$. Then, the equivalent resistance $r(V_{\rm DD})$ can be expressed by

$$r(V_{\rm DD}) = \frac{V_{\rm DD}}{\sum_{i=1}^{N_{tr}} I_i} \approx \left(\sum_{i=1}^{N_{tr}} \frac{(V_{\rm DD} - V_T)}{k_i} \cdot \left(\frac{W_i}{L_i}\right)\right)^{-1} \tag{2}$$

where I_i, k_i, L_i, and W_i are the drain current, conductivity factor, channel length, and channel width of individual transistors, respectively, and V_T is the threshold voltage. From (2), the equivalent resistance of a switching circuit can be approximated to a function of $V_{\rm DD}$. Meanwhile, since the equivalent resistance can be also derived from the supply voltage level and current profile via Ohm's law, the resistance can be expressed with a scaling factor by

$$r(V_{\rm DD}) = r(V_0) \cdot {\rm SR}(V_{\rm DD}) \tag{3}$$

where $V_{\rm DD}$ is the supply voltage, $r(V_0)$ is the equivalent resistance derived from current profile at nominal supply voltage V_0, and ${\rm SR}(V_{\rm DD})$ is the piecewise resistance scaling function fit from voltage and current profiles at different $V_{\rm DD}$ levels.

Fig. 6 exemplifies the advantage of this scaling method over conventional methods. A four-stage clock tree is selected for

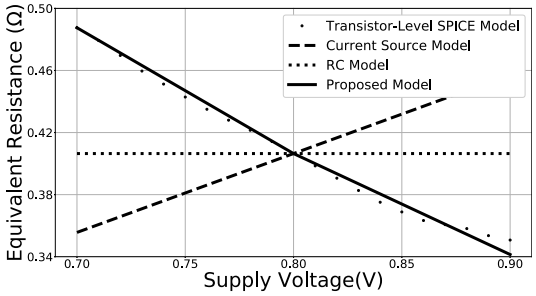

Fig. 6. Comparison of equivalent resistance during clock switching. Constant supply voltage varies from 0.70 to 0.90 V.

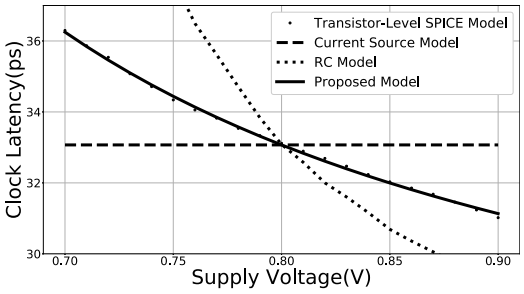

Fig. 7. Clock latency estimation comparison. Constant supply voltage varies from 0.70 to 0.90 V.

demonstration, in which four different modeling methods are compared. The result labeled transistor-level SPICE model is obtained by simulating the transistor-level clock tree netlist, and it is the reference. The current source model is based on the current profile that is obtained from the transistor-level SPICE simulation result at nominal voltage. The *RC* model is constructed according to [15], and the parameters are tuned manually so that clock latency and peak switching current are equal with the transistor-level simulation result at nominal voltage. The proposed model uses Verilog-A to implement the time–voltage-variant resistor. The resistance vector is scaled according to (3). With these four models, we varied the supply voltage level and measured the peak switching current for 100 clock cycles. Then, we divide the supply voltage by the averaged peak switching current to obtain the equivalent resistance. From the result, we can see that the *RC* model and current source model underestimate the resistance at the low supply voltage and overestimate it at the high supply voltage; and consequently, the current is also misestimated. On the other hand, the proposed model based on the scaled resistance correlates closely with the transistor-level SPICE model simulation result as we expected.

3) Time Vector Modeling: Suppose a given path delay D is divided into N intervals and Δt_n denotes the nth interval. Assuming intervals are sufficiently short, the interval duration is determined by average voltage V_{An} during the interval since the interval is impacted by transistor switching speed. This transistor switching includes *RC* charging and discharging processes with *RC* time constant, and hence, the interval can also be scaled by time scaling function similarly to resistance vector elements

$$\Delta t_n(V_{An}) = \Delta t_n(V_0) \cdot \mathrm{ST}_n(V_{An}) \tag{4}$$

where $\mathrm{ST}_n(V_{An})$ is the time scaling function for nth interval. When the intervals are evenly distributed along the path, we use a single time scaling function $\mathrm{ST}(V_{An})$ as the representative. In this case, the path delay is expressed as

$$D = \sum_{n=1}^{N} (\ \Delta t_n(V_0) \cdot \mathrm{ST}(V_{An})\ ). \tag{5}$$

Then, the time vector element t_n becomes

$$t_{n+1} = t_n + \Delta t_n(V_0) \cdot \mathrm{ST}(V_{An}). \tag{6}$$

At a constant supply voltage V_{DD}, path delay (5) can be simplified as

$$D(V_{\mathrm{DD}}) = D(V_0) \cdot \mathrm{ST}(V_{\mathrm{DD}}) = \sum_{n=1}^{N} \Delta t_n(V_0) \cdot \mathrm{ST}(V_{\mathrm{DD}}). \tag{7}$$

Time scaling function $\mathrm{ST}(V_{\mathrm{DD}})$ can be extracted from the circuit simulation or static timing analysis with libraries at different voltages. With (3) and (6), we can scale the RP of (1) and deduce the clock latency under both constant supply voltage and dynamic supply noise by (7) and (5).

Fig. 7 shows the estimated latency of the four-stage clock tree. The transistor-level SPICE model, current source model, and *RC* model are constructed with the same configurations as shown in Fig. 6. The proposed model uses Verilog-A to implement the time–voltage-variant resistor. The time vector is scaled according to (7). We can see *RC* model and current source model either overestimate or underestimate the path delay under different supply voltages. The proposed model based on scaled latency, on the other hand, correlates closely with the transistor-level SPICE simulation result.

4) Operation Mode Transition: In the multicore cluster, an individual core may transit across various operation modes. These modes have different current consumptions and then generate different dynamic supply noises. To replay the voltage-current-timing behavior around the mode transition, the RP is prepared for each operation mode. When a core transits from an original mode to a new mode at simulation time t, the RP module of the original mode is disabled, which means the current through this RP module is set to zero. Meanwhile, the RP module of the new mode is activated, and the equivalent resistance of this RP module will be, hereafter, updated by the simulation engine. Such a transition process can be described with the Verilog-A logic interface along with traditional Verilog test bench.

An example of mode transition is described in Algorithm 1. Suppose a data path RP module has three operation modes, which are shut-down mode, reset mode, and normal mode. The mode transition can be controlled by two signal pins named "Reset" and "Active." Depending on the logic level of control signal pins, the intended RP module is scheduled for simulation.

5) Parasitic Impedance Modeling: For the parasitic impedance part, the equivalent circuit model shown in Fig. 8 is characterized with small-signal analysis, where C_1 and

Algorithm 1 Operation Mode Transition Algorithm

Input: Reset, Active

Main Routine:

```
if Active signal is not set then
  Enable shut-down mode RP module
  Disable other modes' RP module
else
  if Reset is enabled then
    Enable reset mode RP module
    Disable other modes' RP module
  else
    Enable normal operation mode RP module
    Disable other modes' RP module
  end if
end if
```

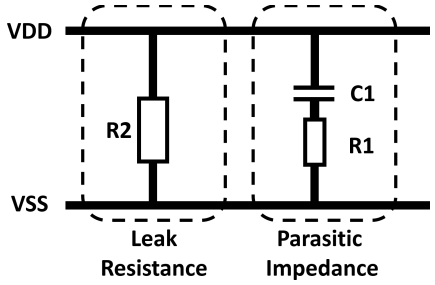

Fig. 8. Parasitic impedance model.

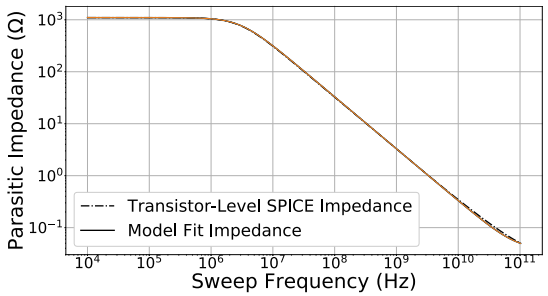

Fig. 9. Parasitic impedance extracted by small-signal analysis.

R_1 represent the parasitic impedance and R_2 is chip leak resistance.

Let us show an example of the extracted parasitic impedance of the processor core used for the experiments in the next section. By sweeping frequency of the small ac signal from 1 kHz to up to 1000 GHz, the equivalent impedance is obtained as Fig. 9. Then, the parameters R_1, C_1, and R_2 are derived by least squares fitting. Since the leakage current is included in RP, we remove R_2 and keep only C_1 and R_1 as the parasitic impedance part.

6) Critical Path Replica Modeling: The critical path replica structure is demonstrated in Fig. 10. The replica interface will duplicate the clock signal, supply voltage (VDD), and ground voltage (VSS) to the critical path circuit. Therefore, the critical path circuit is isolated from the main power supply. The replica interface is implemented in Verilog-A. The critical path circuit may accommodate a set of critical paths, and they can be, for example, a transistor-level netlist or a mathematical model. In this paper, we simply use transistor-level netlist to model the set of critical paths. These critical paths are selected based on static timing analysis at various supply voltages. More sophisticated critical path selection and synthesis methods are well discussed in [1], [3], and [7]–[11]. During the model simulation, the worst critical path slack is measured cycle-by-cycle.

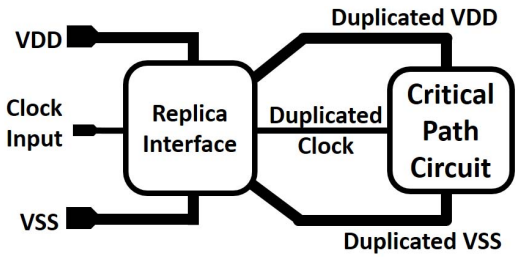

Fig. 10. Critical path replica model.

The multicore chip load is composed of individual core load models. Also, a global clock distribution network is modeled as a time–voltage-variant resistor model and attached to the multicore chip load. Then, the clock skew of n-core chip load is derived by

$$\text{Skew} = \max |D_i - D_j| \quad (\forall i, j \in n) \tag{8}$$

where D_i and D_j are the clock latency to the clock terminals of sequential elements in cores i and j, respectively. The clock latency is derived by (5).

Since the critical path delay is reproduced by the critical path replica model, the worst timing slack at each clock cycle is derived by

$$\text{Slack}(i) = T_{\text{clk}}(i+1) - T_{\text{clk}}(i) - T_{\text{setup}} - T_{\text{path}}(i) \tag{9}$$

where $T_{\text{clk}}(i)$ is the time of the clock rising edge for ith clock cycle, T_{setup} is the setup time of sequential element, and T_{path} is the critical path delay.

D. Core Load Model Characterization

This section summarizes the characterization procedure of the individual core load model. The individual core load model is composed of three submodels demonstrated in Fig. 4. The parasitic impedance model is characterized by small-signal analysis. The critical path replica model can be characterized from static timing analysis. As for the time–voltage-variant resistor model, both resistance vector and time vector need to be characterized to form RP. The items and scaling functions of resistance vector and time vector are characterized through the following process.

Step 1: Generate current profile at nominal voltage V_0 and measure path delay or clock latency $D(V_0)$.

Step 2: Convert current profile into RP pair $(r(V_0)\ t_n(V_0))$.

Step 3: Obtain current profile for tens of clock cycles at different supply voltages, measure clock latency $D(V_{\text{DD}})$, and derive RP pair $(r(V_{\text{DD}})\ t_n(V_{\text{DD}}))$.

Algorithm 2 RP Module Simulation Procedure

Input: V_{DD}, V_{in_signal}
Output: I, V_{out_signal}
Initialization :
Set leak resistance
Main Routine :
if V_{in_signal} is changed **then**
  for $n = 1$ to N **do**
    Obtain r_n and t_n from RP.
    Calculate the resistance value with (3), and time interval with (4).
    Schedule the next resistance update time, which is derived by (6).
    Copy the V_{in_signal} value to V_{out_signal} once the time after the input signal is given becomes larger than the path delay in (5).
  end for
end if

Step 4: Run fitting process and generate scaling functions for resistance vector and timing vector, according to (3) and (4).
Step 5: Compose RP.

In Step 1, current profile at a constant voltage can be generated by either traditional transistor-level simulation or more sophisticated power estimation tools. In Step 2, the RP pair $\left(t_n(V_0)\ r_n(V_0)\right)$ is constructed with temporal discretization and Ohm's law. In Step 3, tens of clock cycle simulations are needed to derive latency and RP as sample data, which will be used to build the scaling functions in Step 4. In Step 5, the final RP is composed of time and resistance vectors defined by (1).

E. Resistance Profile Simulation Procedure

Suppose an RP during a clock cycle is composed of N RP elements. Once a clock rising edge is detected, the first RP element will be selected to deduce equivalent resistance as $r_1(V_{\mathrm{DD}})$. Then, the time to update the next RP element is also deduced with (4). Once resistance is determined at a given simulation time, the current value is computed by Ohm's law in a circuit simulator. Such a procedure is performed until all the RP elements $(t_n(V_{\mathrm{DD}})\ r_n(V_{\mathrm{DD}}))$ are simulated. As a special case for the clock tree RP module, once the simulation time after the clock signal is given is larger than the clock path delay, which is derived by (5), the input clock signal will be copied to the output clock signal port. Hence, the clock propagates with the computed clock path latency. Finally, the output clock signal is duplicated to the critical path replica model, and critical path slack is measured cycle-by-cycle by (9). The RP simulation procedure is described in Algorithm 2.

This algorithm can be implemented with Verilog-A, and hence, our model can be cosimulated with Verilog and SPICE modules. By applying a similar approach to other subcircuit modules or modes, we can model larger-scale complex processors.

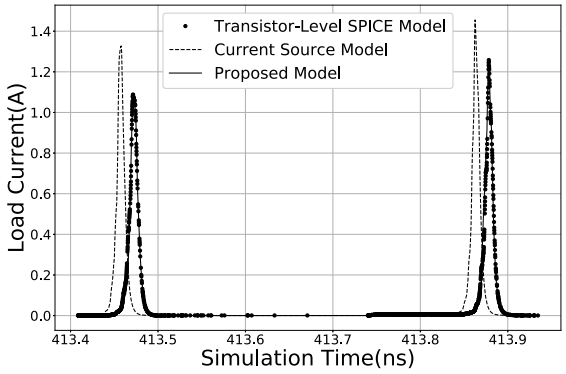

Fig. 11. Current waveform comparison within one clock cycle.

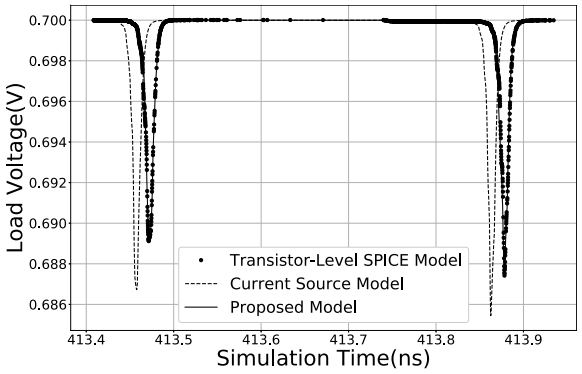

Fig. 12. Load voltage waveform comparison within one clock cycle.

III. Experiment Evaluation

This section shows the experimental results to validate the proposed model. We first demonstrate the simulation quality for an individual core load model. Second, we conduct system-level experiments by building up a multicore PDN system with the chip load model and off-chip PDN for demonstrating the timing impact under different off-chip PDN configurations.

A. Individual Core Experiment

For the individual core experiment, we prepared a 32-bit OpenRISC processor synthesized with NanGate 15-nm open cell library. The number of cells is over 17 k, the clock frequency for the core processor logic is 1.2 GHz, and the average clock latency is 114.9 ps at 0.8-V supply voltage. A cyclic redundancy check (CRC) checksum program is given to OpenRISC as workload. The characterization for 500-cycle operation finished within 2 h in this test case.

First, we illustrate the reproducibility of current and voltage waveforms comparing current source model, full SPICE netlist simulation, and the proposed on-chip load model. Then, the voltage source of 0.7 V is connected to a two-port PDN described by S-parameter. The chip load model is connected to the output load port of PDN. The resulting current waveform is shown in Fig. 11, and the load voltage waveform is shown in Fig. 12. The waveform of the proposed model is depicted with a solid line, from which we can find that

TABLE I
AVERAGE PEAK LOAD CURRENT AND AVERAGE CLOCK LATENCY COMPARISON AT VARIOUS SUPPLY VOLTAGES

Supply Volt.(V)	Peak Curr.(A)		Err(%)	Latency (ps)		Err(%)
	SPICE	Model		SPICE	Model	
0.70	1.38	1.36	1.4%	131.7	132.2	0.4%
0.73	1.54	1.50	2.7%	125.8	126.1	0.3%
0.77	1.74	1.70	2.7%	119.0	119.4	0.4%
0.80	1.91	1.87	2.0%	114.9	115.3	0.4%
0.83	2.10	2.03	3.2%	111.4	111.7	0.3%
0.87	2.35	2.27	3.3%	107.5	107.7	0.3%
0.90	2.50	2.47	1.2%	104.9	105.1	0.2%
Avg.	-	-	**2.4%**	-	-	**0.3%**

both the current and voltage waveforms correlate closely with the transistor-level SPICE simulation result (dotted line). The interdependency among voltage, current, and switching time is also replayed. On the other hand, the current source (dashed line) overestimates voltage noise and underestimates timing delay.

Second, we evaluate the accuracy of the individual core load model quantitatively at different supply voltages from 0.7 to 0.9 V. The results are listed in Table I. This evaluation simulated for 200 clock cycles. For the peak current evaluation, we calculated the errors for 400 current peaks and computed the average of them, where 400 peaks are 200 clock cycles multiplied by two peaks per clock cycle. The average error for individual peak currents is 2.4%. On the other hand, conventional current source and *RC* model cannot attain such accuracy, and the average peak current errors are 17.6% and 10.5%, respectively. For the clock latency evaluation, the average latency error of the proposed model is 0.3%, whereas the average errors for the current source and *RC* models are 6.3% and 11.4%, respectively. In particular, the current source model suffered up to 38.5% error in peak current estimation, and *RC* model had up to 39.2% error in latency estimation.

Third, to validate the individual core load model under dynamic supply noise, we injected a sinusoidal noise with 100-mV amplitude whose frequency ranged from 100 MHz to 1 GHz, where 100 MHz is roughly 10× lower and 1 GHz is almost similar to the clock frequency. We simulated 100 clock cycles for both full-SPICE netlist and the proposed on-chip load model. Figs. 13 and 14 show the clock latency comparison. We can see both the clock latencies are well correlated. The average latency errors are 1.5% for 100-MHz noise and 2.6% for 1-GHz noise. The peak current under dynamic noise is also compared in Figs. 15 and 16. The average peak current errors are 2.3% for 100-MHz noise and 2.2% for 1-GHz noise.

B. Multicore PDN System Experiment

For larger system level experiments, we build up a multicore PDN system. The high-level schematic is demonstrated in Fig. 3, in which, four VRGs provide 16-phase 0.8-V dc supply voltage. Multiport PDN consists of S-parameter and *RLC* elements to model printed circuit board and package

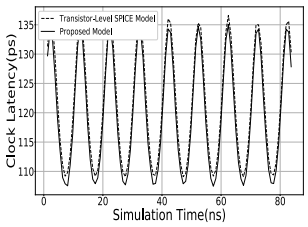

Fig. 13. Clock latency estimation with 100-MHz supply noise.

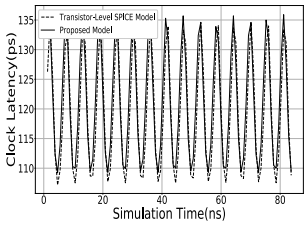

Fig. 14. Clock latency estimation with 1-GHz supply noise.

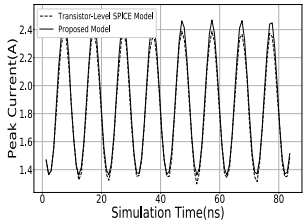

Fig. 15. Peak current estimation with 100-MHz supply noise.

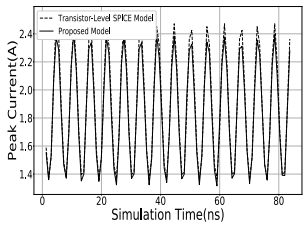

Fig. 16. Peak current estimation with 1-GHz supply noise.

circuit. At the chip load side, the connection between 16-core cluster power-ground mesh is depicted in Fig. 17. For each mesh grid, the segment resistance and inductance are 50.4 mΩ, and 5.6 fH, respectively. The clock signal propagates through a global clock tree shown in Fig. 18. The main process to construct the multicore load model is done by python scripts, which takes around 15 min to convert a set of given current profiles and netlist to chip load model. Extra manual work is also needed for writing glue logic and testbench scenarios. Assuming a template is given for the glue logic and testbench, this manual work takes minutes to hours, depending on the size and complexity of the core.

Using this PDN system, we first verify the timing information accuracy of individual core load model. The core load model is connected to the center of power-ground mesh, which is the position of core #6 in Fig. 17. Four current sources were connected to the adjacent grids to mimic the transient process of neighboring cores. These current sources increase their current consumption from 40 to 400 mA at 470 ns, then

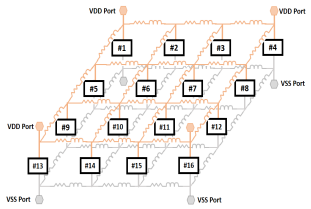

Fig. 17. Sixteen-core cluster with power-ground mesh.

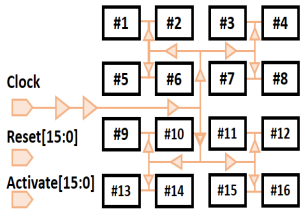

Fig. 18. Sixteen-core cluster with clock tree and control signal.

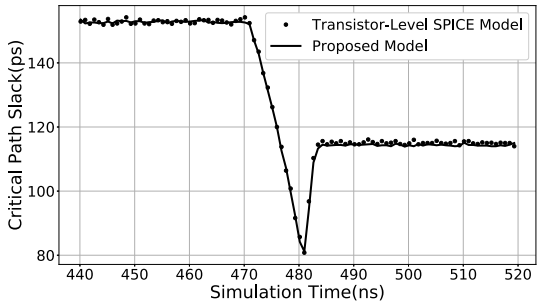

Fig. 19. Cycle-by-cycle critical path slack comparison during transient process.

drop back to 200 mA in 2 ns. We compared the cycle-by-cycle critical path slack between transistor-level SPICE netlist and the proposed chip load model. The slack comparison result is shown in Fig. 19, where the average estimation error of the path slack is 0.1% and the maximum error is 2.6%. In this simulation, the simulation with full transistor-level SPICE netlist takes 68 537 s, while that with the proposed model takes 172 s, which means over 300× run-time reduction. Note that this run-time reduction is more significant when the system under evaluation is larger.

Next, we evaluate the on-chip timing information for different PDN configurations and operation mode transition scenarios. In scenario 1, we activate four cores at the beginning, which are core #1, #2, #5, and #6 in Fig. 17. Then, we activate the other 12 cores simultaneously after 462 ns, followed by 15-ns reset operation mode, then switch to the normal operation mode. The CRC checksum program is used as the workload in the normal operation mode. In scenario 2, we activate the same four cores at the beginning as scenario 1, but the remaining 12 cores are activated in a gradual process; that is, every four cores are activated after 5 ns. In both scenarios, we vary the LSC capacitance from 0.08 to 20 nF, and then measure the critical path slack of core #6, which is located near the center of the power-ground mesh. The cycle-by-cycle slack is shown in Fig. 20.

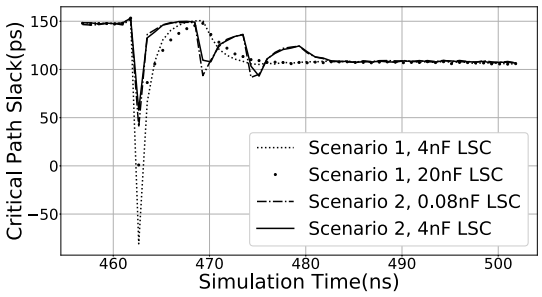

Fig. 20. Cycle-by-cycle critical path worst slack of core #6.

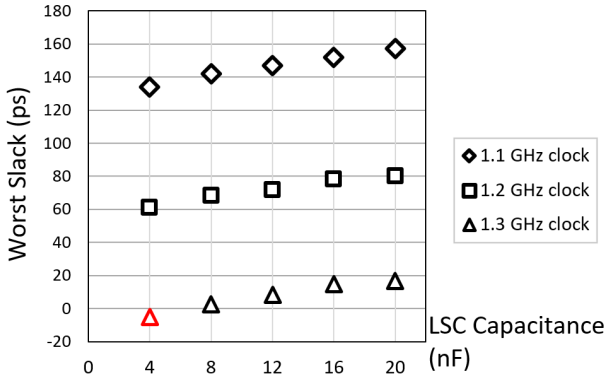

Fig. 21. Worst timing slack under different LSC configurations. Three clock frequencies are inputted to the multicore system.

From the simulation result, off-chip PDN designers can assess the LSC effectiveness under different mode transition procedures. For example, when 12 cores are enabled simultaneously, at least 20-nF LSC capacitance is required to fix setup timing violation for core #6, which is shown as dot lines in Fig. 20. On the other hand, when the mode transition is scheduled in a gradual way, 4-nF LSC is sufficient to ensure 50-ps critical path slack. In this experiment setup, the average simulation runtime is 1087.5 s. This run-time range enables off-chip designers to explore PDN configurations over various mode transition scenarios.

Third, we perform an experiment that tunes multicore system performance with different PDN configurations. In this experiment, we use 16 core load models to form a multicore cluster. As a core load configuration, we turn-on eight cores at the beginning and then turn-on remaining eight cores at 470 ns. Each core switches to reset mode for 15 ns before entering into normal operation mode. The CRC checksum program is used as the workload in the normal operation mode. As for off-chip PDN configuration, we vary the input clock frequency from 1.1 to 1.3 GHz and vary the LSC capacitance from 4 to 20 nF. The worst timing slack among the 16 cores is evaluated.

Fig. 21 shows the result of the worst slack. From the simulation result, off-chip PDN designers can find the effectiveness of LSC capacitance on retrieving timing slack. For example, when LSC is increased from 4 to 20 nF, an extra timing slack of 20 ps is attained. When 1.3-GHz clock is driving the system, a negative timing slack of −5.1 ps is presented by the load model, which is shown as a red triangle

in Fig. 21. The timing data are helpful for off-chip PDN designers to assess the noise impact on chip performance. On the other hand, by increasing the LSC to 20 nF, the worst timing slack is improved to 16.7 ps, which means the chip timing constraint under 1.3-GHz frequency is satisfied with 20-nF LSC configuration. Such an off-chip PDN optimization becomes feasible with the proposed on-chip load model.

IV. Conclusion

In this paper, we proposed a multicore chip load model that could replay the load current and timing information under supply voltage noise. The model also supports extensive design exploration with operation mode variation and different PDN parameters. The experiment shows that the proposed model achieves much better correlation compared with the traditional current source-based model and *RC*-based model, while over 300× run-time reduction is achieved compared with full SPICE netlist simulation. The off-chip PDN modification experiments show the proposed model can guide off-chip PDN designers with on-chip timing information.

References

[1] P. N. Whatmough, S. Das, and D. Bull, "Power integrity analysis of a 28 nm dual-core ARM Cortex-A57 cluster using an all-digital power delivery monitor," *IEEE J. Solid-State Circuits*, vol. 52, no. 6, pp. 1643–1654, Jun. 2017.

[2] M. Saint-Laurent and M. Swaminathan, "Impact of power-supply noise on timing in high-frequency microprocessors," *IEEE Trans. Adv. Packag.*, vol. 27, no. 1, pp. 135–144, Feb. 2004.

[3] X. Wang, D. Zhang, D. Su, L. Winemberg, and M. Tehranipoor, "A novel peak power supply noise measurement and adaptation system for integrated circuits," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 24, no. 5, pp. 1715–1727, May 2016.

[4] R. Thomas, K. Barber, N. Sedaghati, L. Zhou, and R. Teodorescu, "Core tunneling: Variation-aware voltage noise mitigation in GPUs," in *Proc. Int. Symp. High Perform. Comput. Archit. (HPCA)*, Mar. 2016, pp. 151–162.

[5] S. Das, P. Whatmough, and D. Bull, "Modeling and characterization of the system-level power delivery network for a dual-core ARM Cortex-A57 cluster in 28 nm CMOS," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design (ISLPED)*, Jul. 2015, pp. 146–151.

[6] A. Todri, M. Marek-Sadowska, and J. Kozhaya, "Power supply noise aware workload assignment for multi-core systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Nov. 2008, pp. 330–337.

[7] R. Bertran *et al.*, "Voltage noise in multi-core processors: Empirical characterization and optimization opportunities," in *Proc. 47th Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec. 2014, pp. 368–380.

[8] A. Drake *et al.*, "A distributed critical-path timing monitor for a 65 nm high-performance microprocessor," in *IEEE Int. Solid-State Circuits Conf. (ISSCC) Dig. Tech. Papers*, Feb. 2007, pp. 398–399.

[9] J. Kim, K. Choi, Y. Kim, W. Kim, K. Do, and J. Choi, "Delay monitoring system with multiple generic monitors for wide voltage range operation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 26, no. 1, pp. 37–49, Jan. 2018.

[10] Q. Liu and S. S. Sapatnekar, "Capturing post-silicon variations using a representative critical path," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 29, no. 2, pp. 211–222, Feb. 2010.

[11] K. A. Bowman *et al.*, "A 45 nm resilient microprocessor core for dynamic variation tolerance," *IEEE J. Solid State Circuits*, vol. 46, no. 1, pp. 194–208, Jan. 2011.

[12] W. Cui, P. Parmar, J. Morgan, and U. Sheth, "Modeling the network processor and package for power delivery analysis," in *Proc. EMC*, vol. 3, Aug. 2005, pp. 690–694.

[13] L. Zheng, Y. Zhang, and M. S. Bakir, "Full-chip power supply noise time-domain numerical modeling and analysis for single and stacked ICs," *IEEE Trans. Electron Devices*, vol. 63, no. 3, pp. 1225–1231, Mar. 2016.

[14] H. H. Chen and J. S. Neely, "Interconnect and circuit modeling techniques for full-chip power supply noise analysis," *IEEE Trans. Compon., Packag., Manuf. Technol. B*, vol. 21, no. 3, pp. 209–215, Aug. 1998.

[15] Y. Ogasahara, T. Enami, M. Hashimoto, T. Sato, and T. Onoye, "Validation of a full-chip simulation model for supply noise and delay dependence on average voltage drop with on-chip delay measurement," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 10, pp. 868–872, Oct. 2007.

[16] S. Lin and N. Chang, "Challenges in power-ground integrity," in *Proc. ICCAD*, Nov. 2001, pp. 651–654.

[17] Y. Shim and D. Oh, "System level modeling of timing margin loss due to dynamic supply noise for high-speed clock forwarding interface," *IEEE Trans. Electromagn. Compat.*, vol. 58, no. 4, pp. 1349–1358, Aug. 2016.

[18] G. Bai, S. Bobba, and I. N. Hjj, "Static timing analysis including power supply noise effect on propagation delay in VLSI circuits," in *Proc. 38th Design Automat. Conf.*, Jun. 2001, pp. 295–300.

[19] J. Chen, T. Kanamoto, H. Kando, and M. Hashimoto, "An on-chip load model for off-chip PDN analysis considering interdependency between supply voltage, current profile and clock latency," in *Proc. IEEE 22nd Workshop Signal Power Integr. (SPI)*, May 2018, pp. 1–4.

Jun Chen received the B.E. and M.E. degrees in control theory and engineering from Tongji University, Shanghai, China, in 2004 and 2007, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Systems Engineering, Osaka University, Suita, Japan.

From 2008 to 2016, he was with Synopsys Inc., Shanghai, where he was engaged in research and development of routing congestion, power and placement optimization flow. His current research interests include computer-aided-design for digital integrated circuits and power and signal integrity analysis.

Hajime Kando was born in Machida, Japan in 1968. He received the B.S. degree in electronics from Kougakuin University, Tokyo, Japan, in 1991.

He was with Murata Manufacturing Company Ltd., Kyoto, Japan, where he was involved in researching and developing surface acoustic devices, RF devices, and power devices. He is currently a member of the Innovative Technology Development Department, Yasu, Japan, where he is researching a new technology for RF devices and Power devices.

Toshiki Kanamoto (M'08) received the B.S. and M.S. degrees in physics from Nihon University, Tokyo, Japan, and the Ph.D. degree in information science from Osaka University, Osaka, Japan.

From 1991 to 2003, he was with Mitsubishi Electric Corporation, Tokyo. From 2003 to 2016, he was with Renesas Technology and Renesas Electronics Corporation, Kodaira, Japan, where he was engaged in research and development of the physical design and verification technologies for LSI and power devices. Since 2016, he has been a Professor with the Department of Electronics and Information Technology, Graduate School of Science and Engineering, Hirosaki University, Aomori, Japan.

Dr. Kanamoto is currently a Senior Member of the Institute of Electronics, Information and Communication Engineers and the Information Processing Society of Japan. He has been a member of the Japan Electronics and Information Technology Industries Association (JEITA), since 2001.

Cheng Zhuo (S'06–M'12–SM'16) received the B.S. and M.S. degrees in electronic engineering from Zhejiang University, Hangzhou, China, in 2005 and 2007, respectively, and the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor, MI, USA, in 2010.

He is currently a Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His current research interests include 3-D integration, hardware acceleration, and power and signal integrity.

Dr. Zhuo was a recipient of the 2012 ACM SIGDA Technical Leadership Award, the 2016 DAC Best Paper Nomination, and the 2017 JSPS Invitation Fellowship. He has served on the technical program committees of many international conferences and is an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, *Elsevier Integration*, and IEEE VLSI CAS NEWSLETTER.

Masanori Hashimoto (S'00–A'01–M'03–SM'11) received the B.E., M.E., and Ph.D. degrees in communications and computer engineering from Kyoto University, Kyoto, Japan, in 1997, 1999, and 2001, respectively.

He is currently a Professor with the Department of Information Systems Engineering, Graduate School of Information Science and Technology, Osaka University, Suita, Japan. His current research interests include the design for manufacturability and reliability, timing and power integrity analysis, reconfigurable computing, soft error characterization, and low-power circuit design.

Dr. Hashimoto was a recipient of the Best Paper Awards from ASP-DAC in 2004 and RADECS in 2017, and the Best Paper Award of the IEICE Transactions in 2016. He was on the Technical Program Committee of international conferences, including DAC, ICCAD, ITC, Symposium on VLSI Circuits, ASP-DAC, and DATE. He serves/served as an Associate Editor for the IEEE TRANSACTIONS ON VLSI SYSTEMS, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS I, *ACM Transactions on Design Automation of Electronic Systems*, and *Elsevier Microelectronics Reliability*.