PAPER Special Section on VLSI Design and CAD Algorithms

Linear Time Calculation of On-Chip Power Distribution Network Capacitance Considering State-Dependence

Shiho HAGIWARA[†a)], **Koh YAMANAGA**[††], **Ryo TAKAHASHI**[†††], ***Student Members***, **Kazuya MASU**[†], ***and*** **Takashi SATO**[††††], ***Members***

SUMMARY A fast calculation tool for state-dependent capacitance of power distribution network is proposed. The proposed method achieves linear time-complexity, which can be more than four orders magnitude faster than a conventional SPICE-based capacitance calculation. Large circuits that have been unanalyzable with the conventional method become analyzable for more comprehensive exploration of capacitance variation. The capacitance obtained with the proposed method agrees SPICE-based method completely (up to 5 digits), and time-linearity is confirmed through numerical experiments on various circuits. The maximum and minimum capacitances are also calculated using average and variance estimation. Calculation times are linear time-complexity, too. The proposed tool facilitates to build an accurate macro model of an LSI.

key words: *capacitance, power distribution network, state-dependency, integrated circuit modeling, electromagnetic interference*

1. Introduction

Implementation of power distribution network (PDN) is one of the most important steps in designing modern electrical systems. Low design-quality PDN could cause serious faults such as supply voltage fluctuation, timing variation, electro-magnetic interference, etc. Large scale simulation of the system PDN, including package, printed circuit board, and LSI chip is necessary to estimate the behavior of the system. Because this simulation includes a large number of components, the simulation is intractable in many cases. Some of the components are replaced with their macro models, or equivalent circuit models, to reduce problem size. As the equivalent circuit, an LSI chip is often modeled using current sources and a few linear circuit elements including resistors and capacitors. Figure 1 shows one of such models, LECCS [1]. Among others, the capacitance model (on-chip PDN capacitance, C_{ckt}) of an LSI is particularly important because it determines resonant and anti-resonant frequencies of the supply network, combined with the equivalent inductance of the system. Careful attention is required to determine capacitance values and ranges in the equivalent model.

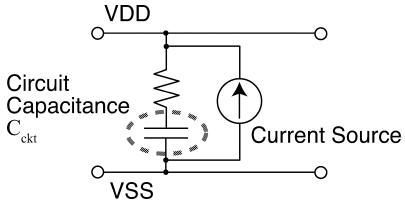


Fig. 1 Linear equivalent circuit and current source (LECCS model) [1].

The bottom-up approach for obtaining the capacitance is not as easy as it looks because the capacitance between power supply (VDD) and ground (VSS) varies with its input at the package pin and internal states of the circuit [2]–[4]. In [2], the average capacitance over all input states is used as the representative capacitance of a standard logic cell. Ignoring the state dependence may lead to time-dependent error considering the transient change of the logic state during circuit operations. Authors of [4] proposed to model capacitance statistically by its mean and variance considering the state-probability of the gates using a switching activity simulator. However, it still is very difficult to obtain the range of capacitance variation correctly because the calculated capacitance would vary significantly according to the slight change of a logical correlation setting. To avoid the correlation uncertainty, the authors of [3] proposed a logic-simulation based algorithm to find the worst-case input vector that maximizes C_{ckt}. To make this algorithm feasible, capabilities of calculating time-domain logic changes and accurate capacitance calculation due to the logic changes are both required. Thus far, SPICE-level circuit simulator is the only tool that satisfies the above requirements. When a circuit simulator is employed for the capacitance calculation, the size of the analyzable circuit becomes highly limited.

In this paper, we propose a state-dependent capacitance calculation that can analyze large scale circuits with SPICE-compatible accuracy. The proposed method utilizes built-in logic simulator for detailed internal state calculation and a capacitance look-up table (CLUT) for fast and state-dependent capacitance calculation. The change of wire capacitances associated with the state-changes is also correctly considered. Time-complexity of the proposed calculation is linear to the number of gates by construction, as compared to SPICE simulations that have super-linear complexity.

The rest of this paper is organized as follows. In Sect. 2,

Manuscript received March 19, 2010.
Manuscript revised June 19, 2010.

[†]The authors are with the Integrated Research Institute, Tokyo Institute of Technology, Yokohama-shi, 226-8503 Japan.
[††]The author is with Murata Manufacturing Co., Ltd., Nagaokakyo-shi, 617-8555 Japan.
[†††]The author is with The University of Tokyo, Tokyo, 153-8505 Japan.
[††††]The author is with Kyoto University, Kyoto-shi, 606-8501 Japan.
a) E-mail: paper@lsi.pi.titech.ac.jp

DOI: 10.1587/transfun.E93.A.2409

Copyright © 2010 The Institute of Electronics, Information and Communication Engineers

we explain capacitance components in a logic circuit, which have to be considered in this analysis. In Sect. 3, key concepts to realize a linear-time capacitance calculation are described. How a capacitance for a given input vector pattern is calculated is also described. In Sect. 4, application examples are presented, which confirms time-linearity and SPICE-compatible accuracy. Finally, conclusions are stated in Sect. 5.

2. Composition of On-Chip PDN Capacitance

The proposed capacitance calculation tool consists of the following three steps, 1) enumerate all capacitors that potentially contribute to the total PDN capacitance, 2) recognize connections of electrode of each capacitor, and 3) summate capacitances according to series/parallel connections. In this section, 1) and 2) are described.

2.1 On-Chip Capacitor Enumeration

The major components of the total capacitance of a logic circuit (C_{ckt} in Fig. 1) are wire capacitance (C_{wire}), logic cell capacitance (C_{cell}). Figure 2 depicts the capacitances inside an LSI.

C_{wire} further consists of the capacitances between pairs of signal wires, between pairs of power lines, and between pairs of power line and signal wire. The n- and p-wells on the substrate are supplied with VDD and VSS, respectively, for device isolation. Thus, capacitances between signal wire and substrate can also be considered as part of the capacitances between signal wire and power line. These capacitances can be obtained by parasitic extractors.

C_{cell} consists of capacitances associated with MOS transistors and coupling capacitances between wires inside standard cells. Here, junction capacitances formed between n-well and substrate, and between p-well and substrate are both considered to form capacitors between VDD and VSS.

2.1.1 Wire Capacitance

Capacitances between wires do not always contribute to C_{ckt}. If the two electrodes of a capacitor are in the equal potential, it has to be eliminated. Let us consider a capacitance between signal wires j and k in Fig. 3(a). When the wires j and k are both logical high (see the upper figure) both wires are connected to VDD through the on-resistances of pMOS transistors in the standard cells driving the wires. In this situation, $C_{j,k}$ is not effective and thus it should be eliminated. On the other hand, when wire k is logical low while j is high, $C_{j,k}$ counts. Similarly, as shown in Fig. 3(b), capacitances between a signal wire and power lines may or may not count as capacitance.

Formally, a total wire capacitance C_{wire} is represented as follows.

$$C_{wire} = \sum_{j}^{n_{\text{wire}}} \sum_{k}^{n_j} C_{j,k}(b_j \oplus b_k). \quad (1)$$

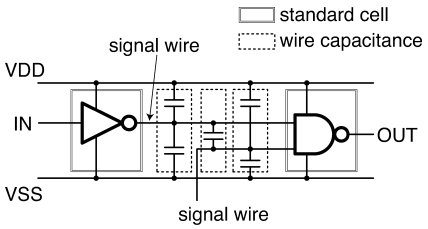
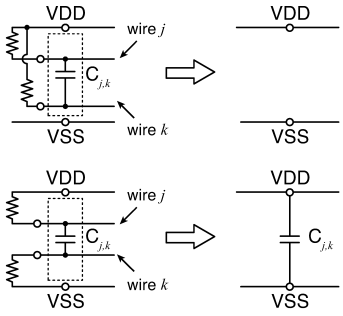
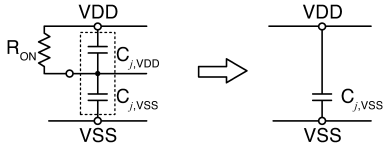


Fig. 2 On-chip capacitances.



(a) Capacitances between pairs of signal wires.



(b) Capacitances between power line and signal wire.

Fig. 3 Contribution of wire capacitances to the total capacitance.

Here, $C_{j,k}$ is a coupling capacitance between wires j and k. n_j is the number of wires that have connection of capacitance with the wire j, and n_{wire} is the number of all wires in a circuit. b_j is the logical state of the wire j, which is either '1' or '0.' Here, b_j is 1(0) for VDD (VSS) wire. Hence, $C_{j,k}$ is either $C_{j,k}$ or zero depending on the states of the signal wires. Accordingly, C_{wire} is considered state-dependent.

2.1.2 Standard Cell Capacitance

Capacitance of a transistor is composed of gate- and junction-related capacitances, both of which depend on bias voltages. Given one input pattern for a standard cell, voltages of all internal nodes inside the cell are determined. The parasitic capacitances in the transistors are also state-dependent but they result from a different cause from the case of wire capacitance.

The total capacitance C_{cell_i} of a standard cell i is the sum of capacitances associated with the transistors and wires inside the logic cell. For a cell with n input pins[†], C_{cell_i} takes one of 2^n different values. Because the location of capacitors and their values are different depending on the transistor models, we run SPICE simulation to obtain C_{cell_i} from VDD-VSS impedance.

[†]In case of sequential logic cells, inner states are also required, as will be described in Sect. 3.1

2.2 Electrical Connections of Capacitances

With low-frequency assumption where on-resistances of transistors are much smaller than the impedance of capacitors, on-resistances are considered as short. With the pre-calculated logic analysis, the signal wires can be regarded to have either VDD or VSS potential. Then, all capacitance elements, C_{cell_i} and $C_{j,k}$, are connected in parallel between VDD and VSS. The total capacitance is now just a summation of the parallel capacitances.

3. State-Dependent Capacitance Calculation

In this section, we describe the flow to calculate the state-dependent capacitance. The flow is roughly divided into two phases: at phase 1, we calculate C_{cell} and C_{wire} according to Sects. 3.2 and 3.3, respectively. Then in phase 2, we add all the capacitances. Before the capacitance calculation, we must prepare capacitance look-up table (CLUT) used for C_{cell} calculation. The CLUT can be commonly used for the designs that utilize the same standard cell library. Thus, the CLUT generation is a one-time procedure for a particular process.

Figure 4 shows the proposed capacitance calculation flow. Solid boxes are the calculation process. Dashed box shows the CLUT generation. Note here that the all processes for capacitance calculation are $O(n_{\text{cell}})$, so the time complexity of the proposed method is linear to n_{cell}.

3.1 Capacitance Look-Up Table Generation

We first prepare the CLUT that records capacitances of all input-states for all standard cells. First, we obtain SPICE netlist of the standard logic cell that includes the following elements.

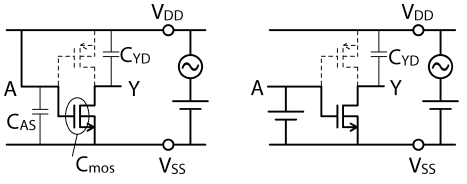
- MOS transistor geometries (diffusion area and perimeter) to calculate diffusion capacitances.
- Wire capacitance inside the logic cell. The wire capacitors are in between
  - signal wires,
  - signal and power (VDD or VSS) wires, and
  - power wires (this includes well capacitance).

Next, using the obtained netlist, frequency domain SPICE analysis for a standard cell is conducted to obtain impedance between VDD and VSS. The capacitance of a logic cell is calculated from the imaginary part of the impedance [2]. We run these analyses and capacitance calculation over all standard cells and all input vector combinations.

In the above analysis, it is critically important to connect input pins to either VDD or VSS depending on the input logic state (Fig. 5(a)). Use of independent voltage sources to set input voltages, as in Fig. 5(b), results in wrong calculation. Because the independent voltage source is AC short, capacitance of pMOS transistor (C_{mos}) and coupling capacitance between wires A and VSS (C_{AS}) are unintentionally eliminated by shorting both terminals of the capacitances. This becomes a source of error. Considering the connection inside the driving logic cell, the node has to be connected to either VDD or VSS.

A procedure to determine inner-states

Capacitance of a sequential cell depends on both input states and the stored states. Thus, the state node inside the sequen-



(a) recommended connection (b) wrong connection

Fig. 5 Simulation bench for look-up table generation.

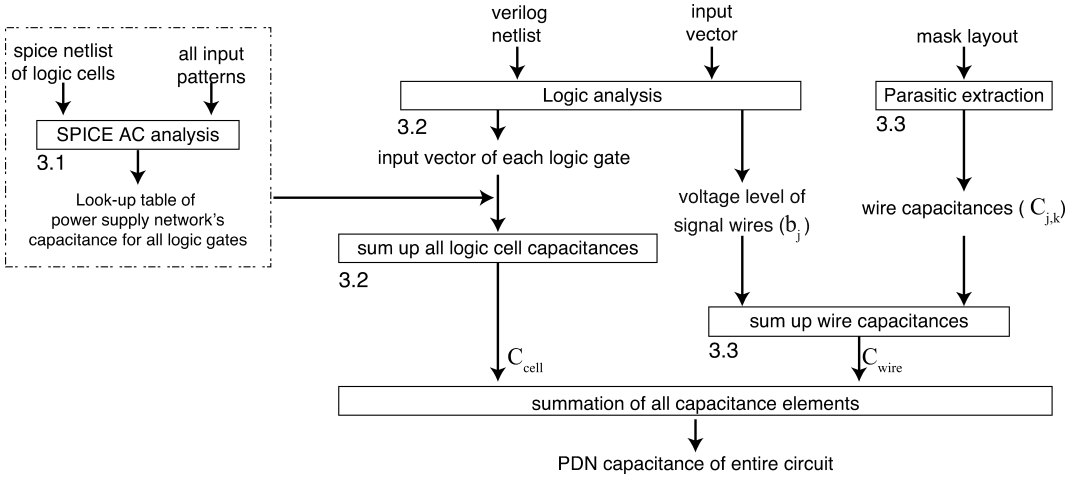


Fig. 4 Proposed capacitance calculation flow.

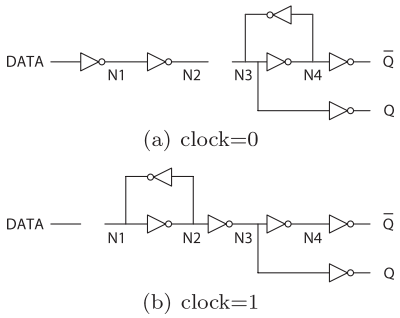


Fig. 6 A flip-flop example.

tial cell has to be added as an index of the capacitor table in the CLUT. Automatic determination of the *key node*, with which all states of the nodes inside the sequential circuit can be determined, is important for automated CLUT generation. The key node has to be recorded as an index in the CLUT. Figure 6 shows a simple example of a schematic of D flip-flop (D-FF). Node voltages of N3, N4 and Q in Fig. 6(a) are undetermined by the given input state (DATA). Disconnection of the transistor switch, which is not illustrated for simplicity, prevents from downstream propagation of the logic values. In Fig. 6(b), states of all nodes except for DATA are undetermined. Such nodes are called *floating nodes* bellow. The problem here is that "how we can find the key node(s) that determines the states of all nodes of a set of floating nodes." Generally, the number of key nodes is one for a one-bit FF, because the number of key node corresponds to the number of feed-back loop in the cell.

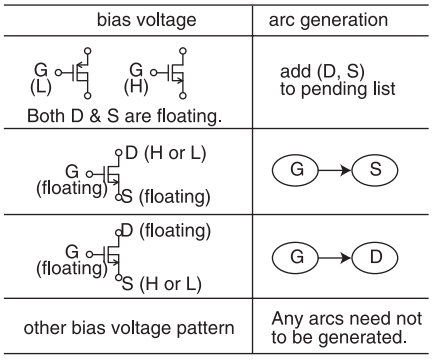
The key node can be found through the following procedure. In this procedure, no prior knowledge of the functionalities of input pins is necessary. We first give an input pattern to propagate logic states according to the connection of MOS transistors. If there are any floating nodes, we move on to the key node to determine the voltages of those nodes. A directed acyclic graph is first formed. The nodes in the graph are the floating nodes, and the arc corresponds to a transistor and indicates voltage-determination dependency between nodes. The propagation of the node voltages is represented by the directed graph. The following pseudo code describes the graph generation flow. Figure 7 shows its graphical explanation. The node that has no predecessors is the key node. By treating the key node in addition to the input nodes, all node voltages are determined and the state-dependent capacitance can be calculated. Figure 8(a) is a part of transistor-level schematic diagram of Fig. 6(a) and Fig. 8(b) shows the generated graph. The node N3 is not a floating node and thus is not included in the graph, because its state is propagated from the input pin DATA. From Fig. 8(b), the key node for the D-FF is N3 when CLK pin is logical low.

In the actual circuit, a key node corresponds to an output node. Considering the use of the CLUT, it is more convenient to use output node as an index rather than using the key node directly. Being an output node as an index of the CLUT, the implementation of the capacitance calculation in the next subsection becomes simple.

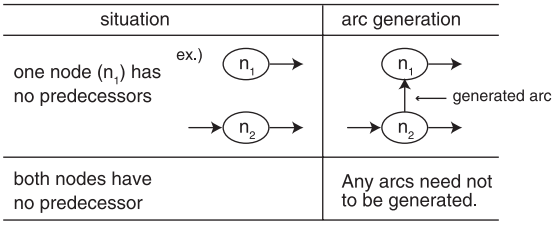
Algorithm 1 The graph generation flow.

```
for all transistor tr in the target circuit do
    g:= tr's gate node
    s:= tr's source node
    d:= tr's drain node
    if g is floating then
        if s is floating and d is not floating then
            connect an arc from g to s
        else if d is floating and s is not floating then
            connect an arc from g to d
        end if
    else if (tr is pMOS AND g is logical high)
                OR (tr is nMOS AND g is logical low) then
        append (d, s) to pending_list
    end if
end for
for all (n1, n2) in pending_list do
    if n1 has no predecessors AND n2 is unreachable from n1 then
        connect an arc from n2 to n1
    else if n2 has no predecessors AND n1 is unreachable from n2 then
        connect an arc from n1 to n2
    end if
end for
```

step 1. Generate arcs according to the bias voltages of transistors



step 2. Generate arcs corresponding to the pairs of nodes in pending list



**If there are any paths from n2 to n1, when generating an arc from n1 to n2, its generation must stop.

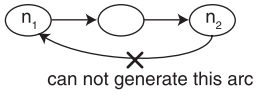
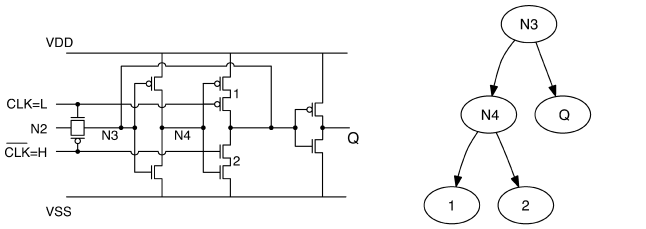


Fig. 7 A graph generation steps.

3.2 Capacitance Calculation of Standard Cell

We begin C_{ckt} calculation with a logic simulation using built-in logic simulator for a given input pattern (includ-



(a) A part of schematic of Fig. 6 (a) (b) The graph generated from (a)

Fig. 8 Example of a floating gate dependency graph.

ing the output pins of flip-flops) to obtain voltage level of all signal wires. Logic-state propagation only for the combinational circuits is necessary because the outputs of the sequential circuits are explicitly given as the input pattern. In the logic simulation, a circuit graph, which is again a directed acyclic graph, is composed according to the logic-dependence. Then, the nodes are topologically sorted to determine node order for efficient logic state propagations. Calculation complexities of both operations are proportional to n_{cell}.

With the logic simulation, input-states of all standard cells are obtained. The capacitances of the standard cells are now obtained by consulting the CLUT with calculated logic states of the pins. Finally, we add all capacitances to obtain the total cell capacitance, C_{cell}.

3.3 Wire Capacitance Calculation

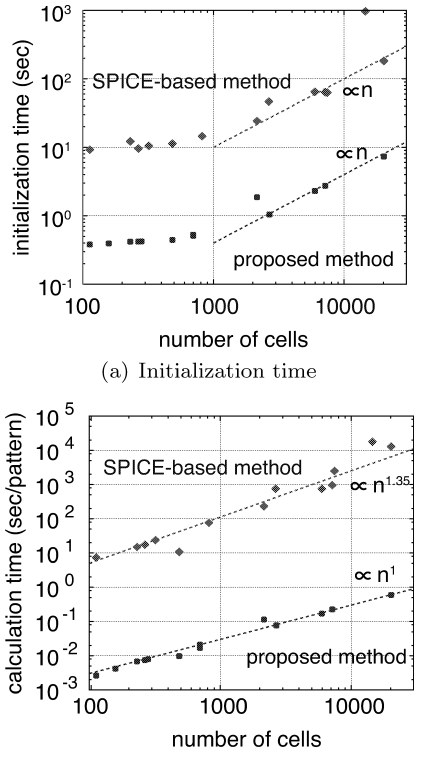
We obtain wire capacitances ($C_{j,k}$ in Eq. (1)) from a mask-layout of a circuit by parasitic extractor. States of each wire, b_j, are obtained by the above logic simulation. Using Eq. (1), C_{wire} is calculated. The calculation complexity of Eq. (1) is $O(n_{\text{near}} \times n_{\text{wire}})$. Here, n_{wire} is approximately proportional to n_{cell}. $n_{\text{near}\,j}$ is a constant bounded by about 20 that is independent of n_{cell} since capacitance is shielded by the wires nearby. Thus, C_{wire} calculation is also $O(n_{\text{cell}})$.

4. Numerical Experiments

All calculations in this section are based on the routed layouts using a standard cell library of 0.18 μm CMOS process. Power supply voltage is 1.8 V. The capacitance is defined at the frequency of 1 MHz. In the higher frequency, the effect of resistances connected in parallel with capacitors becomes larger, which makes the capacitance calculation from the imaginary part of impedance overestimated. Therefore, the capacitance calculations are conducted in low frequency.

4.1 Comparison with SPICE-Based Method

We compare calculation time and accuracy of the proposed method with SPICE frequency domain analysis. The bench mark circuits are from ISCAS89 [5]. We implemented the proposed procedure in scripting language, Python. Experiments are run on a computer with four Dual-Core AMD Opteron (3 GHz) CPUs and 64 GB RAM memory.



(a) Initialization time

(b) Calculation time per pattern

Fig. 9 Time scalability of the proposed method.

4.1.1 Calculation Time

Figure 9(a) shows initialization time versus circuit size, i.e. the number of standard cells n_{cell}. In this evaluation, time for SPICE netlist generation using parasitic extractor is excluded. Time for CLUT generation† is also excluded from the result of the proposed method. Initialization time of the proposed method includes reading files, circuit graph composition and topological sort. The initialization time is constant when n_{cell} is small in which read-in time of the CLUT is dominant. Initialization time of SPICE includes reading files, model parameter calculation, and circuit matrix composition. It is also linear to the number of logic cells.

Figure 9(b) shows capacitance calculation time for one input-pattern. Calculation time of the proposed method is linear to n_{cell} as explained in Sect. 3. On the other hand, calculation time of SPICE is approximately in proportion to ${n_{\text{cell}}}^{1.35}$ which is determined by the sparse matrix inversion. For the circuit of 20k gates, the number of input-patterns that the proposed method can calculate is four orders magnitude larger than SPICE-based method.

4.1.2 Accuracy

Figure 10 compares capacitances calculated using a SPICE-

†8 hours and 15 minutes for 467 standard cells.

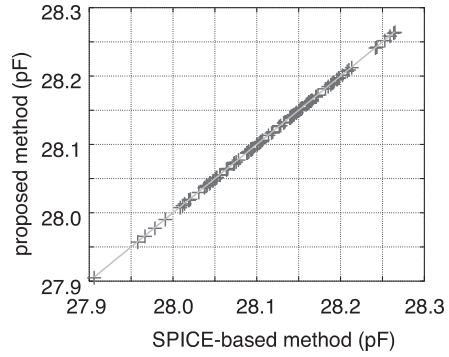


Fig. 10 Circuit capacitance calculation results comparison.

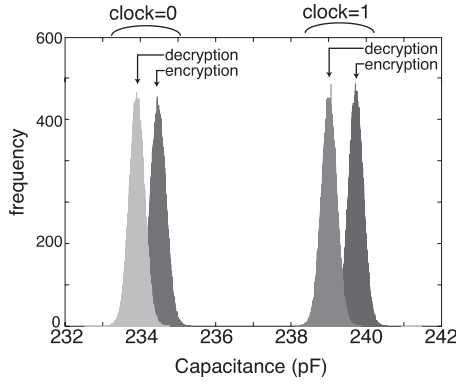


Fig. 11 Capacitance distribution of single DES IP core.

based method and the proposed method. The bench mark circuit is s15850 ($n_{\text{cell}} = 2681$). In Fig. 10, the vertical and the horizontal axes represent capacitances by the proposed method and that of SPICE-based method, respectively. Each point corresponds to the calculation time for one input pattern. The capacitances between the proposed method and SPICE agree completely (up to 5 digits).

4.2 Estimation of Circuit Capacitance Range

The proposed method substantially reduces capacitance calculation time without compromising accuracy. However, the possible input pattern, which is exponential to the total number of input pins and internal states, is still too large to calculate exhaustively. Obtaining a capacitance range, or the maximum and the minimum capacitances, is almost impossible even for a moderate-sized circuit. Because the total circuit capacitance is obtained by adding small capacitance components, its distribution is considered as normal by the central limit theorem. When the circuit capacitance is modeled by Gaussian, the maximum and the minimum capacitances can be estimated by using two statistical parameters, average and variance. In this section, we propose a quick way to estimate these parameters.

Figure 11 shows capacitance distribution of a single DES IP core [6]. The figure is obtained by Monte Carlo calculation of 10^6 samples. There are four peaks. Each peak corresponds to different state combinations between clock signal and mode (encryption or decryption) selector pin. Large capacitance differences occur among the state combinations. The reason of the difference is explained as follows. The capacitance of flip-flop is different according to the clock state. In circuits containing many flip-flops, capacitances of flip-flops change synchronously with 100% correlation because they are connected to the same clock signal. As a result, circuit capacitance changes dramatically according to the state of the clock state. We call such input or internal pins that greatly change the total circuit capacitance as "influential pin." The mode signal is the other influential signal for the DES IP core. As can be seen in the figure, four distributions have similar shape, implying that the above two signals are dominantly influential and the other pins and internal states behave more randomly. The capacitance distribution of different influential pin state has to be analyzed separately since each combination potentially forms different Gaussian distribution.

4.2.1 Average of Circuit Capacitance

The probabilities of a node being in either logical high or low are uneven. Hence, an average of circuit capacitance, $\overline{C_{ckt}}$, can not be approximated just as a sum of averages of each standard cell's capacitance. The state probabilities for all nodes have to be correctly considered to estimate $\overline{C_{ckt}}$ accurately. Let $P(Y = 1)$ be the probability that the node Y which is the output pin of a standard cell is logical high.

$$P(Y = 1) = \sum_{\boldsymbol{s}} P(\boldsymbol{x} = \boldsymbol{s}) \cdot f(\boldsymbol{s}) \tag{2}$$

$$P(\boldsymbol{x} = \boldsymbol{s}) = \prod_{i=1}^{n_{bit}} P(x_i = s_i) \tag{3}$$

Here, $\boldsymbol{x} = (x_0, \cdots, x_{n_{bit}})$ is the input vector of a standard cell, n_{bit} is the dimensions of $\boldsymbol{x}$, and $\sum_{\boldsymbol{s}}$ means summation for all possible input vectors $(0, 0, \cdots, 0)$ to $(1, 1, \cdots, 1)$. $f(\boldsymbol{s})$ is output state when the input vector is $\boldsymbol{s}$. $P(Y = 0)$ is obtained as follows.

$$P(Y = 0) = 1 - P(Y = 1) \tag{4}$$

To obtain state probabilities for all nodes, we assign $P(Y = 1) = P(Y = 0) = 0.5$ as the state probabilities of input pins and flip-flop output. Either 1.0 or 0.0 are assigned for influential pins. The probabilities of all other nodes whose states are not assigned are calculated by propagating probabilities using Eqs. (2) and (4).

Estimation of $\overline{C_{ckt}}$ using state probabilities becomes following.

$$\overline{C_{ckt}} = \sum_{i}^{n_{gate}} \overline{C_{cell_i}} + \sum_{j}^{n_{\text{wire}}} \overline{C_{wire_j}} \tag{5}$$

$$\overline{C_{cell_i}} = \sum_{\boldsymbol{s}} P(\boldsymbol{x} = \boldsymbol{s}) \cdot C_{cell_i}(\boldsymbol{s}) \tag{6}$$

$$\overline{C_{wire_j}} = \sum_{k}^{n_j} P(b_j \neq b_k) \cdot C_{j,k} \tag{7}$$

$P(b_j \neq b_k)$ is obtained as follows.

$$P(b_j \neq b_k) = P(b_j = 1) \cdot P(b_k = 0) + P(b_j = 0) \cdot P(b_k = 1) \tag{8}$$

Time required to calculate $\overline{C_{ckt}}$ by the above equations is also linear to the number of cells.

We compared the proposed average calculation with Monte Carlo simulations of 10^4 samples. The relative errors between the proposed estimations by Eqs. (5)–(7) and that of Monte Carlo results are less than 0.5%. On the other hand, assuming state probabilities of 0.5 to all nodes in a circuit, which is the assumption in conventional estimation, resulted in the maximum relative error of 1.8%.

4.2.2 Variance of Circuit Capacitance

The variance of C_{ckt}, ${\sigma_{ckt}}^2$ is obtained as follows.

$$\sigma_{ckt}^2 = \sum_{i}^{n_{gate}} {\sigma_{cell_i}}^2 + \sum_{j}^{n_{\text{wire}}} \sum_{k}^{n_j} {\sigma_{C_{j,k}}}^2 \tag{9}$$

Here, ${\sigma_{cell_i}}^2$ is the variance of C_{cell_i} and ${\sigma_{j,k}}^2$ is the variance of a wire capacitance between wires j and k. The covariances between standard cells' capacitances and wire capacitances are ignored because there are both positive and negative covariances and they cancel out each other. ${\sigma_{cell_i}}^2$ is calculated by

$${\sigma_{cell_i}}^2 = \sum_{\boldsymbol{s}} (C_{cell_i}(\boldsymbol{s}) - \overline{C_{cell_i}})^2. \tag{10}$$

A wire capacitance between wires j and k becomes $C_{j,k}$ or 0 and its average is $C_{j,k}/2$. Hence, ${\sigma_{j,k}}^2$ is

$${\sigma_{C_{j,k}}}^2 = \frac{{C_{j,k}}^2}{4}. \tag{11}$$

Calculation time of variance estimation is also linear to the number of cells.

We again compared the variance estimation by Eq. (9) with that of Monte Carlo simulations. The maximum relative error of σ_{ckt} is 33% and the average is 11%. The error is larger for small circuits. The circuits that have more than 5,000 gates have 12% error at the maximum. The average is 8%.

4.2.3 Circuit Capacitance Range Calculation

Using the above equations, we obtain parameters to define Gaussian distributions. Let n_{inf} be the number of influential pins. With the state combination of the influential pins, there are $2^{n_{\text{inf}}}$-averages and $2^{n_{\text{inf}}}$-variances. The maximum capacitance C_{max_l} and the minimum capacitance C_{min_l} are estimated using the average $\overline{C_{ckt_l}}$ and the standard deviation σ_{ckt_l} as follows ($l = 1, 2, \cdots, 2^{n_{\text{inf}}}$).

$$C_{max_l}, C_{min_l} = \overline{C_{ckt_l}} \pm \sigma_{ckt_l} \sqrt{2(n_p \log 2 - \log(\sigma_{ckt_l}\sqrt{2\pi}))} \tag{12}$$

Here, n_p is the number of input pins and internal states excluded the influential pins. The maximum capacitance of a circuit, C_{max}, is $\max_l(C_{max})$. The minimum capacitance $C_{min} = \min_l(C_{min})$. The capacitance range $\Delta C = C_{max} - C_{min}$ normalized by the nominal value $C_{nom} = (C_{max} + C_{min})/2$ are 8–17% for ISCAS89 benchmark circuits and DES IP core. Because resonant frequency of a PDN system is proportional to $1/\sqrt{LC}$, circuit capacitance fluctuation of 8% and 17% means that resonant frequency varies 4% and 8%, respectively. It is important to consider the change of on-chip PDN capacitance when we simulate the behavior of a PDN system.

5. Conclusion

This paper proposed a calculation method for analyzing capacitance of power distribution network considering state dependence. Linear time complexity to the number of gates has been achieved by the use of capacitance look-up table and the built-in logic simulator. The calculation accuracy is almost identical to SPICE simulation. Proposed method enables accurate and fast capacitance calculation considering the state-dependency of the logic circuit. The calculation of the maximum and the minimum of circuit capacitances without doing Monte Carlo simulations are also proposed. The capacitance fluctuation due to state-dependence has the range of 8–17%., which corresponds to the range of of resonant frequency variation of 4–8%. We should consider the circuit capacitance fluctuation when we simulate the behavior of a PDN system.

Acknowledgment

This work was partially supported by NEDO and Special Coordination Funds for Promoting Science and Technology, and VLSI Design and Education Center (VDEC) in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

References

[1] H. Osaka, D. Tanaka, O. Wada, Y. Toyota, and R. Koga, "Linear equivalent circuit and current source for I/O (LECCS-I/O) modeling of IC power current for EMI simulation," J. Japan Institute of Electronics Packaging, vol.7, pp.517–524, 2004.
[2] S. Hayashi, F. Minami, and M. Yamada, "Analysis method of dynamic IR-Drop," 16th Workshop on Circuits and Systems in Karuizawa, pp.49–54, April 2003.
[3] S. Bobba and I. Hajj, "Input vector generation for maximum intrinsic decoupling capacitance of VLSI circuits," Proc. IEEE International Symposium on Circuits and Systems, vol.5, pp.195–198, May 2001.
[4] M. Badaroglu, G. Van der Plas, P. Wambacq, L. Balasubramanian, K. Tiri, I. Verbauwhede, S. Donnay, G. Gielen, and H. De Man, "Digital circuit capacitance and switching analysis for ground bounce in ICs with a high-ohmic substrate," IEEE J. Solid-State Circuits, vol.39,

no.7, pp.1119–1130, July 2004.
[5] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," Proc. IEEE International Symposium on Circuits and Systems, vol.3, pp.1929–1934, May 1989.
[6] R. Usselmann, "DES/triple DES IP cores," http://www.opencores.org/projects.cgi/web/des/overview, May 2007.

Shiho Hagiwara received a B.E. and M.E. degrees in Electrical and Electronic Engineering from Tokyo Institute of Technology, Japan, in 2006 and 2008, respectively. She is studying toward an Ph.D. degree in Department of Electronics and Applied Physics, Tokyo Institute of Technology. Her research interests include CAD algorithms for LSI design and design for manufacturing.

Koh Yamanaga received B.E. and M.E. degrees in Materials Science and Engineering from Kyushu University, Fukuoka, Japan in 2001, 2003, respectively, and a Ph.D. degree in Electronics and Applied Physics from Tokyo Institute of Technology in 2010. In 2003, he joined Murata Manufacturing Co. Ltd., where he has been engaged in research on electromagnetic compatibility (EMC) for power distribution network.

Ryo Takahashi received the B.E. degree in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2009. He is currently working toward the M.S. degree at University of Tokyo, Tokyo, Japan.

Kazuya Masu received the B.E., M.E. and Ph.D. degrees in Electronics Engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1977, 1979 and 1982, respectively. He was with the Research Institute of Electrical Communication, Tohoku University, Sendai, Japan since 1982. Since 2000, he has been with Precision and Intelligence Laboratory, Tokyo Institute of Technology, Yokohama, Japan and is currently a professor in Integrated Research Institute, Tokyo Institute of Technology, Yokohama, Japan. He was a visiting Professor in Georgia Institute of Technology in 2002 and 2005. His current interests are signal integrity and GHz signal propagation in multilevel interconnect of Si ULSI, reconfigurable RF circuit technology, performance evaluation and prediction based on interconnect wire length distribution, and BEOL process technology. He is a member of the IEEE, the Japan Society of Applied Physics (JSAP), the Institute of Electrical Engineers of Japan, and the Electrochemical Society.

Takashi Sato received the B.E. and M.E. degrees from Waseda University, Tokyo, Japan, and the Ph.D. degree from Kyoto University, Kyoto, Japan. He was with Hitachi, Ltd., Tokyo, Japan from 1991 to 2003, with Renesas Technology Corp., Tokyo, Japan, from 2003 to 2006, and with Tokyo Institute of Technology, Yokohama, Japan. In 2009, he joined the Graduate School of Informatics, Kyoto University, Kyoto, Japan, where he is currently a professor. He was a visiting industrial fellow at the University of California, Berkeley from 1998 to 1999. His research interests include CAD for nanometer-scale LSI design, fabrication-aware design methodology, and performance optimization for variation tolerance. Dr. Sato is a member of IEEE. He received the Beatrice Winner Award at ISSCC 2000 and the Best Paper Award at ISQED 2003.