

Optimal Design of the Power Delivery Network for Multiple Voltage-Island System-on-Chips

Behnam Amelifard and Massoud Pedram, *Fellow, IEEE*

Abstract - This paper introduces techniques for power efficient design of power delivery network in multiple voltage-island System-on-Chip (SoC) designs. The first technique is targeted to SoC designs with static voltage assignment, while the second technique is pertinent to SoC designs with dynamic voltage scaling capability. Conventionally a single level configuration of DC-DC converters, where exactly one converter resides between the power source and each load, is used to deliver currents at appropriate voltage levels to different loads on the chip. In the presence of dynamic voltage scaling capability, each DC-DC converter in this network should be able to adjust its output voltage. In the first part of this paper it is shown that in a SoC design with static voltage assignment, a multi-level tree topology of suitably chosen DC-DC converters and voltage regulators between the power source and loads can result in higher power efficiency in the power delivery network. The problem is formulated as a combinatorial problem and is efficiently solved by dynamic programming. In the second part of the paper, a new technique is presented to design the power delivery network for a SoC design to support dynamic voltage scaling. In this technique the power delivery network is composed of two layers. In the first layer, DC-DC converters with fixed output voltages are used to generate all voltage levels that are needed by different loads in the SoC design. In the second layer of the power delivery network, a power switch network is used to dynamically connect the power supply terminals each load to the appropriate DC-DC converter output in the first layer. Experimental results demonstrate the efficacy of both techniques.

Index Terms— Low-power design, power delivery network, voltage regulator, DC-DC converter, energy efficiency, voltage island, system-on-chip

I. INTRODUCTION

THE power delivery network (PDN) is a critical design component in system-on-chip (SoC) designs. A robust PDN is required to achieve a high level of power supply integrity. If improperly designed, this network could be a major source of noise, such as IR-drop, ground bounce, and electromagnetic interference (EMI) [1]. In today's high-performance microprocessors, it is typical for the circuit to draw over 100A current from the PDN in a fraction of nano-second, yielding the derivative of the current over 100 GA/s. With careful design, however, the PDN can tolerate large

variations in load currents while maintaining the supply voltage level across the chip within a desired range [2].

Emerging low-power design techniques have made the design of PDN an even more challenging task. More precisely, multiple voltage domains are being introduced on the System-on-a-chip (SoC) in order to minimize the overall power dissipation of the system while meeting a performance constraint. This means that it is possible to have multiple relatively small logic blocks operating at different voltages. The voltage of each logic block may be fixed or change dynamically based on workload monitoring. This is also known as the multiple voltage island approach [3]. In these systems, it is required that the PDN delivers power at appropriate voltage levels to different functional blocks (FB's) while incurring the minimum power loss. A typical PDN design methodology for a high-performance SoC comprises of three steps:

- Establishing a target impedance to be met across a broad frequency range for the PDN,
- Designing a proper system-level decoupling network i.e., specifying components to meet that impedance,
- Selecting the right voltage regulator modules¹.

A target impedance level (in terms of the magnitude of impedance) of several 10s or 100s of milliohms is usually established. Decoupling capacitors are used to try to achieve this target impedance level up to frequency that is at least several times the clock frequency.

A. Establishing Target Impedance of the PDN

A methodology for designing a good PDN is to define a target impedance for the network that should be met over a broad frequency band [4]. This parameter can be computed by assuming $\alpha\%$ allowable ripple (noise) on the supply voltage, V_{dd} , and using value of the maximum switching current drawn by the circuit, I_{peak} . The target impedance can then be calculated as [5]:

$$Z_{target} = \frac{\alpha\% \times V_{dd}}{I_{peak}} \quad (1)$$

The voltage requirements of Intel Pentium 4 microprocessor's core are 1.5 V and 1.327 V for V_{max} and V_{min} , respectively. According to the electrical specification of the processor, the maximum current demand is 50A with a slew rate of 200A/ μ s, while the power consumption is 70 W at the operating clock

B. Amelifard was with the Department of Electrical Engineering-Systems, University of Southern California. He is currently with Qualcomm Inc., San Diego, CA 92122 USA (e-mail: behnama@qualcomm.com).

M. Pedram is with the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, CA 90089 USA (e-mail: pedram@ceng.usc.edu).

¹ "DC-DC converter" and "voltage regulator module" are used interchangeably throughout this paper.

frequency of 2GHz. The power delivery system should be designed to have its impedance less than or equal to the target impedance of $2m\Omega$ from DC up to 2GHz. As another example, for a complex high-performance design done in the 65nm node and a supply voltage of 1.1V, the peak power dissipation is 104 Watts [6], and therefore, $I_{peak} = 104/1.1 = 94A$. If 5% ripple is allowed on the voltage supply, the calculated target impedance will be $0.6m\Omega$. From the general scaling theory, the clock frequency f_{clk} and current demand I_{peak} are increasing, while the power supply voltage V_{dd} is decreasing. Therefore, to satisfy the power supply noise constraint, the target impedance of the power supply is expected to decrease while it must be met over a wider frequency range.

B. Designing a Proper System-Level Decoupling Network

Since the current drawn by digital circuits can change suddenly with different frequencies, the target impedance should be met over a broad frequency range to guarantee the ripple on the voltage supply does not exceed the allowable value. To meet this requirement, on-chip and off-chip decoupling capacitors (decaps) must be suitably placed in the design. Decaps play an important role in the PDN because they act as charge reservoirs providing instantaneous current for switching circuits. Current surface-mount ceramic capacitors provide good IC decoupling up to around 100-300MHz [7]. Decoupling in higher frequencies can be achieved by deploying on-chip capacitors. The amount of on-chip capacitance that can be added is limited to the real estate on-chip. Fabrication data demonstrate that for 90nm technologies, tunneling gate leakage of a 1nF decap is in the order of milliamperes [8] and for more advanced CMOS process technologies it is expected to be even higher. Therefore, the leakage current of the decaps adds to the total power consumption of the circuit and shortens the battery lifetime. These facts emphasize that to achieve a low-power and low-cost design, the added decap should be minimized. In the past, much research has been conducted to address the problem of decap allocation. In [9], for example, the problem of decap allocation during initial floorplanning stage was formulated as a linear program. In [10] the authors proposed a technique for sizing and placing decaps in a standard cell layout. In [11] the authors presented a multigrid-based technique for simultaneously optimizing the power grid and decap allocation. With the aid of macromodeling and the concept of an effective radius of a decap, the authors of [12] proposed an efficient charge-based method for decap allocation.

C. Selecting the Right Voltage Regulator Modules

Every electronic circuit is designed to operate off of some supply voltage, which is usually assumed to be relatively constant, e.g., 1.2V with $\pm 5\%$ ripple. A voltage regulator module (VRM) provides this substantially constant DC output voltage regardless of changes in load current or input voltage (this statement assumes that the load current and input voltage are within the specified operating range for the part). Assume that the range of input voltages and load currents over which a

regulator can maintain a target voltage level within the specified tolerance band (e.g., 1.3V with $\pm 5\%$ ripple) has been specified. The regulator's power efficiency is calculated as the ratio of the power that is delivered to the load to the power that is extracted from the input source, i.e.,

$$\eta = \frac{V_{out}I_{out}}{V_{in}I_{in}} \quad (2)$$

Power efficiency is one of the most important figures of merit for a voltage regulator and is a function of the input voltage and output current of the VRM. Fig. 1 shows the efficiency of a commercial VRM as a function of the input voltage and output current.

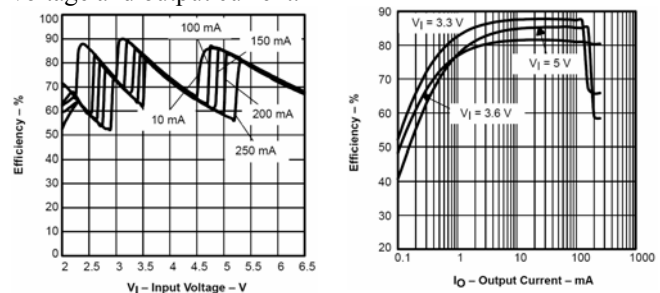


Fig. 1. The efficiency of TPS60503 as a function of input voltage and output current [13].

Each VRM has an associated cost which depends on its complexity, silicon area, and passive element costs. For example, because of their inductors, regulated inductor-based VRM's are usually the most expensive type of DC-DC converters. Linear regulators, on the other hand, are typically the least expensive ones.

In a complex SoC design, there are many functional blocks (FB's) providing various functionality. Examples of processing elements are DSP or CPU cores. Examples of other FB's are interface blocks, MPEG encoder/decoder blocks, RF front-end, on-chip memory, and various controllers. Each of these functional blocks has different voltage and current requirements which have traditionally been met by utilizing one or more off-chip VRM's. In a multi-voltage SoC, however, keeping the VRM's off-chip not only increases the total cost of the system, but also increases the system size, lowers the system reliability, and creates more rigid requirements on the VRM due to losses on the board. On the other hand, one of the main advantages of deploying on-chip regulators is that because the VRM's are located close to the load, the impedance between each VRM and its load becomes smaller, resulting in lower noise injection on the power supply lines [2]. Consequently, utilizing on-chip voltage regulators have become attractive for low-power applications, particularly in compact handheld devices [8] [14].

Fig. 2 depicts the role of VRM's in providing appropriate voltage levels to different FB's on a "static" voltage-island SoC. Typically a (single-level) star topology of VRM's, where only one converter resides between the power source and each load, is used to deliver currents with appropriate voltage levels to different loads on the chip (c.f. Fig. 2). In the first part of this paper we show that using a (multi-level) tree topology¹ of

¹ The graph representation of the VRM network has a tree structure, i.e., no VRM can be driven by more than one other VRM.

suitably chosen VRM's between the power source and loads yields higher power efficiency in the PDN. We formulate the problem of selecting the best set of regulators in a VRM tree topology as a dynamic program and efficiently solve it.

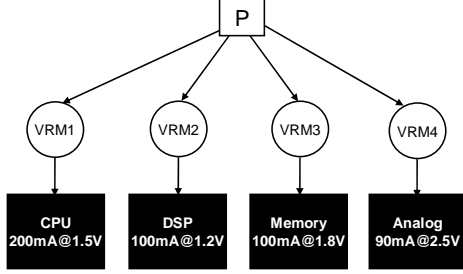


Fig. 2. The role of VRM tree in providing appropriate voltage level for each FB.

In the conventional technique to support DVS for different FB's, which is depicted in Fig. 3, each FB has its own VRM with multiple output voltage levels [15, 16]. The power manager selects the supply level that VRM_i provides to the FB_i. In the second part of the present paper we show that this architecture, despite its simplicity, has several shortcomings and propose a new technique to address the problem of PDN design to support dynamic voltage scaling. Preliminary results of this paper have been appeared in [17, 18].

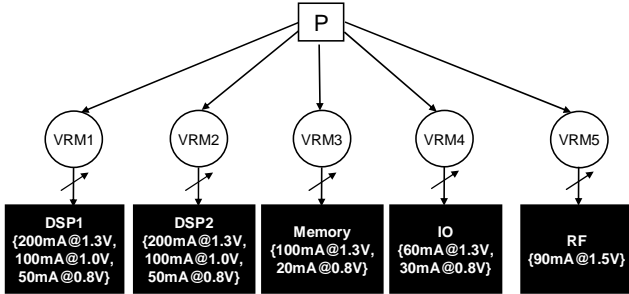


Fig. 3. The role of VRM tree in providing appropriate voltage level for each FB on a SoC with DVS option. The output voltage of each VRM is changed dynamically.

The remainder of this paper is organized as follows. In Section II the problem of VRM tree optimization for a static voltage-island SoC for minimum power dissipation is discussed and an efficient algorithm is proposed to optimally select the best set of regulators in the VRM tree. In Section III a new architecture for the design of power delivery network on a SoC with DVS option is proposed and an algorithm is presented to optimally select the best set of VRM's in this network. Section IV is dedicated to the experimental results, while Section V concludes the paper.

II. VRM TREE OPTIMIZATION FOR MINIMUM POWER DISSIPATION IN A STATIC VOLTAGE ISLAND SOC

The VRM tree optimization (RMTO) problem is defined as follows.

RMTO Problem: Given is:

- A library \mathcal{R} of VRM's and for each $r \in \mathcal{R}$, its output voltage $v_{r,out}$, the minimum and maximum input voltages $v_{r,in}^{\min}$ and $v_{r,in}^{\max}$, the maximum load current

$i_{r,out}^{\max}$, and the VRM efficiency η_r as a function of load

current and input voltage,

- A power source P with a nominal voltage of V_P ,
- A set \mathcal{F} of FB's, and for each $f \in \mathcal{F}$, its required voltage V_f and average current demand I_f .

The goal is to build a tree topology of VRM's that connects P to all FB's and minimizes the PDN power loss from the power source to the loads while meeting the voltage and current constraints. From here on we focus on this RMTO problem statement.¹

It should be noted that the power delivered to the FB's is independent of the topology of the VRM tree and is given by,

$$P_{FBs} = \sum_{f \in \mathcal{F}} V_f I_f. \quad (3)$$

Therefore, to minimize the power loss in the PDN from the power source to the loads, one needs to minimize the power drawn from the power supply. Given that the voltage of the power supply is fixed, the objective of RMTO problem is to minimize the current drawn from the power supply. We assume that each VRM can provide only one output voltage (multi-output VRM's are considered as multiple VRM's, each with its own fixed voltage output).

Although RMTO problem definition does not put any constraints on the depth of the VRM tree that drives the loads, in practice, such a constraint is useful. The reason is that utilizing a VRM tree with a large number of internal levels tends to increase the number of regulators, which in turn increases their cost and chip area overhead with little (if any) benefit in terms of improving the power efficiency of the PDN. For this reason, in this work, we only consider up to two levels of regulators in the VRM tree, i.e., the (node) depth of the tree is 4, with one corresponding to the power source one corresponding to the loads and up to two internal levels dedicated to VRM's. Our solution, however, can be easily extended to handle VRM trees with higher depth.

To improve the efficiency of our solution technique by implicitly considering a large class of tree topologies under one class representative, it is convenient to introduce an ideal VRM whose efficiency is 100% and whose output voltage and thus output current are equal to its input voltage and current, respectively. This ideal VRM (really a lossless buffer) is added to library \mathcal{R} of VRM's. Note that ideal VRM's are inserted on every path from the tree root to a leaf node in the tree so that the logical depth of each such path is exactly four (cf. Fig. 4).

Definition 1: A VRM satisfies *monotone input current (MIC) property* if its input current is a monotone increasing function of its output current independent of the input voltage.

Notice that the monotone input current property may hold in spite of the non-monotone power efficiency characteristics for a VRM. This is because of the way that power efficiency is defined and its relation to input and output voltages and currents. More precisely, the monotone input current property

¹ An interesting variant of the problem, which we do not address here, is as follows. Given a cost associated with each regulator, minimize the power loss in the PDN while ensuring that the total cost of the VRM tree does not exceed a cost budget.

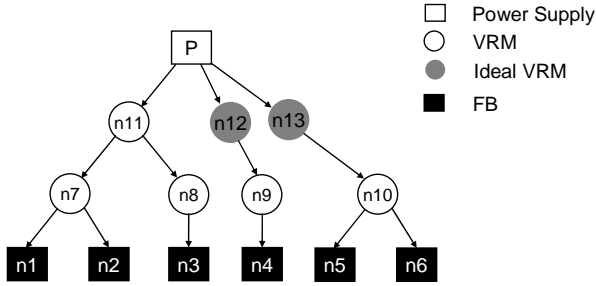


Fig. 4. A VRM tree after inserting ideal VRM's.

holds as long as the VRM has a single mode, where the basic feedback loop in the regulator which performs the output and line regulation does not change its parameters (reference voltage levels, sensing network parameters, switch configuration, etc) in response to applied input voltages. There are, however, VRM's that may operate as say 2X charge pump or 1.5X charge pump or even an LDO depending on the applied input voltage. Such VRM's tend to exhibit a non-monotone input current vs. output current behavior. In the remaining of this paper we assume that each VRM has a single mode of operation (multi-mode VRM's are considered as multiple VRM's, each with a single mode of operation).

 TABLE I
 NOTATION USED IN RMTO ALGORITHM

\mathcal{R}	Set of all VRM's including the ideal VRM
\mathcal{F}	Set of all FB's
\mathcal{U}	Set of all output voltages of the VRM's
\mathcal{V}_n	Set of candidate input voltages for node n
$\mathcal{V}_{n,r}$	Set of candidate input voltages for n when r is the VRM of n
\mathcal{C}_n	Set of candidate VRM's for internal node n of the tree
T	Topology of VRM tree
$\xi(n)$	Optimum VRM selection for node n
L_i	Set of all level i internal nodes, $i = 1, 2$
V_f, I_f	Voltage level and current demand of FB $f \in \mathcal{F}$
$v_{r,out}$	Output voltage level of regulator r
$v_{r,in}^{\min}, v_{r,in}^{\max}$	Minimum and maximum input voltage levels of regulator r
$i_{c_n,out}^{\max}$	Maximum output current of regulator r
$V_{out}(n)$	Output voltage of a node n
$I_{out,r}(n), I_{in,r}(n)$	Output and input current of node n given that regulator r is assigned to this node
$\mu_r(v_{in}, i_{out})$	Efficiency of regulator r as a function of its input voltage v_{in} and output current I_{out}
$\Psi_n(v_{in})$	One dimensional table in node n with the key v_{in} and the value of input current of node.

If the tree topology is fixed (-F option), then the selection of the appropriate regulator for each node can be done optimally by using dynamic programming starting from the leaf nodes. This algorithm, called RMTO-F, will be presented in the next section. Table I introduces notations which will be used in RMTO-F algorithm.

$$\begin{aligned}
 & RMTO-F(\mathcal{R}, \mathcal{F}, V_P, T) \{ \\
 & \text{For each second level node } n \{ \\
 & \quad \text{If } (V_{f_1} \neq V_{f_2} : f_1, f_2 \in FO(n), f_1 \neq f_2) \text{ Exit}(0); \\
 & \quad V_{out}(n) = V_f : f \in FO(n); \\
 & \quad I_{out}(n) = \sum_{f \in FO(n)} I_f; \\
 & \quad \mathcal{C}_n = \{r \in \mathcal{R} | v_{r,out} = V_{out}(n), v_{r,out}^{\max} > I_{out}(n)\}; \\
 & \quad \mathcal{V}_n = \emptyset; \\
 & \quad \text{For each } v_{in} \in \mathcal{U} \{ \\
 & \quad \quad \text{For each } r \in \mathcal{C}_n \\
 & \quad \quad \quad I_{in,r}(v_{in}) = \frac{V_{out}(n) \times I_{out}(n)}{v_{in} \times \eta_r(v_{in}, I_{out}(n))}; \\
 & \quad \quad \quad r = \arg \min_{r \in \mathcal{C}_n} (I_{in,r}(v_{in})); \\
 & \quad \quad \quad \text{If } r \neq \emptyset \{ \\
 & \quad \quad \quad \quad \Psi_n(v_{in}) = I_{in,r}(v_{in}); \\
 & \quad \quad \quad \quad \mathcal{V}_n = \mathcal{V}_n \cup v_{in}; \\
 & \quad \quad \quad \} \\
 & \quad \quad \} \\
 & \quad \} \\
 & \text{For each first level node } n \{ \\
 & \quad \mathcal{C}_n = \{r \in \mathcal{R} | v_{r,out} \in \bigcap_{i \in FO(n)} \mathcal{V}_i, v_{r,in}^{\min} \leq V_P \leq v_{r,in}^{\max}\}; \\
 & \quad \text{For each } r \in \mathcal{C}_n \{ \\
 & \quad \quad V_{out}(n) = v_{r,out}; \\
 & \quad \quad I_{out,r}(n) = \sum_{m \in FO(n)} \Psi_m(V_{out}(n)); \\
 & \quad \quad I_{in,r}(n) = \frac{V_{out}(n) \times I_{out,r}(n)}{V_P \times \eta_r(V_P, I_{out,r}(n))}; \\
 & \quad \quad \} \\
 & \quad \quad (\xi(n), \xi(m) : m \in FO(n)) = \arg \min_{r \in \mathcal{C}_n} (I_{in,r}(n)); \\
 & \quad \quad \} \\
 & \quad \} \\
 & \}
 \end{aligned}$$

Fig. 5. RMTO-F algorithm for VRM tree optimization when tree topology is fixed.

A. RMTO for Fixed Tree Topology

The algorithm for solving RMTO-F problem is shown in Fig. 5. This algorithm starts with the nodes in the second internal level of the tree T . If any such node is connected to two FB's with different input voltage requirements, then the tree will not be a feasible VRM tree (a precise definition is provided later) and the algorithm terminates; otherwise, the output current of the node is calculated as the sum of the current demands of all leaf nodes (FB's) that are connected to it. Next all candidate VRM's with compatible output voltage and current characteristics are evaluated. Since the input voltage of the second-level node is not known at this time, the power efficiency of each candidate VRM for the node in question cannot be calculated directly. Furthermore because this node can be driven by any first-level VRM node, all voltage values in \mathcal{U} are enumerated. Next, for each enumerated voltage value, the power efficiency of each *matching* VRM (i.e., one that accepts the voltage value as its input voltage) is obtained from the efficiency curves for that regulator. This information is then used to compute the input current of the second-level node as the minimum of the input currents of the matching VRM's. The calculated input current is stored in a lookup

table with the key set to the input voltage of the second-level node and the value set to the input current of that same node.

The first-level nodes are visited next. For each such node n , all candidate output voltages $v_{out}(n)$ (defined as the voltages in the intersection of all \mathcal{V}_m 's, where m denotes a fanout of n) are considered. Next a set of output voltages are identified where each of these output voltages show up in input current vs. input voltage lookup tables that are stored at each fanout of n . For every such output voltage, the sum of the input currents of the driven second-level nodes is computed and set as the target output current of the first-level node. Next based on the output current of that first-level node and the known input voltage of the same node (which is the same as the output voltage of the power source for the VRM tree), the optimum VRM assignment for the first-level node is determined by enumerating all possible VRM's that match at that node, i.e., a VRM assignment is chosen that minimizes the input current of the first-level node (and hence the output current demand on the power source along the edge that leads to that node) while providing the output current needed by driven second-level nodes under the selected output voltage assignment for the first-level node.

1) Complexity Analysis

Lemma 1: The complexity of RMTO-F algorithm is $O(|\mathcal{R}|^2 |\mathcal{F}| \lg |\mathcal{F}|)$.

Proof: The worst case running time of the algorithm is when there are $|\mathcal{F}|$ second level nodes and only one first level node in the VRM tree. In this case, the complexity of building the lookup tables for the second-level nodes is $O(|\mathcal{U}| |\mathcal{R}| |\mathcal{F}|)$. When the first level node is visited, the algorithm needs to find all candidate output voltages for this node. This requires computing the intersection of candidate input voltages in the fanout nodes of the first-level node. Since there are $|\mathcal{F}|$ sets of candidate voltages and each set contains at most $|\mathcal{U}|$ voltages, the complexity of intersection operation is $O(|\mathcal{F}| \cdot |\mathcal{U}| \lg |\mathcal{U}|)$. For each voltage in the intersection set, all candidate VRM's should be enumerated. Therefore, the complexity of finding the optimal VRM for the first level node is $O(|\mathcal{R}| |\mathcal{F}| \cdot |\mathcal{U}| \lg |\mathcal{U}|)$. Since each VRM can provide only one output voltage, $|\mathcal{U}| \leq |\mathcal{F}|$. Therefore the algorithm complexity is $O(|\mathcal{R}|^2 |\mathcal{F}| + |\mathcal{R}|^2 |\mathcal{F}| \lg |\mathcal{F}|)$, which proves the result. ■

B. RMTO for Variable Tree Topology

The optimal solution for the VRM tree problem when the tree topology may be varied (-V option) is found by enumerating all feasible trees with exactly two internal nodes and $|\mathcal{F}|$ leaf nodes.

Definition 2: A VRM tree topology is *feasible* when (i) it has an exact depth of 4, i.e., every path from the root to a leaf node comprises of a zeroth level node corresponding to the tree root, a third-level node corresponding to the leaf node, with two levels of internal nodes in between; (ii) the leaf nodes under any second-level internal node in the tree have the same voltage assignments.

Since each VRM can only provide one output voltage level, the number of VRM's in a feasible VRM tree topology cannot be less than the number of distinct voltage levels of the FB's. The number of possible combinations for the first level of the tree is the power set of the number of second-level nodes in that tree. After generating each feasible tree instance T , the RMTO-F algorithm is used to find the optimum solution for the corresponding T (c.f. Fig. 6).

1) Efficient Generation of Feasible Trees

One issue with RMTO-V procedure is that the number of feasible trees with n leaves appears to be quite large; fortunately, in the RMTO problem, many of the generated trees are isomorphic (c.f. Fig. 7).

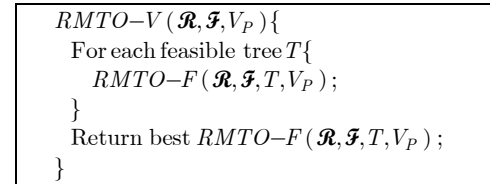


Fig. 6. RMTO-V algorithm for VRM tree optimization.

Definition 3: Two VRM trees T_1 and T_2 are called *inter-isomorphic* if by a change of labeling in the intermediate vertices of one tree, it becomes equal to the other; otherwise, they are called *non-inter-isomorphic*. The set of all non-inter-isomorphic trees comprising of exactly two internal levels and n leaf nodes is denoted by $\mathcal{T}_2(n)$.

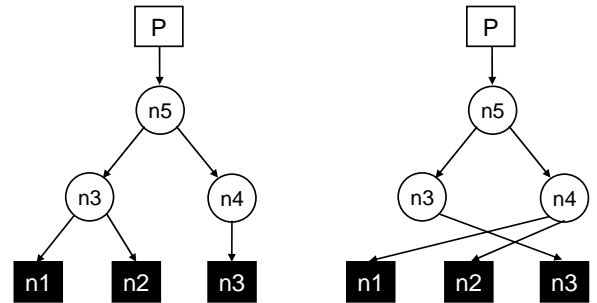


Fig. 7. Two inter-isomorphic trees.

It is clear that to find the optimal solution of VRM tree problem when the tree topology may be varied, only the set of non-inter-isomorphic feasible trees should be enumerated. In the remaining of this section we provide a mathematical framework to efficiently generate the set of non-inter-isomorphic trees.

Definition 4: A *partition* of set U is a collection of disjoint nonempty subsets of U such as U_1, \dots, U_k whose union is U . Each of these subsets is called a *part*.

The number of partitions of a set with n elements is the n 'th Bell number which can be computed from the following recurrence [19],

$$B_n = \sum_{k=0}^{n-1} \binom{n-1}{k} B_k; B_0 = 1. \quad (4)$$

Definition 5: For each n and $m \leq n$, the Stirling number of the second kind, denoted as $\left\{ \begin{smallmatrix} n \\ m \end{smallmatrix} \right\}$, is the number of ways of

partitioning a set of n elements into m nonempty sets. These numbers can be computed from the following recurrence [19]:

$$\begin{Bmatrix} n \\ m \end{Bmatrix} = \begin{Bmatrix} n-1 \\ m-1 \end{Bmatrix} + m \begin{Bmatrix} n-1 \\ m \end{Bmatrix}; \quad \begin{Bmatrix} n \\ 1 \end{Bmatrix} = \begin{Bmatrix} n \\ n \end{Bmatrix} = 1. \quad (5)$$

From the definition, one can see Bell numbers are related to Stirling numbers of the second kind by the following relation:

$$B_n = \sum_{k=0}^n \begin{Bmatrix} n \\ k \end{Bmatrix}. \quad (6)$$

Definition 6: A restricted growth function (RGF) of length n is a function $\phi : \{1, \dots, n\} \rightarrow \mathbb{N}$ which satisfy the following conditions [20]:

$$\begin{aligned} \phi(1) &= 1 \\ \phi(i) &\leq \max\{\phi(1), \dots, \phi(i-1)\} + 1 \quad (2 \leq i \leq n). \end{aligned} \quad (7)$$

Each RGF is represented as an n -tuple $[f(1), \dots, f(n)]$, where each entry in the tuple is a positive integer. The set of all RGF's of length n is denoted as Φ^n .

For example, both $[1, 1, 2, 1, 3]$ and $[1, 2, 3, 4, 2]$ are RGF's of length five, while $[1, 2, 4, 3, 1]$ and $[1, 1, 2, 2, 4]$ are not. Moreover, it can be easily verified that $\Phi^3 = \{[1, 1, 1], [1, 1, 2], [1, 2, 1], [1, 2, 2], [1, 2, 3]\}$.

Lemma 2: There is a one-to-one correspondence between Φ^n and the set of all partitions of $\{1, \dots, n\}$ [20].

In particular, each $\phi \in \Phi^n$ represents a partition of $\{1, \dots, n\}$ into $m \leq n$ groups, where 1 by convention belongs to the first part, i belongs to the $\phi(i)^{\text{th}}$ part, and $\max\{\phi(1), \dots, \phi(n)\} = m$. For example, the RGF $[1, 1, 2, 3, 2, 4, 1]$ represents the partition $\{\{1, 2, 7\}, \{3, 5\}, \{4\}, \{6\}\}$. For each RGF ϕ , $\nabla\phi$ is defined as $\nabla\phi = \max\{\phi(1), \dots, \phi(n)\}$. From Lemma 2 it is seen that $|\Phi^n| = B_n$. Moreover, one can see that the number of $\phi \in \Phi^n$ with $\nabla\phi = m$ is exactly $\begin{Bmatrix} n \\ m \end{Bmatrix}$.

Theorem 1: There is a one-to-one correspondence between the set of $\Lambda^n = \bigcup_{\phi \in \Phi^n} \{\phi\} \times \Phi^{\nabla\phi}$ and the set of all non-inter-isomorphic trees with exactly two internal levels and n leaf nodes, i.e., $\mathcal{T}_2(n)$.

Proof: We show that each $T \in \mathcal{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$ and vice versa. Without losing generality we assume leaf nodes of T are numerically labeled from left to right. The labeling starts from one and monotonically increments. By using the algorithm shown in Fig. 8, we assign new labels to the intermediate nodes. In this algorithm, $|L_l|$ is the number of nodes at level l . By using this algorithm, each $T \in \mathcal{T}_2(n)$ can be represented by the canonical form $([\phi_{3,1}, \dots, \phi_{3,|L_3|}], [\phi_{2,1}, \dots, \phi_{2,|L_2|}])$. Fig. 9 shows the labeling of a VRM tree after applying *VRM_tree_labeling* algorithm. It is seen that the canonical representation of this tree is $([1, 2, 2, 1, 3, 4], [1, 2, 3, 1])$.

By inspecting *VRM_tree_labeling* algorithm, it can be

easily verified that both $\phi_3 = [\phi_{3,1}, \dots, \phi_{3,|L_3|}]$ and $\phi_2 = [\phi_{2,1}, \dots, \phi_{2,|L_2|}]$ are RGF and $(\phi_3, \phi_2) \in \Lambda^n$. Moreover, if two trees are mapped to one $\lambda \in \Lambda^n$, according to Definition 3 they are inter-isomorphic and therefore they both cannot be in $\mathcal{T}_2(n)$. This means that each $T \in \mathcal{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$. Next, we show that each $\lambda \in \Lambda^n$ can be mapped to one and only one $T \in \mathcal{T}_2(n)$.

Assume that $\lambda = ([\phi_{3,1}, \dots, \phi_{3,n}], [\phi_{2,1}, \dots, \phi_{2, \nabla\phi_3}]) \in \Lambda^n$. The *build_VRM_tree* algorithm given in Fig. 10 shows how a $T \in \mathcal{T}_2(n)$ can be constructed from λ . The inputs of this algorithm are two RGF's ϕ_2 and ϕ_3 , where $(\phi_3, \phi_2) \in \Lambda^n$. It can be easily verified that if $\lambda_1, \lambda_2 \in \Lambda^n$ and $\lambda_1 \neq \lambda_2$, they cannot be mapped to a unique $T \in \mathcal{T}_2(n)$; therefore, each $\lambda \in \Lambda^n$ can be mapped to one and only one $T \in \mathcal{T}_2(n)$ and since it was shown that each $T \in \mathcal{T}_2(n)$ can be mapped to one and only one $\lambda \in \Lambda^n$, it is concluded that there is a one-to-one correspondence between Λ^n and $\mathcal{T}_2(n)$. ■

```

VRM_tree_labeling(T) {
  For l = 3 to 2 {
    largest_label = 0;
    For i = 1 to |L_l| {
      c_node = node of T with level l and label i;
      p_node = parent(c_node);
      If p_node does not have a label
        label(p_node) = ++largest_label;
        phi_{l,i} = label(p_node);
    }
  }
}
    
```

Fig. 8. VRM_tree_labeling algorithm.

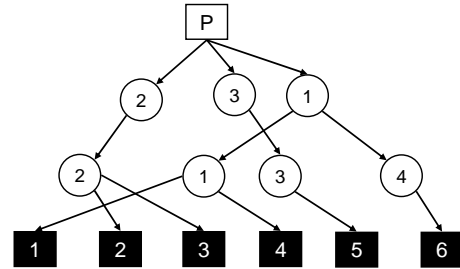


Fig. 9. An example of VRM tree labeling.

Lemma 3: The number of all non-inter-isomorphic trees with exactly two internal levels and n leaf nodes is obtained from,

$$|\mathcal{T}_2(n)| = \sum_{m=1}^n B_m \begin{Bmatrix} n \\ m \end{Bmatrix}. \quad (8)$$

Proof: According to Theorem 1, there is a one-to-one correspondence between the set of $\Lambda^n = \bigcup_{\phi \in \Phi^n} \{\phi\} \times \Phi^{\nabla\phi}$ and $\mathcal{T}_2(n)$; therefore, $|\mathcal{T}_2(n)| = |\Lambda^n|$. The number of elements in Λ^n is calculated as,

$$\begin{aligned}
 |\Lambda^n| &= \sum_{\phi \in \Phi^n} |\Phi^{\nabla\phi}| = \sum_{m=1}^n \left\{ \begin{matrix} n \\ m \end{matrix} \right\} |\Phi^m| \\
 &= \sum_{m=1}^n \left\{ \begin{matrix} n \\ m \end{matrix} \right\} B_m.
 \end{aligned} \tag{9}$$

The second equality is due to the fact that the number of $\phi \in \Phi^n$ with $\nabla\phi = m$ is $\left\{ \begin{matrix} n \\ m \end{matrix} \right\}$. Therefore, Equation (8) holds. ■

```

build_VRM_tree( $\phi_3, \phi_2$ ) {
    Construct a rooted trees such that:
         $|L_3|=n, |L_2|=\nabla\phi_3$ , and  $|L_1|=\nabla\phi_2$ ;
    Label nodes in each level  $l$  from 1 to  $|L_l|$ ;
    For  $l = 3$  to 2 {
        For  $i = 1$  to  $|L_l|$ 
            parent(node of  $T$  with level  $l$  and label  $i$ ) =  $\phi_{l,i}$ ;
        }
    For  $i = 1$  to  $|L_1|$ 
        parent(node of  $T$  with level  $l$  and label  $i$ ) = root;
    }
}
    
```

Fig. 10. Build_VRM_tree algorithm.

Table II shows the number of non-inter-isomorphic trees with 2 internal levels and n leaves. From the table data, it is seen that by using only non-inter-isomorphic trees, the number of enumerations required to find the optimal solution becomes more manageable.

TABLE II

NUMBER OF NON-INTER-ISOMORPHIC TREES WITH N LEAVES					
n	1	2	3	4	5
$ \mathcal{T}_2(n) $	1	3	12	60	358

An efficient algorithm for solving the RMTO-V problem is presented in Fig. 11. This algorithm uses Theorem 1 to enumerate the set of all non-inter-isomorphic trees and finds the topology of optimal VRM tree and corresponding regulator assignment. It should be noted that although according to Equation (8) the time complexity of this algorithm is exponential in the number of leaf nodes, because the number of different voltage domains is small, the runtime of the algorithm is quite reasonable.

```

RMTO-V*( $\mathcal{R}, \mathcal{F}, V_P$ ) {
    For each  $\phi_3 \in \Phi^{|\mathcal{F}|}$  {
        For each  $\phi_2 \in \Phi^{\nabla\phi_3}$  {
            build_VRM_tree( $\phi_3, \phi_2$ );
            RMTO-F( $\mathcal{R}, \mathcal{L}, V_P, T$ );
        }
    }
    Return best RMTO-F( $\mathcal{R}, \mathcal{F}, T, V_P$ );
}
    
```

Fig. 11. Efficient RMTO-V algorithm for VRM tree optimization.

C. Practical Issues

1) Noise Consideration

One practical issue in the proposed VRM tree topology is the propagation of the digital cores noise to the power supply of the analogue cores. The effect of this noise on system

operation can be reduced by isolating sensitive FB's from the noisy ones. Isolation can be performed through VRM isolation and/or distribution network isolation. In VRM isolation, sensitive blocks have their own VRM's which cannot be shared with noisy blocks. In this case, to find the power optimal VRM tree by using the RMTO-V algorithm, some of the tree topologies are not allowed in the enumeration. Distribution network isolation, on the other hand, is achieved by providing separate distribution networks for different blocks [2]. Assume FB1 and FB2 are driven by a single VRM and FB1 is a very sensitive circuit which should be isolated from the noisy circuit FB2. By providing separate distribution networks for FB1 and FB2 back to a common point X, most of the power supply noise generated by FB2 is dropped across the impedance of its private distribution impedance, Z_{FB2} and thus does not affect FB1 [2].

2) Effect of Current Profiles

Current profiles of the loads play a key role in the design of an efficient VRM tree. To motivate the need for considering the load profile of the FB's, consider the following example. Assume that to provide a FB with a desired voltage level, a buck converter is needed and the only candidate converters are those shown in Fig. 12. Now, if the load profile of the FB is $\{(200mA, 90\%), (100mA, 10\%)\}$, i.e., in 90% of the time the FB consume 200mA and in 10% it consumes 100mA current, then using the VRM (b) is more efficient whereas with a load profile of $\{(200mA, 10\%), (100mA, 90\%)\}$ VRM (a) is a better choice.

In the following, we describe how to account for the effect of load profiles in the RMTO-F algorithm. To begin with, for simplicity, we assume that the profiles of different FB's are independent of one another. In the next section, we show how to account for the correlations among load profiles.

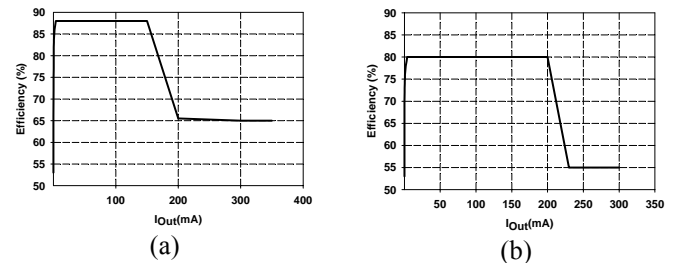


Fig. 12. The efficiency curves of two commercial buck VRM (TPS60502 [21] and TPS60503 [13]).

Assume that m FB's, f_1, f_2, \dots, f_m , with the same required voltage level V are connected to a node n . The current profiles of the FB's are expressed as $\{(I_i^j, \alpha_i^j)\}$ where I_i^j and α_i^j are the current demand and the probability of f_i being in its j^{th} state. Notice that for every i , $\sum_{j \in S(i)} \alpha_i^j = 1$, where $S(i)$ is the set of states of the load profile of f_i . When calculating the efficiency and input current of a candidate regulator c_n for n , $i_{out}(n)$ becomes a piecewise-linear function; so, instead of having a constant value for the efficiency and input current of node n , we need

to model both of them as piecewise-linear functions. That is,

$$\begin{aligned} \eta^{k_1, \dots, k_m}(c_n, v_{in}(n)) &= \mu_r(v_{in}(n), I_1^{k_1} + \dots + I_m^{k_m}) \\ i_{in}^{k_1, \dots, k_m}(c_n, v_{in}(n)) &= \frac{v_{out}(n) \times (I_1^{k_1} + \dots + I_m^{k_m})}{v_{in} \times \eta^{k_1, \dots, k_m}(c_n, v_{in}(n))} \\ \Pr(S(k_1, \dots, k_m)) &= \prod_{j=1}^m \alpha_j^{k_j}, \\ &\text{for } k_i \in S(i), 1 \leq i \leq m \end{aligned} \quad (10)$$

where $\eta^{k_1, k_2, \dots, k_m}$ and $i_{in}^{k_1, k_2, \dots, k_m}$ are the efficiency and input current when f_i is in state k_i and $\Pr(S(k_1, k_2, \dots, k_m))$ is the probability of such an event. Notice that the number of states in node n is the product of the number of states in its fanout nodes. An example of generating the piecewise linear input current for the fanin node is shown in Fig. 13. In this figure it is assumed that the VRM shown in Fig. 12(a) has been used and $V_{out}/V_{in} = 0.5$.

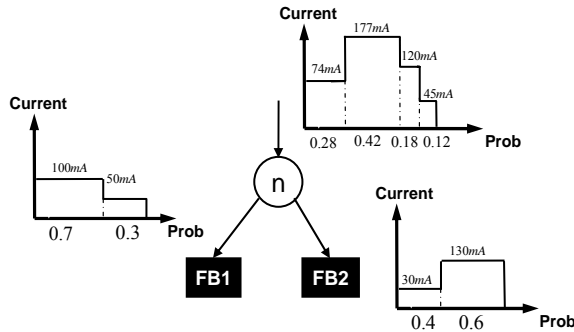


Fig. 13. Piecewise-linear modeling of the input current of a VRM

The average input current of node n , which is used in optimization, can be obtained from

$$i_{in}^{avg} = \sum_{k_i \in S(i), 1 \leq i \leq m} i_{in}^{k_1, \dots, k_m}(c_n, v_{in}(n)) \times \Pr(S(k_1, \dots, k_m)). \quad (11)$$

The candidate VRM c_n at node n should satisfy the constraint that,

$$\max_{k_i \in S(i), 1 \leq i \leq m} (I_1^{k_1} + \dots + I_m^{k_m}) \leq i_{c_n, out}^{max}. \quad (12)$$

3) Effect of Correlations among Current Profiles

The correlation between the load profiles of FB's could be used to design a more efficient VRM tree. To motivate the problem, consider two corner case examples. In the first case, the load currents of the FB's are *positively correlated* in the sense that both FB's have the same peak and off-peak load intervals. An example of such a case is two processor cores that work in parallel. In this case both processors achieve their minimum and maximum currents at the same intervals (c.f. Fig. 14(a)). On the other hand, in some cases, the load profiles of the FB's are *negatively correlated*, i.e., when one FB is in its low-load state, the other one is in the high-load state and vice versa (c.f. Fig. 14(b)). An instance of such a scenario occurs by using activity migration technique for dynamic thermal management in which the peak junction temperature is controlled by moving computation between multiple replicated units [22].

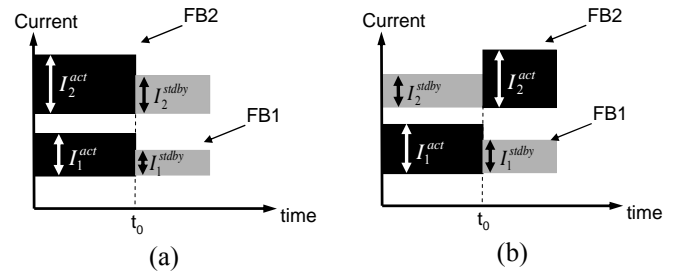


Fig. 14. (a) Positively correlated FB's (b) negatively correlated FB's.

It is clear that these two scenarios put different constraints on the VRM tree design. For example, when two FB's are negatively correlated, it is more likely that by sharing a single VRM for both of them, a more power-efficient VRM network can be achieved. Rather minor changes need to be made to the RMTO-F algorithm so that it can handle the effect of load profile correlations. In the remainder of this section, we describe how to account for the effect of load profiles in the RMTO-F algorithm. In our system modeling framework, it is assumed that the transition of the system into different current demand states can be described as a time-homogenous Markov chain with M states, and hence, current demand transitions can be captured by a stationary time-independent transition matrix $[p_{ij}]$. In each state of this Markov chain, the average current demand (I_{avg}) of all FB's is specified. Clearly, no two states will have the same I_{avg} assignments. Let π_i denote the probability of being in state i of this Markov chain. In vector $\pi = [\pi_i]$ entries π_i sum to one and satisfy

$$\pi_i = \sum_{j=1}^M \pi_j p_{ji} \quad (13)$$

Assume that m FB's, f_1, f_2, \dots, f_m , with the same required voltage level V are connected to a node n . The current profiles of the FB's are expressed as $\{I_i^j\}$ where I_i^j and is the current demand of f_i when it is in j^{th} state. When calculating the efficiency and input current of a candidate regulator c_n for n , $i_{out}(n)$ becomes a piecewise-linear function; so, instead of having a constant value for the efficiency and input current of node n , we need to model both of them as piecewise-linear functions. That is,

$$\begin{aligned} \eta^j(c_n, v_{in}(n)) &= \mu_r(v_{in}(n), \sum_{i=1}^m I_i^j) \\ i_{in}^j(c_n, v_{in}(n)) &= \frac{v_{out}(n) \times (\sum_{i=1}^m I_i^j)}{v_{in} \times \eta^j(c_n, v_{in}(n))} \end{aligned} \quad (14)$$

where η^j and i_{in}^j are the efficiency and input current when f_i 's are in state j .

The average input current of node n , which is used in optimization, can be obtained from

$$i_{in}^{avg} = \sum_{j=1}^M \pi_j \cdot i_{in}^j(c_n, v_{in}(n)). \quad (15)$$

Also notice that the candidate VRM c_n at node n should satisfy the constraint that,

$$\max_{1 \leq j \leq M} (\sum_{i=1}^m I_i^j) \leq i_{c_n, out}^{max}. \quad (16)$$

III. VRM TREE OPTIMIZATION FOR MINIMUM POWER DISSIPATION IN A SoC WITH DVS OPTION

Dynamic power management (DPM) is a feature of the run-time environment of a system that dynamically reconfigures itself to provide the requested services and performance levels with a minimum activity level on its FB's. The fundamental principle for the applicability of DPM is that systems (and their FB's) experience non-uniform workloads during operation time. Such an assumption is valid for most systems, both when considered in isolation and when inter-networked. A second assumption of DPM is that it is possible to predict, with a certain degree of confidence, the fluctuations of workload [23]. At the physical level, DPM is usually performed through assignment of appropriate voltage levels and corresponding clock frequencies to different FB's of the system. This is also known as dynamic voltage scaling (DVS). In a SoC with DVS option, an on-chip power manager decides when to switch the *SoC power-performance state (PPS)*, where each PPS corresponds to a particular combination of voltage level (and associated clock frequency) assignments to various FB's in the SoC.

The PDN of a DVS-enabled SoC is required to deliver power at appropriate voltage levels to different functional FB's while incurring the minimum power loss in the PDN. In the conventional technique to support DVS for different FB's, which is depicted in Fig. 3, each FB has its own VRM with multiple output voltage levels [15, 16]. The power manager selects the supply level that VRM_i provides to the FB_i.

This architecture, despite its simplicity, has several shortcomings: i) the number of VRM's used in the PDN is equal to the number of FB's i.e., when the number of FB's that can accept multiple voltage levels becomes large, the number of VRM's increases, which in turn increases the chip area and cost, ii) design of variable output voltage VRM is quite challenging and its cost is correspondingly higher than that of a fixed output voltage VRM, iii) unlike the VRM's with fixed- V_{out} where the power conversion efficiency is highly optimized for a specific output voltage level, the power conversion efficiency of the multiple- V_{out} VRM varies as a function of the chosen V_{out} and may sometimes degrade severely from one V_{out} to next [24].

Based on these observations, in the next section we propose a new technique to address the problem of PDN design to support dynamic voltage scaling.

A. Power Efficient PDN to enable DVS

In our technique, which is depicted in Fig. 15, the PDN is composed of two layers. In the first layer of PDN, which is called the *power conversion network (PCN)*, VRM's are used to generate all voltage levels that may be needed by different FB's in the SoC design. This is accomplished by using fixed- V_{out} VRM's; so, if \mathcal{U} is the set of all voltage levels required by any FB's, then there must be at least $|\mathcal{U}|$ VRM's in the PCN. Usually this number is small since many of the FB's share the same set of allowed voltage levels. In the second layer of PDN, a *power switch network (PSN)* is used to dynamically connect the power supply terminals of each FB to the appropriate VRM output in the PCN.

In our system modeling framework, it is assumed that the transition of the system into different PPS's can be described as a time-homogenous Markov chain, and hence, PPS transitions can be captured by a stationary time-independent transition matrix $[p_{ij}]$ (c.f., Fig. 16). In each state of this Markov chain, the supply voltage level of all FB's is specified. Clearly, no two states will have the same supply voltage assignments. Let π_i denote the probability of being in state i of this Markov chain. In vector $\pi = [\pi_i]$ entries π_i sum to one and satisfy

$$\pi_i = \sum_{j \in \mathcal{S}} \pi_j p_{ji} \quad (17)$$

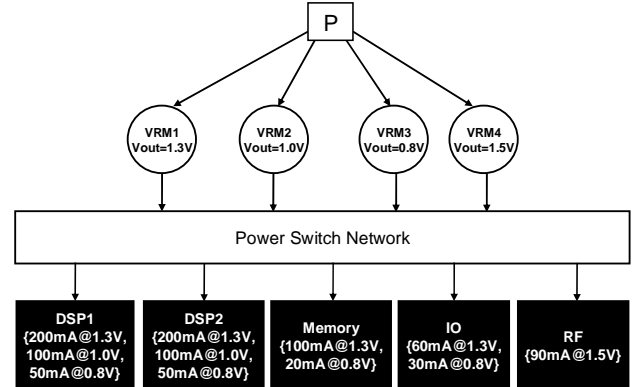


Fig. 15. The proposed architecture of PDN to support dynamic voltage scaling. The output voltage of each VRM is fixed.

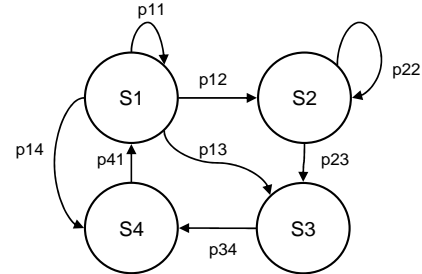


Fig. 16. Operating states and state transition of a system.

Additionally, for simplicity, in this section it is assumed that the current demands of every FB when it is working with each of its voltage levels is specified and is constant. In the next section it will be shown how to change the problem formulation to handle the general case when the current demands of FB's follow some probability distribution function around a mean value. Moreover, it is assumed that level shifters have been included in the SoC to enable communication among FB's operating on different supply voltages. Now, the question becomes how to design the PCN to achieve minimum power loss in the power distribution network, and how to design the PSN to make sure that all FB's receive the desired supply voltage levels.

1) Power Conversion Network Optimization

The PCN optimization supporting dynamic voltage scaling (PCODS) problem is defined next.

PCODS problem: Given is

- A library \mathcal{R} of VRM's and for each $r \in \mathcal{R}$, its cost c_r , output voltage $v_{r,out}$, the minimum and maximum input

voltages $v_{r,in}^{\min}$ and $v_{r,in}^{\max}$, the maximum load current $i_{r,out}^{\max}$, and the VRM's power conversion efficiency η_r as a function of the load current and input voltage,

- A power source P , with the nominal voltage of V_P ,
- A set \mathcal{F} of FB's, and for each $f \in \mathcal{F}$, the required voltages and the corresponding current demands,
- A Markov chain model \mathcal{S} of the system where the required supply voltage level of each FB is specified in each state of the Markov chain.

The objective is to build a network of VRM's that connects P to all FB's and minimizes a weighted sum of total power consumption and total cost of the VRM's used in the PCN, i.e., $V_P I_P + \lambda \sum_{r \in PCN} c_r$, while meeting the voltage and current constraints.

In PCODS problem, λ is a parameter which defines the tradeoff between power-efficiency and cost of the PCN. For example, if $\lambda = 0$, then PCODS will optimize the power efficiency while $\lambda = \infty$ will result in the least-cost PCN.

TABLE III
NOTATION USED IN THE RMTO ALGORITHM

\mathcal{S}	Set of all states of the Markov chain model of the system
π_i	Probability of being in state i of \mathcal{S}
p_{ij}	Transition probability from state i to state j of \mathcal{S}
\mathcal{V}_f	Set of required voltage levels by FB $f \in \mathcal{F}$
\mathcal{W}	Set of voltage levels required by all FB's; i.e., $\mathcal{W} = \bigcup_{f \in \mathcal{F}} \mathcal{V}_f = \{V_1, V_2, \dots, V_m\}$
$V_{f,s}$	Required voltage of FB $f \in \mathcal{F}$ in state $s \in \mathcal{S}$
$I_{f,s}$	Required current of FB $f \in \mathcal{F}$ in state $s \in \mathcal{S}$
$I_{f,v}$	Required current of FB $f \in \mathcal{F}$ when its voltage level is $v \in \mathcal{V}_f$ ($I_{f,v} = I_{f,s} : V_{f,s} = v$)
$I_{r,s}^{in}$	Input current of regulator r in state $s \in \mathcal{S}$
\mathcal{D}_i	FB voltage domain corresponding to $V_i \in \mathcal{W}$; i.e., $\mathcal{D}_i = \{f \in \mathcal{F} : V_i \in \mathcal{V}_f\}$
$I_{avg,r}$	Average input current of regulator r over all states

Before giving details of how PCODS can be solved, in Table III we define the notation used in the remainder of the chapter. The other notation used in this section is from Table I.

We assume that if a FB requires the same voltage V in two different states, it is always powered up by an identical VRM. This assumption implies that the number of power switches in PSN to deliver power to FB $f \in \mathcal{F}$ is exactly $|\mathcal{V}_f|$, and thus, it reduces not only the complexity of PSN, but also the power loss of the PSN during PPS transitions.

It should be noted that the power delivered to the FB's is independent of the topology of PCN and is calculated as,

$$P_{FBs} = \sum_{f \in \mathcal{F}} \sum_{s \in \mathcal{S}} \pi_s V_{f,s} I_{f,s}. \quad (18)$$

Since each FB may have more than one voltage level, FB voltage domains \mathcal{D}_i 's may be overlapping. For each voltage level $V_i \in \mathcal{W}$, one or more VRM's should be used to

deliver

power to the corresponding FB voltage domain \mathcal{D}_i . Assume that the topology of the VRM tree delivering power to \mathcal{D}_i is known. In this case, when the system is in state s , the output current of a VRM r with output voltage V_i that delivers power to a subset $\mathcal{D}_i^j \subseteq \mathcal{D}_i$ can be computed as,

$$I_{r,s}^{out} = \sum_{f \in \mathcal{D}_i^j, V_{f,s} = V_i} I_{f,s}. \quad (19)$$

Therefore, the input current of VRM r in state s is obtained as,

$$I_{r,s}^{in} = \frac{V_i \times I_{r,s}^{out}}{V_P \times \eta_r(V_P, I_{r,s}^{out})} \quad (20)$$

and the average input current of r which is drawn from the power supply is,

$$I_{avg,r} = \sum_{s \in \mathcal{S}} \pi_s I_{r,s}^{in}. \quad (21)$$

The average current drawn from the power supply by the FB voltage domain \mathcal{D}_i is then computed as,

$$I_{avg}(\mathcal{D}_i) = \sum_{r \in \mathcal{R}_i} I_{avg,r}^{in} \quad (22)$$

where \mathcal{R}_i is the set of all VRM's used to power up \mathcal{D}_i . The total cost of the VRM's used in this topology to deliver power to \mathcal{D}_i is,

$$C_{\mathcal{D}_i} = \sum_{r \in \mathcal{R}_i} c_r. \quad (23)$$

Therefore, the average current drawn from the power supply by this PCN and the total cost of VRM's in the PCN can be written as,

$$I_{avg} = \sum_i I_{avg}(\mathcal{D}_i) \quad (24)$$

$$C_{PCN} = \sum_i C_{\mathcal{D}_i}. \quad (25)$$

To deliver power to FB's in each \mathcal{D}_i , different options are available (c.f., Fig. 17 for a pictorial explanation). In the first option, which is the lowest-cost one, only one VRM is used to deliver power to all FB's in each \mathcal{D}_i . The other option is to use one VRM per FB. The drawback of this option is that the number of VRM's increases with the number of FB's. Because of the non-monotone dependency of power conversion efficiency on the delivered output current, neither solution may be that optimal from a power-efficiency viewpoint, i.e., a design in between the two extremes may be the best one. Furthermore, because objective function in the general formulation of the PCODS problem is a weighted sum of the power consumption and the cost of the PCN, by enumerating other solutions a better tradeoff between power-efficiency and cost may be achieved. Therefore, all possible "set partitioning" solutions of \mathcal{D}_i (as defined by Definition 4) should be enumerated when searching for the optimal VRM assignment to \mathcal{D}_i .

Definition 7: In a set partitioning of \mathcal{D}_i , the *required voltage* of each part is V_i whereas the *current demand* of a part in a given state is the summation of the current demands of all FB's in that part for the specified state.

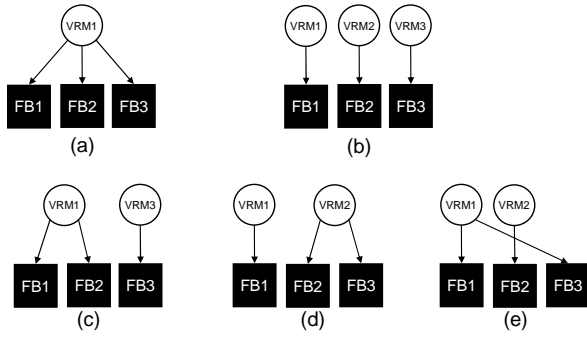


Fig. 17. Different options for delivering power to three FB's, each requiring the same voltage level in some state of its operation. The output voltages of all VRM's are the same.

Definition 8: A *valid VRM assignment* for a set partitioning of \mathcal{D}_i is the assignment of one VRM to each part such that the constraints of each VRM are satisfied, i.e., for each VRM r the input voltage of VRM is between $v_{r,in}^{\min}$ and $v_{r,in}^{\max}$, the required voltage of the part is $v_{r,out}$, and the maximum current demand of the part over all states is less than or equal to $I_{r,out}^{\max}$.

Definition 9: An *optimum VRM assignment* for a set partitioning of \mathcal{D}_i such as $\{\mathcal{D}_i^1, \dots, \mathcal{D}_i^n\}$ is a valid VRM assignment that minimizes $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$. Here, $I_{avg,j}$ and c_j denote the input current and associated cost of designated VRM to part \mathcal{D}_i^j , respectively.

Theorem 2: A valid VRM assignment for a set partitioning of \mathcal{D}_i is optimum if and only if $V_P I_{avg,j} + \lambda c_j$ is minimized in every part, \mathcal{D}_i^j .

Proof: Assume \mathcal{D}_i is partitioned into n nonempty subsets $\{\mathcal{D}_i^1, \dots, \mathcal{D}_i^n\}$. Each valid VRM assignments to a part is shown as a pair of input current and cost of the corresponding VRM, i.e., (I_{avg}, c) . The set of all valid VRM assignments to a part \mathcal{D}_i^j is shown as $\mathcal{Z}_j = \{(I_{avg}, c)\}$. The optimum VRM assignment to the set partitioning of \mathcal{D}_i is the selection of one tuple $(I_{avg,j}, c_j)$ from each \mathcal{Z}_j such that $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j = \sum_j (V_P I_{avg,j} + \lambda c_j)$ is minimized. It can be seen that this objective function is minimized exactly if the value of $V_P I_{avg,j} + \lambda c_j$ is minimized for each tuple $(I_{avg,j}, c_j)$ in \mathcal{Z}_j . ■

The conclusion of Theorem 2 is that in order to determine the optimum VRM assignment for a set \mathcal{D}_i , all set partitioning solutions for \mathcal{D}_i should be enumerated. For each partitioning solution, the VRM, r , which satisfies the constraints and minimizes $V_P I_{avg} + \lambda c$ for every part must be found, and subsequently, the partitioning solution that results in the minimum value of $\sum_j V_P I_{avg,j} + \lambda \sum_j c_j$ shall be identified as the optimum one.

```

optPCN( $\mathcal{R}, \mathcal{F}, \mathcal{S}, V_P$ ) {
  For each  $V_i \in \mathcal{U} = \{V_1, \dots, V_m\}$  {
     $\mathcal{D}_i = \{f \in \mathcal{F} : V_i \in \mathcal{U}_f\}$ ;
     $\psi(V_i) = \text{suboptPCN}(\mathcal{R}, \mathcal{F}, \mathcal{S}, V_P, V_i, \mathcal{D}_i)$ ;
  }
}
    
```

```

suboptPCN( $\mathcal{R}, \mathcal{F}, \mathcal{S}, V_P, \mathcal{D}_i$ ) {
  optCost =  $\infty$ ;
  optVRMs = {};
  For each non-empty partition of  $\mathcal{D}_i$  such as  $\{\mathcal{D}_i^1, \dots, \mathcal{D}_i^n\}$ 
  For each  $\mathcal{D}_i^j, 1 \leq j \leq n$  {
    Select best VRM  $r$  that minimizes  $V_P I_{avg,r} + \lambda c_r$ ;
     $cost_j = V_P I_{avg,r} + \lambda c_r$ ;
     $VRM_j = r$ ;
  }
  newCost =  $\sum_j cost_j$ ;
  If ( $newCost < optCost$ )
    optCost = newCost;
    optVRMs =  $\{VRM_j\}$ ;
}
  Return ( $optCost, optVRMs$ );
}
    
```

Fig. 18. The optPCN algorithm for solving PCODS.

Based on the above discussion, Fig. 18 shows optPCN algorithm to solve PCODS problem. Basically it starts by constructing \mathcal{D}_i sets and for each \mathcal{D}_i it finds the best VRM assignment by using Theorem 2.

Theorem 3: The optPCN algorithm finds the optimum solution to the PCODS problem.

Proof: The optimality of optPCN algorithm is immediate from Theorem 2 and the fact that for each \mathcal{D}_i , all set partitioning solutions are enumerated. ■

Theorem 4: The worst-case running time of optPCN algorithm is $O(|\mathcal{R}| |\mathcal{S}| |\mathcal{F}| B_{|\mathcal{F}|})$, where $|\mathcal{R}|$, $|\mathcal{S}|$, and $|\mathcal{F}|$ denote cardinalities of the corresponding sets and B is the corresponding Bell number.

Proof: For each voltage level required by FB's, $V_i \in |\mathcal{U}|$, we need to partition the corresponding FB voltage domain \mathcal{D}_i into k non-empty parts and for each part find the best VRM. The number of these parts is $B_{|\mathcal{D}_i|}$. For any such partitioning solution, we need to find the best set of k VRM's for its constituent parts. The best VRM for a part is found by enumerating all VRM's with output voltage V_i . The number of these VRM's is denoted by R_i . For each such VRM, we need to calculate the average input current of the VRM among all states of the Markov chain; the number of operations to find the best set of k VRM's for a partitioning solution is $|\mathcal{S}| |\mathcal{F}| R_i$. Therefore, the number of operations to find the best set of VRM's for V_i is $|\mathcal{S}| |\mathcal{F}| R_i B_{|\mathcal{D}_i|}$ and the complexity of the whole procedure is $\sum_i |\mathcal{S}| |\mathcal{F}| R_i B_{|\mathcal{D}_i|}$. For each i , $\mathcal{D}_i \subseteq \mathcal{F}$; therefore $|\mathcal{D}_i| \leq |\mathcal{F}|$, and it follows that,

$$\sum_i |S| |\mathcal{F}| R_i B_{|\mathcal{D}_i|} \leq |S| |\mathcal{F}| B_{|\mathcal{F}|} \sum_i R_i. \quad (26)$$

In equation (26) equality holds if and only if for each i , $\mathcal{D}_i = \mathcal{F}$. This condition requires that for each $f \in \mathcal{F}$, the set of required voltage levels be equal to \mathcal{U} , i.e., $\mathcal{V}_f = \mathcal{U}$. On the other hand, since each VRM in the library can provide only one output voltage, $\sum_i R_i \leq |\mathcal{R}|$. Therefore, the worst case running time of the algorithm is $O(|\mathcal{R}| |S| |\mathcal{F}| B_{|\mathcal{F}|})$. ■

From Theorem 4 one can see that optPCN algorithm has exponential complexity in the number of FB's; however, since the number of FB's is small, in practice the runtime of the algorithm is quite reasonable.

B. Effect of time-varying currents

In the formulation of PCODS problem, it is assumed that the current demand of each FB is a constant value independent of the system PPS. In this section it is shown how to modify the problem formulation to handle the case when the current demands of various FB's follow some probability density function (pdf).

We assume the current demands of different FB's can be modeled as independent Gaussian distribution functions (the case that the demands follow some other probability distribution function can be addressed in a similar manner). In this case, because the output current of a VRM which is connected to a number of FB's is a sum of independent Gaussian random variables (c.f., Equation (19)), it will also be a Gaussian random variable, whose mean and variance respectively are the sum of means and sum of variances of the current demand distributions in the corresponding FB's. This continuous-time random variable is approximated with a discrete-time random variable function which has the probability $\Pr(j)$ in interval $[I_{\min} + j \times \Delta I, I_{\min} + (j+1) \times \Delta I)$ (for $0 \leq j < (I_{\max} - I_{\min}) / \Delta I$) as shown in Fig. 19.

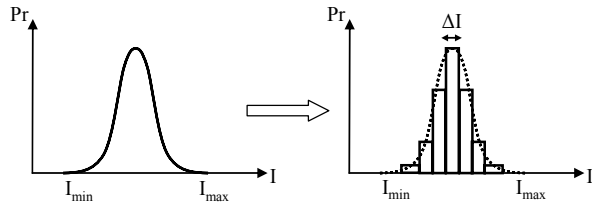


Fig. 19. Approximating the continuous distribution with a discrete one.

Since the efficiency of the VRM and hence its input current are functions of the output current, equation (20) should be modified to account for this dependency,

$$I_{r,s}^{in} = \sum_{j=0}^L \Pr(j) \frac{V_i \times (I_{\min} + j \times \Delta I)}{V_P \times \eta_r (V_P, I_{\min} + j \times \Delta I)} \quad (27)$$

where $L = (I_{\max} - I_{\min}) / \Delta I - 1$. Selecting a smaller value for ΔI results in a better approximation for input current of the VRM, but also increases the algorithm runtime.

C. Power Switch Network Optimization

Power switch network (PSN) performs the function of switching the supply voltage level of the FB's when a new PPS is commanded by the power manager. Fig. 20 depicts a PSN for delivering three different voltage levels to a FB. The switches in the PSN are controlled by a *power switch*

controller (PSC) which is zero-hot coded, i.e., at any given time only one of its outputs is zero, and hence, only one PMOS transistors in ON.

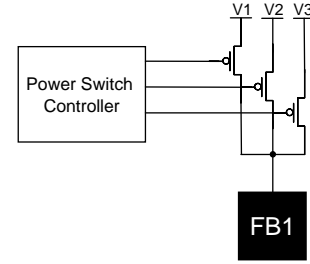


Fig. 20. A PSN for delivering three different voltage levels to a FB.

The number of PMOS transistors needed for each FB f in the PSN is $|\mathcal{V}_f|$. The PMOS transistor which is required to deliver voltage level $v \in \mathcal{V}_f$ to an $f \in \mathcal{F}$ and its width are respectively denoted as $M_{f,v}$ and $W_{f,v}$. This PMOS transistor should be large enough so that the voltage-drop between its drain and source does not exceed a tolerable value.

In the steady state, when FB f is supplied with $v \in \mathcal{V}_f$, the current that flows through the ON PMOS transistor $M_{f,v}$ is the current demand of f at voltage v , i.e., $I_{f,v}$. Since this transistor is in triode region, its current can be derived from the alpha-power model [25] as,

$$I_{f,v} = I_{M_{f,v}} = k \frac{W_{f,v}}{L_{eff}} \left(\frac{V_{gs} - V_t}{v - V_t} \right)^{\alpha/2} V_{ds} \quad (28)$$

where L_{eff} is the effective length of the transistor, V_{gs} , V_{ds} , and V_t are the gate-to-source, drain-to-source, and threshold voltage of the transistor, respectively. Note that k and α are technology. Now, if the maximum tolerable voltage-drop at the supply of the FB is ΔV , the minimum required width for $W_{f,v}$ will be computed as,

$$W_{f,v}^{\min} = \frac{I_{f,v} L_{eff}}{k \Delta V} \quad (29)$$

1) PSN Power Consumption

When the state of the system changes from PPS i to j , some energy is consumed to turn ON/OFF some of the power switches. Assume that the power manager changes the state of the system at regular time intervals with a frequency of f_{PM} . If C_{PMOS} is the total capacitance which is charged or discharged during this transition, then the power consumption for this transition is calculated from

$$P_{dyn,i \rightarrow j} = p_{i \rightarrow j} V_{dd}^2 f_{PM} C_{PMOS} \quad (30)$$

where $p_{i \rightarrow j}$ denotes the transition probability from PPS i to j which can be computed as,

$$p_{i \rightarrow j} = \pi_i p_{ij} \quad (31)$$

So, the power consumption of the PMOS switches is calculated as

$$P_{overhead} = \sum_{i,j} \left(\frac{1}{2} p_{i \rightarrow j} V_{dd}^2 f_{PM} \times \sum_{f: V_{f,i} \neq V_{f,j}} (C_{f,V_{f,i}} + C_{f,V_{f,j}}) \right) \quad (32)$$

where $C_{f,v}$ is the input capacitance of $M_{f,v}$, i.e., $C_{f,v} = W_{f,v}LC_{ox}$.

Equation (32) is the power consumption overhead of our solution compared to the conventional one, where one multiple-output VRM is used for each FB to provide it with appropriate voltage levels.

IV. EXPERIMENTAL RESULTS

The algorithms proposed in this paper have been implemented in C++ and evaluated on a set of test-benches. All experiments have been performed on a Linux server with 1.5GHz CPU and a 14-GB memory. A collection of thirty DC-DC commercially available regulators from Texas Instruments and National Semiconductors were chosen to create the library of VRM's. The power conversion efficiency of each VRM was modeled as a piecewise-linear function of input voltage and output current based on the data sheets for the VRM. The cost of each VRM was assumed to be its dollar cost for a 1000-unit purchase. Note that we did not have access to the efficiency curves and cost of the unpackaged DC-DC converters.

A. Static Voltage Islands

In the first set of experiments, we studied the efficiency of first proposed technique in reducing the power consumption of PDN in static voltage island SoC designs. More precisely, we compared the results of our RMTO-V with the results of the optimal VRM assignment in a star topology. Table IV shows the number of functional blocks in each test-bench along with the reduction of power loss in the PDN achieved by applying our algorithm (power loss in the PDN is the difference between the power delivered to FB's and the power drawn from the power source P). Also shown in this table is the increase in PDN cost of each test-bench as a result of applying our technique (PDN cost is the total cost of VRM's used in the network). The supply voltage V_p of each test-bench is 2.5V. From this table one can see that by applying RMTO-V, on average 18% power reduction in PDN can be achieved with a small PDN cost overhead.

TABLE IV
POWER REDUCTION AND COST INCREASE OF RMTO-V TECHNIQUE

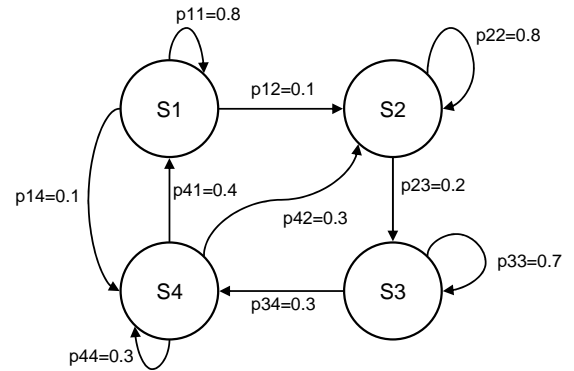
TB	$ \mathcal{F} $	PDN Power Reduction (%)	PDN Cost Increase (%)	Runtime (sec)
TB1	4	25.3	14.0	<1
TB2	5	21.6	11.1	<1
TB3	6	18.8	23.4	<1
TB4	6	19.8	8.9	<1
TB5	7	20.7	7.9	3
TB6	8	18.0	6.5	14
TB7	8	15.1	2.6	11
TB8	9	15.3	2.2	63
TB9	9	14.4	1.9	84
TB10	10	12.2	1.7	650
Average		18.1	8.0	-

B. Dynamic Voltage Islands

In the second set of experimental results, we performed two experiments to compare the performance of the proposed

technique with the conventional VRM assignment to support dynamic voltage scaling in a system. In the first experiment, we used optPCN algorithm with $\lambda = 0$ to find the most power-efficient PCN based on our solution. The best multiple-output VRM assignment to minimize the power consumption of the system based on the conventional solution was also generated for comparison purposes. The results of this experiment are reported in

Table V, where the first column gives the name of the test-bench (Details of the first test-bench of this table are provided in Fig. 21.), the second column gives the number of FB's in the problem, and the third column gives the number of states in the Markov chain model of the system. Column 4 and 5 show PDN power loss and cost reduction in the proposed solution compared to those of the conventional solution. Finally, the last column shows the runtime of optPCN algorithm for finding the optimal set of VRM in the PCN. From Table V, one can see that the proposed technique reduces the power loss of PDN by an average of 31%. Additionally, in most cases it also reduces the PDN cost. The average PDN cost reduction is 6.5%. Finally, one can see that the runtime of optPCN algorithm is quite reasonable.



S1: $\{V_{DSP1}=1.3, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5\}$
 S2: $\{V_{DSP1}=1.0, V_{DSP2}=1.3, V_{MEM}=1.3, V_{IO}=1.3, V_{RF}=1.5\}$
 S3: $\{V_{DSP1}=0.8, V_{DSP2}=1.0, V_{MEM}=1.3, V_{IO}=0.8, V_{RF}=1.5\}$
 S4: $\{V_{DSP1}=0.8, V_{DSP2}=0.8, V_{MEM}=0.8, V_{IO}=0.8, V_{RF}=1.5\}$

Fig. 21. Test-bench TB11. Current demands of FB's are similar to those in Fig. 3.

TABLE V
POWER AND COST REDUCTION OF PDN RESULTING FROM OPTPCN

TB	$ \mathcal{F} $	$ \mathcal{S} $	PDN Power Reduction (%)	PDN Cost Reduction (%)	Runtime (sec)
TB11	5	4	38.5	1.1	<1
TB12	6	4	40.4	5.0	<1
TB13	6	4	41.6	-6.7	<1
TB14	8	5	34.2	-2.8	<1
TB15	7	5	31.29	-0.5	<1
TB16	7	4	43.29	-8.4	<1
TB17	9	5	13.3	15.2	2
TB18	10	4	9.6	23.8	3
TB19	10	10	30.1	29.7	13
TB20	12	10	27.9	8.1	70
Average			31.0	6.5	-

In the second experiment, we studied the tradeoff between

the power-efficiency of the PDN and its cost. More precisely, in addition to designing the optimal PCN for $\lambda = 0$ by running optPCN algorithm, the algorithm was invoked for other values of λ for which the PCN power loss does not increase beyond 10% of its optimal value. The cost reduction of the PDN for this set of test-benches is reported in Table VI. It is seen that on average by allowing about 9% increase in the PDN power loss, the cost of PDN can be lowered by 41%.

TABLE VI
TRADING OFF POWER FOR COST OF PDN IN THE PROPOSED TECHNIQUE

<i>TB</i>	PDN Power Increase (%)	PDN Cost Reduction (%)
TB11	10.0	53.0
TB12	4.3	46.9
TB13	8.9	32.34
TB14	8.9	57.9
TB15	9.8	32.1
TB16	9.7	31.2
TB17	9.6	39.3
TB18	10.0	45.4
TB19	9.6	26.1
TB20	10.0	52.9
Average	9.1	41.7

V. CONCLUSION

We presented two new techniques for optimal design of power delivery network for multiple voltage island system-on-chips. First we showed that by using a tree topology of suitably chosen voltage regulators between the power source and loads, one can achieve higher power efficiency in the power delivery network of static voltage island designs. We formulated the problem of optimizing the VRM tree as a dynamic program and solved it efficiently. Second we presented a technique to design an efficient power delivery network for systems with dynamic voltage scaling capability. In this technique, the PDN is composed of two layers: PCN and PSN. In PCN, fixed- V_{out} VRM's are used to generate all voltage levels that may be needed by different FB's in the system. PSN is subsequently used to dynamically connect the power supply terminals of each FB to the appropriate VRM output in the PCN. We showed that this technique not only reduces the cost of the power conversion network, but also results in a more power-efficient power delivery network. We further described an algorithm to select the best VRM's to achieve a design target in the new PDN. Experimental results have demonstrated the efficacy of proposed problem formulations and solutions.

REFERENCES

- [1] S. Chun, "Methodologies for modeling simultaneous switching noise in multi-layered packages and boards," Ph.D. dissertation, Georgia Institute of Technology, 2002.
- [2] W. Dally and J. Poulton, *Digital Systems Engineering*. New York, NY: Cambridge University Press, 1998.
- [3] D. E. Lackey, P. S. Zuchowski, *et al.*, "Managing power and performance for System-on-Chip designs using voltage islands," in *Proc. of International Conference on Computer Aided Design*, 2002, pp. 195-202.
- [4] L. Smith, R. Anderson, *et al.*, "Power distribution system design methodology and capacitor selection for modern CMOS technology," *IEEE Trans. on Advanced Packaging*, vol. 22, no. 3, Aug. 1999, pp. 284-291.
- [5] S. Park, J. Kim, *et al.*, "Multilayer Power Delivery Network Design for High-speed Microprocessor System," in *Proc. of Electronic Components and Technology Conference*, 2003, pp. 1613-1618.
- [6] Semiconductor Industry Association, International Technology Roadmap for Semiconductors, 2003 edition, [online] <http://public.itrs.net/>.
- [7] W. Borland, "Decoupling of high performance semiconductors using embedded capacitor technology," in *Proc. of International Symposium on the Applications of Ferroelectrics*, 2006, pp. 1-4.
- [8] P. Hazucha, T. Karnik, *et al.*, "Area-efficient linear regulator with ultra-fast load regulation," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 4, Apr. 2005, pp. 933-940.
- [9] S. Zhao, K. Roy, and C. Koh, "Decoupling capacitance allocation for power supply noise suppression," in *Proc. of International Symposium on Physical Design*, 2001, pp. 66-71.
- [10] H. Su, S. S. Sapatnekar, and S. R. Nassif, "An algorithm for optimal decoupling capacitor sizing and placement for standard cell layouts," in *Proc. of International Symposium on Physical Design*, 2002, pp. 68-73.
- [11] K. Wang and M. Marek-Sadowska, "On-chip power-supply network optimization using multigrid-based technique," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 3, Mar. 2005, pp. 407-417.
- [12] M. Zhao, R. Panda, *et al.*, "A fast on-chip decoupling capacitance budgeting algorithm using macromodeling and linear programming," in *Proc. of Design Automation Conference*, 2006, pp. 217-222.
- [13] Texas Instruments, "TPS60503 Datasheet," [online] <http://www.ti.com/lit/gpn/tps60503>
- [14] G. Patounakis, Y. W. Li, and K. L. Shepard, "A fully integrated on-chip DC-DC conversion and power management system," *IEEE Journal of Solid-State Circuits*, vol. 39, no. 3, Mar. 2004, pp. 443-451.
- [15] T. D. Burd and R. W. Brodersen, "Design issues for dynamic voltage scaling," in *Proc. of International Symposium on Low Power Electronics and Design*, 2000, pp. 9-14.
- [16] K. J. Nowka, G. D. Carpenter, *et al.*, "A 32-bit PowerPC system-on-a-chip with support for dynamic voltage scaling and dynamic frequency scaling," *IEEE Journal of Solid-State Circuits*, vol. 37, no. 11, Nov. 2002, pp. 1441-1447.
- [17] B. Amelifard and M. Pedram, "Optimal selection of voltage regulator modules in a power delivery network," in *Proc. of Design Automation Conference*, 2007, pp. 168-173.
- [18] B. Amelifard and M. Pedram, "Design of an efficient power delivery network in an SoC to enable dynamic power management," in *Proc. of International Symposium on Low Power Electronics and Design*, 2007, pp. 328-333.
- [19] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics: A Foundation for Computer Science*. Reading, MA: Addison-Wesley, 1990.
- [20] D. L. Kreher and D. R. Stinson, *Combinatorial Algorithms: Generation, Enumeration, and Search*. Boca Raton, FL: CRC Press, 1999.
- [21] Texas Instruments, "TPS60502 Datasheet," [online] <http://www.ti.com/lit/gpn/tps60502>
- [22] S. Heo, K. Barr, and K. Asanovic, "Reducing power density through activity migration," in *Proc. of International Symposium on Low Power Electronics and Design* 2003, pp. 217-222.
- [23] A. Iranli and M. Pedram, "System-Level Power Management: An Overview," in *The VLSI Handbook*, W.-K. Chen, Ed. New York, NY: CRC Press, 2006.
- [24] A. Stratakos, "High-efficiency low-voltage DC-DC conversion for portable applications," Ph.D. dissertation, University of California, Berkeley, 1998.
- [25] T. Sakurai and A. R. Newton, "A simple MOSFET model for circuit analysis," *IEEE Trans. on Electron Devices*, vol. 38, no. 4, Apr. 1991, pp. 887-894.