

# Noise Minimization During Power-Up Stage for a Multi-Domain Power Network

Wanping Zhang<sup>1,2</sup>, Yi Zhu<sup>2</sup>, Wenjian Yu<sup>3</sup>, Amirali Shayan<sup>2</sup>, Renshen Wang<sup>2</sup>, Zhi Zhu<sup>1</sup>, Chung-Kuan Cheng<sup>2</sup>

<sup>1</sup>{wanpingz, zzhu}@qualcomm.com, Qualcomm Inc. 5775 Morehouse Dr., San Diego, CA, U.S.A

<sup>2</sup>{w7zhang, y2zhu, amirali, rewang, ckcheng}@ucsd.edu, UC San Diego, La Jolla, CA, U.S.A

<sup>3</sup>yu-wj@tsinghua.edu.cn, Tsinghua University, Beijing 100084, China

**Abstract** – With the popularity of Multiple Power Domain (MPD) design, the multi-domain power network noise analysis and minimization is becoming important. This paper describes an efficient heuristic algorithm to arrange the power-up sequence in a multi-domain power network in order to minimize the noise. We present a formulation of this problem and show it is NP-complete. Therefore, we propose a simulated annealing (SA) based algorithm with preprocessing. Experimental results show that the proposed algorithm can minimize the noise close to the minimal values. In terms of efficiency, the SA algorithm is more than hundreds of times faster than the enumerating method and the running time scales well for these cases with the number of domains. In addition, we discuss the trade off between power-up efficiency and noise.

## I Introduction

Power supply is becoming a major concern in VLSI design with technology scaling. The voltage violation in the power supply network has adverse impact on the performance and reliability. Power network noise not only lead to longer gate delay [1], but also cause logic failure with excessive voltage variation [2].

The power network noise is often characterized by the voltage violation area [3-5], which describes the accumulating effect of noise. The violation area at node  $j$  is defined as:

$$A_j = \int_0^T \max(V_{\min} - v_j(t), 0) dt \quad (1)$$

where  $V_{\min}$  is the lowest voltage level allowed for a power line. Fig. 1 shows the illustration of the violation area (the shaded area between  $t_1$  and  $t_2$ ).

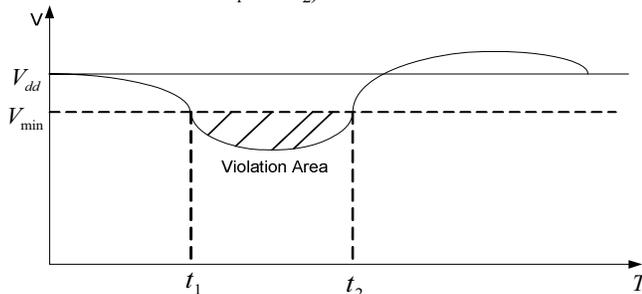


Fig.1. Illustration of violation area.

Multiple power domain (MPD) is becoming popular in the modern SoC design. In order to handle different performance objectives and constraints among different blocks, a new approach is to partition the internal logic of the chip into multiple power domains [6]. The power-up sequencing is one of the major challenges in MPD design for noise reduction. It is not practical to bring up all the power supplies at the same time, because excessive noise will be

introduced due to the rush current. Hence, it is beneficial to design a power-up sequence to enable different power domains in a well-defined order, which results in less noise and therefore assures correct function [6].

Lots of previous work discussed the importance of the power-up sequence in the initialization stage in order to minimize noise. Salmon and Dour [7] showed that the voltage level shifting circuitry associated with the core logic is able to initialize properly only when the core logic voltage supply lines are ramped prior to the I/O voltage supply lines. Ranjan [8] designed a circuit that can turn each transistor stage on and off in order, so as to avoid drawing huge current which leads to excessive voltage violation. A power switch design was developed to minimize rush current [9], and a sequential power-up scheme was established [10]. The above techniques consider the power-up sequence in transistor or logic gate level. We will extend this sequencing problem into multi-domain power network.

In current MPD designs, we need to take into account of the sequence of different power domains for minimizing the overall power noise. Also, we make sure all power domains are completely powered up before proceeding to other tasks. For example, the CPU may wait until the rest of the chip is powered up before booting [6]. As a result, all domains need to start powering up before a particular time point, which is referred as “deadline” in this paper.

The main contributions of this work are:

- (1) A power-up sequencing problem at power-domain level is formulated, where the voltage violation area of power network needs to be minimized.
- (2) The problem considering inter-domain timing relationship is proved to be NP-complete.
- (3) An efficient method is proposed to find the power-up sequence with the minimum power network noise. This method is based on the simulated annealing (SA) algorithm with a domain ordering technique.
- (4) The relationship between the total time to power up all domains and the overall noise is analyzed, which helps designers make the trade-off decision.

In the next section, the multiple power domain design and power noise analysis are introduced. Section III presents the formulation and a brief NP-completeness proof. The proposed algorithm is given in Section IV. The last two sections include the numerical results and conclusions, respectively.

## II. Background

Multiple power domain (MPD) design partitions the chip into several blocks based on their functionalities and characteristics. A power domain is a logic entity as well as a

collection of design elements that share a primary power supply [6]. According to [10], the benefits to introduce MPD design can be summarized in three aspects as follows. Firstly, separated system development can be performed in each domain. Secondly, as different domains work independently, we are able to apply various power gating schemes based on the functionality of a particular block, in order to reduce leakage power consumption. Thirdly, clock frequency for each domain could also be changed for the sake of dynamic power reduction.

One of the most important issues to power up all domains is the stability of VDD/GND lines. Turning on the power switches may cause a large rush current on the power lines. Those large rush currents will make the inductance components more significant and therefore more switching noise will be introduced. Previous research has shown that poor rush current management or power supply noise can potentially corrupt retention registers, which may lead to unsafe state [6].

#### A. Power Domain Power-Up Sequence

Turning on transistors in sequence can avoid drawing large amounts of current. The switches are grouped into several sets, and were turned on with delay in between [6]. Similarly, the power-up sequence for all domains is also critical for limiting the rush current, so that there may not cause voltage spikes that could corrupt registers.

Fig. 2 shows the rush current during the power-up stage of one domain. The rush current leads to large voltage violation spikes, because of large IR drop and switching noise. If all switches are turned on at the same time, the overall noise is unaffordable. Consequently, switches need to be turned on in sequence to avoid the excessive voltage drop. As a result, we can see there are several voltage spikes, but with smaller peak. When turning on all domains, we need to design a sequence to power up every domain in order to minimize total noise.

There are some timing relationships between domains due to the signal or data transition. For example, if domain  $A$

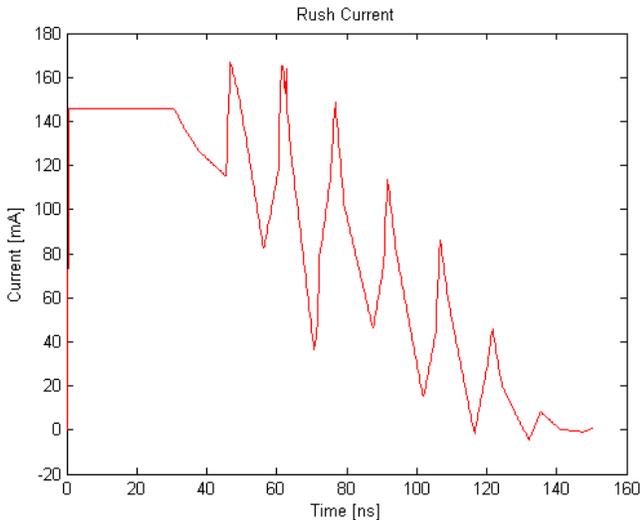


Fig. 2. Rush current during the power-up stage.

wants to get data in the tenth cycle after powering up from domain  $B$ ,  $B$  needs to be turned on in time so that its data will be available when  $A$  acquires it. More applicably, there was a real industry hierarchical power distribution design with a power tree [9]. Those domains on lower levels of the hierarchy can be in the powered-on state only if the domains on higher levels are on. Therefore, when designing the power sequence for all domains, we need to consider the inter-domain timing relationships as constraints.

#### B. Power Noise Analysis

To analyze the total noise when powering up all domains, the idea of superposition is utilized. We assume that the power network is a linear time invariant system, the voltage drop at one node is the superposition of those voltage drops caused by all domains individually. In this sense, we divide the analysis work into two steps. Firstly, we simulate the voltage response at the observation node with each domain working respectively, and then obtain the voltage drops. Secondly, we analyze the voltage noise with the superposition of all the voltage drops.

### III. Problem Statement

#### A. Problem Formulation

Fig. 3 illustrates the power-up sequence for multiple domains. Each row corresponds to the power status for a domain, and there are  $D$  domains. For each domain, one square represents the power status in one clock cycle. The blank square denotes power off, while the dark square denotes power on.  $X_i$  ( $1 \leq i \leq D$ ) is the cycle when the  $i$ th domain switches to power on. Therefore, the voltage response contributed by the  $i$ th domain keeps zero during the previous  $X_i - 1$  cycles. The nonzero voltage waveform can be generated by shifting the response powered up at the  $X_i$  cycle. Based on the superposition idea, the overall voltage drop would be the summation of the drops contributed by all the domains.

The noise minimization problem during power-up stage can be formulated to be an optimization work as shown in Fig. 5. This problem is to find a power-up time sequence for multiple domains, denoted by  $X_1, X_2, \dots, X_D$ , to minimize the voltage violation area for a given observation node at power network. The related parameters are listed in Fig. 4. We sample the voltage waveform for each domain with  $P$  time

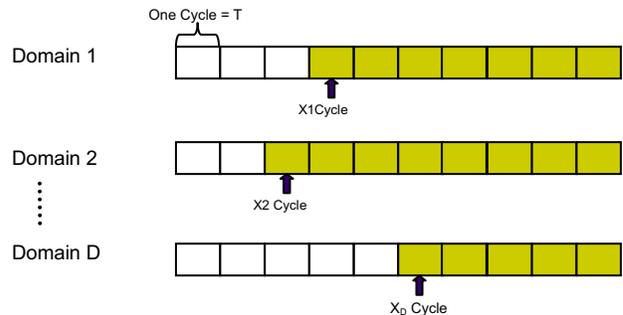


Fig.3. Illustration of power-up sequence for multiple domains.

|                   |   |
|-------------------|---|
| $D$ :             | the number of domains;  |
| $T$ :             | the time period of clock;   |
| $P$ :             | the number of time samples to describe the voltage response contributed by one domain;  |
| $d$ :             | the interval between adjacent time sample points;                                       |
| $X_i$ :           | the starting cycle when domain $i$ becomes power on;                                    |
| $L_i$ :           | the last power-up cycle (deadline) for domain $i$ ;                                     |
| $V_{dd}$ :        | nominal high-level voltage;   |
| $V_{min}$ :       | minimal voltage requirement; Voltage is considered to be violation if below this value; |
| $Cutoff$ :        | the allowed maximum voltage drop, i.e. $V_{dd} - V_{min}$ .                             |
| $V_{sup}^i$ :     | superimposed voltage drop for the $i$ th sampling point.                                |
| $V_{violate}^i$ : | violated voltage amount for the $i$ th sampling point.                                  |

Fig. 4. Parameter description for power-up sequencing problem.

|  |                         |
|--|-------------------------|
| Power-up sequencing problem statement  |                         |
| Objective function:  |                         |
| $\min \sum_{i=1}^P V_{violate}^i \cdot d$  |                         |
| where $V_{violate}^i = \begin{cases} V_{sup}^i - cutoff, & \text{if } V_{sup}^i - cutoff > 0 \\ 0, & \text{otherwise} \end{cases}$ |                         |
| Constraints:   |                         |
| (1) Inter-domain timing relationships,   | i.e.,                   |
| $X_j + a_{jk} \leq X_k \leq X_j + b_{jk};$   |                         |
| (2) Deadline to start powering on, i.e.,   | $0 \leq X_i \leq L_i$ . |
| Decision Variables:  |                         |
| $X_1, X_2, \dots, X_D.$  |                         |

Fig.5. The power-up sequencing problem.

points, whose intervals are  $d_i$  nano-seconds ( $1 \leq i \leq P$ ). Based on the violation area definition in Fig. 1, the violated amount for the  $i$ th sampling point is the amount exceeding a tolerable cutoff. Then the violation area can be approximated by the multiplication of violated voltage and sampling interval. The inter-domain timing relationship will become the constraints as shown in Fig. 5. Constraint (1) means domain  $k$  can start powering up no earlier than  $a_{jk}$  cycles and no later than  $b_{jk}$  cycles after domain  $j$ . The deadlines for powering up also impose additional constraints.

### B. NP Complete Proof

This problem is NP-complete, and the proof is given as follows.

*Lemma:* The Power-up sequencing problem (Fig. 5) is NP-complete.

*Proof:* The power-up sequencing problem is in NP, because verifying whether the violation area corresponding to a given sequence is less than a particular value or not can be done in polynomial time. To prove it is NP-hard, we reduce a known NP-complete problem, i. e. the partition problem [11], to the power-up sequencing problem.

The partition problem is to decide whether a given set of  $m$  integers  $A_1, \dots, A_m$  with the total sum  $S$  can be partitioned into two subsets that have the same sum  $S/2$ . We would like

to reduce it to the decision version of the power on sequence problem, i.e. can we find a sequence such that the total violation area is less than or equal to a constant  $K$ .

From an instance of the partition problem, we construct an instance of the power-up sequence with  $m$  domains. Each domain has two cycles and one sampling point per cycle. The voltage drop at this two sampling points for domain  $i$  are  $V_{drop}^i[1]=A_i$  and  $V_{drop}^i[2]=0$ . Let us suppose  $Cutoff = S/2$ . Because there are only two cycles with one sampling point per cycle, each domain can shift 0 or 1 cycle, which means  $X_i = 0/1$ . We can show that the partition problem has a solution if and only if the power-up sequence has a solution with violation area 0.

When a partition problem has a solution of two subsets with equal sum  $S/2$ , we can start the domains which correspond to the first subset in the first cycle, and start those domains which correspond to the second subset in the second cycle. Thus we have  $V_{sup}^1 = S/2$ , and  $V_{sup}^2 = S/2$ . As the  $Cutoff = S/2$ , the violation area is 0. Conversely, when there is a solution to the power-up sequencing problem with violation area 0, we can partition the set based on the cycle where each domain is started. Hence we have proved the power-up sequencing problem is NP-complete.

## IV. The SA Based Method Finding the Optimal Power-Up Sequence

The proposed method consists of three parts: domain ordering, a greedy algorithm to get an initial solution, and the simulated annealing (SA) based searching. We first order the position of domains in the solution to speed up the searching. Then, a greedy initial solution is obtained. The SA based searching algorithm solves the optimal powering up sequence for minimizing the total violation area.

### A. Domain Ordering

We order the domains as a sequence to generate the feasible solution. Because in one solution  $X_1, X_2, \dots, X_D$ , the domain which appears earlier will restrict the possible range for these domains that appear later based on the constraints. Therefore, if we have determined the starting cycles of those domains which have more constraint relationships with others, the following domains will have less search space. As a result, the searching efficiency will be greatly improved.

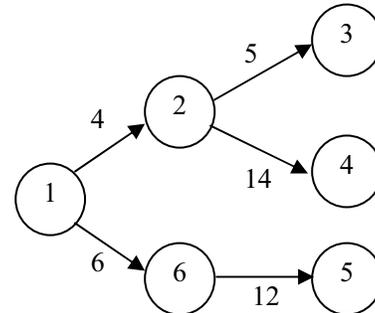


Fig.6. A constraint graph modeling the inter-domain relationships.

The inter-domain timing constraints are modeled as a directed graph, shown in Fig. 6. Every node represents one domain. A directed edge  $(A, B)$  points from one domain to another, which means domain  $B$  depends on domain  $A$ . The number on each edge is the range of cycles that domain  $B$  can choose based on the constraints given by domain  $A$ . For example, if one of the inter-domain constraints are: Domain 2 needs to power up after 10 cycles but before 14 cycles of domain 1, which means the freedom of domain 2 based on domain 1 is 4. Following the same rule, we construct the directed inter-domain relationship graph as show in Fig. 6.

The ordering algorithm is described in Fig. 8, and Fig. 7 shows the related parameters, where sequence  $P$  is the output of the algorithm. The main idea is: we want to firstly select the domain which can constrain more other domains and let these domains as less freedom as possible. The evaluation expression in (2) means the average freedom of the domains controlled by domain  $i$ . For a particular domain  $i$ , its value becomes smaller if domain  $i$  gives less freedom per domain that it controls. As the example in Fig. 6, the first domain to choose is domain 1, because it is the only one with zero input degree. After we delete domain 1 and its output edge, the next domain to choose is domain 2, because its evaluation value is  $(5+14)/2=9.5$ , while the evaluation value for domain 6 is 12. If we continue this algorithm, the ordering result for the six domain power network in Fig. 6 is: 1, 2, 6, 3, 4, 5.

$S$ : the set of domains that have not been ordered;  
 $P$ : the ordered sequence of domains;  
 $Freedom_{ij}$ : the selection range of domain  $j$  based on the constraint from domain  $i$ ;  
 $OutDegree_i$ : the output degree for domain  $i$ , which is the number of domains constrained by domain  $i$ .

Fig. 7. Parameter description for the ordering algorithm.

### B. Greedy Initial Solution

We propose a greedy algorithm to find the initial solution for the SA based algorithm. As shown in Fig. 9, the algorithm scans the ordered domains one by one. For each domain, it makes the local optimal decision which causes the minimal superimposed violation area. Finally, it outputs the

Ordering Algorithm: Given the constraint graph  
 $P=\emptyset$ ;  $S$ =all the domains;  
While  $S \neq \emptyset$  do  
  If (there is no domain in  $S$  which has output degree)  
    Add  $S$  to the end of  $P$ ;  
    break;  
  EndIf;  
  Among the domains in  $S$  without input degree and with output degree, choose the domain  $i$  with  

$$\min \frac{\sum_j Freedom_{ij}}{OutDegree_i} ; \quad (2)$$
  
  Delete domain  $i$  and its out edges from  $S$ ;  
  Add domain  $i$  to the end of  $P$ ;  
EndWhile.

Fig.8. The domain ordering algorithm.

### Greedy Initial Solution Algorithm:

For  $i = 1$  to  $D$ ,  
  According to the values of  $X_j$ ,  $j < i$ , and the freedoms associated with the constraints on domain  $i$ , determine the possible values for  $X_i$ ;  
  Choose the value of  $X_i$  such that the superimposed violation area caused by domains from 1 to  $i$  is minimal;  
EndFor.

Fig. 9. A greedy algorithm to find the initial solution.

initial values of  $X_1, X_2, \dots, X_D$ .

### C. Simulated Annealing Based Algorithm

The simulated annealing based algorithm is presented in Fig. 10. The cost function is the total violation area for a given power-up sequence. The voltage waveforms at observation node caused by each domain have been simulated in advance, and then the voltage drops by each domain will be obtained. So, the voltage drops are shifted with  $X_i$  time cycles and then superimposed to easily produce the actual voltage drops. Then, the violation area below  $V_{min}$ , i.e. the cost function, is computed.

### Simulated Annealing Based Algorithm:

Ordering();  
Seq = GreedyInitialSolution();  
Temp = Initial\_Temperature;  
Iteration = 0;  
Repeat  
  Neighbor(Seq, Seq');  
  Cost = ViolationArea(Seq');  
  dif = Cost - ViolationArea(Seq);  
  if( Cost < minCost)  
    minCost = Cost;  
    minSeq = Seq';  
  end if  
  r = Random(0, 1);  
  if (r < exp(-dif / Temp) )  
    Seq = Seq';  
    Temp = Temp \* Temperature\_Adjustment;  
  end if  
  Iteration ++;  
Until Temp == Freezing\_Point or Iteration > maxIter;  
End.

Fig.10. Simulated annealing based algorithm.

The neighbor search plays a crucial role in a SA based algorithm. Given the current power-up sequence, we need to perturb it to produce a new sequence. We randomly choose a domain, and determine its possible range based on the constraints and current starting cycle of the other domains. The new starting cycle for this domain will be chosen within the possible range. This neighbor search method guarantees that all the constraints are met without further checking.

The conventional cooling schedule and stopping criterion are adopted in our algorithm. With higher temperature, the algorithm has high probability to accept the current solution even though it is not better than the current best solution. Therefore, the algorithm searches within a larger space. However, when temperature becomes lower, the algorithm will have high probability to accept the solution that is better

than the current best. The related parameters are determined experimentally.

## V. Experimental Results

We have implemented the SA based algorithm and an enumerating method for comparison in C language. The experiment environment is a PC with 3.2GHz Pentium 4 processor. Firstly, we give a case to show how the whole SA based method works, and analyze the relationship between the power-up deadline and the minimum violation area. Then, ten test cases with different domain numbers are discussed, for which the computational results from different methods are compared.

### A. A Case with Eight Power Domains

We consider a power network for a 7mm×7mm chip with eight domains. The power network is modeled with a RLC netlist. The  $V_{dd}$  is 1.2V. For a given observation node, we simulate its voltage responses with only one domain working. The voltage waveforms obtained with HSPICE are shown in Fig. 11. The clock cycle is 5ns, and the simulation spans 100 cycles. From the figure, we can see all domains power up during the first 30 cycles, while the rush current leads to very sharp voltage drop. After each domain is fully charged and works normally, the voltage drop becomes much smaller.

For this case, we assume the maximal allowed voltage drop is 0.1V, and there are several inter-domain timing relationships that need to be considered as constraints. With the simulated responses, the proposed method can be used to search the power-up sequence for the domains. If the power-up deadlines are all 50 cycles, the obtained minimal violation area with the proposed SA based algorithm is 1143.6 mV·ns. The corresponding power-up sequence is: 0, 14, 1, 22, 3, 32, 46, 50, which means the first domain becomes power on at the 0th cycle, the second domain becomes power on at the 14th cycle, and so forth.

In order to discuss the relationship between power-up deadline and the minimum violation area, we analyze the eight-domain case with different power-up deadlines. The deadline increases from 20 cycles to 60 cycles with step size of 5 cycles. The minimum violation areas obtained by the proposed method are shown in Fig. 12, where the optimal values from the enumerating method are also shown for comparison. From the figure, we can see that the minimum violation area decreases as the deadline increases. This means, the less tight deadline will give every domain more choices to make power-up sequence arrangement, and therefore reduces the voltage noise on power network. Fig. 12 also helps designers to make tradeoff between the power-up schedule and the induced power noise. Long power-up stage introduces less power noise, but defers following tasks. In Fig. 12, the curves of the SA based method and the enumerating method match with each other very well. This suggests the high accuracy of the proposed SA based algorithm.

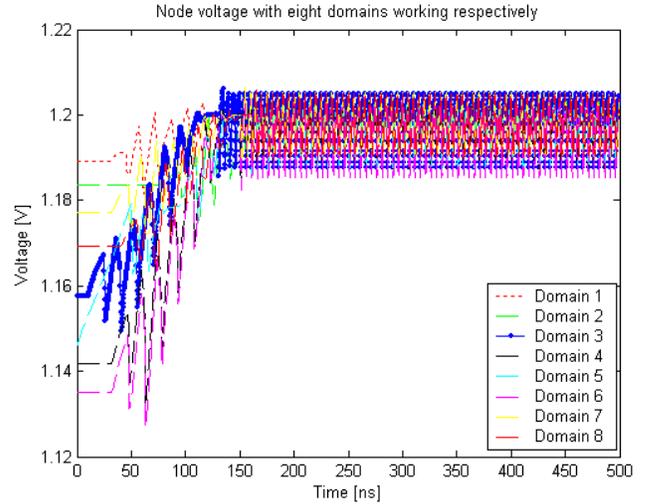


Fig.11. Voltage waveforms with only one domain working.

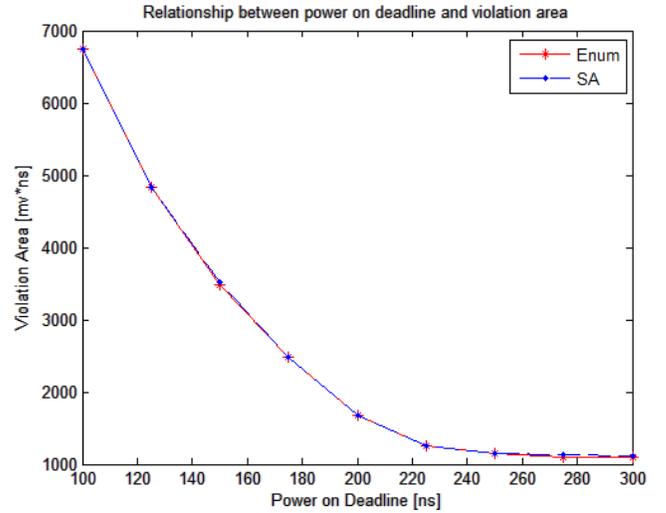


Fig.12. Relationship between the power-up deadline and the minimum violation area.

### B. More Results with Different Test Cases

There are ten test cases with the number of domains varying from 4 to 20. Three methods to calculate the voltage violation area are compared. The first one is the enumerating method which exhaustively searches all possible power-up sequences. The second one is the proposed SA based method. The last one is the greedy algorithm in Fig. 9, which gets a locally minimal solution. The enumerating method has the worst case complexity of  $O\left(\prod_{i=1}^D L_i\right)$ , where  $L_i$  is the deadline for domain  $i$ . So, its search space grows exponentially with the number of domains, and it is computationally prohibitive for large cases.

For four small cases with fewer domains, the computational results of the three methods are listed in Table I. The violation areas obtained from the three methods are in the third, fourth, and fifth columns, respectively. The enumerating method's result, A\_Enum, is the golden value

TABLE I  
Comparison between the enumerating method and SA based method for small cases

| Circuit Name | # of Domain | A_Enum (mV·ns) | A_SA (mV·ns)  | A_Greedy (mV·ns) | T_Enum (s) | T_SA(s) | Speed Up (SA over Enum.) |
|--------------|-------------|----------------|---------------|------------------|------------|---------|--------------------------|
| Ckt 1        | 4           | 196.6 (1)      | 196.6 (1.00)  | 675.9 (3.44)     | 36         | 2       | 18.0                     |
| Ckt 2        | 4           | 489.6 (1)      | 497.5 (1.02)  | 1133.1 (2.31)    | 37         | 2       | 18.5                     |
| Ckt 3        | 8           | 1356.8 (1)     | 1377.9 (1.01) | 11289 (8.32)     | 4192       | 10      | 419.2                    |
| Ckt 4        | 8           | 1143.6 (1)     | 1143.6 (1.00) | 1734.4 (1.52)    | 2313       | 9       | 257                      |

TABLE II  
Comparison between the enumerating method and SA based method for large cases

| Circuit Name | # of Domain | A_Enum (mV·ns) after 10 hours | A_SA (mV·ns) | A_Greedy (mV·ns) | T_SA (s) |
|--------------|-------------|-------------------------------|--------------|------------------|----------|
| Ckt 5        | 12          | 4452.9                        | 1128.5       | 3002.8           | 28       |
| Ckt 6        | 12          | 4650.5                        | 1240.2       | 3001.4           | 34       |
| Ckt 7        | 16          | 3712.3                        | 2572.6       | 4411.3           | 56       |
| Ckt 8        | 16          | 2537.1                        | 1310.4       | 3064.9           | 65       |
| Ckt 9        | 20          | 4081.4                        | 1859.3       | 14136.1          | 114      |
| Ckt 10       | 20          | 3616.9                        | 1713.3       | 13419.5          | 96       |

of the minimal violation area. For the results of other two methods, their ratio to A\_Enum is given in the following parentheses (see Table I). We notice that the SA based method has very high accuracy, whose error is less than 2%. And, the greedy solution is not accurate, whose result is from 1.5x to 8x larger than the accurate value. In last three columns of Table I, the computational time comparison between the enumerating and SA based methods is exhibited. For the four small cases, the speedup ratio of the proposed method is from several tens to several hundreds.

The six larger cases involve 12, 16, or 20 domains. So, the search space becomes very huge, and the enumerating method cannot complete the computation within 10 hours. We manually terminate the enumerating method after 10 hours' runtime and present the best solution it finds. The computational results are listed in Table II. As we can see, the proposed method gives much better solution than the enumerating with 10 hours, with very short computational time. And, the result from the latter may be even worse than the greedy solution. On the other hand, the greedy solution is also far from the actual minimum violation area. To compare the computational speed, we find out that the proposed SA based method is at least 300x faster than the enumerating method for the large cases.

## VI. Conclusions

As the multiple power domain design is becoming popular in modern SoC design, more research focus on power-up sequencing to minimize the overall noise. In this paper, we formulate the power-up sequencing problem in the power domain level, and prove its NP-Completeness. Hence, an SA based algorithm with ordering and greedy initial solution is proposed. Experiments with industry cases show that our SA based algorithm is as accurate as optimal solution, but much more efficient than enumerating. The proposed algorithm is helpful to find a power-up sequence for all the domains with minimal noise. Furthermore, we also show the trade off between the time to power up and the

overall noise.

## Acknowledgments

The authors would like to acknowledge the support of NSF CCF-0811794 and California MICRO Program.

## References

- [1] Y. Ogasahara, T. Enami, M. Hashimoto, et al., "Validation of a full-chip simulation model for supply noise and delay dependence on average voltage drop with on-chip delay measurement," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 10, pp. 868-872, Oct. 2007.
- [2] G. Bai, S. Bobba, and I. N. Hajj, "Simulation and optimization of the power distribution network in VLSI circuits," in *Proc. ICCAD*, Nov. 2000, pp. 481-486.
- [3] J. Fan, I-Fan Liao, S. X.-D. Tan, et al., "Localized on-chip power delivery network optimization via sequence of linear programming," in *Proc. ISQED*, Mar. 2006.
- [4] Z. Qi, H. Li, S. X.-D. Tan, et al., "Fast decap allocation algorithm for robust on-chip power delivery," in *Proc. ISQED*, Mar. 2005, pp 542-547.
- [5] W. Zhang, Y. Zhu, W. Yu, et al., "Finding the worst voltage violation in multi-domain clock gated power network," in *Proc. DATE*, Mar. 2008, pp 537-540.
- [6] M. Keating, D. Flynn, R. Aitken, A. Gibbons, K. Shi, *Low Power Methodology Manual*, Springer, 2007.
- [7] J. Salmon, N. Dour, "Circuit for independent power-up sequence of a multi-voltage chip," *US Patent 6236250*, 2001.
- [8] N. Ranjan, "Mixed voltage, multi-rail, high drive, low noise, adjustable slew rate input/output buffer," *US Patent 5862390*, 1999.
- [9] Y. Kanno, et al., "Hierarchical power distribution with 20 power domains in 90nm low-power multi-CPU processor," in *Proc. ISSCC*, Feb. 2006, pp. 2200-2209.
- [10] T. Hattori, et al., "A power management scheme controlling 20 power domains for a single-chip mobile processor," in *Proc. ISSCC*, Feb. 2006, pp. 2210-2219.
- [11] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Mathematics-Worth Publishers, Incorporated, 1979.