



Clock buffer polarity assignment under useful skew constraints



Deokjin Joo, Taewhan Kim*

Department of Electrical and Computer Engineering, Seoul National University, Seoul, Republic of Korea

ARTICLE INFO

Keywords:

Clock trees
Setup/hold times
Useful clock skew
Timing
Clock polarity assignment
Power/ground noise

ABSTRACT

Clock trees, which deliver the clock signal to every clock sink in the whole system, switch actively at high frequency, which makes them one of the most dominant sources of noise. While many clock polarity assignment (PA) techniques were proposed to mitigate the clock noise, no attention has been paid to the PA under useful skew constraints. In this work, we show that the clock PA problem under useful skew constraints is intractable and propose a comprehensive and scalable clique search based algorithm to solve the problem effectively. In addition, we demonstrate the applicability of our solution by extending it for PA under delay variation environment. Through experiments with ISPD'10 benchmark circuits, we show that our proposed clock PA algorithm is able to reduce the peak noise by 10.9% further over that of the conventional global skew bound constrained PA. Finally, we compare our PA technique against decoupling capacitor embedding technique which is a commonly used method for noise reduction.

1. Introduction

The rapid advancement in CMOS technology scaling has enabled the development of high performance and highly integrated chips. However, the increased power density requires the scaling of supply voltages to keep the power consumption under budget. This scaling then leads to the decrease of the noise margins, causing circuits to be more susceptible to power/ground noise. Power/ground noise is caused by simultaneous switching of circuits as they draw/drain current from/to the power/ground rails, inducing voltage fluctuations. Especially in synchronous high speed circuits, clock network is a major source of the noise, where its clock buffers switch simultaneously at high frequency, at the rising and falling edges of the clock signal. This noise adversely affects not only the signal integrity of chip but also the circuit performance due to the voltage drop/rise at the power/ground supply voltage rails [1]. To mitigate this problem, several techniques have been developed, including decoupling capacitor insertion, clock skew scheduling and polarity assignment (PA).

Decoupling capacitors (decaps) are the most popular and straight forward method for reducing power supply noise. This technique, which has been in use for over 40 years [2], is achieved by intentionally placing a large capacitor in the power distribution network. Although a powerful technique, decaps incur large area overhead.

Clock skew scheduling is a technique for improving circuit robustness, which is sometimes referred to as *useful skew scheduling*. This is done by deliberately introducing clock signal arrival time differences at

the clock sinks to meet certain goals that the designer sets.¹ Fishburn [3] borrowed time from paths with more time slack for more critical paths to improve circuit performance. Wang et al. [4] proposed to utilize clock skew to improve timing yield. There are several works (e.g., [5–9]) that have utilized the clock skew to reduce simultaneous switching noise by spreading the peak current over time domain. Benini et al. [5] firstly proposed to reduce the peak current through clock skew scheduling. Vittal et al. formulated and solved the problem as 0–1 integer linear program. Lam et al. [7] proposed a graph based approach. Later, Huang et al. [8] extended the problem to consider multi-domain clock systems. Most recently, Vijayakumar et al. [9] proposed a fast heuristic method that can find a near-optimal solution within minutes on large circuits.

On the other hand, clock polarity assignment (PA) techniques provide another means to disperse noise [10–14]. Those techniques involve replacing some of the buffers in the clock trees to inverters, thus changing the *polarity* of the clock signal. Then, to compensate for the flip-flops (FFs) which are affected by the introduction of inverters, they are replaced with negative-edge triggered FFs. Fig. 1 illustrates the idea behind the clock PA. Buffers, which are constructed by cascading two differently sized inverters, draw larger current from the power rail at the rising edge of the clock signal compared to the falling edge, as illustrated in Fig. 1(a). Inverters on the other hand behave oppositely, as shown in Fig. 1(b). Consequently, by mixing buffers and inverters in a clock tree, the designer is able to divert the timing of the switching current. We divide the time period into many intervals, which are

* Corresponding author.

E-mail address: tkim@ssl.snu.ac.kr (T. Kim).

¹ (Global) clock skew is defined to the difference between the latest and the earliest clock signal arrival times at the clock sinks.

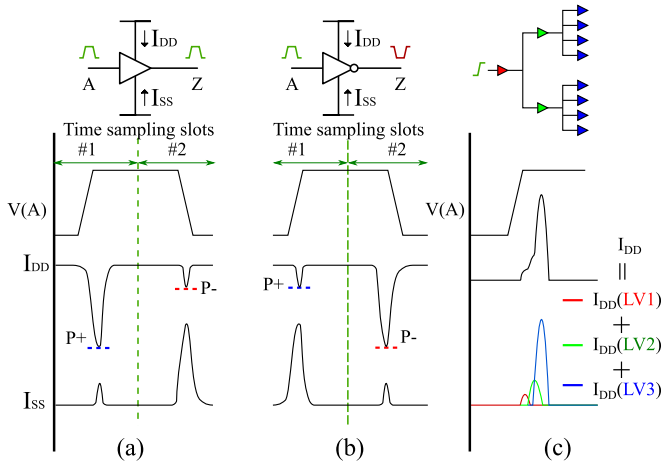


Fig. 1. The idea behind clock buffer polarity assignment. Mixing buffers and inverters in a clock network enables to spread I_{DD}/I_{SS} current over the time period. (a) Buffers draw larger I_{DD}/I_{SS} current at rising/falling edge of clock signal. (b) Inverters exhibit opposite behavior of (a). (c) Peak noise occurs around the time when the leaf buffers (blue color) switch. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

called *time sampling slots* in this work, and take the maximum I_{DD}/I_{SS} current values in each slot to calculate the upper bound of the noise values. $P+/P-$ are used to denote the peak current values of I_{DD} at the rising/falling edge of the clock, as shown in Fig. 1.

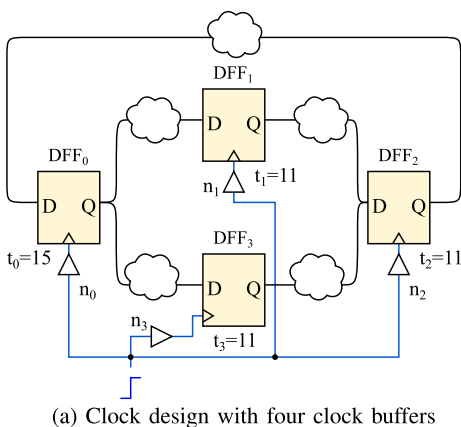
Nieh et al. [10] firstly proposed the idea of PA. They divided the clock tree into two subtrees and replaced the root driving buffer in one of the subtrees with an inverter, assigning polarity onto half of the clock tree. Although this reduced the noise for the whole chip, the buffers in each subtree still switched simultaneously that locally, the noise still remained as a problem. To address this issue, Samanta et al. [11] proposed to mix buffer and inverters throughout the clock tree, so that roughly half of the clock buffering elements are inverters. While this reduced the clock noise significantly, this increased the clock skew. Chen, Ho, and Hwang [12] focused the PA on *leaf buffering elements*. They observed that leaf buffers, which are directly incident to sinks (FFs), outnumber the non-leaf buffers, making leaves the dominant source of noise emitted by the clock tree. (See Fig. 1(c).) Hence, by assigning polarity only to the leaves, it was possible to reduce the noise while minimally impacting the clock skew. Jang et al. [15] proposed an integrated approach to the PA combined with buffer sizing to utilize the clock skew and further reduce the noise. Lu and Taskin [16] attempted to assign polarity to non-leaf at the expense of the increase of clock

skew. Later, in [13] they proposed to perform skew tuning on the polarity assigned clock trees to reduce the clock skew at the worst corner. Joo and Kim [14] proposed a method for better estimation of noise by fine-grained sampling on the noise current waveform. Kang and Kim [17] considered the delay variations in the PA. They performed PA which minimizes the power/ground noise while meeting the skew yield constraint caused by the clock skew variation. Recent researches show that polarities may be adjusted after chip fabrication by using XOR gates and double edge triggered flip-flops, which makes clock-gating-mode aware noise reduction possible [18,19].

While there are plenty of research works that addressed the PA problem, one common feature of all previous works is that they are all *global clock skew bounded* approach. However, for high performance circuits, it is necessary to set a tight clock skew bound since the available time margin is not enough. This means that it becomes much harder to exploit the clock PA under the tight clock skew bound constraint to minimize the worst noise. In contrast, the clock PA under useful skew constraints will be more effective than the global clock skew bound constrained PA in the sense that it is able to check the setup and hold time constraints between sinks *individually* in the course of the PA where some sink pairs have loose time margins while some have tight ones. To the authors' knowledge, this is the first work to assign clock polarity under useful clock skew constraints. In this work, we focus on the leaf clock buffering elements only and propose a comprehensive solution to the problem of clock PA integrated with buffer/inverter sizing to reduce clock switching noise (a preliminary version of this paper was presented in [20]). Precisely, (1) we show the PA problem under useful skew constraints is NP-complete; (2) we propose a clique search based scalable algorithm that is able to trade-off between the solution quality and run time; (3) the proposed algorithm produces library based (practical) solution, so that the optimized buffers and inverters can be taken from the given library; (4) we demonstrate that our method can be effectively extended to the PA under delay variation environment; and (5) we compare the effectiveness of our PA technique against the decap embedding technique. Also, we observe that decap embedding and PA can be applied without conflict.

2. Motivational example

Consider a small clock tree shown in Fig. 2(a). It has four clock sinks which are labeled as DFF_0, \dots, DFF_3 , each of which is driven by its distinct leaf clock buffer, as indicated by n_0, \dots, n_3 . The initial clock signal arrival times to DFF_0 through DFF_3 are 15, 11, 11, and 11, respectively, as indicated by t_0, t_1, t_2 , and t_3 . Assume that the setup and hold time constraints are pre-calculated and given as in Fig. 2(b). Given



$$\begin{matrix} -3 & \leq & t_0 - t_1 & \leq & 2 \\ -5 & \leq & t_1 - t_2 & \leq & 4 \\ -3 & \leq & t_0 - t_3 & \leq & 3 \\ -4 & \leq & t_3 - t_2 & \leq & 2 \\ -3 & \leq & t_2 - t_0 & \leq & 2 \end{matrix}$$

(b) Setup and hold time constraints

Type	Δt	Noise	
		P+	P-
B1	0	10	3
B2	+2	12	3
I1	0	3	9
I2	+1	3	11

(c) Library of buffers/inverters

#	Cell Type Selection				Clock Skew	Noise		
	n_0	n_1	n_2	n_3		P+	P-	Worst
1	B1	B2	B2	B2	2	46	12	46
2	B1	B2	B2	I2	3	37	20	37
3	B1	B2	I2	B2	3	37	20	37
4	B1	B2	I2	I2	3	28	28	28
5	I1	B2	B2	B2	2	39	18	39
6	I1	B2	B2	I2	3	30	26	30
7	I1	B2	I2	B2	3	30	26	30
8	I1	B2	I2	I2	3	21	34	34

(d) Eight feasible PA with sizing

Fig. 2. An illustration of clock buffer PA problem. (a) A small clock tree with four clock buffers and four sinks (FFs). (b) Setup and hold time constraints between the sinks. (c) Available types of buffer/inverter and their impact on clock signal arrival times (Δt). Our brute force analysis with given information reveals that of $4^4 = 256$ search space, only eight are *feasible* assignments which cause no time constraint violation. (d) The eight feasible clock PA of the design in (a) using the library in (c) that satisfies the time constraints in (b). Assignment #4 leads to the lowest peak noise while the conventional clock skew bound constrained PA finds assignment #5 as the least peak noise, which is 39% higher than that in assignment #4.

that each of the four clock buffers is initially a buffer instance of type $B1$, we can calculate the arrival time change Δt experienced at its driven FF resulting from the replacement with an instance of other buffer/inverter type, as tabulated in Fig. 2(c). Note that since two clock buffers of the same type in different locations may drive different load capacitances, even when they are to be replaced with another clock buffers of the same type, it is practically required to separately calculate the two values of Δt and their peak power and ground ($P+$ and $P-$) currents. However, for the sake of simplicity, let us assume that all the clock buffers in the example have the same Δt values and peak power and ground currents.

Now, we are ready to perform PA/buffer sizing. In this example, we resort to brute force method to explore the design space completely. Out of 4^4 combinations, it is found that only eight are *feasible* in that they cause no violation to the constraints given in Fig. 2(b). The eight assignments are listed in the table of Fig. 2(d). The upper bound values of power/ground noise of an assignment are calculated by summing the $P+/P-$ values given in the library of the assigned types. For example, #3 assignment (B1, B2, I2, B2) has $P+$ value of 37 since the $P+$ values are given as (10, 12, 3, 12) in the table in Fig. 2(c) and the sum of them is 37. $P-$ can be calculated likewise. After the computation of noise upper bounds, the assignment with the minimum worst case ($=28 = \min(\max(P+, P-))$) noise is selected as the best assignment, which is assignment #4.

On the other hand, the previous clock PA and buffer sizing algorithms can only take one clock skew bound for their clock skew specification. Thus, to satisfy all setup and hold time constraints, the designer must select the tightest constraint as the clock skew bound ($=2$ in this example). Under this tight constraint, only assignments #1 and #5 are feasible, which results in the peak noise of 39, which is 39% higher than that of the useful clock skew optimization result. This example clearly shows that the bounded skew approach may severely limit the exploration of search space and a useful clock skew approach is essential to fully explore the search space in order find a clock PA with a minimal peak noise.

3. Problem formulation

Problem 1 (USEFULMIN). (PA combined with buffer sizing for peak current minimization under setup and hold time constraints) Given a buffer library B , an inverter library I , a set L of leaf buffering elements, and a set S of time sampling slots, find a mapping function $\phi: L \mapsto \{B \cup I\}$ that minimizes the quantity of

$$\max_{s \in S} \left\{ \sum_{e_i \in L} \text{noise}(\phi(e_i), s) \right\} \quad (1)$$

under a set C of setup and hold time constraints:

$$\text{LB}(e_i, e_j) \leq t_i(\phi) - t_j(\phi) \leq \text{UB}(e_i, e_j), \quad \forall i, j, i \neq j$$

where $i \neq j$ and for all $(e_i, e_j) \in L \times L$ and $\text{LB}(e_i, e_j)$. $\text{UB}(e_i, e_j)$ can be $-\infty$ and ∞ , respectively, if there is no time constraint between e_i and e_j . The term $\text{noise}(\phi(e_i), s)$ is the value of peak current estimation at a time sampling slot s caused by the switching of e_i when it is assigned with $\phi(e_i) \in \{B \cup I\}$.

Note that $P+$ and $P-$ values in the motivational example are short names for $\text{noise}(\phi, s_1)$ and $\text{noise}(\phi, s_2)$, respectively, when $|S| = 2$, in which the peak current noise sampling slots s_1 and s_2 will be used as the high/low periods in the clock cycle. Increasing the number of time sampling slots can improve the resolution of noise estimation. Moreover, it is desirable to sample both I_{DD} and I_{SS} as slots since the objective is to minimize the worst current.

4. Proof of intractability

Since Jang's work [15], it is known that assigning polarity to

minimize $P+$ and $P-$ noise is intractable. Intuitively, even under the special condition with $|S| = 2$ and no clock skew constraints, since decreasing the noise at the negative edge of the clock increases the noise at the positive edge and vice versa, the task of minimizing the worst of the two noise values is equivalent to well-known PARTITION problem, which is NP-complete. In the following, we formally prove USEFULMIN problem is intractable by showing that the decision version of USEFULMIN problem is NP-complete.

Problem 2 (DECISION-USEFULMIN). For a USEFULMIN instance with (B, I, L, S, C) and a noise limit constant k is there a mapping ϕ such that (i) the value of Expr. (1) is less than or equal to k and (ii) satisfies all given setup and hold time constraints C ?

Problem 3 (PARTITION). Given a list a_1, a_2, \dots, a_N of positive integers, is there a partition, i.e., a subset $A \in \{1, 2, \dots, N\}$ such that the equality

$$\sum_{i \in A} a_i = \sum_{i \notin A} a_i$$

is satisfied?

Theorem 1. PARTITION is NP-complete.

Theorem 2. DECISION-USEFULMIN is NP-complete.

Proof. Firstly, DECISION-USEFULMIN is NP. When a problem instance and a solution candidate ϕ is given, the computation of $\text{noise}(\phi)$ and the verification of setup and hold times are achievable in polynomial time, as the clock signal arrival times can be calculated in polynomial time and there are $O(|L|^2)$ clock skew constraints.

Let us now map any instance of PARTITION into DECISION-USEFULMIN as follows, in polynomial time and space. Let there be only one type of buffer and inverter in the library:

$$B = \{b\} \quad \text{and} \quad I = \{v\}$$

Two slots are allocated in the set of time sampling slots S , i.e., $S = \{s_1, s_2\}$ and $|S| = 2$. For set of leaf buffering elements L , one element e_i is for allocated for each positive integer a_i so that $|L| = N$. Now, we define noise values as follows so that they correspond to a_i values in PARTITION:

$$\text{For all } i = 1, 2, \dots, N,$$

$$\begin{aligned} \text{noise}(\phi(e_i) = b, s_1) &:= a_i \text{noise}(\phi(e_i) = b, s_2) := 0 \text{noise}(\phi(e_i) = v, s_1) \\ &:= 0 \text{noise}(\phi(e_i) = v, s_2) := a_i \end{aligned}$$

For the set C of setup and hold time constraints, no constraint is imposed, so that e_i may be mapped freely:

$$\text{LB}(e_i, e_j) \leq t_i(\phi) - t_j(\phi) \leq \text{UB}(e_i, e_j), \quad \text{where } \text{LB}(e_i, e_j) = -\infty$$

$$\text{UB}(e_i, e_j) = \infty \quad \forall i, j, i \neq j$$

Since $|S| = 2$, Expr. (1) can be re-written as:

$$\max \{ \text{Noise}(s_1), \text{Noise}(s_2) \}, \quad (2)$$

where

$$\text{Noise}(s_1) = \sum_{e_i \in L} \text{noise}(\phi(e_i), s_1) \text{Noise}(s_2) = \sum_{e_i \in L} \text{noise}(\phi(e_i), s_2)$$

Note that $\text{Noise}(s_1)$ and $\text{Noise}(s_2)$ correspond to $\sum_{a_i \in A} a_i$ and $\sum_{a_i \notin A} a_i$ in PARTITION, respectively, by construction of noise .

Finally, we define the noise constraint $k := \frac{1}{2} \sum_{i \in \{1, 2, \dots, N\}} a_i$. Since $\text{noise}(\phi(e_i), t) \geq 0 \quad \forall i$, for Expr. (2) to be less than or equal to k , $c = \text{Noise}(s_1) = \text{Noise}(s_2)$ must hold: while migrating noise from slot s_1 to s_2 decreases $\text{Noise}(s_1)$, $\text{Noise}(s_2)$ must increase; $\text{Noise}(s_1) < k$ implies $\text{Noise}(s_2) > k$ and vice versa. Hence, unless both the Noise values are equal to k , Expr. (2) $> k$.

The DECISION-USEFULMIN solution instance found by this mapping can be converted back to the PARTITION solution instance A in $O(|L|)$ time, by adding i to set A when $\phi(e_i) = b$. \square

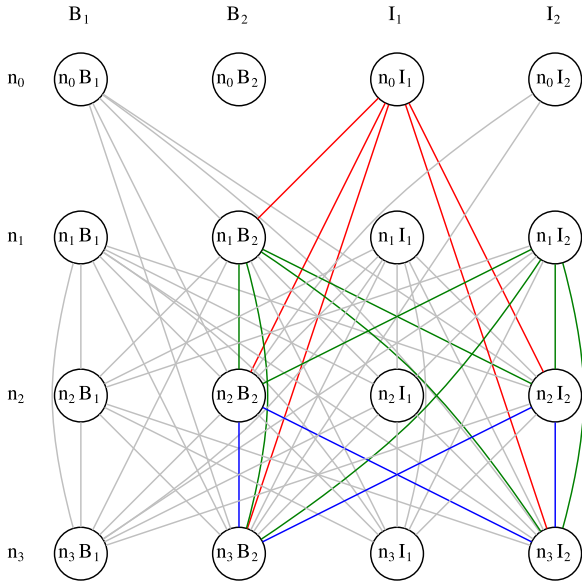


Fig. 3. Transformation of the problem instance in Fig. 2 into a search problem in a graph $G(V, E, W)$. $|L| \times (|B| + |I|)$ vertices are created, in which each vertex represents a mapping, e.g., vertex (n_1, I_1) is the representation of assigning leaf node $n_1 \in L$ with $\phi(n_1) = I_1 \in I$. Two vertices are connected if their assignments do not cause time violation and no conflict of assignment occurs. Any clique of size $|L|$ in G is a feasible solution.

5. Proposed method

5.1. Approaches for solving USEFULMIN

(1) *ILP formulation and LP relaxation:* It is possible to formulate USEFULMIN problem into 0–1 integer linear programming (ILP). However, its linear programming (LP) relaxation is of little help for our problem. The reason is that the ILP formulation has underlying maximum clique search process, as will be shown later, and the LP relaxation yields only $(0, 1/2, 1)$ for each variable. It is known that the variables are rarely mapped to 0 or 1 [21].

(2) *Formulating into maximum clique problem:* Consider the USEFULMIN problem instance of the clock tree in Fig. 2, which is then represented by a weighted graph $G(V, E, W)$ as shown in Fig. 3: (i) for each pair $(n_i, B_j/I_j)$ of leaf buffers $n_i \in L$ and buffers/inverters $B_j/I_j \in B \cup I$, there is a unique vertex in V , and $|V| = |L| \times (|B| + |I|)$. (ii) There exists an edge in E between vertices $(n_i, B_j/I_j)$ and $(n_k, B_l/I_l)$, $i \neq k$, if and only if assigning n_i with B_j/I_j and n_k with B_l/I_l causes no setup

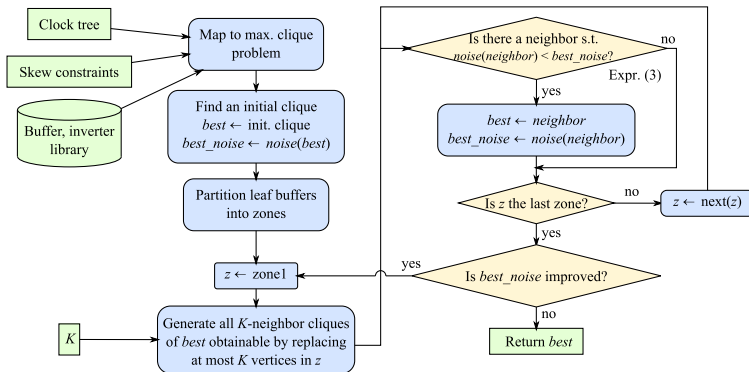
and hold time violation. For example, there is no edge between (n_0, I_1) and (n_1, I_2) in Fig. 3 since their mapping leads to the violation of $-3 \leq t_0 - t_1 \leq 2$ in Fig. 2(b). Note that the vertices in the same row in Fig. 3 have no edge between them. This forbids a leaf buffer to be assigned to more than one type of buffer/inverter. In addition, there will be edges between all possible pairs of nodes in rows marked n_1 and n_3 since there is no skew constraint at all in the initial clock tree between the sinks corresponding to n_1 and n_3 . (iii) Weight $w_i \in W$ assigned to a node $(n_i, B_j/I_j)$ represents the set of power/ground currents at the sampling slots in S when B_j or I_j assigned to n_i switches. Let $w_i(s_j)$ be the power/ground current at sampling slot s_j when the buffer/inverter assigned to leaf buffer e_i switches. Then, the problem of finding the clock PA under the useful skew constraints is equivalent to the problem of finding a clique $Q \subset V$ of size $|L|$ in G that minimizes the (noise) quantity of

$$\max_j \left\{ \sum_{q_i \in Q} w_i(s_j) \right\}, \quad j = \{1, \dots, |S|\} \quad (3)$$

Since there is no edge between the vertices in the rows of G in Fig. 3, the problem of finding $|L|$ -clique with the minimum value of Expr. (3) can be translated to finding a maximum clique in G with the minimum value of Expr. (3). Thus, if the size of maximal clique in G is less than $|L|$, there is no feasible PA that meets all useful skew constraints. For example, in Fig. 3, there are eight cliques of size 4 that can be found from the subgraph defined by vertices $\{(n_0, B_1), (n_0, I_1), (n_1, B_2), (n_2, B_2), (n_2, I_2)\}$, and $\{(n_3, B_2), (n_3, I_2)\}$, which correspond to the eight feasible assignments in Fig. 2(d). Among the assignments, assignment #4 produces the least value of Expr. (3).

(3) *Scalable algorithm for clique exploration:* The problem of finding a maximum clique with least cost is known to be not only intractable but also hard to approximate [21]. Hence, we propose to employ local search heuristics to find a good feasible solution, as summarized in Fig. 4.

We start by mapping the USEFULMIN problem instance to a maximum clique problem instance. To use local search heuristic, we first need to find an initial clique of cardinality $|L|$ to start the local search. A trivial solution is the unoptimized one, where no buffers are changed. However, note that, since the initial clique for the local search determines the quality of the final solution, it is desirable to use a previous skew bound constrained clock PA/buffer sizing algorithm to find an initial clique. Then, we iteratively search for cliques that yield better results. We search them by finding K -neighbors of the clique found in the current iteration. Clique X is called a K -neighbor of Y if X can be formed by replacing K or less vertices of Y . Since the designer is



```

1: function USEFULMIN( $B, I, L, S$ )
2:   parameter  $K$ 
3:    $G(V, E) \leftarrow \text{map\_to\_clique}(B, I, L, S)$ 
4:    $\text{best\_clique}, \text{best\_noise} \leftarrow \text{find\_init\_clique}(B, I, L, S)$ 
5:    $Z \leftarrow \text{partition\_zones}(L)$ 
6:    $\text{prev\_clique} \leftarrow \text{nil}$ 
7:    $\text{prev\_best} \leftarrow \infty$ 
8:   while  $\text{best\_noise} < \text{prev\_best}$  do
9:      $\text{prev\_best} \leftarrow \text{best\_noise}$ 
10:     $\text{prev\_clique} \leftarrow \text{best\_clique}$ 
11:    for  $z \in Z$  do
12:       $N \leftarrow K\_neighbors(K, z, \text{best\_clique})$ 
13:       $\text{best\_clique}, \text{best\_noise} \leftarrow \min_{c \in N}(\text{noise}(c))$ 
14:    end for
15:  end while
16:  return  $\text{best\_clique}, \text{best\_noise}$ 
17: end function

```

Fig. 4. The flow of our USEFULMIN algorithm. The input problem instance is mapped to a weighted maximum clique problem instance. In the mapped problem, any clique of maximum cardinality is a clock PA/buffer sizing that is compliant to all input clock setup and hold time constraints. Since finding maximum clique is a difficult task, heuristics is applied to perform local search to find a good solution. Starting from the clique/solution corresponding to the input clock tree with no time violation, at most K vertices in the clique are replaced to generate all K -neighbors of the clique. If any resulting clique improves the noise value, the clique is kept as the current best clique. Based on the fact that noise is a local phenomenon, search space is further reduced by partitioning the vertices into zones by local proximity and allowing at most K vertices to be replaced within the zone of interest at each step. The local search is repeated until there is no noise reduction.

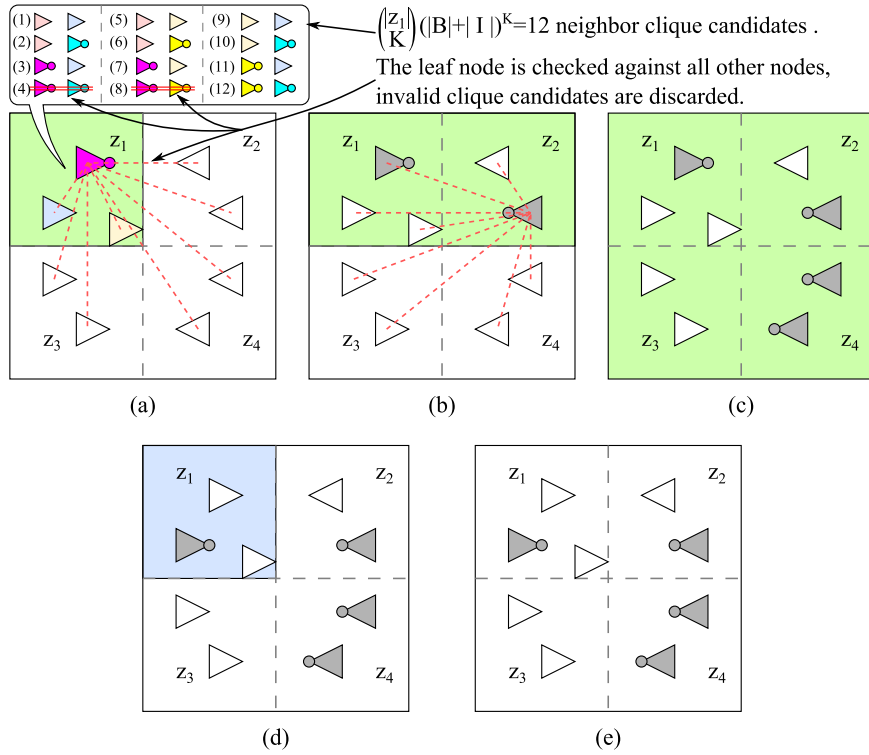


Fig. 5. An example illustrating the procedure of USEFULMIN algorithm. (a) The first zone z_1 is optimized. Each of the 2-neighbor clique candidates is checked if it forms a clique globally. (b) Among the candidates, the one with the least value of Expr. (3) is frozen and the optimization continued to z_2 . (c) All zones are optimized. The optimization restarts from z_1 . (d) A better assignment in z_1 has been discovered. (e) USEFULMIN terminates when no improvement is made.

able to control the value of parameter K , it is possible to trade-off between the solution quality and run time.

Theoretically, raising K increases the search space significantly since the number of candidates to be examined is $O\left(\binom{|L|}{K} (|B| + |I|)^K\right)$. However, by reflecting the fact that noise is a local phenomenon, we can partition the leaf buffers in L into zones by their proximity and perform the optimization in zone-by-zone manner, which greatly reduces the search space: for each zone, we find K -neighbor cliques where the K vertices are only chosen from the zone. From the K -neighbors, we keep the neighbor clique with the least noise as the new best clique and move on to the next zone. When all zones are visited, we start the search again from the first zone, as the new best clique may have better neighbor cliques. This exploration is repeated until no improvement is made.

Fig. 5 shows an example of the execution of the zone based USEFULMIN algorithm. Assume that $K=2$ and $|B| + |I| = 2$. The leaf buffering elements are partitioned into zones by their locations. In Fig. 5(a), zone z_1 is optimized. Since there are $|z_1| (=3)$ leaf buffering elements in the zone and $K=2$, $\binom{|z_1|}{K} (|B| + |I|)^K = 12$ 2-neighbor clique candidates are generated from this zone. Each candidate is then checked if it globally forms a clique. Among the candidates that form cliques, the one with the least value of noise (Expr. (3)) is frozen and the optimization continues to the next zone, as shown in Fig. 5(b). This process continues until there are no more zones to optimize, as shown in Fig. 5(c). Since the new clique in Fig. 5(c) has new neighbor cliques, the zone-by-zone optimization is repeated, subsequently generating results in Fig. 5(d) and (e).

The run time analysis of the zone based algorithm is as follows. Suppose there are $|Z|$ zones. Then, there are $n = |L|/|Z|$ leaf nodes in each zone on average. Since there are $O\left(\binom{n}{K} (|B| + |I|)^K\right)$ K -neighbor candidates for each zone, $O(n^{K-1} (|B| + |I|)^K |L|)$ K -neighbors are searched in the whole circuit. The overall runtime of a single iteration is $O(n^{K-1} (|B| + |I|)^K |L|) \times (O(K|L|) + O(2|S|))$, where $O(K|L|)$ time is used for checking if the new set of vertices form a clique and $O(2|S|)$ is for

incrementally computing noise. Simplifying the expression yields $O(Kn^{K-1} (|B| + |I|)^K |L|^2)$, assuming $|S|$ is much smaller than $K|L|$. To speed up the computation, the designer may halt the iterative neighbor search when there is little improvement over the prior iteration. Also, it is possible to set $K=1$ for drastic speedup, if necessary.

5.2. Extension: variation aware polarity assignment

The flexibility of the proposed framework allows integration of other design factors into PA. Such examples may be the PA on multi-corner multi-mode clock trees and the yield aware PA. In this section, we demonstrate yield aware PA. Although several methods of PA had been proposed, relatively less attention has been paid to process variations. Lu and Taskin [13] reported clock skew at the worst corner. By greedily finding the paths that have the greatest difference of clock arrival times and tuning the buffer polarity associated with the paths after the initial clock PA, they were able to trade-off the worst corner clock skew with increased noise. Kang and Kim [17] proposed a more systematic approach. They used statistical static timing analysis (SSTA) on the clock tree to examine the yield of each pair of the leaf buffering elements. Precisely, they calculated the statistic arrival time difference for each pair of leaf buffers which were optimized to satisfy the yield constraint, while noise was minimized heuristically. However, the heuristic has no direct control over the design yield, which is the probability of the whole clock tree meeting all clock skew constraints.

Here, we propose a design yield aware PA heuristic. Given yield constraint γ , we make the following modifications to the proposed algorithm:

- In mapping the problem to a maximum clique problem, we create an edge when the pair of vertices in the graph satisfy the clock tree yield constraint γ . This corresponds to finding pair choices that meet the pair-wise yield constraint in Kang's algorithm.
- During the local search, the cliques now have two parameters *noise* and *yield*. When the current best clique does not satisfy the design

yield γ , the clique with higher yield is kept as the best clique. When the current best and the neighbor cliques both satisfy γ , we keep the one with lower *noise*. The final yield depends on the given initial clique at the start of the local search. To guarantee γ , both the unoptimized input clock tree and the initial clique must satisfy γ . Such clique can be the trivial solution from the unoptimized clock tree.

6. Experimental results

6.1. Experimental setup

The proposed algorithm `USEFULMIN` was implemented in C++ and Python language on a Linux machine with Intel i5-4670 CPU and 8 GB RAM. Clock trees were generated for ISPD'10 high performance clock network synthesis contest benchmarks with the algorithm in [22], using Nangate 45 nm Open Cell Library [23] and employing only `BUF_X8` as buffering elements. Since ISPD'10 benchmarks have only clock sink information and no circuit/individual clock skew constraint information, the setup and hold time constraints were generated randomly within [60, 90] ps range for upper bounds and [-90, -60] ps for lower bounds. To compare the results with that of a skew bound constrained clock PA/buffer sizing approach, we selected and implemented `WAVEMIN` algorithm in [14]. To partition the leaf buffering elements, the circuits were bisected until each zone had no more than 10 elements.

6.2. Assessing the performance of `USEFULMIN` over bounded skew algorithm [14]

The simulation results are summarized in Table 1. # Constr. and skew range columns show the information on clock skew generation. For each benchmark, clock skew constraints were randomly generated so that the number of constraints is equal to 10 times the number of clock sinks. Since `WAVEMIN` is a clock skew bounded algorithm, it was run with the tightest clock skew bound (=60 ps). `USEFULMIN` used the solution from `WAVEMIN` as its initial clique and searched neighbor cliques with $K=5$. It is worth noting that `USEFULMIN` performs better than `WAVEMIN` under tighter clock skew conditions since `USEFULMIN` may exploit looser individual clock skew constraints that `WAVEMIN` cannot. Overall, our algorithm reduces the peak noise by 49.1% and 10.9% further on average over that of the base case and the conventional PA, respectively. Fig. 6 compares our `USEFULMIN` algorithm with optimal ILP formulation. For the ILP solver, SCIP [24] was used. The two curves in Fig. 7 show how `USEFULMIN` algorithm trades the noise value with run time as the setting of parameter K changes in the module of K -neighbor clique search. It reveals that `USEFULMIN` algorithm can effectively control the noise quality while taking the execution time into account.

In many clock PA methods, noise and clock skew are in trade-off relationship to some degree, as reducing the clock skew tends to make the buffers switch simultaneously. This is first reported in [15]. We acquired Fig. 8 by relaxing the clock skew constraints and applying `USEFULMIN`. Clock skew relaxation of 80 ps means that any clock skew constraint that is tighter than 80 ps is relaxed to 80 ps. It can be observed that the relaxation of clock skew constraints leads to the reduction in clock noise.

Finally, Fig. 9 shows the geometric distributions of the voltage fluctuation in circuit 07 in ISPD'10 optimized by `WAVEMIN` and our `USEFULMIN`. The comparison shows that by carefully spreading buffers and inverters while meeting all local skew constraints, `USEFULMIN` reduces the regional noises more effectively than the other.

6.3. Assessing the performance of delay variation aware polarity assignment

Since both skew tuning [13] and pairwise [17] methods are incapable of buffer sizing, we defined $B = \{\text{BUF_X8}\}$ and $I = \{\text{INV_X4}\}$. `INV_X4` was chosen as it has the closest matching clock signal propagation delay to `BUF_X8`. Like other experiments, useful clock skew constraints were randomly generated in the range of [60, 90] ps, where both skew tuning and pair-wise methods were given 60 ps as their clock skew bound. In the experiments, we assumed that buffer/inverter and interconnect delays are spatially correlated normally distributed random variables. Spatial correlations were modeled using the grid model proposed in [25]. During the local search phase in our proposed algorithm, each 3σ value of the distributions were set to 5% of their nominal delays. During exploration, design yield was computed using statistical max operation as proposed in [25]. Given (correlated) normal distributions (d_1, d_2, d_3, \dots), $\max(d_1, d_2, d_3, \dots)$ operation computes approximated normal distribution of the maximum value.

Table 2 summarizes the results of variation aware clock PA. Design yield is obtained by running Monte Carlo simulation on 1000 randomized instances of the clock tree. γ column shows the yield constraint input to the algorithms. In one case (circuit 01), our algorithm fails to meet γ constraint by a few percent point. This is attributed to the fact that the statistical max operation is an approximation rather than the true distribution. However, it is evident that our algorithm is more capable of keeping the yield constraint compared to the other two algorithms, which heuristically increase the design yield. Considering that the yield and noise are in trade-off relation, the increase of yield comes at the cost of increased noise. However, in all circuits, our algorithm maintains comparable noise while keeping the design yield constraint γ . Particularly in circuits 01 and 02 of ISPD'10, our algorithm reduced considerable noise while maintaining higher yield than other algorithms. This shows that our useful skew approach can exploit the individual skew constraints to reduce noise, even under

Table 1

The comparison of our useful skew constrained PA (`USEFULMIN`) and bounded skew constrained PA (`WAVEMIN` [14]). # Constr. and # Lvs columns indicate the number of useful clock skew constraint and the number of leaf buffering elements in the clock tree, respectively. `WAVEMIN` forces tight skew bound of 60 ps for each pair of clock sinks, while `USEFULMIN` exploits individual skew constraints during the PA, reducing the peak noise by 10.9% further. *Area* indicates the total area of leaf buffers/inverters in the optimized clock trees.

ISPD 2010 Ckt	# Cnstr.	# Lvs	Base (no PA)		WAVEMIN [14]			USEFULMIN			Improvement	
			Noise (mA)	Area (μm^2)	Noise (mA)	Area (μm^2)	Run	Noise (mA)	Area (μm^2)	Run time (s)	Noise (%)	Area (%)
01	11 070	221	235.1	22.38	155.3	21.24	56.21	122.1	13.73	2547.70	21.38	35.38
02	22 490	454	433.9	45.97	281.9	43.83	291.00	242.4	29.62	24229.49	14.01	32.42
03	12 000	113	106.1	11.44	45.58	6.00	12.34	42.56	6.49	25.26	6.63	-8.3
04	18 450	116	115.3	11.75	52.85	8.58	44.46	48.21	6.60	145.36	8.78	23.07
05	10 160	49	51.83	4.96	35.12	4.84	12.48	27.78	3.09	16.90	20.90	36.05
06	9810	77	74.26	7.80	43	6.28	10.02	40.25	4.88	14.62	6.40	22.22
07	19 150	131	125.4	13.26	73.26	13.20	34.73	67.13	8.51	127.49	8.37	35.55
08	11 340	89	90.88	9.01	51.43	7.05	13.04	51.05	5.88	12.29	0.74	16.59
Average											10.90	24.13

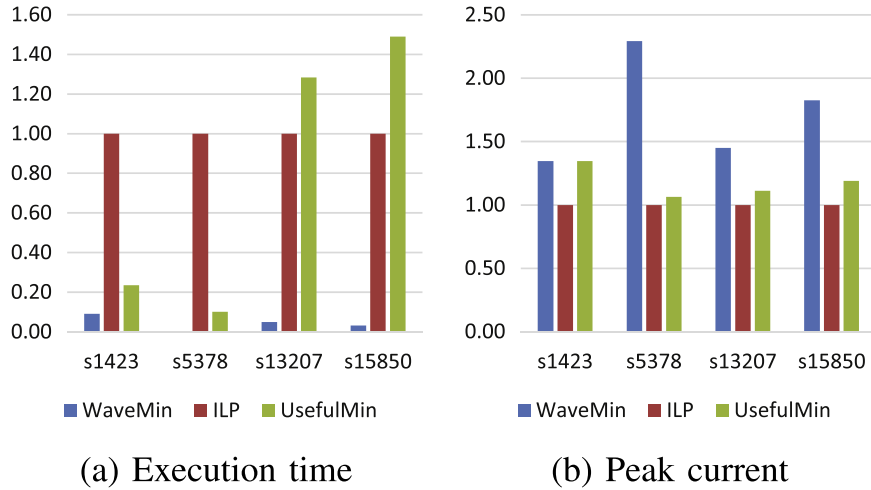


Fig. 6. Normalized comparison of our USEFULMIN, conventional WAVEMIN [14], and optimal ILP formulation. ISCAS'89 benchmarks were used since ISPD'10 benchmarks were too large for the ILP solver. In small circuits, the heuristic iteration can take longer than ILP (0.18 s vs. 0.25 s for s15850). However, as circuits become larger, ILP overtakes (3.9 s vs. 0.8 s in s5378, where s5378 is the largest of the 4 benchmark circuits).

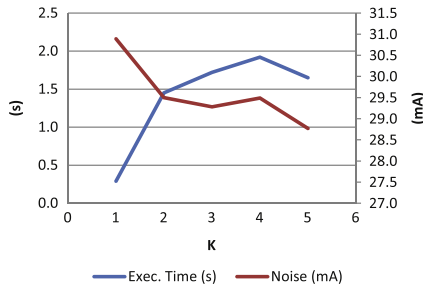


Fig. 7. The effect of parameter K on the optimization of circuit 05 in ISPD'10 by USEFULMIN algorithm. Increasing (decreasing) K expands (shrinks) the search space, so that the likelihood of finding a better (worse) solution increases while spending more (less) time.

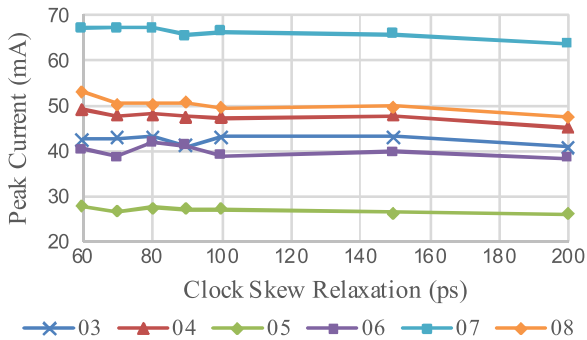


Fig. 8. The relationship between clock skew constraint and clock noise. ISPD 2010 benchmark circuits 03–08 were chosen as they have similar scale of noise. The clock skew constraints are relaxed from 60 ps (no relaxation) to 200 ps and the noise is measured after optimizing the circuits with USEFULMIN. To some degree, the relaxation of the constraints leads to the decrease of the noise, as relaxing the constraints tends to disperse the timings of clock buffers' switching.

clock delay variations.

6.4. Comparison of effectiveness against decoupling capacitors

Inserting on-chip decoupling capacitors (decaps) is a simple and effective technique to mitigate power/ground noise. In order to evaluate the effect of decaps by monitoring the voltage fluctuations, the power delivery network (PDN) must be modeled first. In the previous experiments, the PDN was modeled as a resistive network. Here, we use more refined model as follows.

Figs. 10 and 11 illustrate the PDN model used in the experiments,

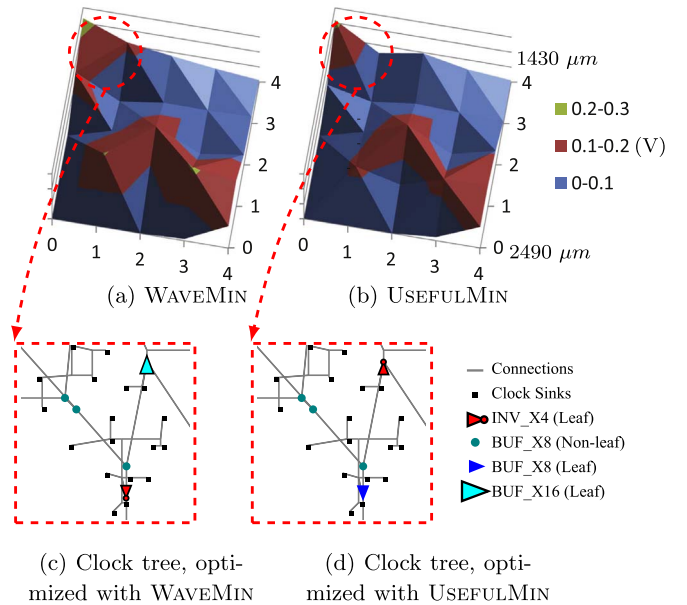


Fig. 9. Geometric distribution of voltage fluctuation in circuit 07 in ISPD'10. Units are in Volts. (a) WAVEMIN [14] optimized the voltage drops successfully. (b) Our USEFULMIN optimized the noise further by exploiting useful skews. Subfigures (c) and (d) show a small section of the clock tree near grid (0, 4). Initially, all of the buffering elements are BUF_X8. Then, (c) WAVEMIN replaces many leaf nodes with buffers and inverters of different sizes for noise reduction. (d) Our USEFULMIN discovers that it is possible to further reduce noise by removing BUF_X16 (cyan triangle) in (c) and allocating BUF_X8 (blue triangle). Note that their locations were changed to satisfy the clock skew constraint. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

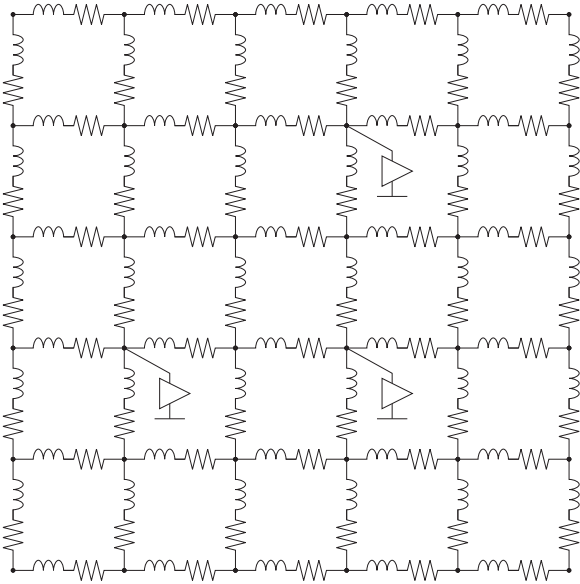
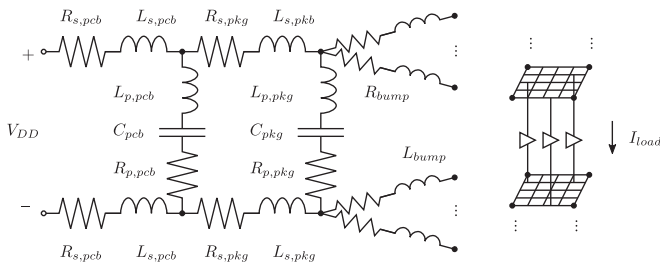
where the on-chip PDN is modeled as described in [26]. Each RL segment of the on-chip PDN shown in Fig. 10 has parameters of $R = 0.007 \Omega/\mu\text{m}$ and $L = 0.5 \text{ pH}/\mu\text{m}$. In the experiments, each grid cell was $100 \mu\text{m} \times 100 \mu\text{m}$ in size. As a result, the PDNs were partitioned into meshes of 32×6 to 131×71 , depending on the benchmark circuit size. The buffering elements were connected to the closest grid point. The off-chip PDN is also modeled with the model and parameters in [27], where the parameters are as shown in Table 3. The input clock signal had 30 ps of slew and frequency of 1 GHz. To measure the voltage fluctuations in the PDN, HSPICE simulation was executed on the ISPD'10 clock network synthesis benchmark circuits.

Table 4 presents the observed voltage fluctuations. We compare the

Table 2

γ column shows the yield constraint input to the algorithms. Design yield is obtained by running Monte Carlo simulation on 1000 randomized instances of the clock tree. The results show that our algorithm is more capable of keeping the yield constraint than the other two algorithms. In ISPD'10 circuits 01 and 02, our algorithm reduced considerable noise while maintaining higher yield than other algorithms. This shows that the useful skew approach can exploit the individual skew constraints to reduce noise, even under clock delay variations.

Benchmark circuit	γ	Average peak current (mA)			Average improvement (%)		Yield (%)			
		Tuning [13]	Pairwise [17]	Ours	vs. [13]	vs. [17]	Base	Tuning [13]	Pairwise [17]	Ours
01	0.83	120.10	123.68	92.93	22.62	24.87	84.70	76.4	73.1	81.1
02	0.39	222.75	230.50	195.54	12.22	15.17	40.70	28.6	27.2	39.4
03	0.98	50.40	51.10	51.47	-2.13	-0.73	99.60	94.9	93.8	98.6
04	0.98	54.79	55.43	58.77	-7.25	-6.02	99.70	94	94.4	98.9
05	0.98	27.93	27.78	27.92	0.03	-0.50	99.99	98.8	98.6	99.7
06	0.98	39.54	39.65	40.12	-1.46	-1.17	100.00	99.4	99.7	100
07	0.98	59.76	62.09	66.45	-11.18	-7.01	99.90	96.3	96.5	99.1
08	0.98	47.59	47.45	45.40	4.60	4.32	100.00	99.7	99.8	99.8
Average					2.18	3.61				

**Fig. 10.** Model of the on-chip power delivery network.**Fig. 11.** Model of the off-chip power delivery network (PDN). The off-chip and on-chip PDNs are connected through bumps.**Table 3**

HSPICE simulation parameters of the PDN in Fig. 11.

$R_{s,pcb}$	0.094 m Ω	$R_{p,pcb}$	0.166 m Ω
$L_{s,pcb}$	21 pH	$L_{p,pcb}$	0 pH
$R_{s,pkg}$	1 m Ω	C_{pcb}	240 μ F
$L_{s,pkg}$	120 pH	$R_{p,pkg}$	0.54 m Ω
R_{bump}	20 m Ω	$L_{p,pkg}$	5.61 pH
L_{bump}	30 pH	C_{pkg}	26 μ F

result of that by WAVEMIN [14] algorithm and our USEFULMIN algorithm here, since the power and ground voltage noise makes more sense when the PDN is fully modeled. In all circuits, our method successfully reduced more power and ground noise than WAVEMIN, with average

reduction of 20.54% and 15.80% for power and ground noise, respectively. Also, both PA algorithms slightly reduced power consumption. However, the amount is not large, as PA techniques work mainly by determining which edge the power consumption will happen. In the following experiments, we only use USEFULMIN method, since it out-performs WAVEMIN.

Table 5 shows the optimization results of USEFULMIN on ISPD'10 benchmarks, with or without decap embedding. While inserting on-chip decap is a simple method of noise reduction, we observed that embedding excess amount of capacitance leads to increase of noise. Since PDNs have a self-resonant frequency and embedding capacitors lower this frequency, excess capacitance could cause the resonant frequency to be near the operating frequency of the chip, increasing the noise [28]. In the experiments, we embedded one “unit” decap for each point in the on-chip PDN mesh. For optimized decap sizing, we tried {0.001, 0.0025, 0.005, 0.01, 0.02, 0.04, 0.08, 0.16, 0.32, 0.64, 1, 2, 5} pF as the unit decap for each un-optimized circuit and selected the best one. For example, Circuit 01 has unit decap of 0.01 pF and has 81 \times 81 mesh points, resulting in total capacitance of 65.61 pF while in Circuit 06, there are 21 \times 10 unit decaps of 0.32 pF, making total capacitance of 67.2 pF.

During the decap embedding, we tried to minimize max(Power Noise, Ground Noise), since the objective is to minimize the worst noise which occurs at different edges of the clock. For example, in Circuit 07, it can be observed that both power and ground noise is reduced while in Circuit 05, ground noise is reduced at the cost of increased power noise, although the worst noise is reduced from 0.296 V to 0.229 V. The average improvement row in Table 5 is computed by averaging the improvement of the worst noise, i.e., average improvement of max(PN, GN). By comparing decap only columns and USEFULMIN columns, it can be observed that USEFULMIN consistently improves noise, more than decap embedding. Also, decap + USEFULMIN columns suggest that PA is a useful technique for the designers which can go on top of decap embedding technique independently, reducing noise even further than applying only one of the techniques.

7. Conclusions

This work proposed a scalable solution to the problem of the clock PA combined with sizing to minimize the peak current induced by the switching of clock signals. Unlike the conventional (global) clock skew bound constrained approaches, our method exploited individual clock skew constraints to further reduce the peak current. Precisely, we formulated the problem into the maximal clique exploration problem and employed a K -neighbor search scheme to trade-off the run time and quality of PA. In addition, we demonstrated that our solution can effectively extend its applicability to the PA under delay variation environment. For designing high speed systems with tight time margin,

Table 4

Comparison of the noise in circuits optimized with WAVEMIN [14] and our USEFULMIN. Voltage noise observed in ISPD 2010 clock network synthesis benchmark circuits. PN and GN columns are power and ground noise, respectively. They are the maximum peak-to-peak voltage fluctuations observed in the PDN mesh. PWR columns are the average power consumption of the circuit in one clock cycle (=1 ns), measured after reaching steady state.

ISPD 2010 Ckt	Base			WAVEMIN [14]			USEFULMIN			Improvement		
	PN(V)	GN (V)	PWR (W)	PN (V)	GN (V)	PWR (W)	PN (V)	GN (V)	PWR (W)	PN (%)	GN (%)	PWR (%)
01	0.437	0.273	0.065	0.146	0.184	0.056	0.119	0.171	0.056	18.49	7.07	0.00
02	0.386	0.661	0.117	0.156	0.29	0.109	0.139	0.25	0.111	10.90	13.79	-1.83
03	0.471	0.58	0.034	0.145	0.228	0.028	0.122	0.212	0.028	15.86	7.02	0.00
04	0.472	0.568	0.041	0.168	0.247	0.036	0.134	0.209	0.035	20.24	15.38	2.78
05	0.205	0.296	0.015	0.115	0.177	0.014	0.072	0.113	0.014	37.39	36.16	0.00
06	0.261	0.403	0.023	0.111	0.178	0.021	0.087	0.146	0.021	21.62	17.98	0.00
07	0.487	0.448	0.036	0.199	0.274	0.032	0.135	0.205	0.03	32.16	25.18	6.25
08	0.343	0.296	0.021	0.091	0.079	0.019	0.084	0.076	0.019	7.69	3.80	0.00
Average										20.54	15.80	0.90

Table 5

Voltage noise observed in ISPD 2010 clock network synthesis benchmark circuits, applying decoupling capacitor and/or USEFULMIN PA method. PN and GN columns are power and ground noise, respectively. They are the maximum peak-to-peak voltage fluctuations observed in the PDN mesh. PWR columns are the average power consumption of the circuit in one clock cycle (=1 ns), measured after reaching steady state. Average improvement row is computed by calculating improvement (%) of max (PN, GN) for each circuit then averaging them.

ISPD 2010 Ckt	Decap size (pF)	Base (0 pF, no PA)			Decap only (no PA)			USEFULMIN (0 pF)			Decap + USEFULMIN		
		PN (V)	GN (V)	PWR (W)	PN (V)	GN (V)	PWR (W)	PN (V)	GN (V)	PWR (W)	PN (V)	GN (V)	PWR (W)
01	65.61	0.437	0.273	0.065	0.419	0.393	0.065	0.119	0.171	0.056	0.118	0.164	0.057
02	93.01	0.386	0.661	0.117	0.554	0.629	0.126	0.139	0.25	0.111	0.184	0.206	0.114
03	30.72	0.471	0.58	0.034	0.507	0.519	0.033	0.122	0.212	0.028	0.154	0.187	0.027
04	51.52	0.472	0.568	0.041	0.513	0.52	0.04	0.134	0.209	0.035	0.156	0.182	0.034
05	103.68	0.205	0.296	0.015	0.229	0.229	0.015	0.072	0.113	0.014	0.08	0.083	0.013
06	67.2	0.261	0.403	0.023	0.336	0.347	0.023	0.087	0.146	0.021	0.101	0.119	0.021
07	33.28	0.487	0.447	0.036	0.442	0.404	0.036	0.135	0.205	0.03	0.147	0.167	0.031
08	27.2	0.337	0.296	0.021	0.323	0.289	0.021	0.084	0.076	0.019	0.083	0.07	0.019
Average improvement (%)					9.73			63.53			68.82		

our proposed approach could be useful in mitigating the clock noise, which otherwise the conventional PA approaches could rarely achieve. Finally, we compared our PA technique against decap embedding technique which is a commonly used method for noise reduction. The results indicated that our PA technique was more effective than decaps. Plus, both methods could be applied to the circuit for further noise reduction.

Acknowledgments

This research was supported by the ITRC program of ITTP by MSIP (ITTP-2015-H8501-15-1005) in Korea, NRF grant funded by the MSIP (2015R1A2A2A01004178), and Brain Korea 21 Plus Project in 2015.

References

- [1] L. Chen, M. Marek-Sadowska, F. Brewer, Buffer delay change in the presence of power and ground noise, *IEEE Trans. Very Large Scale Integr. Syst.* 11 (June (3)) (2003) 461–473.
- [2] T. Charania, A. Opal, M. Sachdev, Analysis and design of on-chip decoupling capacitors, *IEEE Trans. Very Large Scale Integr. Syst.* 21 (April (4)) (2013) 648–658.
- [3] J.P. Fishburn, Clock skew optimization, *IEEE Trans. Comput.* 39 (July (7)) (1990) 945–951.
- [4] Y. Wang, W.-S. Luk, X. Zeng, J. Tao, C. Yan, J. Tong, W. Cai, J. Ni, Timing yield driven clock skew scheduling considering non-Gaussian distributions of critical path delays, in: *Proceedings of IEEE/ACM Design Automation Conference*, 2008, pp. 223–226.
- [5] L. Benini, P. Vuillod, A. Bogliolo, G.D. Micheli, Clock skew optimization for peak current reduction, *J. VLSI Signal Process. Syst. Signal Image Video Technol.* 16 (June (2)) (1997) 117–130.
- [6] A. Vittal, H. Ha, F. Brewer, M. Marek-Sadowska, Clock skew optimization for ground bounce control, in: *Proceedings of IEEE/ACM International Conference on Computer-Aided Design*, 1996, pp. 395–399.
- [7] W.-C. Lam, C.-K. Koh, C.-W.A. Tsao, Power supply noise suppression via clock skew scheduling, in: *Proceedings of International Symposium on Quality Electronic Design*, 2002, pp. 355–360.
- [8] S.-H. Huang, C.-M. Chang, Y.-T. Nieh, Fast multi-domain clock skew scheduling for peak current reduction, in: *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2006, pp. 254–259.
- [9] A. Vijayakumar, V.C. Patil, S. Kundu, An efficient method for clock skew scheduling to reduce peak current, in: *29th International Conference on VLSI Design and 15th International Conference on Embedded Systems (VLSID)*, 2016, pp. 505–510.
- [10] Y.-T. Nieh, S.-H. Huang, S.-Y. Hsu, Opposite-phase clock tree for peak current reduction, *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* E90-A (December (12)) (2007) 2727–2735.
- [11] R. Samanta, G. Venkataraman, J. Hu, Clock buffer polarity assignment for power noise reduction, *IEEE Trans. Very Large Scale Integr. Syst.* 17 (June (6)) (2009) 770–780.
- [12] P.-Y. Chen, K.-H. Ho, T. Hwang, Skew-aware polarity assignment in clock tree, *ACM Trans. Des. Autom. Electron. Syst.* 14 (2) (2009). <http://dx.doi.org/10.1145/1497561.1497574>.
- [13] J. Lu, B. Taskin, Clock buffer polarity assignment with skew tuning, *ACM Trans. Des. Autom. Electron. Syst.* 16 (4) (2011). <http://dx.doi.org/10.1145/2003695.2003709>.
- [14] D. Joo, T. Kim, A fine-grained clock buffer polarity assignment for high-speed and low-power digital systems, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 33 (March (3)) (2014) 423–436.
- [15] H. Jang, D. Joo, T. Kim, Buffer sizing and polarity assignment in clock tree synthesis for power/ground noise minimization, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 30 (January (1)) (2011) 96–109.
- [16] J. Lu, B. Taskin, Clock buffer polarity assignment considering capacitive load, in: *Proceedings of International Symposium on Quality Electronic Design*, 2010, pp. 765–770.
- [17] M. Kang, T. Kim, Clock buffer polarity assignment considering the effect of delay variations, in: *Proceedings of International Symposium on Quality Electronic Design*, 2010.
- [18] J. Lu, B. Taskin, Clock tree synthesis with XOR gates for polarity assignment, in: *IEEE Computer Society Annual Symposium on VLSI*, 2010, pp. 17–22.
- [19] J. Lu, Y. Teng, B. Taskin, A reconfigurable clock polarity assignment flow for clock gated designs, *IEEE Trans. Very Large Scale Integr. Syst.* 20 (6) (2012) 1002–1011.
- [20] D. Joo, T. Kim, Clock buffer polarity assignment utilizing useful clock skews for power noise reduction, in: *Proceedings of IEEE Asia and South Pacific Design Automation Conference*, 2016, pp. 226–231.
- [21] I.M. Bomze, M. Budinich, P.M. Pardalos, M. Pelillo, The maximum clique problem, in: *Handbook of Combinatorial Optimization*, Springer, New York, 1999.
- [22] T.-Y. Kim, T. Kim, Clock tree synthesis for TSV-based 3D IC designs, *ACM Trans. Des. Autom. Electron. Syst.* 16 (October (4)) (2011) 48:1–48:21.
- [23] Open Cell Library v2009 07, Nangate Inc (<http://www.nangate.com/openlibrary>), 2009.

- [24] T. Achterberg, SCIP: solving constraint integer programs, *Math. Program. Comput.* 1 (July (1)) (2009) 1–41.
- [25] H. Chang, S. Sapatnekar, Statistical timing analysis under spatial correlations, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 24 (September (9)) (2005) 1467–1482.
- [26] M. Popovich, High performance power distribution networks with on-chip decoupling capacitors for nanoscale integrated circuits (Ph.D. thesis), University of Rochester, 2007.
- [27] M.S. Gupta, J.L. Oatley, R. Joseph, G.Y. Wei, D.M. Brooks, Understanding Voltage Variations in Chip Multiprocessors Using a Distributed Power-Delivery Network, 2007, pp. 1–6.
- [28] R. Jakushokas, M. Popovich, A.V. Mezhiba, S. Köse, E. Friedman, *Power Distribution Networks With On-Chip Decoupling Capacitors*, Springer Science & Business Media, New York, 2010.