Clock Buffer Polarity Assignment Utilizing Useful Clock Skews for Power Noise Reduction

Deokjin Joo and Taewhan Kim
Department of Electrical and Computer Engineering, Seoul National University, Seoul, Korea
{jdj,tkim}@snucad.snu.ac.kr

Abstract— Clock trees, which deliver the clock signal to every clock sink in the whole system, are one of the most active components on a chip 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 effectively extending it for PA under delay variation environment. Through experiments with ISPD'10 benchmark circuits, it is shown 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.

I. 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 of designs, 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 and actively in 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 deal with this problem, various *clock polarity assignment* (PA) techniques have been proposed [2, 3, 4, 5, 6]. 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 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.

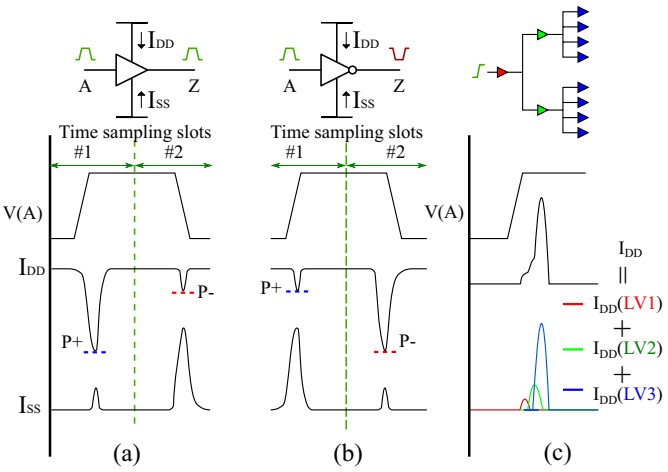


Fig. 1. The idea behind clock buffer polarity assignment. (a) A buffer draws larger I_{DD}/I_{SS} current at rising/falling edge of clock signal. (b) An inverter exhibits opposite behavior of (a). (c) Peak noise occurs around the time when the leaf buffers (blue color) switch.

There are plenty of research works [2, 3, 4, 5, 6, 7, 8, 9, 10, 11] that addressed the PA problem. One common feature of all previous works is that they are all *global clock skew bounded* [1] 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

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

978-1-4673-9569-4/16/$31.00 ©2016 IEEE

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.

The task of determining clock arrival times to every sink is referred to as *useful skew scheduling* [12]. Note that even though there are several works (e.g., [13, 14]) that have utilized the clock skew scheduling to minimize the peak noise, none of them have applied the clock PA combined with buffer sizing. In this work, we propose a comprehensive solution to the problem of clock PA integrated with buffer/inverter sizing to reduce clock switching noise. 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.

II. Motivational Example

Consider a small clock tree shown in Fig. 2(a). It has four clock sinks, each of which has its distinct driving clock buffer. 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 in Fig. 2(b). Given that each of the four clock buffers are initially a buffer instance of type $B1$, we can calculate the arrival time change Δt of its driven FF resulting from replacing it with an instance of other buffer/inverter type, as shown 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. 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 every setup and hold time constraint, 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.

III. Problem Formulation

Problem 1 (UsefulMin). *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*

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

under a set C of setup and hold time constraints:

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

where $LB(e_i, e_j)$ and $UB(e_i, e_j)$ can be $-\infty$ and ∞ respectively, if there is no time constraint between e_i and e_j. The term $noise(\phi(e_i), s)$ 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 $\phi(e_i) \in \{B \cup I\}$.

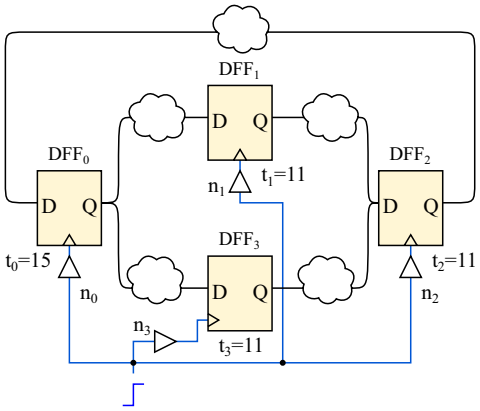
Note that P+ and P- values in the motivational example are short names for $noise(\phi, 1)$ and $noise(\phi, 2)$, respectively when $|S| = 2$, in which the peak current noise sampling slots 1 and 2 will be used as the high/low periods in the clock cycle. Increasing the number of time sampling slots can improve noise estimation. Moreover, both I_{DD} and I_{SS} should be sampled as slots since the objective is to minimize the worst current.

IV. The Proposed Algorithm

A. Proof of Intractability

The problem is intractable since the decision version of UsefulMin, decision-UsefulMin is NP-complete. Due to space limitations, we only sketch the proof:

Proof. We first define decision-UsefulMin. decision-UsefulMin is the same as Problem 1 except the objective: the problem is not minimizing the quantity of Eq. (1) but checking if it is possible to find a solution that makes Eq. (1) less than or equal to a given value z. decision-WaveMin is in



(a) Clock design with four clock buffers

-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

(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

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

(d) Eight feasible PA with sizing

Fig. 2. An illustration of clock buffer PA problem. (a) A small clock tree with 4 clock buffers and 4 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 8 are *feasible* assignments which cause no time constraint violation. (d) The 8 feasible clock PA of the design in (a) using the library in (c) that satisfies the time constraints in (b).

NP since the verification of solution can be done in polynomial time. On the other hand, it is known that the decision version of PA problem under bounded clock skew constraint is NP-complete [6]. Since useful clock skew is more general constraint, DECISION-USEFULMIN is "no easier than" DECISION-WAVEMIN and can be used to solve the bounded clock skew problems. Hence, DECISION-USEFULMIN is NP-hard. □

B. Approaches for Solving UsefulMin

B.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 since the LP relaxation yields 1/2 for each variable and only few of them are mapped to 0 or 1 [15]. The reason is that solving the ILP has underlying maximum clique search process, as will be shown later.

B.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 a precise analysis leads to find 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

$$max\{\sum_{q_i \in Q} w_i(s_j), j = 1, \cdots, |S|\}. \quad (2)$$

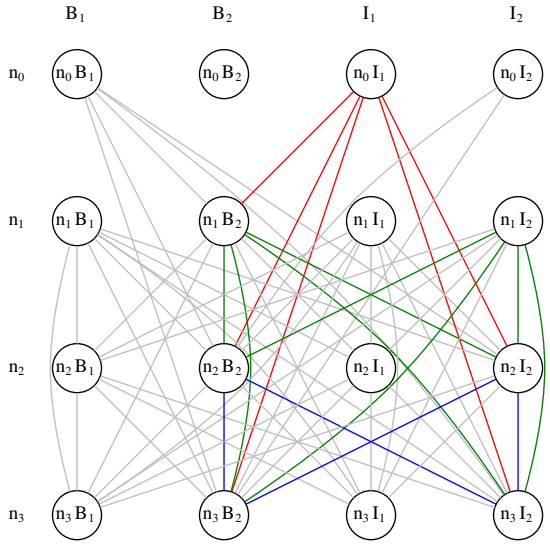


Fig. 3. Transformation of the problem instance in Fig. 2 into a search problem in a graph $G(V, E, W)$.

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 Eq.(2) can be translated to finding a

maximum clique in G with the minimum value of Eq.(2). 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 Eq.(2).

B.3 Scalable Algorithm for Clique Exploration

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 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 $O(\binom{|L|}{K}(|B| + |I|)^K)$. 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.

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 $O(\binom{n}{K}(|B| + |I|)^K)$ 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 speedup 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.

C. 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. In [5], Lu and Taskin 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. In [8], Kang and Kim 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 global clock skew of the whole clock tree.

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 doesn't 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.

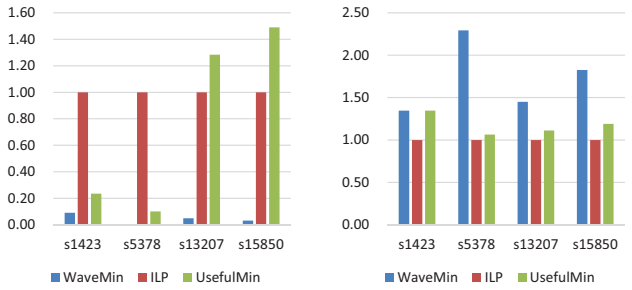
V. Experimental Results

A. Experimental Setup

The proposed algorithm UsefulMin was implemented in C++ language on a Linux machine with Intel i5-4670 CPU and 8GB RAM. Clock trees were generated for ISPD'10 high performance clock network synthesis contest benchmarks with the algorithm in [16], using Nangate 45nm Open Cell Library [17] 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]

TABLE I
THE COMPARISON OF OUR USEFUL SKEW CONSTRAINED PA (USEFULMIN) AND SKEW BOUND CONSTRAINED ASSIGNMENT (WAVEMIN [6]). # CONSTR. AND # LVS COLUMNS INDICATE THE NUMBER OF USEFUL CLOCK SKEW CONSTRAINT AND THE NUMBER OF LEAF BUFFERING ELEMENTS IN THE CLOCK TREE, RESPECTIVELY. *Area* INDICATES THE TOTAL AREA OF LEAF BUFFERS/INVERTERS ALLOCATED BY THE PA.

ISPD 2010 Ckt	# Cnstr.	# Lvs	Base		Wavemin [6]			UsefulMin			Improvement	
			Noise (mA)	Area (μm^2)	Noise (mA)	Area (μm^2)	Run time (s)	Noise (mA)	Area (μm^2)	Run time (s)	Noise (%)	Area (%)
01	11070	221	235.1	22.38	155.3	21.24	56.21	122.1	13.73	2547.70	21.38	35.38
02	22490	454	433.9	45.97	281.9	43.83	291.00	242.4	29.62	24229.49	14.01	32.42
03	12000	113	106.1	11.44	45.58	6.00	12.34	42.56	6.49	25.26	6.63	-8.3
04	18450	116	115.3	11.75	52.85	8.58	44.46	48.21	6.60	145.36	8.78	23.07
05	10160	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	19150	131	125.4	13.26	73.26	13.20	34.73	67.13	8.51	127.49	8.37	35.55
08	11340	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**



(a) Execution time (b) Peak current

Fig. 4. Normalized comparison of our UsefulMin, conventional WaveMin, 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 (0.18s vs. 0.25s for s15850). However, as circuits become larger, ILP overtakes (3.9s vs. 0.8s in s5378, where s5378 is the largest of the 4 benchmark circuits).

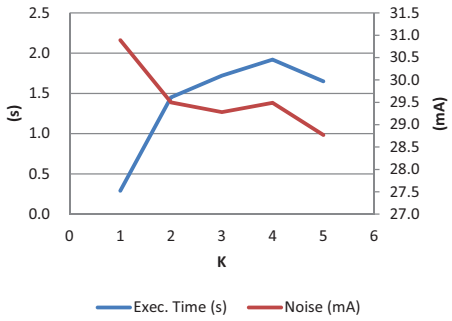


Fig. 5. The effect of parameter K on the optimization of circuit 05 in ISPD'10 by UsefulMin algorithm.

ps range for upper bounds and [-90, -60] 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 [6].

B. Assessing the Performance of UsefulMin over Bounded Skew Algorithm [6]

The simulation results are summarized in Table I. 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 $K = 5$. Overall, our algorithm reduces the peak noise by 49.1% and 10.9% further on average over that of no PA and the conventional PA, respectively. On average UsefulMin reduces the power by 4.9% over that of WaveMin. The minimum and maximum average power improvement are -2.5% in circuit 03 and 12.5% in circuit 03, respectively, which reveal similar trend of that of area improvement. Fig. 4 compares our UsefulMin algorithm with optimal ILP formulation. For the ILP solver, SCIP [18] was used. The two curves in Fig. 5 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 into account the execution time.

C. Assessing the Performance of Delay Variation Aware Polarity Assignment

Since both skew tuning [5] and pairwise [8] methods are incapable of buffer sizing, we defined $B = \{BUF\_X8\}$ and $I = \{INV\_X4\}$. INV_X4 was chosen so, as it had 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. In the experiments, we assumed that buffer/inverter and interconnect delays are spatially correlated normally

TABLE II

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 circuits 01 and 02 in ISPD'10, 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)			Yield (%)			
		Tuning [5]	Pairwise [8]	Ours	Base	Tuning [5]	Pairwise [8]	Ours
01	0.83	120.10	123.68	92.93	84.7	76.4	73.1	81.1
02	0.39	222.75	230.50	195.54	40.7	28.6	27.2	39.4
03	0.98	50.40	51.10	51.47	99.6	94.9	93.8	98.6
04	0.98	54.79	55.43	58.77	99.7	94	94.4	98.9
05	0.98	27.93	27.78	27.92	99.99	98.8	98.6	99.7
06	0.98	39.54	39.65	40.12	100	99.4	99.7	100
07	0.98	59.76	62.09	66.45	99.90	96.3	96.5	99.1
08	0.98	47.59	47.45	45.40	100	99.7	99.8	99.8

distributed random variables. Spatial correlations were modelled using the grid model proposed in [19]. 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 [19]. Given (correlated) normal distributions d_1, d_2, d_3, ..., $max(d_1, d_2, d_3, ...)$ operation computes approximated normal distribution of the maximum value.

Table II summarizes the results of variation aware clock PA. 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. Compared to the other methods which only have indirect control over yield, in all circuits, our algorithm maintains comparable noise while keeping the design yield constraint γ.

VI. 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, our proposed approach would be useful in mitigating the clock noise, which otherwise the conventional PA approaches could rarely achieve.

VII. Acknowledgements

This research was supported by the ITRC program of ITTP by MSIP (IITP-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, and F. Brewer. Buffer delay change in the presence of power and ground noise. *IEEE TVLSI*, Vol. 11, No. 3, pp. 461–473, June 2003.

[2] Y.-T. Nieh, S.-H. Huang, and S.-Y. Hsu. Opposite-phase clock tree for peak current reduction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science*, Vol. E90-A, No. 12, pp. 2727–2735, Dec. 2007.

[3] R. Samanta, G. Venkataraman, and J. Hu. Clock buffer polarity assignment for power noise reduction. *IEEE TVLSI*, Vol. 17, No. 6, pp. 770–780, June 2009.

[4] P.-Y. Chen, K.-H. Ho, and T. Hwang. Skew-aware polarity assignment in clock tree. *ACM TODAES*, Apr. 2009.

[5] J. Lu and B. Taskin. Clock buffer polarity assignment with skew tuning. *ACM TODAES*, Oct. 2011.

[6] D. Joo and T. Kim. A fine-grained clock buffer polarity assignment for high-speed and low-power digital systems. *IEEE TCAD*, Vol. 33, No. 3, pp. 423–436, Mar. 2014.

[7] Y. Ryu and T. Kim. Clock buffer polarity assignment combined with clock tree generation for power/ground noise minimization. In *Proc. IEEE/ACM ICCAD*, Nov. 2008.

[8] M. Kang and T. Kim. Clock buffer polarity assignment considering the effect of delay variations. In *Proc. ISQED*, 2010.

[9] H. Jang, D. Joo, and T. Kim. Buffer sizing and polarity assignment in clock tree synthesis for power/ground noise minimization. *IEEE TCAD*, Vol. 30, No. 1, pp. 96–109, Jan. 2011.

[10] J. Lu and B. Taskin. Clock tree synthesis with XOR gates for polarity assignment. In *Proceedings of the 2010 IEEE Annual Symposium on VLSI*, pages 17–22, July 2010.

[11] J. Lu, Y. Teng, and B. Taskin. A reconfigurable clock polarity assignment flow for clock gated designs. *IEEE TVLSI*, Vol. 20, No. 6, pp. 1002–1011, 2012.

[12] J. P. Fishburn. Clock skew optimization. *IEEE Transactions on Computers*, Vol. 39, No. 7, pp. 945–951, July 1990.

[13] K. Wang, Y. Ran, H. Jiang, and M. Marek-Sadowska. General skew constrained clock network sizing based on sequential linear programming. *IEEE TCAD*, May 2005.

[14] W.-C. Lam, C.-K. Koh, and C.-W. A. Tsao. Power supply noise suppression via clock skew scheduling. In *Proc. ISQED*, pages 355–360, Mar. 2002.

[15] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. Pelillo. The maximum clique problem. In *Handbook of Combinatorial Optimization*. Springer, 1999.

[16] T.-Y. Kim and T. Kim. Clock tree synthesis for TSV-based 3D IC designs. *ACM TODAES*, Vol. 16, No. 4, pp. 48:1–48:21, Oct. 2011.

[17] Open cell library v2009 07, Nangate Inc. `http://www.nangate.com/openlibrary`, 2009.

[18] T. Achterberg. SCIP: Solving constraint integer programs. *Mathematical Programming Computation*, Vol. 1, No. 1, pp. 1–41, July 2009.

[19] H. Chang and S. Sapatnekar. Statistical timing analysis under spatial correlations. *IEEE TCAD*, Sept. 2005.