A Fine-Grained Clock Buffer Polarity Assignment for High-Speed and Low-Power Digital Systems

Deokjin Joo, *Student Member, IEEE,* and Taewhan Kim, *Senior Member, IEEE*

***Abstract*—The clock buffer polarity assignment is one of the effective design schemes to mitigate the power/ground noise caused by the clock signal propagation in high-speed digital systems. This paper overcomes a set of fundamental limitations of the conventional clock buffer polarity assignment methods, which are: 1) the unawareness of the signal delay (i.e., arrival time) differences to the leaf clock buffering elements; 2) the ignorance of the effect of the current fluctuation of nonleaf clock buffering elements on the total peak current waveform; and 3) the inability of supporting low-power digital designs with multiple (dynamically operating) power modes. Clearly, not addressing 1 and 2 in the polarity assignment may cause a severe inaccuracy on the peak current estimation, which results in unnecessarily high peak current. Moreover, without tackling 3, designs may suffer from clock skew violation in some of the power modes, affecting circuit speed or reliability. To overcome the limitations, we propose a completely new fine-grained approach to the clock buffer polarity assignment combined with buffer sizing, formulating the problem into a multiobjective shortest path problem and solving it effectively for designs with a single power mode, while exploiting the flexibility of our multiobjective shortest path formulation for designs with multiple power modes. Through experiments using benchmark circuits, it is shown that the proposed approach is able to produce designs with 17% lower peak current and 20% lower power noise on average, compared with the results produced by the best ever known method.**

***Index Terms*—Adjustable delay buffer, buffer sizing, clock skew, clock tree synthesis, multiple power modes, polarity assignment, power/ground noise.**

I. Introduction

As THE CMOS process technology scales down, it becomes possible to use much lower supply voltages in very large scale integration design. The use of lowered supply voltage then enables reducing the power consumption in the circuit. However, the use of lower supply voltage causes the circuit to be more susceptible to the power and ground noise, i.e., voltage fluctuation in the power and ground rails. This noise also adversely affects circuit performance such as the delay of switching signal [2], [3]. The major sources of the voltage fluctuation are attributed to the input and output drivers and the internal logic circuitry, especially those that switch near either rising or falling edge of clock signal [4]. In a synchronous high-speed circuit, the buffered clock tree consumes a considerable amount of power since its clock signal is one of the most actively switching sources in the circuit. It is reported that the amount of clock power consumed by a clock distribution network with clocked loads typically accounts for one third to one half of the total chip power dissipation [5]. This implies that the clock tree is one of the major sources of power and ground noise.

Ideally, the clock signal should reach all sequential elements at the same time from the clock source. However, in practice, there exists some timing difference between the clock signal paths from the clock source to the sequential elements due to variations on path lengths and buffer characteristics on the paths. The largest difference among the arrival times of clock signal is called clock skew, and achieving zero clock skew is a practically very difficult task. A doable solution is to limit the clock skew in a certain bound that can tolerate all variations caused by the clock skew. Furthermore, it should be noted that as the applications run on a digital system are complex and diverse, designing a system with multiple (dynamically operating) power modes, in which the voltage applied to some design module varies as the power mode changes, is regarded as an effective strategy to save power consumption. For designs with multiple power modes, optimizing the structure of a clock tree to meet the clock skew constraint for every power mode is an important task. However, such an optimization of clock tree may cause a high power and ground noise in some power mode if the effect of clock tree optimization on the noise is not carefully taken into account.

Extensive research works on clock tree optimization, such as clock routing, clock buffer insertion/sizing, and wire sizing, have been performed to minimize clock skew for designs with a single power mode [6]–[12]. In addition, as the importance of maintaining the clock skew for designs with multiple power modes has been aware, recently a number of post-silicon tuning methods, particularly replacing some of normal buffers in the clock tree with adjustable delay buffers (ADBs), have been developed to cope with the clock skew problem [13]–[17], in which they resolve the clock skew violation caused by the dynamic change of power mode by properly replacing some buffers with ADBs. Note that some works have also utilized ADBs in designs of a single power mode to

Manuscript received June 17, 2013; revised September 11, 2013; accepted October 13, 2013. Date of current version February 14, 2014. This work was supported in part by the Basic Science Research Program under NRF Grant 2011-0029805 in Korea, in part by the CISS of the Global Frontier Project by MSIP (CISS 2011-0031863) in Korea, in part by the ITRC Program of NIPA by MSIP (NIPA-2013-H0301-13-1011) in Korea, and in part by Samsung Electronics Company. A preliminary version of this paper was presented in [1]. This paper was recommended by Associate Editor P. Saxena.

The authors are with the School of Electrical and Computer Engineering, Seoul National University, Seoul, Korea (e-mail: jdj@snucad.snu.ac.kr; tkim@snucad.snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2013.2288698

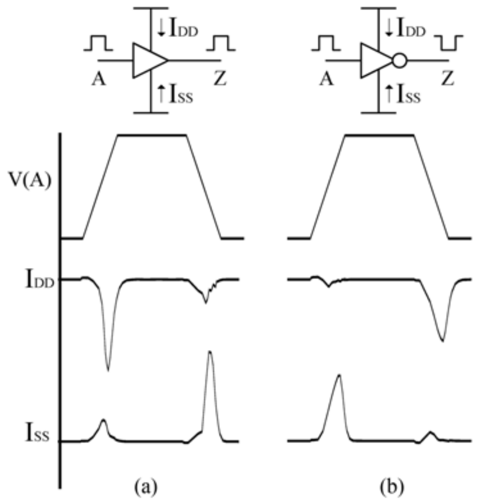


Fig. 1. Idea behind buffer polarity assignment. (a) Buffers exhibit high $I_{\mathrm{DD}}/I_{\mathrm{SS}}$ current at rising/falling edge of clock signal. (b) while inverters emit high $I_{\mathrm{DD}}/I_{\mathrm{SS}}$ current at falling/rising edge.

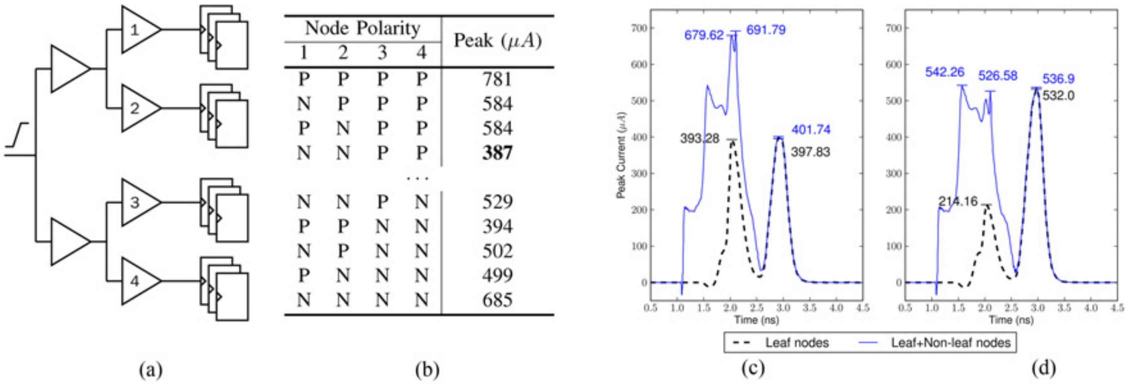
minimize the impact of process, environmental, or statistical variation on clock skew [18]–[21]. However, all the works in [6]–[17] do not address the power/ground noise problem at all.

It has been known that selectively assigning (positive or negative) polarities to (initial) clock buffering elements by properly replacing some of the buffering elements with inverters is an effective way of reducing the power/ground noise.[1] Fig. 1 illustrates the basic idea behind the polarity assignment. A buffer is a chain of unequally sized two inverters and exhibits current noise as shown in Fig. 1(a): at the rising edge of clock signal, the buffer charges, drawing a high I_{DD} current while drawing a low I_{SS} current. For inverters, the opposite case happens as shown in Fig. 1(b). Thus, by mixing buffers and inverters in the buffered clock tree, the designer is able to disperse the current noise from/to V_{DD}/Gnd at rising/falling edge of clock signal. By observing the current waveforms in Fig. 1, several techniques of buffer polarity assignment have been proposed [22]–[29].

Nieh *et al.* [22] first proposed to assign positive polarity onto a half of clock buffers and negative polarity onto the rest half of the clock buffers. Thus, they equally divided the whole clock tree into two subtrees and replaced the buffering element at the root of a subtree with an inverter, so that when the clock signal switches from 0 to 1 (or 1 to 0) all buffers on one subtree charge (or discharge) current from V_{DD} (or to Gnd) while all buffers on the other subtree discharge (or charge) current to Gnd (or from V_{DD}). Note that the buffering elements directly connected to flip-flops (FFs) are called leaf nodes or sinks and the other buffering elements non-leaf nodes or non-sinks. Thus, the FFs connected to a leaf node assigned with negative polarity should be replaced with negative-edge triggered FFs. Even though this simple modification in [22] can reduce the total peak current over the chip up to the limit, it is not able to effectively reduce the power/ground noise in local regions. To overcome this limitation, Samanta *et al.* [23] used the physical placement information of the buffering elements in determining buffers and inverters so that for local regions, roughly half of the buffering elements are assigned with positive polarity and the other half with negative polarity. Although this paper is able to reduce the power/ground noise greatly, sometimes it is likely to cause a long clock skew because the effect of the different delays of inverters and buffers on the clock skew have not been taken into account. Chen *et al.* [24] observed that the peak current occurs at the time when the clock signal arrives at the buffering elements (i.e., leaves) that are directly incident to FFs, as validated by SPICE simulation. Thus, they proposed a method of assigning polarities to the leaves, using the physical placement information of the leaves, with the objective of minimizing the power/ground noise while satisfying the clock skew constraint. In addition, the approach by Ryu and Kim [25] placed more weight on the power/ground noise minimization than the clock tree embedding, thus performing polarity assignment followed by clock tree construction. However, this approach required wire overhead, which is about 5%. Kang and Kim [26] considered the delay variations in the polarity assignment. They performed polarity assignment that minimizes the power/ground noise, while meeting the skew yield constraint caused by the clock skew variation. On the other hand, Jang *et al.* [27] proposed an integrated approach to the polarity assignment combined with buffer sizing to further explore the design space. Lu and Taskin [28] attempted to assign polarity to non-leaf buffering elements, as well as leaf elements by which the peak current noise was reduced by 5.5% further by reducing the noise from the non-leaf elements at the expense of the increase of clock skew. Later, in [29], they proposed to perform skew tuning on the polarity assigned clock trees to reduce the clock skew in the worst corner. Recent research shows that polarities may be adjusted dynamically by using XOR gates and double edge triggered flip-flops, which makes clock gating mode-specific noise reduction possible [30], [31]. The critical flaws of all the previous polarity assignment approaches [22]–[25], [27] are: 1) the unawareness of the signal delay (i.e., arrival time) differences to the leaf nodes; 2) the ignorance of the effect of non-leaf nodes' current fluctuations on the total peak current waveform; and 3) the inability of supporting designs of multiple power modes. Clearly, not addressing 1 and 2 in polarity assignment may cause a severely inaccurate peak current (or peak power/ground noise) estimation. Moreover, multipower mode unaware approaches may cause clock skew violation in some power modes, which would affect circuit speed or reliability. By addressing the limitations, we propose a completely new solution to the problem of clock buffer polarity assignment combined with buffer sizing, employing a fine-grained noise estimation technique, rather than using the peak current values only at the four time sampling points of (V_{DD}, rising), (V_{DD}, falling), (Gnd, rising), and (Gnd, falling), as adopted by the previous works. The contributions of the work are summarized as follows.

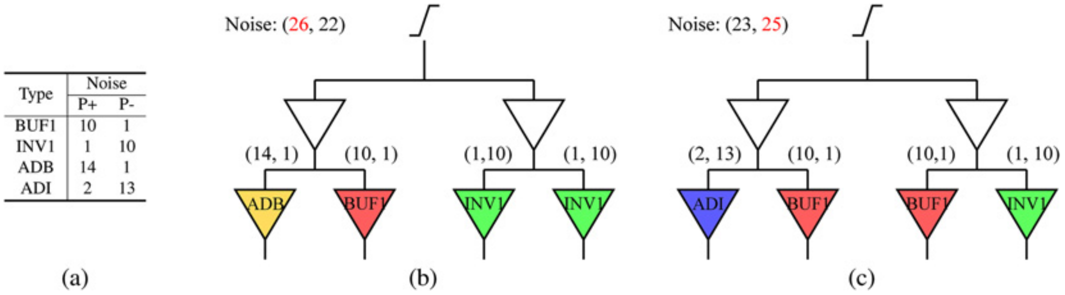
1) We propose a fine-grained clock buffer polarity assignment algorithm that overcomes the limitations of the previous works: a) the unawareness of the signal delay

[1] A buffering element is said to be assigned with a positive polarity or a negative polarity if its output switches in the same direction as or in the opposite direction to that of the clock source, respectively.



Node Polarity				Peak (μA)
1	2	3	4	
P	P	P	P	781
N	P	P	P	584
P	N	P	P	584
N	N	P	P	**387**
		...		
N	N	P	N	529
P	P	N	N	394
N	P	N	N	502
P	N	N	N	499
N	N	N	N	685

Fig. 2. (a) Simple clock tree with four leaf nodes. (b) Expected peak current value by the leaf nodes for each of the possible polarity assignments. N means a negative polarity and P means a positive polarity. The fourth assignment (N, N, P, P) produces the lowest value of total peak current of 387 μA. (c) Current waveforms by non-leaf nodes' noise unaware optimal polarity assignment [= (N, N, P, P) in (b)] to leaf nodes. Dark dotted line is the current waveform from leaf nodes only while blue solid line shows the total current from all clock nodes. (d) Current waveforms resulting from non-leaf nodes' noise aware optimal polarity assignment [= (N, N, P, N) in (b)] to leaf nodes.



Type	Noise	
	P+	P-
BUF1	10	1
INV1	1	10
ADB	14	1
ADI	2	13

Fig. 3. (a) Profile of a buffer BUF1, an inverter INV1, an adjustable delay buffer ADB, and an adjustable delay inverter ADI. P+ and P- denote the noise emitted at the rising and falling edges of clock signal. (b) Optimal polarity assignment using BUF1, INV1, and ADB for a design with two power modes M_1 and M_2. Two numbers (26, 22) in parenthesis at the top indicate the total peak noise corresponding to P+ and P- for power modes M_1 and M_2. Thus, the peak noise is 26. (c) Optimal polarity assignment using ADI together with BUF1, INV1, and ADB. The peak noise is reduced from 26 to 25.

differences to the leaf nodes, and b) the ignorance of the effect of non-leaf nodes' current fluctuations on the peak current waveform. We overcome a) and b) by formulating the polarity assignment problem into a multiobjective shortest path problem and solving it effectively.

2) We propose an extended solution to the problem of polarity assignment 3 for designs with multiple power modes. We propose a systematic solution that effectively exploits the flexibility of our multiobjective shortest path formulation used for designs with a single power mode. The proposed solution meets the clock skew bound in all power modes by doing simultaneous polarity assignment and buffer sizing while considering all power modes. For the designs with larger clock skews where the clock skew bound cannot be met with polarity assignment and buffer sizing, our proposed solution may be applied to the clock trees that have preplaced ADBs, which may be obtained with existing ADB embedding techniques [13]–[17]. In addition, we propose to use a new cell component, which we call adjustable delay inverters (ADIs) combined with the use of ADBs to maximally reduce the peak noise.

3) We include diverse practical analyses such as effect of buffer sizing on polarity assignment, characterizing buffering elements as well as theoretical analyses such as proof of NP-completeness, time complexity of the algorithms, and extensive experimental data to support the feasibility and effectiveness of the proposed approach. The applicability of this paper is extended to include the low power designs of multiple power modes, as well as to the diverse high-speed designs of a single power mode.

II. Observations

Since the leaf buffering elements are the major contributor to the (total) peak current as illustrated by [24], our work also focuses on the polarity assignment on leaf buffering elements. This section includes a number of important observations we have made regarding how the previous works on polarity assignment lack the accuracy in estimating peak current, how the previous works of ADB allocation to support designs with multiple power modes lose the opportunity of reducing peak current, and what factors we should focus on or ignore.

Observation 1 (the effect of current fluctuation by non-leaf buffers): Let us consider the problem of assigning polarity to the four leaf nodes on the clock tree in Fig. 2(a). All possible combinations of polarity assignment by replacing each node with buffer or inverter and the corresponding value of total peak current obtained by summing the peak current values of the nodes are summarized in the table in Fig. 2(b), where P and N indicate positive and negative polarities, respectively.

From the table, we can see that the fourth assignment (N, N, P, P) produces the lowest value of total peak current, which is 387 μA. The dark dotted curve in Fig. 2(c) shows the accumulated current waveform of the leaf nodes for the polarity assignment (N, N, P, P). On the other hand, the blue solid curve in Fig. 2(d) shows the accumulated current waveform of all nodes, including the two non-leaf buffers, from which we can see that the actual value of total peak current is unbalanced, i.e., skewed to the left (at time = 2.2 ps), resulting in the peak current of 691.79 μA. However, the dark dotted curve in Fig. 2(d) shows the current waveform of the leaf nodes when the polarity assignment is (N, N, P, N); thus, the peak is skewed to the right. The blue solid curve in Fig. 2(d) which shows the resulting waveform of all nodes, however, has much reduced peak current, which is around 542 μA. This observation implies that the current fluctuation by non-leaf nodes should be taken into account during the process of polarity assignment of leaf nodes.

Observation 2 (the effect of clock signal delay difference): Another observation from the current waveforms in Fig. 2(d) indicates that by knowing that some leaf nodes may switch at different times due to unequal clock signal propagation delays, the current fluctuation by the non-leaf nodes contributes differently to the (accumulated) current waveforms at the time when the leaf nodes switch. Thus, any time instance in a certain time interval [e.g., time in [1.0, 4.0] in Fig. 2(d)] can be a time sampling candidate at which peak current may occur.

Observation 3 (the unawareness of peak noise in designs with multiple power modes): For the designs with multiple power modes, it is known that ADBs are very useful to meet the clock skew constraint for every power mode. For example, Fig. 3(b) shows an optimal polarity assignment using a buffer BUF1, an inverter INV1, and an adjustable delay buffer ADB in Fig. 3(a) for a design of two power modes. The peak noise is 26. However, if we include another type of delay adjustable element, which we call ADI as shown in Fig. 3(a), a better polarity assignment can be produced, as shown in Fig. 3(c). This observation implies that by carefully performing polarity assignment using ADBs and normal buffers/inverter together with ADIs for designs with multiple power modes, it is possible to reduce the peak current further while satisfying the clock skew constraint for every power mode.

Fig. 4 shows a capacitor bank based ADI we have implemented. The implementation of ADI is almost identical to that of ADB in [16], except the polarity of ADI is opposite to that of ADB. In this ADI implementation, the capacitor bank acts as a variable capacitor between the two inverters, adjusting delay. The number of switched capacitors in the capacitor banks 1 and 2 may be traded off for better control of the ADI delay and area overhead.

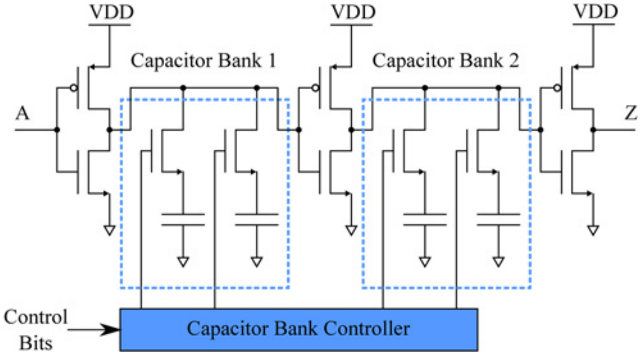


Fig. 4. Our proposed capacitor bank based implementation of ADI. The capacitor banks contain switched capacitors that are dynamically controllable. The number of capacitors in the two banks is a design parameter that controls the granularity of the discrete delay steps and the delay range of the ADI.

TABLE I

Impact of Buffer Sizing and Polarity Assignment to 15 Siblings on a Buffer. Data Was Obtained by Gradually Replacing the Siblings From Buffers to Inverters: Columns T_D, Peak, and Slew Are the Propagation Delay, Peak of the Noise Current, and Slew Rate of the Buffer, Respectively. The Change Has Little Influence on T_D and Slew, But It Has a Direct Influence on the Peak

# Invs	# Bufs	T_D (ps)		Peak (μA)		Slew (ps)	
		rise	fall	I_{DD}	I_{SS}	rise	fall
0	16	31.23	**35.19**	49.26	42.85	29.12	40.63
1	15	**27.34**	38.7	71.49	51.81	29.26	42.15
2	14	28.31	40	90.61	63.04	32.25	43.24
3	13	29.48	40.29	106.9	75.72	34.53	44.19
4	12	28.54	40.17	215.6	88.11	37.58	45.23
5	11	28.56	40.43	132.9	100.6	40.91	46.46
6	10	31.58	41.7	143.4	112.4	44.54	48.03
7	9	30.23	43.59	313.1	157.6	46.68	50.38
8	8	31.42	45.93	160.8	133.8	49.42	52.65
9	7	33.12	45.96	170.3	143.5	52.82	53.36
10	6	**34.27**	45.96	197.1	152.3	54.08	54.53
11	5	30.77	46.07	220.1	160.7	57.99	55.92
12	4	32.57	46.44	239.8	168.4	60.15	57.56
13	3	34.8	47.09	256.7	175.7	62.89	59.08
14	2	32.15	**47.27**	271.4	182.4	67.35	60.5
15	1	31.78	47.17	284.2	188.7	69.88	61.89

Observation 4 (the effect of polarity assignment and buffer sizing on siblings): Table I shows an HSPICE simulation result on a clock tree with 16 leaf buffering elements. The 16 buffering elements are driven by a parent buffer sized as BUF_X16 (R_{out} = 397.6 Ω). The first column indicates the number of buffers, all of which were initially BUF_X4 (C_{in} = 1 fF), replaced with inverters of INV_X8 (C_{in} = 2.2 fF). For example, #Invs = 3 means that three out of 16 buffers are replaced with inverters and the remaining 13 buffers are left unchanged. By replacing BUF_X4 with INV_X8, both the polarity and buffer size are affected, making the effect of the replacement more observable. We measured the propagation delay (T_D in the table) and clock slew[2] of a buffer and the peak current at its power rail, while its siblings are replaced. It is observed from the table that the clock signal delay at the rising time increases by up to 7.5 ps and the slew degrades by up to 40.76 ps, both of which occur between two extreme (unrealistic) polarity assignments (i.e., those in the first and last rows in the table). From the practical point of view, we can see from the table that the slew changed by a local update in the number of #Invs and #Bufs is less than 4 ps. Furthermore, the slew degradation is less of a concern in real clock trees

[2] We used 20%–80% rise time or 80%–20% fall time for slew.

where the parent buffers have better driving strength by having more options for buffer/inverter sizing. Hence, it is acceptable to assume that the result of polarity assignment or buffer sizing to a buffering element has a negligible effect on the delay and clock slew of its siblings, but the peak current varies significantly. This observation indicates that when we consider a polarity assignment or a buffer sizing to a leaf buffering element, we can ignore the delay and slew change of its siblings, and focus only on minimizing peak current.

III. Problem Formulation

We formally describe the polarity assignment problem combined with buffer sizing as follows.

Problem 1 (WaveMin): (Polarity assignment combined with buffer sizing for peak current minimization) Given a buffer library B, an inverter library I, a set L of leaf buffering elements (i.e., sinks), a set S of time samplings, and clock skew constraint κ, find a mapping function $\phi : L \mapsto \{B \cup I\}$ that minimizes the quantity of

$$\max_{s\in S}\left\{\sum_{e_i\in L} noise(\phi(e_i), s)\right\} \qquad (1)$$
$$\text{s.t } t_{skew}(\phi) \le \kappa$$

where $t_{skew}(\phi)$ is the clock skew induced by mapping ϕ and $noise(\phi(e_i), s)$ is the value of peak current estimation at a time sampling point s caused by the switching of node e_i when it is assigned with $\phi(e_i) \in \{B \cup I\}$.

Note that $|B| = |I| = 1$ corresponds to polarity assignment without buffer sizing. S represents the set of not only discretely sampled times of interest such as the rising and falling edges of clock signal but also power lines of interest. For example, S may contain four times: when V_{DD} and Gnd are on the rising edge of clock tree; when V_{DD} and Gnd on the falling edge. As S includes more (meaningful) time sampling points, the peak current estimation would be more accurate.

In the following, we show that WaveMin is NP-complete by reducing the decision version of PeakMin [27], which is NP-complete, to Decision-WaveMin problem which is the decision version of WaveMin.

Problem 2 (Decision-WaveMin): For a WaveMin instance with (L, B, I, S, κ) and a constant c, is there a mapping ϕ such that the value of (2) is less than or equal to c?

Problem 3 (Decision-PeakMin): For a set L of leaf buffering elements, a buffer library B, an inverter library I, clock skew bound κ, and a constant c, find a mapping function $\phi : L \mapsto \{B \cup I\}$ such that c is greater than the quantity of

$$\max\left\{\sum_{\phi(e_i)\in B} peak(\phi(e_i)),\ \sum_{\phi(e_i)\in I} peak(\phi(e_i))\right\} \qquad (2)$$
$$\text{s.t. } t_{skew}(\phi) \le \kappa$$

where $t_{skew}(\phi)$ is the clock skew induced by mapping ϕ and $peak(\phi(e_i))$ indicates the amount of peak current on $\phi(e_i)$ over time period $[0, \infty)$.

Theorem 1: Decision-PeakMin is NP-complete [27].

Theorem 2: Decision-WaveMin is NP-complete.

Proof: It is easy to see that Decision-WaveMin is in NP, because a nondeterministic algorithm needs only to guess a mapping of buffering elements in L to a buffer in B or an inverter in I, and check in polynomial time if the value of (2) is less than or equal to c or not.

We transform Decision-PeakMin to Decision-WaveMin: by passing all parameters of Decision-PeakMin to Decision-WaveMin with $|S| = 2$ composed of V_{DD} rail at the rising and falling edges of the clock, every instance of Decision-PeakMin can be exactly mapped to an instance of Decision-WaveMin, in which the summation term for buffers in (2) is V_{DD} rail at the rising edge of the clock, which corresponds to the summation term in (2) in a time sampling point in S, while the summation term for inverters in (2) is V_{DD} rail at the falling edge, which corresponds to the summation term in (2) in the other time sampling point in S. ■

Since WaveMin is a generalized version of PeakMin [27], as illustrated in the proof of Theorem 2, we will borrow a number of key concepts from the work in [27] such as feasible time interval to satisfy the skew constraint κ and local zones by which the circuit is divided into tiles to tackle the noise minimization tile by the tile basis.

IV. Background and Preprocessing

Since the work of PeakMin [27] has provided a useful basis, we recapitulate the terminologies it used and review its approach using an example in the first subsection. Then, in the second subsection, a preprocessing step, which is to extract noise data as well as time sampling points for all combinations of buffers/inverters in $B \cup I$ and sinks in L, is described.

A. Review of PeakMin [27]

Fig. 5 shows an example of clock tree with four leaf nodes (i.e., sinks) e_1, e_2, e_3, and e_4, in which we assume the leaf nodes are all initially assigned to (i.e., sized by) BUF_X2 in buffer library B, resulting in the arrival times[3] of 69, 70, 71, and 70 to the FFs driven by the leaf nodes. We assume the clock skew constraint $\kappa = 5$. Furthermore, we assume buffer library $B = \{$BUF_X1, BUF_X2$\}$ and inverter library $I = \{$INV_X1, INV_X2$\}$. Table II details B and I used in the presentation.

For the clock tree in Fig. 5 with libraries B and I in Table II, PeakMin performs the polarity assignment in three steps. (Step 1) The first step is, for each sink, to collect all (distinct) arrival times of the sink resulting from the trials of assigning the sink to all elements in B and I. For example, in Fig. 5 and Table II, the arrival time of sink e_2 is 70 when BUF_X2, whose delay is 19, is assigned to e_2. If the other three types, i.e., BUF_X1, INV_X1, and INV_X2, whose delays are, respectively, 24, 21, and 17, are assigned to e_2, the arrival times will be 75, 72, and 68. Thus, the collected arrival times are $\{68, 70, 72, 75\}$. This process is applied to every sink and the set of arrival times is extracted. Then, the sets are merged into one. The numbers arranged in the bottom of grids

[3] We will simply say the arrival times of sinks unless it causes confusion.

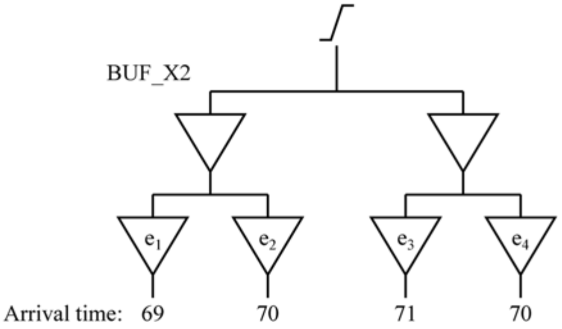


Fig. 5. Example of clock tree with four leaf nodes e_1, e_2, e_3, and e_4. It is assumed that all leaf nodes are initially sized by BUF_X2 in buffer library B, generating arrival times of 69, 70, 71, and 70 to the FFs driven by the leaf nodes.

TABLE II

CHARACTERIZATION OF $B = \{\text{BUF\_X1, BUF\_X2}\}$ AND $I = \{\text{INV\_X1, INV\_X2}\}$. T_D REPRESENTS THE SIGNAL PROPAGATION DELAY, P+ AND P- INDICATE THE VALUES OF THE PEAK I_{DD} AT THE RISING AND FALLING EDGES OF THE INPUT, RESPECTIVELY. (FOR BREVITY, WE OMIT HERE THE VALUES OF P+ AND P- OF I_{SS})

Type	T_D	P+	P-
BUF_X1	24	130	13
BUF_X2	19	255	44
INV_X1	21	13	130
INV_X2	17	44	255

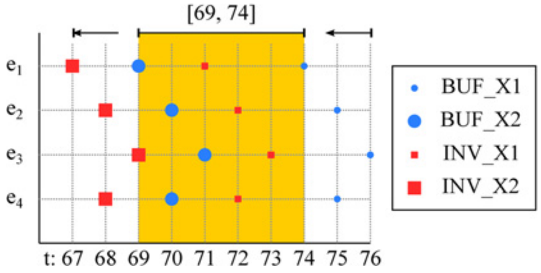


Fig. 6. Illustration of intervals of arrival times for the example in Fig. 5 and Table II. Each dot in the grid represents a buffer or an inverter. For example, the large red dot located at position (68, e_2) indicates that e_2 has arrival time of 68 when INV_X2 is assigned to it. Each arrival time and the clock skew constraint ($\kappa = 5$) defines an interval. For example, the yellow area indicates interval [74-κ, 74] = [69, 74] of arrival time $t = 74$.

in Fig. 6 show the distribution of arrival times. (Step 2) The second step is to convert the arrival times into intervals. For an arrival time t, its time interval is defined as $[t - \kappa, t]$. For example, the yellow area in Fig. 6 shows time interval $(= [74 - \kappa, 74] = [69, 74])$ of $t = 74$. Since interval [69, 74] contains at least one buffer or inverter in each row of sinks e_1, e_2, e_3, and e_4 in Fig. 6, which is called feasible time interval, polarity assignment to the sinks is possible using the buffers and inverters in the interval, while clock skew constraint (κ) is satisfied. PeakMin collects all the feasible intervals of the arrival times. (Step 3) In the last step, for each feasible interval obtained in step 2, PeakMin performs polarity assignment with sizing to minimize the peak noise and selects the solution corresponding to the interval which has the lowest peak noise. PeakMin formulated the polarity assignment problem with sizing into a Knapsack problem and solved it optimally in pseudopolynomial time.

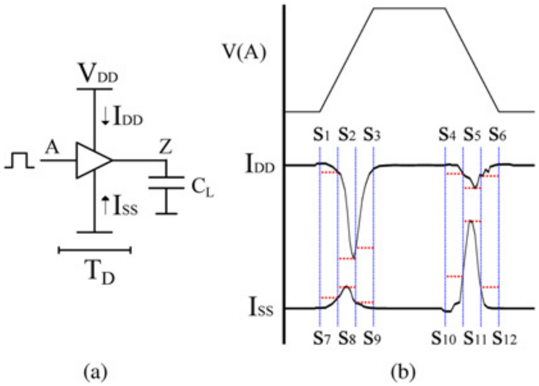


Fig. 7. Characterizing a buffer in B assigned to a sink. (a) Clock pulse is applied to the input of the buffer. Then, the current waveforms of I_{DD} and I_{SS}, and the signal propagation time T_D of the buffer are measured and recorded. (b) Only the hot spots of waveforms of I_{DD} and I_{SS} are captured as most of the nonzero sampled values are located near the rising and falling edges of the input. There are 12 sampling points, $s_1, s_2, \cdots, s_{12}$ and $s_1, \cdots, s_6$ are from I_{DD} and $s_7, \cdots, s_{12}$ from I_{SS}. Inverters in I are also similarly characterized.

B. Time Sampling Points

To compute (2), it is required to measure the value of $noise(\phi(e_i), s)$. That is, the current waveforms of I_{SS} and I_{DD} for each buffer/inverter in $B \cup I$ must be known. Instead of running a full-fledged HSPICE simulation on the clock tree, every combination of buffers/inverters in $B \cup I$ and sinks in L can be characterized to calculate the approximate values of the corresponding *noise* function. Fig. 7(a) shows a node in clock tree on which we focus to extract noise data. By applying a clock pulse to the input A, the current waveforms of I_{DD} and I_{SS} and the signal propagation time T_D of the buffer are measured and recorded as a data entry in the lookup table *noise*. We use the linear interpolation method to build *noise* function. Note that we capture only the hot spots of waveforms of I_{DD} and I_{SS} since the sampled values in the current waveforms are mostly zero and the nonzero values are located near the rising and falling edges of the input. For example, in Fig. 7(b), times $s_1, s_2, \cdots, s_{12}$ are selected as the time sampling points to form S in (2).

Note that the waveforms depend on the input slew as well. We have measured the average clock slew in the clock trees and used slew of 20 ps during profiling, where 20 ps is the value of 1 to 3 ps sharper than the average clock slew. The rationale is that with the sharper clock transition, upper bounds for noise can be obtained as I_{SS} and I_{DD} would make sharper transition too. However, the input clock slew must not be too different from the one observed in the clock tree since it would lead to inaccurate estimation.

V. POLARITY ASSIGNMENT FOR SINGLE POWER MODE DESIGNS

This section describes two algorithms to solve the polarity assignment for designs of a single power mode. One is an

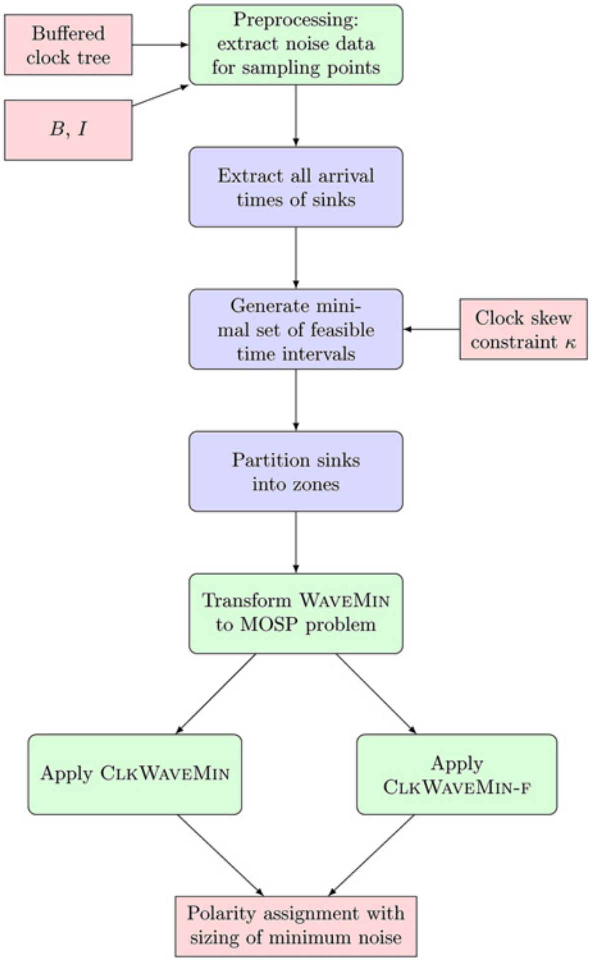

Buffered clock tree
B, I
Preprocessing: extract noise data for sampling points
Extract all arrival times of sinks
Generate minimal set of feasible time intervals
Clock skew constraint κ
Partition sinks into zones
Transform WaveMin to MOSP problem
Apply ClkWaveMin
Apply ClkWaveMin-f
Polarity assignment with sizing of minimum noise

Fig. 8. Flow of our proposed polarity assignment for designs with a single power mode.

approximation algorithm called ClkWaveMin and the other is a fast heuristic algorithm called ClkWaveMin-f.

A. Overview

Fig. 8 shows the flow of our proposed clock polarity assignment. The inputs to our polarity assignment framework are a synthesized buffered clock tree, libraries B and I, and clock skew constraint κ, from which the preprocessing of extracting noise data and sampling points is performed, followed by generating all feasible time intervals of the arrival times that are computed by mapping every element in $B \cup I$ to sinks.

Since power/ground noise is a local effect, we divide the design into several zones and apply our algorithms to the zones one by one to minimize the peak current at each zone, which targets the maximum peak current value as the objective cost to be minimized. Now, for the rest of this section we focus on the discussion of ClkWaveMin and ClkWaveMin-f to be applied to a time interval $[t - \kappa, t]$ with a zone z_i.

We transform the WaveMin problem into the min–max problem (or sometimes called max ordering problem in some literature [32]) which we can solve by solving the multi-objective shortest path (MOSP) problem, for which we use a fully polynomial ϵ-approximation algorithm devised by Warburton [33]. Our formulation of WaveMin problem to the MOSP problem is described in Section V-B, by which we then use Warburton's approximation algorithm to solve the transformed MOSP problem that is a fully polynomial algorithm in time and space criteria: $O(rn^3(n/\epsilon)^{2r})$ time and $O(rn(n/\epsilon)^r)$ space where r is the arc weight dimension and n is the number of vertices in MOSP graph.

B. Mapping WaveMin Problem to MOSP Problem

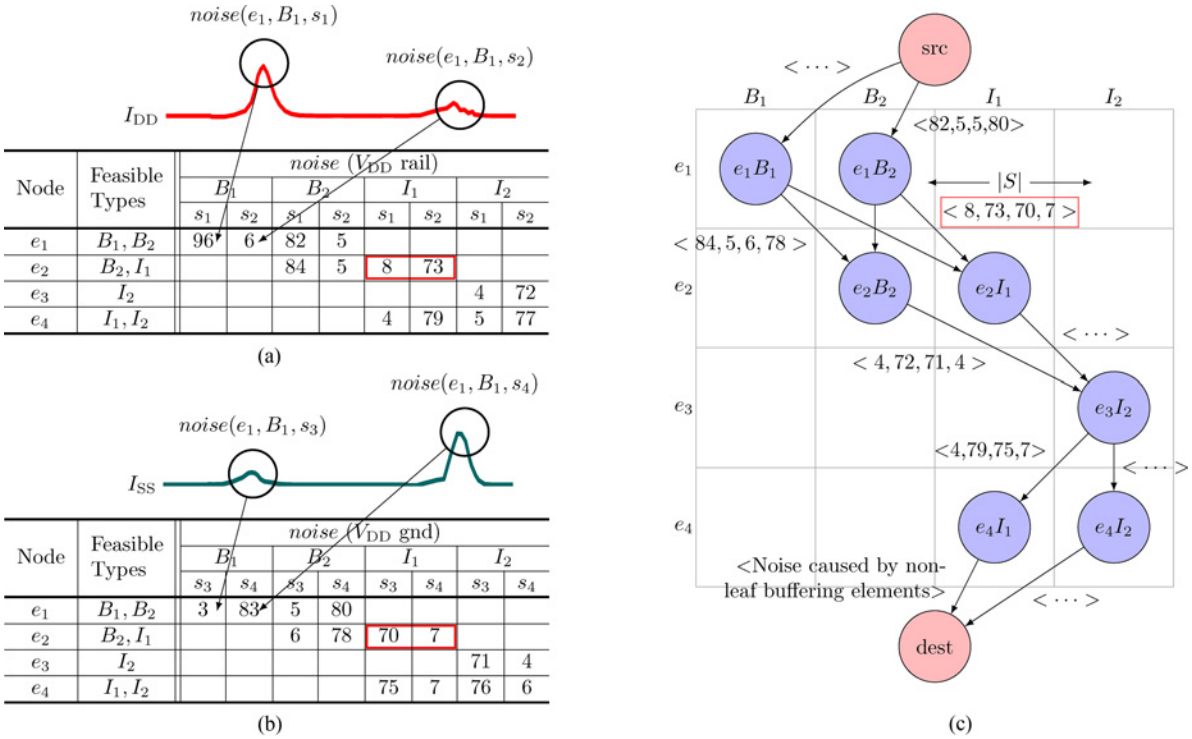
We first formally define the MOSP problem.

Problem 4 (MOSP): Given a directed graph $G = (V, A)$, r dimensional vector weight $w \in W(a)$ for each arc $a \in A$ and two vertices $s, t \in V$, find all Pareto-optimal paths[4] from s to t, where the cost of a path is defined as the sum of arc weights along the path.

Even for $r = 2$, it is known that the decision version of MOSP problem is NP-complete [34]. Fig. 9 shows an example of converting an instance of WaveMin in an interval $[t_1 - \kappa, t_1]$ to a graph of MOSP problem. Column *Feasible types* in the tables in Fig. 9(a) and (b) are the buffers and inverters in $B \cup I$ that can be assigned to the corresponding sink in L without violating clock skew constraint, and the numbers in the entries of the tables represent the corresponding noise values of I_{DD} and I_{SS}. For example, the number (= 96) in the entry at location (e_1, B_1, s_1) in Fig. 9(a) indicates that the peak noise of I_{DD} at time s_1 is 96 when sink e_1 is assigned with buffer B_1, and the number (= 75) in the entry at location (e_4, I_1, s_3) in Fig. 9(b) indicates that the peak noise of I_{SS} at time s_3 is 75 when sink e_4 is assigned with inverter I_1. Note that the WaveMin instance has four time sampling slots $s_1, \cdots, s_4$ where s_1 and s_2 are the sampling slots for I_{DD} noise waveform and s_3 and s_4 are for I_{SS}. The transformed MOSP graph of the WaveMin instance in Fig. 9(a) and (b) is shown in Fig. 9(c). The MOSP graph has vertices with row (representing sinks) and column (representing elements in $B \cup I$) properties, and each vertex corresponds to a distinct feasible assignment of a sink to a buffer or inverter in $B \cup I$ in the WaveMin instance. For example, the vertex labeled with e_2B_2, i.e., located at the intersection of row e_2 and column B_2 corresponds to the option of assigning sink e_2 with buffer B_2 in Fig. 9(a). A vertex in row i has an incoming arc from every vertex in row $i-1$. The MOSP graph has two dummy vertices called *src* and *dest*. The *src* is directed to every vertex in the first row and every vertex in the last row is directed to *dest*. For an arc (u, v) where v is at row r and column c, the arc weight is defined as $w(u, v) = (noise(e_r, c, s_1), \cdots, noise(e_r, c, s_{|S|}))$. For example, any arc directed to vertex e_2I_1 in Fig. 9(c) has arc weight of $w(\cdot, e_2I_1) = (noise(e_2, I_1, s_1),\ noise(e_2, I_1, s_2),\ noise(e_2, I_1, s_3),\ noise(e_2, I_1, s_4)) = (8, 73, 70, 7)$, as shown in the red box in Fig. 9(c). One exception is vertex *dest*. For the arcs directed to *dest*, the arc weights are assigned to reflect the noise caused by the non-leaf buffering elements of the clock tree to account for observation 1 in Section II. Algorithm 1 describes the conversion of a WaveMin instance to an MOSP graph.

The multidimensional distance $w(u, v)$ is assigned as the estimated noise value when option v is selected for the final

[4] It corresponds to finding all nondominated paths in the graph, that is, paths for which it is not possible to find a better total weight on a vector entry without getting worse on some of the other entries.



Node	Feasible Types	noise (V_{DD} rail)							
		B_1		B_2		I_1		I_2	
		s_1	s_2	s_1	s_2	s_1	s_2	s_1	s_2
e_1	B_1, B_2	96	6	82	5				
e_2	B_2, I_1			84	5	8	73		
e_3	I_2							4	72
e_4	I_1, I_2					4	79	5	77

Node	Feasible Types	noise (V_{DD} gnd)							
		B_1		B_2		I_1		I_2	
		s_3	s_4	s_3	s_4	s_3	s_4	s_3	s_4
e_1	B_1, B_2	3	83	5	80				
e_2	B_2, I_1			6	78	70	7		
e_3	I_2							71	4
e_4	I_1, I_2					75	7	76	6

Fig. 9. Example of converting an instance of ClkWaveMin with interval $[t_1 - \kappa, t_1]$ to an MOSP graph. For each fixed time interval, the feasibility of buffer and inverter types for each node can be calculated and the corresponding *noise* values for each noise slot can be determined. In this example, there are four slots $s_1, \cdots, s_4$, where s_1, s_2 are sampling slots for I_{DD} noise waveform and s_3, s_4 are for I_{SS}. The MOSP graph has vertices with row and column properties. For example, the vertex (labeled as e_2I_1) located at row e_2 column I_1 corresponds to the option of assigning node e_2 with inverter type I_1. The graph has two additional nodes *src* and *dest* marked as red color. A vertex in row i has incoming arcs from all the vertices in row $i-1$. For an arc (u, v) where v is at row r column c, the arc weight is defined as $w(u, v) = (noise(r, c, s_1), \cdots, noise(v, c, s_{|S|}))$. For example, any arc which is directed to vertex e_2I_1 has arc weight of $w(\cdot, e_2I_1) = (noise(e_2, I_1, s_1), noise(e_2, I_1, s_2), noise(e_2, I_1, s_3), noise(e_2, I_1, s_4)) = (8, 73, 70, 7)$. One exception is vertex *dest*. For the arcs directed to *dest*, the arc weights are assigned to reflect the noise caused by the non-leaf buffering elements of clock tree.

assignment; hence, the distance of path $s \rightsquigarrow t$ represents the (accumulated) noise, and the vertices in between the path indicate the corresponding assignments. For example, if vertex e_2B_2 is on path $s \rightsquigarrow t$, node e_2 should be assigned with a buffer of type B_2. The degree of MOSP graph G is $O(|B|+|I|)$ since a node can have at most $|B|+|I|$ incoming and at most $|B|+|I|$ outgoing arcs. Therefore, the number of arcs in G is bounded by $O(2(|B|+|I|)|L|+2) = O(|L|)$, since there are only limited available types of buffers and inverters, meaning that $|B|+|I|$ is a constant. Last, arc weight dimension r equals $|S|$.

The resulting problem is solved with Warburton's algorithm [33] and all approximated Pareto-optimal paths from s to t are found. Among the retrieved paths, we take the path with the minimum worst distance as our WaveMin solution. The path is a valid solution to WaveMin problem because the MOSP graph is directed acyclic since arc (u, v) exists between vertices u and v only if $row(v) - row(u) = 1$. The overall runtime of Warburton's approximation algorithm is given as $O(rn^3(n/\epsilon)^{2r})$ and substituting r and n yields $O(|S||L|^3((|B|+|I|)\cdot|L|/\epsilon)^{2|S|})$. The final selection of min–max solution among $O(r(n/\epsilon)^r)$ Pareto-optimal solutions has execution time of $O(r \times r(n/\epsilon)^r + r(n/\epsilon)^r) = O(|S|^2((|B|+|I|)\cdot|L|/\epsilon)^{|S|})$.

C. Fast Algorithm

In addition to using Warburton's approximation algorithm, we propose a fast version ClkWaveMin-f with lower time and space complexity than ClkWaveMin. In contrast to ClkWaveMin that tries to find an optimal or approximate shortest path, ClkWaveMin-f performs the polarity assignment vertex by vertex basis iteratively, by selecting and assigning a buffer or an inverter with the least noise-worsening first from its current state. Let *sum* denote the *noise* expectation contributed by the currently selected set of vertices in the MOSP graph $G(V, A)$, as well as all the non-leaf nodes in the clock tree. Then, for each unselected vertex $v \in V$, $M(v) = max(sum(s_i) + noise(v, s_i),\ s_i \in S)$ is calculated and the vertex with the minimum $M(v)$ is selected as the vertex of choice in this iteration. For next iteration, *sum* is updated and the other vertices in the same row as v are removed from V to prevent the leaf node associated to v from further sizing or polarity assignment. The iteration continues until there is no more vertex in V. The space used by ClkWaveMin-f is $O(|S||L|)$ since there are $O(|L|)$ vertices in the MOSP graph and the running time is $O(|S||L|^2)$.

VI. Polarity Assignment for Multiple Power Mode Designs

In real designs, there may be arrival time variations induced by many causes. It has been shown in [27] that buffer/inverter sizing can be utilized to satisfy the clock skew for clock trees that has multiple operating points due to thermal variations.

Algorithm 1 Conversion of WaveMin instance to MOSP graph.

```
function WAVEMIN 2MOSP($L$, $\kappa$, $noise$, $B$, $I$, $S$)
    $V \leftarrow \emptyset$;                                   ▷ Vertices
    $A \leftarrow \emptyset$;                                   ▷ Arcs
    for $e_i \in L$ do                                          ▷ Vertex construction
        for $type \in$ feasible subset of $B \cup I$ for $e_i$ do
            // Allocate and place vertices at proper place
            $v \leftarrow$ new_vertex();
            row($v$) $\leftarrow i$;
            column($v$) $\leftarrow type$;
            $V \leftarrow V \cup \{v\}$;
        end for
    end for
    Create and prepend a row, as the new first (0-th) row;
    Place a dummy node $src$ in the first row;
    for $r \in$ rows do                                         ▷ Arc construction
        $q \leftarrow$ next_row($r$);
        for all $(u, v)$, where $u \in r$ and $v \in q$ do
            $a = (u, v)$;
            $A \leftarrow A \cup \{a\}$;
            $type$ = column($v$);
            // $S$ is the set of sampling points
            weight($a$) $\leftarrow noise(e_r, type, S)$;
        end for
    end for
    $r \leftarrow$ the current last row;
    Create and append a row, as the new last ((r+1)-th) row;
    Place a dummy node $dest$ in the last row;
    for all vertices $u$ in row $r$ do                          ▷ Arcs to $dest$ vertex
        Allocate and add a new arc ($u$, $dest$) in $A$;
        weight($a$) $\leftarrow noise$(non-leaf, $S$)           ▷ Supporting observation 1 in Section II
    end for
    return $G(V, A)$;
end function
```

However, they assumed that the peak noise is invariant with respect to the temperature since the peak noise is the greatest at the coolest state, and this value can be used as a pessimistic upper bound of the noise for other operating points. Unfortunately, this assumption is invalid in multiple power mode designs, in which the primary source of the delay variations is the local adjustment of the power supply voltage, V_{DD}. In this section, we use the concept of intersection of intervals to satisfy the clock skew constraint and provide a method for minimizing noise in multipower mode designs.

Consider the example of clock tree shown in Fig. 10 with two power modes M_1 and M_2 such that in M_1, both of the voltage islands $A1$ and $A2$ operate at V_{DD} =1.1 V, by which all leaf nodes (i.e., sinks) have arrival time of 70, while in M_2, $A2$ operates at V_{DD} =0.9V, which increases the arrival times of e_3 and e_4 from 70 to 78 (+4 from the parent node of e_3 and e_4 and another +4 from each of e_3 and e_4). The clock tree must support both M_1 and M_2 under some bounded clock skew constraint. Let the skew bound κ be 5 in this example. Clearly, the clock skew in Fig. 10 is violated in M_2.

TABLE III
CHARACTERIZATION OF B = {BUF_X1, BUF_X2} AND I = {INV_X1, INV_X2}. T_D REPRESENTS THE SIGNAL PROPAGATION DELAY, P+ AND P- INDICATE THE VALUES OF THE PEAK I_{DD} AT THE RISING AND FALLING EDGES OF THE INPUT. (FOR BREVITY, WE OMIT HERE THE VALUES OF P+ AND P- OF I_{SS})

Type	V_{DD} =0.9V			V_{DD} =1.1V		
	T_D	P+	P-	T_D	P+	P-
BUF_X1	27	120	10	24	130	13
BUF_X2	23	234	36	19	255	44
INV_X1	24	10	120	21	13	130
INV_X2	22	36	234	17	44	255

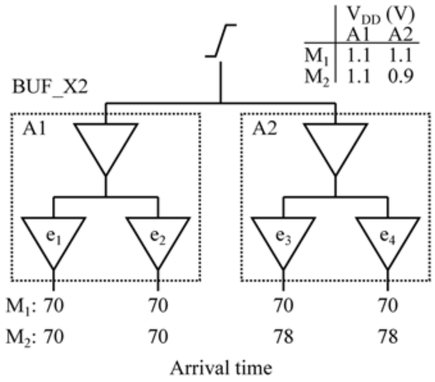


Fig. 10. Example of clock tree that has two voltage islands $A1$ and $A2$ such that in power mode M_1, both $A1$ and $A2$ operate at V_{DD} =1.1 V and in power mode M_2, $A1$ operates at 1.1 V while $A2$ operates at 0.9 V. All nodes are initially assigned with BUF_X2.

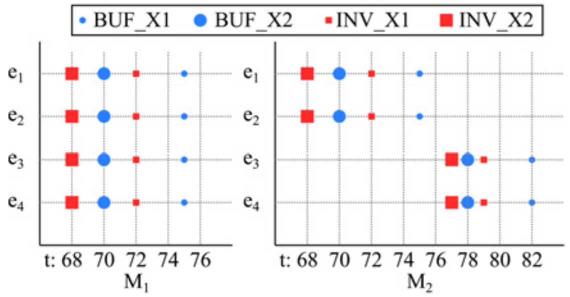


Fig. 11. Illustration of intervals of arrival times for the example in Fig. 10 and Table III. Each dot in the grids represents a buffer or inverter. For example, the large red dot located at position (68, e_3) in M_1 indicates that e_3 has arrival time of 68 when INV_X2 is assigned to it in power mode M_1.

To tackle this problem, we first compute the sets of feasible intervals for all power modes, and then intersect them to identify, for each sink in L, the buffer/inverter types in $B \cup I$ that can be assigned to the sink. For example, Fig. 11 illustrates all intervals for power modes M_1 and M_2 in Fig. 10. With $\kappa = 5$, in M_1 there are time intervals [70, 75], [67, 72], [65, 70], and [63, 68] defined by arrival times 75, 72, 70, and 68, and all of them are feasible intervals. In M_2, there are eight intervals but only intervals [74, 79], [73, 78], and [72, 77] are feasible. With feasible intervals in all power modes, we are now ready to obtain intersections of feasible intervals in different power modes. Fig. 11 involves 12 intersections between M_1 and M_2, i.e., {[70, 75], [67, 72], [65, 70], [63, 68]}×{[74, 79], [73, 78], [72, 77]}. For example, intersection (70, 79) (= [65, 70]×[74, 79]) denotes that interval [65, 70] of M_1 and [74, 79] of M_2

TABLE IV
NODE-TO-TYPE FEASIBILITY INFORMATION OF ALL FEASIBLE INTERSECTIONS, WHEN THE CLOCK SKEW BOUND IS $\kappa = 5$

Intersection	Node	BUF_X1	BUF_X2	INV_X1	INV_X2
(75, 79)	e_1	**fsbl**	infsbl	infsbl	infsbl
	e_2	**fsbl**	infsbl	infsbl	infsbl
	e_3	infsbl	**fsbl**	**fsbl**	infsbl
	e_4	infsbl	**fsbl**	**fsbl**	infsbl
(75, 78)	e_1	**fsbl**	infsbl	infsbl	infsbl
	e_2	**fsbl**	infsbl	infsbl	infsbl
	e_3	infsbl	**fsbl**	infsbl	infsbl
	e_4	infsbl	**fsbl**	infsbl	infsbl
(72, 77)	e_1	infsbl	infsbl	**fsbl**	infsbl
	e_2	infsbl	infsbl	**fsbl**	infsbl
	e_3	infsbl	infsbl	infsbl	**fsbl**
	e_4	infsbl	infsbl	infsbl	**fsbl**

fsbl: assignment with no skew violation
infsbl: assignment that causes skew violation

are chosen, which means extracting, for each sink, a maximal subset of buffers and inverters that are contained in both of the sets of feasible buffers and inverters in [65, 70] of M_1 and [74, 79] of M_2. In Fig. 11, since [65, 70] of M_1 has {BUF_X2, INV_X2} for sink e_1, {BUF_X2, INV_X2} for e_2, {BUF_X2, INV_X2} for e_3, and {BUF_X2, INV_X2} for e_4 while [74, 79] of M_2 has {BUF_X1} for e_1, {BUF_X1} for e_2, {BUF_X2, INV_X1, INV_X2} for e_3, and {BUF_X2, INV_X1, INV_X2} for e_3, intersection (70, 79) returns ϕ (= {BUF_X2, INV_X2}∩{BUF_X1}) for e_1, ϕ (= {BUF_X2, INV_X2}∩{BUF_X1}) for e_2, {BUF_X2, INV_X2} (= {BUF_X2, INV_X2}∩{BUF_X2, INV_X1, INV_X2}) for e_3, and {BUF_X2, INV_X2} (= {BUF_X2, INV_X2}∩{BUF_X2, INV_X1, INV_X2}) for e_4. An intersection $(t_i, \cdots, t_j)$ is called a feasible intersection if the resulting set of buffers and inverters for every sink is not empty and called an infeasible intersection, otherwise. Thus, (70, 79) is an infeasible intersection.

The example in Fig. 11 has three feasible intersections (75, 79), (75, 78), and (72, 77) among 12 possible intersections. The intersection results are summarized in Table IV in which *fsbl* indicates that its buffer or inverter is feasible to use in that interval and *infsbl* indicates that it is not feasible. As long as only the feasible types are selected, the clock skew is satisfied for all power modes. The difficulty lies in minimizing the noise for multiple modes as there are multiple different noise values from multiple modes to optimize. In this noise optimization problem, the objective is to minimize the worst case noise. In other words, noises in M_1 and M_2 for the example in Figs. 10 and 11 have the same priority or weight; if we concatenate the noise values from all the modes into one vector, this is still a valid cost formulation of MOSP problem. Hence, we translate the noise from each power mode as an extra dimension in the MOSP problem formulation. Fig. 12 shows the MOSP graph of the intersection (75, 79). As with optimization of single power mode, MOSP graph vertices represent which buffer or inverter types are available to each sink. The arc weights are composed of noise from multiple modes. For example, the arc from e_1B_1 to e_2B_1 has weight of <130, 13, 120, 10> where 130 and 13 are from P+ and P- columns of V_{DD} =1.1 V and 120 and 10 are from V_{DD} =0.9 V in BUF_X2 row of Table II. Optimizing this MOSP problem (without approximation) yields noise of <268, 268, 280 266> with the assignment of BUF_X1 to e_1, BUF_X1 to e_2, INV_X1 to e_3, and INV_X1 to e_4, resulting in clock skew of 3 in M_1 and 4 in M_2. Thus, the worst noise for the feasible intersection (75, 79) is 280. Likewise, the worst noise for the other intersections (75, 78) and (72, 77) is each 770. Consequently, the best solution is from (75, 79) since its noise is the least.

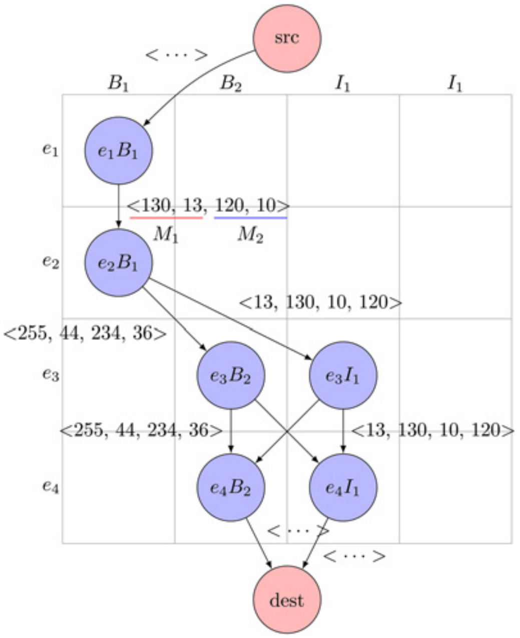


Fig. 12. Updated MOSP graph supporting intersection (75, 79) in Fig. 11. The cost formulation of MOSP problem is still valid.

Although ClkWaveMin can endure some degree of clock skew, the arrival time variation may be too large in designs of multiple power modes so that it is impossible to satisfy the clock skew without the use of ADBs. Fig. 13 is the flow of ClkWaveMin-M, an extension of ClkWaveMin for multiple power mode designs. Given a synthesized clock tree and clock skew constraint κ, the clock signal arrival times in each power mode are calculated by ClkWaveMin and noise is minimized, if it is possible to satisfy κ with only polarity adjustments and buffer/inverter sizing. If it fails, ADBs are inserted to satisfy κ; then, ClkWaveMin is executed again, in which the inverter library I contains an ADI in Fig. 4, as well as the normal inverters of different size. Note that ADBs that have been already allocated must not be replaced with buffers or inverters since ADBs are essential to meet the clock skew bound in multiple power modes; each ADB can be replaced with an ADI or stay as ADB. Likewise, non-ADBs may not become ADBs or ADIs since this replacement leads to unnecessary increase of area. This restriction is handled during feasible buffer/inverter type computation by checking if the leaf node is an ADB or not. After the ADB insertion, at least one WaveMin solution exists for the ADB inserted clock tree—the trivial solution in which no buffer sizing and polarity assignment are applied.

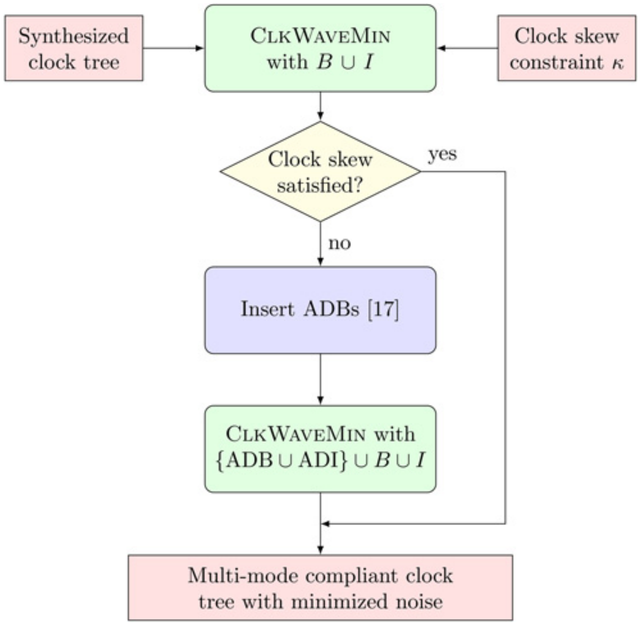

Synthesized clock tree
ClkWaveMin with $B \cup I$
Clock skew constraint κ
Clock skew satisfied?
yes
no
Insert ADBs [17]
ClkWaveMin with $\{ADB \cup ADI\} \cup B \cup I$
Multi-mode compliant clock tree with minimized noise

Fig. 13. Flow of ClkWaveMin-M, an extension of ClkWaveMin to support multiple power mode designs. Note that module Insert-ADB resolves the clock skew violation and the subsequent module ClkWaveMin performs the polarity assignment with library $B \cup I \cup ADB \cup ADI$ while retaining the satisfaction of clock skew constraint.

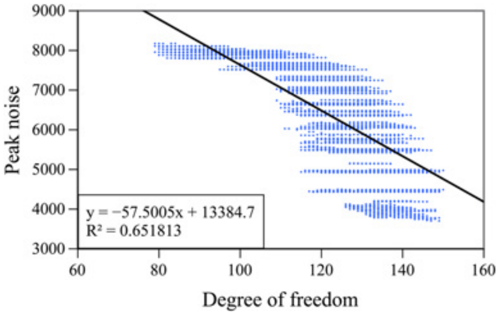


Fig. 14. Relationship between peak noise and the degree of freedom which measures the flexibility of polarity assignment of a feasible intersection. The plot has been acquired by optimizing s35932 circuit in ISCAS'89 benchmark set.

One of the bottlenecks of this optimization is the intersection process. In [27], the time complexity of the intersection process is $O(|L|^{(M+1)} \cdot (|B|+|I|)^{(M+1)})$, where M is the number of power modes. The complexity increases exponentially as the number of modes increases. In thermal mode, this was a less concern since only a few coolest and hottest modes may be considered. Although even the brute force method may have a fast execution time in practice, depending on the input size—this is because most of the intersections are not feasible and pruned early during execution—it is possible to improve the performance through the use of the concept of degree of freedom: given a feasible intersection, the degree of freedom is calculated by simply counting the total number of the buffers and inverters produced by the intersection for all sinks. For instance, in Table IV, the degree of freedom of intersection (75, 79) is 6 and (75, 78) is 4. As illustrated in Fig. 14, it is observed that there is a negative correlation between the degree of freedom and peak noise; the more the freedom is, the lower the noise is. Hence, we use the degree of freedom to prune out less free intersections during the intersection process.

VII. Experimental Results

A. Experimental Setup

The proposed algorithms ClkWaveMin, ClkWaveMin-f and ClkWaveMin-M have been implemented in C++ language on a Linux machine and tested on ISCAS'89 benchmark circuits. The benchmarks were synthesized using Synopsys' Design Compiler and clock trees were synthesized as zero skew trees (<10 ps clock skew in HSPICE simulations) with Synopsys' IC Compiler, using Nangate 45-nm Open Cell Library [35]. RC extractions were performed on IC Compiler and HSPICE simulation was done on the clock trees. To measure the clock noise in the power/ground network, the power grid model in [36] was used and voltage fluctuations at the source/drain of each buffer/inverters were measured. In addition, to synthesize ISPD'09 CTS contest benchmarks, we have employed the algorithm in [37].

We also implemented the best ever known polarity assignment algorithm ClkPeakMin [27] for the comparison with our algorithms. All leaf nodes were attempted to be assigned to any of BUF_X8, BUF_X16, INV_X8, and INV_X16. The benchmark circuits were partitioned into a square grid of zones, where the grid size had been determined empirically as 50 × 50 μm. Larger zones tend to yield better optimization results [27] since the optimizer can consider more leaf buffering elements than smaller zones, although some saturation point exists. Moreover, excessively large zones should be avoided since it leads to prolonged optimization time due to increased subproblem size and the zones may suffer from locally large noise, due to concentrated buffer/inverter in some local area within the zone. On average, each zone contained 4.3 nodes for ISCAS'89 benchmarks and 4.9 nodes for ISPD'09 benchmarks. In particular, benchmark design S35932 has 7.1 nodes in each zone on average.

B. Assessing the Performance of Approximation Algorithm ClkWaveMin Over ClkPeakMin [27]

Table V summarizes the comparison of the results produced by ClkPeakMin [27] and ClkWaveMin when clock skew bound is set to κ = 20 ps. V_{DD} and Gnd noises are the maximum voltage fluctuations observed in the power and ground grids, respectively. In summary, ClkWaveMin reduces the peak current by 15.6% on average.

C. Assessing the Performance of Fast Algorithm ClkWaveMin-f Over ClkWaveMin

Table VI shows comparison with results by ClkWaveMin using various time sampling points and our fast ClkWaveMin-f ($|S|$ = 158). For $|S|$ = 4, from I_{SS} and I_{DD} waveforms, two values from each current profile were obtained by extracting the maximum value from the first and the second halves of the waveform. We can see that the use of more sampling points leads to a further reduction in peak current. Furthermore,

TABLE V

Comparison of Results by ClkPeakMin [27] and ClkWaveMin When κ = 20 ps, ϵ = 0.01, $|S|$ = 158. Column n Denote the Total Number of Buffering Elements, Including Both Non-Leaf Nodes and Leaf Nodes and $|L|$ Is the Number of Leaf Buffering Elements

Bench-mark Circuit	n	$\lvert L\rvert$	ClkPeakMin [27]			ClkWaveMin			Improvement		
			V_{DD} noise (mV)	Gnd noise (mV)	Peak curr. (mA)	V_{DD} noise (mV)	Gnd noise (mV)	Peak curr. (mA)	V_{DD} noise (%)	Gnd noise (%)	Peak curr. (%)
s13207	58	50	3.84	6.06	6.45	4.19	6.62	7.25	-9.19	-9.17	-12.39
s15850	22	19	1.48	2.00	3.01	1.48	2.00	3.01	0.00	0.00	0.00
s35932	323	246	7.54	8.66	21.59	4.40	8.39	15.59	41.72	3.03	27.79
s38417	304	228	7.79	7.55	19.83	3.78	7.04	11.88	51.46	6.78	40.09
s38584	210	169	5.38	6.54	16.92	4.30	7.94	11.58	20.02	-21.40	31.56
ispd09f31	328	111	0.95	1.51	75.50	0.94	1.50	62.17	1.05	0.66	17.66
ispd09f34	210	69	0.53	0.93	49.12	0.96	1.51	46.85	-81.13	-62.37	4.62
Average									3.42	-11.78	**15.62**

TABLE VI

Comparison With ClkWaveMin (ϵ = 0.01) Varying the Number of Time Points and ClkWaveMin-f ($|S|$ = 158, κ = 20 ps)

Bench-mark circuit	ClkPeakMin		ClkWaveMin						ClkWaveMin-f	
			$\lvert S\rvert = 4$		$\lvert S\rvert = 8$		$\lvert S\rvert = 158$		$\lvert S\rvert = 158$	
	Peak curr. (mA)	Exec. time (ms)	Peak curr. (mA)	Exec. time (ms)	Peak curr. (mA)	Exec. time (ms)	Peak curr. (mA)	Exec. time (ms)	Peak curr. (mA)	Exec. time (ms)
s13207	6.5	<0.01	7.2	<0.01	7.2	<0.01	7.2	<0.01	7.25	<0.01
s15850	3.0	<0.01	3.0	<0.01	3.0	<0.01	3.0	<0.01	3.01	<0.01
s35932	21.6	0.05	16.9	0.19	15.6	1.07	15.6	1.02	18.8	0.01
s38417	19.8	0.04	13.0	0.08	11.9	0.51	11.9	0.49	11.4	<0.01
s38584	16.9	0.03	13.6	1.02	11.6	0.7	11.6	0.66	10.3	0.01
ispd09f31	75.5	0.01	71.0	0.02	71.0	0.02	62.2	0.07	68.7	0.01
ispd09f34	49.1	<0.01	50.8	0.01	50.8	0.01	46.9	0.02	54.9	<0.01

our fast greedy algorithm ClkWaveMin-f produces the result close to that by ClkWaveMin with 158 sampling points, but run time is significantly fast. For the seven benchmark circuits, ClkWaveMin found near-optimal assignments that were expected to have lower noise than that by ClkWaveMin-f. However, the results from more accurate HSPICE simulation show that ClkWaveMin-f sometimes yields a superior polarity assignment than ClkWaveMin. This is mainly because of the modeling inconsistency between HSPICE and our noise model: $noise(\phi(e_i), s)$ is affected by the polarity/sizing of neighboring nodes.

D. Impact of Process Variations on ClkWaveMin

Even though the optimizations are based on the nominal values, to investigate the effect of process variation on the products of ClkPeakMin and ClkWaveMin, Monte Carlo simulations were run on the clock trees, where the trees were optimized with κ = 100 ps and $|S|$ = 158. Wire widths, wire lengths, buffer/inverter widths, and threshold voltages were randomized in which all the variables follow the Gaussian random distribution of $N(\mu, \sigma^2)$, where μ is the variables' respective nominal value and σ satisfies σ/μ = 5%. For each benchmark circuit, 1000 randomized instances were generated for HSPICE simulations.

On average, 95.5% and 83.9% of the clock trees produced by ClkPeakMin and ClkWaveMin satisfied the clock skew bound κ, respectively. This is because some of the circuits optimized by ClkWaveMin had the nominal clock skews that were very close to κ so that they were more sensitive to the variations; ClkWaveMin tries to disperse the noise waveform over time slots, but this leaves less room for variations. The average values of peak current and V_{DD}/Gnd noise were close to that of Table V. Since the circuits have different noise values, we normalized the standard deviations of each benchmark circuit as $\hat{\sigma}/\hat{\mu}$, where $\hat{\mu}$ is the observed average value and $\hat{\sigma}$ is the observed standard deviation. The average normalized standard deviations for peak current, V_{DD}, and Gnd noises were 0.054, 0.082, and 0.084, respectively, when optimized by ClkPeakMin. For ClkWaveMin, we obtained 0.062, 0.086, and 0.086. Since neither of the algorithms are variation tolerant, these similar figures would be expectable.

E. Assessing the Performance of ClkWaveMin-M Supporting Designs With Multiple Power Modes

ClkWaveMin-M has been applied to the benchmark circuits, given four power modes. Each benchmark was partitioned into four to ten power domains with each having two operating modes at supply voltage levels of 0.9 V and 1.1 V. Table VII summarizes the results of ClkWaveMin-M. While any ADB embedding algorithms may be used, we employed the algorithm in [17], which is known to insert a minimum number of ADBs in multiple power modes to resolve the clock skew violation. The optimization results produced by ClkWaveMin-M have been compared with the noise-unaware clock trees (denoted as ADB-embedded-only in Table VII) produced by [17] which inserts ADBs to meet the clock skew constraint for every power mode. It is evident from the table that ClkWaveMin-M reduces noise on multiple power mode designs, without violating clock skew bound. On average, ClkWaveMin-M achieves 16.38% peak current reduction. One

TABLE VII

RESULT PRODUCED BY CLKWAVEMIN-M THAT SUPPORTS DESIGNS WITH MULTIPLE POWER MODES

Bench-mark ckt.	Skew Bound (ps)	ADB-embedding-only [27]				CLKWAVEMIN-M					Improvement		
		Peak curr. (mA)	V_{DD} noise (mV)	GND noise (mV)	#ADBs	Peak curr. (mA)	V_{DD} noise (mV)	GND noise (mV)	#ADBs	#ADIs	Peak curr. (%)	V_{DD} noise (%)	GND noise (%)
s13207	90	21.53	6	6.53	33	18.91	5.68	8.11	33	0	12.17	5.33	-24.20
	110	20.12	5.82	7.95	33	17.37	5.83	6.7	33	0	13.67	-0.17	15.72
	130	19.71	5.42	7.21	53	19.12	6.99	8.11	53	0	2.99	-28.97	-12.48
s15850	90	8.77	2.4	3.78	18	7.33	2.92	2.81	18	0	16.42	-21.67	25.66
	110	8.83	2.27	3.67	33	8.33	2.65	3.14	33	0	5.66	-16.74	14.44
	130	8.88	2.26	3.85	0	8.36	3.46	3.26	0	0	5.86	-53.10	15.32
s35932	90	99.27	29.84	26.7	75	104	33.03	27.96	69	6	-4.76	-10.69	-4.72
	110	100.4	30.98	26.66	164	93.05	29.93	24.71	164	0	7.32	3.39	7.31
	130	97.44	30	26.38	293	78.39	23.93	23.24	293	0	19.55	20.23	11.90
s38417	90	74.94	22.05	20.43	113	69.46	21.62	21.5	109	4	7.31	1.95	-5.24
	110	80.4	23.88	21.61	127	84.55	27.1	19.51	127	0	-5.16	-13.48	9.72
	130	78.4	23.72	17.51	288	53.46	18.08	22.93	288	0	31.81	23.78	-30.95
s38584	90	56.06	16.12	15.72	279	51.45	15.75	15.81	213	66	8.22	2.30	-0.57
	110	58.23	15.36	16.16	167	51.01	14.95	16.25	101	66	12.40	2.67	-0.56
	130	57.91	16.75	16.57	275	56.02	16.86	15.02	261	14	3.26	-0.66	9.35
f31	90	90.71	0.54	0.94	12	62.65	0.27	0.51	12	0	30.93	50.00	45.74
	110	92.29	0.54	0.94	2	51.36	0.5	0.9	2	0	44.35	7.41	4.26
	130	79.27	0.53	0.94	69	47.97	0.28	0.51	69	0	39.49	47.17	45.74
f34	90	65.17	0.53	0.93	18	49.88	0.52	0.92	18	0	23.46	1.89	1.08
	110	50.03	0.53	0.94	32	36.28	0.51	0.89	32	0	27.48	3.77	5.32
	130	57.32	0.53	0.92	31	33.54	0.27	0.5	31	0	41.49	49.06	45.65
Average											**16.38**	3.50	8.50

interesting data to note is S15850 with skew bound of 130 ps. It has no ADB allocated, yet the buffer sizing managed to satisfy the clock skew constraint for all modes.

The reasons that only a fraction of ADBs were replaced with ADIs is that: 1) while ADBs are located at both leaf and non-leaf positions, only the ones at the leaf positions are subject to ClkWaveMin and may be replaced with ADIs, and 2) since ADIs have longer signal propagation delay than that of ADBs, during feasible type computation, ADIs were mostly pruned. As shown in Fig. 4, there are three inverters in an ADI that causes ADIs to have longer delays than ADBs. Currently, in our implementation, the first inverter that directly receives the incoming clock signal has nMOS width of 45 nm which is the smallest feature size allowed by the technology. Thus, it is impossible to reduce the ADI size. Instead, the designer might choose to have larger ADBs so that the signal propagation delay is balanced. However, this will cause ADBs to occupy larger area, and in this experiment, we chose to have the unbalanced ADBs and ADIs.

VIII. CONCLUSION

We addressed a new problem of clock buffer polarity assignment combined buffer sizing to overcome a set of fundamental limitations of the conventional clock buffer polarity assignment methods, which are: 1) the unawareness of the signal delay differences to the leaf clock buffering elements; 2) the ignorance of the effect of the current fluctuation of non-leaf clock buffering elements on the total peak current waveform; and 3) the inability of supporting low-power digital designs with multiple power modes. We showed that our proposed polarity assignment algorithms can be used effectively to reduce the peak current caused by clock signal propagation in both the diverse high-speed designs [38] with single power mode and the low power designs with multiple power modes. However, some circuits experienced V_{DD}/Gnd noise degradations despite the improvement of peak current noise. Since V_{DD}/Gnd noises are the cause of propagation delay degradation of circuits, future works should consider not only the clock trees but also the power distribution networks so that the voltage noises can be improved.

REFERENCES

[1] D. Joo and T. Kim, "Wavemin: A fine-grained clock buffer polarity assignment combined with buffer sizing," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2011, pp. 522–527.

[2] S. Chowdhury and J. Barkatullah, "Estimation of maximum currents in MOS IC logic circuits," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 9, no. 6, pp. 642–654, Jun. 1990.

[3] L. Chen, M. Marek-Sadowska, and F. Brewer, "Buffer delay change in the presence of power and ground noise," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 11, no. 3, pp. 461–473, Jun. 2003.

[4] K. Tang and E. Friedman, "Simultaneous switching noise in on-chip CMOS power distribution networks," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 4, pp. 487–493, Aug. 2002.

[5] N. H. Weste and D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 3rd ed. Reading, MA, USA: Addison-Wesley, 2005.

[6] C. J. Alpert, A. Devgan, and S. T. Quay, "Buffer insertion with accurate gate and interconnect delay computation," in *Proc. ACM/IEEE Design Autom. Conf.*, Jun. 1999, pp. 479–484.

[7] J. Cong, C. Koh, and K. Leung, "Simultaneous buffer and wire sizing for performance and power optimization," in *Proc. IEEE/ACM Int. Symp. Low Power Electron. Design*, Aug. 1996, pp. 271–276.

[8] C. C. N. Chu and M. D. F. Wong, "An efficient and optimal algorithm for simultaneous buffer and wire sizing," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 18, no. 9, pp. 1297–1304, Sep. 1999.

[9] I.-M. Liu, T.-L. Chou, A. Aziz, and M. D. F. Wong, "Zero-skew clock tree construction by simultaneous routing, wire sizing and buffer insertion," in *Proc. ACM Int. Symp. Phys. Design*, Apr. 2000, pp. 33–38.

[10] T. Okamoto and J. Cong, "Buffered Steiner tree construction with wire sizing for interconnect layout optimization," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1996, pp. 44–49.

[11] J.-L. Tsai, T.-H. Chen, and C.-P. Chen, "Zero skew clock-tree optimization with buffer insertion/sizing and wire sizing," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 23, no. 4, pp. 565–572, Apr. 2004.

[12] K. Wang, Y. Ran, H. Jiang, and M. Marek-Sadowska, "General skew constrained clock network sizing based on sequential linear programming," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 24, no. 5, pp. 773–782, May 2005.

[13] Y.-S. Su, W.-K. Hon, C.-C. Yang, S.-C. Chang, and Y.-J. Chang, "Value assignment of adjustable delay buffers for clock skew minimization in multi-voltage mode designs," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2009, pp. 535–538.

[14] Y.-S. Su, W.-K. Hon, C.-C. Yang, S.-C. Chang, and Y.-J. Chang, "Clock skew minimization in multi-voltage mode designs using adjustable delay buffers," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 29, no. 12, pp. 1920–1930, Dec. 2010.

[15] K.-Y. Lin, H.-T. Lin, and T.-Y. Ho, "An efficient algorithm of adjustable delay buffer insertion for clock skew minimization in multiple dynamic supply voltage designs," in *Proc. IEEE Asia-South Pacific Design Autom. Conf.*, Jan. 2011, pp. 825–830.

[16] K.-H. Lim, D. Joo, and T. Kim, "An optimal allocation algorithm of adjustable delay buffers and practical extensions for clock skew optimization in multiple power mode designs," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 32, no. 3, pp. 392–405, Mar. 2013.

[17] J. Kim, D. Joo, and T. Kim, "An optimal algorithm of adjustable delay buffer insertion for solving clock skew variation problem," in *Proc. 50th Annu. Design Autom. Conf.*, Jun. 2013, pp. 90:1–90:6.

[18] S. Hu and J. Hu, "Unified adaptivity optimization of clock and logic signals," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2007, pp. 125–130.

[19] V. Khandelwal and A. Srivastava, "Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation," in *Proc. ACM Int. Symp. Phys. Design*, Apr. 2007, pp. 11–18.

[20] J.-L. Tsai and L. Zhang, "Statistical timing analysis driven post-silicon-tunable clock-tree synthesis," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2005, pp. 575–581.

[21] E. Takahashi, Y. Kasai, M. Murakawa, and T. Higuchi, "A post-silicon clock timing adjustment using genetic algorithms," in *Proc. Symp. VLSI Circuits*, Jun. 2003, pp. 13–16.

[22] Y.-T. Nieh, S.-H. Huang, and S.-Y. Hsu, "Minimizing peak current via opposite-phase clock tree," in *Proc. IEEE/ACM Design Autom. Conf.*, Jun. 2005, pp. 182–185.

[23] R. Samanta, G. Venkataraman, and J. Hu, "Clock buffer polarity assignment for power noise reduction," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2006, pp. 558–562.

[24] P.-Y. Chen, K.-H. Ho, and T. Hwang, "Skew-aware polarity assignment in clock tree," *ACM Trans. Design Autom. Electron. Syst.*, vol. 14, no. 2, pp. 31:1–31:17, Apr. 2009.

[25] Y. Ryu and T. Kim, "Clock buffer polarity assignment combined with clock tree generation for power/ground noise minimization," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 2008, pp. 416–419.

[26] M. Kang and T. Kim, "Clock buffer polarity assignment considering the effect of delay variations," in *Proc. Int. Symp. Quality Electron. Design*, Mar. 2010, pp. 69–74.

[27] H. Jang, D. Joo, and T. Kim, "Buffer sizing and polarity assignment in clock tree synthesis for power/ground noise minimization," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 30, no. 1, pp. 96–109, Jan. 2011.

[28] J. Lu and B. Taskin, "Clock buffer polarity assignment considering capacitive load," in *Proc. Int. Symp. Quality Electron. Design*, Mar. 2010, pp. 765–770.

[29] J. Lu and B. Taskin, "Clock buffer polarity assignment with skew tuning," *ACM Trans. Design Autom. Electron. Syst.*, vol. 16, no. 4, pp. 49:1–49:22, Oct. 2011.

[30] J. Lu and B. Taskin, "Clock tree synthesis with XOR gates for polarity assignment," in *Proc. IEEE Annu. Symp. VLSI*, 2010, pp. 17–22.

[31] J. Lu, Y. Teng, and B. Taskin, "A reconfigurable clock polarity assignment flow for clock gated designs," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 6, pp. 1002–1011, Jun. 2012.

[32] Z. Tarapata, "Selected multicriteria shortest path problems: An analysis of complexity, models and adaptation of standard algorithms," *Int. J. Appl. Math. Comput. Sci.*, vol. 17, pp. 269–287, Jun. 2007.

[33] A. Warburton, "Approximation of Pareto optima in multiple-objective, shortest-path problems," *Oper. Res.*, vol. 35, pp. 70–79, Feb. 1987.

[34] M. Ehrgott, *Multicriteria Optimization*. Secaucus, NJ, USA: Springer-Verlag New York, 2005.

[35] Nangate Inc., (2009) *Open Cell Library v2009 07* [Online]. Available: http://www.nangate.com/openlibrary

[36] Q. Zhu, *Power Distribution Network Design for VLSI*. New York, NY, USA: Wiley, 2004.

[37] T.-Y. Kim and T. Kim, "Clock tree synthesis for TSV-based 3-D IC designs," *ACM Trans. Design Autom. Electron. Syst.*, vol. 16, no. 4, pp. 48:1–48:21, Oct. 2011.

[38] B.-H. Park, Y. Kim, B.-D. Kim, T. Hong, S. Kim, and J. K. Lee, "High performance computing: Infrastructure, application, and operation," *J. Comput. Sci. Eng.*, vol. 6, no. 4, pp. 280–286, Dec. 2012.

Deokjin Joo (S'11) received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2009 and 2011, respectively. He is currently pursuing the Ph.D. degree at the School of Electrical Engineering and Computer Science, Seoul National University.

His current research interests include clock tree synthesis for low power and thermal resilient design.

Taewhan Kim (SM'08) received the B.S. degree in computer science and statistics and the M.S. degree in computer science from Seoul National University, Seoul, Korea, and the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, IL, USA, in 1993.

He is currently a Professor with the School of Electrical Engineering and Computer Science, Seoul National University. After graduation, he was with Lattice Semiconductor Corporation and Synopsys, Inc., San Jose, CA, USA, for six years, specializing in design automation tool development. He has published over 170 technical papers in international journals and conferences, including the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION, ACM TODAES, DAC, ICCAD, and ASPDAC. His current research interests include computer-aided design of integrated circuits ranging from the architectural synthesis through physical designs, specifically focusing on power, thermal, noise, reliability, and 3-D integrated circuit design issues.

Dr. Kim is the Co-Editor-in-Chief of the *International Journal of Computing Science and Engineering*.