Variation-Aware Clock Network Design Methodology for Ultralow Voltage (ULV) Circuits

Xin Zhao, *Student Member, IEEE,* Jeremy R. Tolbert, *Student Member, IEEE,* Saibal Mukhopadhyay, *Member, IEEE,* and Sung Kyu Lim, *Senior Member, IEEE*

***Abstract*—This paper presents a design methodology for robust and low-energy clock networks for ultralow voltage (ULV) circuits. We show that both clock slew and skew play important roles in achieving high maximum operating frequency ($F_{\max}$) and low clock energy in ULV circuits. In addition, clock networks in ULV circuits are highly sensitive to process variations. We propose a variation-aware methodology that controls both clock skew and slew to maximize $F_{\max}$ and minimize clock power. In addition, we implement dynamic programming (DP)-based ULV clock routing and buffering methods (deferred merging and embedding) for deterministic and statistical conditions. Experimental results show that our clock network design method achieves lower energy (more than 20% savings) at comparable or even higher $F_{\max}$ compared with the existing methods.**

***Index Terms*—Clock network design, ultralow voltage, variation aware.**

I. Introduction

Ultralow voltage (ULV) circuits, where the supply voltage is around or even below the threshold voltage of transistors, have emerged as an attractive option for ultralow-power digital computing. Many ultralow-power battery-operated applications with a stringent energy budget can benefit from operating in ULV, such as biological monitoring systems, radio-frequency identification devices, wireless sensor networks, and others. Although speed is not the primary goal, high-frequency operation has been demonstrated in the range of tens to hundreds of megahertz [1] with ULV circuits.

The clock network is a global interconnect that provides clock signal to flip-flops (FFs) for synchronization. This network contributes a significant amount of power consumption and dedicates the overall system performance. In the ULV domain, clock slew plays a major role in robustness of the clock network. This is because in ULV, buffer delay and FF timings (setup, clock-to-q, and hold) are strong functions of clock slew [2], [3]. In addition, ULV circuits are more sensitive to process and environmental variations, especially threshold voltage random variability caused by the random dopant fluctuation and process variations [3]–[5]. Fig. 1 shows Vgs–Ids curves of the NMOS and PMOS from 45-nm predictive technology model [6], where the nominal value of threshold voltage (Vt) is 621 mV and −575 mV (see the light curves), respectively. The supply voltage (Vdd) is set to 550 mV that is around the Vt. The threshold voltage variation with 1-σ swing of 10 mV are considered in 1000 Monte Carlo SPICE simulation. The results are shown in the two groups of dark curves indicating the Vt variation. In the ULV domain, the device current depends exponentially on threshold voltage. Hence, threshold voltage variability can cause a significant variation in clock skew and slew, thereby degrading the timing margins. As a result, the operating frequency is usually reduced to ensure correct operation. Therefore, the clock design methodology for ULV circuits requires: 1) efficient control on *both* clock slew and skew; 2) robustness in the presence of variations; 3) consideration of frequency target; and 4) low-energy clock operation.

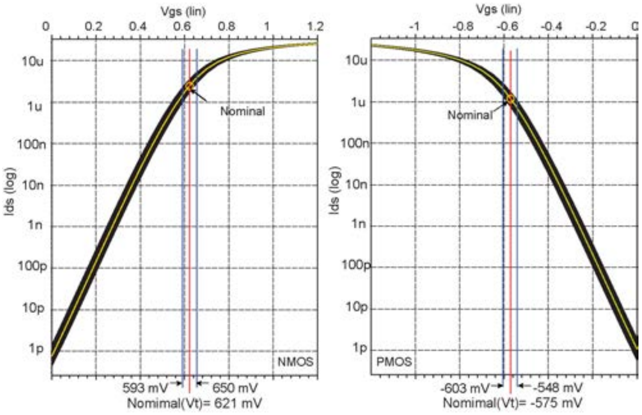


Fig. 1. Vgs–Ids curves of NMOS and PMOS, where the nominal Vt is 621 mV and −575 mV (light curves), respectively. Our design method is for ULV clock network design. The supply voltage is set to 550 mV and the 1-σ Vt variation is 10 mV. One thousand Monte Carlo simulation results are shown in two groups of dark curves indicating the Vt variation.

In this paper, we develop a variation-aware methodology for robust and low-energy clock network design for ULV circuits. The contributions of this paper are as follows.

1) We present comprehensive studies based on extensive experimental results that show the impact of clock

Manuscript received September 22, 2011; revised January 12, 2012; accepted February 18, 2012. Date of current version July 18, 2012. This work was supported by the National Science Foundation, under CCF-0917000, the National Science Foundation Graduate Research Fellowship, under Grant DGE-0644493, and the Semiconductor Research Corporation, under Task ID 1836.075. This paper was recommended by Associate Editor D. Sylvester.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: xinzhao@gatech.edu; jeremy.r.tolbert@gmail.com; saibal.mukhopadhyay@ece.gatech.edu; limsk@ece.gatech.edu).

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.2012.2190825

skew and clock slew control on power consumption, performance, and variation tolerance in ULV circuits.

2) We develop a variation-aware ULV clock network design methodology. For clock skew management, we construct the routing topology and insert buffers to minimize the delay differences among the clock paths under both nominal and statistical conditions. We also show how to efficiently control clock slew bound at each sink under both nominal and statistical conditions.
3) We implement robust and low-energy clock tree synthesis algorithms for ULV clock networks, which are based on dynamic programming and deferred merging and embedding techniques, so called DP+DME algorithm. Our algorithms generate and save multiple solutions to achieve minimum clock energy while satisfying given upper bounds for clock slew and skew.
4) Experimental results show that our clock design method efficiently controls the clock skew and slew in both nominal and statistical conditions and constructs ULV clock networks with low clock energy at a high maximum operating frequency. We outperform state-of-the-art ULV clock-routing methods [2], [7] in terms of performance and energy under both nominal and statistical conditions.

The remainder of this paper is organized as follows. Section II presents the summary of related work and its limitations. Section III presents comprehensive study on the clock skew and clock slew variability control impact on clock performance and energy. Section IV formulates the ULV clock synthesis problem and presents our clock design methodology. Section V presents our DP and DME-based clock synthesis algorithm for both nominal and statistical conditions. Experimental results and extensive discussions are presented in Sections VI and VII, respectively, and we conclude in Section VIII.

II. Background

The history on ULV clock network design is very brief. Existing works focus on minimizing either clock slew or clock skew but not both. Tolbert *et al.* [8] pointed out the importance of clock slew control for the reliability of subthreshold circuits. They developed a subthreshold buffer model that considered the impact of slew on delay. They also discussed reliable clock system design [8] that controls clock slew while minimizing the energy. In addition, it was shown that traditional clock synthesis methods for superthreshold circuits are not feasible. However, they did not consider the impact of clock skew on performance and energy. Seok *et al.* [7] compared buffered and unbuffered H-tree (UnBH) topologies for various technology, circuit sizes, and supply voltages. For UnBH, clock skew can be well controlled to near zero, but clock slew may become worse. To counter the slew effect, a larger driver is required, which results in a significant energy penalty. On the other hand, the buffered H-tree (BufH) may have unbalanced loadings at buffers depending on the buffer level, which can also cause large skew variability. In addition, both of these works are primarily circuit-level studies and did not present any design method for clock network synthesis. As large-scale ULV circuits are emerging, methodology is becoming essential for automated synthesis of robust (low-slew and low-skew) and low-energy clock network.

DP-based buffer insertion is one of the common methods in superthreshold circuits for either timing optimization or clock routing, which can be classified into wirelength-driven, timing-driven, and maximum slew-constraint-driven with power or area minimization [9]–[14]. The basic flow is as follows. The multiple feasible buffering solutions with certain costs are stored and propagated, and a global optimal solution is determined later. Most of the existing work fix slew violations by upper bounding the buffer loading. However, this is not sufficient for ULV clock synthesis. Since buffer delay heavily depends on the input slew, bounding slew in a certain range does not guarantee well management on delay, nor the resulting clock skew. Moreover, buffer insertion still takes the merits of repairing the slew, but at the same time leads to more randomness. Extra cares should be paid on both slew and skew variability control in ULV clock network design.

III. Motivation

A. Clock Slew and Skew Impact on Timing of ULV Circuits

Our work is motivated by the impact of both clock skew and slew on the cycle time. The schematic in Fig. 2 shows a generic logic path composed of fan-out-4 NAND gates between two registers. It includes the clock-to-q ($T_{\text{CLK-Q}}$), the setup time (T_{setup}), the combination logic path delay (T_{logic}), and the difference of the clock arrival times (Skew). The minimum cycle time ($T_{\min}$) and the maximum clock frequency ($F_{\max} = 1/T_{\min}$) for the above system is as follows:

$$T_{\min} = T_{\text{CLK-Q}}(\text{FF1}) + T^{\max}_{\text{Logic}} + T_{\text{setup}}(\text{FF2}) + \text{Skew} \qquad (1)$$

where $T^{\max}_{\text{Logic}}$ is the maximum logic path delay. This circuit operates at 550 mV supply voltage, where nominal threshold voltage for NMOS and PMOS is 621 mV and −575 mV, respectively. First, a larger skew increases $T_{\min}$, and thus decreases $F_{\max}$. Second, clock slew directly alters the timing metrics $T_{\text{CLK-Q}}$ and T_{setup}, leading to a long cycle time. Clock slew could vary the hold time as well [3], [8].

Fig. 3 shows the path length (=N) versus $F_{\max}$ trend for four cases in ULV circuits: 1) Optimal: 1.5 ns clock slew and 0 ns clock skew; 2) Skew-only: clock skew (CLK1 arrives later than CLK2) is 5% of the optimal period, and the clock slew is 1.5 ns; 3) Slew-only: 10 ns clock slew and 0 ns skew; and 4) Slew+skew: slew around 10 ns and skew is 5% of the optimal period. We observe that due to the additional amount of skew (or slew), Case 2 (or Case 3) obtains lower operating frequency than Case 1 at each logic length. In Case 4, where both skew and slew degradation are applied, $F_{\max}$ is degraded from 12% to 20%, depending on the length of the logic path. This clearly shows that both clock skew and slew affect the operating frequency significantly in ULV circuits.

Fig. 4 shows detailed timing metrics with respect to the slew in ULV circuits. The slew impact on T_{logic}, $T_{\text{CLK-Q}}$, and T_{setup} is shown in Fig. 4(a), where the path length is fixed to

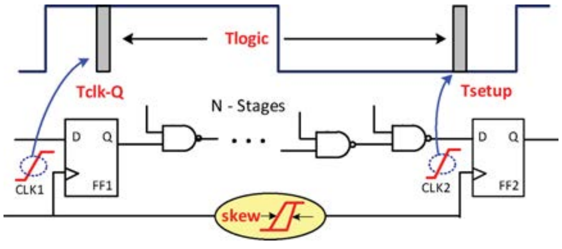


Fig. 2. Data path in synchronous ULV circuit.

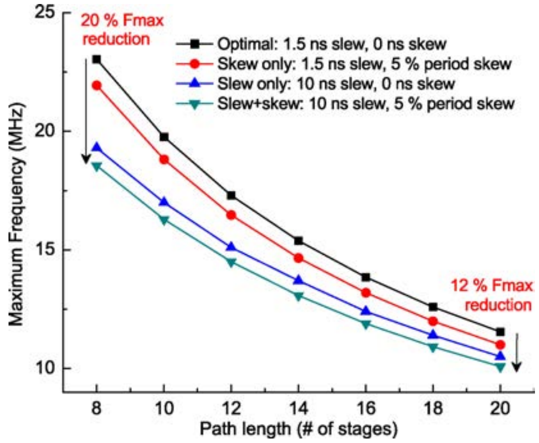


Fig. 3. Impact of the logic path length on $F_{\max}$ of ULV circuit in 550 mV Vdd under four combinations of clock skew and slew.

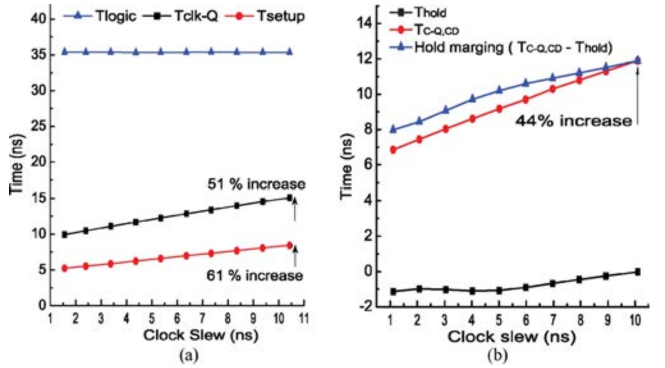


Fig. 4. Impact of clock slew on setup and hold-time constraints. (a) Slew impact on T_{logic}, $T_{\text{CLK-Q}}$, and T_{setup} in ULV circuit, where the path length is set to 10. (b) Slew impact on T_{hold}, $T_{\text{C-Q,CD}}$, and hold margin.

10. We observe that both $T_{\text{CLK-Q}}$ and T_{setup} increase by 51% and 61% if the slew increases from 1.5 ns to 10.5 ns. Note that data slew is recovered within the FFs. The logic delay remains unaffected by the clock slew change. As ULV systems target higher frequencies (i.e., as the logic path reduces), the timing metrics ($T_{\text{CLK-Q}}$ and T_{setup}) become a large portion of the cycle time and thus cannot be ignored. This clearly demonstrates the importance of clock slew on ULV circuit performance.

The impact of clock slew on hold time (T_{hold}), clock-to-q contamination delay ($T_{\text{C-Q,CD}}$), and hold margin ($T_{\text{C-Q,CD}}$-T_{hold}) are shown in Fig. 4(b). For small clock slew, the hold time is negative and large in magnitude. As the slew increases, the hold-time transitions from a large negative number to a smaller negative number (eventually arriving at zero). The small negative number means that the requirement of the hold edge has been pulled to a time closer to the clock edge. Therefore, as the slew increases, the hold time increases as well. Much like the worse-case clock-to-q, the clock-to-q contamination delay is directly dependent on the slew distribution. For an increasing clock slew, the variation of contamination delay can be as much as 44% larger. Note that the hold-time constraint can be expressed as follows:

$$\text{Skew} - T_{\text{logic, CD}} < T_{\text{C-Q,CD}} - T_{\text{hold}} \qquad (2)$$

where $T_{\text{logic, CD}}$ is the logic contamination delay. The hold margin in Fig. 4(b) increases from 8 ns to 12 ns with input slew increases from 1 ns to 10 ns.

B. Impact of Buffer Placement on ULV Clock Wires

In ULV circuits, the effect of interconnect resistance is negligible [2], [7] due to the large resistance of the driving buffer. This also allows the interconnect to be modeled as a lumped capacitance. Seok *et al.* [2] pointed out that adding a repeater in the middle of a capacitive interconnect does not help reduce the delay. However, we observe that a buffered interconnect is advantageous to improve slew under comparable power consumption. This means that to achieve the same slew, using many but smaller buffers consumes lower power than using one large buffer to drive the entire clock network.

Fig. 5 demonstrates the impact of the buffer count, buffer location, and wirelength on the sink slew, B2 input slew, and the src-to-sink delay in ULV circuits. The interconnect length varies from 500 μm to 3000 μm. We compare two cases: 1) using a large driver (B0) for a long interconnect, and 2) using two small buffers (B1 and B2) to drive the interconnect. The slew and src-to-sink delay in the two-buffer case are normalized to the one-buffer case results.

While keeping B1 at the source location, we first notice that moving B2 toward the sink causes the sink slew to decrease up to 70–80% and outperforms the one-buffer case as shown in Fig. 5(a). The improved sink slew presents similar trends under different lengths. Second, when B2 moves toward the sink, it also leads to increased slew at the input of B2 as shown in Fig. 5(b). This is mainly due to the increased loadings of B1. The high input slew may result in larger delay variation and skew at the sink, thus requires careful buffer placement policy for skew reduction. Third, the src-to-sink delay, shown in Fig. 5(c), increases faster for the longer interconnect, when B2 gets closer to the sink. This is mainly due to the larger buffer delay caused by interconnect parasitic capacitance. Note that both cases consume similar power. This shows that a buffered clock tree has the potential to improve slew and achieve lower power, but care must be taken to control clock skew in ULV circuits.

C. Buffer Delay Dependency on Input Slew

In ULV, buffer delay heavily depends on the input slew and loading. A comparison between superthreshold and ULV circuits is shown in Fig. 6. The superthreshold design (in blue) operates at 1.1 V with 1 ns clock period; the ULV design uses supply voltage of 0.55 V and 100 ns clock period. Both the input slew and buffer delay are normalized to the

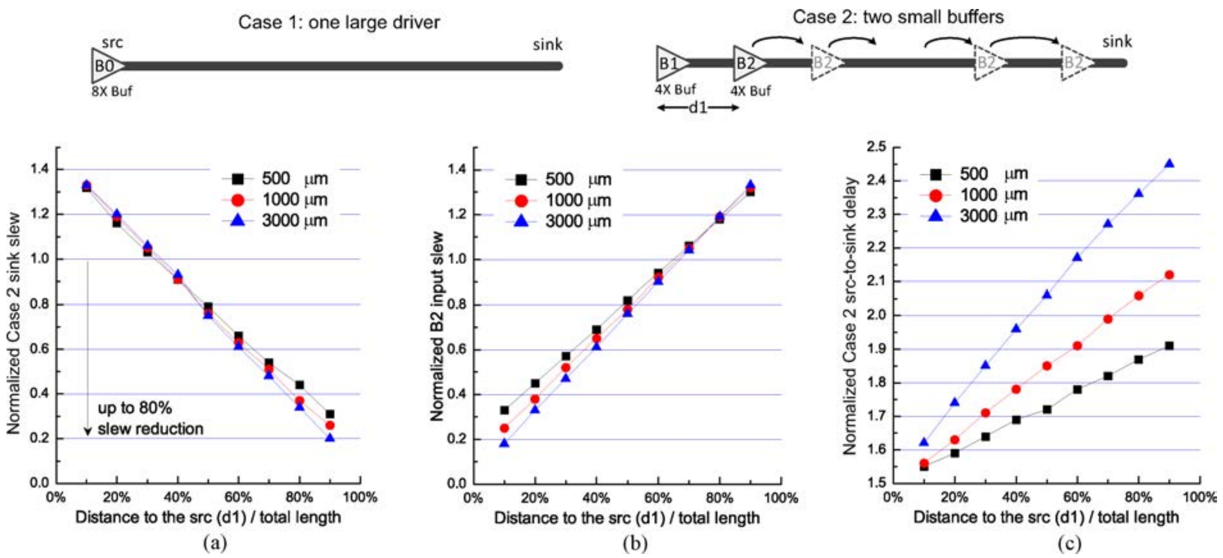


Fig. 5. Impact of the buffer count, buffer location, and wirelength on the (a) sink slew, (b) B2 input slew, and the (c) src-to-sink delay in ULV circuits. The total length varies from 500, 1000, and 3000 μm. Slew and delay results of Case 2 are normalized to Case 1.

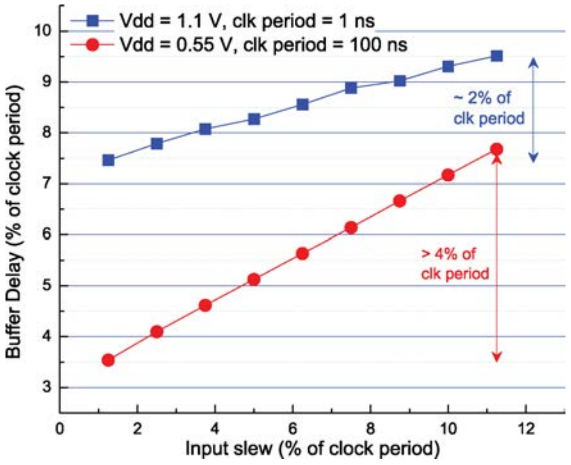


Fig. 6. Buffer delay dependency with input slew. The buffer delay and input slew are normalized to the corresponding clock period.

corresponding clock period. It is well known that buffer delay is sensitive to the input slew. We find out that this dependency becomes stronger (in larger slope) in ULV domain. It means that an input slew variation could result in much larger delay variation therefore more skew variability. Upper bounding the buffer loading, which has been widely applied in either clocking or timing/slew-driven buffering, is not sufficient to handel the resulting delay variations thus corresponding clock skew. Under the same amount of relative input slew changes, the buffer delay in ULV is 2× large of that in superthreshold. Therefore, obtaining a well-controlled clock skew requires the clock synthesis method take extra care on the slew impact on clock delay variations.

D. Effect of Variation

The ULV circuit is much sensitive to the process and environmental variations. Especially, since drain current exponentially depends on threshold voltage, this exponential impact through the threshold voltage variability is a dominant source affecting the functionality and performance. A larger variation of clock slew or skew in ULV circuit increases the risk of timing violations, and would result in large degradation in $F_{\max}$, and thus requires efforts on controlling variabilities.

The impact of slew and skew variability on the cycle time can be observed from the following equation:

$$T^0_{\min} + \Delta T_{\min} = T^0_{\text{CLK-Q}} + T^{\max}_{\text{Logic}} + T^0_{\text{setup}} + \text{skew}^0 + \delta(\text{skew} + \eta(\text{slew})) \qquad (3)$$

where the nominal values are with a head of 0. The clock skew variation $\delta(\text{skew})$ directly contributes to the cycle time change $\Delta T_{\min}$. Let $\eta()$ denote the function representing the slew impact on timing metrics ($T_{\text{CLK-Q}}$, T_{setup}). A slew variation will cause both timing metrics vary. As a result, both slew and skew variability leads to the $F_{\max}$ variation.

Based on the measurement data and observations from Drego *et al.* [15], threshold voltage variation in ULV circuits can be modeled as random variables with spatial uncorrelated. In this paper, we follow this statement and mainly focus on the randomness from the threshold voltages.

IV. ULV Clock Network Design Methodology

The ULV clock tree synthesis (CTS) problem is formulated as follows; given a set of clock sinks, a target sink slew, and an upper bound for clock skew, the *ULV-CTS* is to construct a buffered clock tree such that: 1) clock slew at sink node is under the given constraint; 2) clock skew is under the given constraint; 3) clock power is minimized; and 4) the clock skew and clock slew variability are controlled. Given various slew and skew upper bound constraints, we can generate clock networks with different $F_{\max}$.

We develop a ULV clock buffering and routing method that consists of two steps: 1) abstract tree generation that determines the hierarchical connection among the sink nodes,

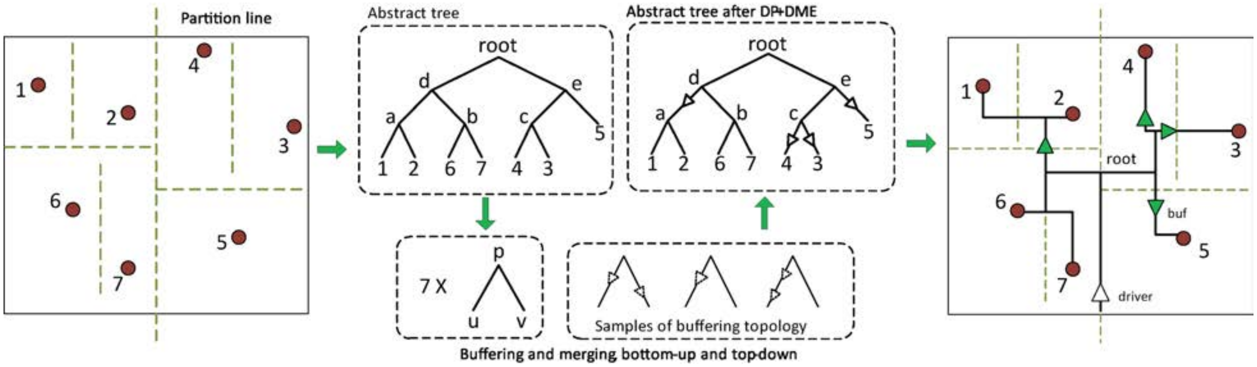


Fig. 7. Illustration of our ULV clock network design flow. We bipartition the given sink set based on their coordinates, and construct an abstract tree that indicates the hierarchical connection among the clock nodes and the routing sequences. We then propagate the slew, delay, buffering, and routing solutions from sinks to the root node recursively, and obtain a set of candidate solutions for each clock node. Last, we select the legal low-power solution for the root note and propagate it to its children in a top-down fashion to construct the final clock topology.

intermediate nodes, and the root node, and 2) clock routing and buffer insertion that decide the clock wire topology, buffer count, and buffer placement.

Our method resembles the DME routing [16], but we have added the following enhancements to handle clock skew, slew, and various variation sources in ULV clock networks.

1) *Efficient control of both skew and slew for F_{max}*: We introduce upper bounds for target skew and slew in the clock network design. By tightening or relaxing the bounds, we can generate clock networks with various $F_{\max}$ and energy. For instance, a tighter slew bound results in higher $F_{\max}$ but requires more clock buffers and thus higher clock energy; a tighter skew bound leads to higher $F_{\max}$ at the cost of high energy, etc. We limit the node capacitance for buffer outputs to satisfy the slew bound. In addition, we determine the routing topology, buffer placement, and buffer count for skew control.
2) *Low clock energy*: Note that clock energy depends on the clock wirelength and buffer count. Our buffer insertion process stores multiple buffering solutions and selects the optimal one with the lowest clock energy under the slew and skew constraints.
3) *Robustness to variations*: We use look-up tables (LUTs) for buffer timing in statistical condition. We compute and constrain the weighted skew that consists of the standard deviation of the path delay and the covariance between buffers.

Fig. 7 shows an illustration of our methodology named DP+DME. In abstract tree generation, a classical technique so-called method of means and medians [17] is used to decide how to merge clock nodes in an hierarchical fashion. Given an abstract tree, the buffering and routing problem can be divided into many subproblems with a similar structure: merging child nodes to their parent node.

The DP+DME algorithm consists of two steps: generating feasible solutions bottom-up and selecting an optimal solution top-down. First, it visits each node bottom-up from the sink nodes to the root node based on the given abstract tree. It generates a set of feasible solutions with respect to various input slew values for each node. Then DP+DME propagates the solution toward an upper level, until the solution set for the root node is obtained. Section V describes the technical details of buffer insertion in DP+DME algorithm.

Second, we select the optimum solution for the root node that has the lowest power and satisfies the skew–slew constraints, and recursively decide the solution for the children in a top–down fashion. During this process, each node in the abstract tree is assigned a solution, and the clock network is constructed correspondingly.

In Fig. 7, seven iterations of merging and buffering are performed on a given pair of children u and v and their parent node p. Each iteration consists of exploring feasible solutions for node p in a bottom-up fashion and selecting the optimal solution for each node in a top–down fashion.

In addition, we created LUTs based on SPICE simulations to obtain buffer timing during ULV clock synthesis. The input parameters for LUT generation consists of the input slew and the loading capacitance of a buffer, where both parameters are assigned with a certain range. Given an input slew value and a loading capacitance, SPICE Monte Carlo simulation is then applied with threshold voltage variations. As a result, we can obtain the deterministic values of buffer delay and output slew, standard deviation of buffer delay and output slew, as well as the covariance between buffer delays. Note that the ranges of input parameters for LUT generation determine the simulated distribution of buffer timing (e.g., output slew and buffer delay). In this paper, the input slew is swept from 0 ns to 25 ns with a step of 1.25 ns, and the range of loading capacitance is set from 0 fF to 300 fF with a step of 20 fF. Several scripting files have been created to automatically generate the LUTs.

V. ULV Clock Synthesis Algorithms

A. *Deterministic DP+DME*

The buffering and merging problem for the entire abstract tree can be divided into many subproblems with similar structure of merging child nodes to their parent node [see Fig. 8(a)]. We assign a solution set Γ_p for each merging node p in the abstract tree, where each solution $\gamma_p \in \Gamma_p$ is a 6-tuple, $\gamma_p = \{S, M, D_{\min}, D_{\max}, C, P\}$:

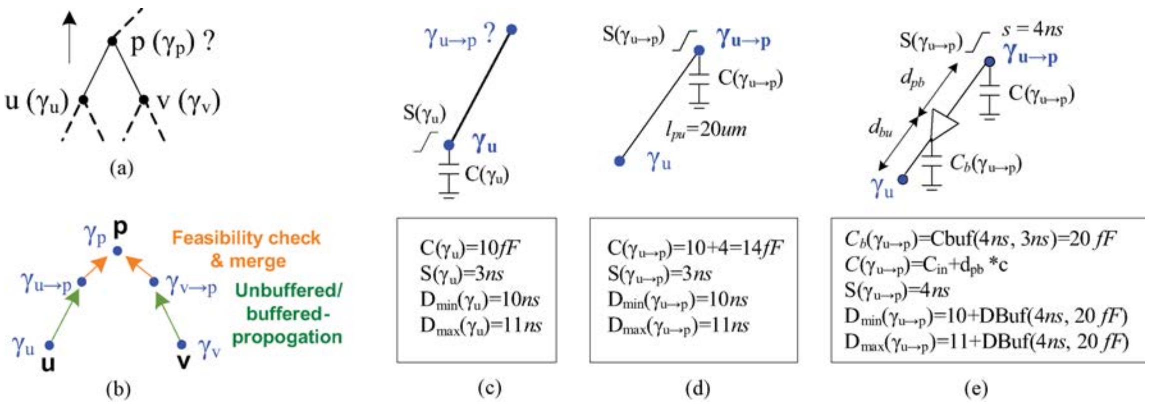


Fig. 8. Illustration of determining a solution of node p (γ_p) by merging nodes u (γ_u) and v (γ_v) using deterministic DP+DME. The DP+DME is composed of many subproblems with the similar structure as in (a), where solution γ_p is determined by first unbuffered or buffered propagating γ_u (γ_v) to the solution $\gamma_{u\to p}$ ($\gamma_{v\to p}$) and then applying feasibility check and merging to γ_p as shown in (b). Given a solution of γ_u as in (c), the solution $\gamma_{u\to p}$ can be obtained by unbuffered propagation in (d) or buffered propagation in (e). The per-unit-length wire capacitance is set to 0.2 fF/μm, and the buffer loading CBuf() and delay DBuf() are obtained in the LUT.

1) S is the slew at node p;
2) M is the style of merging its child nodes;
3) $D_{\min}$ and $D_{\max}$ are the minimum and the maximum delay from node p to its sink nodes;
4) C is the loading capacitance at node p;
5) P is the cost of the corresponding merging style, which is the power consumption in this problem.

We use $S(\gamma_p)$, $D_{\min}(\gamma_p)$, $D_{\max}(\gamma_p)$, $C(\gamma_p)$, and $P(\gamma_p)$ to represent each element in γ_p.

In bottom-up buffering solution construction, it is impossible to obtain the accurate slew due to its top–down propagation property. To accurately acquire the slew and its affecting delay, we enumerate a set of feasible slew value s for each node that $s \in \{s_1, s_2, \ldots, s_n\}$, where $s_{\text{i}} - s_{\text{i-1}} = g$, s_1 and s_n are the lower or upper bound, and g is the granularity.

Without loss of generality, considering merging node u and node v to node p, we first propagate the solution of γ_u (γ_v) to node p, and obtain the solution $\gamma_{u\to p}$ ($\gamma_{v\to p}$). We then determine the solution for node p by merging $\gamma_{u\to p}$ and $\gamma_{v\to p}$ to γ_p with a feasibility check [see Fig. 8(b)]. Depending on, if buffers are inserted along the edge pu (pv), the propagation is classified into buffered or unbuffered-propagation.

In unbuffered-propagation, no buffer is along edge pu. The elements in solution $\gamma_{u\to p}$ are determined as

$$S(\gamma_{u\to p}) = S(\gamma_u) \tag{4}$$

$$D_{\min}(\gamma_{u\to p}) = D_{\min}(\gamma_u) \tag{5}$$

$$D_{\max}(\gamma_{u\to p}) = D_{\max}(\gamma_u) \tag{6}$$

$$C(\gamma_{u\to p}) = C(\gamma_u) + c \times l_{pu} \tag{7}$$

$$P(\gamma_{u\to p}) = P(\gamma_u) \tag{8}$$

where l_{pu} is the length of edge pu and c is the per unit length capacitance of wires. Equations (4)–(8) mean that interconnect has negligible effect on delay and slew propagation. Note that if the unbuffered-propagation merging passes the feasibility check from (18) to (21), the $l_{pu} = d_{pu}$ as in (25). Fig. 8(d) shows an example of determining the solution $\gamma_{u\to p}$ given the solution of γ_u in Fig. 8(c).

In buffered-propagation, a buffer is inserted along edge pu. A set of feasible slew values are assigned at node p. For each slew value s, solution $\gamma_{u\to p}$ for node p are obtained as

$$S(\gamma_{u\to p}) = s \tag{9}$$

$$C_{\text{b}}(\gamma_{u\to p}) = \text{CBuf}(S(\gamma_{u\to p}), S(\gamma_u)) \tag{10}$$

$$D_{\min}(\gamma_{u\to p}) = \text{DBuf}(S(\gamma_{u\to p}), C_{\text{b}}(\gamma_{u\to p})) + D_{\min}(\gamma_u) \tag{11}$$

$$D_{\max}(\gamma_{u\to p}) = \text{DBuf}(S(\gamma_{u\to p}), C_{\text{b}}(\gamma_{u\to p})) + D_{\max}(\gamma_u) \tag{12}$$

$$C(\gamma_{u\to p}) = C_{\text{in}} + c \times d_{pb} \tag{13}$$

$$P(\gamma_{u\to p}) = P(\gamma_u) + \text{PBuf}(S(\gamma_{u\to p}), C_{\text{b}}(\gamma_{u\to p})). \tag{14}$$

We first calculate the buffer loading on edge pu as $C_{\text{b}}(\gamma_{u\to p})$ in (10). The DBuf(), CBuf(), and PBuf() denote LUT operation to obtain the buffer delay, loading, and power. For instance, CBuf($S(\gamma_{u\to p})$, $S(\gamma_u)$) obtains the buffer loading given input slew $S(\gamma_{u\to p})$ and output slew $S(\gamma_u)$. DBuf($S(\gamma_{u\to p})$, $C_{\text{b}}(\gamma_{u\to p})$) denotes the buffer delay given input slew $S(\gamma_{u\to p})$ and loading $C_{\text{b}}(\gamma_{u\to p})$. C_{in} is the input capacitance of the buffer. Fig. 8(e) shows an example of determining the solution $\gamma_{u\to p}$ if a buffer is inserted along edge pu. $\gamma_{v\to p}$ follows the similar equations from (4) to (14).

When merging $\gamma_{u\to p}$ and $\gamma_{v\to p}$ into γ_p

$$D_{\min}(\gamma_p) = \min(D_{\min}(\gamma_{u\to p}), D_{\min}(\gamma_{v\to p})) \tag{15}$$

$$D_{\max}(\gamma_p) = \max(D_{\max}(\gamma_{u\to p}), D_{\max}(\gamma_{v\to p})) \tag{16}$$

$$C(\gamma_p) = C(\gamma_{u\to p}) + C(\gamma_{v\to p}). \tag{17}$$

A feasible solution should satisfy all of the following:

$$S(\gamma_{u\to p}) = S(\gamma_{v\to p}) = s \tag{18}$$

$$C_{\text{b}}(\gamma_{u\to p}) \geq C(\gamma_u), \text{ if edge pu is buffered} \tag{19}$$

$$C_{\text{b}}(\gamma_{v\to p}) \geq C(\gamma_v), \text{ if edge pv is buffered} \tag{20}$$

$$\text{Skew}(\gamma_p) = D_{\max}(\gamma_p) - D_{\min}(\gamma_p) \leq \text{skewBnd}. \tag{21}$$

If the conditions from (18) to (21) are satisfied, we save γ_p into Γ_p as a candidate, and the remaining elements in γ_p are obtained as

$$S(\gamma_p) = s \tag{22}$$

$$P(\gamma_p) = P(\gamma_{u\to p}) + P(\gamma_{v\to p}). \tag{23}$$

Let L be the minimum merging distance between nodes u and v. In buffered-propagation, let $d_{\mathrm{b}u}$ ($d_{\mathrm{b}v}$) denote the distance between a buffer to node u (v), which is obtained as

$$d_{\mathrm{b}u} = \frac{C_{\mathrm{b}}(\gamma_{u\rightarrow p}) - C(\gamma_u)}{c}. \tag{24}$$

For the unbuffered-propagation cases, $d_{\mathrm{b}u}$ ($d_{\mathrm{b}v}$) is set to zero. Then the merging distance d_{pu} between nodes p and u (d_{pv} between p and v) is determined as

$$d_{pu} = \max(\frac{L - d_{\mathrm{b}u} - d_{\mathrm{b}v}}{2}, 0) + d_{\mathrm{b}u} \tag{25}$$

$$d_{pv} = \max(\frac{L - d_{\mathrm{b}u} - d_{\mathrm{b}v}}{2}, 0) + d_{\mathrm{b}v}. \tag{26}$$

Correspondingly, the merging style M stores the merging distances of d_{pu}, d_{pv}, $d_{\mathrm{b}u}$, and $d_{\mathrm{b}v}$, and a merging segment for node p following the classic DME procedure is obtained.

For a sink node p, we choose the feasible slew value s under the given slew bound $slewBnd$. For each $s \in \{s_1, \ldots, s_n\}$, we create a solution γ_p with $S(\gamma_p) = s$, $D_{\min}(\gamma_p) = D_{\max}(\gamma_p) = P(\gamma_p) = 0$, and $C(\gamma_p) = C_{\mathrm{in}}^{\mathrm{FF}}$, where $C_{\mathrm{in}}^{\mathrm{FF}}$ is the input capacitance of the FF.

B. Pruning the Solutions

Considering that each child node has a number of candidate solutions, with a combination of the feasible slew for node p, the solution space would be dramatically expanded and lose the efficiency. However, with fewer solutions for node p, it is more difficult to derive a good solution. In addition, the slew granularity g also affects the runtime and final quality. The finer the g, the more candidates for node p, the higher possibility to consume less power, but the longer runtime. To guarantee a high quality with reasonable runtime, we define a control parameter K, which is the maximum number of solutions for each feasible slew. We increasingly sort the solutions based on each slew s and keep the first K solutions having the smallest power for each feasible slew. We discuss the efficiency of using K and g and their impact on quality and runtime in the experimental result Section VI-C.

C. Statistical DP+DME

We implement the statistical DP+DME algorithm that efficiently controls the skew and slew variability during clocking and buffering procedure. The statistical DP+DME involves many augments on delay and skew randomness. The solution structure is extended as a 7-tuple, $\gamma_p = \{S, M, D_{\min}, D_{\max}, C, P, \sigma_D\}$:

1) S is the sample mean of slew at node p;
2) $D_{\min}$ and $D_{\max}$ are the sample mean of the minimum and the maximum delay from node p to its sink nodes;
3) σ_D denotes the largest standard deviation of the path delay from node p to the sinks;
4) M is the style of merging its child nodes;
5) C is the loading capacitance at node p;
6) P is the cost of the corresponding merging style, which is the power consumption in this problem.

The statistical DP+DME utilizes variation-aware LUTs, which include the sample mean and standard deviation of delay and slew with respect to the input slew and loading capacitance, as well as the covariance between buffer delays.

Most of the elements in solution γ_p follow the similar propagation policy as the deterministic DP+DME method. The $\sigma_D(\gamma_{u\rightarrow p})$ is updated as follows.

In the case of buffered-propagation

$$\sigma_D^2(\gamma_{u\rightarrow p}) = \sigma_D^2(\gamma_u) + \mathrm{VBuf}(S(\gamma_{u\rightarrow p})C_{\mathrm{b}}(\gamma_{u\rightarrow p})) + \mathrm{Cov}. \tag{27}$$

We calculate the covariance between buffers along the path pu and each of the buffers connecting to it. We then add the largest covariance value (Cov) into the $\sigma_D(\gamma_{u\rightarrow p})$. The VBuf() denotes the variance of the buffer along path pu with input slew $S(\gamma_{u\rightarrow p})$ and loading $C_{\mathrm{b}}(\gamma_{u\rightarrow p})$ obtained from the LUT.

After merging $\gamma_{u\rightarrow p}$ and $\gamma_{v\rightarrow p}$ to γ_p, $\sigma_D(\gamma_p)$ is obtained as follows:

$$\sigma_D(\gamma_p) = \max(\sigma_D(\gamma_{u\rightarrow p}), \sigma_D(\gamma_{v\rightarrow p})). \tag{28}$$

The statistical skew depends on not only the differences among the average path delays but also the delay variance. We use a weighted sum (SSkew) to represent this dependency

$$\mathrm{SSkew}(\gamma_p) = \alpha \times (D_{\max}(\gamma_p) - D_{\min}(\gamma_p)) + \beta \times \sigma_D(\gamma_p) \tag{29}$$

where weights α=1, β=2 are used to express the worst-case skew variability, and the feasibility check for the skewBnd constraint (21) is updated as

$$\mathrm{SSkew}(\gamma_p) \leq \mathrm{skewBnd}. \tag{30}$$

For simplicity, the weighted equation (29) is used to control the skew variability. Our method is flexible to employ more sophisticated method as [18]. In addition, slew variability can be directly improved by applying a tighter slew constraint.

VI. Simulation and Discussions

A. Experimental Settings

Our clock design method has been implemented using C++/STL on Linux. We focus on the 45-nm technology ULV clock network design. The per-unit-length wire resistance and capacitance are $0.1\,\Omega/\mu$m and $0.2\ fF/\mu$m, respectively. Our clock network uses 6× buffers. The nominal values of threshold voltage (Vt) is 621 mV and −575 mV for NMOS and PMOS, respectively. The supply voltage (Vdd) is set to 550 mV that is around the Vt and the 1-σ threshold voltage swing is 10 mV.

All experimental results are reported from SPICE simulation, including clock skew, slew, and energy per cycle. We use LUT-based buffer modeling during clock network construction and evaluate the designs using SPICE: 1) we extract the layout information of the FFs; 2) apply our ULV clock synthesis method to construct a buffered clock tree; and 3) extract a clock netlist for SPICE simulation. In statistical condition, the Vt uncertainty is modeled as random variables with spatial uncorrelated [15]. We apply 1000 Monte Carlo simulations for each design and report μ+2σ skew and slew like the existing work [7]. We created five benchmark circuits: three

TABLE I
INFORMATION OF BENCHMARK AND ENERGY PER CYCLE (pJ)

					Energy per Cycle	
ckt	Function	#Gates	#FFs	Area	Logic+Wire	H-Tree
				(μm×μm)	w/o clock	(% of tot)
ckt1	FIR filter	3823	148	331×315	2.1	1.1 (34%)
ckt2	Multiplier	3952	320	376×412	4.4	2.5 (36%)
ckt3	FIR filter	16 185	499	664×664	6.3	4.1 (39%)
ckt4	FIR filter	30 833	619	857×924	13.4	5.5 (29%)
ckt5	Quick sort	4828	768	518×546	7.6	4.8 (39%)

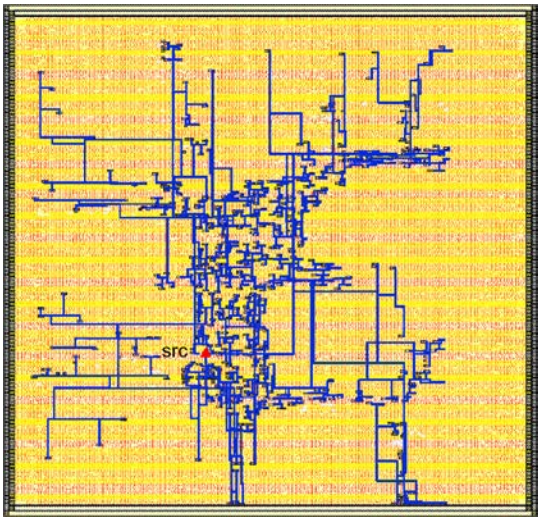


Fig. 9. Sample layout of a clock network for a FIR filter with 619 clock sinks and 24 937 μm wirelength.

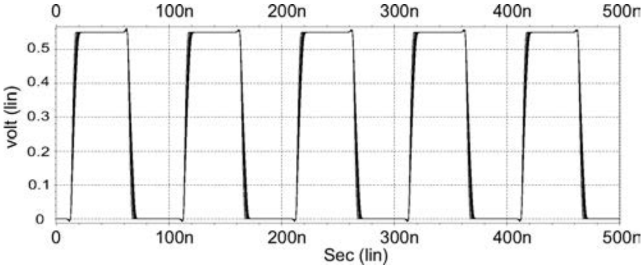


Fig. 10. Clock waveforms for a FIR filter with 148 FFs at 10 MHz frequency. Clock skew is 1.2 ns, and it distributes from 3.1 ns to 5.3 ns with an average value of 4.7 ns.

finite impulse response (FIR) filters, a multiplier, and a design implementing quick sort as shown in Table I.

Fig. 9 shows a clock network for our FIR filter, which is seen in front of the logic cells and highlighted FFs. The clock network has 619 clock sinks, a total wirelength of 24 937 μm, and die area of 857 μm×924μm.

Fig. 10 shows the clock waveforms from SPICE for a FIR filter (ckt1) in ULV operation. This superimposes 148 waveforms from the FFs. The clock skew is 1.2 ns, which can be observed by the width of waveforms at 50% Vdd. The clock slew values for all the sinks are from 3.1 ns to 5.3 ns with an average of 4.7 ns.

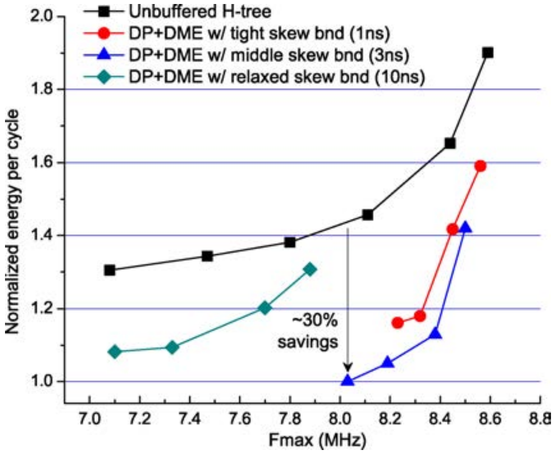


Fig. 11. $F_{\max}$ versus energy per cycle in nominal condition for ckt1.

B. Impact of Slew and Skew Bounds on F_{max} and Energy

Fig. 11 shows the impact of skew and slew upper bounds on $F_{\max}$ and clock energy per cycle for ckt1. We show four curves of nominal results: one is for UnBHs and the other three are generated by our DP+DME clock synthesis technique. The UnBH takes an advantage of skew minimization. However, it needs a large driver for entire network to ensure small clock slew at sink nodes. Therefore, we upsize the driver (an inverter chain) to improve the $F_{\max}$. As a result, the overall clock energy of the UnBH increases significantly. In the case of DP+DME, we present three groups of results, where clock networks in each group are designed under the same skew bounds (1 ns, 3 ns, and 10 ns) but different slew bounds (from 3 ns to 8 ns).

First, Fig. 11 demonstrates the tradeoff between high $F_{\max}$ and low clock energy, i.e., design for a higher $F_{\max}$ consumes more clock energy. In each DP+DME curve under the same skew constraint, a tighter slew bound improves $F_{\max}$. Meanwhile, the clock energy per cycle increases due to more buffers that are inserted for tighter slew control. Second, the design of 3 ns skew bound consumes lower energy than 1 ns skew bound. This is mainly because the 3 ns skew constraint reserves more feasible solutions during clock network construction, which helps to obtain a low-energy clock network. However, using relaxed skew bound of 10 ns cannot hold this benefit, since it allows larger clock skew thus requires more buffers for tighter slew to reach a similar $F_{\max}$ as 1 ns or 3 ns skew bound. Third, compared with the UnBH design targeting at skew minimization only, our method achieves up to 30% energy reduction in the frequency range from 8.0 MHz to 8.4 MHz. This is because a buffered clock tree has shorter wirelength and a smaller driver than the UnBH design. We note that a higher $F_{\max}$ target will shorten the energy gap between ours and the UnBH. Under the relaxed skew bound, the clock energy increases slower than that of tight skew bound as the slew bound increases. But, the relaxed skew bound results in lower $F_{\max}$ than other two curves. Thus, we see that by controlling both the skew and slew bounds, we design a low-energy clock network for a given target $F_{\max}$ more effectively.

TABLE II

DETERMINISTIC CLOCK ROUTING RESULTS UNDER VARIOUS SLEW AND SKEW BOUNDS INCLUDING WIRELENGTH (μm), BUFFER COUNT, SKEW (ns), SLEW (ns), AND CLOCK ENERGY PER CYCLE (pJ)

Slew	Skew	WL	#Bufs	Skew		Min Slew		Max Slew		Avg. Slew		Energy Per Cycle	
Bound	Bound			LUT	SPICE	LUT	SPICE	LUT	SPICE	LUT	SPICE	SPICE	Ratio
3	0.5	6678	40	0.27	0.38	2.5	2.5	3.0	2.9	2.7	2.6	1.17	1.00
	1.0	5268	40	0.81	0.78	1.5	1.5	3.0	2.9	2.5	2.4	1.09	0.93
	2.0	5159	39	1.56	1.70	2.0	1.9	3.0	2.9	2.5	2.4	1.08	0.92
5	0.5	7761	28	0.50	0.51	3.5	3.5	5.0	4.8	4.1	4.0	1.14	0.97
	1.0	5864	27	1.00	0.98	3.0	3.0	5.0	4.8	4.0	3.8	1.01	0.86
	2.0	5127	25	1.79	1.92	2.0	2.0	5.0	4.8	3.8	3.7	0.94	0.80
7	0.5	6104	16	0.50	0.50	5.5	5.4	6.5	6.3	5.6	5.4	0.91	0.78
	1.0	5501	16	0.75	0.73	5.0	4.9	6.5	6.3	5.3	5.2	0.87	0.74
	2.0	4968	16	1.26	1.23	4.0	4.0	6.5	6.3	5.1	5.0	0.84	0.72

Both LUT and SPICE simulation results are included in clock timing.

C. Deterministic Clock Routing Results

1) *Efficiency of Slew and Skew Control:* Table II shows clock synthesis results under various upper bounds for the slew and skew using deterministic DP+DME algorithm. We show the wirelength (μm), buffer count, clock skew (ns), slew distribution (min/max/avg) (ns), and clock energy per cycle (pJ). First, LUT-based modeling approach is trustworthy to estimate the buffer timing during clock synthesis. By comparing the value from LUT and SPICE simulation, the differences are within 0.2 ns. Second, for each design, both clock skew and slew from SPICE simulation are under the given upper bounds. This demonstrates the efficiency of our deterministic DP+DME algorithm in controlling both clock skew and slew for ULV circuits.

2) *Runtime Versus Quality Tradeoff:* Fig. 12 shows the comparisons for normalized clock energy per cycle and runtime using two slew granularity g={0.5 ns, 0.25 ns} and the maximum allowed solution count K for each feasible slew value, where K={2, 5, 10, 20}, correspondingly. The resulting clock energy and overall runtime are closely related to these two settings in the DP algorithm. A lower energy clock design can be obtained by either using finer slew granularity or storing more intermediate solutions. However, at the same time, a significant amount of runtime has to be paid to find an optimal solution in DP+DME algorithm. We choose 0.5 ns slew granularity and allow maximum ten solutions for each feasible slew in the DP+DME for the consideration of both low-energy clock design and reasonable runtime.

D. Statistical Versus Deterministic Methods

Fig. 13 shows the efficiency of our variation-aware methodology. We compare the deterministic and statistical DP+DME techniques in terms of the μ+2σ skew, the worst-case skew, and the clock energy. There are two major differences between these two methods: 1) statistical DP+DME uses variation-aware LUTs, which include the sample mean and standard deviation of delay and slew with respect to the input slew and loading capacitance, as well as the covariance between buffer delays, and 2) we employ the statistical skew bound to cope with the control on the variation-caused skew uncertainty. We observe that both μ+2σ and the worst-case skew are efficiently reduced by using the statistical method with marginal energy penalty.

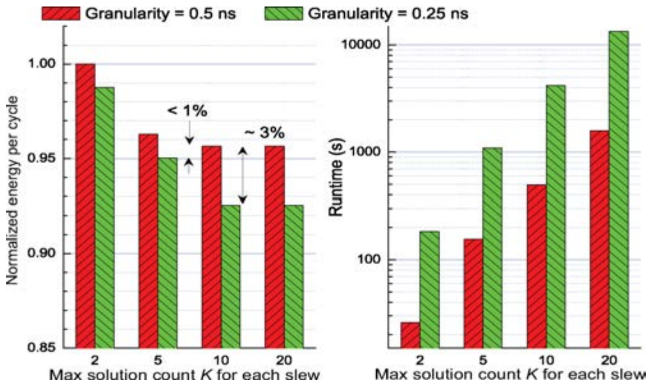


Fig. 12. Impact of the slew granularity g and the maximum number of solutions K for each feasible slew value on the energy per cycle and runtime, where g={0.5 ns, 0.25 ns} and K={2, 5, 10, 20}.

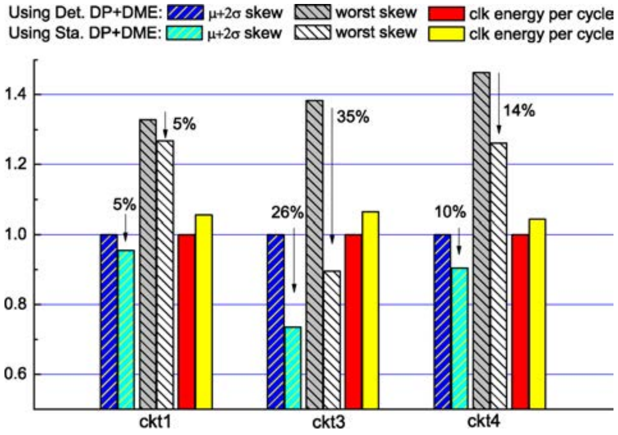


Fig. 13. Comparisons between deterministic and statistical DP+DME techniques in μ+2σ skew and the worst-case skew.

E. Comparison With Existing Works

For ULV circuit clock tree routing, Tolbert *et al.* [2] focused on clock slew minimization only for low power and reliability. Seok *et al.* [7] compared UnBH and BufH topologies in ULV operation under different supply voltages, technologies, and

TABLE III
COMPARISONS BETWEEN LSHS METHOD [2] AND OUR DP+DME ALGORITHM IN NOMINAL AND STATISTICAL CONDITIONS

ckt	Method	In Nominal Condition					In Statistical Condition						Reduction %			
		WL	#B	Skew	Slew	EPC	Skew				MaxSlew		Skew	EPC	Skew	Skew
					Max		μ	σ	μ+2σ	Worst	μ	σ	Nom.		μ+2σ	Worst
ckt1	LSHS	4968	31	16.7	3.8	0.99	17.7	1.8	21.2	23.5	4.9	0.6				
	Our DP+DME	5725	32	1.9	3.8	1.04	4.5	2.0	8.5	13.7	4.9	0.6	88.7	−5.7	59.7	41.6
ckt2	LSHS	10 760	65	14.7	3.9	2.11	16.8	1.6	19.9	23.3	5.2	0.6				
	Our DP+DME	11 061	66	3.1	3.8	2.13	6.4	1.9	10.2	13.3	5.1	0.6	79.0	−1.1	48.6	43.1
ckt3	LSHS	19 497	109	15.2	3.9	3.53	17.8	1.6	20.9	25.2	5.6	0.6				
	Our DP+DME	20 719	113	3.0	3.8	3.64	7.4	1.7	10.8	16.5	5.5	0.6	80.3	−3.1	48.3	34.7
ckt4	LSHS	26 603	127	15.4	3.9	4.43	19.1	2.2	23.6	28.3	5.8	0.6				
	Our DP+DME	27 794	131	1.7	3.8	4.54	8.5	2.1	12.8	17.1	5.7	0.6	89.1	−2.4	45.8	39.5
ckt5	LSHS	21 101	144	15.2	3.9	4.64	18.8	1.8	22.4	31.2	5.8	0.6				
	Our DP+DME	23 703	151	2.6	3.8	4.86	8.1	1.8	11.7	15.5	5.6	0.6	83.2	−4.8	47.6	50.3

We show the wirelength (μm), skew (ns), slew (ns), and energy per cycle (EPC in pJ).

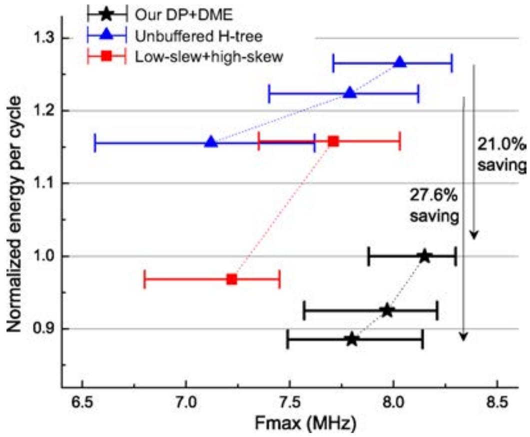


Fig. 14. $F_{\max}$ distribution versus energy per cycle in statistical condition for ckt1. We compare three methods: our DP+DME technique, UnBHs [7], and LSHS [2].

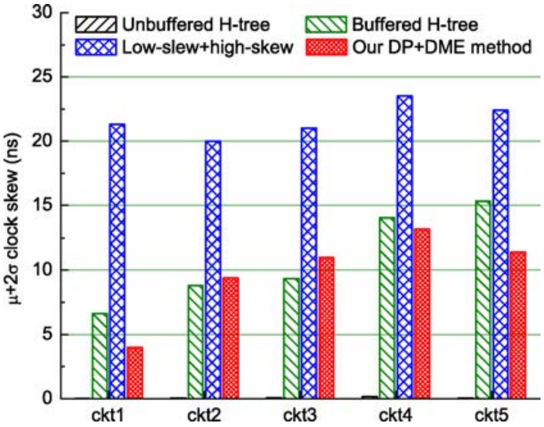


Fig. 15. Comparison of UnBH, BufH [7], LSHS [2], and our DP+DME methods in $\mu + 2\sigma$ skew.

design sizes. Our proposed method is to generate buffered clock tree by controlling both slew and skew for low-clock energy and high $F_{\max}$. To obtain the results of [2], we apply tight slew bounds and relaxed skew bounds (LSHS) as suggested by the authors. This is suggested because using relaxed skew bound leads to the lowest clock energy per cycle for a specific slew value. To reproduce the results of [7], we construct symmetric UnBH and BufH, and design an inverter chain as a clock driver for the entire H-tree. We take into account the energy from the clock driver as well as clock buffers, sinks load, and interconnect.

We first show a detail comparison between our DP+DME algorithm and the LSHS [2] in Table III; both methods focus on buffered clock tree design in ULV clock systems. Our DP+DME algorithm is assigned a tighter skew bound but the same slew bound as the LSHS method. Table III shows the comparisons in both nominal and statistical conditions. The LSHS method takes the advantages of less clock energy, but suffers large amount of skew and its variability. Our algorithm reduces the skew by up to 89.1% in nominal value, 59.7% in 2σ skew, and 50.3% in the worst-case skew, and sacrifices a small amount of energy per cycle around 1–5.7%.

Fig. 14 shows the $F_{\max}$ versus clock energy per cycle tradeoff in statistical condition for ckt1. We compare UnBH [7], LSHS method [2], and our DP+DME technique. We observe the tradeoff between high $F_{\max}$ and low clock energy in all these results. To improve $F_{\max}$ in UnBHs, we upsize the driver. We also tighten the slew bound from 7 ns to 4 ns for the LSHS method [2]. In our DP+DME, we try several combinations of slew and skew bounds and report the results. Compared with UnBHs [7], our method achieves more than 21.0–27.6% energy savings around the 8 MHz $F_{\max}$. Compared with the LSHS method [2], we obtain more than 13% $F_{\max}$ improvement with marginal energy penalty (around the normalized energy of 1.0) or more than 20% energy savings at comparable $F_{\max}$ of 8 MHz.

Figs. 15–18 show the comparisons among UnBH [7], BufH [7], LSHS method [2], and our statistical DP+DME algorithm for benchmark ckt1–ckt5. These comparisons include the $\mu + 2\sigma$ skew, $\mu + 2\sigma$ maximum slew, $F_{\max}$ distribution, and normalized clock energy per cycle. The LSHS method uses 4 ns slew bound.[1]

First, the UnBH design achieves the minimum skew because of its symmetric structure and negligible interconnect resistance (see Fig. 15). Our DP+DME algorithm achieves

[1] The slew bound is to limit the mean value of the maximum slew. The resulting $\mu + 2\sigma$ slew will exceed the bound. We observe that a tighter slew bound helps to narrow the overall slew distribution.

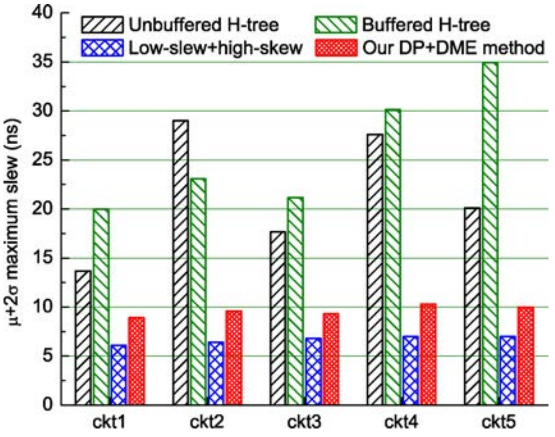


Fig. 16. Comparison of UnBH, BufH [7], LSHS [2], and our DP+DME methods in $\mu + 2\sigma$ maximum slew.

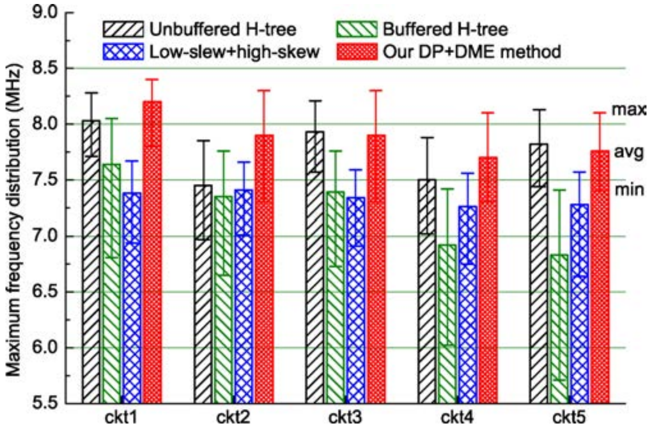


Fig. 17. Comparison of UnBH, BufH [7], LSHS [2], and our DP+DME methods in $F_{\max}$ distribution.

comparable or even better skew compared with BufH design, whereas the LSHS method results in the largest skew degradation.

Second, the LSHS method shows the advantage in minimizing the variation-aware slew (see Fig. 16). Both unbuffered and BufHs have high slew. This large slew degradation can be reduced by upsizing the driver or the internal clock buffers with inevitable penalty of extreme larger clock energy.

Third, because we efficiently control both clock skew and slew, our method obtains a high $F_{\max}$ (see Fig. 17) and outperforms the other three methods in achieving the lowest clock energy (i.e., 10–40% energy savings, see Fig. 18). Fig. 19 shows the variation-aware $F_{\max}$ distribution of the four methods for ckt1. Our clock network achieves the highest sample mean $F_{\max}$ of 8.2 MHz and narrower standard deviation of 0.08 MHz. This is comparable with UnBH result and much better than other two methods.

In addition, the LSHS method achieves lower energy than UnBHs at the cost of low $F_{\max}$. The BufH consumes the highest energy but in low $F_{\max}$, which matches the observations [7]. Therefore, we conclude that a simultaneous management of variation-aware slew and skew proves to be an efficient way to obtain a low-energy and robust clock network targeting at a high $F_{\max}$ in ULV circuits.

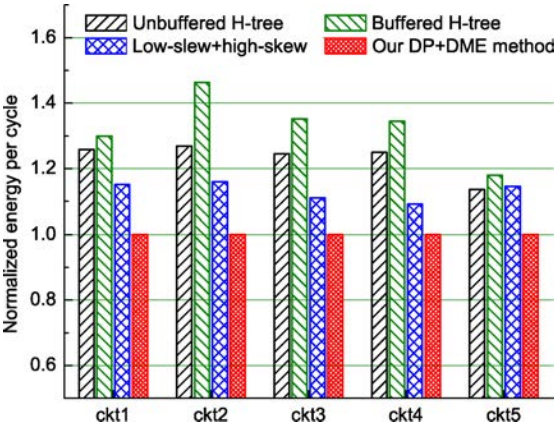


Fig. 18. Comparison of UnBH, BufH [7], LSHS [2], and our DP+DME methods in normalized energy per cycle.

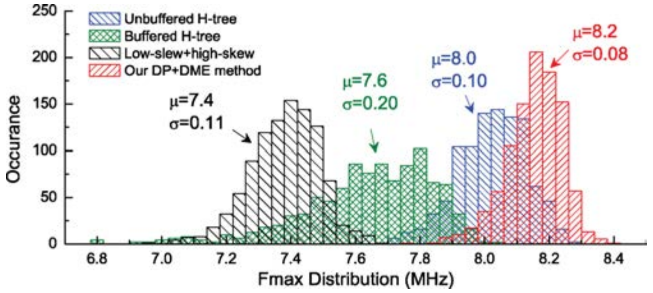


Fig. 19. Comparison of $F_{\max}$ distribution among UnBH, BufH, LSHS, and our DP+DME for ckt1.

VII. Extensive Discussions

A. Results of 130-nm Technology

We extend our discussion on 130-nm technology. The comparisons of UnBH, BufH, low-slew and high-skew (LSHS), and our DP+DME method are shown in Table IV for ckt1. The threshold voltage of NMOS and PMOS from 130-nm PTM model are 378 mV and −4321 mV, respectively. The supply voltage is set to 300 mV with 1-σ threshold voltage swing as 10 mV. The per-unit-length wire resistance and capacitance are $0.028\,\Omega/\mu$m and $0.267\ fF/\mu$m, respectively. The interconnect parasitic resistance is much lower in 130 nm than in 45-nm technology.

We have consistent observations that: 1) our DP+DME algorithm achieves comparable Fmax distribution and 19% less power than UnBH; 2) compared with the LSHS and BufH methods, we obtain both higher Fmax and more than 30% power reduction; and 3) UnBH results in near-zero skew and LSHS method has lowest slew. But our DP+DME method controls both slew and skew variability.

B. Impact of Wire Thickness

In this section, we discuss the impact of wire thickness on ULV clock performance. Note that our DP+DME algorithm is able to consider various interconnect parasitics or using different technology. To reduce the parasitic resistance, we use thin-global interconnect with 0.8 μm thick, 0.6 μm wide, and 0.4 μm space, which refers to the metal layers [19]. The corresponding per-unit-length wire resistance and capacitance

TABLE IV

COMPARISONS OF UnBH, BufH, LSHS, AND OUR DP+DME METHODS IN 130-NM TECHNOLOGY

Method	Skew (ns)		Slew (ns)		Fmax (MHz)		EPC (pJ)	
	μ	σ	μ	σ	μ	σ	SPICE	Ratio
UnBH	0.00	0.00	4.08	0.59	18.17	0.10	0.50	1.19
BufH	0.96	0.36	3.89	0.58	17.89	0.18	0.55	1.31
LSHS	5.99	0.59	1.61	0.17	16.70	0.17	0.57	1.34
Our	1.11	0.26	2.12	0.22	18.20	0.10	0.42	1.00

TABLE V

COMPARISONS OF UnBH, BufH, LSHS, AND OUR DP+DME METHODS USING THICK WIRES

Method	Skew (ns)		Slew (ns)		Fmax (MHz)		EPC (pJ)	
	μ	σ	μ	σ	μ	σ	SPICE	Ratio
UnBH	0.00	0.00	11.36	1.71	8.08	0.10	1.02	1.25
BufH	3.54	1.33	14.37	2.27	7.69	0.19	1.06	1.29
LSHS	16.98	1.69	4.75	0.62	7.43	0.10	0.95	1.16
Our	2.73	0.60	6.73	0.85	8.18	0.08	0.82	1.00

TABLE VI

COMPARISONS OF UnBH, BufH, LSHS, AND OUR DP+DME METHODS FOR A SMALL CIRCUIT

Method	Skew (ns)		Slew (ns)		Fmax (MHz)		EPC (pJ)	
	μ	σ	μ	σ	μ	σ	SPICE	Ratio
UnBH	0.00	0.00	9.01	1.36	8.22	0.09	0.79	1.05
BufH	4.76	0.91	10.92	1.51	7.81	0.13	0.83	1.12
LSHS	5.96	0.85	4.89	0.67	8.08	0.07	0.82	1.10
Our	2.50	0.58	6.44	0.88	8.22	0.08	0.75	1.00

are $0.046\,\Omega/\mu$m and $0.177\ fF/\mu$m, respectively. The comparisons of UnBH, BufH, LSHS, and our DP+DME method are shown in Table V for ckt1.

We have similar observations that our algorithm can result in less energy and high Fmax than other three methods. The interconnect delay is in the order of picoseconds, whereas the buffer delay is in the order of nanoseconds. Our algorithm can efficiently control the clock skew and slew by balancing the buffer delay and managing the loadings and input slew. Therefore, using less parasitic resistance does not affect the performance of our algorithm.

C. Impact of Circuit Size

We have created a small circuit to discuss the impact of circuit size, which has $120\,\mu$m $\times 130\,\mu$m footprint area, and 159 clock sinks. The comparisons of four methods are shown in Table VI. Our method achieves comparable Fmax to the UnBH and 5% power reduction. For smaller circuits, it is much easier for the UnBH to obtain lower slew, thus does not cause large power overhead as in large circuits. In addition, our DP+DME still outperforms other two methods in achieving large Fmax and less power. Therefore, our method is more efficient in large-scale ULV circuits for high Fmax and low energy.

D. Impact of Vdd Variations

Our algorithm uses LUTs to store the statistical delay and slew with respect to input slew and loading capacitance, as well as the covariance between buffer delays. Therefore, our algorithm is able to consider other types of variations. We have included both supply voltage variations (with nominal value of 550 mV and 1-σ swing of 10 mV) and threshold voltage variations. The comparisons of UnBH, BufH, LSHS, and our DP+DME methods are shown in Table VII. Our method is efficient to obtain the highest Fmax than other three methods and achieve 16–31% energy reduction.

TABLE VII

COMPARISONS OF UnBH, BufH, LSHS, AND OUR DP+DME METHODS UNDER BOTH POWER SUPPLY AND THRESHOLD VOLTAGE VARIATIONS

Method	Skew (ns)		Slew (ns)		Fmax (MHz)		EPC (pJ)	
	μ	σ	μ	σ	μ	σ	SPICE	Ratio
UnBH	0.01	0.00	13.03	4.19	7.98	0.25	1.08	1.27
BufH	1.93	0.62	14.05	4.49	7.81	0.29	1.12	1.31
LSHS	17.66	5.62	4.07	1.31	7.44	0.37	0.99	1.16
Our	1.32	0.42	5.61	1.81	8.35	0.15	0.85	1.00

VIII. CONCLUSION

In this paper, we studied the methodology of low-energy and robust clock network design for ULV circuits. We observed that both clock slew and skew need to be accurately controlled to ensure a high maximum operating frequency ($F_{\max}$) in ULV circuits. We showed that buffer insertion is an important mean to achieve this goal. We proposed a variation-aware methodology that controls both clock skew and slew to maximize $F_{\max}$ and minimize clock power. In addition, we implemented the DP-based ULV clock routing and buffering methods (DP+DME) in both deterministic and statistical conditions. Experimental results showed that we are able to construct clock trees that are variation aware, low power, and high performance ($F_{\max}$) while satisfying the given slew and skew constraints for ULV operations.

REFERENCES

[1] B. Paul, A. Raychowdhury, and K. Roy, "Device optimization for digital subthreshold logic operation," *IEEE Trans. Electron Devices*, vol. 52, no. 2, pp. 237–247, Feb. 2005.

[2] J. R. Tolbert, X. Zhao, S. K. Lim, and S. Mukhopadhyay, "Analysis and design of energy and slew aware subthreshold clock systems," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 30, no. 9, pp. 1349–1358, Sep. 2011.

[3] N. Verma, J. Kwong, and A. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE Trans. Electron Devices*, vol. 55, no. 1, pp. 163–174, Jan. 2008.

[4] B. Zhai, S. Hanson, D. Blaauw, and D. Slyvester, "Analysis and mitigation of variability in subthreshold design," in *Proc. Int. Symp. Low Power Electron. Des.*, 2005, pp. 20–25.

[5] R. Vaddi, S. Dasgupta, and R. Agarwal, "Device and circuit co-design robustness studies in the subthreshold logic for ultralow-power applications for 32 nm CMOS," *IEEE Trans. Electron Devices*, vol. 57, no. 3, pp. 654–664, Mar. 2010.

[6] *Predictive Technology Model (PTM)* [Online]. Available: http://ptm.asu.edu

[7] M. Seok, D. Blaauw, and D. Sylvester, "Robust clock network design methodology for ultra-low voltage operations," *IEEE Trans. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 2, pp. 120–130, Jun. 2011.

[8] J. R. Tolbert and S. Mukhopadhyay, "Accurate buffer modeling with slew propagation in subthreshold circuits," in *Proc. Int. Symp. Qual. Electron. Des.*, 2009, pp. 91–96.

[9] G. E. Tellez and M. Sarrafzadeh, “Minimal buffer insertion in clock trees with skew and slew rate constraints,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 4, pp. 333–342, Apr. 1997.

[10] C. Albrecht, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky, “On the skew-bounded minimum-buffer routing tree problem,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 7, pp. 937–945, Jul. 2003.

[11] L. van Ginneken, “Buffer placement in distributed RC-tree networks for minimal elmore delay,” in *Proc. IEEE Int. Symp. Circuits Syst.*, May 1990, pp. 865–868.

[12] C. J. Alpert, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky, “Minimum buffered routing with bounded capacitive load for slew rate and reliability control,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 3, pp. 241–253, Mar. 2003.

[13] S. Hu, C. Alpert, J. Hu, S. Karandikar, Z. Li, W. Shi, and C. Sze, “Fast algorithms for slew-constrained minimum cost buffering,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 11, pp. 2009–2022, Nov. 2007.

[14] J. Lillis, C.-K. Cheng, and T.-T. Lin, “Optimal wire sizing and buffer insertion for low power and a generalized delay model,” *IEEE J. Solid-State Circuits*, vol. 31, no. 3, pp. 437–447, Mar. 1996.

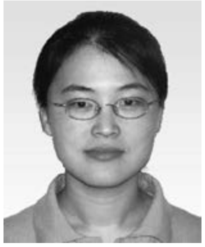
[15] N. Drego, A. Chandrakasan, and D. Boning, “Lack of spatial correlation in MOSFET threshold voltage variation and implications for voltage scaling,” *IEEE Trans. Semicond. Manuf.*, vol. 22, no. 2, pp. 245–255, May 2009.

[16] K. Boese and A. Kahng, “Zero-skew clock routing trees with minimum wirelength,” in *Proc. 5th Annu. IEEE Int. ASIC Conf. Exhib.*, Sep. 1992, pp. 17–21.

[17] M. Jackson, A. Srinivasan, and E. Kuh, “Clock routing for high-performance ICs,” in *Proc. ACM Des. Automat. Conf.*, 1990, pp. 573–579.

[18] A. Agarwal, V. Zolotov, and D. Blaauw, “Statistical clock skew analysis considering intradie-process variations,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 23, no. 8, pp. 1231–1242, Aug. 2004.

[19] *FreePDK45:Metal Layers* [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK45:Metal_Layers



Xin Zhao (S’07) received the B.S. degree from the Department of Electronic Engineering in 2003, and the M.S. degree from the Department of Computer Science and Technology Department in 2006, both from Tsinghua University, Beijing, China. She is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

Her current research interests include computer-aided design for very large-scale integrated circuits (ICs), especially on physical design for low power, robustness, and 3-D ICs.

Ms. Zhao was nominated for the Best Paper Award at the International Conference on Computer-Aided Design in 2009 and the IEEE Transactions on Computer-Aided Design in 2012.

Jeremy R. Tolbert (S’08) received the B.S. degree in electrical engineering from the University of Michigan, Ann Arbor, in 2007, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2011. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Georgia Institute of Technology.

His current research interests include low-power circuits and systems, techniques for robust sub-threshold design, and energy-efficient processing for mobile computing.

Mr. Tolbert was a recipient of the National Science Foundation’s Graduate Research Fellowship.

Saibal Mukhopadhyay (S’99–M’07) received the B.E. degree in electronics and telecommunication engineering from Jadavpur University, Kolkata, India, in 2000, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 2006.

He was a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, where he was involved in high-performance circuit design and technology-circuit codesign focusing primarily on static random access memories. Since September 2007, he has been an Assistant Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA. He has authored or co-authored over 120 papers in refereed journals and conferences. He owns five U.S. patents. His current research interests include analysis and design of low-power and robust circuits in nanometer technologies.

Dr. Mukhopadhyay received the National Science Foundation CAREER Award in 2011, the IBM Faculty Partnership Award in 2009 and 2010, the SRC Inventor Recognition Award in 2008, the SRC Technical Excellence Award in 2005, the IBM Ph.D. Fellowship Award for 2004 to 2005, the Best in Session Award at the 2005 SRC TECHCON, and the Best Paper Award at the 2003 IEEE Nano and 2004 International Conference on Computer Design.

Sung Kyu Lim (S’94–M’00–SM’05) received the B.S., M.S., and Ph.D. degrees from the Department of Computer Science, University of California at Los Angeles (UCLA), Los Angeles, in 1994, 1997, and 2000, respectively.

In 2001, he joined the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he is currently an Associate Professor. His current research interests include architecture, circuit, and physical design for 3-D integrated circuits and 3-D system-in-packages. He is the author of *Practical Problems in VLSI Physical Design Automation* (New York: Springer, 2008).

Dr. Lim received the Design Automation Conference Graduate Scholarship in 2003, the National Science Foundation Faculty Early Career Development (CAREER) Award in 2006, and the ACM Special Interest Group on Design Automation (SIGDA) Distinguished Service Award in 2008. He was on the Advisory Board of the ACM SIGDA from 2003 to 2008. He was an Associate Editor of the IEEE Transactions on Very Large Scale Integration Systems from 2007 to 2009. He was a Technical Program Committee Member of several ACM and IEEE conferences on electronic design automation. His work was nominated for the Best Paper Award at ISPD’06, ICCAD’09, CICC’10, DAC’11, and DAC’12.