Analysis and Design of Energy and Slew Aware Subthreshold Clock Systems

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

***Abstract*—In this paper, we analyze the effect of clock slew in subthreshold circuits. Specifically, we address the issue that variations in clock slew at the register control can cause serious timing violations. We show that clock slew variations can cause frequency targets to deviate by as much as 28% from the design goals. Based on these observations, we recognize the importance of clock slew control in subthreshold circuits. We propose a systematic approach to design the clock tree for subthreshold circuits to reduce the clock slew variations while minimizing the energy dissipation in the tree. The combined approach, including the wire sizing and dynamic nodal capacitance control, can achieve better slew control (and better timing control) at lower energy in subthreshold circuits.**

***Index Terms*—Design automation, reliability, system analysis and design.**

I. Introduction

TRANSISTORS OPERATING in the subthreshold region constitute an attractive technology for ultralow power mobile applications such as micro-sensors and biomedical devices. When the primary goal is to save energy, subthreshold logic can allow for significant power reduction by operating at a supply voltage lower than the threshold voltage of the devices. Even though low power is the main focus, it is still innate for the designer to optimize secondary parameters such as robustness and performance. As a result of these efforts, works have been presented to optimize energy and delay, while performing computations with minimal error [1]–[3]. Additionally, efforts have been made to optimize devices such that circuits can be operated at medium frequencies in the order of tens to hundreds of megahertz [4].

In addressing the design of an optimal energy-delay subthreshold system, the clock network plays a significant role. Delivering robust clock signals to hundreds or even thousands of flip-flops requires the clock tree to be optimally designed to handle issues of delay, skew, and jitter. In subthreshold, the signal slew (10% to 90% transition time) also has capacity to affect system performance [5], [6]. Additionally, due to its high switching activity, the clock network can contribute up to 40% of the total dynamic power [7]. The same trend is expected when an above threshold system is scaled to subthreshold voltages. Thus, designing a low-power yet robust clock tree is a critical challenge to implement large scale subthreshold systems. This challenge is increasingly difficult because subthreshold designs are always constrained by the requirement of robustness. As the supply voltage of digital circuits is scaled below the device threshold, the characteristics of the transistor change. An immediate observation is that the current in the subthreshold regime has an exponential dependence on gate voltage, threshold voltage, and additional parameters that are functions of the process. This is in contrary to the above threshold design, whose dependence has been noted to be linear or square [8]. As a result of these effects, small variations in the subthreshold regime have been known to follow this exponential dependence, and care has been taken to control these effects [9]–[13].

The purpose of this paper is to analyze the impact of clock slew in subthreshold design and propose a technique for a low-power slew controlled clock tree design. We examine the inherent slew variations in a clock tree and show that the slew variations can cause a direct increase in cycle time computations. We propose a systematic approach to design the clock tree for subthreshold circuits to reduce the clock slew variations while minimizing the energy dissipation in the tree. We will show that a tighter nodal capacitance control is necessary to control the slew in a subthreshold clock tree, which can increase the energy dissipation. Recognizing that the wire resistances have a negligible effect in subthreshold circuits, we will show proper wire sizing is necessary to reduce the clock energy. Finally, we propose a dynamic nodal capacitance control technique that allows larger slew at the earlier nets of the tree while controlling it more aggressively near the sink nodes.

The rest of this paper is organized as follows. Section II addresses the motivation. Section III provides current techniques that can be used for low energy slew control in subthreshold. Section IV proposes and analyzes dynamic C_{MAX} selection as a new approach to clock tree design. Section V presents a discussion regarding process, voltage and temperature variations. Section VI summarizes with a conclusion.

Manuscript received February 10, 2011; accepted March 14, 2011. Date of current version August 19, 2011. This work was supported by the National Science Foundation, under Grant CCF-0917000, the National Science Foundation Graduate Research Fellowship, under Grant DGE-0644493, the SRC Interconnect Focus Center, and Intel Corporation. This paper was recommended by Associate Editor I. Bahar.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: jeremy.r.tolbert@gatech.edu; xinzhao@ece.gatech.edu; limsk@gatech.edu; saibal@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.2011.2144595

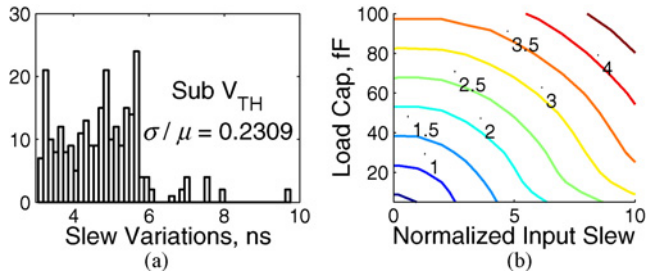


Fig. 1. (a) Deterministic slew variations at the sink nodes of a subthreshold clock tree designed using above threshold concepts. (b) Normalized output slew contours and their dependence on input slew and total load capacitance for an inverter.

II. Motivation

The purpose of this section is to demonstrate to the reader the effects of clock slew, and how it can directly impact the system cycle. The input slew of a logic gate can cause the output delay to change in the range of 50–100% [5], [6]. The output slew of a logic gate has a strong dependence on the device dimensions, gate and drain voltages, as well as the load capacitance. More recently, this effect has been designated a concern for flip-flop design in the subthreshold region [13]. When designed with above threshold methods, subthreshold clock trees exhibit significant slew variations at the sink nodes (i.e., the nodes directly connected to the latches) that will increase the probability of timing violations. As an example, the focus of this experimental section will be based on a clock network designed using an above threshold, zero skew clock tree design algorithm, with a slew control method that limits the maximum capacitance driven by each internal and external clock node (i.e., hereafter, referred to as the nodal capacitance). The power supply of this design was then scaled to below the device threshold. In this clock tree, inverting buffers were used to reduced the number of devices and thus save on power. The clock tree was designed using a 65 nm predictive technology model (PTM) with $V_{TP} = -378$ mV and $V_{TN} = 429$ mV [14]. The power supply was 300 mV and the clock tree has 267 sink nodes, each driving multiple flip-flops. The design used to define the clock sink destinations is the IBM r1 benchmark [15].

A. Deterministic (Design Induced) Slew Variations

Fig. 1(a) depicts the deterministic slew distribution at the sink nodes for the subthreshold clock tree described above.

Deterministic or design induced variations occur as a function of load capacitance, routing distances, and buffer placement. Under this definition, it is unlikely that all sink nodes will have the same slew, which results in a deterministic variation across the chip. In an ideal case, each individual chip would have the same spread of deterministic slew variations. The slew at the clock sink nodes is important because they are the control for the latches and flip-flops in a chip. The coefficient of variation, CV, is a normalized measure of dispersion of a probability distribution and is defined as the ratio of standard deviation (σ) to mean (μ)

$$CV = \frac{\sigma}{\mu}. \tag{1}$$

In the subthreshold clock tree the CV is 23%, showing there is a wide distribution of slew. Well controlled CVs are in the range of 10–15%.

This distribution of slew shown in Fig. 1(a) corresponds to the slew variation across different sink nodes of the tree. This slew variation is caused by variation in the output slew of the inverter stages of the clock tree. Fig. 1(b) shows the effect of input slew and load capacitance on the output slew of an inverter. The line contours represent that the output slew of an inverter has a strong dependence on the input slew and load capacitance. This is important for clock tree paths, because it is composed of numerous inverters, and the slew and capacitance often vary from node to node. For a smaller output slew, a smaller input slew and load capacitance are required. The recovery of the slew (i.e., output slew smaller than the input slew) through an inverter stage is an important consideration for the clock tree path. If not controlled, the slew can become progressively higher as signals progress down a clock path. For example, with a load capacitance of 80 fF and input slew of 1 ns, the output slew is 3X the input slew. The slew recovery is a strong function of the load capacitance. Fig. 1 shows that if the load capacitance is reasonable, the inverter can recover slew even for a large input slew. Hence, controlling the capacitance driven by each node in the clock tree is very important to control slew propagation through the tree and reduce the slew variation at the sink nodes.

B. Clock Slew Impact on Cycle Time

A direct motivation of this paper is the impact of clock skew and slew on the cycle time. The schematic in Fig. 2(a) shows a generic logic path composed of fan-out-4 NAND gates between two registers. The minimum cycle time and the maximum clock frequency for the above system is given by

$$T_{\min} \geq t_{c-q} + t_{\text{logic}} + t_{su} - \delta \tag{2}$$

$$F_{\max} = 1/T_{\min} \tag{3}$$

where t_{c-q} denotes the maximum propagation delay of the register (clock-to-q), t_{logic} is the maximum delay of the combinational logic, t_{su} is the setup time for the registers, and δ the clock skew. The graph in Fig. 2(c) plots three $F_{\max}$ cases for this subthreshold path as we alter the total number of stages (N). In each case, skew and slew requirements are as follows:

1) *optimal*: 0 ns clock skew and 1.0 ns clock slew;
2) *skew only*: a skew that is 5% of the optimal period and 1.0 ns clock slew;
3) *slew + skew*: takes into account both skew and slew, where skew is 5% of the optimal period and slew is around 10 ns.

The 10 ns slew is chosen in the experiment to reflect the maximum slew obtained from the clock tree that was first designed in above threshold and then operated at subthreshold condition. As expected, the skew reduces the operating frequency. When both slew and skew are considered, $F_{\max}$ can be reduced from 14% to 28% depending on the length of the logic path. In essence, as subthreshold systems target higher

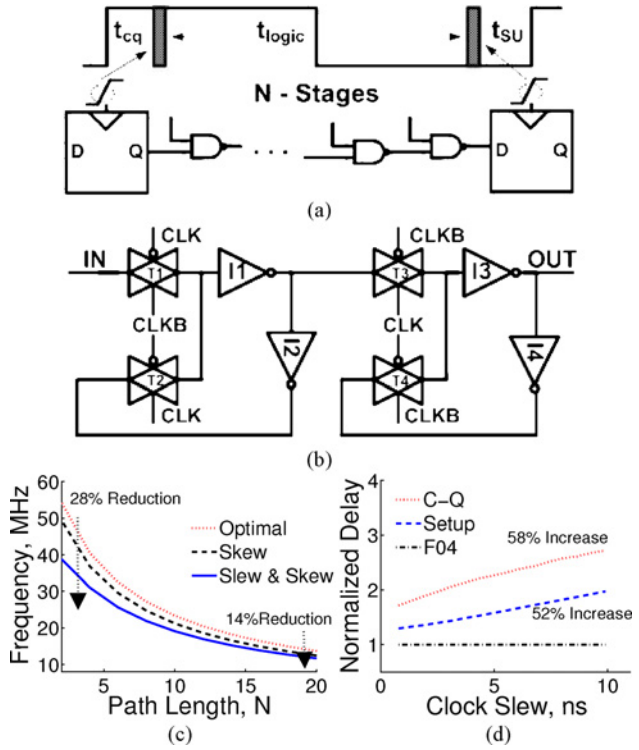


Fig. 2. (a) Sample logic path simulated with F04 nand delays. (b) Transmission gate flip-flop used for subthreshold experiments. (c) Impact of clock skew and clock slew on frequency. (d) Timing variations for a clock tree designed in above threshold lowered to subthreshold.

frequencies (i.e., as the logic path reduces), the effect of slew on cycle time cannot be ignored.

To understand the effect of clock slew directly, we have analyzed how these variations impact the setup time and clock-to-q. Fig. 2(b) depicts a commonly used transmission gate flip-flop configuration that can be used in subthreshold operation [16]. The plot in Fig. 2(d) shows how the slew distribution directly affects the timing metrics described above. They have been normalized to the delay of a fan-out-4 gate, t_{FO4}, since the focus of this paper is on the general behavior. When slew variations are applied to clock, the setup time is directly proportional to it. In severe cases the setup time can vary by 52% worse than the best case achieved. This variation in setup time reduces the time available to compute logic, and in some cases will cause errors in the logic by violating the setup requirements. For the given slew distributions, the clock-to-q delay can be 58% worse than its best case value.

C. Deterministic (Design Induced) Timing Variations

In the previous subsections, we have discussed that: 1) a clock tree design causes slew variations, and 2) slew variations have the potential to cause severe timing violations in subthreshold. In this section, we make the connection that the design of the clock tree contributes to these timing variations. Fig. 3(a) and (b) shows the distribution of the timing metrics that impact cycle time. Fig. 3 reiterates the concept that the setup time and clock-to-q are worsened by clock slew, as the deterministic variations are plotted.

Clock slew directly impacts timing metrics; a smaller slew variation translates to smaller setup and clock-to-q variations.

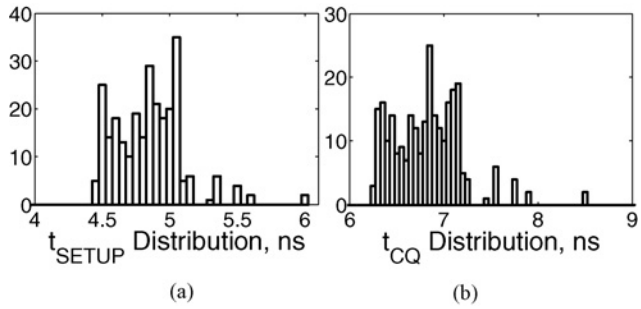


Fig. 3. (a) Setup time and (b) Clock-Q distributions for a clock tree designed in above threshold and then the power supply is scaled to subthreshold voltages. The variations are normalized by the best case scenario.

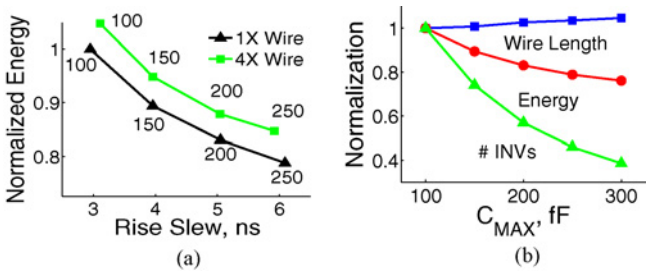


Fig. 4. Summary of experimental methods to reduce the slew variations in subthreshold. (a) shows that increasing C_{MAX} increases slew and reduces power. The numbers indicate the fixed C_{MAX} in fF. (b) Impact of C_{MAX} constraint on wire length, energy, and buffer count.

By reducing the deterministic clock slew variations, an optimal maximum frequency can be met.

III. Techniques for Low Energy Slew Control in Subthreshold

The findings from Section II show that it is necessary to control and reduce the variations associated with clock slew for robust subthreshold operation. The focus of this section is on investigating techniques to design a clock tree in subthreshold that provides a smaller slew variation.

A. Smaller C_{MAX} Requirements in Subthreshold

Conventional, above threshold methods for controlling clock slew currently exist and rely on limiting the maximum nodal capacitance, C_{MAX} a buffer can drive [17]–[20]. In general, when a buffer reaches a node that exceeds the designated C_{MAX}, buffer insertion is required to reduce the load. In our analysis of an above threshold clock tree scaled to operate in subthreshold, the C_{MAX} was defined by above threshold methods. In above threshold, there is more available current to drain the charge from the node; therefore we should impose a smaller C_{MAX} requirement for optimal subthreshold clock trees. Fig. 4(a) shows the results of a clock tree designed in subthreshold, with varying C_{MAX}. Keep in mind that this figure of merit C_{MAX} represents the maximum nodal capacitance a node can have, and in most cases the actual load capacitance of a node is less than the C_{MAX}. The results show that it is possible to better control the slew in subthreshold reducing the average rise slew from 6 ns to 3 ns. At the same time we are

reducing the slew, we are increasing the energy by nearly 20%. This behavior is exhibited because as we reduce the C_{MAX} each node can drive, the total interconnect capacitance remains constant and we need to insert more buffers to compensate for the reduced C_{MAX}. Fig. 4(b) shows this trend of energy and number of inverters as a function of C_{MAX}. Additionally, the total wire length of the design has small changes, because whenever an inverter is removed a small wire segment takes the place of the removed inverter. Note that going from a C_{MAX} of 100 fF to 300 fF, the number of inverters decreases by 60%, but the energy only decreases by nearly 20%. A large portion of this is unaffected energy comes from the large interconnect capacitance. In summary to design a subthreshold clock tree, we require a smaller C_{MAX} than in above threshold.

B. Minimum Wire Width in Subthreshold

The wire interconnect contributes to a significant portion of the energy in a clock network. Reducing the capacitance in interconnect without sacrificing delay in the clock path can help to address this energy component. When modeling the interconnect as a distributed RC line, the design rule of thumb is that rc wire delays should only be considered when the line being modeled has reached a critical length, L_{crit} [16]

$$L_{\mathrm{crit}} >> \sqrt{\frac{t_d}{0.38rc}} \tag{4}$$

where t_d is the gate delay, r the resistance per unit length, and c the capacitance per unit length. Based on the above equation, the L_{crit} for a 1 ns gate in 65 nm (using the PTM model) should be much greater than 4 mm. This value also assumes a minimum wire width to reduce the interconnect capacitance. A recent work showed a 65 nm subthreshold chip with a 2.29 mm × 1.86 mm area [21]. Using this as a benchmark we could expect a worse case wire length of just over 4 mm. Recent products in 65 nm technology from Intel show similar results in regards to maximum wire lengths [22]. Since we expect the length of the wire to be much less than the critical length, it may be possible to neglect the distributed rc behavior of the wire. When $L << L_{\mathrm{crit}}$, the wire can be modeled primarily as a lumped capacitance. This is possible in subthreshold as the device resistance is much higher than the wire resistance and we are operating at lower frequencies. Since rc wire delays are negligible, wire resistance does not play a critical role in determining wire delay and all wires can be designed with minimum width. In above threshold it is not possible to neglect rc wire delays as the operating frequencies are higher and driver delays are much smaller. Therefore the wires are usually larger than minimum width.

The major advantage of reducing the wire width is the corresponding decrease in the wire capacitance. Referring back to Fig. 4(a), we observe that reducing the wire width in a subthreshold clock tree reduces the energy without sacrificing slew. The above threshold clock tree used 4X minimum wires to ensure the rc delays were handled correctly. When this tree is lowered to subthreshold voltages, the 4X wire width only adds capacitance and thus energy to the system. This result can allow us to design the clock tree in subthreshold with minimum widths to reduce wire cap and neglect the wire resistance compared to the driver resistance.

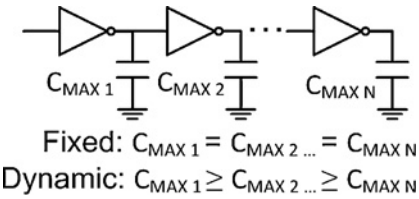


Fig. 5. Dynamic C_{MAX} selection is the proposed technique to reduce slew variations at the sink nodes, while ensuring the energy does not increase.

IV. Optimizing Subthreshold Clock Tree for Reduced Slew and Energy With Dynamic C_{MAX}

The previous section has provided insight into reducing the slew (and variation) while at the same time considering energy. While they each have their pros, alone they cannot provide an optimal subthreshold clock tree for slew control. As a result, we propose a new technique for optimal clock tree design in subthreshold, based on regressive C_{MAX} near clock sinks. Since the proposed technique allows the C_{MAX} to vary dynamically from one level to another level, we refer to this method as Dynamic C_{MAX} instead of the conventional approach (referred to as Fixed C_{MAX}) where C_{MAX} is constant across all levels. We compare the slew, skew, and energy behaviors of the clock trees designed using Dynamic C_{MAX} against the ones designed using Fixed C_{MAX}. The Fixed C_{MAX} trees are first designed in the above threshold voltage and the supply voltage is simply scaled to study their performance/energy at subthreshold levels. We first explain the basic concept of Dynamic C_{MAX}. Next we study the behavior of broad selection of Dynamic C_{MAX} based clock trees to understand what traits the desired trees have in common, and use that information as a basis for future designs. Finally, we summarize the results to show how tighter C_{MAX}, lower wire-width, and Dynamic C_{MAX} helps design better subthreshold clock tree compared to simply scaling the supply voltage of a clock tree from above threshold to subthreshold level. In essence, we reinforce the principle that new design concepts are required for designing a clock tree in subthreshold and simply scaling the voltage of an above threshold tree is not sufficient.

A. Principles of Dynamic C_{MAX}

From the results of Fig. 4(a), we know that we can reduce the power in the clock network by increasing the C_{MAX} that each node drives. At the other end of the spectrum, a smaller C_{MAX} will control slew better. For the purposes of controlling slew, we are most interested in reducing variations at the sink nodes, which are directly attached to flip-flops. We still need to control slew at other nodes, but we can relax constraint in order to save energy (fewer or smaller buffers). To reduce energy while achieving a proper slew at the sinks, C_{MAX} should vary at each level of the clock tree, reducing the closer we get to the sink. Additionally, if we ensure the wire widths as minimum, we should receive more power savings. Fig. 5 summarizes

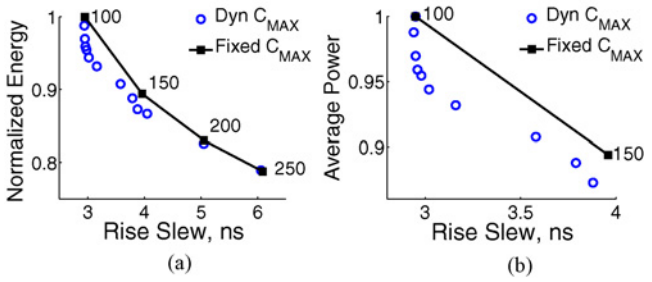


Fig. 6. (a) Preliminary results of slew control and energy savings of Dynamic C_{MAX} selection. (b) Results zoomed in for a rise slew of 3–5 ns. The numbers indicate the fixed C_{MAX} in fF. By traversing the dynamic C_{MAX} path, we can achieve better slew control at targeted power.

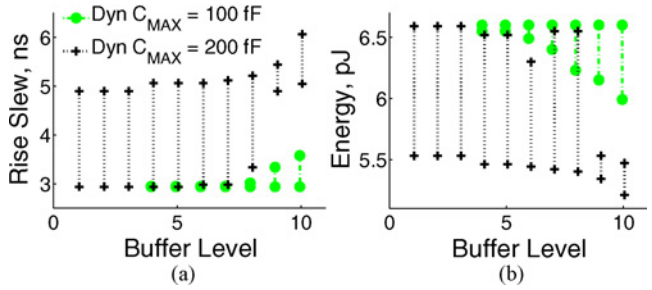


Fig. 7. Effect of independent dynamic C_{MAX} values and their impact on the (a) final clock slew and (b) total clock tree energy.

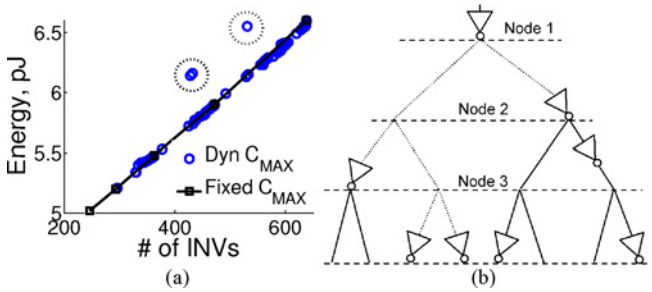


Fig. 8. (a) Correlation of inverter count and clock tree energy. (b) Example binary clock tree network with merging node definitions. With Dynamic C_{MAX} selection, different C_{MAX} values are selected at each buffer level, denoted by the buffer output.

the proposed technique and the previous method with Fixed C_{MAX}. In the proposed technique of Dynamic C_{MAX}, the C_{MAX} values selected are reduced from the source (buffer level 1) to the sink (buffer level N). We cannot guarantee that the C_{MAX} values selected are the *exact* values that a level will drive. We can, however, guarantee that the C_{MAX} values will limit the *maximum capacitance* a level can drive. Fig. 6 shows the potential advantage that Dynamic C_{MAX} selection has compared to Fixed C_{MAX} methods. It is apparent that the axes in the figure represent a energy versus robustness plane. In an optimal case, there is zero slew and zero energy, leading us toward the origin. Based on this, curves that are closer to the origin represent a better energy-robustness tradeoff. Fig. 6(a) shows the trend for a selection of points, while Fig. 6(b) zooms into a region where slew is small. In Fig. 6(b), we see that it is possible to reduce the power in the clock network by nearly 6% while maintaining the same slew, by employing Dynamic C_{MAX} selection.

B. C_{MAX} Selection Trends of Clock Trees

To understand the trends of Dynamic C_{MAX} based clock trees, we simulated 79 trees designed using the rules of Dynamic C_{MAX} selection. The trees were all implemented to deliver clock to the same r1 IBM benchmark as before, which has 267 sinks. Additionally, there are definitions for 10 buffer levels and thus 10 dynamic C_{MAX} selections required. The Dynamic C_{MAX} values ranged from 100 fF to 400 fF in 50 fF intervals. All 79 trees were uniquely selected to provide a broad range of dynamic clock trees, so that the results could provide insight into general trends. The authors believe that in lieu of all 7^{10} possible combinations, this is the best scenario for investigation.

1) *C_{MAX} Value Selection and Level Placement:* Fig. 7(a) shows how dynamic C_{MAX} values for each clock tree level can affect slew delivered to the latches. The endpoints (for a given level) denote the minimum and maximum ranges for slew when the given Dynamic C_{MAX} value has been placed at that level. For example, of all the 79 Dynamic C_{MAX} trees selected, when level 5 has a C_{MAX} of 200 fF, the final clock slew is in the range of 3–5 ns.

Since we are reporting results for a per level basis, we do not directly know how the remaining levels in the designated tree have been selected. That is, given that level 5 has a C_{MAX} value of 200 fF, based on the Fig. 7(a) we cannot definitely say we know the C_{MAX} values of levels 1 to 4, and 6 to 10. We can however, indirectly know that the following holds true:

$$C_{MAX(1...4)} \geq C_{MAX(5)} \geq C_{MAX(6...10)}. \quad (5)$$

The interesting features to note for Fig. 7(a) are that when level 5 has C_{MAX} value of 100 fF, the slew at the latches is well controlled. This is because (5) has to hold true, and since 100 fF is the smallest cap value selected, when level 5 is 100 fF, levels 6–10 are 100 fF as well. This fact results in a well controlled slew delivered to the latches.

Another interesting point to make is looking at level 10. When the C_{MAX} value selected at level 10 is 100 fF, we have smaller min/max values compared to when the C_{MAX} value is 200 fF. This reinforces the concept that a smaller C_{MAX} value near the sink (level 10) has the best chance to reduce the slew.

Fig. 7(b) shows an inverse trend with Dynamic C_{MAX} values based on the energy in the entire clock tree. What we notice here is that although a smaller C_{MAX} at level 10 can provide lower slew at the latches, it can also increase the energy in the entire system as well. The latter point is denoted by the range at level 10 of 6–6.6 pJ, when C_{MAX} is 100 fF. On the contrary, a larger C_{MAX} value at level 10 can yield lower energy, by sacrificing the slew delivered to the latches.

The last thing to note from Fig. 7 is that the further away we are from the sink level (10), the more disparity we have between min/max values for slew and energy. This occurs when the C_{MAX} value is 200 fF, because there is more flexibility in C_{MAX} values for the levels near the sink. Ultimately a C_{MAX} value closer to the sink (level 10) has more restrictions in the slew and energy than a C_{MAX} value selected

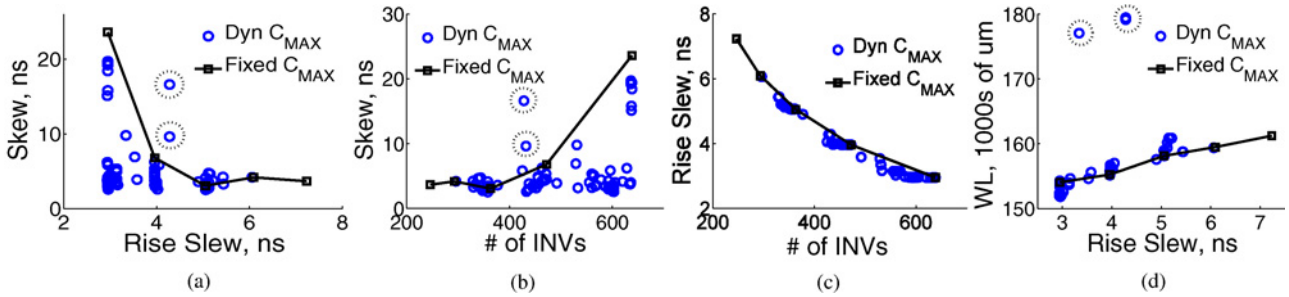


Fig. 9. Indirect relationship between the clock slew (delivered to the latches) with (a) clock skew and (d) wirelength. The relationship between inverter count with (b) skew and (c) slew.

near the source (levels 1–3). In general, selecting a C_{MAX} value for each node can turn designing an optimal clock tree into a lesson in pure combinatorics. With that being said, it should be possible to design an optimal tree using combinatorics and the previously described relationships between C_{MAX}, slew and energy.

2) *Energy is Directly Related to Inverter Count:* By changing the C_{MAX} values at different levels, we are directly altering the number of inverters in the clock tree system. From Fig. 8(a), we recognize the relationship between the inverter count and clock tree energy. The direct trend hints that most of the power associated in the clock trees are attributed to the inverters. As a reiteration, to control the number of inverters (and thus energy) in a clock tree, we can change the C_{MAX} values at each level. This can be seen from the previously described Fig. 7(b).

To explain the direct correlation, we can examine the structure of a clock tree network. Assume (for simplicity) that the clock distribution network is the form of the binary tree shown in Fig. 8(b), a merging node is defined as the junction where the clock tree splits into two directions. Buffer levels are at the output of a buffer, where C_{MAX} values need to be defined. Recall that in Fig. 7(b), at the clock tree sink, (buffer level 10), when the C_{MAX} value is 100 fF, it has a higher energy than when it is 200 fF. There are two factors that could contribute to this increase in energy. First, from our binary tree model and reducing C_{MAX}, we know that the clock nodes near the sink are guaranteed to have more inverters than at the higher levels (near the source). This is based on the nature of the binary tree and by reducing the C_{MAX} values near the sink; we are forced to introduce more inverters. Essentially, we are introducing a large fraction of inverters by reducing C_{MAX} at the lowest level. Therefore, energy is strongly controlled by the C_{MAX} values, and more importantly, C_{MAX} values selected nearest to the sink.

3) *Dynamic C_{MAX} Can Reduce Slew, Skew, Wirelength:* Knowing that inverter count is directly related to the clock tree energy, there are also indirect trends related to the clock slew that can be considered. Fig. 9 compares how the clock slew and inverter count are correlated to important metrics in clock tree design: clock skew, slew, and wirelength. For Fixed C_{MAX}, we considered a constant C_{MAX} for all levels and generated the clock tree using conventional low-skew clock tree generation methods [23], [24]. We analyzed the generated trees to estimate the slew, skew, number of inverters, and wirelength. The different points in the Fixed C_{MAX} lines for all subplots of Fig. 9 represent the trees generated by different Fixed C_{MAX} values. Fig. 9(a) shows that when designing with a Fixed C_{MAX} method, there is an inverse relationship between clock skew and the clock slew delivered to the latches. With the Fixed C_{MAX} method, smaller C_{MAX} values are used everywhere to achieve smaller slew. For a given buffer size a tighter C_{MAX} at all levels requires more inverters in the tree. We also observe that a tighter C_{MAX} constraints results in larger deterministic skew. Note that if random process variation is considered, larger number of inverters also implies more sources of variations, which could further increase the random variation in skew. In summary, when designed using Fixed C_{MAX}, there exists an inverse relationship between slew, and skew: when the C_{MAX} is less, slew is larger and skew is smaller. Fig. 9(b) and (c) supports the observed inverse trend of skew and slew for Fixed C_{MAX} trees. Hence, from a design perspective, a tradeoff is required when using the Fixed C_{MAX} method, as it is only possible to achieve low slew or low skew. The observed trends for a Dynamic C_{MAX} tree are different. As with the Fixed C_{MAX} method, the slew reduces with increasing the number of inverters (assuming same buffer sizes for a single design). However, by employing various Dynamic C_{MAX} topologies, we can increase the number of inverters in a design, and still achieve a reduced skew design. Essentially, Dynamic C_{MAX} method allows more degrees of freedom in placement of the inverters as nodes at different levels have different C_{MAX} constraints. Consequently, for same number of inverters, there exist several different instances of clock tree (for same design) with varying skew [Fig. 9(b)]. For example, in Fig. 9(b) when the number of inverters is nearly 600, a Dynamic C_{MAX} tree can achieve a reduced slew as low as 2 ns, or as high as 20 ns. That is, with an optimal selection of Dynamic C_{MAX} values for different levels, it is possible to achieve low slew and low skew by reducing the C_{MAX} near the sink, and increasing the C_{MAX} near the source [Fig. 9(a)].

Fig. 9(d) shows how the wirelength is correlated with the slew rate delivered to the latches. The increase in the number of inverters with an increase in the slew can be explained based on the absence of inverters. As stated before, when the C_{MAX} values are large, there are fewer inverters since each inverter has a larger limit of capacitance it can drive. With fewer inverters the slew will tend to increase, because there is less control. Additionally, with fewer inverters there will be more wirelength to compensate for the removed inverters.

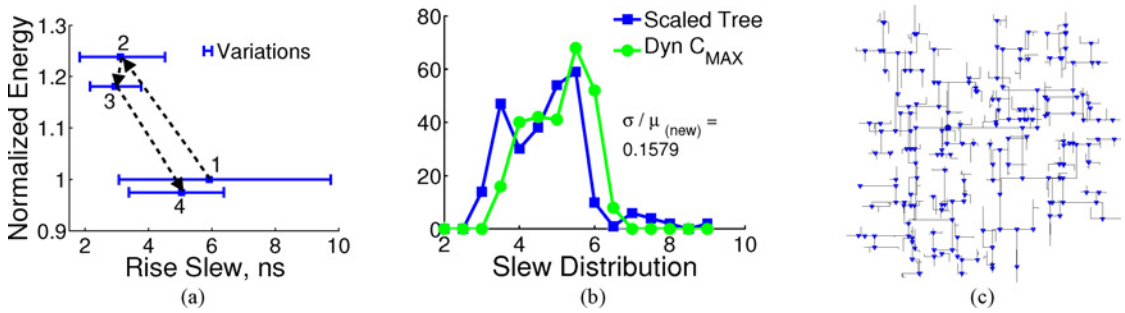


Fig. 10. (a) Summary of combined approaches to reduce deterministic slew variations in subthreshold. (b) Dynamic C_{MAX} deterministic slew distribution compared with the original clock tree. The new slew distribution has a coefficient of variation of 0.1579, better than the original 0.2309 of Fig. 1. (c) Clock tree (4) routed using dynamic C_{MAX} and combined methods.

TABLE I
SUMMARY OF FIXED AND DYNAMIC C_{MAX} VALUES WITH COMBINED SLEW REDUCTION TECHNIQUES

Tree	1 Source	2	3	4	5	6	7	8	9	10 Sink	Norm. Energy	μ Slew (ns)
#1 (4x WL)	250 fF	250 fF	250 fF	250 fF	250 fF	250 fF	250 fF	250 fF	250 fF	250 fF	1.00	5.92
#2 (4x WL)	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	1.23	3.11
#3 (1x WL)	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	100 fF	1.18	2.95
#4 (1x WL)	300 fF	300 fF	300 fF	300 fF	300 fF	200 fF	200 fF	200 fF	200 fF	200 fF	0.97	5.05

If we compare the Fixed C_{MAX} and Dynamic C_{MAX} selection trends there are a few points to consider. First, they both follow the explained trend above that a larger slew is correlated with a longer wirelength. Second, for smaller slew targets, it is possible to achieve a smaller wirelength using Dynamic C_{MAX} selection compared to Fixed C_{MAX}. This can be seen in the figure where the points are located around 3 ns of slew.

4) *Explanation of Outliers:* During our preceding discussion of the trends of dynamic clock trees, we have neglected the outliers that do not follow the trends. These points can be seen from the dotted circles in Figs. 8 and 9. Since Dynamic C_{MAX} can only limit the maximum capacitance a level can drive, it has no bearing on the minimum capacitance. If for example

$$C_{MAX(6...10)} = 300\,\text{fF} \tag{6}$$

we cannot assure that

$$C_6 \geq C_8 \geq C_7 \geq C_9 \geq C_{10} \tag{7}$$

where C_i denotes the actual capacitance at the level after the clock tree has been routed. Recall from Fig. 1(b) that when the actual capacitance does not follow the trend in (7), the slew is not well controlled and can become unpredictable. An algorithm to control the upper and lower capacitance values would assure optimal design.

5) *Limitations of Buffer Reduction:* In Section IV-B2, we learned the direct correlation between energy and inverter count. While it is true that shallower trees may be better for skew as described in [25], there may be secondary effects of slew depending on the size of the circuit. In an attempt to reduce the overall power by continuing the trend of reducing the number of inverters, we designed a 1-buffer H-tree to investigate the limitations of buffer reduction. A comparison of this design and the previously described designs can be seen in Table II. We recognize that the skew is improved using a 1-buffer H-tree compared to the previous options, but the slew has increased drastically. In this design, the large size of the IBM r1 benchmark circuit has contributed to a long wire length and thus extra capacitance. Additionally, a large buffer has been introduced in the 1-buffer H-tree to accommodate this and the energy is still greater than that of the dynamic CMAX tree. In essence, the 1-buffer H-tree is best for minimum skew and is appropriately used for small-scale designs. However, for large-scale designs, using a clock-tree with dynamic CMAX design, it is possible to achieve a near optimal tree with good slew, skew, and minimum energy.

C. Summary and Results of Dynamic C_{MAX}

In this section, we summarize the effects of using tighter C_{MAX}, reduced wire-width, and Dynamic C_{MAX}. As stated before, we implemented several subthreshold clock trees in 65 nm CMOS, based off the PTM Model at V_{DD} = 300 mV, and the best were selected for comparison. Fig. 10(a) represents a plane of transitions as a different technique is employed at each number to reduce the slew variations. The clock tree "1" represents the base-line of comparison (i.e., a standard clock tree designed in above threshold using Fixed C_{MAX} = 250 fF), scaled to subthreshold voltages. The details of C_{MAX} values used for these trees are shown in Table I. The results are depicted for rise slew information, but a similar trend exists for fall slews. The techniques to reduce energy and slew were applied as follows:

1) **(1 to 2)** the C_{MAX} was reduced from 250 to 100 fF and the tree was redesigned using Fixed C_{MAX};
2) **(2 to 3)** the wire width was reduced from 4X to 1X minimum sized, and then redesigned with Fixed C_{MAX};

TABLE II
METHODOLOGY COMPARISON FOR IBM R1 BENCHMARK

Sub-V_T Design Method	Energy (pJ)	Min Slew (ns)	Max Slew (ns)	Mean Slew (ns)	Skew (ns)	WL (μm)
Fixed C_{MAX} [17]–[20]	6.60	2.15	3.77	2.95	25.9	154 048
1-buffer H-Tree [25]	5.89	35.37	39.64	38.15	2.77	151 727
Dynamic C_{MAX} [this work]	5.47	3.38	6.38	5.05	3.39	158 954

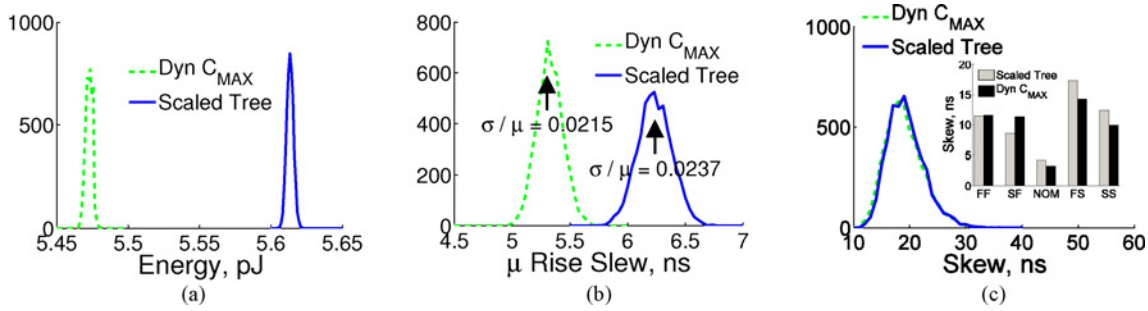


Fig. 11. Monte Carlo simulation of varying threshold voltages for clock tree. (a) Energy. (b) Slew. (c) Skew. The slew variations are reported as an average and the skew variations are reported as a worst case for each Monte Carlo simulation.

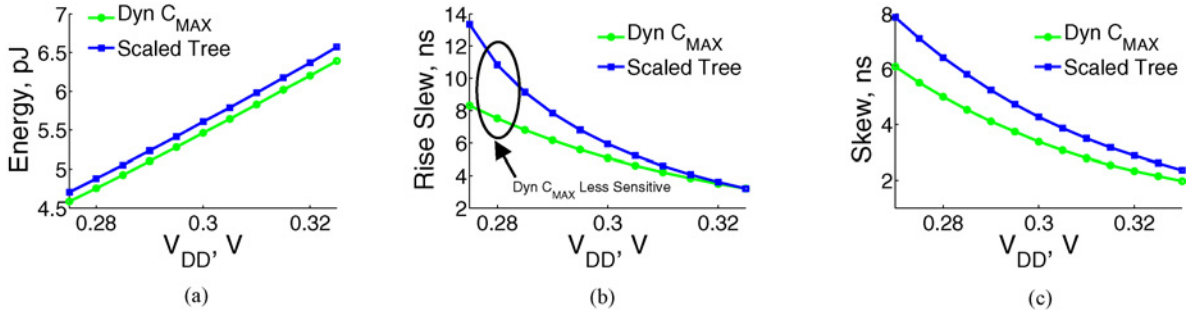


Fig. 12. Supply voltage sweep and its impact on clock tree. (a) Energy. (b) Slew. (c) Skew. At low voltages, the Dynamic C_{MAX} tree is less sensitive to supply voltage variations.

3) **(3 to 4)** Dynamic C_{MAX} selection was employed with a minimum wire width.

With a starting point of "1" and ending point of "4," this proves that it is possible to reduce the slew without increasing power in a subthreshold clock tree. Fig. 10(a) and (b) shows that our final clock tree, "4," has smaller slew variations compared to the original above threshold design, scaled to subthreshold. Fig. 10(b) shows that the clock tree designed with Dynamic C_{MAX} selection has reduced the variations in slew from the range of 3–10 ns to the range of 3–6 ns. Additionally, the coefficient of variation for the new clock tree is 0.1579 compared to the larger dispersion of 0.2309 [Fig. 1(a)]. Lastly, Fig. 10(c) shows the final clock tree designed with Dynamic C_{MAX} methods, implemented for the r1 IBM testbench.

V. PROCESS, VOLTAGE, AND TEMPERATURE VARIATIONS

Random slew variations are induced by process variations and have the potential to be more severe in designs because they add to deterministic slew variations. The prior work on the area of subthreshold design such as [9] has shown that in the subthreshold region of operation the local variability due to effects like random dopant fluctuations can dominate the device threshold, V_{TH}, variability. Therefore, we have considered local variability in the clock buffers while simulating the random slew variations.

Using the learned techniques to design a dynamic clock tree, we have proven it is possible to reduce the slew variations without an energy penalty in the clock tree. While this has remained a focus of this paper, maintaining a stable design under process, voltage and temperature variations are extremely important in subthreshold designs. In this section we study the effect of using Dynamic C_{MAX} on the impact of PVT variation on the robustness and energy of the clock tree.

To model the effects of process variations in subthreshold we have applied an independent variation to the threshold voltage of each transistor in the clock tree. A 5000 point Monte Carlo simulation was performed using a Gaussian distribution with 3σ value of +/−10% of the nominal V_{TH}. Fig. 11(a) and (b) depicts the clock tree energy and average slew as a result of the MC simulation under process variations. There is a clear advantage to designing a tree using dynamic C_{MAX} selection, because it maintains a reduced energy and slew

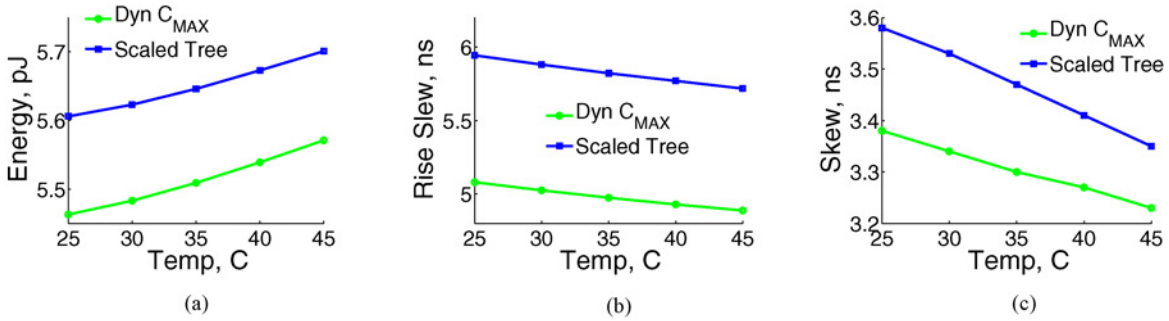


Fig. 13. (a) Energy, (b) Slew, and (c) Skew response to a temperature sweep shows that a dynamic clock tree is the preferable design.

under variations. Fig. 11(c) demonstrates Monte Carlo the skew variations at all process corners. The thing to note here is that with a smaller energy and slew, the dynamic clock tree has similar or improved skew for all corners.

A supply voltage sweep of +/−20% of the nominal V_{DD} will provide a broad range to examine the clock tree response. Fig. 12(a) shows how the resulting supply voltage sweeps impacts the clock tree energy, with the dynamic clock tree always more efficient than the scaled tree. Fig. 12(b) is more interesting, because it shows how the average slew has a non-linear dependence on voltage. Additionally, we can see at lower voltages, the slope of the dynamic clock tree is less than that of the scaled tree. This means that in subthreshold, the slew is less sensitive to supply voltage variations when a clock tree is designed with a Dynamic C_{MAX} methodology. Fig. 12(c) shows the skew dependence on voltage. In general, dynamic C_{MAX} appears to be more robust across subthreshold supply variations. Fig. 13 examines how temperature variations affect the clock energy, slew and skew. We can observe that the dynamic C_{MAX} has lower energy, slew and skew even under different temperature conditions.

VI. Conclusion

In this paper, we explained how the design of a subthreshold clock tree can impact the slew variations, which will in turn corrupt the flow of data in a logic path. This notion provided the motive to design an optimal subthreshold clock tree with slew control. We concluded that the following guidelines should be used when designing an optimal clock tree in subthreshold with slew control. The maximum allowable nodal capacitance should be small in subthreshold, minimum wire sizes should be used at all times, and the maximum nodal capacitance can be controlled dynamically to allow more slew propagation near the root of the tree while saving power. On the other hand, near the sink-nodes the maximum nodal capacitance should be reduced to better control the slew.

We presented a systematic approach combining the above three guidelines for subthreshold clock tree design that has the potential to reduce the timing metric variations by 50% or more while maintaining the power advantage of subthreshold design. Additionally, the flip-flop design will also have an impact on how severe the timing variations are. By co-designing these efforts, it is will be possible to further reduce the variations in slew, thus reducing the probability of timing violations.

VII. Appendix

The clock routing algorithm used in this paper includes two major steps: 1) abstract tree generation, and 2) slew-aware buffering and embedding. Given a set of clock sinks, we first generate an abstract tree based on the method of means and medians algorithm [26]. The objective of abstract tree generation is to decide the connection among the sink nodes, internal nodes, and the clock source while minimizing the wirelength. Then, the routing topology and geometric locations of all the nodes are determined by a two-phase slew-aware buffering and embedding method. Our method follows the classic deferred-merging and embedding flow [23] in the above-threshold clock network design. But the major difference is we insert buffers during the clock routing as well. We first visit the abstract tree by a bottom-up manner. For a pair of nodes, we create a set of feasible candidate solutions for their parent node, including the merging distances and merging styles. This bottom-up phase aims at generating zero-skew solutions, and inserting buffers so that loading capacitance of each buffer does not exceed the user-specified maximum value (C_{MAX}). The second phase is to choose the optimum solution among the candidates by visiting the abstract tree in a top-down order. The outcomes are the entire clock routing topology with the exact locations of the internal nodes, buffers, and the clock source.

References

[1] A. Wang and A. Chandrakasan, "A 180 mV FFT processor using subthreshold circuit techniques," in *Proc. Int. Solid-State Circuits Conf.*, 2004, pp. 292–293.

[2] A. Wang, A. Chandrakasan, and S. Kosonocky, "Optimal supply and threshold scaling for subthreshold CMOS circuits," in *Proc. Symp. VLSI*, 2002, pp. 5–9.

[3] B. H. Calhoun and A. Chandrakasan, "Characterizing and modeling minimum energy operation for subthreshold circuits," in *Proc. Int. Symp. Low Power Electron. Design*, 2004, pp. 90–95.

[4] B. C. Paul, A. Raychowdhury, and K. Roy, "Device optimization for digital subthreshold logic operation," *IEEE Trans. Electron Devices*, vol. 51, no. 9, pp. 300–301, Feb. 2005.

[5] N. Hedenstierna and K. O. Jeppson, "CMOS circuit speed and buffer optimization," *IEEE Trans. Comput.-Aided Des.*, vol. 6, no. 2, pp. 270–281, Mar. 1987.

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

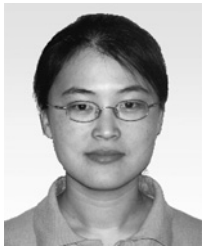
[7] N. Magen, A. Kolodny, U. Weiser, and N. Shamir, "Interconnect-power dissipation in a microprocessor," in *Proc. Int. Workshop SLIP*, 2004, pp. 7–13.

[8] T. Sakurai and A. R. Newton, "Alpha-power model, and its application to CMOS inverter delay and other formulas," *IEEE J. Solid-State Circuits*, vol. 25, no. 2, pp. 584–594, Apr. 1990.
[9] B. Zhai, S. Hanson, D. Blaauw, and D. Slyvester, "Analysis and mitigation of variability in subthreshold design," in *Proc. Int. Symp. Low Power Electron. Design*, Aug. 2005, pp. 20–25.
[10] J. Kwong and A. Chandrakasan, "Variation-driven device sizing for minimum energy sub-threshold circuits," in *Proc. Int. Symp. Low Power Electron. Design*, Oct. 2006, pp. 8–13.
[11] N. Jayakumar and S. P. Khatri, "A variation-tolerant sub-threshold design approach," in *Proc. Des. Autom. Conf.*, 2005, pp. 716–719.
[12] N. Lotze, M. Ortmanns, and Y. Manoli, "Variability of flip-flop timing at sub-threshold voltages," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2008, pp. 221–224.
[13] N. Verma and A. Chandrakasan, "Nanometer MOSFET variation in minimum energy subthreshold circuits," *IEEE J. Solid-State Circuits*, vol. 55, no. 1, pp. 163–174, Jan. 2008.
[14] *Predictive Technology Model* [Online]. Available: http://www.eas.asu.edu/∼ptm
[15] R. S. Tsay, "Exact zero skew," in *Proc. Int. Conf. Comput.-Aided Des.*, 1991, pp. 336–339.
[16] J. Rabaey, A. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits*. Englewood Cliffs, NJ: Prentice-Hall, Jan. 2003.
[17] G. E. Tellez and M. Sarrafzadeh, "Minimal buffer insertion in clock trees with skew and slew rate constraints," *IEEE Trans. Comput.-Aided Design*, vol. 16, no. 4, pp. 333–342, Apr. 1997.
[18] C. J. Alpert, A. B. Kahng, L. Bao, I. I. Mandoiu, and A. Z. Zelikovsky, "Minimum buffered routing with bounded capacitive load for slew rate and reliability control," *IEEE Trans. Comput.-Aided Design*, vol. 22, no. 3, pp. 241–253, Mar. 2003.
[19] C. Albrecht, A. B. Kahng, L. Bao, I. I. Mandoiu, and A. Z. Zelikovsky, "On the skew-bounded minimum-buffer routing tree problem," *IEEE Trans. Comput.-Aided Design*, vol. 22, no. 7, pp. 937–945, Jul. 2003.
[20] 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.*, vol. 26, no. 11, pp. 2009–2022, Nov. 2007.
[21] J. Kwong and A. P. Chandrakasan, "A 65 nm sub-Vt microcontroller with integrated SRAM and switched capacitor DC-DC converter," *IEEE J. Solid-State Circuits*, vol. 44, no. 1, pp. 115–126, Jan. 2009.
[22] *Intel Products* [Online]. Available: http://ark.intel.com
[23] K. Boese and A. Kahng, "Zero-skew clock routing trees with minimum wirelength," in *Proc. 5th Annu. IEEE Int. ASIC Conf. Exhibit*, Sep. 1992, pp. 17–21.
[24] R. S. Tsay, "Exact zero skew," in *Proc. Int. Conf. Comput.-Aided Des.*, 1991, pp. 336–339.
[25] M. Seok, D. Blaauw, and D. Sylvester, "Clock network design for ultra-low power applications," in *Proc. Int. Symp. Low-Power Electron. Des.*, Aug. 2010, pp. 271–276.
[26] M. Jackson, A. Srinivasan, and E. Kuh, "Clock routing for high performance ICs," in *Proc. ACM Des. Automat. Conf.*, 1990, pp. 573–579.

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 working toward the Ph.D. degree in electrical and computer engineering from the School of Electrical and Computer Engineering, 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 is currently sponsored by the Graduate Research Fellowship of the National Science Foundation.



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

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

Ms. Zhao was the recipient of the Best Paper Award Nomination at the International Conference on Computer-Aided Design in 2009.

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, Los Angeles, in 1994, 1997, and 2000, respectively.

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

Dr. Lim received the National Science Foundation Faculty Early Career Development Award in 2006. He was on the Advisory Board of the ACM Special Interest Group on Design Automation (SIGDA) from 2003 to 2008 and received the ACM SIGDA Distinguished Service Award in 2008. He has served the technical program committees of several conferences on electronic design automation, including the ACM Design Automation Conference and the IEEE International Conference on Computer-Aided Design. He has been leading the Cross-Center Theme on 3-D Integration for the Focus Center Research Program since 2010.

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 is currently an Assistant Professor with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta. Prior to joining the Georgia Institute of Technology, he was with the IBM T. J. Watson Research Center, Yorktown Heights, NY, as a Research Staff Member and worked on high-performance circuit design and technology-circuit co-design focusing primarily on static random access memories. His current research interests include analysis and design of low-power and robust circuits in nanometer technologies.

Dr. Mukhopadhyay was a recipient of the NSF CAREER Award in 2011, the IBM Faculty Partnership Award for 2009 and 2010, the SRC Inventor Recognition Award in 2009, the SRC Technical Excellence Award in 2005, the IBM Ph.D. Fellowship Award for 2004 to 2005, the Best in Session Award at 2005 SRC TECNCON, and the Best Paper Award at the 2003 IEEE Nano and 2004 International Conference on Computer Design.