High Performance Low Swing Clock Tree Synthesis with Custom D Flip-Flop Design

Can Sitik, Leo Filippini
Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104, USA
E-mail:{as3577, lf348}@drexel.edu

Emre Salman
Electrical and Computer Engineering
Stony Brook University
Stony Brook, NY 11794, USA
E-mail: emre.salman@stonybrook.edu

Baris Taskin
Electrical and Computer Engineering
Drexel University
Philadelphia, PA 19104, USA
E-mail: taskin@coe.drexel.edu

***Abstract*—Low swing clocking is a low power design methodology that scales the clock voltage to decrease power consumption of the clock distribution networks, with an expected degradation in the performance. In this work, a novel low swing clock tree synthesis methodology is combined with a custom low swing clock-aware D flip-flop (DFF) design. The low swing clocking serves to reduce the power dissipation whereas the custom low swing-aware DFF serves to preserve the performance of the IC. The experimental results performed on the three largest circuits of ISCAS'89 benchmarks operating at 1GHz in the 32nm technology show that the proposed methodology can achieve an average of 16% power savings in the clock tree compared to its full swing counterpart, while satisfying the same clock skew (50ps) and slew (150ps) constraints at the worst case corner of operation. Moreover, the clock-to-output delay of the low swing DFF does not increase compared to traditional full swing DFF, while consuming only 1% more power.**

I. Introduction

Clock distribution networks constitute an important part of the IC design due to their direct effect on the performance and the power consumption [1]. Thus, the tradeoff between the power consumption and the performance of clock networks is well studied in the literature [2–5]. Clock trees are good candidates for low power applications [4, 5], where the power budget is the main concern, whereas high performance architectures investigate different clock distribution network topologies to target higher timing performance with higher power consumption [2, 3]. Low swing clocking is one of the investigated techniques for low power design [6–10]. In low swing clocking, the power savings on the clock buffers and interconnects obtained through the voltage scaling trades off timing performance, due to higher delay and slower switching (slew) at the clock sinks.

The current art of low swing clocking is effective for low power applications that do not demand performance. However, the applicability of low swing clocking remains limited for high performance designs due to following issues: i) Larger number of buffers and greater interconnect delay increase the insertion delay on the clock path, leading to excessive clock skew and/or excessive buffering to minimize the skew, ii) increased switching time at the clock buffer output (clock slew), especially in the sub-45nm designs, leading to excessive buffering to satisfy timing constraints, iii) the effect of the low swing clock on the local timing and DFF power consumption when synchronizing DFF cells running at full data swing, and iv) the decrease in the expected power savings of the low swing clocking operation, induced by the efforts to minimize performance degradation. A previous work introduces the use of level-shifting buffers at the final level of the low swing clock tree in order to synchronize full swing flip flops [6]. Level-shifting buffers at the final level restore the clock signal so that the flip-flops are driven at the full swing, therefore, the issue (iii) is addressed. However, restoring the clock signal to full swing degrades the power savings as the capacitance at the final level of the clock tree, which constitutes most of the total capacitance, is charged to full swing, failing to address issue (iv). Another work considers a combination of a low skew clock tree design scheme and a low swing latch design, addressing issues (i) and (iv) [8]. However, the clock-to-output delay on this latch is significantly increased, and consequently, it is stated in this work that these latches can only be used for non-critical paths, failing to address issue (iii). A third work proposes custom buffer topologies of full-swing-to-reduced-swing, reduced-swing-to-reduced-swing and reduced-swing-to-full-swing in order to obtain a low swing clock tree between the full swing clock source to full swing flip-flop cells [7]. This work fails to address the issue (iv), by restoring the clock signal to full swing at the final level of the clock tree and degrades power savings. Moreover, the output low swing voltage levels of the full-swing-to-reduced-swing and reduced-swing-to-reduced-swing buffers depend on the output capacitances of the buffers, therefore it is hard to obtain a stable low swing voltage level. A fourth work introduces a low swing clock tree design methodology that addresses the issues (i), (ii) and (iv) [10]. This work fails to address the issue (iii) as it does not consider clock slew or the increased impact of DFF power in low swing clocking for technologies scaled to sub-45nm.

In this work, these four issues are addressed simultaneously through a combination of a novel clock slew-aware low swing clock tree synthesis method with a low swing clock-aware DFF design. The low swing DFF is designed to satisfy the same clock-to-output delay and the power consumption compared to a typical full swing DFF topology, for the same clock slew at both the full and low swing operation. Then, the clock tree synthesis is performed satisfying the same clock slew

978-1-4799-3765-3/14 $31.00 © 2014 IEEE
DOI 10.1109/ISVLSI.2014.53

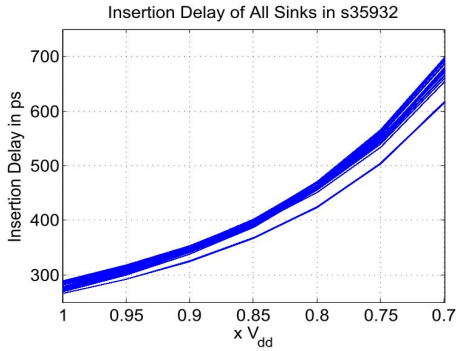


Fig. 1. Insertion delay profile of all 1728 sinks in s35932 with low swing clocking

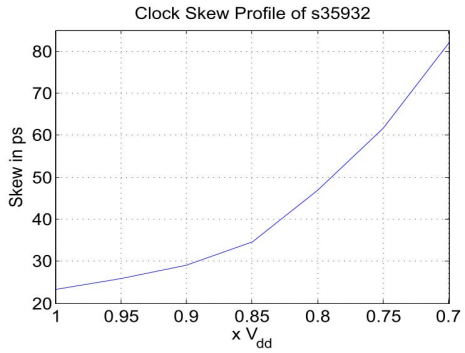


Fig. 2. The clock skew profile of s35932 with low swing clocking

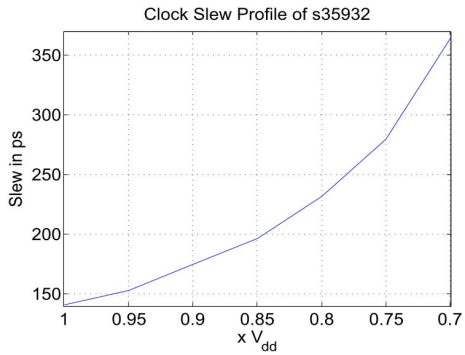


Fig. 3. The clock slew profile of s35932 with low swing clocking

constraint, via a novel clock slew-aware buffering scheme. The proposed novel methodology has the following features:

1) The custom low swing DFF design is performed at the fixed system slew constraint for both power and clock-to-output delay compatibility with the traditional full swing DFF topology,
2) Accurate delay and slew characterizations of clock buffers and interconnects are performed at low swing voltage levels that are not available in the lookup tables of the timing library,
3) A slew-aware buffering scheme is developed that accounts for not only capacitance but also resistance of the clock interconnects for better accuracy at sub-45nm,
4) The increase in the insertion delay due to low swing operation on the clock tree is methodically compensated by embedding a buffer insertion/wire snaking scheme within the clock tree synthesis for skew minimization.

The output of this methodology is a low swing clock tree, running at the same frequency, satisfying the same clock skew, clock slew constraints with the same clock-to-output delay as its full swing counterpart, while saving significant power. The proposed methodology is implemented within the IC design flow, by including the listed features into an automated clock tree synthesis (CTS) algorithm. Thus, it is highly practical for easy integration into existing industrial tools.

The rest of the paper is organized as follows. The preliminaries of low swing clocking are briefed in Section II. In Section III, the proposed methodology is introduced. The experimental results are presented in Section IV. The paper is finalized with concluding remarks in Section V.

II. Preliminaries

In Section II-A, the changes in the clock buffer and interconnect timing in the low swing operation are discussed. In Section II-B, the power consumption of the clock tree and the sink DFF cells are investigated at various voltage levels.

A. Delay and Slew Characteristics in Low Swing Clock Trees

In low swing clocking, the lower power supply on the clock buffers increases the buffer delay and its output switching time (slew). These increased buffer and interconnect delays increase the insertion delay of the clock branches, increasing the clock skew under variations. In order to highlight this change, a sample circuit, s35932 of ISCAS'89 benchmarks is selected. A clock tree with a 150ps slew constraint and a 50ps skew constraint at 1GHz is built at the full swing (0.95V at the worst case corner of this technology) using the buffers from SAED32nm library [11] at 125°C and SS corner. The clock voltage is scaled from 100% to 70% of full V_{dd} with 5% decrements. It is shown in Figure 1 that the insertion delay increases significantly (from ≈250ps to ≈650ps) when the clock voltage is scaled to low swing values. Furthermore, the spread of the insertion delay margin increases. Thus, the effect on the clock skew is exacerbated: As shown in Figure 2, the clock skew is increased from ≈25ps to ≈80ps. Given the clock skew constraint of 50ps, it is observed that the timing is violated when the low swing clocks ($< 80\%V_{dd}$ in this example) are used on this clock topology originally operational at full swing clocking.

Another big challenge that arises in low swing clocking, particularly for sub-45nm technologies, is the negative effect of high interconnect resistance on the clock slew, which causes an increase in the number of repeaters (i.e. clock buffers). In order to show this effect, s35932 of ISCAS'89 benchmarks is selected to observe the change in the clock slew with a varying low swing voltage level, from 100% to 70% of V_{dd} with 5% decrements. It is observed in Figure 3 that the clock slew is

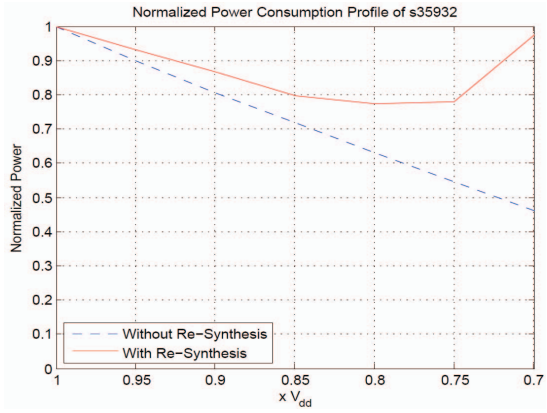


Fig. 4. Power profile of s35932 with low swing clocking with and without a re-synthesis. When the clock voltage is scaled without a re-synthesis to fix timing violations, significant power savings can be obtained (blue dashed curve) at the expense of clock slew degradation shown in Figure 3. A re-synthesis of the clock tree is necessary in order to satisfy the slew constraint (150ps for this example) at each voltage level, leading to the tradeoff in power savings peaking at 75% of V_{dd} for this circuit (red solid curve).

more than doubled, switching from a full swing operation at 100% to a low swing level at 70% of V_{dd}. The same slew constraint (e.g. as that in full swing clocking) is possible at low swing clocking through buffering the existing topology. However, it introduces an extravagant power dissipation due to the necessity of high number of clock buffers to satisfy slew constraint each clock sink. Moreover, the increase in the number of clock buffers increases the number of buffer levels in the clock tree, therefore the insertion delay (and clock skew) increases further than the profile shown in Figure 1, where the slew violation was not considered (thus not compensated for by inserting buffers). Due to this large discrepancy in slew and skew, and the failure to efficiently correct these violations through buffering, a re-synthesis of the clock tree is necessary at low swing voltages.

B. Power Characteristics in Low Swing Clocking

The potential power savings in low swing clocking depend not only on the power on the clock branches but also on the DFF cells (or on all synchronous components) driven by the clock signal. One significant effect of low swing clocking occurs when the low swing clock signal synchronizes a traditional master-slave DFF cell (i.e. running at full data swing). The low swing voltage on the clock pin prevents certain PMOS transistors in the DFF cell from completely turning off, resulting in short circuit power consumption, and a possible failure. Thus, this increase in the DFF power consumption limits the impact of low swing clocking. In this paper, a novel low swing DFF is introduced (described in Section III-A) in order to address this issue.

As the negative effect of low swing clocking on DFF power is addressed with the use of the proposed low swing DFF, lower voltage scaling becomes feasible. However, as shown in Figure 3 (and discussed in Section II-A), the clock slew increases significantly at lower clock voltage swings. The alternative to scaling the voltage on the original full-swing clock network is the re-synthesis of a clock network at low swing voltage nodes. In order to identify the performance of the either approach, the power consumption profile of s35932 from ISCAS'89 benchmarks is analyzed in two cases:

1) Scaling the clock voltage of a clock tree synthesized at full swing, allowing clock slew constraints to violate (i.e. status of the previous art),
2) Re-synthesizing the clock tree at each voltage level in a slew-aware manner.

These two profiles are shown in Figure 4, at 1GHz frequency and 150ps (15% of the clock period) at the worst case corner of operation. As shown in Figure 4, scaling the clock voltage directly on the original full-swing clock tree can achieve significant power savings, at the expense of extravagant clock slew shown in Figure 3. For high performance ICs where the slew constraint is identical to the full swing operation, the higher number of clock buffers that are necessary to satisfy the slew constraint outweighs the savings obtained through low swing clocking. In Figure 4, the power savings start leveling around 80% of V_{dd} and the power consumption start increasing from 75% to 70% of V_{dd}. Thus, 75% of V_{dd} is selected to be the low swing voltage level in this experimental setup.

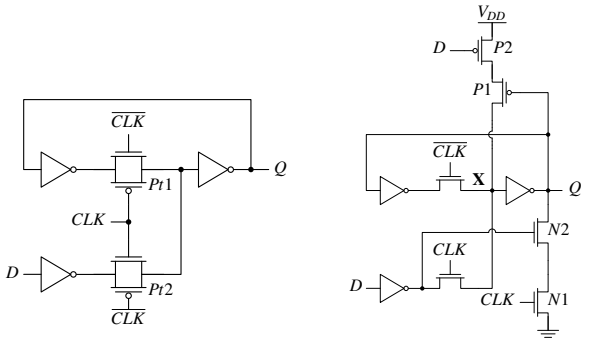
III. Methodology

The proposed methodology has 3 mains steps:

1) The design of the custom low swing DFF at the target slew constraint (Section III-A),
2) Timing characterization of buffers and interconnects at the target technology, only necessary when the timing information is not readily available at selected low swing level (Section III-B),
3) Low swing clock tree synthesis with the skew and the slew constraints (Section III-C).

A. Low Swing DFF Design

As highlighted in Section II-B, the design of a low swing DFF is necessary in order to preserve the power savings of the low swing operation, limiting the short circuit power that is exacerbated when a typical DFF cell is used with low swing clocking. Furthermore, the transistors within the low swing DFF are sized in order to obtain the same clock-to-output delay as the full swing DFF, despite running at a low swing level. A typical latch topology in the traditional master-slave latch-based DFF cell is investigated here. Figure 5(a) shows the classical implementation of a latch using transmission gates. In low-swing clock applications, the latch in Figure 5(a) can present issues. In particular, the PMOS transistors $Pt1$ and $Pt2$ in the transmission gates fail to completely turn off when CLK is high, hence providing a conductive path even in the idle states. For these reasons, the low swing latch shown in Figure 5(b) is proposed. Without transistors $N1$, $N2$, $P1$, and $P2$, this latch is equivalent to the one in Figure 5(a), other than the PMOS transistors in the transmission gates. Transistor $P1$ is used as a level restorer, and it ensures that node X charges up to V_{DD} when D is low. However, node Q must be lower



(a) Typical full-swing latch (b) Novel low-swing latch

Fig. 5. Typical latch topology vs. Novel low swing latch

than V_{DD} in order for $P1$ to start conducting. Thus, transistor $N2$ is used to discharge node Q when D is low so that $P1$ can restore the level of node X. $N1$ guarantees that $N2$ changes Q only when CLK is high. Although $P1$ helps X charge up to full V_{dd}, it is detrimental during discharge operation. Transistor $P2$ ensures that $P1$ is off when node X is being discharged (this happens when D and CLK are high). After setting up this topology, the transistors are sized considering both clock-to-output delay equivalence to full swing case and the power consumption.

B. Timing Characterization of Clock Buffers and Interconnects

This step is necessary when the timing models of the clock buffers and the interconnects are not readily available. Two methods to perform delay and slew characterizations of i) clock buffers and ii) clock interconnects are proposed as follows.

It is known that the delay $D(B)$ and the output slew $Slew_{out}(B)$ of a clock buffer B depend input slew $slew_{in}(B)$ and output capacitance $Cap_{out}(B)$ of the buffer. In SPICE-accurate simulations, it is observed that a linear fit is possible for both delay $D(B)$ and the output slew $Slew_{out}(B)$ estimations that results in an accuracy within ≈1ps for each buffer. In particular, the delay of a clock buffer $D(B)$ can be written as:

$$D(B) = K_{slew}^{delay} \times Slew_{in}(B) + K_{cap}^{delay} \times Cap_{out}(B) + K_{delay}, \quad (1)$$

where K_{slew}^{delay} and K_{cap}^{delay} are the coefficients for the input slew $Slew_{in}(B)$ and output capacitance $Cap_{out}(B)$, respectively, for delay computation. K_{delay} is the intrinsic delay of the buffer. As for the output slew $Slew_{out}(B)$, it is observed that the input slew does not have a significant effect; therefore the output slew of a buffer B can be estimated as:

$$Slew_{out}(B) = K_{cap}^{slew} \times Cap_{out}(B) + K_{slew}, \quad (2)$$

where K_{cap}^{slew} is the coefficient of the output capacitance for slew computation and K_{slew} is the intrinsic output slew. Given the clock slew constraint (for instance 150ps), these coefficients are obtained by sweeping the input slew and the output capacitance around the slew constraint.

The wire delay can be estimated with the well-known Elmore delay [12] with sufficient accuracy. The slew degradation on a wire segment T can be estimated using the Bakoglu [13] metric simplified for an ideal wire input with zero input slew:

$$Slew_{ideal}(T_i, T_f) = \ln 9 \times D(T_i, T_f) \quad (3)$$

where $D(T_i, T_f)$ is Elmore delay of the wire segment T from its initial point T_i to final point T_f. This result can be extended for wires with non-zero input slews, by using the PERI model estimation [14]. In this estimation, the output slew of the wire segment T is estimated at its final position T_f as:

$$Slew_{wire}(T_f) = \sqrt{Slew_{wire}(T_i)^2 + Slew_{ideal}(T_i, T_f)^2} \quad (4)$$

where $Slew_{wire}(T_i)$ is the input slew of the wire segment T estimated at its initial position T_i. In a buffered RC network, the output of a buffer is the input of a wire segment and vice versa. Thus, Eq. (1) to Eq. (4) are used to estimate the slew and the delay propagation on the clock tree.

C. Low Swing Clock Tree Synthesis

In this section, the proposed slew-aware low swing clock tree synthesis methodology is introduced. The algorithm (Algorithm 1) adopts the well-known zero-skew-tree deferred merge embedding (ZST-DME) algorithm [15] to merge two nodes into one at each step. The merging cost is inspired by [5], which considers both the capacitance and the delay as the cost metric. In this work, this cost is modified to consider the slew and the delay, in order to accurately capture the impact of higher wire resistance for sub-45nm technologies. In Algorithm 1, lines 5-15 identify whether a merging pair is feasible. If a feasible pair is identified, lines 16-30 describe the merge process including a novel embedded skew minimization scheme. If a feasible pair is not identified, a buffering process is proposed in lines 31-36 to help satisfy the slew constraint (e.g. that caused the infeasible merge process). The feasibility is monitored by checking if the slew constraint can be satisfied with the presence of a buffer at the temporary merging point $T_{i,j}$ of child nodes i and j by calculating the maximum slew $T_{i,j}$ produces, using Eq. (2), Eq. (3) and Eq. (4) (Lines 7-10). If no feasible point is found, the buffers are inserted at the unmerged nodes, and their capacitance, delay and slew constraint parameters are updated (Lines 32-35). If a feasible pair is available (i.e. satisfying the slew constraint) for merging (Line 16), the one with the minimum cost (recorded at Line 11) is initialized as node k to merge child nodes i and j at this new node (Lines 17-20). After the maximum and the minimum delay from k to the child nodes are updated, it is checked if the difference between the maximum and the minimum is larger than the skew constraint $skew_{const}$ (Line 21). The case of a skew violation is solved through moderate wire snaking for small violations and through buffer insertion for larger violations: If the skew violation is larger than the intrinsic buffer delay K_{delay}, buffer insertion is preferred versus exorbitant amounts of wire snaking that would have been necessary (Lines 22-23). Otherwise, wire snaking is performed (Lines 24-25).

Algorithm 1 Slew-Aware Low Swing Clock Tree Synthesis

Input: Buffer library, Timing models at the voltage level, Skew and slew constraints ($skew_{const}$,$slew_{const}$)
Output: Clock buffer and interconnect locations

Initialize nodes, $Num\_of\_unmerged = Num\ of\ sinks$
$D^{max}(i)$=$D^{min}(i)$=0, $slew_{const}(i)$=$slew_{const}$ for each node i
while $Num\_of\_unmerged > 1$ **do**
  $Cost_{curr}$=∞
  for i *in Nodes* **do**
    for j *in Nodes* **do**
      $Slew_{out}(T_{i,j}) = K_{cap}^{slew} \times Cap_{out}(T_{i,j}) + K_{slew}$
      $X = max[Slew_{ideal}(T_{i,j},i),\ Slew_{ideal}(T_{i,j},j)]$
      $Slew_{wire}(T_{i,j}) = \sqrt{X^2 + Slew_{out}(T_{i,j})^2}$
      if $Cost(i,j) < Cost_{curr}$ && $Slew_{wire}(T_{i,j}) < Min[slew_{cons}(i), slew_{cons}(j)]$ **then**
        $Cost_{curr}$=Cost(i,j)
        $Temp_{slew}$=$Slew_{wire}(T_{i,j})$
      end if
    end for
  end for
  if $Cost_{curr}$!= ∞ **then**
    Num_of_unmerged$--$
    Initialize new node k=$T_{i,j}$
    $D^{max}(k)$=max[$D^{max}(i)$+D(i,k),$D^{max}(j)$+D(j,k)]
    $D^{min}(k)$=min[$D^{min}(i)$+D(i,k),$D^{min}(j)$+D(j,k)]
    while $D^{max}(k)$-$D^{min}(k) > skew_{const}$ **do**
      if $D^{max}(k)$-$D^{min}(k) > K_{delay}$ **then**
        Insert a buffer at the lower delay node
      else
        Apply wire snaking at the lower delay node
      end if
      Update $D^{max}(k)$ and $D^{min}(k)$ using Eq. (1)
      Update $Slew_{const}(i)$ and $Slew_{const}(j)$
    end while
    $Slew_{const}(k)$= $\sqrt{min[Slew_{const}(i), Slew_{const}(j)]^2 - Temp_{slew}{}^2}$
  else
    for i *in Unmerged Nodes* **do**
      Insert buffer, update $D^{max}(i)$, $D^{min}(i)$ using Eq. (1)
      $slew_{const}(i)$=$slew_{const}$
    end for
  end if
end while

This embedded skew minimization scheme helps build a skew balanced clock tree at each level; therefore it has a potential to minimize the buffering/wiring cost at the upper levels of the clock tree, which is highlighted in the experimental results in Section IV. This procedure continues until the number of unmerged nodes is one, which is the source of the clock tree.

IV. Experimental Results

The proposed algorithm is implemented in *Perl* and the output circuits are tested using HSPICE of *Synopsys* with SAED32nm technology [11] at 1GHz. Three largest circuits from ISCAS'89 benchmarks are selected (# of sinks ranging from 1238 to 1728) for experimental analysis. It is important to note here that 1) ISPD'10 benchmarks cannot be used for this work, as they do not contain any DFF information, rather they have capacitance information to model the clock pin of DFFs and 2) ISCAS'89 benchmarks have comparable circuit size with ISPD'10 benchmarks (1728 vs. 2249 for the largest number of sinks). The logic synthesis of the RTL-level netlists is performed using *Design Compiler* of *Synopsys* and the placement is performed using *IC Compiler* of *Synopsys*. The largest buffer from SAED32nm library (NBUFFX32) [11] is used as the clock buffer. The wire models are adopted from [16], whose per unit values are R=8Ω/μm and C=0.2fF/μm. The clock skew constraint is set to 50ps (5% of the period) and the clock slew constraint is set to 150ps (15% of the period) at the worst case corner of operation. The low swing voltage level is set to 75% of V_{dd}, as explained in Section II-B. The experimental results are then verified at 3 corners:

1) Worst Corner: V_{dd}=0.95V, 125°C and SS transistors,
2) Nominal Corner: V_{dd}=1.05V, 25°C and TT transistors,
3) Best Corner: V_{dd}=1.16V, -40°C and FF transistors,

where at each corner, the low swing voltage level is set to 75% of the V_{dd} of that corner (for example it is set to 0.95x0.75=0.7125V at the worst corner). In order to highlight the effectiveness of the proposed slew-aware clock tree synthesis methodology, the power consumption, clock skew, clock slew and the clock-to-output delay are compared against a full swing clock tree synthesized with the same slew-aware algorithm, and a custom implementation of the low swing clock tree using the method in [10], shown in Table I. As shown in Table I, the proposed low swing clock tree synthesis method can satisfy both skew (50ps) and slew (150ps) at the worst case, unlike the method in [10] with the SAED32nm technology. Although the method in [10] can achieve a significant 45% savings in the clock tree power, it critically violates the timing constraints. The slew is almost doubled; therefore resulting in a potentially non-functional circuit due to slew violations. It is also shown in Table I that this violation of the slew constraint causes an average of 142.8ps degradation in the clock-to-output delay, which is as significant as 14.3% of the clock period at 1GHz. In the proposed methodology, both skew and slew constraints are satisfied, resulting in an average of 2ps smaller clock-to-output delay, compared to its full swing counterpart. Furthermore, it provides significant power savings of 16% on average.

In order to highlight the compatibility of the novel low swing DFF, its power consumption is compared to the traditional full swing DFF, which is shown in Table II. The design of the novel low swing DFF enables the low swing clocking operation by i) keeping the clock-to-output delay of the low swing DFF the same and ii) limiting the increase in DFF power consumption to 1% compared to its full swing counterpart at the worst case corner. This novelty addresses a big concern in low swing clocking under PVT variations.

TABLE I
THE COMPARISON OF CLOCK TREE POWER (CP), CLOCK SKEW (SK.), CLOCK SLEW (SL.) AND CLOCK-TO-OUTPUT (C2Q) DELAY OF A FULL SWING CLOCK TREE, THE PREVIOUS WORK IN [10] AND THE PROPOSED CLOCK SLEW-AWARE LOW SWING CLOCK TREE AT THE WORST CASE CORNER. *SAT* INDICATES THE SKEW (50PS) OR THE SLEW (150PS) CONSTRAINT IS SATISFIED, *VIO* INDICATES VIOLATION

Circuits	Full Swing Clock Tree				[10]				Proposed Low Swing Clock Tree			
	CP(mW)	Sk.(ps)	Sl.(ps)	C2Q(ps)	CP(mW)	Sk.(ps)	Sl.(ps)	C2Q(ps)	CP(mW)	Sk.(ps)	Sl.(ps)	C2Q(ps)
s38584	3.82	38.9	145.6	111.9	2.14	40.4	268.1	249.5	3.05	46.4	137.8	110.6
s38417	4.17	22.9	148.5	112.7	2.28	45.2	283.6	258.5	3.55	47.1	139.3	110.0
s35932	4.04	23.3	145.4	112.3	2.24	48.8	284.0	257.3	3.48	48.5	141.5	110.3
	Avg. Change Compared to Full Swing				-45%	SAT	VIO	+142.8	-16%	SAT	SAT	-2.0

TABLE II
THE COMPARISON THE POWER CONSUMPTION OF THE TRADITIONAL FULL SWING DFF AND THE PROPOSED NOVEL LOW SWING DFF

Circuits	Power Consumption (mW)	
	Full Swing DFF	Low Swing DFF
s38584	1.40	1.42
s38417	1.66	1.68
s35932	1.96	1.99
	Average Change	+1%

TABLE III
CLOCK SKEW, NUMBER OF CLOCK BUFFERS AND TOTAL CLOCK INTERCONNECT LENGTH WITH AND WITHOUT EMBEDDING SKEW MINIMIZATION SCHEME INTO CTS

Circuits	Clock Skew(ps)		Num of Buffers		Inter. Length(μm)	
	Without	With	Without	With	Without	With
s38584	144.1	46.4	60	59	13668	13719
s38417	191.0	47.1	67	70	15546	15582
s35932	82.6	48.5	69	68	15120	15171
Range of Change			-1/+3		+36/+51	

In order to highlight the effectiveness of the skew minimization scheme embedded into the algorithm (Lines 21-29 in Algorithm 1), the clock skew, the number of buffers and the total interconnect length are presented in Table III. As stated in Section III-C, embedding the skew minimization scheme within the CTS algorithm may increase but also decrease the overall clock buffer and interconnect cost. The skew minimization scheme embedded into the algorithm synthesizes the clock tree by limiting the clock skew at each merging segment; therefore having a balanced segment may lead to less buffer and capacitance cost at the upper levels of the clock tree, unlike a post-CTS clock skew minimization scheme. Actually, this phenomenon can be observed for s38584 and s35932, where the number of buffers is one less than the scheme without the skew minimization despite the slight increase in the interconnect length. s38417 is the only case where both the buffer and the interconnect costs increase, where the highest clock skew decrease is observed (from 191.0ps to 47.1ps).

V. CONCLUSION

In this paper, a novel low swing clock tree synthesis method is combined with a novel low swing clock-aware DFF design targeting power savings for high performance designs. The previous art of low swing clocking schemes can achieve significant power savings with a performance degradation in timing. In this work, a slew-aware low swing clock tree synthesis method is introduced in order to satisfy the same clock skew and slew constraint as the full swing clock tree, while saving substantial amount of power. Furthermore, a novel low swing DFF is designed to preserve the local timing performance (i.e. clock-to-output delay) with the same power budget as a typical full swing DFF. The proposed methodology is implemented within the IC design flow, therefore, it is highly practical for automation purposes.

REFERENCES

[1] R. Shelar and M. Patyra, "Impact of local interconnects on timing and power in a high performance microprocessor," *IEEE Transactions on Computer-Aided Design (TCAD) of Integrated Circuits and Systems*, vol. 32, no. 10, pp. 1623–1627, 2013.

[2] N. Kurd and et al., "A multigigahertz clocking scheme for the Pentium 4 microprocessor," *IEEE Journal of Solid-State Circuits (JSSC)*, vol. 36, no. 11, pp. 1647–1653, Nov. 2001.

[3] C. Sitik and B. Taskin, "Multi-voltage domain clock mesh design," in *Proceedings of IEEE International Conference on Computer Design (ICCD)*, Sept. 2012, pp. 201–206.

[4] D. J. Lee, M. C. Kim, and I. Markov, "Low-power clock trees for CPUs," in *Proceedings of IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, Nov. 2010, pp. 444–451.

[5] R. Chaturvedi and J. Hu, "Buffered clock tree for high quality IC design," in *Proceedings of IEEE International Symposium on Quality Electronic Design (ISQED)*, March 2004, pp. 381–386.

[6] J. Pangjun and S. Sapatnekar, "Low-power clock distribution using multiple voltages and reduced swings," *IEEE Trans. on Very Large Scale Integration (TVLSI) Systems*, vol. 10, no. 3, pp. 309–318, June 2002.

[7] H. A. Asgari, F., and M. Sachdev, "A low-power reduced swing global clocking methodology," *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems*, vol. 12, no. 5, pp. 538–545, May 2004.

[8] Q. Zhu and M. Zhang, "Low-voltage swing clock distribution schemes," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS)*, May 2001, pp. 418–421.

[9] D. Markovic, J. Tschanz, and V. De, "Feasibility study of low-swing clocking," in *the Proceedings of the International Conference on Microelectronics*, vol. 2, May 2004, pp. 547–550.

[10] C. Sitik and B. Taskin, "Skew-bounded low swing clock tree optimization," in *Proceedings of ACM Great Lakes Symposium on VLSI (GLSVLSI)*, May 2013, pp. 49–54.

[11] *Synopsys 32nm Generic Library*, Synopsys Inc., 2012.

[12] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, 1948.

[13] H. B. Bakoglu, *Circuits, Interconnects, and Packaging for VLSI*. Addison-Wesley, 1990.

[14] C. Kashyap, C. Alpert, F. Liu, and A. Devgan, "Closed-form expressions for extending step delay and slew metrics to ramp inputs for RC trees," *IEEE Transactions on Computer-Aided Design (TCAD) of Integrated Circuits and Systems*, vol. 23, no. 4, pp. 509–516, April 2004.

[15] K. Boese and A. Kahng, "Zero-skew clock routing trees with minimum wirelength," in *Proceedings of IEEE International ASIC Conference and Exhibit*, 1992, pp. 17–21.

[16] S. Natarajan and et al., "A 32nm logic technology featuring 2nd-generation high-k + metal-gate transistors, enhanced channel strain and 0.171μm^2 SRAM cell size in a 291Mb array," in *IEEE International Electron Devices Meeting (IEDM)*, 2008, pp. 1–3.