FinFET-Based Low-Swing Clocking

CAN SITIK, Drexel University
EMRE SALMAN, Stony Brook University
LEO FILIPPINI, Drexel University
SUNG JUN YOON, Stony Brook University
BARIS TASKIN, Drexel University

13

A low-swing clocking methodology is introduced to achieve low-power operation at 20nm FinFET technology. Low-swing clock trees are used in existing methodologies in order to decrease the dynamic power consumption in a trade-off for 3 issues: (1) the effect of leakage power consumption, which is becoming more dominant when the process scales sub-32nm; (2) the increase in insertion delay, resulting in a high clock skew; and (3) the difficulty in driving the existing DFF sinks with a low-swing clock signal without a timing violation. In this article, a FinFET-based low-swing clocking methodology is introduced to preserve the dynamic power savings of low-swing clocking while minimizing these three negative effects, facilitated through an efficient use of FinFET technology. At scaled performance constraints, the proposed methodology at 20nm FinFET leads to 42% total power savings (clock network+DFF) compared to a FinFET-based full-swing counterpart at the same frequency (3 GHz), thanks to the dynamic power savings of low-swing clocking and 3% power savings compared to a CMOS-based low-swing implementation running at the half frequency (1.5 GHz), thanks to the leakage power savings of FinFET technology.

Categories and Subject Descriptors: B.7.1 [**Integrated Circuits**]: Types and Design Styles

General Terms: Design

Additional Key Words and Phrases: FinFET, clock tree, low power, VLSI, EDA

ACM Reference Format:
Can Sitik, Emre Salman, Leo Filippini, Sung Jun Yoon, and Baris Taskin. 2015. FinFET-based low-swing clocking. ACM J. Emerg. Technol. Comput. Syst. 12, 2, Article 13 (August 2015), 20 pages.
DOI: http://dx.doi.org/10.1145/2701617

1. INTRODUCTION

In sub-32nm CMOS technologies, conventional techniques such as nonuniform channel doping are not sufficiently effective mechanisms to reduce the sub-threshold current [Salman and Friedman 2012]. According to ITRS [ITRS Technology Working Groups 2010], enhanced electrostatic control of the transistor channel is a critical requirement for sub-25nm technologies since the threshold voltage should be decreased to produce a sufficient gate overdrive voltage without increasing the sub-threshold current. Vertical multigate devices, and particularly FinFET technology, have been used to

This research is supported in part by Semiconductor Research Corporation (SRC) under contract nos. 2013-TJ-2449 and 2013-TJ-2450.
S. J. Yoon is currently affiliated with Texas A&M University.
Authors' addresses: C. Sitik (corresponding author), Department of Electrical and Computer Engineering, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104; email: as3577@drexel.edu; E. Salman, Department of Electrical and Computer Engineering, Stony Brook University, Stony Brook, NY; L. Filippini, Department of Electrical and Computer Engineering, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104; S. J. Yoon, Texas A&M University, College Station, TX; B. Taskin, Department of Electrical and Computer Engineering, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104.
Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2015 ACM 1550-4832/2015/08-ART13 $15.00
DOI: http://dx.doi.org/10.1145/2701617

achieve this objective [Doyle et al. 2003]. In FinFET devices, the gate terminal achieves enhanced control of the transistor channel, thereby significantly reducing the leakage current. This is particularly important for low- and ultra-low-power microelectronics that demand high computational performance. The availability of FinFET technology is critical to address the prohibitive leakage problem encountered at sub-32nm CMOS technologies while enabling high performance (e.g., frequency scaling). Developments of FinFET-technology-aware design techniques are emerging to address the exacerbated leakage power problem in sub-32nm CMOS technologies that enable frequency scaling. FinFET-based circuits can be used to replace the incumbent CMOS-based circuits with existing methods, leading to improvements in static (i.e., leakage) power consumption. Alternatively, in a more methodical manner, FinFET-circuit-based low-power design methodologies can be developed. Unlike a straight-forward replacement, FinFET-aware methodologies would combine the leakage power superiority of FinFETs with: (1) dynamic power minimization techniques and (2) FinFET-specific design features, such as an increased frequency of operation, changing the driving capability of cells, and the use of low gate delay of FinFET devices. This article describes one such advancement in the science of clock tree synthesis, facilitating the use of FinFET-based technology.

The proposed design methodology achieves multiple advancements in: (i) low-swing clock tree synthesis, (ii) a novel DFF topology to accommodate low-swing clocks, and (iii) a FinFET-based clock buffer and DFF modeling for SPICE-accurate simulation of 20nm FinFET technology. The outcome is a design methodology for low-swing clock tree synthesis in 20nm FinFET technology that demonstrates: (i) substantial dynamic power savings of 36% in the clock tree power (excluding DFF) and 42% in total power (clock network+DFF) compared to its full-swing complement in FinFET technology, compounded with (ii) substantial leakage power savings, leading to a total power savings of 3% while doubling (i.e., 2×) the frequency compared to its CMOS counterpart running with a low-swing clock. If the improvements through the combination of low-swing clocking and use of FinFETs are compared against a traditional, full-swing implementation with CMOS technology, 24% power savings is obtained in total power (clock network+DFF) compared to a CMOS-based full-swing implementation, while doubling the clock frequency (3 GHz). FinFET-based design of the logic (e.g., circuit) elements synchronized by this clocking system, while not investigated in this work due to a lack of an academically available FinFET cell library, would also demonstrate substantial power savings, adding to the savings reported in this work for the clock network and clock sinks (i.e., DFFs). Note that these potential savings in the circuit elements with a FinFET-based implementation are not impacted (positively or negatively) by the proposed low-swing clocking because the clock-to-output delay of the clock network is methodically preserved in low-swing operation. As such, any additional low-power design methodology can be used on the logic network, independent of the savings reported in this work for the clock tree network.

The proposed low-swing clock tree design methodology is not an easy procedure due to the following issues: (1) currently, there are no standard buffer or flip-flop libraries available for academic use for FinFET technology; (2) the timing characterization of these FinFET-based buffer and flip-flop designs needs to be performed at full- and low-swing voltage nodes so as to be used by clock tree design methodologies; (3) differences in the input capacitance and timing characteristics of FinFET buffers, compared to CMOS buffers, and their relationship to interconnect scaling at sub-32nm technology, necessitate a methodical approach as opposed to a straight-forward CMOS replacement approach; and (4) a novel, low-swing flip-flop-aware clock tree design methodology is necessary to provide clock trees with the slew specifications of the original full-swing clock tree. The proposed methodology in this work is implemented with: (i) a

custom FinFET-based clock buffer with a selected driving capability considering the original design constraints of the full-swing system; (ii) a custom low-swing flip-flop design (low-swing DFF) that has almost identical performance (skew, clock-to-output delay) compared to traditional full-swing flip-flops; and (iii) characterization of the timing of the clock buffer, and the input capacitance of the low-swing DFF. The clock tree design methodology uses a DME-based clustering stage and a novel buffering scheme to achieve low skew values while minimizing the power consumption of clock buffers and interconnects.

The rest of the article is structured as follows. The low-swing clocking techniques are introduced in Section 2, by discussing the existing methods and previous studies. The preliminaries and background for the FinFET-based low-swing clock tree methodology are analyzed in Section 3. In Section 4, the proposed methodology is introduced and explained in detail. The experimental results are presented in Section 5 and the article is finalized with concluding remarks in Section 6.

2. LOW-SWING CLOCKING

In this section, previous work on low-swing clocking is introduced and the methods are discussed in Section 2.1, while a discussion on single-V_{dd} versus dual-V_{dd} low-swing clocking is presented in Section 2.2.

2.1. Previous Work on Low-Swing Clocking

Clock distribution network design is one of the critical steps of IC design flow, as clock networks are known to consume a significant portion of the (dynamic) power consumption [Shelar and Patyra 2013] and have direct effect on performance (frequency). Furthermore, static leakage power dissipation is a growing concern for trees with a high number of buffers, particularly at sub-32nm technologies. Thus, the trade-off between performance and power consumption is well studied by both industry [Kurd et al. 2001; Shamanna et al. 2010; Xanthopoulos et al. 2001] and academia [Chaturvedi and Hu 2004; Lee et al. 2010; Sitik and Taskin 2013a]. High-end microprocessors target to minimize power consumption while satisfying high performance constraints [Kurd et al. 2001; Shamanna et al. 2010; Sitik and Taskin 2013a; Xanthopoulos et al. 2001], whereas low-power ASICs and mobile devices target to maximize performance with tight low-power constraints [Chaturvedi and Hu 2004; Lee et al. 2010]. Operating the clock network with a lower swing than the rest of the IC is one of the techniques explored in order to decrease voltage swing on the switching capacitance to minimize dynamic power consumed by the clock network [Asgari et al. 2004; Kim and Kang 2002; Markovic et al. 2004; Pangjun and Sapatnekar 2002; Zhu and Zhang 2001]. Low-swing operation can be adopted for low-power IC design, however, its applicability is limited due to the following issues.

(1) There is higher clock skew induced by the higher insertion delay due to clock buffers and interconnects operating at a lower voltage swing.
(2) There is less timing slack in the local datapaths, negatively affected by low voltage swing. Unless a methodical approach is proposed as in this work, the slew on the clock pin of flip-flops elevates, which negatively affects the local datapath timing. Thus the development of a low-swing DFF cell is critical, particularly in the targeted FinFET-based technology.
(3) Reduction in the quadratic dynamic power savings of low-swing clocking is necessary to address the clock skew and timing slack degradation of an otherwise straight-forward attempt at scaling the clock-swing voltage. Timing degradation through such a straight-forward scaling might lead to a nonfunctional circuit due to timing failure.

A number of previous solutions exist to address these issues, however: (i) they remain insufficient to address all issues at the same time and (ii) a FinFET-based methodology has not been investigated before. One previous approach considers the use of level-shifting buffers to interface a low-swing clock tree with full-swing flip-flops [Pangjun and Sapatnekar 2002]. With the level-shifting buffers, flip-flop sinks are driven with full-swing, therefore, issue (2) is addressed. However, driving the sink capacitance with a full-swing signal sacrifices most of the power savings, as the last level of the clock tree has the most capacitance. Another approach introduces a low-swing latch design combined with a low-skew clock tree design methodology, addressing issues (1) and (3) [Zhu and Zhang 2001]. However, the data delay on this latch design is doubled, and it is emphasized in this work that these latches are not suitable for critical paths, failing to address issue (2). A third approach introduces custom buffers of full-swing-to-reduced-swing, reduced-swing-to-reduced-swing, and reduced-swing-to-full-swing in order to design a low-swing clock tree that effectively bridges the full-swing clock source to full-swing clock sinks [Asgari et al. 2004]. This approach again fails to address issue (3) by converting the clock signal to full swing at the last level of the clock tree. Furthermore, their low-swing value depends highly on full-swing-to-reduced-swing buffer design. A recent work introduced a low-swing clock tree design methodology that addresses three issues at the same time with a pragmatic approach of using a standard buffer and flip-flop library [Sitik and Taskin 2013b]. However, the low-swing voltage level cannot be scaled more than 80% of full swing due to limitations on the standard cell library that is considered, failing to address issue (3) effectively. In this article, three issues are addressed by developing a novel clock tree design methodology that incorporates a low-overhead low-swing flip-flop design. The use of FinFET technology inherently minimizes the overhead of higher insertion delay compared to the incumbent CMOS technology. Also, the low-swing DFF introduced in this article is methodically built to have a clock-to-output delay as low as that of a typical full-swing flip-flop. This property, that preserves the system design constraints from the original full-swing design, is accomplished by methodically building the clock tree that drives the target low-swing DFFs: setting different input slew constraints on the clock tree, depending on the desired clock-to-output delay values at each sink DFF. Finally, the power savings of the proposed methodology are significant, due to: (1) increased dynamic power savings, thanks to (a) the low-swing clock tree and (b) low-swing-aware flip-flop design; and (2) the low-leakage structure of FinFET technology.

2.2. Single-V_{dd} vs. Dual-V_{dd} Low-Swing Clocking

Implementation of low-swing clocking can be realized in two ways, depending on the number of unique voltage sources on the chip: single- or dual-V_{dd} configurations. In the single-V_{dd} implementation, it is assumed that the only power grid available in the design is the full-swing V_{dd}. Thus, implementation of low-swing clocking includes level converting and low-swing buffers in order to create and manipulate the low-swing voltage level [Asgari et al. 2004]. This implementation is the only option when another power grid is not pre-planned, because the overhead of creating another power grid can be cost prohibitive. It is observed that the single-V_{dd} design implementation is not the ideal case for low-swing clocking, as the low-swing voltage level obtained may not be robust and hence consume greater leakage power [Asgari et al. 2004].

Alternatively, in most contemporary designs (especially for low-power implementations), additional power grids are already pre-planned (and placed before the clock tree synthesis step). Thus, it is compliant with common practice to assume the presence of a low-swing power grid. This is the preferred implementation method in this article, in order to fully benefit from low-swing clocking. Assuming the presence of a low-swing power grid (or multiple grids) floorplanned to be used for low-power applications of

Table I. Full- vs. Low-Swing Insertion Delay Characteristics at 20nm FinFET and 32nm CMOS Technologies for s38584

	20nm FinFET	32nm CMOS
Minimum Insertion Delay when CLK @V_{dd} (ps)	197.96	194.48
Maximum Insertion Delay when CLK @V_{dd} (ps)	220.19	223.77
Minimum Insertion Delay when CLK @$0.7\times V_{dd}$ (ps)	249.96	419.27
Maximum Insertion Delay when CLK @$0.7\times V_{dd}$ (ps)	279.35	474.75

the logic circuit, the clock tree also benefits from a low-swing grid in order to further decrease the power consumption, making this approach a perfect candidate for ultra-low-power applications.

3. PRELIMINARIES AND BACKGROUND

In this section, preliminaries about insertion delay, power consumption, and the effect of low-swing clocking on local timing are analyzed thoroughly with comparisons between typical 32nm CMOS and 20nm FinFET technologies. The change in insertion delay of clock trees depending on the low-swing voltage level is presented in Section 3.1. The power consumption of clock trees and the flip-flop sinks at various low-swing voltage levels is analyzed in Section 3.2. The effects of the low-swing voltage level and clock slew on the local timing are investigated in Section 3.3.

3.1. Insertion Delay in Low-Swing Clock Trees

Insertion delay is defined as the total (i.e., buffer and interconnect) delay measured from the source to each sink of a clock network. Although the insertion delay itself is not a design specification, higher insertion delay causes high clock skew and increased power consumption. Thus, minimizing clock insertion delay is one of the design objectives. The insertion delay is expected to increase when the clock swing (magnitude) is decreased, due to the slower switching transition of clock buffers. On the other hand, the gate delays in FinFET technology are negligibly low, particularly when compared to the delays of the interconnects. As such, the effect of low supply voltage on the gate/buffer delay must be reconsidered methodically for efficient implementation of low-swing clock trees in a FinFET-based technology.

In order to highlight this phenomenon, the insertion delays of a full-swing and a low-swing clock tree for the 20nm FinFET and 32nm CMOS technologies are compared on s38584 of ISCAS'89 benchmarks. 20nm FinFET models are obtained from PTM models [Sinha et al. 2012], and 32nm CMOS models are obtained from the SAED 32nm library of Synopsys, both of which are simulated using HSPICE of Synopsys. A clock tree is synthesized at each technology so as to have similar insertion delays at full swing. The two clock trees are simulated at full swing (V_{dd}) and low swing ($0.7\times V_{dd}$) in order to observe the respective changes in insertion delay. The minimum and maximum insertion delays of the FinFET- and CMOS-based clock tree at full- and low-swing operation are presented in Table I. It is shown in Table I that the insertion delay increases severely in low-swing operation of 32nm CMOS, even when the FinFET and CMOS technologies have similar insertion delays at full-swing operation. In order to visualize this phenomenon, the insertion delays of each of the clock sinks in s38584 are presented by normalizing them to their full-swing delays in Figure 1 from 100% of V_{dd} to 70% of V_{dd} with 5% decrements. It is shown in Figure 1 that the insertion delay of a clock sink in 32nm CMOS technologies increases by almost 120%, whereas this increase is $\approx$26% for the 20nm FinFET technology for clock signal magnitude scaling down to $0.7\times V_{dd}$. This example shows the superiority of FinFET technology over its counterpart in CMOS technology, by having small sensitivity in timing against a change in the supply voltage. With this observation, it is concluded here that FinFET

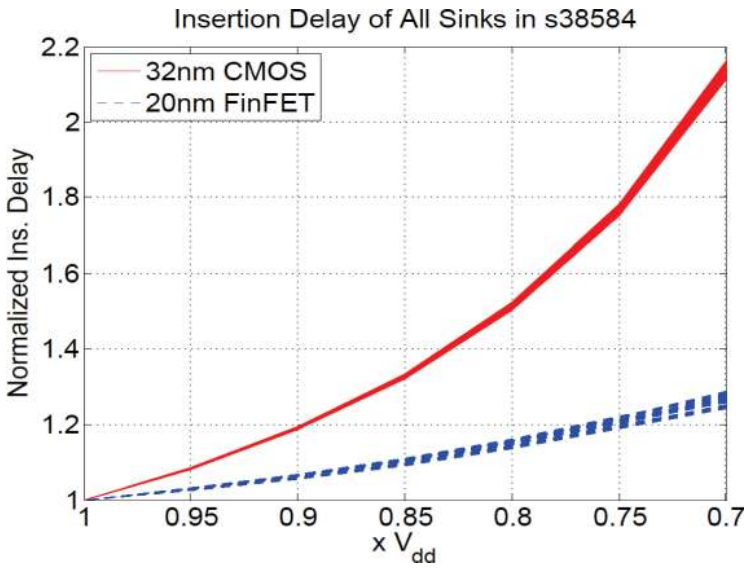


Fig. 1. Insertion delay characteristic of 20nm FinFET and 32nm CMOS technologies at different voltage levels interpolated from 100% to 70% of V_{dd} with 5% decrements.

technology inherently improves one quality aspect of low-swing trees by potentially providing a low insertion delay.

3.2. Power Consumption in Low-Swing Clock Trees

The power consumption of a clock tree can be divided into two parts: dynamic and static (leakage) power consumption. The dynamic power consumption is formulated as

$$P_{dyn} = \alpha C_{total} f V_{swing}^2, \tag{1}$$

where α is the switching factor, f the operating frequency, and V_{swing} the supply voltage of clock buffers. The total capacitance C_{total} is

$$C_{total} = \sum_{\forall i} C_i^{buffer} + \sum_{\forall j} C_j^{sink} + \sum l \times C_{unit} \tag{2}$$

with a design which has i clock buffers with an input capacitance of C_i^{buffer}, j sinks with an input capacitance of C_j^{sink}, and an interconnect capacitance of C_{unit} per unit length. Note that short-circuit power is neglected since transition times are sufficiently small due to tight constraints in slew. The low-swing clock trees target to save substantial (quadratic) dynamic power by scaling clock swing [V_{swing} in Eq. (1)]. For instance, a 30% drop-off from full-swing clock magnitude can lead to 51% savings in dynamic power dissipation. However, it is also important here to note that the number of clock buffers needs to be increased to satisfy the same timing metrics in low-swing operation as those of the full-swing tree. This increase in number of clock buffers sacrifices some of the power savings obtained through V_{swing} scaling by increasing C_{total}. Thus, it is concluded here that the low-swing clock tree design methodology should efficiently place clock buffers in order to satisfy the same performance (timing) constraints as the full-swing clock tree, while trading off dynamic power savings by minimizing the increase in C_{total}.

Table II. Leakage Power Consumption of Low- vs. Full-Swing Implementations of Clock Trees Built on s38584, using 20nm FinFET and 32nm CMOS Technologies

CLK Level	20nm FinFET (μW)	32nm CMOS (μW)
Full Swing (V_{dd})	15.3	319.5
Low Swing ($0.7\times V_{dd}$)	13.1	92.5

Given a FinFET-based clock buffer library, the static (leakage) power consumption depends on the number of clock buffers.

$$P_{static} \approx \sum_{\forall i} P_{leakage}^{buffer_i} \tag{3}$$

Thus, minimizing the number of clock buffers i for capacitance-induced dynamic power minimization opportunistically minimizes the static power consumption of FinFET-based buffers. Furthermore, lowering the V_{dd} level on the clock tree also decreases the leakage due to the lower electrical field on the channels. There are other popular methods than the proposed use of FinFET-based technology to address the leakage power problem in sub-32nm nodes. For instance, this issue can be addressed with power-gating, which is a widely used method to gate power grids in order to save power by cutting off the power supply on the transistors when they are idle. However, this method cannot be applied to flip-flop sinks, as they carry the state of the operation. High-V_{th} transistor libraries can be considered as another method to address this issue. However, higher V_{th} increases the gate delay, therefore their applicability remains limited, especially on the clock buffers and gates on critical paths. With the emergence of FinFET technology, the static (leakage) power consumption is minimized inherently while enabling frequency scaling thanks to low gate delay. In order to highlight the low-leakage property of FinFET technology compared to its counterpart in CMOS, a clock tree is designed for s38584 of the ISCAS'89 benchmarks, using 20nm FinFET [Sinha et al. 2012] and 32nm CMOS [Synopsys, Inc. 2012] libraries. A leakage power comparison of the two technologies is presented in Table II. As shown in Table II, the leakage power consumption in FinFET decreases significantly compared to CMOS technology. In the full-swing implementation, the leakage power decreases to 15.3μW from 319.5μW. When the low-swing implementation is considered, the leakage decreases in CMOS technology to 92.5μW, however, it is still $\approx$7$\times$ as large as its FinFET counterpart.

3.3. Effect of Low-Swing Clocking on Local Paths

The effects of clock performance constraints (skew and slew) must be fully understood in order not to degrade the timing performance at a low-swing clock tree implementation. The study in Sitik and Taskin [2013b] shows that not only clock slew but also the magnitude of the clock swing (i.e., in low-swing clocking) are key factors that affect the clock-to-output delay. Thus, the equivalence in terms of timing performance is defined as the equivalent effect on the clock-to-output delay. In effect, this equivalence defines different slew constraints at different low-swing clock voltage, as the clock-to-output delay primarily depends on the clock slew, assuming that the output capacitances are the same.

Another issue which arises in low-swing clocking is the increased power consumption at the clock sinks (e.g., flip-flops). By convention, only the input capacitance of a clock sink (C_j^{sink}) is considered to contribute to the power dissipation of a clock network, as standard cell libraries readily have optimized DFF cells whose power consumption does not directly depend on the clock tree. However, these library cells are optimized assuming the presence of a full-swing clock signal. Thus, the effect of low-swing clocking

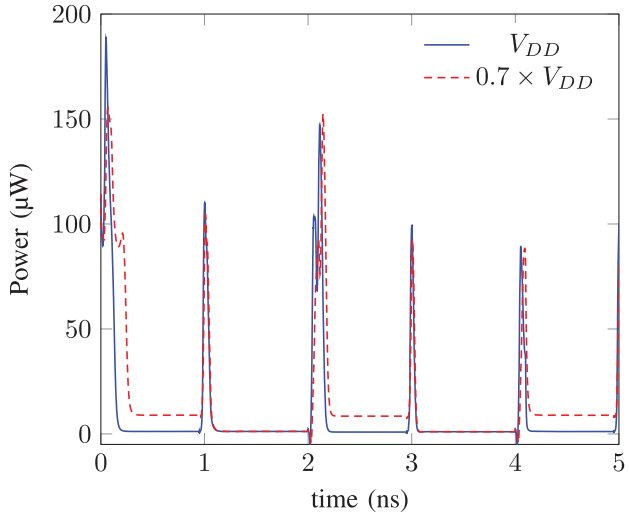


Fig. 2. Power consumption of the DFFX1 from Synopsys standard library when CLK@V_{dd} and CLK@$0.7\times V_{dd}$. As shown in the graph, when operated with a low-swing clock (at $0.7\times V_{dd}$), the power consumption increases significantly. In this example case, the average power consumption is 9.9μW when the clock is at full swing and 13.0μW when at low. In the figure, the source of the extra power consumption occurs at the static region (i.e., non-peaks), highlighting the need for a low-swing clock aware DFF design.

can degrade their power consumption significantly, making DFF power consumption a part of the clock tree design process when low-swing clocking is considered. In order to highlight this issue, the power consumption of a standard minimum-size DFF cell, which is selected to be "DFFX1" from the SAED 32nm library of Synopsys [Synopsys, Inc. 2012], is presented in Figure 2 at low-swing (CLK@$0.7 \times V_{dd}$) and full-swing clock (CLK@V_{dd}) input cases. It is shown in Figure 2 that the power consumption of a DFF cell that is optimized for full-swing operation can severely increase (31% for this example) at low-swing operation due to a large contention current. It is concluded here that both the low-swing DFF topology and the clock network design methodology should consider both the clock-to-output delay and the power consumption in order not to degrade the performance of the system to enable low-swing clocking.

4. DESIGN METHODOLOGY

The proposed design methodology that enables implementation of low-swing clock trees at FinFET technology consists of four major steps:

(1) the design of a FinFET-based clock buffer library, which is an optional step only required if a FinFET-based buffer library is not readily available;
(2) the timing characterization of the buffer library, which is another optional step only required if the timing models of the buffer library are not readily available;
(3) design of a low-swing D flip-flop, which is a required step as the low-swing DFF directly affects the overall performance and power consumption; and
(4) a low-swing DFF-aware clock tree design considering the timing constraints (slew and skew) and power consumption.

To our best knowledge, a FinFET-based standard cell library is currently not available for academic use. Thus a clock buffer is custom designed so as to have similar driving capabilities as the reference NBUFFX32 cell in the baseline CMOS-based buffer library (Synopsys SAED32nm [Synopsys, Inc. 2012]). This is important for a

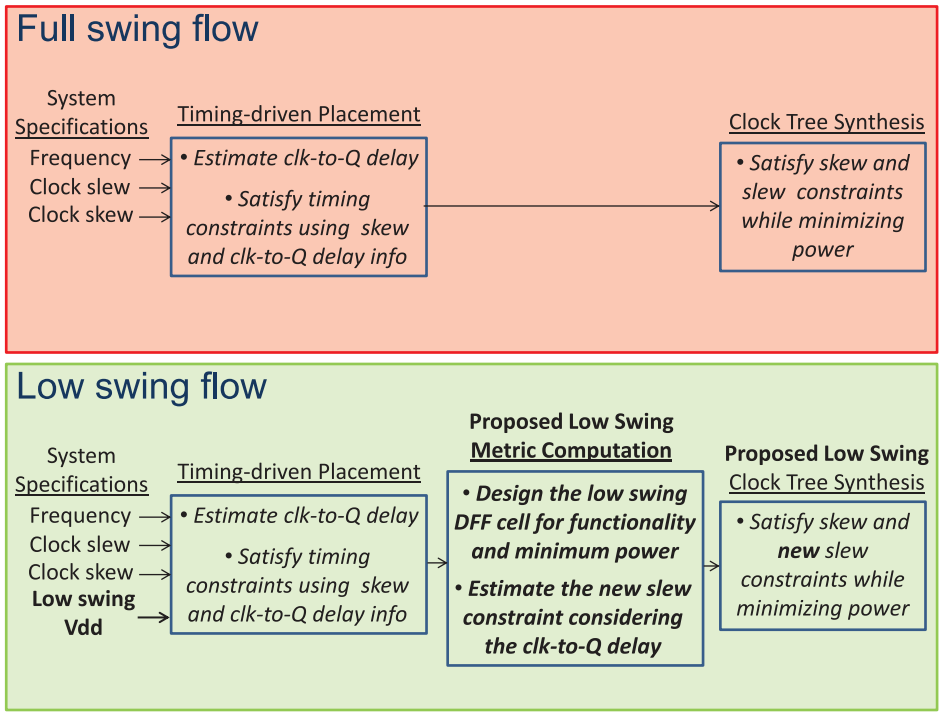


Fig. 3. In the full-swing flow, the placement information can directly be used, as the data and clock swing are the same. In the proposed low-swing flow, first the low-swing DFF is designed considering the low-swing voltage level, then clock-to-output delay estimation obtained in the placement step is used to define a new slew constraint in order to obtain the same performance at local paths.

fair comparison between implementations in the incumbent CMOS technology and that proposed in the FinFET technology, explained in detail in Section 4.1. The timing characterization of this custom-designed buffer is introduced in Section 4.2 to be used at the clock tree design step. After these optional steps, which are only performed due to the absence of a FinFET library, the required steps of the proposed flow are summarized in Figure 3 with a comparison to the typical clock tree synthesis flow. In the proposed flow, the low-swing flip-flop is designed considering the functionality, robustness, and power consumption which are introduced in Section 4.3. Also, a new slew metric is defined for low-swing clocking in order to obtain (almost) the same clock-to-output delay compared to its full-swing counterpart. Next, the clock tree synthesis is performed, which is presented in Section 4.4, considering the skew and new slew metric to target (almost) the same performance on local paths while saving significant power at the clock tree via low-swing clocking.

4.1. FinFET-Based Clock Buffer Design

This step is optional in the presence of a FinFET-based clock buffer library. The clock buffer design presented here does not define the novelty of the presented work of this article, but is needed to have a complete design flow. The design is limited to a single buffer only in order not to divert the focus of the article to the (potentially suboptimal) quality of the created buffer library. Instead, having only a single buffer type highlights the elegance of the proposed low-swing flip-flop topology and the low-swing clock tree design methodology.

Table III. Custom-Designed FinFET-Based Buffer vs. CMOS-Based NBUFFX32 of SAED32nm Library

Technology	1st Inverter (P/N)	2nd Inverter (P/N)	Output Load	Output Slew (ps)
20nm FinFET	2fin/2fin	29fin/25fin	200μm wire	38.7
32nm CMOS	800nm/420nm	800nm/420nm	200μm wire	37.2

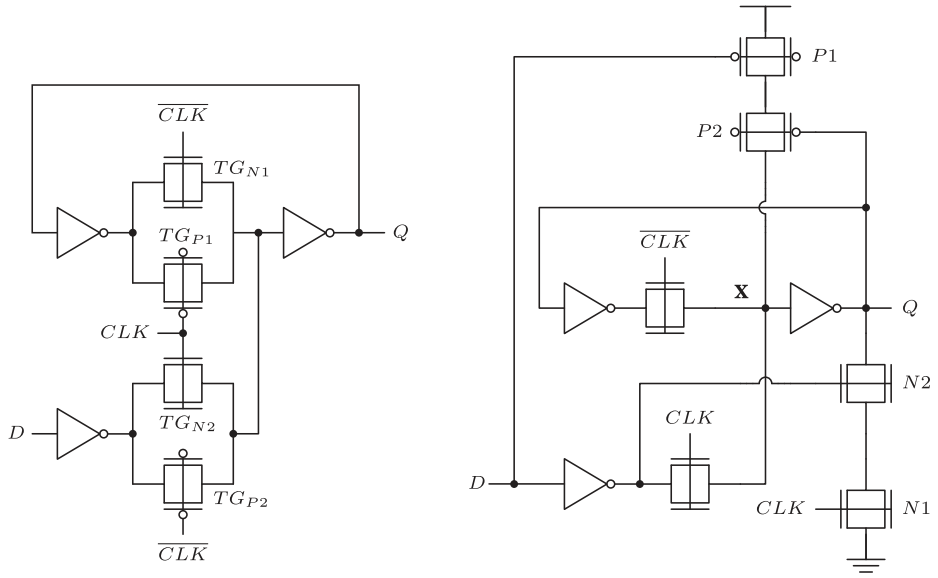
The custom clock buffer is designed to have a similar driving capability as the maximum-size buffer of the SAED32nm library of Synopsys, which is NBUFFX32. As the width and length are process parameters in a FinFET technology, the sizing of the selected buffer is performed by changing the number of fins. It is a discrete number (as the number of fins is an integer) whose optimal value cannot be found by running continuous optimization methods. Instead, the desired number of fins is found with an iterative search. First, the second inverter in the buffer is sized to achieve similar (close to but not the same due to the discrete number of fins) driving characteristics as NBUFFX32 when the input slew is at 50ps. Representative of a typical clock-branch length at the technology node, a 200μm wire is connected as the output load. In this scenario, the number of fins is computed to be 29 for PFET and 25 for NFET. After this optimization, the number of fins of the first inverter is changed so as to drive the second inverter with a similar slew value. The number of fins for the NFET and the PFET of the first inverter of the buffer is determined to be 2. With this optimization, the designed FinFET-based buffer and NBUFFX32 of SAED32nm library drive the 200μm wire load with a similar output slew ($\approx$39ps versus $\approx$37ps, respectively). A complete comparison of these two buffers is summarized in Table III. It is targeted here that having similar driving characteristics for both the FinFET and CMOS buffer enables a fair comparison of technologies when two synthesized clock trees are compared using these two buffers.

4.2. Timing Characterization of FinFET-Based Clock Buffer Design

The methodology used for timing characterization of the FinFET-based buffer introduced in Section 4.1 is presented in this section. The timing characterization of clock buffers can be performed in various ways. One option is using higher-order models [Li and Acar 2005; Raja et al. 2008] that can accurately estimate the timing characteristic of buffers at the expense of runtime. Another approach is storing empirical data in lookup tables in order to decrease runtime with similar accuracy. For the proposed clock tree design methodology, a third option is used because simpler first-order models are mostly sufficient for the clock tree synthesis. The first-order model helps achieve substantial accuracy with simpler metrics which are also more easily integrated into the clock tree design and optimization process. A commonly used model for the clock buffers is the RC model, where each buffer is modeled as a resistance (R_{out}) and a capacitance (C_{out}). With this model, it is easier to combine buffer delay with the wire delay, which is also modeled as an equivalent RC network. Despite its simplicity, these models produce substantially accurate timing metrics (such as Elmore delay [Elmore 1948]). With this observation, the FinFET-based custom clock buffer designed in Section 4.1 is characterized by sweeping the output capacitance to estimate R_{out} and C_{out}. As the wire capacitance is dominant for sub-32nm designs, the wire length at the output is swept with desired data points, and R_{out} and C_{out} can be obtained with a linear fit equation. For instance, the output load is swept from a 20μm wire to 200μm with 10 data points for the experimental setup of this work. The accuracy of the adopted timing characterization metrics is empirically demonstrated in the experimental results.

4.3. Low-Swing DFF Design

Existing low-swing flip-flop topologies primarily utilize dynamic nodes, degrading robustness [Hu et al. 2007; Mahmoodi-Meimand and Roy 2004]. Also, the clock-to-output



(a) classical full-swing latch (b) proposed low-swing latch

Fig. 4. Latch schematics with FinFETs.

delay is typically sacrificed unless a large number of transistors is added. For example, in Zhang and Sun [2007], a sense amplifier is used which further complicates the design process. The proposed topology in this article is based on the most commonly used edge-triggered static DFF, and addresses these issues while providing a robust output signal and a low clock-to-output delay with a relatively small number of transistors. Figure 4(a) shows the classical implementation of a latch (that is used to build a DFF) using FinFET-based transmission gates. For instance, the combination of PFET TG_{P1} and NFET TG_{N1} constitutes a transmission gate (e.g., on the latch data storage path). In low-swing clock applications (e.g., 0 to $V_{low\ swing}$) the circuit shown in Figure 4(a) can present issues. Specifically, the P-type transistors TG_{P1} and TG_{P2} within the transmission gates fail to completely turn off when the clock is high, hence providing a conductive path and eventually leading to increased power consumption (which is shown to be ≈31% for DFFX1 of the SAED library in Section 3.3) or functional failure. For these reasons, the circuit of Figure 4(b) is novelly introduced instead, where the P-type transistors within the transmission gates are removed. Operation of the proposed topology is as follows: Since an N-type transistor alone cannot pass a strong 1 (logic high), node X of Figure 4(b) would not reach V_{DD} when CLK is not at full-swing voltage. Transistor $P2$ is a level restorer and ensures that node X reaches V_{DD} when D is low. In order for $P2$ to start conducting, however, node Q should be at a voltage lower than V_{DD}. This is achieved by $N2$: when D is low, $N2$ is turned on and starts to lower the voltage at node Q so that $P2$ can restore the level of node X. Finally, $N1$ guarantees that $N2$ changes Q only when CLK is high. Note that $P1$ is added to prevent contention current when node X is being discharged.

In order to demonstrate the performance of the proposed low-swing flip-flop topology in terms of addressing the mentioned issues, a simple example is presented. Using 20nm FinFET technology, both a traditional full-swing flip-flop (using two of the latches shown in Figure 4(a) in a master-slave latch configuration) and the proposed

Table IV. Clock-to-Output Delay (C2Q) and Power Consumption Comparison of Conventional DFF Topology and the Proposed Low-Swing DFF Topology in 20nm FinFET Technology

	Power (μW)	C2Q (D=1)(ps)	C2Q (D=0)(ps)
Conventional DFF when Data @V_{dd}, CLK @V_{dd}	5.4	21.4	20.4
Conventional DFF when Data @V_{dd}, CLK @0.7×V_{dd}	7.6	31.9	31.3
Proposed DFF when Data @V_{dd}, CLK @0.7×V_{dd}	2.7	28.8	28.2

low-swing flip-flop (using two of the latches shown in Figure 4(b) in a master-slave latch configuration) are implemented to compare their performance (i.e., clock-to-output delay) and power consumption when the low-swing voltage is at $0.7\times V_{dd}$. The clock-to-output (C2Q) delay and the power consumption of a conventional DFF topology when the clock input is at full V_{dd}, the conventional DFF topology when the clock input is at $0.7\times V_{dd}$, and the proposed DFF topology when the clock input is at $0.7\times V_{dd}$ are compared in Table IV. It is important to note here that the supply voltages of datapaths are set as full V_{dd} in all cases. It is shown in Table IV that the clock-to-output delay of the conventional DFF increases by ≈10ps when a low-swing clock at $0.7\times V_{dd}$ is applied. Furthermore, it is important to note that the contention current through the PFETs not completely turned off (conventional DFF with clock at $0.7\times V_{dd}$ case) creates large spikes, which results in unstable behavior and possible failures. On the other hand, this increase is 7.4ps and 7.8ps for the $D = 1$ and $D = 0$ cases, respectively, for the proposed DFF topology, limiting the increase in clock-to-output delay by 2.2% and 2.3% of the clock period at 3 GHz compared to a traditional full-swing flip-flop. It is also shown that the power consumption of conventional DFF increases significantly (from 5.4uW to 7.6uW) when a low-swing clock is applied, as in the case of 32nm CMOS (shown in Figure 2). On the other hand, the power consumption of the proposed DFF topology decreases significantly (from 5.4uW to 2.7uW). Thus, the proposed low-swing flip-flop topology addresses both the performance and power consumption issues encountered when a traditional flip-flop topology is used for low-swing operation, thereby enabling the implementation of low-swing clock trees with a small overhead at the local paths.

4.4. Low-Swing Clock Tree Design

In this section, the proposed low-swing clock tree design methodology is introduced. The inputs of this algorithm are: (i) a placed design with input capacitance information of flip-flop sinks; (ii) the timing models of the clock buffer and clock wire; and (iii) the skew and new slew constraints as presented in Table IV of Section 4.3. With these inputs, the algorithm returns the buffer and wire locations as well as their connectivity as the output. The pseudocode for the proposed methodology is given in Algorithm 1.

The algorithm starts by initializing each sink as a node and setting a bottom-up delay (*bottomup_delay*) of 0 to each node (lines 1–3), as these nodes are the leaves of the clock tree. Then, the algorithm searches the minimum-cost (*min_cost*) pair by iterating through all pairs (lines 7–15). In the algorithm, cost is defined as the total capacitance: the sum of the capacitance of two nodes (i and j) to be merged, and the capacitance of the interconnect that connects them ($wire_{i,j}$). Once the minimum cost is found, their merging point is calculated so as to have the same delay using the well-known deferred merge embedding for zero-skew trees (ZST-DME) [Boese and Kahng 1992] routing algorithm (line 10). If a feasible pair is found (line 16), two nodes are merged into a new node (k) with the cost (*min_cost*) defined to be the new capacitance metric of the created node. If wire snaking is necessary, it is avoided assuming a low-power application, and the merging point is placed on the child node where wire snaking is (e.g., would have been) necessary (line 17). The bottom-up delay metric of new node k is updated as the delay from the merging point to its child ($wire\_delay_{i,k}$) plus the bottom-up delay of this child (*bottomup_delay*(i)). Note that the delay from this

ALGORITHM 1: FinFET-Based Low-Swing Clock Tree Design Methodology

Input: DFF placement, timing models for the buffers and the interconnects, the skew and the maximum capacitance constraint *max_cap*.
Output: Locations of clock buffers, clock interconnects and their connectivity.

```
1  Initialize each sink as a node;
2  for i in nodes do
3      bottomup_Delay(i) = 0;
4  end
5  while num_of_nodes >1 do
6      min_cost = max_cap;
7      for i in nodes do
8          for j in nodes do
9              if (merging_cost_{i,j} < min_cost) then
10                 min_cost = merging_cost_{i,j};
11                 found_i = i;
12                 found_j = j;
13             end
14         end
15     end
16     if (min_cost <= max_cap) then
17         Find_merging_point(found_i, found_j, bottomup_delay(found_i), bottomup_delay
           (found_j));
18         Create new node k, delete found_i and found_j;
19         bottomup_delay(k) = (wire_delay_{found_i,k} + bottomup_delay(found_i) +
           wire_delay_{found_j,k} + bottomup_delay(found_j))/2;
20     end
21     else
22         Buffer all nodes;
23         Update their input capacitance;
24     end
25 end
```

node to its two children is not the same when wire snaking is necessary (as the wire snaking is avoided); the bottom-up delay at this node is defined to be the average of the two bottom-up delays calculated using bottom-up delays of two children (line 18). If no feasible pair is found (line 16), all nodes are buffered, although they do not have the same capacitance (lines 22–23). The purpose here is to create a balanced clock tree, that is, reaching each clock sink through the same number of levels. Having this uniform structure is very important for low-swing clock tree synthesis, as lack of uniformity can vary insertion delays significantly, resulting in excessive clock skew. After buffering is performed, the capacitance values of these nodes are updated to be the input capacitance of the buffer, and the same algorithm is followed recursively until only one node, that is, the parent node of the clock tree, is left (lines 5–25). The DME-based algorithm successfully synthesizes a buffered bounded skew tree by combining a well-known routing approach with a novel buffering scheme, addressing the lack of automation tools in FinFET technology. Given a standard cell library in FinFET technology, more advanced methodologies can seamlessly be used in this step to further improve the quality of results presented in this article.

5. EXPERIMENTAL ANALYSIS

In order to analyze the quality of results, the proposed methodology is implemented using a 20nm FinFET technology in comparison to a 32nm CMOS technology. The

experimental setup is introduced in Section 5.1 and experimental results for various cases are presented in Sections 5.2 to 5.5.

5.1. Experimental Setup

The low-swing DFF cell introduced in Section 4.3 is implemented with the 20nm PTM models [Sinha et al. 2012] for FinFET and 32nm SAED Library of Synopsys for CMOS. The proposed clock tree design methodology is coded in Perl and tested on the three largest circuits (s38584, s38417, and s35932) of ISCAS'89 benchmarks. It is noted here that ISPD'10 clock network contest benchmarks are not usable for the proposed methodology in this article, as they lack logic information for the local paths. Instead, they have capacitance locations, modeling clusters of flip-flop cells as a single capacitance. Thus, the proposed low-swing DFF cells and low-swing DFF-aware clock tree design methodology cannot be tested on these benchmarks. However, the maximum number of sinks in ISPD'10 benchmarks (2249) is comparable to the maximum number of sinks in ISCAS'89 benchmarks (1728 for s35932). Logic synthesis of the RTL-level netlists of the benchmark circuits is performed using Design Compiler of Synopsys, and placement of the flip-flops is performed using IC Compiler of Synopsys at 1.5 GHz, at SAED32nm CMOS technology. As a complete cell library for 20nm FinFET technology is not available, the placement is performed using the same library at 3 GHz for the tests of FinFET nodes. This placement is not accurate as the FinFET-based cell library does not exist, but it serves to provide an attainable, reproducable approximation to the expected placement in absence of FinFET libraries. After the proposed low-swing clock tree is designed, the power and skew analyses are performed using HSPICE simulator of Synopsys. The full-swing voltage levels are set as default values in each technology, as 1.05V and 0.9V for the 32nm CMOS and 20nm FinFET technologies, respectively. The low-swing voltage level is assumed 70% of the nominal V_{dd} level at each case. The wire models for the 32nm technology are obtained from Natarajan et al. [2008], which has unit resistance and unit capacitance of R=8Ω/μm and C=0.2fF/μm, respectively, with the default 50nm wire width. The wire models for 20nm technology are adopted from the 22nm technology in Auth et al. [2012], which projects 13% less capacitance from the 32nm technology, resulting in a C=0.174fF/μm. The per-unit resistance remains the same as reported in Auth et al. [2012], which is R=8Ω/μm. The clock slew and clock skew requirements of FinFET- and CMOS-based circuits are selected to enable rational comparison: As discussed in Section 3.3, the slew constraint is used as a proxy for clock-to-output delay, where the clock-to-output delay is methodically mandated to remain the same regardless of the presence of a low-swing operation. To this end, for FinFET-based circuits, the clock slew constraint is set to 50ps for the full-swing case and 25ps for the low. It is shown in Section 3.3 that the clock-to-output delay at 70% of V_{dd} is $\approx$28ps, almost the same as its full-swing counterpart at a clock-to-output delay of $\approx$21ps with these slew constraints. For CMOS-based circuits at 1.5 GHz (at half the frequency), the slew constraint is set to 100ps, as opposed to 50ps in FinFET-based circuits. This constraint is set the same (100ps) for both low- and full-swing implementations because their clock-to-output delays are measured to be identical when the same input is applied. The clock skew constraint is set to 50ps for both technologies, targeting the same topology/insertion delay performance for both low- and full-swing implementations. This is considered reasonable, as the placements are performed using the same cell library for CMOS- and FinFET-based circuits (due to lack of a FinFET cell library).

There are three primary goals of the experiments:

(1) to evaluate applicability of low-swing clocking in a FinFET-based technology as a dynamic power-saving (36% at the same frequency) methodology (Section 5.2);

Table V. Clock Buffer Metrics (number of clock buffers/total buffer capacitance) and Clock Interconnect Metrics (total interconnect length/total interconnect capacitance) of FinFET-Based Low-Swing (LS) and FinFET-Based Full-Swing (FS) Clock Trees with Information on Benchmark Circuits

Circuit	Floorplan Size (μm$\times\mu$m)	Number of Sinks	Clock Buffer Info (# / fF)		Clock Interconnect Info (μm/fF)	
			FS	LS	FS	LS
s38584	250×251	1239	86/12.9	220/33.0	14783/2572.2	14551/2531.9
s38417	262×264	1461	94/14.1	247/37.1	16826/2927.7	16374/2849.1
s35932	211×212	1728	88/13.2	259/38.9	16683/2902.8	16343/2843.7

(2) to demonstrate that low-swing clocking permits a higher frequency (2×) at the same power budget (3% power savings) with a methodical design of a low-swing FinFET-based library compared to its CMOS counterpart (Section 5.3); and
(3) to demonstrate that low-swing clocking provides significant power savings (46% power savings) at the same frequency budget with a methodical design of a low-swing FinFET-based library compared to its CMOS counterpart (Section 5.4).

5.2. Low- vs. Full-Swing Clocking in FinFET Technology

The experimental results of the proposed FinFET-based low-swing clocking methodology compared to its full-swing counterpart are presented in this section. The total clock buffer capacitance and total clock interconnect capacitance values are reported in Table V, including the number of sinks and floorplan size information of each benchmark circuit. The columns FS and LS demonstrate these clock network metrics for full- and low-swing operation with FinFETs, respectively.

The increase in number of clock buffers is expected for low-swing operation as the clock slew constraint is set to 25ps (as opposed to 50ps in the full-swing case) at this case to accommodate low-swing DFF design. Partially compensating for the increase in number of clock buffers, the total interconnect length is smaller in the low-swing case, thanks to the clustering algorithm working more efficiently with the increased number of buffers in low-swing clocking. It is observed in Table V that the increase in total capacitance is countered by a decrease in interconnect capacitance, serving to minimize the overhead of low-swing clocking on the total switching capacitance. This observation is supported by the dynamic power consumption comparison of full- and low-swing implementations of FinFET-based clock trees. A comparison of the FinFET-based full-swing clock tree against the FinFET-based low-swing clock tree is presented in Table VI. In Table VI, it is shown that there are significant power savings which are recorded, on average, to be 36% for the clock tree (excluding DFF power) and 52% for the DFF cells. Strikingly, these savings are recorded despite the number of clock buffers almost tripling to provide a sharper clock slew to the low-swing DFFs. It is iterated for a theoretical interpretation of the empirical result here that the savings are on the order of 36%, smaller than the expected 51% of ideal V_{dd}^2 savings (0.7×0.7=0.49). The reason behind this degradation in power savings is the fact that the input capacitance of the low-swing DFF cell is higher than that of the traditional full-swing DFF (0.228fF versus 0.063fF). However, this degradation in expected power savings is compensated by the savings in DFF power, thanks to the novel low-swing DFF cell introduced in this work. It is shown in Table VI that power savings at the DFF-only cells are 52%. The savings in power consumption of FinFET-based clock networks through low-swing clocking are realized via degrading the timing performance by a negligible amount. It is shown in Table VI that the degradation in clock skew and of the clock-to-output delay are 14.7ps and 7.3ps, respectively, totaling a margin of 22ps within a clock cycle, which is still as low as 6.6% of the clock period at high-speed 3 GHz operation.

For comparison purposes, the same algorithm is used to implement low- and full-swing CMOS-based clock trees on the same benchmark set. The results are presented

Table VI. Performance and Power Comparison of FinFET-Based Low- (LS) and Full-Swing (FS) Clock Trees at 3 GHz (reported separately for the clock network and DFF cells)

	CT Power (mW)		DFF Power (mW)		Skew (ps)		Max Slew (ps)		Max C2Q Delay (ps)	
Circuit	FS	LS	FS	LS	FS	LS	FS	LS	FS	LS
s38584	11.64	7.31	5.84	2.86	18.9	28.2	47.6	24.6	20.9	28.1
s38417	13.02	8.25	6.97	3.33	21.1	41.7	47.9	24.9	21.0	28.7
s35932	12.76	8.44	8.27	3.89	19.6	33.7	49.3	24.7	21.1	28.1
Avrg. Imp. over FS		36%		52%		−14.7	SATISFIED			−7.3

Table VII. Performance and Power Comparison of CMOS-Based Low- and Full-Swing Clock Trees at 1.5 GHz (reported separately for the clock network and DFF cells)

	CT Power (mW)		DFF Power (mW)		Skew (ps)		Max Slew (ps)		Max C2Q Delay (ps)	
Circuit	FS	LS	FS	LS	FS	LS	FS	LS	FS	LS
s38584	9.18	6.17	4.12	3.99	42.1	45.1	94.1	86.3	62.5	63.3
s38417	10.45	7.16	4.84	4.72	32.2	47.7	93.4	91.0	62.2	63.6
s35932	10.24	7.54	5.72	5.52	27.9	37.6	95.0	82.2	62.5	63.2
Avrg. Imp. over FS		30%		3%		−9.4	SATISFIED			−1.0

in Table VII. It is observed in Table VII that power savings can also be obtained in CMOS technology via low-swing clocking, as widely reported in literature [Asgari et al. 2004; Sitik and Taskin 2013b; Zhu and Zhang 2001]. However, due to the superiority of FinFET technology (low power, decreased performance overhead due to low buffer delay), the power savings in the low-swing clock tree in FinFET technology are 6% more than its CMOS counterpart (36% versus 30%). Furthermore, power savings at the DFF-only cells are as low as 3% in CMOS technology. The reason behind this improvement in DFF power is the fact that the transistor delay (the inverter chain in DFF cells) increases more significantly in CMOS technology than its FinFET counterpart, as shown in Section 3.1. Thus, transistors are upsized more in CMOS technology to obtain comparable clock-to-output delay to its full-swing counterpart, degrading power savings at the DFF cell.

5.3. FinFET-Based Low-Swing Clocking at High Performance

In this section, the superiority of the methodical implementation of a FinFET-based low-swing operation compared to its CMOS counterpart is highlighted as a method for providing higher frequency at the same power budget. The power consumption of the clock tree only (excluding DFF) and total power consumption (clock network+DFF) of FinFET and CMOS technologies for both full- and low-swing implementations are compared in Tables VIII and IX, respectively, normalized to FinFET low-swing implementation (FinFET LS). Compared to a CMOS low-swing implementation (CMOS LS), the proposed FinFET low-swing implementation (FinFET LS) can achieve 2× frequency with a 15% ((0.87–1.00)/0.87) degradation in power consumption of the clock tree only (excluding DFF), as shown in Table VIII. When total (clock network+DFF) power consumption is considered, as shown in Table IX, the proposed FinFET low-swing implementation (FinFET LS) provides marginally better power consumption at 3% ((1.03–1.00)/1.03) while running at 2× frequency compared to CMOS low-swing implementation (CMOS LS), thanks to the novel low-swing DFF design of this work. The improvements through combination of low-swing clocking and use of FinFETs are reported in comparison against a traditional, full-swing implementation with CMOS technology (CMOS FS). The proposed FinFET low-swing implementation (FinFET LS) provides 20% ((1.25–1.00)/1.25) smaller power consumption in the clock tree only (excluding DFF) compared to a typical full-swing CMOS implementation (CMOS FS) running at half speed (1.5 GHz), shown in Table VIII. The total power

Table VIII. Clock Tree Power (excluding DFF) of Proposed FinFET-Based Low-Swing (LS) vs. Full-Swing (FS) Clocking in FinFET (at 3 GHz) and CMOS (at 1.5 GHz) Technologies (normalized to FinFET-based low swing)

	FinFET (@3GHz)		CMOS (@1.5GHz)	
Circuit	FinFET LS	FinFET FS	CMOS LS	CMOS FS
s38584	7.31	11.64	6.17	9.18
s38417	8.25	13.02	7.16	10.45
s35932	8.44	12.76	7.54	10.24
Normalized Average Power	1.00×	1.56×	0.87×	1.25×

Table IX. Total (clock network+DFF) Power of Proposed FinFET-Based Low-Swing (LS) vs. Full-Swing (FS) Clocking in FinFET (at 3 GHz) and CMOS (at 1.5 GHz) Technologies (normalized to FinFET-based low swing)

	FinFET (@3GHz)		CMOS (@1.5GHz)	
Circuit	FinFET LS	FinFET FS	CMOS LS	CMOS FS
s38584	10.17	17.48	10.16	13.30
s38417	11.58	19.99	11.98	15.29
s35932	12.33	21.03	13.06	15.96
Normalized Average Power	1.00×	1.72×	1.03×	1.31×

consumption (clock network+DFF) provides even better power savings at 24% ((1.31–1.00)/1.31), thanks to the novel low-swing DFF design as shown in Table IX.

5.4. FinFET-Based Low-Swing Clocking for Ultra-Low Power

With the use of FinFET technology, it is desired not only to reduce leakage power but also to improve performance through frequency scaling. However, FinFET technology can also be used in ultra-low-power applications, taking advantage of the low-power structure of FinFETs at the same frequency constraints as CMOS technology. In order to investigate this scenario for low-power operation only, power consumption is also compared while both technologies are running at 1.5 GHz. The comparison of clock tree power consumption only (excluding DFF) is presented in Table X, and comparison of the combined (clock tree+DFF) power consumption is presented in Table XI. In this scenario of low-power operation without frequency scaling, the proposed FinFET low-swing implementation (FinFET LS) can achieve 44% ((1.80–1.00)/1.80) power savings at the clock tree only (excluding DFF) compared to CMOS low-swing implementation (CMOS LS), as shown in Table XII at the same frequency constraint (1.5 GHz). When total (clock network+DFF) power consumption is considered as shown in Table XI, the proposed FinFET low-swing implementation (FinFET LS) provides 46% ((1.84–1.00)/1.84) power savings compared to CMOS low-swing implementation (CMOS LS) running at the same frequency (1.5 GHz), thanks to the novel low-swing DFF design of this work. Similar to Section 5.3, the improvements through combination of low-swing clocking and use of FinFETs are reported in comparison against a traditional, full-swing implementation with CMOS technology (CMOS FS). The proposed FinFET low-swing implementation (FinFET LS) provides 61% ((2.57–1.00)/2.57) power savings in clock tree only (excluding DFF) compared to a typical full-swing CMOS implementation (CMOS FS) running at half speed (1.5 GHz), shown in Table X. The total power consumption (clock network+DFF) is at 57% ((2.33–1.00)/2.33), shown in Table XI.

5.5. Leakage Power Comparison

It is highlighted in Section 3.2 that FinFET technology is superior due to its subthreshold-leakage-immune structure. In order to highlight this effect, leakage power consumption is compared in both technologies to provide a deeper analysis

Table X. Clock Tree Power (excluding DFF) of Proposed FinFET-Based Low-Swing (LS) vs. Full-Swing (FS) Clocking in FinFET (at 1.5 GHz) and CMOS (at 1.5 GHz) Technologies (normalized to FinFET-based low swing)

	FinFET (@1.5GHz)		CMOS (@1.5GHz)	
Circuit	FinFET LS	FinFET FS	CMOS LS	CMOS FS
s38584	3.55	5.45	6.17	9.18
s38417	4.01	6.09	7.16	10.45
s35932	4.06	5.94	7.54	10.24
Normalized Average Power	1.00×	1.51×	1.80×	2.57×

Table XI. Total (clock network+DFF) Power of Proposed FinFET-Based Low-Swing (LS) vs. Full-Swing (FS) Clocking in FinFET (at 1.5 GHz) and CMOS (at 1.5 GHz) Technologies (normalized to FinFET-based low swing)

	FinFET (@1.5GHz)		CMOS (@1.5GHz)	
Circuit	FinFET LS	FinFET FS	CMOS LS	CMOS FS
s38584	5.67	9.17	10.16	13.30
s38417	6.50	10.53	11.98	15.29
s35932	6.95	11.21	13.06	15.96
Normalized Average Power	1.00×	1.62×	1.84×	2.33×

Table XII. Leakage Power Comparison at Low-Swing (LS) and Full-Swing (FS) of Both 20nm FinFET and 32nm CMOS Technologies in μW

	20nm FinFET		32nm CMOS	
Circuit	LS	FS	LS	FS
s38584	119.5	94.6	694.0	764.2
s38417	140.2	110.2	700.7	762.7
s35932	163.9	126.2	825.8	876.7
Average normalized to FinFET LS	1.00×	0.78×	5.24×	5.67×

of power dissipation in FinFET-based low-swing clocking. The leakage power comparison of low- and full-swing implementations in both FinFET and CMOS technologies is presented in Table XII. It is shown in Table XII that experimental results verify the superiority of FinFET technology in terms of decreasing leakage power consumption. The leakage power increases going from full- to low-swing implementation within FinFET technology due to a larger number of clock buffers. However, it is still more than 5× better compared to both full- and low-swing implementations of CMOS technology.

6. CONCLUSION

In this article, a novel FinFET-based low-swing clock tree design methodology is introduced. It is shown that using FinFET technology methodically in clock tree design can achieve significant power savings. The introduced clock tree design methodology combines the leakage-power-immune structure and low delay characteristics of FinFET technology to achieve substantial power savings while doubling the frequency. It is also shown that the proposed methodology can achieve significant power savings within FinFET technology via low-swing clocking. The clock tree design methodology is combined with a novel low-swing DFF design that can accommodate low-swing clock signals while achieving (almost) the same timing performance and saving further power compared to its full-swing counterpart. The proposed methodology is applicable to both ultra-low-power applications via achieving significant power savings at the same frequency and low-power/high-performance applications via achieving substantial power savings at twice the frequency.

REFERENCES

F. H. A. Asgari and M. Sachdev. 2004. A low-power reduced swing global clocking methodology. *IEEE Trans. VLSI Syst.* 12, 5, 538–545.

C. Auth, C. Allen, A. Blattner, D. Bergstrom, M. Brazier, M. Bost, M. Buehler, V. Chikarmane, T. Ghani, T. Glassman, R. Grover, W. Han, D. Hanken, M. Hattendorf, P. Hentges, R. Heussner, J. Hicks, D. Ingerly, P. Jain, S. Jaloviar, R. James, D. Jones, J. Jopling, S. Joshi, C. Kenyon, H. Liu, R. McFadden, B. Mcintyre, J. Neirynck, C. Parker, L. Pipes, I. Post, S. Pradhan, M. Prince, S. Ramey, T. Reynolds, J. Roesler, J. Sandford, J. Seiple, P. Smith, C. Thomas, D. Towner, T. Troeger, C. Weber, P. Yashar, K. Zawadzki, and K. Mistry. 2012. A 22nm high performance and low-power CMOS technology featuring fully-depleted tri-gate transistors, self-aligned contacts and high density MIM capacitors. In *Proceedings of the Symposium on VLSI Technology (VLSIT'12).* 131–132.

K. D. Boese and A. B. Kahng. 1992. Zero-skew clock routing trees with minimum wire-length. In *Proceedings of the 5th Annual IEEE International ASIC Conference and Exhibit (ASIC'92).* 17–21.

R. Chaturvedi and J. Hu. 2004. Buffered clock tree for high quality IC design. In *Proceedings of the 5th International Symposium on Quality Electronic Design (ISQED'04).* 381–386.

B. S. Doyle, S. Datta, M. Doczy, S. Hareland, B. Jin, et al. 2003. High performance fully-depleted tri-gate CMOS transistors. *IEEE Electron Device Lett.* 24, 4, 263–265.

W. C. Elmore. 1948. The transient response of damped linear networks with particular regard to wideband amplifiers. *J. Appl. Phys.* 19, 1, 55–63.

Y. Hu, Z. Li, and R. Zhou. 2007. A new type of high-performance low-power low clock-swing TSPC flip-flop. In *Proceedings of the 7th International Conference on ASIC (ASICON'07).* 130–133.

ITRS Technology Working Groups. 2010. *International Technology Roadmap for Semiconductors (ITRS) 2010 Update*. ITRS Technology Working Groups.

C. Kim and S. M. Kang. 2002. A low-swing clock double-edge triggered flip-flop. *IEEE J. Solid-State Circ.* 37, 5, 648–652.

N. A. Kurd, J. S. Barkarullah, R. O. Dizon, T. D. Fletcher, and P. D. Madland. 2001. A multigigahertz clocking scheme for the Pentium 4 microprocessor. *IEEE J. Solid-State Circ.* 36, 11, 1647–1653.

D. J. Lee, M. C. Kim, and I. L. Markov. 2010. Low-power clock trees for CPUs. In *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (IC-CAD'10).* 444–451.

P. Li and E. Acar. 2005. A waveform independent gate model for accurate timing analysis. In *Proceedings of the IEEE International Conference on Computer Design (ICCD'05).* 363–365.

H. Mahmoodi-Meimand and K. Roy. 2004. Dual-edge triggered level converting flip-flops. In *Proceedings of the International Symposium on Circuits and Systems (ISCAS'04).* 661–664.

D. Markovic, J. Tschanz, and V. De. 2004. Feasibility study of low-swing clocking. In *Proceedings of the International Conference on Microelectronics (ICMEL'04).* 547–550.

S. Natarajan, M. Armstrong, M. Bost, R. Brain, M. Brazier, C.-H. Chang, V. Chikarmane, M. Childs, H. Deshpande, K. Dev, G. Ding, T. Ghani, O. Golonzka, W. Han, J. He, R. Heussner, R. James, I. Jin, C. Kenyon, S. Klopcic, S.-H. Lee, M. Liu, S. Lodha, B. McFadden, A. Murthy, L. Neiberg, J. Neirynck, P. Packan, S. Pae, C. Parker, C. Pelto, L. Pipes, J. Sebastian, J. Seiple, B. Sell, S. Sivakumar, B. Song, K. Tone, T. Troeger, C. Weber, M. Yang, A. Yeoh, and K. Zhang. 2008. A 32nm logic technology featuring 2nd-generation high-k + metal-gate transistors, enhanced channel strain and 0.171μm2 SRAM cell size in a 291Mb array. In *Proceedings of the IEEE International Electron Devices Meeting (IEDM'08).* 1–3.

J. Pangjun and S. S. Sapatnekar. 2002. Low-power clock distribution using multiple voltages and reduced swings. *IEEE Trans. VLSI Syst.* 10, 3, 309–318.

S. Raja, F. Varadi, M. Becer, and J. Geada. 2008. Transistor level gate modeling for accurate and fast timing, noise, and power analysis. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'08).* 456–461.

E. Salman and E. Friedman. 2012. *High Performance Integrated Circuit Design*. Mc-Graw Hill Professional.

G. Shamanna, N. Kurd, J. Douglas, and M. Morrise. 2010. Scalable, sub-1W, sub-10ps clock skew, global clock distribution architecture for Intel Core i7/i5/i3 microprocessors. In *Proceedings of the IEEE Symposium on VLSI Circuits (VLSIC'10).* 83–84.

R. S. Shelar and M. Patyra. 2013. Impact of local interconnects on timing and power in a high performance microprocessor. *IEEE Trans. Comput.-Aided Des. Integr. Circ. Syst.* 32, 10, 1623–1627.

S. Sinha, G. Yeric, V. Chandra, B. Cline, and Y. Caoi. 2012. Exploring sub-20nm Fin-FET design with predictive technology models. In *Proceedings of the ACM/IEEE Design Automation Conference (DAC'12).* 283–288.

C. Sitik and B. Taskin. 2013a. Multi-corner multi-voltage domain clock mesh design. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI'13).* 209–214.

C. Sitik and B. Taskin. 2013b. Skew-bounded low swing clock tree optimization. In *Proceedings of the ACM Great Lakes Symposium on VLSI (GLSVLSI'13)*. 49–54.

Synopsys Inc. 2012. *Synopsys 32nm Generic Library*. Synopsys Inc.

T. Xanthopoulos, D. W. Bailey, A. K. Gangwar, M. K. Gowan, A. K. Jain, and B. K. Prewitt. 2001. The design and analysis of the clock distribution network for a 1.2 GHz alpha microprocessor. In *Proceedings of the IEEE International Solid-State Circuits Conference (ISSCC'01)*. 402–403.

J. Zhang and Y. Sun. 2007. A low clock swing, power saving and generic technology based D flip-flop with single power supply. In *Proceedings of the International Conference on ASIC (ASICON'07)*. 142–144.

Q. K. Zhu and M. Zhang. 2001. Low-voltage swing clock distribution schemes. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'01)*. 418–421.

Received December 2013; revised July 2014; accepted September 2014