Circuits and Algorithms to Facilitate Low Swing Clocking in Nanoscale Technologies

Weicheng Liu*, Emre Salman*, Can Sitik†, Baris Taskin†, Savithri Sundareswaran‡ and Benjamin Huang‡
*Stony Brook University, Stony Brook, NY 11794
[weicheng.liu, emre.salman]@stonybrook.edu
†Drexel University, Philadelphia, PA 19104
‡Freescale Semiconductor Inc, Austin, TX 78735

***Abstract*—A low swing clocking methodology applicable to industrial ICs and design flows is proposed. The proposed methodology consists of (1) a novel flip-flop cell that enables low swing operation at the sinks without degrading timing performance, (2) a new clock tree synthesis algorithm to satisfy the original skew/slew constraints within a voltage scaled clock network, and (3) skew scheduling method to fix possible timing violations in a gated, low swing clock network. Experimental results on ISCAS'89 benchmark circuits designed with a 45 nm technology demonstrate 37% and 22% reduction in, respectively, clock tree and flip-flop power consumption while satisfying the original timing constraints (*i.e.* same clock frequency, skew and slew) of the full swing clock network.**

I. Introduction

Power consumption has become a critical design objective for almost any application [1]. Low swing signaling is a well known approach to reduce dynamic power in long, capacitive interconnects [2]. Low swing clock distribution has also been proposed since clock networks consume a significant portion of the overall dynamic power [3]–[5]. Existing works on low swing clocking suffer from multiple issues, making these approaches impractical for industrial circuits. These issues are: (1) level shifters are used to restore clock signal back to full swing at the last stage of a clock network (before the clock pin of the flip-flops), significantly sacrificing power savings, (2) important constraints such as skew and slew are not systematically considered, and (3) clock gating is not considered. All of these issues are resolved with the proposed methodology, as described in this paper.

In the proposed low swing clocking methodology, the voltage of the clock network is scaled by utilizing a low voltage power grid whereas the data signals are maintained at full swing to ensure the same performance. The proposed methodology includes a novel low swing flip-flop cell, a new clock tree synthesis algorithm, and a method to exploit useful skew in heavily gated, low swing clock trees. Each of these components alleviates an important issue in low swing clocking, as depicted in Fig. 1.

The rest of the paper is organized as follows. Challenges of low swing clocking and contributions of this work are highlighted in Section II. The proposed methodology and simulation results are presented in Section III. Finally, the paper is concluded in Section IV.

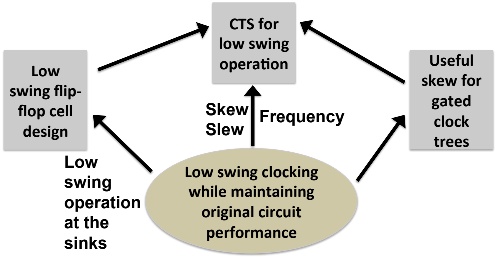

CTS for low swing operation
Low swing flip-flop cell design
Useful skew for gated clock trees
Skew
Slew
Frequency
Low swing operation at the sinks
Low swing clocking while maintaining original circuit performance

Fig. 1. **Components of the proposed low swing clocking methodology.**

II. Challenges and Contributions

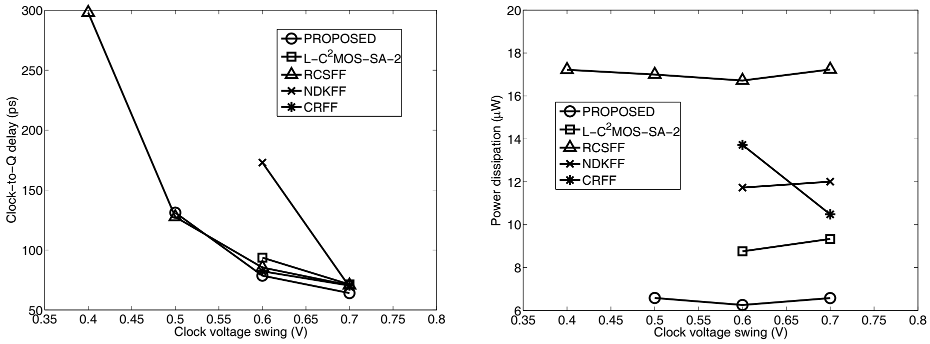
Several challenges exist in applying low swing clocking to industrial circuits with heavily gated clock networks. These challenges and proposed solutions are summarized here.

A. Low Swing Operation at the Sinks: New Flip-Flop Cell

Traditional low swing clocking methodologies restore clock signals back to full swing at the sinks since conventional flip-flops cannot be reliably used with a low swing clock signal. A low swing clock signal either causes significant contention/short circuit current (thereby significantly increasing the power consumption) and/or increase clock-to-Q delay (thereby possibly violating the timing constraints). An important disadvantage of restoring back to full swing signal is a significant reduction in power savings since the last stage of a clock network consumes large power due to high capacitance. Thus, to maximize power savings, a novel flip-flop cell is developed, as further discussed in Section III-A.

B. Satisfy Skew and Slew Constraints: Novel CTS

At scaled supply voltages (as required for low swing operation), clock insertion delay increases, which in turn increases clock skew under variations. Furthermore, the driving ability of the clock buffers is also degraded, which significantly increases clock slew. Satisfying the slew constraint at low swing operation therefore becomes highly challenging. A higher number of clock buffers is typically required which reduces the power savings. Our results demonstrate that an optimum voltage swing level exists beyond which low swing clocking increases overall power due to excessive buffering (assuming tight slew constraints) [6]. The proposed clock tree synthesis (CTS) algorithm for low swing operation is discussed in Section III-B.



(a) (b)

Fig. 2. Comparison of the proposed topology with existing low voltage flip-flop cells: (a) Clock-to-Q delay vs. clock voltage swing, (b) power consumption vs. clock voltage swing.

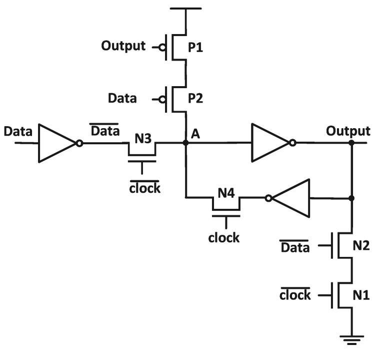


Fig. 3. **Latch within the proposed flip-flop cell that can reliably work with a low swing clock signal (as low as half the nominal supply voltage).**

C. Consider Gated Clock Trees: Exploit Useful Skew

Practical clock networks are heavily clock gated. A common practice is to use integrated clock gating (ICG) cells that consist of a sequential element to prevent glitches in the gated clock signal. An ICG cell has two input pins (clock and Enable signals) and an output pin (gated clock signal). The proposed low swing methodology should efficiently consider gated trees. Despite full swing operation of the data signals, it has been observed that low swing gated clock trees may violate timing constraints within an Enable path. Specifically, in low swing operation, the delay between an ICG and flip-flops (gated by this ICG) increases. Thus, the timing of the Enable path suffers due to skew between the launching flip-flop and the capturing latch (within the ICG). Useful skew is exploited for gated low swing clock trees to alleviate this issue, as further discussed in Section III-C.

III. Proposed Methodology and Results

A. Low Swing Flip-Flop Cell

The first component of the proposed methodology is a novel flip-flop cell that can reliably work with a low swing clock signal while the incoming data signal is at full swing. In a typical flip-flop, clock signals drive both NMOS and PMOS transistors. If the same flip-flop is used with a low swing clock signal, the PMOS transistors driven by the clock signal fail to completely turn off when the clock signal is at logic "high". This behavior significantly affects the operation reliability (high contention current, large glitches, failure in worst-case operating points) of a traditional flip-flop driven by a low swing clock signal [7].

The master latch of the proposed flip-flop is shown in Fig. 3. Note that the slave latch is identical to the master latch. Rather than using transmission gates, pass gate with an NMOS transistor (N3) is utilized as the switch. Thus, when the low swing clock signal is at logic high, N3 can completely turn off. Pass gates with NMOS transistors, however, cannot transfer a full voltage to the output. This issue is critical since the incoming data signal operates at full swing. Thus, node A cannot reach a full V_{DD} which increases the short-circuit and leakage current in the following stages in addition to increasing clock-to-Q delay. To alleviate these issues, a pull-up network consisting of two PMOS transistors (P1 and P2) is added. Finally, a pull-down logic (N1 and N2) is also added to enhance propagation delay and setup time. Specifically, when data and clock signals are at logic low, the pull-down logic is active and quickly pulls the output node to ground, triggering the pull-up network. Thus, node A quickly reaches full V_{DD}.

The proposed flip-flop is compared with existing low voltage flip-flop cells: L-C^{2}MOS-SA-2 [8], RCSFF [9], NDKFF [10], and CRFF [11]. Clock-to-Q delay and power consumption are illustrated for each flip-flop in Fig. 2. According to these results, the proposed topology achieves the least power consumption and clock-to-Q delay until a swing voltage of approximately 0.5 V. At a swing of 0.7 V, an average reduction of 38.1% and 44.4% in, respectively, power consumption and power-delay product is demonstrated.

The proposed flip-flop enables reliable low swing operation at the clock pins of the leaf cells, thereby maximizing the power savings of the proposed methodology. This flip-flop is used by the clock tree synthesis procedure, described in the following section.

B. CTS Algorithm for Low Swing Operation

Enabling low swing clocking on the clock tree is achievable through voltage scaling of the buffers. Clock tree timing performance (*i.e* clock skew and slew), however, is degraded at scaled voltages [12]. Thus, a novel CTS method is required to accommodate low swing operation while satisfying the original skew and slew constraints.

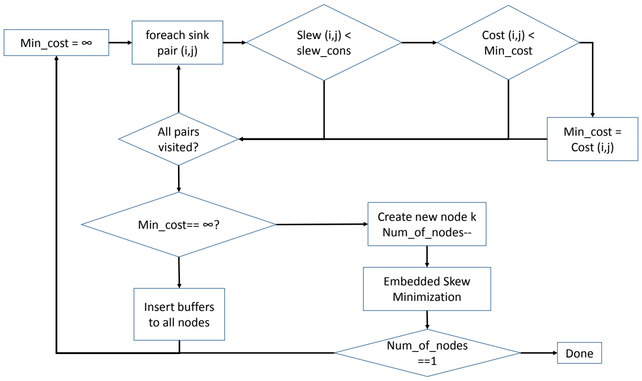


Fig. 4. **Proposed CTS procedure for low swing operation.**

The proposed slew-aware clock tree synthesis algorithm (shown in Fig. 4) adopts the well-known zero-skew-tree deferred merge embedding (ZST-DME) method [13] to merge two nodes into one at each step. The merging cost is inspired by [14], 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. The algorithm starts with initializing the minimum cost as infinity. It is identified whether a merging pair is feasible. If a feasible pair is identified, it is checked if it has the minimum cost, and the minimum cost is updated. This process continues until all pairs are processed. After all pairs are processed, the minimum cost is checked. If minimum cost is infinity, no feasible pairs are found. In this case, a buffering process is followed to help satisfy the slew constraint. Otherwise, the feasible minimum cost pair is merged and initialized as a new node. After the delay, slew and the capacitance of this new node are initialized, it is checked if the difference between the maximum and the minimum is larger than the skew constraint $skew_{const}$. The case of a skew violation is solved through moderate wire snaking for small violations and through buffer insertion for larger violations. 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. This procedure continues until the number of unmerged nodes is one, which is the source of the clock tree. The combined results of the proposed CTS procedure and flip-flop cell are presented in Section III-D.

C. Useful Skew in Gated Low Swing Clock Trees

As mentioned previously, data signals operate at full swing voltage level to maintain the clock frequency constant. Furthermore, the proposed flip-flop does not degrade timing performance despite operating at reduced clock swings. It is, however, observed that the timing performance can still be degraded in low swing operation due to heavy clock gating that is common in practical clock networks.

To better illustrate this point, a simple clock gated circuit is shown in Fig. 5. A conventional data path is from R0 to R1. Conventional clock launch and capture paths are from the clock source to the clock pins of R0, R1, and ICG0 (clock gating cell). The Enable path consists of R1/R3 to ICG0. In addition to these traditional paths, a timing path exists between ICG0 and R2/R3. This path (due to local clock tree) is referred to as the clock propagation path. The delay of this path can be significant when the ICG cell has a large fanout and drives a large number of flip-flops. In low swing clocking, this delay further increases due to lower supply voltage of the clock buffers (B0 and B1 in Fig. 5). Thus, significant skew occurs at the clock pins of R3 and ICG0. This skew degrades the timing characteristics (reduces positive slack) of the Enable path. If it is sufficiently large, timing violation occurs. To fix this issue, useful skew is exploited in this work.

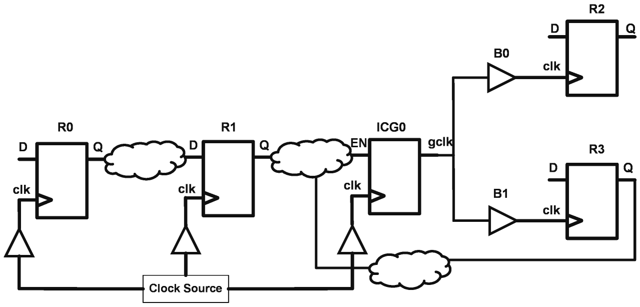


Fig. 5. **Different types of timing paths in a clock gated design.**

TABLE I
LP BASED APPROACH TO EXPLOIT USEFUL SKEW IN A GATED DESIGN.

LP based approach for ICs with clock gating	
Inputs: paths delays and clock period	
Output: skew schedule	
Objective: max $\Sigma(t_{reg,i} - t_{icg,j})$ for single j	
1	$t_i - t_j \geq -DP_{min}(data\ path)$
2	$t_i - t_j \leq T - DP_{max}(data\ path)$
3	$\boldsymbol{t_{icg,j} - t_i \geq -CP_{max}}(propagation\ path)$
4	$\boldsymbol{t_{icg,j} - t_i \leq -CP_{min}}(propagation\ path)$
5	$\boldsymbol{t_i - t_{icg,j} \geq -EP_{min}}(enable\ path)$
6	$\boldsymbol{t_i - t_{icg,j} \leq T - EP_{max}}(enable\ path)$
7	$0 \leq t_i, t_{icg,j} \leq T$

Existing skew scheduling methods cannot be directly used since nonzero clock propagation paths are not considered in prior work [15]. In the proposed approach, each ICG cell is treated as a flip-flop with an associated clock arrival time. The clock propagation delays (between ICG and flip-flops) are treated as clock skew [16]. However, note that the clock signal should arrive to the ICG cell earlier than it arrives to the flip-flops gated by this ICG cell due to positive clock propagation path delay. This constraint is different than conventional data paths where skew can be both positive and negative. Combining this constraint with the traditional, data path related constraints, an improved linear programming solution for skew scheduling in ICs with gated clock trees is obtained, as listed in Table I. The bold lines represent the *new* constraints required for gated clock networks.

At constant clock period, the objective function of skew schedule is to maximize the ICG to flip-flop (gated by this ICG) delay. Thus, after skew scheduling, more delay can be tolerated between the ICG and flip-flops. This characteristic is highly useful for low swing clocking since larger delay possibly causes a timing violation within the Enable path.

TABLE II

THE COMPARISON OF CLOCK TREE POWER (CP IN MW), FLIP-FLOP POWER (DFFP IN MW), CLOCK SKEW (SK. IN PS), CLOCK SLEW (SL. IN PS) AND THE CLOCK-TO-Q DELAY (C2Q IN PS) FOR THE BASELINE FULL SWING (FS) AND PROPOSED LOW SWING (LS) METHODOLOGY.

Circuits	Full Swing (FS)					Proposed Low Swing (LS)				
	CP	DFFP	Sk.	Sl.	C2Q	CP	DFFP	Sk.	Sl.	C2Q
s1423	0.284	0.269	0.2	48.2	166.7	0.121	0.210	0.3	101.2	168.8
s9234	0.365	0.528	0.3	63.8	169.3	0.165	0.412	0.3	133.7	173.0
s5378	0.421	0.637	0.1	80.4	172.9	0.333	0.499	14.3	123.6	172.0
s15850	1.255	1.885	8.3	127.7	180.4	0.786	1.456	12.3	138.3	172.8
s13207	1.343	2.176	28.4	138.0	182.4	0.834	1.686	20.1	140.8	173.8
s38584	3.189	4.656	37.9	118.6	178.5	2.014	3.649	22.5	141.3	172.9
s38417	3.685	5.749	46.9	125.3	179.8	2.319	4.519	39.9	140.3	173.0
s35932	3.908	6.318	9.2	121.3	179.9	2.581	4.959	44.9	142.6	173.3
Norm.	1.00	1.00	<50ps	<150ps	176.2	0.63	0.78	<50ps	<150ps	172.5

The proposed approach is evaluated using the largest ISCAS'89 benchmark circuits. ICG cells are inserted by the tool during the synthesis stage. According to the results, on average, the tolerable delay between ICG and flip-flops increases by 47%, thereby compensating the increase in delay due to low swing clocking.

D. Reduction in Power Consumption

The proposed CTS algorithm is implemented in *Perl* and the new flip-flop cell is custom designed with NCSU 45nm technology [17]. The output circuits are tested using HSPICE. Eight largest circuits from ISCAS'89 benchmarks are selected. The logic synthesis and the physical placement of flip-flop sinks are performed using *SoC Encounter*. The largest size buffer (NBUFFX32) and the wire model (with per unit values of R_{unit}=2Ω/μm and C_{unit}=0.1$fF/\mu m$) are obtained from SAED library [18]. The experiments are performed at the slowest corner (SS/0.9V/125°C) to verify functionality in reduced noise margins. Frequency is set to 1GHz with 150ps slew (15% of the period) and 50ps skew (5% of the clock period) constraints. The low swing voltage level is selected as the level that has minimum power consumption, which is empirically determined as $0.65\times V_{dd}$.

The power consumption of clock tree, the overall power consumption of flip-flop cells, clock skew, maximum clock slew and the maximum clock-to-Q delay of flip-flops are compared against a full swing clock tree with traditional full swing flip-flops. The results are presented in Table II. The proposed slew-aware methodology satisfies both skew and slew constraints simultaneously, while achieving, on average, a significant 37% power savings. Furthermore, the proposed low swing flip-flop achieves, on average, 22% power savings, while providing smaller clock-to-Q delay.

IV. CONCLUSIONS

Methodology has been proposed to facilitate low swing clocking in practical circuits with heavy clock gating. The limitations of existing low swing approaches have been alleviated through a novel flip-flop cell design, a slew-aware CTS algorithm to satisfy skew and slew in low swing clock trees, and a method to exploit useful skew to fix timing violations. Experimental results demonstrate 37% reduction in power consumption while satisfying the original timing constraints (frequency, skew, and slew).

REFERENCES

[1] E. Salman and E. G. Friedman, *High Performance Integrated Circuit Design*. McGraw-Hill, 2012.

[2] H. Zhang, V. George, and R. J.M., "Low-swing on-chip signaling techniques: effectiveness and robustness," *IEEE Transaction on Very Large Scale Integration (VLSI) Systems*, vol. 8, no. 3, pp. 264–272, 2000.

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

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

[5] C. Sitik, E. Salman, L. Filippini, S. J. Yoon, and B. Taskin, "FinFET-Based Low Swing Clocking," *ACM Journal on Emerging Technologies in Computing Systems*, in press.

[6] C. Sitik, L. Filippini, E. Salman, and B. Taskin, "High Performance Low Swing Clock Tree Synthesis with Custom D Flip-Flop Design," in *Proceedings of the IEEE Computer Society Annual Symposium on VLSI*, July 2014, pp. 498–503.

[7] M. Rathore, W. Liu, E. Salman, C. Sitik, and B. Taskin, "A novel static d-flip-flop topology for low swing clocking," in *Proceedings of the ACM Great Lakes Symposium on VLSI*, May 2015, pp. 301–306.

[8] Z. Jianjun and S. Yihe, "A low clock swing, power saving and generic technology based d flip-flop with single power supply," in *IEEE International Conference on ASIC*, Oct. 2007, pp. 142–144.

[9] H. Kawaguchi and T. Sakurai, "A reduced clock-swing flip-flop (rcsff) for 63power reduction," *IEEE Journal of Solid-State Circuits*, vol. 33, no. 5, pp. 807–811, 1998.

[10] M. Tokumasu, H. Fujii, M. Ohta, T. Fuse, and A. Kameyama, "A new reduced clock-swing flip-flop: Nand-type keeper flip-flop (ndkff)," in *Proceedings of the IEEE custom integrated circuits conference*, 2002, pp. 129–132.

[11] D. Levacq, M. Yazid, H. Kawaguchi, M. Takamiya, and T. Sakurai, "Half vdd clock-swing flip-flop with reduced contention for up to 60clock distribution," in *Proceedings of the IEEE European Solid State Circuits Conference*, 2007, pp. 190–193.

[12] 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.

[13] 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.

[14] 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.

[15] W.-P. Tu, S.-H. Huang, and C.-H. Cheng, "Co-synthesis of data paths and clock control paths for minimum-period clock gating," in *Design, Automation And Test in Europe Conference And Exhibition (DATE)*, March 2013, pp. 1831–1836.

[16] W. Liu, E. Salman, C. Sitik, and B. Taskin, "Clock skew scheduling in the presence of heavily gated clock networks," in *Proceedings of the ACM Great Lakes Symposium on VLSI*, May 2015, pp. 283–288.

[17] *FreePDK 45nm Library v1.4*, NCSU, 2011.

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