Exploiting Useful Skew in Gated Low Voltage Clock Trees

Weicheng Liu and Emre Salman
Department of Electrical and Computer Engineering
Stony Brook University, Stony Brook, NY 11794
[weicheng.liu, emre.salman]@stonybrook.edu

Can Sitik and Baris Taskin
Department of Electrical and Computer Engineering
Drexel University, Philadelphia, PA 19104
as3577@drexel.edu, taskin@coe.drexel.edu

***Abstract*—Low swing/voltage clocking is a well-studied approach to reduce dynamic power consumption in clock networks. It is, however, challenging to maintain the same performance at scaled clock voltages due to timing degradation in the Enable paths that are required for clock gating, another highly popular method to reduce dynamic power. A useful skew methodology is proposed in this paper to increase the timing slack of the Enable paths when the clock network is operating at a lower swing voltage. The skew schedule is determined via linear programming. The methodology is evaluated on five largest ISCAS'89 benchmark circuits. The results demonstrate an average 47% increase in the timing slack of the Enable path, thereby facilitating low swing operation without degrading performance.**

I. INTRODUCTION

Power consumption has become one of the primary concerns for almost any application due to increased design complexity, higher integration, and difficulty in scaling the power supply voltage [1]. A well-known approach to minimize the overall on-chip power dissipation is to reduce the supply voltage [2]. For example, near-threshold computing has received considerable attention to achieve optimal energy efficiency [3]. A reduction in supply voltage, however, degrades IC performance, particularly when the nominal supply voltages are low. An alternative approach is to scale only the supply voltage of a clock network, *i.e.*, low voltage/swing clocking since the clock network has significantly high switching capacitance [4]–[7]. Clock networks operating at near-threshold voltages have also been investigated [8].

Existing works on low swing/voltage clocking, however, do not consider the performance requirements of the IC. In practice, low swing clocking introduces two issues related to performance: 1) possible degradation in clock-to-Q delay of the flip-flops due to low swing clock signal, 2) timing degradation in the Enable paths of a gated clock network due to higher clock insertion delay. A possible solution to the first issue is to restore the full swing/voltage operation before the clock signal reaches flip-flops [6]. This approach reduces the power savings since the last stage of a clock network has high switching capacitance. In [9], a novel flip-flop topology has been proposed for a low swing clock signal without any degradation in clock-to-Q delay, addressing issue one. A solution based on useful skew is proposed in this paper to address issue two.

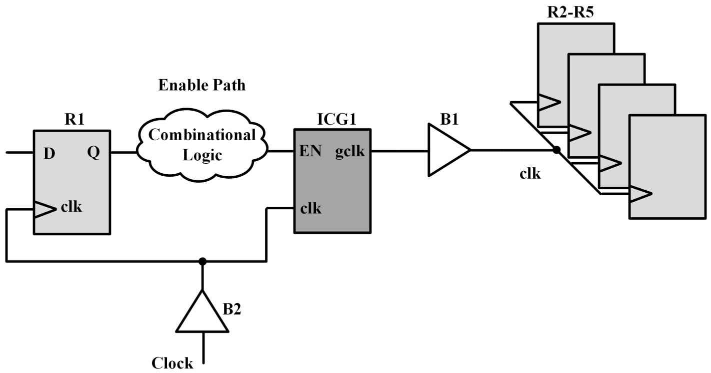

Fig. 1. **A simplified gated clock network consisting of five sinks, an integrated clock gating (ICG) cell, and an Enable path.**

Clock gating is a standard method to reduce dynamic power by deactivating the clock signals of the idle flip-flops. Thus, proposed methods based on low voltage/swing clocking should be able to consider clock gating. A simplified gated clock network is shown in Fig. 1. The clock signal can be gated by an integrated clock gating (ICG) cell, depending upon the Enable path. If the Enable signal is active, the output of the ICG cell (gated clock signal) does not switch even though the clock signal switches, thereby reducing dynamic power in certain clock nets. Unfortunately, a low voltage/swing clock signal degrades the timing of the Enable path even when the data signals are at nominal voltage, as further described in Section II-A.

A useful skew approach is formulated in this paper for gated clock networks operating at a reduced voltage. The methodology is evaluated on largest ISCAS'89 benchmark circuits, demonstrating that the useful skew can effectively fix timing violations (introduced due to low voltage/swing operation) within the Enable paths.

The rest of the paper is organized as follows. Background on low swing operation and problem formulation are provided in Section II. The proposed method is described in Section III. Experimental results on several large ISCAS'89 benchmark circuits are presented in Section IV. Finally, in Section V, the paper is concluded.

This research is supported in part by Semiconductor Research Corporation under Contract No. 2013-TJ-2449 and 2013-TJ-2450.

978-1-4799-5341-7/16/$31.00 ©2016 IEEE

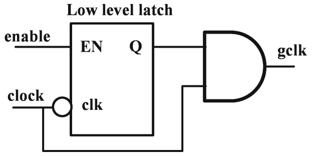

Fig. 2. **Integrated clock gating cell**

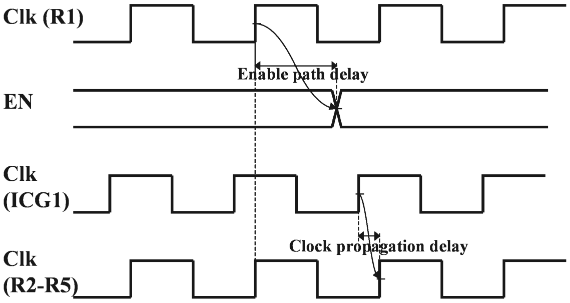

Fig. 3. **Timing graph of the gated clock network shown in Fig. 1.**

II. BACKGROUND

A. Low Swing Operation and Problem Formulation

In gated clock networks, each ICG cell creates a timing path for the Enable signals. Note that an ICG cell consists of a latch, as shown in Fig. 2. Unlike conventional data paths, the output of an Enable path is a clock signal. In practice, an ICG cell can drive a large number of flip-flops. Thus, it is common to have a local clock tree between the ICG and the sinks driven by this ICG. In Fig. 1, this local tree is simply represented by buffer B1.

When the clock voltage is reduced, the delay from ICG to the flip-flops (R2-R5) increases. Thus, clock signal arrives at ICG much earlier than the sinks R2-R5. Assuming negligible clock skew, clock signal arrives at R1 at approximately the same time as R2-R5. Thus, clock signal arrives at ICG (capturing latch of the Enable path) earlier than the R1 (launching flip-flop of the Enable path). Thus, the timing slack of the Enable path is *reduced* by this difference. This issue places a practical limitation on low voltage clocking if performance needs to be maintained.

To better illustrate this issue, signal waveforms at different clock nets and Enable signal are shown in Fig. 3, assuming zero clock skew. During the second clock period, Enable signal changes. Referring to this figure, the Enable paths should satisfy the following max delay constraint,

$$t_{EN} + t_{ICG\ setup} + t_{clock\ propagation} < T_{clock}, \tag{1}$$

where t_{EN}, $t_{ICG\ setup}$ and $t_{clock\ propagation}$ are, respectively, Enable path delay, ICG cell setup time, and clock propagation delay within the local clock tree. The sum of these three variables should be less than one clock period. Since clock signal should always arrive at the ICG cell earlier than the flip-flops gated by this ICG cell (determined by $t_{clock\ propagation}$ that is always positive), the timing slack of the Enable path is reduced. In low swing operation, due to the increase in clock buffer delays, clock propagation delay $t_{clock\ propagation}$ increases. Thus, the reduction in the timing slack of the Enable path is more pronounced. The proposed linear programming based useful skew methodology makes low swing operation more practical by alleviating this timing degradation of the Enable paths.

B. Traditional Useful Skew without Clock Gating

In a sequential timing path P, assume R_i and R_j represent two registers, t_i and t_j are clock arrival times for registers R_i and R_j, respectively. For each data path P in the circuit, two types of timing constraints exist: setup time (max delay) and hold time (min delay) constraints, which are represented, respectively, by (2) and (3),

$$t_i - t_j \leq T - DP_{max}, \tag{2}$$

$$t_i - t_j \geq -DP_{min}, \tag{3}$$

where T is the clock period, DP_{max} and DP_{min} are the maximum and minimum data path delays that include setup and hold time, respectively [10], [11].

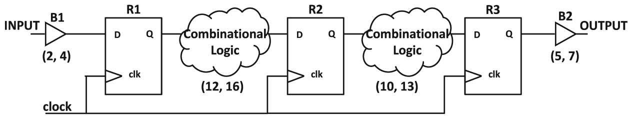

Fig. 4. **Simple sequential circuit consisting of three registers without clock gating.**

A simple sequential circuit with three registers $R1$, $R2$ and $R3$ and without clock gating is shown in Fig. 4. Two buffers $B1$ and $B2$ are inserted at the primary input and the output load, respectively. A pair of delay values (D_{min}, D_{max}) is denoted with each buffer, where $D_{min,buf}$ and $D_{max,buf}$ are the minimum and maximum propagation delay of the buffer, respectively. There are two data paths in this circuit, $R1 \rightarrow R2$ and $R2 \rightarrow R3$, which are also associated with a pair of delay values $(DP_{min,path}, DP_{max,path})$ representing minimum and maximum data path delays.

Conventional useful skew approaches find a set of clock arrival times corresponding to each register, which should satisfy each data path's timing constraints represented by (2) and (3). In [12], the clock skew scheduling methodology is formulated as a simple linear programming (LP) problem where the objective function is to minimize the clock period. For the motivational example shown in Fig. 4, this formulation determines the minimum clock period as 10 units after skew scheduling.

III. PROPOSED METHODOLOGY

A. Useful Skew with Clock Gating

As mentioned in Section II-A, in practice, one ICG cell gates multiple registers since an ICG cell placed at higher levels of a clock tree can save more dynamic power. Thus, in industrial designs, it is common to have *a local clock tree* between an ICG cell and the registers that are gated by this ICG cell. A *clock propagation path* on the local clock tree is

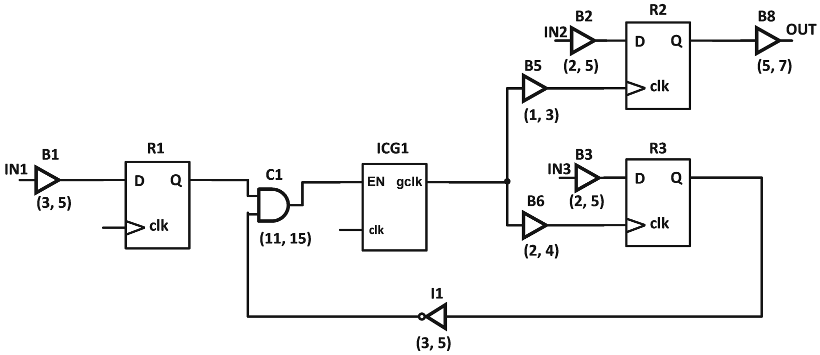

Fig. 5. **Simple sequential circuit consisting of an ICG cell, two registers gated by this ICG cell, a local clock sub-tree, and a timing loop formed by clock propagation path and clock Enable path.**

therefore defined as the path from the output pin of an ICG cell to the clock pin of a register that is gated by this ICG cell. Since an ICG cell typically gates multiple registers, there are more than one clock propagation paths for an ICG cell. The delay of the clock propagation path (the delay between the clock arrival time to the ICG cell and the clock arrival time to the register gated by this ICG cell) is at least the ICG cell delay and is bounded by the longest path within the local tree. Thus, each ICG cell is associated with a lower and upper bound of clock propagation path delay [13].

A simplified motivational example with clock gating is shown in Fig. 5 to better illustrate the aforementioned definitions. For simplicity, the circuit in this example has one ICG cell $ICG1$, gating two registers $R1$ and $R2$. A local subtree including two buffers $B5$ and $B6$ is synthesized to drive the two registers. Each buffer is denoted with a pair of delay values, which indicates the minimum and the maximum clock propagation path delays. The clock Enable (or control) path is from $R1$ to $ICG1$ and consists of a single combinational gate, $C1$. Note that for simplicity, data paths are omitted in this example so that the issues related with clock gating can be emphasized.

Conventional useful methodologies cannot consider the unique challenges introduced by clock gating. In [14], the authors have recently proposed a linear programming approach to investigate the clock gated designs. In this work, useful skew is utilized in a gated design via considering both the data paths and clock Enable paths with the objective function of minimum insertion delay [14]. However, it is assumed that the clock arrival time to an ICG cell is the same as the clock arrival time to the registers gated by this ICG cell. This assumption is impractical since in practice, the clock signal is distributed with a local clock tree that has larger and non-identical clock propagation delays (as depicted in Fig. 5). A method to exploit useful skew in clock gated design with a local sub-tree is proposed in this paper, as described in the following section. Useful skew is exploited to increase the timing slack of the Enable paths.

TABLE I
PROPOSED LP BASED APPROACH TO EXPLOIT USEFUL SKEW IN LOW SWING OPERATION.

LP based approach for ICs with clock gating	
Inputs: path delays and clock period	
Outputs: 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	$0 \leq t_i, t_{icg,j} \leq T$

B. Linear Programming Based Useful Skew

The arrival time of a clock signal to a flip-flop gated by an ICG cell is larger than the arrival time of the clock signal to the ICG cell (see Fig. 5). The lower bound for each clock propagation path delay is determined by the AND gate delay and a local clock tree. This inequality is given by,

$$t_{icg,j} - t_i \leq -CP_{min}, \tag{4}$$

where $t_{icg,j}$ and t_i are the clock arrival times to ICG cell ICG_j and register R_i, respectively. CP_{min} is the minimum clock propagation path delay.

An upper bound on clock propagation path delay is also required to represent the maximum delay of the local clock tree,

$$t_i - t_{icg,j} \leq CP_{max}, \tag{5}$$

where CP_{max} is the maximum delay of the corresponding clock propagation path. Combining the constraints in (4) and (5) with the traditional, data path related constraints, an improved linear programming (LP) solution for useful skew in ICs with gated clock trees is obtained, as shown in Table I. It is important to note that the objective function of the LP is to maximize the ICG-to-DFF delay, thereby increasing the timing slack of the Enable paths. In other words, more delay can be tolerated between ICG and flip-flops.

The bold lines represent the new constraints required for gated clock networks. The first two lines are the data path related constraints whereas lines 3 and 4 are the constraints related with clock propagation paths. Lines 5 and 6 represent the timing constraints of the Enable path. Line 7 is added to limit the global skew within one clock period. The linear programming based solution for the motivational example in Fig. 5 is listed in Table II after substituting clock period with

22 time units. The program determines a set of clock arrival times as $t_1 = 8$, $t_2 = 22$, $t_3 = 21$, $t_{icg,1} = 19$, and $t_{host} = 20$, where $ICG1$ to $R2$ delay increases from 1 to 3 time units, whereas $ICG1$ to $R3$ delay remains the same as 2 time units.

TABLE II
APPLICATION OF THE LP BASED APPROACH TO CIRCUIT SHOWN IN FIG. 5 TO INCREASE THE TIMING SLACK OF THE ENABLE PATHS.

LP based approach for ICs with clock gating
Objective: max $(t_2 - t_{icg,1}$+$t_3 - t_{icg,1})$
s.t. $-3 \leq t_{host} - t_1 \leq 17$
$-2 \leq t_{host} - t_2 \leq 17$
$-2 \leq t_{host} - t_3 \leq 17$
$-5 \leq t_2 - t_{host} \leq 15$
$-3 \leq t_{icg,1} - t_2 \leq -1$
$-4 \leq t_{icg,1} - t_3 \leq -2$
$-11 \leq t_1 - t_{icg,1} \leq 7$
$-14 \leq t_3 - t_{icg,1} \leq 2$
$0 \leq t_1, t_2, t_3, t_{icg,1}, t_{host} \leq 22$

IV. EXPERIMENTAL RESULTS

The proposed useful skew approach to facilitate low swing clocking without degrading the timing slack of the Enable paths is evaluated with the five largest ISCAS'89 benchmark circuits consisting of up to approximately 2000 registers. Synopsys Design Compiler [15] is used to synthesize these benchmark circuits with the 45 nm NanGate open cell library [16]. ICG cells are automatically inserted by the tool during clock tree synthesis. An open source linear programming kit GLPK [17] is used as the linear programming solver, running on a Linux server with Intel Xeon processor. The longest runtime is 120 seconds for s38417.

The experimental results comparing the zero skew and useful skew are listed in Table III. Both zero skew and useful skew cases operate at the same clock period which is the minimum clock period in zero skew. Depending upon the mismatch in the datapaths, up to 86% improvement in Enable slack can be achieved. On average, the slack of the Enable path is increased by 47% after applying the proposed useful skew approach.

The maximum ICG-to-DFF delays are listed in Table IV when the clock period is the minimum theoretical clock period *after useful skew*. If the proposed useful skew approach is adopted in this case, the maximum ICG-to-DFF delays are degraded on average, by 20% as compared to the useful skew case in Table III (with a larger clock period). This result is expected since in Table IV, there is a tighter constraint for clock period.

TABLE III
EXPERIMENTAL RESULTS DEMONSTRATING THE INCREASE IN THE SLACK OF THE ENABLE PATHS AFTER EXPLOITING USEFUL SKEW.

Circuit	Clock Period (ns)	Max ICG-to-DFF Delay (ns)		Increase in Enable Slack
		Zero Skew	Useful Skew	
s13207	2.71	2.13	2.36	10.8%
s15850	4.93	2.78	3.72	33.8%
s35932	3.83	1.20	2.24	86.7%
s38417	4.85	3.47	4.85	39.8%
s38584	4.61	2.12	3.48	64.2%

TABLE IV
EXPERIMENTAL RESULTS DEMONSTRATING THE INCREASE IN THE SLACK OF THE ENABLE PATHS AFTER EXPLOITING USEFUL SKEW AT THE MINIMUM CLOCK PERIOD.

Circuit	Min Clock Period (ns)	Max ICG-to-DFF Delay (ns)	Run Time (s)
s13207	2.26	1.91	0
s15850	4.13	2.91	5
s35932	3.04	1.45	7
s38417	4.66	4.56	120
s38584	4.04	2.92	25

V. CONCLUSION

Enable paths can become a performance bottleneck in low swing/voltage clocking due to higher clock delay from ICG cells to registers. A useful skew methodology has been proposed to increase the timing slack of the Enable paths in gated low swing clock trees. The proposed methodology has been implemented via linear programming and demonstrated with largest ISCAS'89 benchmark circuits. The proposed approach facilitates low swing/voltage clocking in gated designs without degrading performance.

REFERENCES

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

[2] R. Gonzalez, B. M. Gordon, and M. A. Horowitz, "Supply and Threshold Voltage Scaling for Low Power CMOS," *IEEE Journal of Solid-State Circuits*, vol. 32, no. 8, pp. 1210–1216, August 1997.

[3] R.G. Dreslinski *et al.*, "Near-Threshold Computing: Reclaiming Moore's Law Through Energy Efficient Integrated Circuits," *Proceedings of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[4] 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*, vol. 12, no. 2, pp. 13:1–13:20, August 2015.

[5] C. Sitik, W. Liu, B. Taskin, and E. Salman, "Design Methodology for Voltage-Scaled Clock Distribution Networks," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, to appear.

[6] J. Pangjun and S. 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, 2002.

[7] F. H. 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.

[8] J. R. Tolbert, X. Zhao, S. K. Lim, and S. Mukhopadhyay, "Analysis and design of energy and slew aware subthreshold clock systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, no. 9, pp. 1349–1358, September 2011.

[9] 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/IEEE Great Lakes Symp. on VLSI*, May 2015, pp. 301–306.

[10] E. Salman, A. Dasdan, F. Taraporevala, K. Kucukcakar, and E. G. Friedman, "Exploiting setup–hold-time interdependence in static timing analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 26, no. 6, pp. 1114–1125, 2007.

[11] A. Dasdan, E. Salman, F. Taraporevala, and K. Kucukcakar, "Characterizing sequential cells using interdependent setup and hold times, and utilizing the sequential cell characterization in static timing analysis," US Patent No 7,506,293.

[12] J. P. Fishburn, "Clock skew optimization," *Computers, IEEE Transactions on*, vol. 39, no. 7, pp. 945–951, 1990.

[13] 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/IEEE Great Lakes Symp. on VLSI*, May 2015, pp. 283–288.

[14] 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 *Proceedings of the ACM Conference on Design, Automation and Test in Europe*, 2013, pp. 1831–1836.

[15] Synopsys. Design compiler. http://www.synopsys.com.

[16] NanGate. 45nm open cell library. http://www.nangate.com.

[17] GNU. Gnu linear programming kit. https://www.gnu.org/software/glpk/.