Clock Skew Scheduling in the Presence of Heavily Gated Clock Networks

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

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

ABSTRACT

Clock skew scheduling is a common and well known technique to improve the performance of sequential circuits by exploiting the mismatches in the data path delays. Existing clock skew scheduling techniques, however, cannot effectively consider heavily gated clock networks where a local clock tree exists between clock gating cells and registers. A methodology is proposed in this paper to efficiently achieve clock skew scheduling in circuits with gated clock networks. The methodology is implemented via both linear programming and constraint graph based approaches, and evaluated using the largest ISCAS'89 benchmark circuits with clock gating. The results demonstrate up to approximately 21% reduction in clock period while maintaining the power savings achieved by clock gating. A conventional design flow is used for the experiments, demonstrating the applicability of the proposed algorithms to automation.

Categories and Subject Descriptors

B.7 [**Integrated Circuits**]: VLSI (very large scale integration)

General Terms

Design

Keywords

Clock Skew Scheduling, Clock Gating, Low Power

1. INTRODUCTION

In IC design process, clock distribution networks are vital to synchronize all of the sequential elements in a circuit [1]. Due to the process-voltage-temperature (PVT) variations and design margins, the arrival time of the clock signal to each sequential element (latch or flip-flop) is not identical, resulting in non-zero clock skew [2–4]. Historically, the clock skew is managed in three ways: i) zero skew, ii) bounded skew and iii) useful skew approaches, *i.e.*, clock skew scheduling.

The zero skew and bounded skew approaches ensure that the clock arrival time of all of the sequential elements is either identical (for zero skew) or within a margin (for bounded skew). Alternatively, the useful skew approach considers clock skew scheduling where the skew of each sequential element that belongs to the same timing path is individually considered for timing optimization. In clock skew scheduling, the available timing slack at each sequential element is utilized to improve clock period of the IC. Specifically, slower data paths "borrow" time from faster data paths. Thus, skew scheduling exploits the mismatches in the timing characteristics of the data paths to decrease clock period.

Conventional clock skew scheduling techniques rely on linear programming (LP) with a minimum clock period objective [2,5,6] or a graph-based solution to utilize existing graph algorithms [7,8]. In [9], delay insertion methodology in clock skew scheduling is proposed. In [10], a linear programming approach is proposed to minimize the overall delay insertion while maintaining the minimum clock period. In order to mitigate the effect process variations on skew, multi-domain clock skew scheduling [11] is proposed. In [12, 13], two optimal algorithms are developed to implement a multi-domain clock skew scheduling.

The global clock signal has the highest switching activity in an IC, making clock gating a popular technique to reduce dynamic power. Although clock gating is shown to be effective [14], it may introduce timing related challenges. One such challenge is to utilize useful skew since conventional clock gating structures consider zero skew. Furthermore, the timing constraints (setup and hold) and the insertion delay of the local clock tree (between clock gating cells and registers) produce other challenges that need to be addressed during clock skew scheduling, as discussed in this paper.

Existing clock skew scheduling methods (including those mentioned above) consider only *non-gated* clock distribution networks, which is impractical since industrial clock trees are heavily clock gated. A recent methodology proposed in [15] performs clock skew scheduling in a clock gated design where a linear programming framework is used with a minimum insertion delay objective. However, it is assumed that each sequential element has an individual clock gate, which is not practical in modern industrial designs. Fur-

GLSVLSI'15, May 20–22, 2015, Pittsburgh, PA, USA
Copyright 2015 ACM 978-1-4503-3474-7/15/05 ...$15.00
http://dx.doi.org/10.1145/2742060.2742092.

thermore, the clock arrival time at the clock gate and its corresponding sequential element are assumed to be identical, which is typically not the case when there is a local clock tree after the clock gating cell. Thus, a practical clock skew scheduling approach for gated clock trees is required, as proposed in this paper. Specifically, the challenges of skew scheduling in the presence of gated clock networks are addressed to improve timing performance (clock period) while maintaining the power savings achieved by clock gating.

The rest of the paper is organized as follows. Traditional clock skew scheduling methods that utilize linear programming and constraint based approaches are summarized in Section 2. The challenges introduced by clock gating are also discussed. The proposed method is described in Section 3. Experimental results on largest ISCAS'89 benchmark circuits are presented in Section 4. Finally, paper is concluded in Section 5.

2. BACKGROUND AND PROBLEM FORMULATION

Traditional clock skew scheduling is briefly summarized in Section 2.1. Unique challenges introduced due to clock gating are discussed in Section 2.2.

2.1 Traditional Clock Skew Scheduling

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 (1) and (2),

$$t_i - t_j \leq T - DP_{max}, \tag{1}$$

$$t_i - t_j \geq -DP_{min}, \tag{2}$$

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 [16].

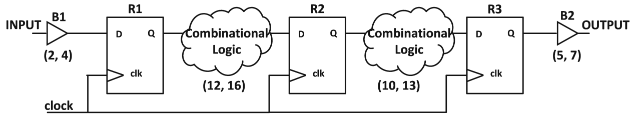


Figure 1: Simple sequential circuit consisting of three registers without clock gating.

A simple sequential circuit with three registers $R1$, $R2$ and $R3$ and without clock gating is shown in Fig. 1. 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.

Table 1: LP based formulation of skew scheduling for the simple circuit shown in Fig. 1.

LP based formulation	
Objective: min T	
1	$-12 \leq t_1 - t_2 \leq T - 16$
2	$-10 \leq t_2 - t_3 \leq T - 13$
3	$-2 \leq t_{host} - t_1 \leq T - 4$
4	$-5 \leq t_3 - t_{host} \leq T - 7$
5	$0 \leq t_1, t_2, t_3, t_{host} \leq T$

Conventional clock skew scheduling approaches find a set of clock arrival times corresponding to each register, which should satisfy each data path's timing constraints represented by (1) and (2). In [5], the proposed clock skew scheduling methodology is formulated as a simple linear programming (LP) problem where the objective function is to minimize the clock period. The linear programming model of the motivational example shown in Fig. 1 is listed in Table 1. Lines 1 to 4 represent the timing constraints of the two data paths and the primary input and the primary output paths. Line 5 is included to limit the maximum global skew within one clock period. The linear programming determines the minimum clock period as 10 units with the following set of skew schedule: $t_1 = 0$, $t_2 = 6$, $t_3 = 9$ and $t_{host} = 6$.

In addition to utilizing linear programming to perform clock skew scheduling, a sequential circuit can also be modeled as a constraint graph $G(V, E)$, in which each vertex represents a register and two edges (with opposite directions) connecting two vertices represent setup and hold time constraints, respectively. In [8], a constraint graph based approach is proposed to optimize clock skew. In this graph-based approach, each data path from R_i to R_j in a sequential path has two edges: 1) an edge (R_j, R_i) with weight $T - DP_{max}$ models the setup time constraint in (1) and 2) an edge (R_i, R_j) with weight DP_{min} models the hold time constraint in (2). In order to synchronize the primary input and the primary output, a special vertex $Host$ is added. This constraint graph provides skew schedule only if no negative weight cycle exists in the constraint graph. The well-known Bellman-Ford algorithm [17] is utilized to detect a negative weight cycle and increase the clock period T until all of the negative weight cycles are eliminated.

Using the circuit of the motivational example in Fig. 1, the constructed constraint graph is shown in Fig. 2(a). The solid lines represent setup time constraints, and the dashed lines represent hold time constraints. After applying the graph-based method, a minimum clock period of 10 units (similar to LP result) is computed with the set of clock arrival times as: $t_1 = 1$, $t_2 = 7$, $t_3 = 10$ and $t_{host} = 7$. As depicted in Fig. 2(b), there is no negative weight cycle after substituting clock period with 10 units.

2.2 Clock Skew Scheduling with Clock Gating

Clock gating is a popular technique to save dynamic power by deactivating the clock signal of the idle registers [14, 18]. Typically, an integrated clock gating (ICG) cell, as shown in Fig. 3, is utilized to prevent the clock signal from switching. The enable pin within an ICG cell creates a clock enable (or control) path in addition to the data paths. Thus, a clock enable (or control) path refers to the combinational logic from the output pin of a register to the enable pin of an ICG cell.

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 reg-

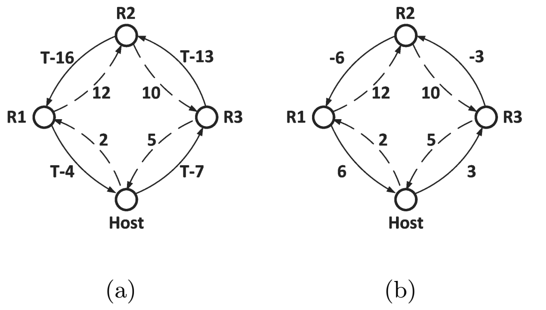


Figure 2: Constraint graph based formulation of skew scheduling for the circuit shown in Fig. 1: (a) constraint graph, (b) after applying a clock period of 10 units eliminating all of the negative weight cycles.

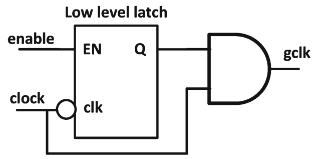


Figure 3: Integrated clock gating (ICG) cell.

isters that are gated by this ICG cell. A *clock propagation path* on the local clock tree is 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.

A simplified motivational example with clock gating is shown in Fig. 4 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 sub-tree 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 clock skew scheduling methodologies cannot consider the unique challenges introduced by clock gating. In [15], 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 [15]. 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. 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. 4). A method to perform clock skew scheduling in clock gated design with a local sub-tree is proposed in this paper, as described in the following section.

Table 2: LP based approach to clock skew scheduling in a clock gated design.

LP based approach for ICs with clock gating	
Objective: min T	
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	$0 \leq t_i, t_{icg,j} \leq T$

3. PROPOSED APPROACH

Since ICG cell has a clock pin, in the proposed approach, each ICG cell is treated as a register with an associated clock arrival time. Since there is a local clock tree between an ICG cell and registers gated by this ICG, the associated clock propagation delays can be treated as clock skew. However, note that the clock signal should arrive to the ICG cell earlier than it arrives to the registers 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. The linear programming based solution to skew scheduling in gated clock trees is described in Section 3.1 whereas the constrained graph based approach is discussed in Section 3.2.

3.1 Linear Programming Based Solution

The arrival time of a clock signal to a register gated by an ICG cell is larger than the arrival time of the clock signal to the ICG cell (see Fig. 4). 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,

$$t_{icg,j} - t_i \leq -CP_{min}, \tag{3}$$

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,

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

where CP_{max} is the maximum delay of the corresponding clock propagation path. Combining the constraints in (3) and (4) 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 2. 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 (control) 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. 4 is listed in Table 3. The program determines the minimum clock period as 22 units and a set of clock arrival times as $t_1 = 0$, $t_2 = 1$, $t_3 = 2$, $t_{icg,1} = 0$, and $t_{host} = 0$.

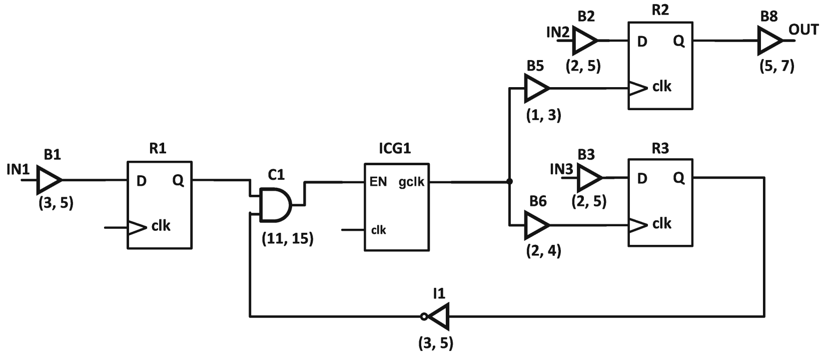


Figure 4: Simple sequential circuit consisting of an ICG cell, two registers gated by this ICG cell, a local clock sub-tree, and a timing loop formed by clock propagation path and clock enable path.

Table 3: Application of the LP based approach to circuit shown in Fig. 4.

LP based approach for ICs with clock gating	
Objective: min T	
s.t.	$-3 \leq t_{host} - t_1 \leq T - 5$
	$-2 \leq t_{host} - t_2 \leq T - 5$
	$-2 \leq t_{host} - t_3 \leq T - 5$
	$-5 \leq t_2 - t_{host} \leq T - 7$
	$-3 \leq t_{icg,1} - t_2 \leq -1$
	$-4 \leq t_{icg,1} - t_3 \leq -2$
	$-11 \leq t_1 - t_{icg,1} \leq T - 15$
	$-14 \leq t_3 - t_{icg,1} \leq T - 20$
	$0 \leq t_1, t_2, t_3, t_{icg,1}, t_{host} \leq T$

3.2 Constraint Graph Based Solution

In addition to linear programming, constraint graph based solution is also proposed to compare the efficacy and confirm the accuracy of the proposed methods. Each ICG cell is treated as a register and added to the directed graph as a vertex. The maximum and minimum clock propagation path delays are treated, respectively, as setup and hold time constraints of a traditional data path. Specifically, (3) is treated as a setup time constraint and modeled by a directed edge (R_i, ICG_j) with weight $-CP_{min}$. Similarly, (4) is treated as a hold time constraint and modeled by a directed edge (ICG_j, R_i) with weight CP_{max}.

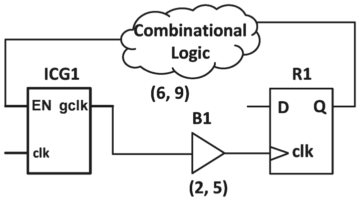
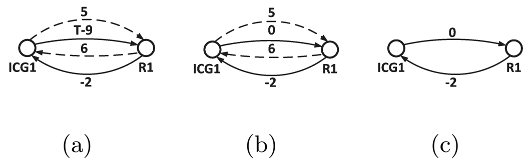


Figure 5: Simple example to illustrate the timing loop formed by an ICG cell and a register gated by this ICG cell.

An important issue in graph based solution of skew scheduling in gated clock networks is a possible timing loop that can form between an ICG cell and one of the registers gated by this ICG cell. Assume that the enable signal of the ICG cell is provided from the output pin of one of the registers that is gated by the same ICG cell (such as $ICG1$ and $R3$ in Fig. 4), then the ICG cell and the register form a loop. Unlike conventional data paths, the clock signal should arrive to the register *later* than it arrives to the ICG. Thus, this timing loop should be broken from the directed graph while still maintaining accurate results. As observed from experimental results on ISCAS'89 benchmark circuits, breaking the loop is necessary to obtain a feasible skew schedule.



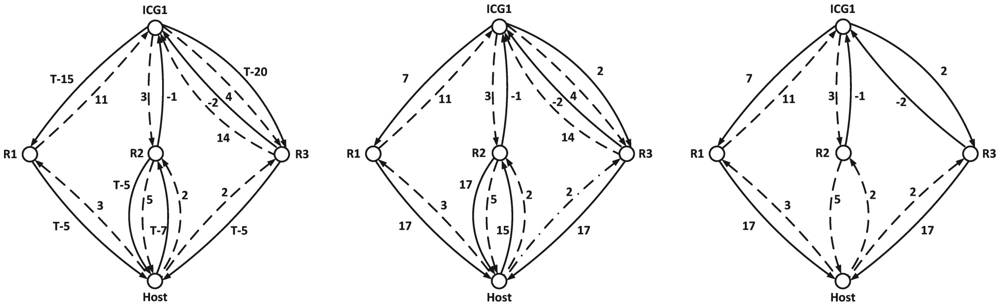
(a) (b) (c)

Figure 6: Constraint graph of the circuit shown in Fig. 5: (a) original graph, (b) after one iteration with clock period as 11 units, (c) after breaking the timing loop.

Table 4: Graph based solution for ICs with clock gating, including the proposed mechanism to break the timing loop.

```
Graph based approach (timing data)
1:   start with a clock period T
2:   for each edge(u,v) with weight w
3:       if (u,v) ∃ in G(V,E)
4:           if weight(u,v)>w
5:               weight(u,v)=w
6:       else
7:           add edge(u,v)
8:   end for
9:   add a source node
10:  for each V ∈ G(V,E) except source node
11:      add edge(source,V) with weight T
12:  end for
13:  apply Bellman-Ford algorithm on G(V,E)
14:  if ∃ negative weight cycle
15:      increase clock period
16:      repeat Line 1-13
17:  else
18:      return clock period T and skew schedule
```

To better describe this issue, consider the example shown in Fig. 5 where the enable signal of $ICG1$ is generated by the output signal of $R1$, forming a timing loop. The constraint graph of this circuit is depicted in Fig. 6. Due to the loop, there are two sets of max and min delay constraints: 1) $t_{icg,1} - t_1 \leq -2$, $t_{icg,1} - t_1 \geq -5$ and 2) $t_1 - t_{icg,1} \leq T - 9$, $t_1 -$



(a) (b) (c)

Figure 7: Constraint graph of the circuit shown in Fig. 4: (a) original graph, (b) after one iteration with clock period as 22 units, (c) after breaking the timing loop.

$t_{icg,1} \geq -6$, as shown in Fig. 6(a). To break the loop, only the tighter constraints of the same directed edge (*i.e.* smaller weight) should be preserved. For example, assume that in one of the iterations, clock period T is determined as 9 units, producing the following inequalities: $-5 \leq t_{icg,1} - t_1 \leq -2$ and $0 \leq t_{icg,1} - t_1 \leq 6$, as shown in Fig. 6(b). Since only the tighter constraint of the same edge should be preserved, the edges with weights 5 and 6 are dropped, breaking the loop, as shown in Fig. 6(c). According to Fig. 6(c), a negative weight cycle exists, indicating that the chosen clock period should be increased. If the process is repeated with a clock period of 11 units, the cycle weight becomes zero, indicating that the minimum clock period has been determined while satisfying the timing constraints.

The pseudo-code of the proposed constraint graph based solution is provided in Table 4. The algorithm takes the timing data as the input and generates a constraint graph in lines 2 to 12. In lines 3 to 5, the timing loops formed by ICG cells and registers gated by the same ICG cells are detected and broken by the proposed method (*i.e.*, preserving only the smaller weight of the same directed edges). In line 13, Bellman-Ford algorithm [17] is utilized to detect negative weight cycles. If found, clock period is increased until all of the negative weight cycles are removed. In line 18, the algorithm returns the minimum clock period and the skew schedule, *i.e.*, clock arrival time to each register and ICG cell.

As an example, the proposed algorithm is applied to the circuit shown in Fig. 4. The original constraint graph that corresponds to this circuit is depicted in Fig. 7(a). T is replaced with the minimum clock period 22 units, producing the graph shown in Fig. 7(b). The timing loop formed by $ICG1$ and $D3$ are broken using the proposed method, producing the final graph shown in Fig. 7(c). The algorithm returns the clock arrival times as $t_1 = 22$, $t_2 = 22$, $t_3 = 22$, $t_{icg} = 20$, and $t_{host} = 22$.

4. EXPERIMENTAL RESULTS

The proposed LP based and constraint graph based approaches for skew scheduling in gated clock networks are evaluated using the largest ISCAS'89 benchmark circuits consisting of up to approximately 2000 registers. Each benchmark is synthesized with Synopsys Design Compiler [19] using the 45 nm NanGate open cell library [20]. ICG cells are inserted by the tool during the synthesis stage. An open source GLPK (GNU Linear Programming Kit) [21] is used as the linear programming solver, running on a Linux system with Intel Xeon processor.

The experimental results are listed in Table 5 for both linear programming and graph based solutions. It is important to note that both solutions provide the same minimum clock period in each circuit, verifying the accuracy of the algorithms. The maximum reduction in clock period after skew scheduling is approximately 21%, which highly depends upon the timing data. In some benchmarks, higher gating percentage corresponds to less reduction in clock period, such as S1423 and S38417. However, this behavior does not hold in other benchmarks such as S38584 where 16% reduction in clock period is achieved with approximately 72% gating. It is also shown in Table 5 that the graph based solution produces smaller global skew than LP based solution.

The run time of both solutions is compared in Fig. 8 for some of the benchmark circuits. LP based solution runs faster than or equal to graph based solution. Note that the graph based approach utilizes Bellman-Ford algorithm with a computational complexity of $O(V \cdot E)$ [17], where V is the overall number of registers and ICG cells in the circuit, and E is the overall number of data paths, enable paths and clock propagation paths. Lines 2 to 8 in Table 4 have a complexity of $O(E)$ and lines 9 to 11 have a complexity of $O(V)$. Therefore, the computational complexity of the graph based method is maintained at $O(V \cdot E)$. The LP based solution utilizes the simplex algorithm and in practice, runs faster. However, note that with certain inputs, simplex algorithm may require exponential time to reach a solution [17].

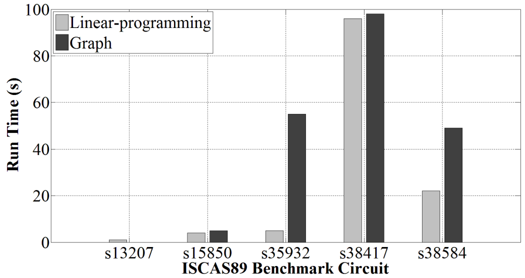


Figure 8: The run time comparison of linear programming and graph based approaches.

Table 5: Experimental results demonstrating the reduction in clock period of gated ISCAS'89 benchmark circuits after clock skew scheduling (CCS).

Circuit	No. of DFFs	No. of ICGs	Gating%	Clock Period (ns)			Max Global Skew (ns)	
				Zero Skew	After CSS	Reduction	LP	Graph
S1423	74	30	90.54%	4.9212	4.6618	5.27%	0.2594	0.2374
S9234	125	15	51.20%	2.9278	2.7369	6.52%	0.7171	0.1909
S13207	240	37	44.17%	2.4436	2.0678	15.38%	0.3897	0.3896
S15850	434	57	66.36%	4.9466	4.1436	16.23%	0.8030	0.1342
S35932	1728	4	0.23%	3.8314	3.0419	20.61%	0.7895	0.0773
S38417	1459	236	49.62%	4.8895	4.6490	4.92%	0.2926	0.2487
S38584	1240	251	71.61%	4.5806	3.8425	16.11%	0.7381	0.0570

5. CONCLUSIONS

Existing clock skew scheduling methods cannot effectively consider clock gating where an ICG cell gates multiple registers. In such cases, a local clock tree typcially exists between the ICG cell and registers gated by this ICG cell, introducing additional and unbalanced *clock propagation paths*. A methodology is proposed in this paper to efficiently implement skew scheduling for gated clock networks. Each ICG cell is treated as a register and additional constraints are included to accurately consider clock propagation paths. A mechanism is also proposed to break the register-to-ICG timing loops in the graph based solution to ensure accuracy. The proposed algorithms are evaluated using the largest ISCAS'89 benchmark circuits with gated clock networks. A conventional design flow is utilized, demonstrating that the proposed algorithms are feasible for automation. Up to 21% reduction in clock period is demonstrated.

6. ACKNOWLEDGMENTS

This research is supported by Semiconductor Research Corporation (SRC) under contract No. 2013-TJ-2449 and 2013-TJ-2450.

7. REFERENCES

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

[2] E. G. Friedman, "Clock Distribution Networks in Synchronous Digital Integrated Circuits," *Proceedings of the IEEE*, Vol. 89, No. 5, pp. 665-692, May 2001.

[3] I. S. Kourtev, B. Taskin, and E. G. Friedman, *Timing Optimization Through Clock Skew Scheduling.* Springer, 2009.

[4] J. Neves and E. G. Friedman, "Optimal clock skew scheduling tolerant to process variations," *Design Automation Conference*, pp. 623-628, June 1996.

[5] J. P. Fishburn, "Clock Skew Optimization," *IEEE Transactions on Computers*, Vol. 39, No. 7, pp. 945-951, July 1990.

[6] T.G.Szymanski, "Computing Optimal Clock Schedules," *ACM/IEEE Design Automation Conference*, pp. 399-404, June 1992.

[7] N.Shenoy, R.K.Brayton, and A.L.Sangiovanni-Vincentelli, "Graph Algorithms for Clock Skew Optimization," *International Conference on Computer-Aided Design*, pp. 132-136, November 1992.

[8] R. Deokar and S. Sapatnekar, "A Graph-theoretic Approach to Clock Skew Optimization," *Int. Symp. on Circuits and Systems*, pp. 407-410, May 1994.

[9] B. Taskin and I. S. Kourtev, "Delay Insertion Method in Clock Skew Scheduling," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 25, No. 4, pp. 651-663, April 2006.

[10] S.-H. Huang, C.-H. Cheng, C.-M. Chang, and Y.-T. Nieh, "Clock Period Minimization with Minimum Delay Insertion," *Design Automation Conference*, pp. 970-975, June 2007.

[11] K. Ravindran, A. Kuehlmann, and E. Sentovich, "Multi-domain Clock Skew Scheduling," *International Conference on Computer-Aided Design*, pp. 801-808, November 2003.

[12] M. Ni and S. O. Memik, "A Fast Heuristic Algorithm for Multidomain Clock Skew Scheduling," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 18, No. 4, pp. 630-637, April 2010.

[13] L. Li, Y. Lu, and H. Zhou, "Optimal and Efficient Algorithms for Multidomain Clock Skew Scheduling," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, Vol. 22, No. 9, pp. 1888-1897, Sept. 2014.

[14] Q. Wu, M. Pedram, and X. Wu, "Clock-gating and Its Application to Low Power Design of Sequential Circuits," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, Vol. 47, No. 103, pp. 415-420, March 2000.

[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," *Design, Automation And Test in Europe Conference And Exhibition (DATE)*, pp. 1831-1836, March 2013.

[16] E. Salman, A. Dasdan, F.Taraporevala, K. Kucukcakar and E. G. Friedman, "Exploiting Setup-Hold Time Interdependence in Static Timing Analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 26, No. 6, pp. 1114-1125, June 2007.

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms.* The MIT Press, 2001.

[18] M. Donno, A. Ivaldi, L. Benini, and E. Macii, "Clock-tree Power Optimization based on RTL Clock-gating," *Design Automation Conference*, pp. 622-627, June 2003.

[19] Synopsys. Design Compiler. http://www.synopsys.com/home.aspx.

[20] NanGate. 45nm Open Cell Library. http://www.nangate.com.

[21] GNU. GNU Linear Programming Kit. https://www.gnu.org/software/glpk/.