

Slew-Driven Clock Tree Synthesis (SLECTS) Methodology to Facilitate Low Voltage Clocking

Weicheng Liu*, Can Sitik†, Emre Salman*, 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

‡NXP Semiconductor Inc, Austin, TX 78735

Abstract—A slew-driven clock tree synthesis methodology, referred to as SLECTS, is proposed for clock networks operating at reduced voltages. SLECTS is developed to i) satisfy tight slew constraints, which might be highly challenging with skew-driven methodologies, particularly at scaled voltages, and ii) reduce the power dissipation of the clock tree, thanks to targeting slew and skew constraints methodically. SLECTS achieves up to 17% power savings compared to a traditional skew-driven methodology, at 3 GHz operation in a 20nm FinFET technology. Furthermore, SLECTS decreases the total buffer size of a large industrial circuit in 16nm FinFET technology by 59%, compared to an industrial vendor tool at similar clock skew and slew constraints.

I. INTRODUCTION

A well-known approach to minimize the overall on-chip power dissipation is to reduce the supply voltage [1]. This approach (low voltage/swing operation) has been extended to clock networks due to high clock net capacitance [2]–[4]. Recently, a clock tree synthesis algorithm and a flip-flop have been proposed for low voltage clocking that target high performance [5]–[7].

One of the primary challenges in low voltage clocking is the difficulty in satisfying slew constraints due to degraded drive ability. Slew-constrained design techniques are proposed in recent work [8], [9] to fix (or avoid) timing violations due to slew. Exploiting slew-awareness as part of the clock tree synthesis (i.e. slew-driven) has not been previously addressed.

The major contribution of this work is the introduction of a slew-driven CTS methodology call (SLECTS), as conceptually illustrated in Fig. 1. Instead of targeting skew minimization as the primary objective and resolving slew violations with buffer insertion, as in traditional skew-driven CTS, SLECTS targets slew optimization at every stage of the synthesis. In contrast to skew-driven CTS resolving slew violations in post-CTS optimization, SLECTS uses buffering more efficiently to constrain skew and slew simultaneously. Due to this efficient slew handling and efficient use of buffering, SLECTS leads to reduced power dissipation while satisfying the slew and skew constraints. SLECTS is developed on the popular deferred merge embedding (DME) algorithm, and features innovations of 1) a new cost metric for the merging process, 2) a new merging point computation method, 3) a new net splitting method.

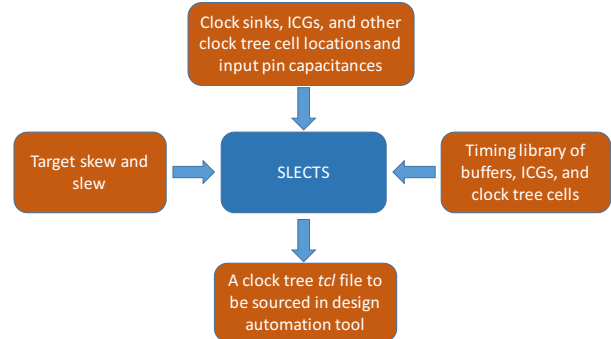


Fig. 1. The input and output of the proposed methodology SLECTS.

According to experimental results, at 2 GHz operation, the final power savings compared to a (traditional, skew-driven) DME implementation satisfying the same skew and slew constraints are 9% and 10% at the available nominal (V_{dd}) and the low voltage ($0.7 \times V_{dd}$) levels in a 20nm technology [10], respectively. These savings improve to 17% for both nominal (V_{dd}) and the low voltage ($0.7 \times V_{dd}$) operations at 3 GHz. The increased savings of 17% (up from 10%) at 3 GHz operation highlight the slew-driven approach of the proposed methodology in performing better in tighter slew constraints (at a higher frequency).

As an additional study, SLECTS is tested on an industrial circuit, which has approximately 1M gates and 75k flip-flop sinks, operating at 1.33 GHz in a 16nm FinFET technology. In this case study, it is shown that SLECTS satisfies tight slew constraints that an industrial vendor tool cannot satisfy at the expense of a 7% increase in total buffer size. Furthermore, SLECTS reduces the total buffer size by 59% compared to the industrial vendor tool at similar slew and skew constraints. These results highlight the applicability of SLECTS as a better suited tool for modern ICs.

II. DME AND PROPOSED NOVELTIES

DME method is a popular technique for clock tree synthesis. Fig. 2 shows a flow chart of DME procedure. The proposed methodology SLECTS is developed within the DME framework. The novelties in SLECTS are highlighted within Step 1, Step 2, and Step 3:

- 1) A cost metric definition for efficient clustering,
- 2) A slew and skew-aware merging point computation,
- 3) A slew and insertion delay-aware net splitting.

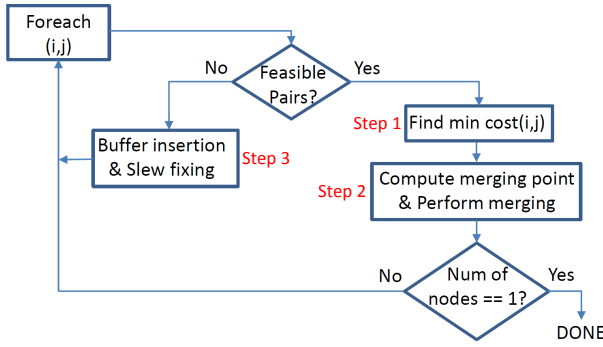


Fig. 2. The flow chart of the DME framework.

Three steps are presented in Sections II-A, II-B and II-C, respectively.

A. Step 1: Merging Pair Selection

As the DME algorithm searches for the minimum cost pair among all pairs, several pair selection techniques and cost definitions are introduced in the literature, which are classified into 2 groups: 1) distance-based [11], and 2) delay-based [12]. Distance-based merging pair selection suffers from the well-known deficiencies of using length as a delay metric. As the pairs are not selected one at a time, this selection results in a sub-optimal clustering. The delay-based approach, is expectedly higher in accuracy in terms of satisfying skew.

The contemporary and common delay-based approach has two drawbacks making it formidable for SLECTS: 1) physically farther nodes can be selected to minimize skew, which is detrimental to slew, 2) considering wire snaking as part of cost metric is inaccurate. Consequently, in this paper, the distance-based approach (similar to [11]) is selected as the cost metric. It is important to note here that using a distance-based cost results in several subtree clusters that have different capacitance and delay values, which would make merging harder at the top-level of a clock tree. However, the potential effects of these mismatches are fixed by buffer insertion and/or wire snaking, and the power overhead of these processes are shown, experimentally, to be less than the slew fixing in traditional skew-driven CTS algorithms.

B. Step 2: Merging Point Computation

After selecting the minimum cost pair, as described in Section II-A, the merging point is determined to perform routing of this pair. In this paper, the skew constraint-based merging regions are constructed in the bottom-up phase, similar to the BST-DME methodology [13]. Unlike BST-DME methodology two phases, SLECTS determines the merging point in the same phase while considering the slew and skew constraint simultaneously. In the proposed merging point computation algorithm, for each pair $i-j$ that is to be merged, each end point of the permissible merging window represents a corner case when the skew within $i-j$ pair is equal to skew constraint.

After the permissible merging window is generated, the *minimum slew point* is computed. The minimum slew point is defined as the point that makes the slew at node i and j equal in order to obtain the minimum slew at both nodes. In

order to estimate this point, the PERI model [14] is used for slew propagation, which estimates the slew degradation $S(W)$ on a wire segment W as:

$$S(W) = \ln(9) \times ED(W) \quad (1)$$

where $ED(W)$ is the Elmore delay [15] of the wire segment W , and estimates the output slew $S_{out}(W)$ of a wire segment W as:

$$S_{out}(W) = \sqrt{S_{in}(W)^2 + S(W)^2} \quad (2)$$

where $S_{in}(W)$ is the input slew of the wire segment. Using Eq. (1) and Eq. (2), the minimum slew point m should satisfy the following equation:

$$S_i^2 - (\ln(9) \times ED(m, i))^2 = S_j^2 - (\ln(9) \times ED(m, j))^2 \quad (3)$$

where S_i and S_j are the target slew values at nodes i and j , respectively. Slews are propagated bottom-up to the internal nodes after each merging. After solving equation 3, the position of point m is checked to identify whether it is within the permissible merging window. If this is the case, m is set as the merging point k . Otherwise, k is set as one of the corner points. For the cases where permissible merging window does not exist, buffer insertion or wire snaking is considered.

C. Step 3: Slew-Aware Net Splitting

Traditional DME-based CTS algorithms consider buffer insertion at the merging points only, and do not consider splitting the net after selecting merging pairs. This would result in slew violations on long distance nets and would not permit the desired voltage and frequency scaling. Thus, the contemporary approach is to synthesize clock tree with slew violations and fix these violations later in the physical design flow, as a post-CTS optimization.

SLECTS satisfies slew constraints while considering the insertion delays of the nodes to be merged. The insertion delay-aware net splitting technique is proposed. It first finds the minimum cost pair (s_i and s_j) and determines which node of the selected pair has a smaller insertion delay. Then, the distance is computed from this lower insertion delay node to generate a new node m .

In the proposed approach, the splitting point is determined as the longest feasible distance from the selected node. The longest feasible distance is computed using the slew constraint, the timing models of buffer and the interconnect metrics. The output slew $S(B)$ of a buffer B is estimated in [16] as:

$$S(B) = K_{cap}^{slew} \times C_{out} + K_{slew} \quad (4)$$

where K_{cap}^{slew} is the capacitance coefficient of output slew, C_{out} is the output capacitance of the buffer B and K_{slew} is the no-load slew of the buffer. The slew propagation on the wire segment is estimated using Eq. (1) and Eq. (2). Combining Eq. (4), Eq. (1) and Eq. (2), the maximum distance L that a net can be split from a node i should satisfy the following equation:

$$(Slew_{const})^2 = (K_{cap}^{slew} \times (L \times C_{unit} + Cap_i))^2 + (\ln(9) \times ED(W))^2 \quad (5)$$

TABLE I

CLOCK POWER COMPARISONS OF DME [12] AND SLECTS SCHEMES OPERATING AT 2 GHz, REPORTED FOR BOTH NOMINAL VOLTAGE ($0.9 \times V_{dd}$) AND LOW VOLTAGE ($0.63 \times V_{dd}$) AT THE WORST CASE CORNER OF 20NM FINFET TECHNOLOGY.

Circuits	Power (mW) at $0.9 \times V_{dd}$		Power (mW) at $0.63 \times V_{dd}$	
	DME	SLECTS	DME	SLECTS
cns03	8.8	8.1	4.5	4.0
cns04	7.8	6.9	3.8	3.3
cns05	3.4	3.0	1.6	1.5
cns06	6.1	5.9	3.0	2.9
cns07	9.7	8.7	4.7	4.3
cns08	6.9	6.2	3.5	3.1
Norm.	1.00	0.91	1.00	0.90

TABLE II

CLOCK POWER COMPARISONS OF DME [12] AND SLECTS SCHEMES OPERATING AT 3 GHz, REPORTED FOR BOTH NOMINAL VOLTAGE ($0.9 \times V_{dd}$) AND LOW VOLTAGE ($0.72 \times V_{dd}$) AT THE WORST CASE CORNER OF 20NM FINFET TECHNOLOGY.

Circuits	Power (mW) at $0.9 \times V_{dd}$		Power (mW) at $0.72 \times V_{dd}$	
	DME	SLECTS	DME	SLECTS
cns03	20.0	15.5	11.0	9.2
cns04	15.4	13.2	9.1	7.7
cns05	7.5	5.9	4.1	3.5
cns06	11.9	11.1	7.1	6.6
cns07	21.3	16.6	12.4	10.0
cns08	13.3	11.8	9.5	7.0
Norm.	1.00	0.83	1.00	0.83

where C_{unit} is the per-unit capacitance of the wire. In this paper, the largest size buffer in the library is used in order to split as large distance as possible in one iteration.

III. EXPERIMENTAL RESULTS

The proposed methodology is implemented with *Perl* and the quality of results is presented with select ISPD'10 benchmarks. In order to investigate the performance of SLECTS against the previous skew-driven methodologies, the power and timing measurements of SLECTS are compared against DME [12] at 20nm FinFET technology [10], operating at 2 GHz and 3 GHz. The skew constraint is set to 50ps, and the slew constraint is set to 10% of the clock period for each frequency. Two voltage levels are considered at each frequency: 1) $0.9 \times V_{dd}$ of this technology (0.9V in the nominal case), 2) Low V_{dd} that is achievable by all benchmarks, which is $0.63 \times V_{dd}$ at 2 GHz and $0.72 \times V_{dd}$ at 3 GHz. All experiments are performed at the worst case corner ($0.9 \times V_{dd}$, SS, -40°C) to identify the lower bound of improvements achieved by SLECTS.

The comparative results are presented in Table I at 2 GHz operation for $0.9 \times V_{dd}$ and $0.63 \times V_{dd}$. The power savings of SLECTS compared to DME [12] are 9% and 10% at $0.9 \times V_{dd}$ and $0.63 \times V_{dd}$, respectively. When the slew constraints are tighter at 3 GHz operation, the power savings of SLECTS are 17% for both $0.9 \times V_{dd}$ and $0.72 \times V_{dd}$, as shown in Table II. This increase in power savings validates the slew-driven approach of SLECTS: The power savings improve when the challenge of slew handling is harder at tighter slew constraints (33ps at 3 GHz vs. 50ps at 2 GHz). Furthermore, it shows the applicability of SLECTS to future nodes, as interconnect resistance is predicted to be higher, and the supply

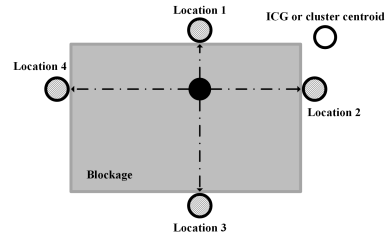


Fig. 3. Buffer movement in block area

voltage (V_{dd}) levels are predicted to be lower, both of which degrades slew.

IV. CASE STUDY: SLECTS IN A LARGE INDUSTRIAL DESIGN

In order to highlight the effectiveness of SLECTS, and compare it to a modern industrial vendor tool, a case study is presented with an industrial design implemented in a 16nm FinFET technology. It consists of approximately 1M gates, 75k flip-flop sinks, and 3k integrated clock gating (ICG) cells. Over 80% of flip-flops are clock-gated. Three different sizes of buffers (CLKBUFF3, CLKBUFF5 and CLKBUFF8) with different driving abilities are used to synthesize clock trees. Metal layers 8 and 9 are selected for clock routing which has a per unit resistance of $6.17\Omega/\mu\text{m}$ and a per unit capacitance of $0.2\text{fF}/\mu\text{m}$. The circuit is operated at 0.8V and 1.33 GHz.

In order to reduce the run time and memory footprint: First, local clock trees of gated flip-flops are synthesized. Then the well-known k-means clustering algorithm [17] is applied on the upper-level clock tree.

In the industrial design, there are macro blocks that generate placement blockages. If a buffer insertion is performed within a placement blockage, the placement legalization step after CTS moves this buffer significantly, potentially resulting in slew violations. To this end, a simple heuristic is implemented within the proposed methodology. In this heuristic, if a buffer insertion within a placement blockage is detected, it is moved to one of the four edges of this placement blockage, as shown in Figure 3. Among these four potential points, the point that is the closest to the ICG (source of a gated local tree) or the cluster centroid (at the upper-level of clock tree) is set as the new location of the buffer. For instance, in the specific case of Figure 3, the buffer is moved to location 2.

The floorplan of the clock tree after applying the proposed CTS methodology is shown in Figure 4. The large gray areas at bottom left and right corners are placement blockages. For comparison purposes, three clock trees are synthesized as follows: 1) using the vendor tool at 70ps target slew, 2) using SLECTS at 70ps target slew, and 3) using SLECTS at 130ps target slew (experimentally determined to achieve similar clock slew as the vendor tool that cannot satisfy 70ps slew constraint). In order to quantify the change in the number of buffers, a metric called *buffer cost* is defined as follows:

$$\text{Buffer cost} = \sum i \times N_i \quad (6)$$

where i is the buffer size, and N_i is the number of buffers with size i . The clock tree results for these three cases are

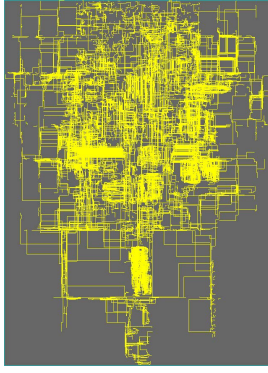


Fig. 4. Floorplan of clock tree output of SLECTS on a 16nm industrial design with approximately 1M gates and 75k sinks.

TABLE III
CTS COMPARISON BETWEEN THE VENDOR TOOL AND SLECTS AT DIFFERENT TARGET SLEW (TS) CONSTRAINTS.

Properties	Vendor tool (TS=70ps)	SLECTS	
		(TS=70ps)	(TS=130ps)
Max buf. slew	108.0ps	75.4ps	117.6ps
Max sink slew	111.0ps	71.6ps	114.5ps
Depth	18	17	14
Skew	≈200ps	≈200ps	≈200ps
CLKBUFF3	2064	553(-73%)	588(-72%)
CLKBUFF5	516	1564(+203%)	474(-8%)
CLKBUFF8	282	297(+5%)	46(-84%)
Buffer cost	11028	11855	4502
Run time	≈45mins	≈8mins	≈8mins
Norm. buffer cost	1.00	1.07	0.41
Norm. run time	1.00	0.18	0.18

presented in Table III. It is shown that the vendor tool has high slew violations (111.0ps) at 70ps target slew, whereas SLECTS has only small violations (due to the slight inaccuracy of the adopted timing model) at 75.4ps. SLECTS achieves these smaller slew values with a trade-off in the number of clock buffers, resulting in a 7% increase in buffer cost. At 130ps target slew, SLECTS achieves similar clock slew as the vendor tool at 70ps target slew (111.0ps for the vendor tool vs. 117.6ps for SLECTS) while reducing the number of clock buffers for each buffer type, as presented in Table III. This significant increase in the number of clock buffer results in 59% decrease in the buffer cost, indicating significant power savings.

The presented results demonstrate that SLECTS handles clock slew efficiently to satisfy tight slew constraints at the expense of a slight increase in buffer cost. Alternatively, at comparable clock slews, SLECTS achieves significant reduction in buffer cost. Note that the interconnect resistance of the clock routing layer in this experimental setup is $5.1\times$ of the one presented in Section III ($6.17\Omega/\mu\text{m}$ vs. $1.2\Omega/\mu\text{m}$), exacerbating the challenge of handling clock slew. Thus, the improved savings in this setup compared to the results presented in Table I and Table II validate the slew-driven claim of this paper, as SLECTS provides improved power savings when the challenge of handling clock slew is harder. Furthermore, the run time of SLECTS is significantly smaller ($0.18\times$) than the vendor tool.

V. CONCLUSION

A slew-driven CTS methodology is proposed in this paper. A new net splitting technique and merge point selection are introduced for power savings. The primary objective is to achieve significant power savings without degrading circuit performance. Furthermore, the SLECTS methodology has been verified on FinFET-based clock trees to achieve voltage scaling for low power or frequency scaling for performance, while providing additional power savings compared to existing methodologies. The proposed methodology can also be easily integrated into design automation tools, similar to traditional skew-driven CTS approaches.

REFERENCES

- [1] E. Salman and E. G. Friedman, *High Performance Integrated Circuit Design*. McGraw-Hill, 2012.
- [2] 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.
- [3] 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.
- [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 (JETC)*, vol. 12, no. 2, pp. 1–13, 2015.
- [5] 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.
- [6] 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.
- [7] 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*, in press.
- [8] Y. Cai, C. Deng, Q. Zhou, H. Yao, F. Niu, and C. Sze, "Obstacle-avoiding and slew-constrained clock tree synthesis with efficient buffer insertion," *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems*, vol. 23, no. 1, pp. 142–155, Jan. 2014.
- [9] J. Lu, W.-K. Chow, and C.-W. Sham, "Fast power- and slew-aware gated clock tree synthesis," *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems*, vol. 20, no. 11, pp. 2094–2103, Nov. 2012.
- [10] S. Sinha and et al., "Exploring sub-20nm finfet design with predictive technology models," in *Proceedings of ACM/IEEE Design Automation Conference (DAC)*, 2012, pp. 283–288.
- [11] M. Edahiro, "A clustering-based optimization algorithm in zero-skew routings," in *ACM/IEEE Design Automation Conference (DAC)*, June 1993, pp. 612–616.
- [12] R. Chaturvedi and J. Hu, "An efficient merging scheme for prescribed skew clock routing," *IEEE Transactions on Very Large Scale Integration (TVLSI) Systems*, vol. 13, no. 6, pp. 750–754, Jun 2005.
- [13] J. Cong, A. B. Kahng, C.-K. Koh, and C.-W. A. Tsao, "Bounded-skew clock and steiner routing," *ACM Trans. Des. Autom. Electron. Syst. (TODAES)*, vol. 3, no. 3, pp. 341–388, Jul. 1998.
- [14] C. Kashyap, C. Alpert, F. Liu, and A. Devgan, "Closed-form expressions for extending step delay and slew metrics to ramp inputs for RC trees," *IEEE Transactions on Computer-Aided Design (TCAD) of Integrated Circuits and Systems*, vol. 23, no. 4, pp. 509–516, April 2004.
- [15] W. C. Elmore, "The transient response of damped linear networks with particular regard to wideband amplifiers," *Journal of Applied Physics*, vol. 19, no. 1, pp. 55–63, 1948.
- [16] C. Sitik, S. Lerner, and B. Taskin, "Timing characterization of clock buffers for clock tree synthesis," in *Proceedings of the IEEE International Conference on Computer Design (ICCD)*, Oct 2014, pp. 230–236.
- [17] J. Hartigan and M. Wong, "A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C*, vol. 28, no. 1, pp. 100–108, 1979.