

# A Survey on Buffered Clock Tree Synthesis for Skew Optimization

Anju Rose Tom<sup>1</sup>, K. Gnana Sheela<sup>2</sup>

<sup>1,2</sup>Electronics and Communication Department, Toc H Institute of Science and Technology, Kerala, India

**Abstract:** Buffered clock tree synthesis has become increasingly critical in an attempt to generate a high performance synchronous chip design. Skew optimization includes the satisfaction of slew constraints and signal polarity. Clock tree approach features the clock tree construction stage with the obstacle aware topology generation algorithm, balanced insertion of candidate buffer positions and a fast heuristic buffer insertion algorithm. With an overall view on obstacles to explore the global optimization space, CTS approach effectively overcomes the negative influence on skew which is brought by the obstacles. A look up table was built through NGSPICE simulation to achieve accurate buffer delay and slew which guarantees overall skew optimization. The accuracy of look up table is demonstrated through huge skew reduction. Additionally, wire length of clock routing trees should be minimized in order to reduce system power requirements and deformation of the clock pulses at the synchronizing elements of the system.

**Keywords:** Clock tree synthesis; Buffer insertion; Skew optimization; Obstacle avoidance.

## 1. Introduction

Clock tree synthesis is an important element and problem in physical design which controls the pace of the whole circuit. As VLSI technology moves into the nanometer territory along with feature shrinking, buffer insertion becomes unavoidable in the CTS flow to reduce delay and keep the signal integrity. Clock tree with low skew, accurate delay and slew information is essential during this approach.

Elmore delay is the first moment of the impulse response and it is widely used model to estimate the interconnect delay in CTS. Despite that Elmore delay model is simple and efficient to compute, the accuracy of the Elmore delay is limited. It indicates that the buffer intrinsic delay is sensitive to input slew and can vary significantly when the input slew changes. However, Elmore delay is not sufficient enough to get the slew information within RC net list. During the clock routing, a track graph is constructed in order to guarantee the obstacle avoidance. Except for lacking a good buffer insertion strategy, the final skew result is limited by the inaccuracy of the Elmore delay model. The NGSPICE simulation is performed immediately after buffer is inserted into the clock tree to check whether the slew constraints are violated. It guarantees better runtime and skew results. [1]

The algorithm obtains notable optimization in skew but the resource usage will be a problem in large scale benchmarks with the obstacles because the algorithm will realize the tree with all paths equal to the longest path. It improves the symmetric tree structure by adopting a hybrid structure in which symmetrical top tree drives bottom level asymmetry trees. The bottom level asymmetry trees prevent the explosive wire length. The algorithm is based on D-C plane. The paper highlights the delay model and the look up table providing accurate delay or slew during buffer insertion. The position of buffer only depends on slew limit which can be optimized. In addition, it may cost a lot of routing resources on balancing the branches since the buffers are not only inserted when the slew on the wire is about to violate the constraint. Buffer positions are relatively flexible on a wire as there are redundant candidate buffers (CBF) to keep the

signal integrity so that the actual buffer insertion positions give consideration to the delay balance[1].

In this stage, a heuristic topology generation algorithm named OBB is proposed for obstacle avoidance. It predicts the obstacles impaction on real routing and attempts to raise the tree level impacted by obstacles. Path difference at upper levels is generally easier to deal with than that at lower levels. This is because the unbalanced delays induced by upper level wire differences involve fewer wires and thus would be easier to repair. The merge nodes have no overlaps with the obstacles but the routing method which directly connects two nodes that may generate wire crossing the obstacles. The combination of the selected solutions from the parts of sampling technique makes the final solution set of node.

The survey paper is organized as follows. Section 2 deals with the literature survey which includes the detailed explanation of various algorithms for efficient buffer insertion. Section 3 describes the comparative analysis study of the obstacle aware topology generation algorithms. Finally, last section concludes with the best approach towards the clock tree synthesis without any obstacles.

## 2. Literature Survey

Ginneken et al (1990) formulated an algorithm for choosing the buffer positions for a wiring tree such that the Elmore delay is minimum. For given required arrival times at the sinks of the wiring tree, the algorithm chooses the buffer such that the required departure time at the source is as late as possible. The topology of the wiring tree, a Steiner tree is assumed given as well as the possible i.e. legal positions of the buffers. The algorithm uses depth first search on the wiring tree to construct a set of time or capacitance pairs that corresponds to different choices. The complexity of the algorithm is  $O(B*B)$  where B is the number of possible buffer positions. An extension of the basic algorithm allows the minimization of the number of buffers as a secondary objective. The tree structure with buffers which distributes the signal over the chip is called fan out trees. An algorithm

is thus designed for fan out trees. However, this algorithm does not take physical information such as capacitance and RC effects of the wire into account[1].

Rapidly increasing design complexity due to small size and higher speed, results in the problem of clock skew and insertion delay. These are the two important parameters which should be considered for successful completion of the design. In this work, a method for minimizing clock skew by buffer insertion and resize is proposed. Clock skew will be minimized during post-CTS timing analysis after placement of standard cells during physical implementation of the design. Also, buffer relocation method is used for minimizing the delay of the cells. The growth of the delay with the length of a wire can be reduced to linear by introducing buffers at fixed distances. Most connections have a tree structure with multiple sinks. The special properties of the Elmore delay model allow the usage of a hierarchical algorithm. The algorithm follows the general two phase bottom up prediction and top down decision approach. The complexity of the algorithm is only linear in the number of possible positions of the buffers. A simple extension of the algorithm allows optimization of the cost of the buffers used [2].

Chao et al (1992) developed the deferred merge embedding (DME) algorithm which embeds any given connection topology to create a clock tree with zero skew while minimizing total wire length. The algorithm always yields exact zero skew trees with respect to the appropriate delay model. The DME algorithm may be applied to either the Elmore or linear delay model, and yields optimal total wire length for linear delay. DME is a very fast algorithm, running in time linear in the number of synchronizing elements. A unified buffer balance algorithm and deferred merge algorithm constructs a clock tree topology using a top-down balanced bipartition (BB) approach and then applies DME to that topology is also presented [3]. In linear time, it embeds any given connection topology into the Manhattan plane to create a clock tree with zero skew while minimizing total wire length. The DME algorithm may be applied to either the Elmore or the linear delay model and yields optimal total wire length for linear delay.

In the design of high performance VLSI systems, minimization of clock skew is an increasingly important objective. Additionally, wire length of clock routing trees should be minimized in order to reduce system power requirements and deformation of the clock pulse at the synchronizing elements of the system. Circuit speed is increasingly limited by clock skew which is the maximum difference in arrival times of the clocking signal. Circuit speed is limited mainly by delay on the longest path through combinational logic and clock skew [4].

Lillis et al (1996) presented efficient and optimal algorithms for timing optimization by discrete wire sizing and buffer insertion. It was able to minimize cost function which is subjected to given timing constraints and focuses on the minimization of dynamic power dissipation. This algorithm is easily adaptable to area minimization. In addition, the algorithm efficiently computes the complete optimal power delay trade off curve for added design flexibility. An

extension of the basic algorithm accommodates a generalized delay model which takes into account the effect of signal slew on buffer delay which will contribute towards the overall delay. Automatic sizing of wire widths is an attractive technique for timing optimization in signal nets particularly with the advent of submicron technology. The benefit of wire sizing lies in shrinking geometries wire resistance is now a significant contribution towards overall delay. As a result, it makes sense to tune the widths of the wires to balance the tradeoff between added capacitance and decreased resistance[5].

The improved Elmore delay formulation and the traditional Elmore delay model are compared according to SPICE simulation environment performances which verify the superior accuracy of the novel delay formulation. Interconnects are now considered as the bottleneck in the design of system-on-chip (SoC) since they introduce delay and power consumption. To deal with this issue, data coding for interconnect power and timing optimization has been introduced. In today's SoCs these techniques are not efficient anymore due to their codec complexity or to their unrealistic experimentations. Based on some realistic observations on interconnect delay and power estimation, the spatial switching technique is proposed. The power consumption reduction can reach up to 12% for a 5-mm bus and more if buses are longer.

Tsai et al (2004) formulated an algorithm based on zero slew clock tree optimization with buffer insertion or sizing and wire sizing. It deals with the study of problem of multi-stage zero skew clock tree construction for minimizing clock phase delay and wire-length. In existing approaches clock buffers are inserted only after the clock tree is constructed. The novelty of this paper lies in simultaneously performing clock tree routing and buffer insertion. It proposes a clustering-based algorithm which uses shortest delay as the cost function. It shows that the feasible positions for clock tree nodes and buffers can be generalized from diagonal segments (merging segments) to rectangles (merging blocks). Buffers are large components and must be placed pair wise disjoint. It also shows that the problem of finding legal positions for buffers such that no buffers overlap can be formulated as a shortest path problem on graphs, and can be solved by the Bellman-Ford algorithm. By making use of the spatial properties of the graphs, it further speedup the Bellman-Ford algorithm. It also includes a new quick and effective legitimate skew clock routing with buffer insertion algorithm. It then analyzes the optimal buffer position in the clock path and concludes the sufficient condition and heuristic condition for buffer insertion in clock net [7].

Although optimizing for power dissipation, none of these disclosures has considered the commonly used power saving techniques for clock tree network such as clock gating and register clustering on meshes. For wire widths from 0.3 to 3 micro meters and buffer width from 1\* to 10\*, the algorithm achieves 45\* delay improvement and 1.25 \* power saving over initial routing. [7]

Alpert et al (2004) developed a speed up technique for optimal buffer insertion with minimum cost and thus reduces the complexity. As gate delays decrease faster than

wire delays for each technology generation, buffer insertion becomes popular method to reduce the interconnection delay. Several modern buffer insertion algorithms are based on Van Ginneken's dynamic programming paradigm. However, Van Ginneken's original algorithm does not control buffering resources and tends to over buffering thereby wasting area and power. This algorithm settles this problem by showing that for arbitrary integer cost functions thus the problem of NP will be completed. It also extends the pre-buffer slack technique to minimize the buffer cost. This technique can significantly reduce the running time and memory in buffer cost minimization problem[8].

Finally, it shows how to deal efficiently with the multi way merge in buffer insertion. For buffer insertion with minimum cost it involves the frame work of Lillis et al and apply the pre-buffer slack pruning technique to both solution set indexed by cost and the range query tree. Since all new candidates are visited in non-decreasing order of cost, the pre-buffer slack pruning technique guarantees the optimality. Multiple buffer types are also considered here. However, this will increase the space complexity by a factor of modulus B. It can avoid the extra space increase by only pruning those candidates dominated by other candidates. There will be less redundant solutions pruned compared with using modulus B trees but the experiments show that there are still many redundant solutions been pruned compared with previously Q pruning technique. To show the advantage of new pruning technique, this has been tested with the new algorithm for buffer insertion with cost constraints.

Six different buffer types based on TSMC 180nm technology is used. The largest buffer is 8X. The sink capacitances range from 2fF to 41 fF. Intrinsic gate delay is identical for all buffers. All algorithms are implemented in C and runs on Sun SPARC workstations with 400 MHz and 2 GB memory. The implemented algorithms also output buffer positions[8].

The buffer cost is the number of buffers. The numbers of buffer positions are the number of sinks. This algorithm is 2 to 17 times faster than Lillis algorithm and uses 1/1.5 to 1/30 of memory. The performance of the algorithm is better when the size of buffer library is large. Usually, the buffer library is large in industry designs so that the algorithm can achieve more significant improvement in practice. During buffer insertion, one problem is to deal with routing tree vertices of out degree greater than 2. Although any such vertex can be replaced by a number of out degree 2 vertices, the order is not unique. Each new vertex, solid or hollow is a possible buffer position. Solid vertices involve merging and buffer insertion. Hollow points involve buffer insertion only. It proves that the buffer cost minimization problem is NP complete in general. On the other hand, it proposes an algorithm using the pre-buffer slack pruning technique for the buffer insertion problem with buffer cost constraints. It also shows an efficient way to merge vertices with degree greater than 2 in buffer insertion. [9]

Pan et al(2006) put forward an algorithm that includes buffered clock network synthesis with cross links by tolerant variation. Clock network synthesis is a key step in the ultra-

deep sub-micron (UDSM) VLSI Designs. It also uses SPICE for tuning the locations of internal nodes and buffer delays, thereby making it slow even for clock networks with a few hundred sinks. In this paper, it proposes a unified algorithm for synthesizing a variation tolerant, balanced buffered clock network with cross links. This approach can make use of ordinary buffers and does not require SPICE for clock network synthesis. SPICE based Monte Carlo simulations show that methodology results in a buffered clock network with 50% reduction in skew variability with minimal increase in wire-length, buffer area and CPU time[9].

In the sub-65nm VLSI technology, the variation effects like manufacturing variation, power supply noise and temperature variation become very significant. As one of the most vital components in any synchronous VLSI chip, the clock distribution network (CDN) is especially sensitive to these variations. The unwanted clock skews caused by the variation effects consume increasing proportion of the clock cycle, thereby limiting chip performance and yield. Thus, making the clock network variation-tolerant is a key objective in the chip designs of today. In this dissertation, it proposes several techniques that can be used to synthesize variation-tolerant clock networks [10].

The contributions can be broadly classified into following four categories: (i) Efficient algorithms for synthesizing link based non-tree clock networks. (ii) A methodology for synthesizing a balanced, variation tolerant, buffered clock network with cross-links. (iii) A comprehensive framework for planning, synthesis and optimization of clock mesh networks. (iv) chip-level clock tree synthesis technique to address issues unique to hierarchical System-On-a-Chip (SOC) designs that are becoming more and more frequent today. Depending on the performance requirements and resource constraints of a given chip, the above techniques can be used separately or in combination to synthesize a variation tolerant clock network. Unwanted clock skews caused by variation effects like manufacturing variations, power-ground noise etc.[11].

Thus, reducing the clock skew variations is one of the most important objectives of any high-speed clock distribution methodology. Inserting cross-links in a given clock tree is one way to reduce unwanted clock skew variations. However, most of the existing methods use empirical methods and do not use delay or skew variation information to select the links to be inserted. This can result in ineffective links being inserted. It considers the delay variation directly, but it is very slow even for small clock trees [12].

Among these sources, "variations" have become one of the more significant challenges in synchronous clock-tree design. These variations can include: manufacturing process variations, operational voltage variations, and ambient temperature variations. Furthermore, OCV causes uncertainty in clock arrival times at circuit components. This uncertainty can cause clock skew and can thereby worsen the timing performance of the data paths between the clock sinks. In order to reduce the effects of OCV, some systems insert shunt connections called "cross links" into a clock-tree structure in a post-processing step. These cross links can

increase the amount of clock-path sharing between registers, thereby improving OCV-tolerance [13].

Huang et al (2007) presented DME based clock routing in the presence of obstacles. A method and service of balancing delay in a circuit design begins with nodes that are to be connected together by a wiring design or by being supplied with an initial wiring design that is to be altered. The wiring design will have many wiring paths, such as a first wiring path, a second wiring path, etc. Two or more of the wiring paths are designed to have matching timing such that the time needed for a signal to travel along the first wiring path is about the same time needed for a signal to travel along the second wiring path, the third path etc.[14].

Li et al [2009] contributed towards the fast algorithm for slew constrained minimum cost buffering. A buffer insertion technique addresses slew constraints while minimizing buffer cost. The method builds initial solutions for the sinks, each having an associated cost, slew and capacitance. As a solution propagates toward a source, wire capacitance and wire slew are added to the solution. When a buffer is selected for possible insertion, the slew of the solution is set to zero while the cost of the solution is incremented based on the selected buffer and the capacitance is set to an intrinsic capacitance of the buffer. The solutions of two intersecting wire branches are merged by adding branch capacitances and costs, and selecting the highest branch slew [15].

A microelectronic integrated circuit chip can generally be thought of as a collection of logic cells with electrical interconnections between the cells, formed on a semiconductor substrate. An IC may include a very large number of cells and require complicated connections between the cells. A cell is a group of one or more circuit elements such as transistors, capacitors, resistors, inductors, and other basic circuit elements grouped to perform a logic function. Cell types include, for example, core cells, scan cells and input/output (I/O) cells. Each of the cells of an IC may have one or more pins, each of which in turn may be connected to one or more other pins of the IC by wires. The wires connecting the pins of the IC are also formed on the surface of the chip [16].

Shih et al (2010) put forwarded fast timing model which is independent buffered clock tree synthesis. In high performance synchronous chip design a buffered clock tree with small clock skew is essential for improving clocking speed. Due to the insufficient accuracy of timing models for modern chip design, embedding simulation into a clock tree synthesis flow becomes inevitable [17].

It proposes an ultra-fast timing model independent approach to perform skew minimization by structure optimization. A symmetrical structure was presented. At each level of symmetrical clock tree, the number of branches, wire length and the inserted buffers are almost the same. It is natural that the clock skew could be minimized if the configurations of all paths from the clock source to sinks are similar. By symmetrically constructing a clock tree, the clock skew can be minimized without referring to simulation information [18].

In particular, the work provides a key insight into the importance of handling practical design issues for real-world clock-tree synthesis. A clock tree typically consumes substantial dynamic power and thus the considerable heat generated by it can cause serious clock-skew variations. In this paper, it also proposes a self-heating-aware buffered clock tree synthesis flow. A mixed integer linear programming (MILP) formulation is proposed to simultaneously model heat spreading, place buffers, and determine a temperature-aware clock tree topology. The formulation is then transformed into a succession of low-complexity feasibility problems to further reduce the runtime. In addition, a fast superposition approach is proposed to incrementally update thermal profiles to reduce simulation time.

Chen et al (2010) explained the clock tree synthesis under aggressive buffer insertion. As VLSI circuits are aggressively scaled down, the interconnections have become performance bottlenecks. Buffer insertion is extensively relied- upon to reduce interconnect delay at the expense of increased power dissipation. Given a routing tree, partial solutions in each tree node are constructed and propagated in a bottom-up fashion. When the optimal solution is identified in the root node, a top-down back-trace is performed to get the optimal buffer assignment. Following this dynamic programming framework, various delay optimization buffer insertion algorithms have been developed in the existing literature, such as a proposal for wire segmenting with buffer insertion; for handling multi-source nets repeater insertion; considering noise and delay optimization simultaneously in buffer insertion; presenting an efficient algorithm for delay-optimal buffer insertion with  $O(n \log n)$  time complexity by employing a sophisticated data structure. [18]

A method of optimizing a logic or clock interconnection tree, comprising: generating an interconnection tree having a source node interconnected by wires to a plurality of sink nodes through a plurality of Steiner nodes and a plurality of candidate buffer nodes; selecting a multi-Vdd buffer for insertion at a selected candidate buffer node within a given routing tree to reduce power consumption while reducing delay or delay difference; inserting buffers at selected candidate buffer nodes between said source and sink nodes within said given routing tree of said interconnect tree; wherein buffers are inserted in said given routing tree so that buffers with lower Vdd are not placed along a routing path of the routing tree before buffers with higher Vdd; and wherein said buffers are inserted without the inclusion of level converters between buffers.

Mittal et al (2011) explained the cross link insertion algorithm for improving tolerance variation in CTS. Clock trees are short wired which is having unique path from source to sink and also more susceptible to process variations. But clock mesh is high wired which is having high cost. Most robust to process variations and has got many paths from source to sink. As a result, clock trees are more preferable compared to clock mesh. Cross link compromises between clock tree and clock mesh. Due to cross link addition, skew between the different nodes are

changed. Links are inserted between two sinks. Cross links have been used to reduce skew variations in clock trees [19].

In earlier studies cross links are inserted between the sinks of DC-connected trees. In this paper, it proposes a link insertion scheme that inserts cross links between internal nodes of a clock tree. In addition to reducing the skew variability, the proposed approach also reduces the total cross link length. The work also improves the correlation of sink delays for those sinks within a sub tree that have similar path lengths to the cross link. Monte-Carlo (MC) simulations on the ISPD-2010 benchmarks showed that our work could handle variations effectively. In addition to meeting all the design constraints, the solutions produced by the approach have on the average 32% lower capacitance than the least capacitance obtained. Clock distribution is one of the key limiting factors in any high speed, sub-100nm VLSI design. Unwanted clock skews, caused by variation effects like manufacturing variations, power-ground noise etc., consume increasing proportion of the clock cycle. Thus, reducing the clock skew variations is one of the most important objectives of any high-speed clock distribution methodology. Inserting cross-links in a given clock tree is one way to reduce unwanted clock skew variations [19].

However, most of the existing methods use empirical methods and do not use delay/skew variation information to select the links to be inserted. This can result in ineffective links being inserted. The work considers the delay variation directly but it is very slow even for small clock trees. In this paper, the authors propose a fast link insertion algorithm that considers the delay variation information directly during link selection process. This algorithm inserts links only in the parts of the clock tree that are most susceptible to variation effects by evaluating the skew sensitivity to variations. Another key feature of the algorithm is that it is compatible with any higher order delay model, unlike the existing algorithms [19].

The effectiveness of our algorithm using HSPICE based Monte Carlo simulations on a set of standard benchmarks. Generally, the sequential elements that are related, for example one element feeding data to the other, are placed closer to each other. The clock skew between any pair of sequential elements that are separated by less than a specified distance is defined as Local Clock Skew (LCS). Cross links inserted in a buffered clock tree has been shown to be effective in reducing the skew variations. In earlier work cross links are inserted between the sinks of DC-connected trees. In this work it includes link insertion scheme that inserts cross links at higher level internal nodes in a clock tree. In addition to reducing the skew variability, the proposed work also reduces the total cross link length [19]. The work also improves the correlation of sink delays as the sinks in a sub tree have the similar path lengths to the cross link. NG Spice based Monte Carlo simulations verifies the effectiveness of the approach.

Chang et al [2012] modeled an algorithm for low power robust clock tree through slew budgeting. Clock skew resulted by process variation becomes more and more serious as technology shrinks. In 2010, ISPD held a high performance clock network synthesis contest; it considered

supply-voltage variation and wire manufacturing variation. Previous works show that the main issue of variation induced skew is on supply-voltage variation. To trade off power and supply-voltage variation induced skew more effectively, it adapts a tree topology which use a timing model independent symmetrical tree at top level to drive the bottom level non-symmetry trees. This method gives top tree more power budget to reduce supply-voltage variation induced skew and greedily saves power consuming in bottom level. [20]

Experimental results are evaluated from the benchmarks of ISPD contest 2010. Compared with state-of-the-art cross link work, the proposed technique reduces 10% of power consumption on average and also improves the run time. Clock power consumption is a factor of capacitance, switching activity, and wire length. Low-power CTS strategies include lowering overall capacitance and minimizing switching activity. Additional features, such as slew shaping and the ability to define skew groups are also beneficial in reigning in clock power. [20] Slew shaping techniques push the majority of cases closer to target slew, eliminates transitions that are overly pessimistic, and meets timing requirements while minimizing dynamic power.

However, getting the best power results from CTS depends on the ability to synthesize the clocks for multiple corners and modes concurrently in the presence of design and manufacturing variability. Multi-corner CTS can measure early and late clock network delays over all process corners concurrently with both global and local variation accounted for. MCMC CTS can make dynamic tradeoffs between either buffering the wire or assigning it to less resistive layers in order to achieve the best delay, area, and power. The effect of MCMC CTS is to minimize functional skew and skew variation across corners. The CTS engine should allow easy setup and accurate representation of all mode/corner/power scenarios, then analyze, synthesize and optimize them concurrently using a single, unified timing graph.

In advanced node designs, clock trees have become extremely complex circuits with different clock tracing per circuit mode of operation. The growth of mode/corner/power states and the large variations of resistance seen across various process corners pose new challenge in minimizing the power used by clock trees. Traditionally, CTS engines attempt to achieve zero skew by balancing the signal arrival time across all the flops regardless of which level of the clock tree they inhabit. However, not all clock ends points need to be balanced with each other. To balance different clock end points, designers have to manually craft multiple CTS specs and perform multiple CTS runs. This method is time consuming and error prone. A better way is for the CTS engine to analyze flop interactions to derive the exact skew balancing requirements at the different clock tree levels, and also across different voltage islands. From a given clock, it could then balance only the end points of the defined skew groups in a single call to the CTS engine. The tool should be able to discover skew groups by analyzing connected components in the timing data structure. Using skew groups saves processing time because the tool isn't trying to balance clock endpoints

that need not be balanced. It also reduces the number of buffers inserted, and eliminates manual CTS specifications and multiple CTS runs. [20]

Designers need clock synthesis and optimization tools that are built to handle MCMM scenarios and that use advanced CTS techniques like slew shaping, skew groups, and intelligent clock gating. With MCMM and low-power CTS optimizations, designers can reclaim a significant amount of power from their clock trees without sacrificing area, timing, or performance, or time to closure.

Finally, Cai et al [2014] formulated clock tree synthesis approach with efficient buffer insertion which results in skew optimization. An obstacle aware algorithm called OBB is proposed to generate the clock tree topology with an overall view on obstacles[21.] A look up table is built through NG Spice simulation to achieve accurate buffer delay and slew which guarantees that the final skew after NG Spice simulation is as satisfactory as expected.

A novel buffer insertion algorithm is developed which uses reasonable number of CBPs to satisfy slew constraints. The CBPs are carefully chosen to ensure the quality of final

solution whose target is skew optimization. An efficient sampling technique is adopted to speed up the buffer insertion algorithm on the basis of skew optimization during the process of buffer insertion. The key improvement is to change the sink's order in the weight ascending list. It assumes that all sinks are of same capacitance to eliminate the influence on the partitioning result brought by the unbalanced capacitance of sinks. It should be noted that this algorithm is valid on different layouts with clock sinks and obstacles. If REF consists of sinks s1, s2, s3 and s4, BB will partition the sink set by solid line where sink ss will finally be connected by the dotted lines. In this situation, the obstacle directly affects the lowest level of the tree and this unbalance will propagate to all the other paths from bottom to up levels[21]. To construct a balanced CBP distribution, algorithm applies heuristic strategies to improve the insertion of CBPs. A balanced CBP insertion algorithm is to eliminate the negative effect caused by the wire snaking for obstacle avoidance. This algorithm obtains 53.2% improvement in skew than the classic balanced bipartition algorithm.

### 3. Comparative Analysis

**Table 3.1:** Various algorithms used for efficient buffer insertion in clock tree synthesis

Author	Algorithm	Advantages	Disadvantages	Results
Cai et al(2014)	1) Obstacle aware topology generation algorithm (OBB) 2) Balanced insertion of candidate buffer positions 3) Fast heuristic buffer insertion algorithm 4) 4.Deferred Merge Embedding algorithm (DME)	1) Skew optimization. 2) Effectively overcomes the negative influence of obstacles over skew. 3) Satisfaction of slew constraints and signal polarity. 4) Accurate buffer delay with shorter runtime. 5) Reduces the routing cost.	1. Time consuming. 2. Buffer intrinsic delay is sensitive to input slew changes. 3. It will not collect the slew information within an RC net list. 4. Accuracy of Elmore delay is limited.	1. The skew and maximum latency are reduced by 69% and 72% on average. 2. The accuracy of the look up table is demonstrated through skew reduction by 87.3% on average. 3.OBB obtains 53.2% improvement in skew. 4. The skew with obstacles shows 6.3% reduction.
Chang et al(2012)	High performance clock network through slew budgeting algorithm	1.Improves power efficiency of buffer insertion. 2. Improves the performance of clock tree network. 3. Less number of embedded SPICE simulations is needed. 4.Latency minimization.	1. Robustness 2. Process variation like decreasing VDD and interconnecting issues. 3. Buffer cannot overlap with a blockage. 4. Buffer placement decision is not much flexible.	1.10% power reduction than state of the art clock network. 2.Hybrid structure makes slew optimization easier. 3. Skew limit is greater than 95%.
Mittal et al(2011)	Cross link insertion algorithm for tolerance improvement in clock tree	1. Reduces the total cross link length. 2. Improves the correlation of sink delays for those sinks within a sub tree. 3. Reduces skew variations in clock trees.	1. Shorter wiring. 2. Unique path from source to sink. 3. More susceptible to process variation. 4. Higher wiring cost.	NG Spice based Monte Carlo simulations verifies the effectiveness of the approach.
Shih et al(2010)	Symmetrical Clock Tree Synthesis algorithm for time independent model	1.Skew minimization by structure optimization. 2.Optimizes slew rates and signal phase latency of clock trees. 3.Minimizes clock skew with marginal wiring overheads.	1. Huge run time 2.Insufficient accuracy of timing models for modern chip design. 3.Embedding simulation into clock tree flow becomes inevitable.	1. The work without ngspice simulation results in average of 7.93X clock skew and requires 46X run time over the approach. 2. With the help of ngspice the average becomes 2.77X clock skew and 24343X run time.

Chen et al(2010)	Clock tree synthesis under aggressive buffer insertion algorithm	<ol style="list-style-type: none"> <li>1. Robust slew control.</li> <li>2. Accurate timing analysis engine for delay and slew estimation.</li> <li>3. Balanced routing scheme for better skew reduction during CTS.</li> </ol>	<ol style="list-style-type: none"> <li>1. Potential buffer insertion locations are restricted to the merge nodes in the clock tree topology.</li> <li>2. In general scenarios, simple buffer arrangement will not be sufficient to meet a hard slew constraint.</li> </ol>	<ol style="list-style-type: none"> <li>1. Maintained reasonable skew.</li> <li>2. For every benchmark worst slew does not exceed the slew limit of 100ps.</li> <li>3. Run time is within the minutes.</li> <li>4. All skews are within 3% of maximum latency.</li> </ol>
Li et al(2009)	Fast algorithm for slew constrained minimum cost buffering	<ol style="list-style-type: none"> <li>1. Minimizes buffer cost.</li> <li>2. Faster performance and predictability of responses.</li> </ol>	Due to the large number of components and the details required by the fabrication process for very large scale integrated (VLSI) devices, physical design is not practical without the aid of computers.	Automation of the physical design process has increased the level of integration, reduced turnaround time and enhanced chip performance.
Huang et al(2007)	Deferred merge embedding based algorithm	<ol style="list-style-type: none"> <li>1. Minimizes wire length.</li> <li>2. Yields exact zero skew trees.</li> <li>3. Very fast algorithm</li> </ol>	Increased switching speed results in lowering of interconnection timing specifications.	Averages 15% total wire length and 13% cost savings.
Pan et al(2006)	Variation tolerant algorithm in clock tree synthesis with cross links	Reduces skew variability.	It deals with only unbuffered clock networks making them impractical.	Results in a buffered clock network with 50% reduction in skew variability with minimal increase in wire-length, buffer area and CPU time.
Alpert et al(2004)	Complexity analysis and speed up technique algorithm for buffer insertion	<ol style="list-style-type: none"> <li>1. Minimizes buffer cost.</li> <li>2. Reduces memory and running time.</li> <li>3. Efficiently deals with multiway merge in buffer insertion.</li> </ol>	<ol style="list-style-type: none"> <li>1. It does not control buffering resources.</li> <li>2. Tends to over buffering.</li> <li>3. Wastage of area and power.</li> </ol>	Algorithm can speed up the running time up to 17 times and reduces the memory to 1/30.
Tsai et al(2004)	Zero skew clock tree optimization	<ol style="list-style-type: none"> <li>1. Zero skew</li> <li>2. Minimizes delay and power in polynomial time.</li> <li>3. More efficient.</li> </ol>	<ol style="list-style-type: none"> <li>1. Clock skew directly affects chip performance.</li> <li>2. Timing plans cannot be met due to physical effects.</li> </ol>	Clock tune achieves 45*delay improvement for buffering and sizing an industrial clock tree with 3101 sink nodes in 16 min.
Lillis et al(1996)	Low power and generalized delay model	<ol style="list-style-type: none"> <li>1. Efficient and optimal algorithm for timing optimization.</li> <li>2. Minimizes cost function over timing constraints.</li> <li>3. Minimizes dynamic power dissipation.</li> <li>4. Algorithm is easily adaptable to area minimization.</li> </ol>	The problem was formulated on the task of minimizing the weighted sum of source to sink Elmore delays in a given routing tree over set of identified critical sinks.	<ol style="list-style-type: none"> <li>1. Impressive run time was obtained typically in the 20-30 second range.</li> <li>2. As scaling factor increases, large variation between observed delays was noted and these variations approaches to 50% in each case.</li> </ol>
Chao et al(1992)	Zero skew clock routing with minimum wire length	<ol style="list-style-type: none"> <li>1. Reduces system power requirement.</li> <li>2. Reduces deformation of the clock pulses at the synchronizing elements of the system.</li> </ol>	Circuit speed is increasingly limited by two factors. They are: delay on the longest path through combinational logic and clock skew.	BB + DME method averages 15% wire length savings over the previous methods and also provides 10% average wire length savings when compared to the method of linear delay.
Ginneken et al(1990)	Buffer placement in distributed RC tree network algorithm	<ol style="list-style-type: none"> <li>1. Minimum Elmore delay.</li> <li>2. Growth of delay with wire length can be reduced to linear.</li> </ol>	The complexity of algorithm is quadratic in the number of legal positions for the buffer and the leaves.	The algorithm was implemented as a part of IBM's logic synthesis system and it is coded in PL/I.

The obstacle aware topology generation algorithm is found to be more effective through comparative studies of other algorithms. The accuracy and efficiency of the buffer insertion algorithm results in the required optimization of the skew and thus signal polarity has met the conditions. The power consumption and the run time have significantly

reduced to the expected level compared to other algorithms. Table 3.1 describes the comparative analysis of different algorithms which is used for the avoidance of obstacles using buffer placements.

## 4. Conclusion

This survey deals with the obstacle avoiding and slew constrained buffered CTS for skew optimization. The concentration was mainly towards slew constrained and skews targeted buffered CTS. A heuristic sampling technique was proposed in the DP based frame work. The integration of signal polarity into basic buffer insertion completes the flow. The obstacle is one important factor that affects the effect of traditional algorithms on buffered CTS. The common fixed length based CBP insertion is also heuristically developed on the consideration of balanced CBP distribution. The classical combination of balanced partition and deferred merge algorithm is improved to obstacle balanced partition and obstacle deferred merge algorithm. This paper reviews the effectiveness and robustness of the algorithms for skew optimization.

## References

- [1] L.P.P.P. van Ginneken, "Buffer placement in distributed RC tree networks for minimal Elmore Delay," in Proc. IEEE Int. Symp. Circuits Syst. May 1990, pp. 865-868.
- [2] F.Niu, Q.Zhou, H.Yao, Y.Cai, J.Yang and C.N. Sze, "Obstacle avoiding and slew constrained buffered clock tree synthesis for skew optimization," in Proc. 21. Glsvlsi, 2011, pp. 199-204.
- [3] T.H.Chao, Y.C.Hsu, J.M.Ho, K.D.Boese and A.B.Kahng, "Zero skew clock routing with minimum wire length," IEEE Trans. Circuits Syst., vol. 39, no. 11, pp.799-814, Nov.1992.
- [4] D.Lee and I.L.Markov, "Contago: Integrated optimization of SoC clock network," in Proc. DATE, 2010, pp. 1468-1473.
- [5] J. Lillis, C.K. Cheng and T.T.Lin, "Optimal wire sizing and buffer insertion for low power and a generalized delay model," IEEE J. Solid State Circuits, vol. 31, no. 3, pp. 437- 447, Mar. 1996.
- [6] Z.Li and W.Shi, "An  $O(bn^2)$  time algorithm for buffer insertion with b buffer types," in Proc. DATE, 2005, pp.1324-1329.
- [7] J.L. Tsai, T.H Chen and C.C.P Chen, "Zero skew clock tree optimization with buffer insertion/sizing and wire sizing," IEEE Trans. Comput. Aided Des.Integr.Circuits Syst. Vol. 23, no.4, pp. 565-572, Apr. 2004.
- [8] W.Shi, Z.Li and C.J. Alpert, "Complexity analysis and speed up techniques for optimal buffer insertion with minimum cost," in Proc. Aspdac, 2004, pp.609-614.
- [9] A.Rajaram and D.Z.Pan, "Variation tolerant buffered clock network synthesis with cross links," in Proc. Ispd. 2006, PP. 157-164.
- [10] X.W. Shih and Y.W. Chang, "Fast timing model independent buffered clock tree synthesis," in Proc. Dac, 2010, pp. 80-85.
- [11] Y.C Chang, C.K. Wang and H.M.Chen, "On constructing low power and robust clock tree via slew budgeting," in Proc. Ispd, 2012, pp. 129-135.
- [12] W.H.Liu, Y.L.Li and H.C.Chen, "Minimizing clock latency range in robust clock tree synthesis," in Proc. 15thAspdac, 2010, pp.389-394.
- [13] S.Hu, C.J. Alpert, J.Hu, S.Karandikar, Z.Li, W.Shi, et al., "Fast algorithms for slew constrained minimum cost

buffering," IEEE Trans.Comput. Aided Des.Vol.24, no.6, pp. 879-891, Jun.2005.

- [14] H. Huang, W.S Luk, W. Zhao and X. Zeng, "DME based clock routing in the presence of obstacles," in Proc., 7<sup>th</sup>Asicon2007, pp. 1225-1228.
- [15] R.Chaturvedi and J.Hu, "Buffered clock tree for high quality IC design" in Proc. 5th ISQED, 2004, pp. 381-386.
- [16] Z.Li and W.Shi, "An  $O(mn)$  time algorithm for optimal buffer insertion of nets with m sinks," in Proc. Aspdac, 2006, pp.320-325.
- [17] X.W. Shih, C.C. Cheng, Y.K. Ho and Y.W. Chang, "Blockage avoiding buffered clock tree synthesis for clock latency range and skew minimization," in Proc. 15th Aspdac, 2010, pp. 395-400.
- [18] W.Shi and Z.Li, "A fast algorithm for optimal buffer insertion," IEEE Trans.Comput.Aided Des., vol.24, no.6, pp. 879-891, Jun.2005.
- [19] Y.Y Chen, C. Dong and D.Chen, "Clock tree synthesis under aggressive buffer insertion," in Proc. 47 thDac., 2010, pp. 86-89.
- [20] T.Mittal and C.K.Koh, "Cross link insertion for improving tolerance to variations in clock network synthesis," in Proc.Ispd., 2011, pp. 29-36.
- [21] Y.Cai and Zhou Q, "Obstacle avoiding and slew constrained clock tree synthesis with efficient buffer insertion," in IEEE Trans.Comput.Aided Des., 2014.

## Author Profile



Technology.

**Anju Rose Tom** completed her B-Tech in Electronics and Biomedical Engineering under Cochin University of Science and Technology. Currently she is pursuing M-Tech in the specialization of VLSI and Embedded System under Cochin University of Science and



**Dr. Gnana Sheela K:** She received her Ph D in Electronics & Communication from Anna University, Chennai. Presently she is working as an Associate Professor in TIST, Kerala. She has published 14 international journal papers. She is a life member of ISTE.