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. MCMM 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 MCMM 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	1. Robust slew control. 2. Accurate timing analysis engine for delay and slew estimation. 3. Balanced routing scheme for better skew reduction during CTS.	1. Potential buffer insertion locations are restricted to the merge nodes in the clock tree topology. 2. In general scenarios, simple buffer arrangement will not be sufficient to meet a hard slew constraint.	1.Maintained reasonable skew. 2. For every benchmark worst slew does not exceed the slew limit of 100ps. 3. Run time is within the minutes. 4. All skews are within 3% of maximum latency.
Li et al(2009)	Fast algorithm for slew constrained minimum cost buffering	1. Minimizes buffer cost. 2. Faster performance and predictability of responses.	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	1. Minimizes wire length. 2. Yields exact zero skew trees. 3.Very fast algorithm	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	1. Minimizes buffer cost. 2. Reduces memory and running time. 3. Efficiently deals with multiway merge in buffer insertion.	1. It does not control buffering resources. 2. Tends to over buffering. 3. Wastage of area and power.	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	1. Zero skew 2. Minimizes delay and power in polynomial time. 3. More efficient.	1. Clock skew directly affects chip performance. 2. Timing plans cannot be met due to physical effects.	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	1. Efficient and optimal algorithm for timing optimization. 2. Minimizes cost function over timing constraints. 3. Minimizes dynamic power dissipation. 4. Algorithm is easily adaptable to area minimization.	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.	1. Impressive run time was obtained typically in the 20-30 second range. 2. As scaling factor increases, large variation between observed delays was noted and these variations approaches to 50% in each case.
Chao et al(1992)	Zero skew clock routing with minimum wire length	1. Reduces system power requirement. 2. Reduces deformation of the clock pulses at the synchronizing elements of the system.	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	1. Minimum Elmore delay. 2. Growth of delay with wire length can be reduced to linear.	The complexity of algorithm is quadratic in the number of legal positions for the buffer and the leafs.	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.,7th 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

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 Technology.

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.