

Obstacle aware delay optimized rectilinear steiner minimum tree routing

Shyamala G, G. R. Prasad

B M S College of Engineering, Bangalore, India

Article Info

Article history:

Received Jan 15, 2019

Revised Mar 18, 2019

Accepted May 20, 2019

Keywords:

Obstacle-avoiding

Rectilinear steiner tree

Slack and slew constraint

VLSI

ABSTRACT

This work presents a method to solve the problem of constructing Rectilinear Steiner Minimum Tree (RSMT) for a group of pins in the presence of obstacles. In modern very large-scale integrated circuit (VLSI) designs, the obstacles, generally blocks the metal and the device layer. Therefore routing on top of blockage is a possible solution but buffers cannot be placed over the obstacle. Modern VLSI design OARSMT construction has long wire length, which results in signal violation. To address this issue a slew constraint interconnect need to be considered in routing over obstacle. This is called the Obstacle-Avoiding Rectilinear Steiner minimum trees (OARSMT) problem with slew constraints over obstacles. The drawback of traditional OARSMT is that they only consider slew constraint, and delay constraints are neglected. It induces high routing resources overhead due to buffer insertion and does not solve global routing solution. This work presents an Obstacle Aware Delay Optimized Rectilinear Steiner Minimum Tree (OADORSMT) Routing to address the delay, slew constraint and reduce the routing resources. Experiments are conducted to evaluate the performance of proposed approach over existing approach in term of wire length and worst negative slack. The experiments are conducted for small and large nets considering fixed and varied obstacles and outcome shows the proposed efficiency over existing approaches. The OADORSMT is designed in such a way where it can be parallelized to obtain better efficiency.

*Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.*

Corresponding Author:

Shyamala G,

B M S College of Engineering,

Bangalore, India.

Email: shyamala.cse@bmsce.ac.in

1. INTRODUCTION

The wide growth of semiconductor industries has led to the evolutionary growth of its design. Current design allows million to billions of transistor to be embedded in to a single on chip (SOC). The evolutionary growth of this has led to many routing challenges such as power, propagation delay, buffer insertion and timing constraint. The rectilinear Steiner minimal tree (RSMT) problem has been adopted by various researchers and has been adopted to solve the fundamental problems in the fields of VLSI (very large-scale integrated circuit) design. The RSMT play a significant role in designing very large-scale integrated circuit.

The rectilinear Steiner minimal tree has been used to solve and connect a set of pins in routing stage and it is also used for computing the delay in the routing path, overall wire length and time constraint in the placement stage. However the RSMT construction is said to an NP-complete deterministic problem [1]. Considering the present design of VLSI there exist many obstacles such as IP blocks, Macro-cells and pre-routed nets. So there exists a harder research challenge in VLSI design in presence of obstacle.

Therefore transmitting over the blockage is a probable solution but positioning the buffer on top of obstacle is not possible. In the presence of blockage the RSMT problem is considered to be more challenging

and problematical. The issue pertaining to transmission of net in presence of obstacle has been wide an application design challenges in field of VLSI automation. The major goal of these designs is to reduce the routing length and resources to connect nets. The interconnection delay is prevailing over logical delay and transistor delay due to the growth of deep submicron technologies [2]. The signal integrity may degrade due to high interconnected resistance when considering longer connection. To solve the signal integrity problem the buffer can be placed, which breaks a wire in to smaller chunks. An important thing to be noted is that the buffers cannot be positioned on top of obstacle. Though transmission over the blockage is possible, the design should be aware of signal integrity issue for longer wires on top blockage that may be problematic in post transmission electrical fixups. The one possible method to address this is to build an avoiding rectilinear Steiner minimum tree (OARSMT).

RSMT construction avoiding the obstacles is said to be OA-RSMT (Obstacle Avoidance-RSMT) problem. The OA-RSMT has been widely adopted by various existing researcher in recent times [3-6]. In [5] and [6] the presented a model that address the issue pertaining to rectangular obstacles. To overcome in [4] presented a technique to address obstacles without dividing rectilinear obstacles into rectangle nature. The OA-RSMT [7, 8] approach can provide a feasible strategy that possess buffer and wires among adjacent blocks. However, it induces overhead, since it consider IP blocks as routing obstacles and considerably degrades routing resources of these IP blocks and also induces delay in routing path.

In reality there exist multiple routing layer and the obstacles are mostly presented in lower metal layer and device layer as a result the wire can be positioned on top of obstacle. The buffers are inserted in to Steiner tree to further split in to subtree in order to solve time and slew constraint. Since the buffers cannot be placed on top of blockages and to utilize routing resources over obstacle, RSMT must be built on to of blockages without violating the slew constraint. This problem is called as RSMT with reusing routing resources over obstacles (RSMT-RERR).

The main objective of our research is to solve the RSMT-RERR problem in the presence of obstacle. To improve the overall circuit performance we consider slew, slack and delay constraint on interconnects that are transmitted on top of obstacle. The slew parameter is a crucial factor in improving overall performance, since violation of slew and slack constraint will lead to delay optimization error and result in degrading performance. The existing model has considered solving slew constraint is more important than timing constraint. In [9] showed that major design are developed based slew on constraint and they considered that the timing constraint [10] can be solved if slew constraint is satisfied. Therefore the highlight of their technique is to avoid routing over the obstacle to solve slew constraint. This problem can be represented as OARSMT problem with slew constraint on top of blockage. However the slew constraint depends on length of wires and delay as a result the traditional OARSMT cannot be applicable for global routing solution since it does not consider the delay parameter and routing over the obstacle cannot assure slew violation and interconnect performance. The solution to this problem is to reduce the routing resource overhead.

This work present a delay optimized routing over the obstacle based RSMT in order to address the routing resource minimization and slack constraint. This model firstly present an accurate delay optimization information and RST is iteratively computed until a convergence is met and delay optimization is done to find the delay resisted path. Secondly model solves the slew and delay optimizer failure to improve the slack performance. Lastly the delay optimizer position of Steiner points is optimized to reduce optimizing cost and wire length.

1.1. The Contribution of Work Can be Classified As Follows:

The proposed methodology present a delay optimized routing on top of obstacle based RSMT which solves the global routing problem [11] and reduces routing resources. The exiting Elmore model induces higher error for slew calculation. To overcome this model incorporates slew calculation presented in [12] in to our model. The proposed model addresses the slew constraint for routing on top of obstacle and improves performance in term of worst negative slack. The proposed approach reduces the number of logic gates required for creating efficient RSMT in order to reduce delay and also reduces the wire length. The paper organization is as follows: The proposed Obstacle aware delay optimized RSMT models are presented in Section two. The experimental study considering various benchmark are presented in penultimate section. The concluding remark and future work is discussed in the last section.

2. LITERATURE SURVEY

In the past several attempts have been made to solve RSMT-RERR problem in the presence of obstacle. In [13] presented a model for high performance synchronous chip design. The buffered clock tree synthesis (CTS) adoption in VLSI design is a crucial factor to achieve high performance. The model addressed the signal polarity correction and slew constraint in obstacle aware topology generation (OBB).

They presented a fast lookup based simulation model namely NGSPICE. The NGSPICE achieves accurate slew and buffer delay by adopting OBB algorithm. The OBB algorithm provides fast heuristic buffer insertion and balances the insertion of buffer placement. The model explored the global routing optimization space and overcomes the negative impact on skew due to obstacles. The outcome shows there model improves skew performance and reduce latency. However the model did not consider minimizing wire length and process variation.

The OARSMT minimize wire length without intersecting obstacle by connecting all pin-vertices through Steiner nodes using horizontal and vertical nodes. To deal with process variation and multiple routing layers, [14] consider OAPD-ST (obstacle avoidance preferred direction Steiner tree) problem and multilayer OARSMT problem. Firstly they presented a multilayer theoretical model from 2D and presented a transformation of multilayer instance to 3D instance. They adopted computational geometric technique to present an efficient technique using existing heuristic algorithm to solve ML-OARSMT. To further improve performance a new Steiner node selection method is presented, which avoids inferior Steiner nodes. The overall outcomes show the model achieves good speed up. However the optimality guarantee is not presented for higher pin nets and did not consider performance evaluation for different unit wire cost.

In [8] presented a model to solve OARSMT problem. They presented a model for Steiner node selection that solves the bottleneck of existing heuristic algorithm. They presented an approach of creating a linear space routing graph, Steiner node positioning with satisfactory Steiner node candidates and resolve issues of existing heuristic routing graph. The model achieves good quality and speedup performance. The model also shows it work well for OAPD-ST problem. However their models did not address RSMT-RERR problem in the presence of obstacle. As a result the routing resources are wasted. To maximize the routing resource over the obstacle, the slew constraint should not be violated when constructing RSMT on top of obstacle and this problem is called as RSMT-RERR. Many approaches have been presented to solve RSMT-RERR [15, 16].

The RSMT-RERR can take full benefit of routing resource over blockage, buffer resources and also saves routing resources outside the blockage. More importantly it reduces the delay, routing congestion, power consumption and minimized wire length. At the same new constraint will further increases complexity and especially connecting multi-pin nets. In [15] adopted length-restricted Steiner minimum tree (LRSMT). The model design is based on reach aware visibility graph. The visibility graph should at least contain one shortest probable path which is length restricted among any vertices pair (obstacle corner points or terminals). The model consists of two phases, preprocessing and post processing. In preprocessing phase, the running time are saved for generation and reducing the size of visibility graph. Here small obstacle with diameter smaller than L are ignored. Then in post processing phase, obstacle-unaware Steiner tree algorithms [17] are adopted to reconnect pins to improve quality. In [15] the blocked Steiner nodes are limited in reach aware visibility graph to assure feasible outcome for all solution. The outcome of connected element that are coincides by the blockages of the outcomes are all paths rather than Steiner graphs which consists of numerous Steiner nodes. However the simplified LRSMT problem increases the wire length and degrades wiring performance.

An efficient model to solve RSMT-RERR is presented in [16], the model adopted [18] for slew rate calculation. This model is called as OARSMT_SC (OARSMT problem with slew constraints over obstacles). In [16] presented an optimal solution to determine OARSMT_SC by extending Hanan grid. They formulated that OARSMT_SC can be divided into a set external and internal trees. The model design is composed of two stages. In first stage, the model computes a candidate set of external and internal trees. In stage two, it chooses and combines a subset of these trees to construct an ideal solution using ILP (Integer Linear Programming). The modified Hanan grids consist of escaping graph and Hanan grids. Extending Hanan grid may not solve LRSMT problem. Therefore their model is not suitable for rapid processing in physical design because of the considerable ILP problem.

Extensive research survey carried out shows that there is a need to develop a model that minimize wire length and reduce slew overhead. It is seen that most of the approaches have adopted OARSMT which induces wastages of routing resource. Routing over the blockage is an efficient method to utilize resource efficiently. However the existing model presented so far is not efficient in addressing sew constraint and minimizing delay. This work overcome these challenges and presents an efficient model namely Obstacle Aware Delay Optimized Rectilinear Steiner Minimum Tree (OADORSMT) Routing to address the delay, slew constraint and reduce the routing resources. In next section the proposed OADORSMT model is presented.

3. PROPOSED OBSTACLE AWARE DELAY OPTIMIZED RSMT MODEL

Here we present obstacle aware delay optimized rectilinear seiner tree construction and routing. The model consists of following phases, Delay Optimization phase, Routing Optimization phase and Minimization of path construction computing cost phase.

Firstly we model delay optimization, here a delay optimized topology is generated to obtain influx time and delay resisted path at each node. Then based on the information obtained in delay optimization phase, a delay optimized rectilinear Steiner tree is constructed.

In routing phase, routing on top of obstacle is considered in our work as shown in Figure 1 rather than routing avoiding obstacle as shown in Figure 2. The routing on top of obstacle will lead to slew constraint error due to delay optimizer failure constraint. Then delay optimization is done on the reconstructed topology, in order to address the slew constraint and improve slack performance. Lastly the delay optimized topology structure is optimized based on delay optimized parameter and further delay optimization is done to obtain the best path and minimize cost. Table 1 shown the notation and symbol used.

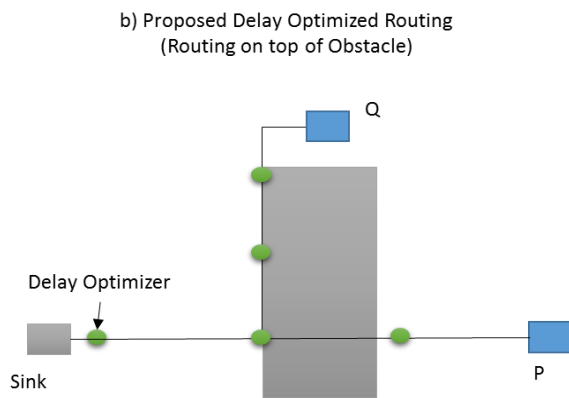


Figure 1. Proposed On top of Obstacle based Rectilinear Steiner Minimum Tree Construction which consist of two sink P and Q and one root sink

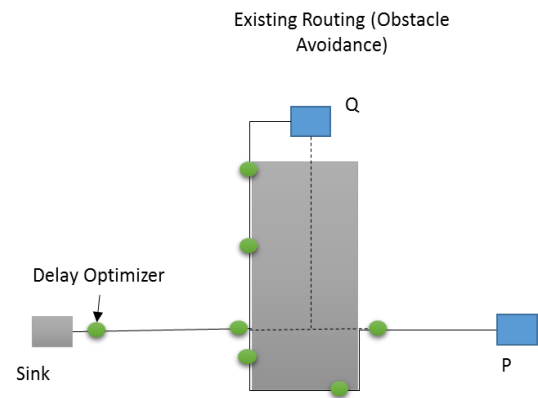


Figure 2. Existing Obstacle Avoidance based Rectilinear Steiner Minimum Tree Construction which consist of two sink P and Q and one root sink

Table 1. Notation and Symbol used

Symbol Used	Abbreviation
H	Set of pins in a transmission area
A	Set of graph
G	A set of non-intersecting rectilinear area
x	Non-intersecting rectilinear block
T	Transmission area
C	Collection of sink
S	Set of horizontal and vertical edge
\mathcal{S}	Set of nodes
K	Distinctive route
z	Steiner nodes
n	Node
m	Wire capacitance size on a particular layer
g_c	Edge size of c
m_c	Wire size of capacitance unit on a particular layer for edge c
t_c	Wire size of resistance unit on a particular layer for edge c
Z_x	Delay due to presence of buffers
T_x	Selected device resistance outcome
$z(n)$	Steiner point of pin n
$C_n(s)$	$M_x(s)$ if s is a root or delay optimized node else M_z if s is a not a root or delay optimized node
M_x	Selected device capacitance outcome
$M_z(s)$	overall capacitance of the sub-tree transmitted at pin s to the closest sink which includes delay optimized input capacitance
$r(n, s)$	Propagation time from pin n to s

The overall flow of proposed Routing On top of Obstacle based Rectilinear Steiner Minimum Tree Construction is presented in Figure 3.

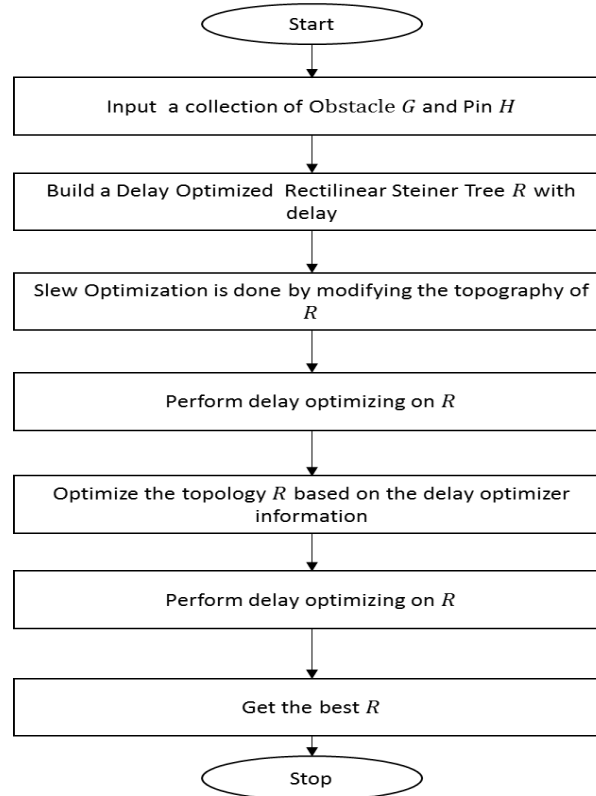


Figure 3. Delay optimized routing on-top of obstacle model

Let consider a set of pins of net $H = \{a_0, a_1, \dots, a_n\}$ in a transmission area with $h + 1$ pins, where a_0 the distinctive root is and others are the sink nodes. Let consider a collection of $G = \{x_0, x_1, \dots, x_i\}$ non-intersecting rectilinear block in a transmission area T . Let for all $a_y \in H$, a_y is not in the region possessed by G . The blockage G is considered for largely populated and complex cells.

The proposed approach produces a delay optimized graph $R(C, S)$ to associate all pins in H where S collection of vertical and horizontal edges and C is the collection of sinks. The set of graph $A = \{R_1, R_2, R_3, \dots, R_g\}$ is considered to be inside the block if R overlap with block G . The graph A is represented as inline graph and the outline block graph R is defined as \mathbb{R} . The delay optimized graph $R_x(S_x, C_x)$ is computed using R once the insertion of nodes \mathbb{S} that are related to delay optimizer selected from delay optimized table X and $S_x = S \cup \mathbb{S}$. There exist a distinctive route $K(a_0, a_y)$ from a_0 to a_y in Steiner graph and presence of delay optimizer along the transmission root will further split the path in to sub level graph. Each split consist of Steiner nodes, a collection of slack optimized sinks and also edges connecting Steiner nodes and slack optimized sinks.

The cumulative delay of the propagating path can be computed by summing up the delay at each level. The Elmore technique is been adopted by various existing researcher to measure delay. This work also adopts the Elmore model to compute delay for wires and for delay optimizer (adding gates) we adopt a switch level delay estimation model. The total delay at each level of the propagating route is formulated as follows.

$$r(z(n), u) = \sum_{c=(y,l) \in k(z(n),n)} t_c g_c \left(\frac{1}{2} m_c g_c + G_n(l) \right) + T_x M_z(z(n)) + Z_x \quad (1)$$

The cumulative propagating delay is obtained by summing each sublevel of propagating path is computed as follows

$$z(a_0, a_y) = \sum_{n \in \mathbb{S} \cap k(a_0, a_y)} r(z(n), n) \quad (2)$$

The worst negative slack V is computed as follows

$$V(R) = \min\{0, V\} \tag{3}$$

where V is the worst slack which is computed as follows

$$V = \min\{\text{slack}(a_y)\} \text{ where } 1 \leq y \leq h \tag{4}$$

where $\text{slack}(a_y)$ is the slack of node a_y which is computed as follows

$$\text{slack}(a_y) = O(a_y) - z(a_0, a_y) \tag{5}$$

where O is the delay optimized influx time. The slew rate is computed using [18] which is as follows

$$S_{v_{out_0}} \left(\sqrt{S_{v_{out}}^2 + S_{step}(v_{in}, v_{out})^2} \right) \tag{6}$$

where $S_{v_{out_0}}$ is the slew at any node v_{out_0} , $S_{v_{out}}$ is the output slew at node v_{out} and $S_{step}(v_{in}, v_{out})$ is the step slew from v_{in} to v_{out} . The step slew is computed as follows

$$S_{step}(v_{in}, v_{out}) = \ln 9 * \text{Elmore}(v_{in}, v_{out}) \tag{7}$$

3.1. Delay Optimized Tree Creation:

The tree creation of RST might require required influx time on internal nodes during phase of tree creation and requires estimation of influx time on each sink. The flow diagram of delay optimized tree creation is shown below in Figure 4.

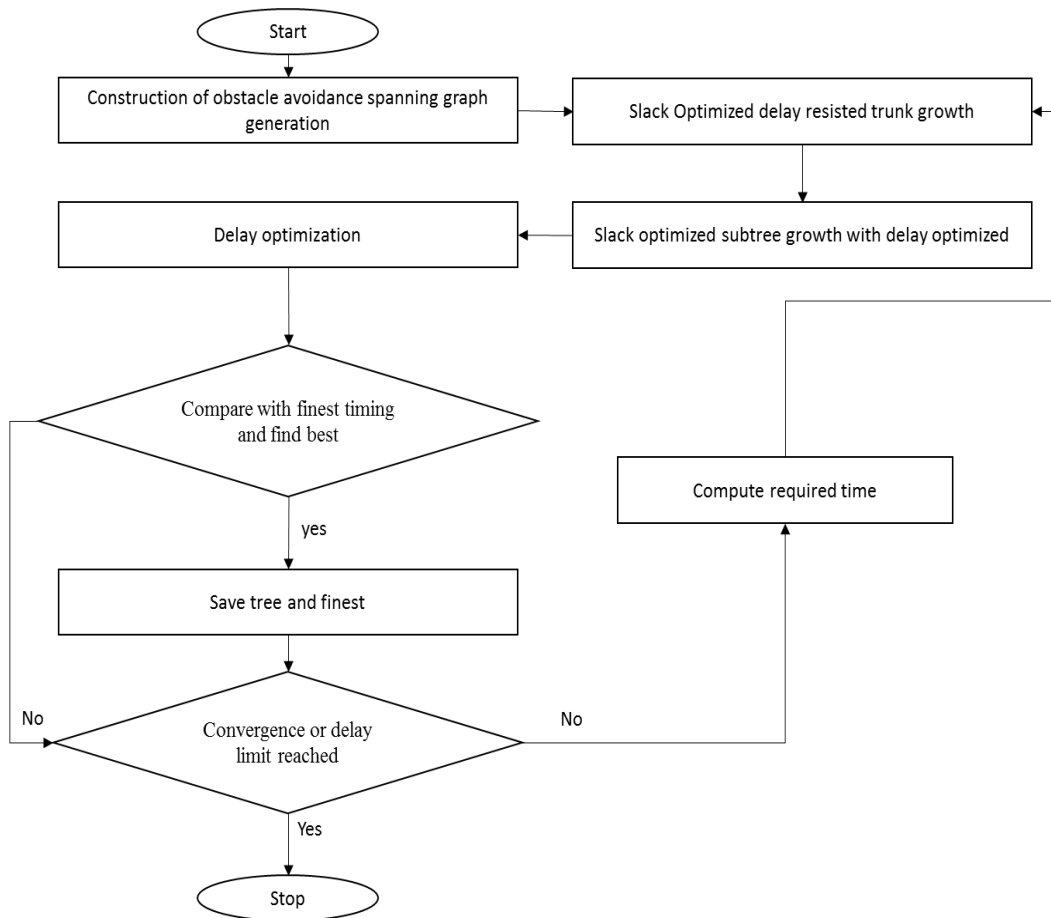


Figure 4. Flow diagram of delay optimized tree creation

First create an initial tree using delay optimized RST model. Then in delay optimization phase, analyse timing and delay optimize the RST. Then the topology and delay optimizer param is saved if it is ideal so far. We further compute the actual influx time based on delay optimized tree. This is done to obtain feedback information (delay optimized time) for topology generation methodology. In next step, all actual delay resisted trunk and sink are recomputed using delay optimized param. In RST model delay resisted trunk is re-positioned and other 2-pin nets are ripped apart and routing path is changed applying maze routing after delay optimized delay resisted trunk growth. Lastly it produces another RST after application of redirection and rectilinearization. The process is continued until time limit is reached or the tree topology converges.

In Figure 5, 6 and 7 shows that the timing and topology converges during each steps. In Figure 5, initial structure directly connects Sink S to root as the required influx time of S is very less. In next step the topology creation model decides to join S to the trunk as shown in Figure 6, since in Figure 5 the delay to S is very less, which can satisfy the required influx time. As a result late branch can be permitted. Allowing addition of late branch will incur more delay on sink P and leads to topology convergence and result in star like RSMT structure.

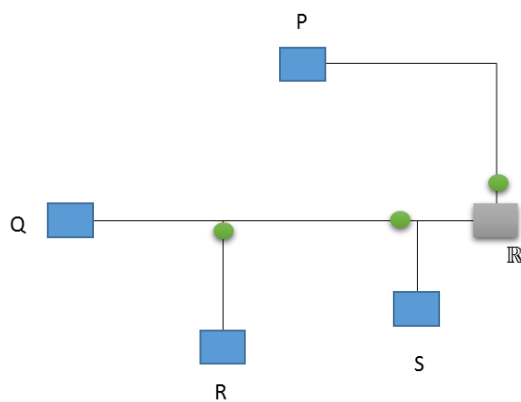


Figure 5. Initial delay resisted tree construction which consist of four sink P , Q , R and S and one root \mathbb{R}

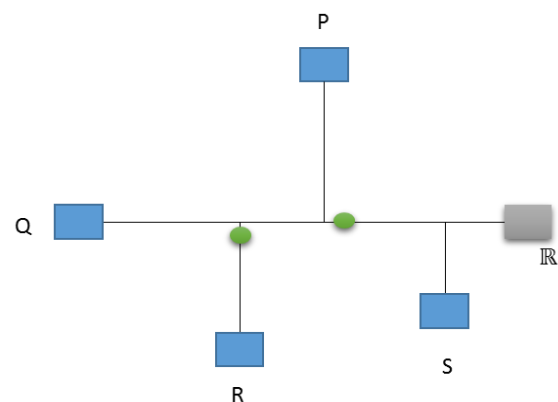


Figure 6. Reconstruction of delay optimized Tree which consist of four sink P , Q , R and S and one root \mathbb{R}

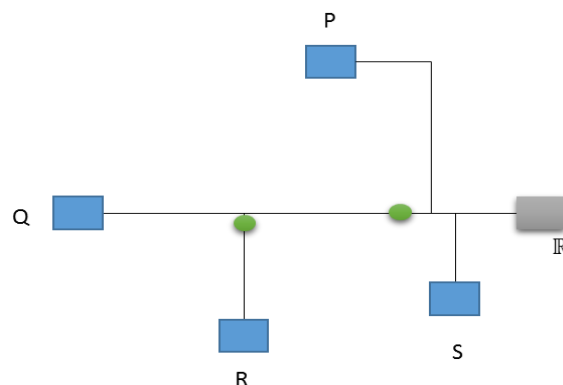


Figure 7. The delay optimized topology convergences which consist of four sink P , Q , R and S and one root \mathbb{R}

3.2. Delay Optimizer Aware Routing on Top of Obstacle

Firstly, the initial tree is created without considering any blockages. The initial tree might induce slew constraint even after insertion of delay optimizer and the tree could overlap over the blockage. Therefore to address it routing over the blockage inside the tree is considered. The objective of our model is to reduce delay and wire length. Therefore our objective function integrates slack and criticality. The objective function minimizes delay of delay resisted path and wire length in non-delay resisted path.

For routing over the blockage inside the tree construction under slack and slew constraint, $\forall \mathcal{R}_y^r \in \mathcal{R}^r$ we first need to identify the probable point. The selection of point is considered to be an optimization problem as follows

$$\min \sum_{y=1}^{|\mathcal{R}^r|} \sum_{l=1}^{|\mathcal{P}_y^r|} B_{yl} \mathcal{M}_{yl} (M_z(\mathcal{R}_y^r) M_y + \gamma) \tag{8}$$

$$\text{Such that } \left(A_{itr1}^r + \sum_{y=1}^{|\mathcal{R}^r|} \sum_{l=1}^{|\mathcal{P}_y^r|} B_{yl} C_{yl}^r \right)^2 + \left(A^r(Z^r) + \sum_{y=1}^{|\mathcal{R}^r|} \sum_{l=1}^{|\mathcal{P}_y^r|} B_{yl} C_{yl}^r \right)^2 \leq \text{slew}_{spec}^t{}^2 \tag{9}$$

$$\sum_{l=1}^{|\mathcal{P}_y^r|} B_{yl} = 1 \quad \forall y \in \{1, 2, 3, \dots, |\mathcal{R}^r|\} \tag{10}$$

where $B_{yl} \mathcal{M}_{yl}(\mathcal{R}_y^r)$ is the product of total downstream capacitance and resistance, which computes cumulative delay of all sink in downstream from C_{yl}^r . \mathcal{R}_y^r is the cumulative absolute param of negative slacks of sinks downstream from \mathcal{R}_y^r . $\mathcal{M}_y = \sum_{a_u} |\text{slack}(a_u)|$ is the delay resisted paths weights below \mathcal{R}_y^r . The weight γ identifies the solution with least computed wire length penalty on non-delay resisted path. To elude affecting delay resisted path, the γ is set very low with respect to $M_z(\mathcal{R}_y^r) M_y$. The objective function minimize changes on delay resisted path, thus satisfy slew constraint. Our model considers delay on delay resisted path and wire length on non-delay resisted path. The example of this is shown in Figure 8, 9 and 10.

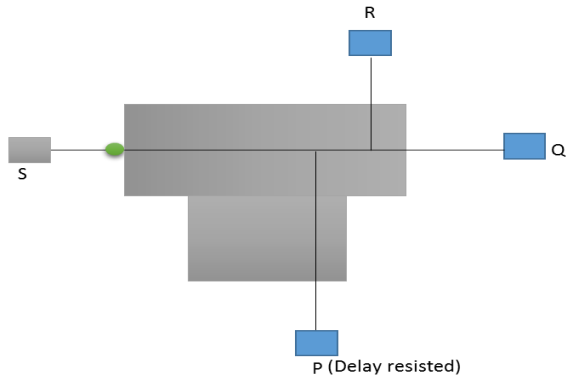


Figure 8. Initial tree creation using delay optimized RST with slew violation which consist of three sink P , Q and R and one root S

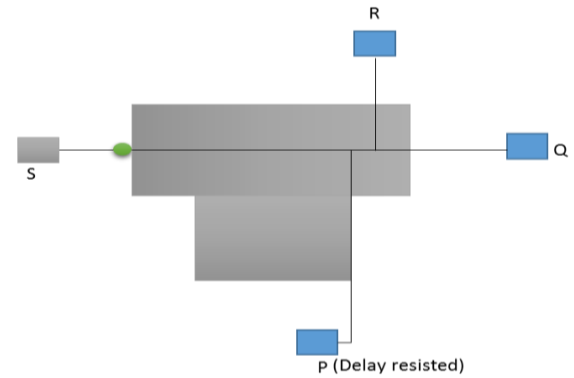


Figure 9. Tree creation using delay optimized RST that fixes slew violation with minimum wire length penalty which consist of three sink P , Q and R and one root S

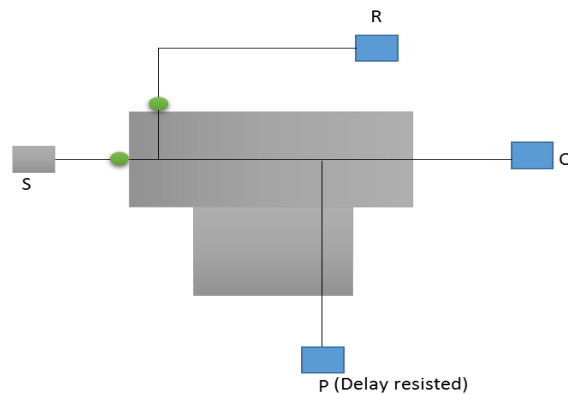


Figure 10. Tree creation using delay optimized RST that fixes slew violation and optimize delay on delay resisted path which consist of three sink P , Q and R and one root S

Figure 10 is preferred than Figure 9, since to satisfy slew constraint, Figure 9 reserves the timing of delay resisted sink by moving the escaping points on non-delay resisted path. The constraint (9) and (10), limits the cumulative slew reduction on \mathcal{R}_1^r has to be in position to pull $slew_1^r$ down below necessities and limits only one position selected for each escaping point respectively.

3.3. Minimization of Path Construction Computing Cost

Our model design permits optimizing topology to reduce delay on delay resisted path by extending wire with slew bounds. As the slew bound occurs below Steiner node, we extract the normalized design with two delay optimizer and one Steiner node as shown in Figure 11. Delay optimizer x_1 and x_2 stay right below for shielding and above the Steiner nodes respectively. We note that the phase driven by x_1 as $phase_1$, phase driven by x_2 as $phase_2$ and that above x_2 as $phase_0$. The elongation amount of g to use slew bound can be computed, due to availability of slew bound in $phase_1$. The distance among x_2 and \mathcal{W} is denoted as $g(x_2, \mathcal{W})$.

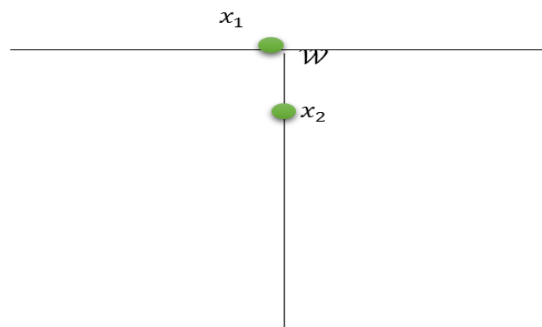


Figure 11. Shows the pattern of slew margin

Let consider an assumption that $g > g(x_2, \mathcal{W})$ and input capacitance of delay optimizer is within bound of wire capacitance, all slew constraint can be fulfilled by placing the Steiner node toward the position of x_2 and x_1 up to the position right beneath the new position of Steiner node. The Figure 12 shows assumption of negligible input capacitance of delay optimizer. The slew of $phase_1$ is within bound of $g > g(x_2, \mathcal{W})$ and the load and slew of $phase_0$ are not changed.

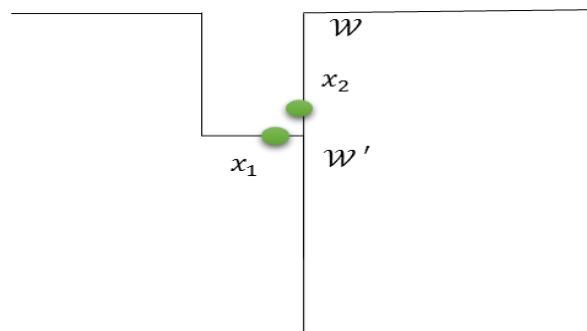


Figure 12. Optimization of delay optimizer considering input capacitance of delay optimizer negligible

Let consider another assumption that $g > g(x_2, \mathcal{W}) + M_x/m_c$ and input capacitance of delay optimizer is not within bound of wire capacitance, all slew constrained can be fulfilled by placing the Steiner node to M_x/m_c above x_1 and moving delay optimizer x_1 up to the position right beneath the new position of Steiner point (where m_c is the wire segment above x_2 init capacitance and M_x is the delay optimizer input capacitance). The Figure 13 shows the delay optimized topology after relocating the Steiner nodes \mathcal{W} to M_x/m_c above x_2 and delay optimizer relocation. The downstream capacitance for $phase_0$ is reduced in accordance to $M_x/m_c * m_c$ due o wire length above x_2 is reduced by M_x/m_c . The delay optimizer x_1 is

attached to $phase_0$ during path construction optimization, that includes M_x into downstream capacitance. As a result the cumulated downstream capacitance remains same for $phase_0$ and the quantity of downstream capacitance of $phase_2$ surges by $M_x/m_c * m_c$ as wire $\mathcal{W}''\mathcal{W}'$ is added beneath x_2 . The input capacitance of x_1 is detached from $phase_2$ where downstream capacitance is redeemed by M_x . Therefore the overall downstream capacitance beneath x_2 remains unchanged and slew of $phase_1$ satisfy slew constraint. The flow diagram of path construction is shown below in Figure 14.

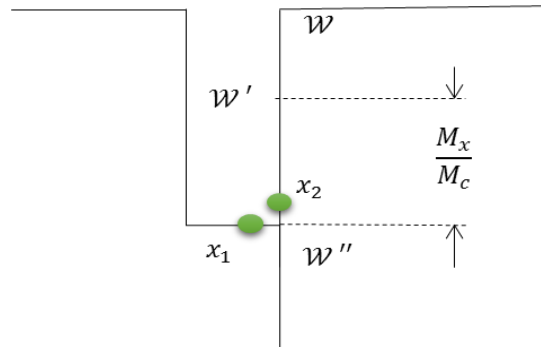


Figure 13. Optimization of delay optimizer considering without neglecting input capacitance of delay optimizer

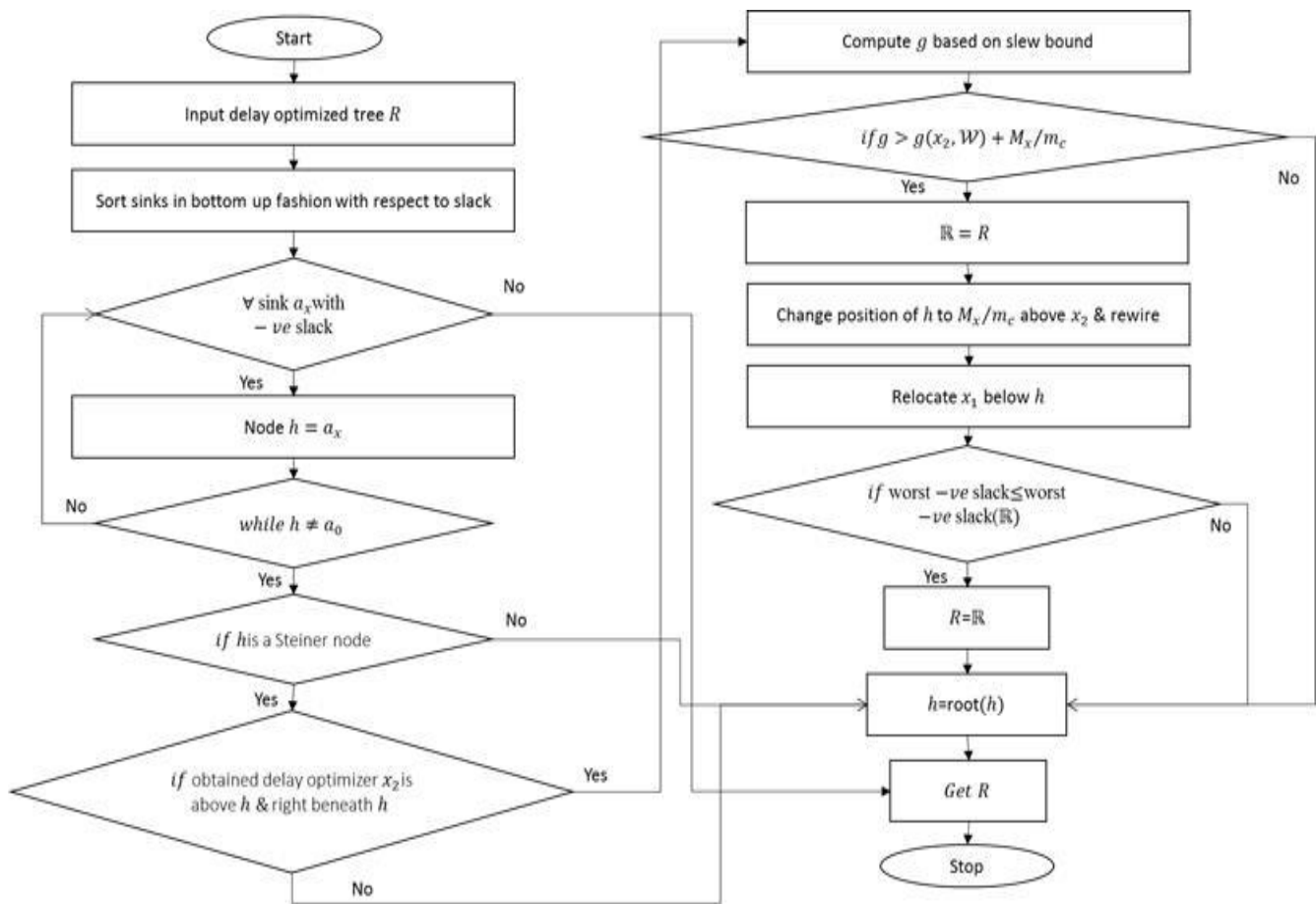


Figure 14. Flow diagram of proposed path construction

The proposed approach performance study is presented in next section.

4. SIMULATION RESULT AND ANNALYSIS

The OADORSMT algorithm is implemented C++ object oriented programming language. The GCC compiler is used to compile the code. The eclipse Kepler IDE used for running the algorithm. The system environment used to run the algorithm is Centos 7.0 Linux operating system, 3.2 Ghz, Intel I-5 Quad core processor and 16GB RAM. The experiment is conducted to evaluate the performance of proposed OAORSMT in term of wire length and worst negative slack performance. The results are compared with [11]. The same benchmark as in [11] has been used in our case study which is shown in Table 2. We have considered two case studies. The case 1 considers fixed obstacle benchmarks and case 2 consider varied obstacle.

4.1. Performance Evaluation Under Fixed Obstacle:

We had considered five case studies RC01-RC-05 as shown in Table 2, the pin size is varied from 10 to 100 and 10 obstacles are considered. The Table 3 shows Wire length performance of proposed OADORSMT and existing approaches. The Table 4 shows Worst negative slack performance of proposed OADORSMT and existing approaches. The experiment result shows performance improvement of OADORSMT over Existing approach [11] in term of wire length reduction and worst negative slack performance. An average reduction of 14.32% wire length reduction and 93.24% for worst negative slack is achieved by OADORSMT over existing approach.

Table 2. Benchmark Details for Fixed Obstacle

Benchmark Case	Number of Pin	Number of Obstacle
RC01	10	10
RC02	30	10
RC03	50	10
RC04	70	10
RC05	100	10

Table 3. Wire Length (WL) Performance for Fixed Osbtacle

Benchmark Case	Existing Approach Wire length (um) [11]	Proposed OADORSMT Wire length (um)
RC01	29140	25632.8
RC02	42970	40040.2
RC03	62270	49162.1
RC04	73870	64424.6
RC05	87320	77424.6
Average	59114	51336.86

Table 4. Worst Negative Slack (WNS) Performance for fixed Obstacle

Benchmark Case	Existing Approach WNS [11]	Proposed OADORSMT WNS
RC01	-214.56	-185.94
RC02	-4474.01	-48.5774
RC03	-847.29	0
RC04	-274.15	-238.215
RC05	-1186.67	0
Average	-1399.336	-94.54

4.2. Performance Evaluation Under Varied Obstacle

We had considered seven case studies RC06-RC-12 as shown in Table 5, the pin size is varied from 100 to 1000 and 100 to 10000 obstacles are considered. The Table 6 shows Wire length performance of proposed OADORSMT and existing approaches. The Table 7 shows Worst negative slack performance of proposed OADORSMT and existing approaches. The experiment result shows performance improvement of OADORSMT over Existing approach [11] in term of wire length reduction and worst negative slack performance. An average reduction of 20.48% wire length reduction and 97.28% for worst negative slack is achieved by OADORSMT over existing approach.

Table 5. Benchmark Details for Varied Obstacle

Benchmark Case	Number of Pin	Number of Obstacle
RC06	100	500
RC07	200	500
RC08	200	800
RC09	200	1000
RC10	500	100
RC11	1000	100
RC12	1000	10000

Table 6. Wire Length (WL) Performance for Varied Obstacle

Benchmark Case	Existing Approach Wire length (um) [11]	Proposed OADORSMT Wire length (um)
RC06	94742	83505.1
RC07	128875	119504
RC08	137116	124050
RC09	149274	136591
RC10	186030	190956
RC11	253039	259234
RC12	1297170	872277
Average	320892.2857	255159.5857

Table 7. Worst Negative Slack (WNS) Performance For Varied Obstacle

Benchmark Case	Existing Approach WNS [11]	Proposed OADORSMT WNS
RC06	-1551.37	0
RC07	-2239.39	0
RC08	1670.44	0
RC09	-3037.44	0
RC10	3695.56	0
RC11	-6749.56	0
RC12	-49733.9	-1571.32
Average	-8277.951429	-224.474

5. CONCLUSION

In this work we have carried out a research problems of OARSMT slew constraint in the presence of obstacle. This has enable has that circumstance that in modern VLSI design, the obstacle generally blocks the metal and the device layer as a result routing over the obstacle is a probable solution but does not permit placing of buffers on top of obstacles. The other issue of existing VLSI design is the presence obstacle considering long wire may cause signal violation constraint. To address this buffer insertion and slew constraint is done on the inter-connect over obstacle but the delay parameter is neglected which result in routing overhead. To address this, this work presented Obstacle aware Delay Optimized Rectilinear Steiner Routing to reduce the routing resource overhead. The experiment are carried out for small and longer net to evaluate the performance of OADORSMT in term of wire length and worst negative slack over existing approach. Two case studies have been considered. For case 1, the obstacle are fixed, an average improvement of 14.32% wire length reduction and 93.24% for worst negative slack is achieved by OADORSMT over existing approach. For case 1, the obstacle are fixed, an average improvement of 20.48% wire length reduction and 97.28% for worst negative slack is achieved by OADORSMT over existing approach. The outcome achieved show the proposed OADORSMT model is efficient in term of wire length reduction and worst negative slack performance. The future work would consider evaluating proposed model considering various benchmarks and consider optimizing the proposed algorithm in order to evaluate under parallel computing model and analyses its parallel efficiency.

REFERENCES

- [1] Garey M. R. and Johnson D. S., "The rectilinear Steiner tree problem is NP-complete," *SIAM Journal on Applied Mathematics*, vol. 32, pp. 826-834, 1977.
- [2] C. J. Alpert, et al., "Buffer Insertion for Noise and Delay Optimization," *IEEE Trans. on Comput.-Aided Des. Integr. Circuits Syst.*, vol. 18, pp. 1633-1645, 1999.
- [3] G. Ajwani, et al., "FOARS: FLUTE Based Obstacle-Avoiding Rectilinear Steiner Tree Construction," *Proc. ISPD*, 2010.
- [4] T. Huang and Evangeline F. Y. Y., "An Exact Algorithm for the construction of Rectilinear Steiner Minimum Trees among Complex Obstacles," *Proc. DAC*, 2011.
- [5] L. Li, et al., "Generation of Optimal Obstacle-avoiding Rectilinear Steiner Minimum Tree," *Proc. ICCAD*, 2009.
- [6] L. Li and Evangeline F. Y. Y., "Obstacle-avoiding Rectilinear Steiner Tree Construction," *Proc. ICCAD*, 2008.
- [7] Huang T., et al., "On the construction of optimal obstacle-avoiding rectilinear Steiner minimum trees," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions*, vol. 30, pp. 718-731, 2011.
- [8] C. H. Liu, et al., "Obstacle-Avoiding Rectilinear Steiner Tree Construction: A Steiner-Point-Based Algorithm," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, pp. 1050-1060, Jul 2012.
- [9] S. Hu, et al., "Fast algorithm for slew constrained minimum cost buffering," *Proc. Design Automation Conf.*, pp. 308-313, 2006.
- [10] Y. H. Lin, et al., "Critical-Trunk-Based Obstacle-Avoiding Rectilinear Steiner Tree Routings and Buffer Insertion for Delay and Slack Optimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 30, pp. 1335-1348, Sep 2011.
- [11] Zhang H. and Ye D. Y., "Key-node-based local search discrete artificial bee colony algorithm for obstacle-avoiding rectilinear Steiner tree construction," *Neural Computing and Applications*, vol. 26, pp. 875-898, 2015.
- [12] H. Zhang, et al., "A heuristic for constructing a rectilinear Steiner tree by reusing routing resources over obstacles," *Integration, the VLSI Journal*, 2016.
- [13] Y. Cai, et al., "Obstacle-Avoiding and Slew-Constrained Clock Tree Synthesis With Efficient Buffer Insertion," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23, pp. 142-155, Jan 2015.
- [14] C. H. Liu, et al., "Efficient Multilayer Obstacle-Avoiding Rectilinear Steiner Tree Construction Based on Geometric Reduction," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, pp. 1928-1941, Dec 2014.
- [15] Held S. and Spirkel S. T., "A fast algorithm for rectilinear Steiner trees with length restrictions on obstacles," *Proceedings of the 2014 on International symposium on physical design. ACM*, pp. 37-44, 2014.

-
- [16] Huang T. and Young E. F. Y., "Construction of rectilinear Steiner minimum trees with slew constraints over obstacles," *Proceedings of the International Conference on Computer-Aided Design. ACM*, pp. 144-151, 2012.
 - [17] Chu C. and Wong Y. C., "FLUTE: Fast lookup table based rectilinear steiner minimal tree algorithm for VLSI design," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 27, pp. 70-83, 2008.
 - [18] Kashyap C. V., et al., "PERI: a technique for extending delay and slew metrics to ramp inputs," *Proceedings of the 8th ACM/IEEE international workshop on Timing issues in the specification and synthesis of digital systems. ACM*, pp. 57-62, 2002.