Optimization Of Link-Based Non-Tree Clock Network Using An Efficient Algorithm

Aswathy Vijayan
P.G Scholar,
SNS College of Technology,Coimbatore

A. Sridevi
Associate Professor,
SNS College of Technology, Coimbatore

Abstract

The Clock network is a vulnerable victim of variations as well as a main power consumer in many integrated circuits. Clock skew is becoming increasingly difficult to control due to variations. Recently, link-based non-tree clock network[1] attracts people's attention due to its appealing trade off between variation tolerance and power overhead. Link based non-tree clock distribution is a cost-effective technique for reducing clock skew variations. Existing non-tree clock routing methods[1] have a few drawbacks. The length of the link is long, link distribution is uneven and is also very complex to design and it does not consider about the delay and skew variations. In this paper, a new sensitivity based algorithm is proposed to overcome these drawbacks. The algorithm is implemented in VHDL language using Altera Quartus II and its effectiveness is validated using SPICE based simulation. Experimental results show that the new algorithm is able to achieve better skew reduction with an average of 5% wire length increase.

1. Introduction

In synchronous VLSI designs, the quality of clock network is vitally important, since almost every data transfer are coordinated by clock signal. The clock network suffers from two simultaneous challenges: variability and power. Clock skew, which is the difference between clock signal delays, is very sensitive to variations. Clock network is also a large power consumer because of its large fan out size and high switching frequency. To address these problems, several solutions like variation aware clock tree routing, buffer/wire sizing, non-tree clock distribution have been proposed. The most effective among these methods is non-tree clock routing because of the existence of multiple paths from clock source to the sinks, which makes the delay in the sinks correlated, resulting in reduced skew variation.

There are several types of non-tree clock distribution network methodologies that have been proposed. These non-tree networks are classified mainly as 1-dimensional and 2-dimensional structures. In a 1-dimensional approach, several clock sinks are attached to single piece of connection.This connection will be driven at multiple points from a binary tree and so skew between any two sinks attached to this interconnect will be small. An example of 1-dimensional non-tree clock network[8,9] is given in Figure.1(a). A key limitation of the 1-dimensional structure is that it does not handle the skew variation between different 1-dimensional regions.

The 2-dimensional non-tree structure[10-12] is also called mesh. Depending upon the location of mesh, from clock source to sink, the mesh structure can be further classified as leaf level, top level and multi level mesh. In leaf level (Figure. 1(b)), the mesh is close to clock sinks. In top level (Figure. 1(c)), the mesh is close to clock source and in multi level (Figure. 1(d)), several meshes are at different levels. In the leaf level mesh approach, a metal wire mesh is overlaid on the entire chip area and driven at multiple points directly from the clock source or through a routing tree. Each clock sink is connected to the nearest point on the mesh. This technique is very effective in suppressing skew variations in microprocessor designs. A leaf-level mesh[10,11] normally consumes enormous wire resources and also consumes larger power. It is also hard to integrate with clock gating, which is a common low-power technique. Therefore, its application is restricted to high-end products such as microprocessors. In contrast, a top-level mesh[12] may consume less wire and power. In the top-level mesh approach, the clock source drives a coarse mesh directly and clock sub-trees are attached to the mesh. The mesh structure is having negligible skew variations, but skew variations within each sub-tree still exist. Most recently, a multilevel mesh[13] approach is

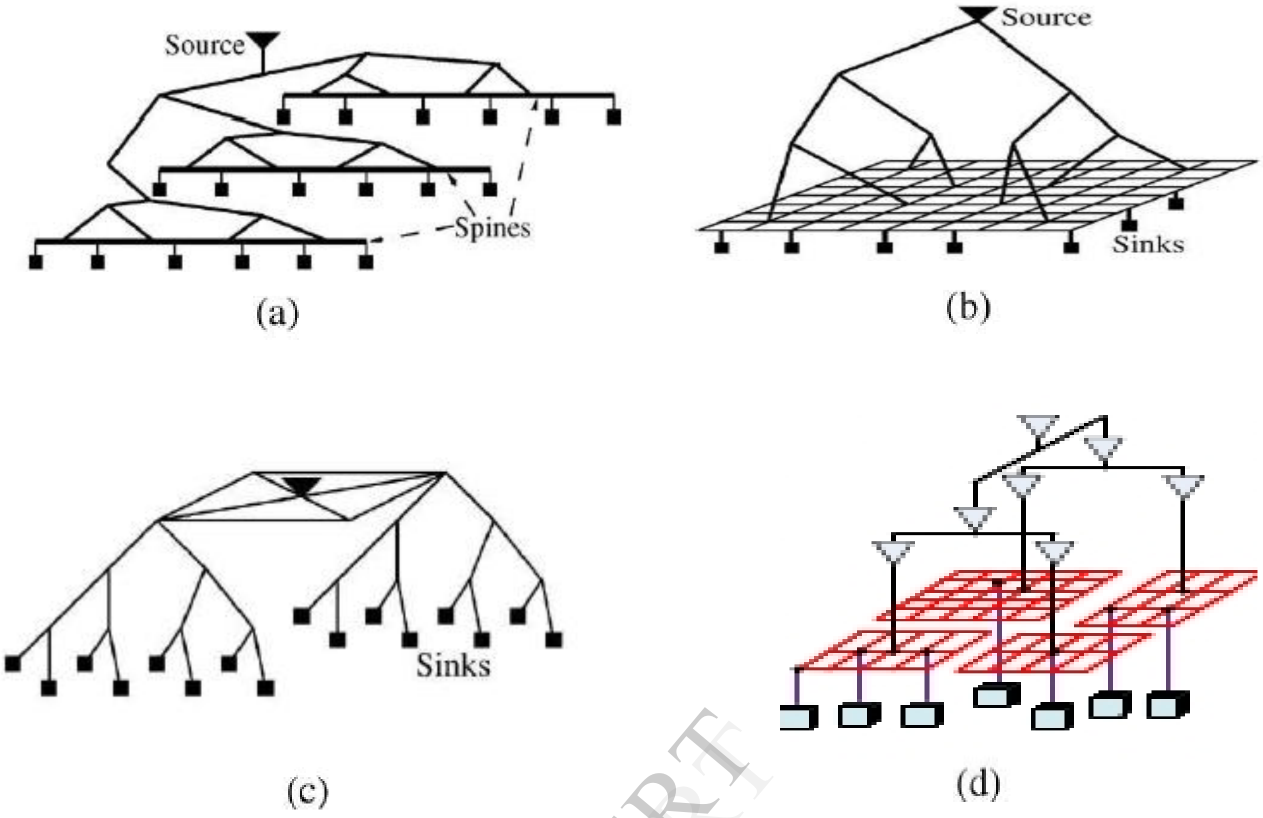

Figure.1. Non-Tree Clock Distribution Network: (a) 1-Dimensional; (b)Leaf Level Mesh; (c)Top Level Mesh; (d)Multi Level Mesh.

also proposed. But this mesh structure is also having larger wire/power overhead.

Recently, a link-based non-tree algorithm has been proposed in which cross links are added in an existing clock tree. By suitably choosing the correct locations of the links, significant reduction in skew variability can be achieved with very small increase in wire length when compared to the original clock tree. A link-based non-tree is shown in Figure. 2. The vital aspect of the link-based methodology is to determine the proper location of the cross links, for which some algorithms were proposed. While these algorithms are highly effective, they from several drawbacks. Some of these are: high complexity, lengthy links that might cause routing problems and uneven distribution of links across all regions of the clock tree that might limit skew variability reduction. The existing algorithms doesn't consider about the delay or skew variation information. So, a fast link insertion algorithm called delay sensitivity based algorithm is proposed in which both delay and skew variations are considered.

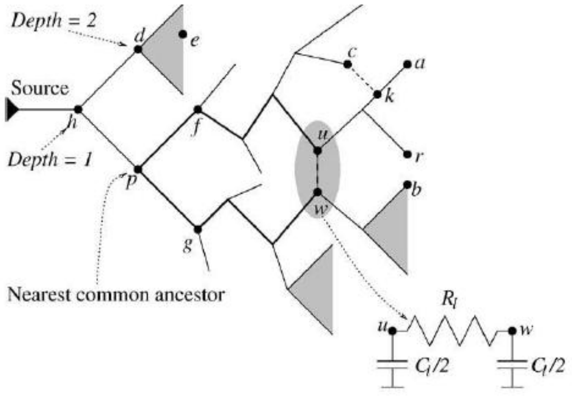

Figure.2. Crosslink Insertion

2. Effect of Link Insertion

In this section, we will review a link-based non-tree clock distribution network. Delay and skew variation of a non-tree *RC* network will be analyzed. An Elmore delay model is employed because of its high fidelity[14] and ease of computation.

2.1.Delay in RC Network

A non-tree *RC* network can be represented by a graph *G*=(*V*,*E*). The graph *G* consists of a node set *V* composed of a source node, several sinks and internal nodes. In an *RC* network, the Elmore delay at node *i* is given by

$$t_i = \sum_j R_{i,j} C_j \qquad (1)$$

where C_j is the ground capacitance at a node j in V . The transfer resistance $R_{i,j}$, is equal to the voltage at node *i* when 1A current is injected into node *j* and all other node capacitors are zero. The *RC* network can be decomposed to a spanning tree *T*, where *T*= (*V*,E_T), having a set of link edges *El* such that $E = E_T \cup E_l$. As an alternative approach which is easier for analysis, the delay from the source node to each node can be evaluated by starting with delays in the tree *T* and then incrementally inserting every link edge in E_l.

In Fig. 2, a network is indicated by the solid lines and a crosslink is inserted between node *u* and *w*. If the link has a wire resistance of R_l and wire capacitance of C_l, the link insertion can be decomposed by inserting a resistor of R_l between *u* and *w* and adding a capacitor $C_l/2$ of at node *u* and *w*. Adding link capacitors will not change the network topology and so, its effect can be estimated easily. If the Elmore delay before the link insertion, from the source to any sink *i* is t_i; then the delay $\widetilde{t_i}$ after only adding the link capacitors is given by

$$\widetilde{t_i} = t_i + \frac{C_l}{2}\left(R_{i,u} + R_{i,w}\right) \qquad (2)$$

The impact of the link resistance *Rl* can be analyzed by using the technique by Chan and Karplus. The delay at node *i* is changed from $\widetilde{t_i}$ to $\widehat{t_i}$ given by

$$\widehat{t_i} = \widetilde{t_i} - \frac{\widetilde{t_u} - \widetilde{t_w}}{R_l + r_u - r_w} r_i \qquad (3)$$

where r_i, r_u and r_w are equal to the Elmore delay at *i*, *u*, and *w*, respectively, when C_u= 1, C_w = −1 and the other node capacitances are zero.

2.2 Skew Variability Between Link Endpoints

If a link is inserted between nodes *u* and *w*, then according to, the skew between *u* and *w* after link insertion is given by:

$$\widehat{q_{u,w}} = \frac{R_l}{R_l + r_u + r_w}\left(q_{u,w} + \frac{Cl}{2}\left(R_{u,u} - R_{w,w}\right)\right) \quad (4)$$

where, $q_{u,w} = t_u - t_w$ is the original skew between nodes *u* and *w*, $\widehat{q_{u,w}}$ is the final skew after the link insertion, C_l is the link capacitance, R_l the link resistance, $R_{u,u}$ and $R_{w,w}$ the transfer resistances of nodes *u* and *w*. Since the effect of capacitance can be easily estimated and removed , equation (4) gets reduced to:

$$\widehat{q_{u,w}} = \frac{R_l}{R_l + r_u + r_w} q_{u,w} \qquad (5)$$

Thus, from equation (5), we can see that the final skew is a scaled value of the original skew with the scaling factor of $\frac{R_l}{R_l + r_u + r_w}$. The scaling factor is always less than 1, therefore, the skew between nodes *u* and *w* is always reduced as a result of link insertion. It has also been proved that inserting a link as close to sink nodes as possible is better in terms of skew variability reduction. For example, in Figure.2, when we want to reduce the skew between nodes *r* and *b*, then it is better for the link to be closer to the nodes *r* and *b* as possible.

3. Existing Algorithms Overview

3.1.Rule Based Algorithms: In the rule-based link insertion, all the possible links are characterized by parameters denoted by *α*, *β* , *γ* and *δ*. The definitions of *α* and *β* are the same as in equation 2.In simple terms, the rule-based methods involve empirically selecting appropriate values of bounds *αmax*, *βmax* , *γmax* and *δmax* and adding only the links that have *α*, *β*, *γ* and *δ* values less than chosen bounds.

3.2.MST Based Algorithm: In MST based algorithm, the minimum rectilinear distance between any two nodes is calculated. By using this algorithm crowding of links between any two nodes can also be avoided. One problem which is possible in constructing a complete MST is that the total wire length might become very high. To avoid this problem, we remove links from the selected MST edges based on *α* and *β* rules used in the rule based node pair selection algorithms.

3.3.Drawbacks of Existing Algorithms:

1 The existing algorithms use parameters such as *α* , *β*, *αmax* and *βmax*. So it makes the user to select the values of the parameters.

2 The existing algorithms do not give any importance for delay or skew variation information.

3 The current methods are not compatible with the use of higher order delay models because

all the parameters used for selecting the links are based on nominal Elmore delays.

4. Sensitivity Based Algorithm

In this section, we propose the delay sensitivity based algorithm that addresses all the drawbacks of the existing algorithms. The basic idea behind sensitivity based link insertion is to insert links between the node pairs that are most susceptible to variation. This is motivated by the fact that a given link has the maximum beneficial effect on the node pairs between which it is inserted. The rest of this section presents the details of the algorithm. First, we introduce the concept of delay sensitivity vector for a given process variation model. This concept helps us to identify the link pairs that are the most susceptible to the variation effects. Following this, we prove that the *α*-rule proposed for un buffered clock trees is also true for buffered clock tree case. Finally, we present our sensitivity based link selection method. The effectiveness of the sensitivity based link insertion algorithm has to be evaluated using VHDL language.Implement the algorithm on clock tree network. Perform the experiments on certain benchmark circuits and compare the results with our previous work.

The sensitivity based algorithm includes the following steps:

1 Decompose the clock tree network into left and right sub treesT_l, T_r.

2 For every pair ($T_{l,i}, T_{r,j}$), find out the weight. Weight= minimum rectilinear distance between leaf pairs using MST based algorithm.

3 Calculate delay sensitivity for each independent segment of the clock tree network.

$$S_{i,j} = \frac{T_i - T_j}{x_k}$$

4 Generate sensitivity vector for node pair and obtain its magnitude, Mi,j.

$$M_{i,j} = \sqrt{\left(\frac{T_i - T_j}{x_k}\right)^2}$$

5 Evaluate the value of the Sensitivity Cost. Sensitivity Costi,j = (1 − α)Mi,j

$$\alpha \leq \frac{R_l}{R_{loop}} \leq \alpha_{max}$$

6 Select the node pairs between which the cross links has to be inserted based on the value of sensitivity cost and α value.

We can conclude that the effectiveness of a link between nodes *i* and *j* is proportional to *Mi,j* and (1-*α*). Any node pair with large value of sensitivity cost can be regarded as a good candidate for link insertion. This is because if *Mi,j* is high, it means that the sink pair *i,j* is very sensitive to variations. Similarly, if the value of *α* is low (or a high value of 1-*α*), the link can significantly reduce the final skew. Thus, we use the *Sensitivity Cost* as a measure of effectiveness of each link.

We must consider a set of guidelines before the insertion of cross links. These guidelines are listed below:

1 Links are always inserted between the nodes with zero nominal skew.

2 Node pairs are selected in such a way that the buffers driving the links are never overloaded. So, the selected sink nodes must be relatively close to each other. This also makes sure that the links are not concentrated in a single place.

3 Since the magnitude of the sensitivity vector is higher for sink pairs that do not share any common element, the method inherently selects pairs with considerably different source to sink paths.

4 Short circuit risk in multi-driver nets is avoided by this method.

4.1.Advantages of Sensitivity Based Algorithm

The sensitivity based link insertion algorithm has the following advantages when compared to the other existing algorithms:

• Because of the use of delay sensitivity, this method finds the sink pairs that are most sensitive to the effects of variations.
• Links insertion can be modified based on a given variation model. This method directly uses the variation information in the form of delay sensitivity, so the links selected will be more effective.
• All the sensitivity information are obtained in a one-time process, our overall approach is much faster than the existing approach.
• The calculation of delay sensitivity is compatible with higher order models including SPICE. Hence our algorithm can be applied even when high accuracy is required.
• Our algorithm does not involve the use of any empirical parameters such as *α*, *β*, *γ*, etc.

Thus, the sensitivity based algorithm is having all the features of a good algorithm.

5. Experimental Results

The algorithm is implemented in VHDL language using Altera Quartus II and the effectiveness of the algorithm is validated using SPICE based simulation.

The tree network can be either considered as an array or as a matrix. Each elements of the matrix can be considered as nodes of the tree network. The resistance and capacitance values (R_l, C_l) of the link are given as inputs. All the delay values are also given.

The VHDL file can be made from Altera Quartus II and in which the minimum rectilinear distance between each and every node is calculated. The minimum rectilinear distance represents the edge weight. The delay sensitivity vector is calculated using the delay values of each sink segments. Based on the value of α and sensitivity cost, select the nodes that are most sensitive to variation. The cross links has to be inserted between those selected nodes.

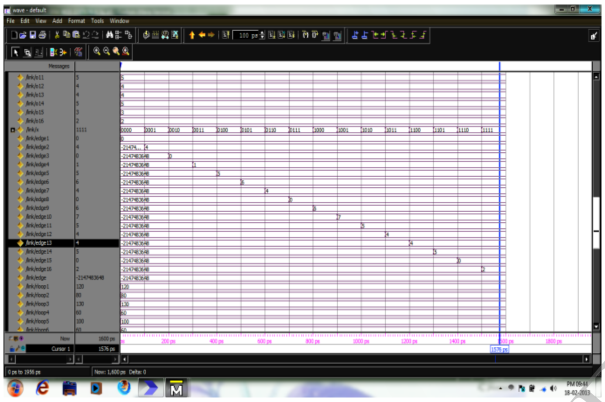

Figure 3 . Output Waveform for the Selection of Nodes Pairs

The algorithm implemented in Altera Quartus II is simulated in Modelsim. Figure 3 shows the output waveform for the selection of node pairs. Here comparison of each edge weight values is shown. Those values that are obtained as zero are most sensitive to variation and between those node pairs we insert a cross link.

After finding the nodes, the cross links are inserted. The cross links are a resistor with two capacitors. The clock tree network is simulated in SPICE. The experimental results show that the clock tree network with link insertion is having lesser delay and wire length when compared with the clock network without link. In order to verify the effectiveness of the sensitivity based link insertion, we compare our results with that of the existing works. The experiments are performed on benchmark circuits and compared with our previous work. The skew reduction in this method is lesser than the earlier works. The RTL schematic for the algorithm is shown in Figure 4.

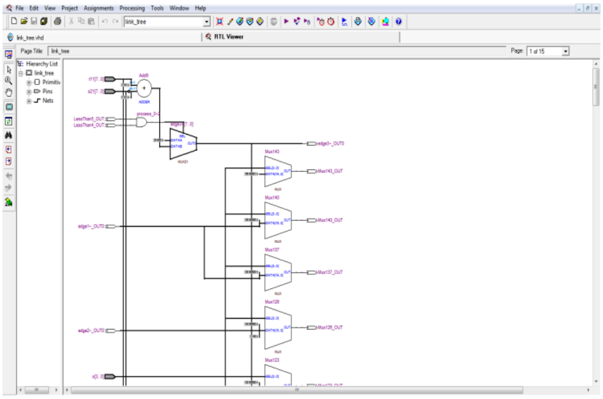

Figure 5. RTL Schematic for the algorithm

The Table I below shows, the effect of link insertion on skew variation or delay variation and power for different clock distribution networks like clock tree, clock mesh networks (leaf level, top level and multi level mesh), clock tree with cross links using Rule based method(Link-R), MST based method(Link-MST) and Sensitivity based algorithm(Link-S). The power delay product of cross link insertion using sensitivity based algorithm is much less when compared to other link insertion techniques.

TABLE I
POWER CONSUMPTION AND TIME DELAY FOR DIFFERENT LINK INSERTION TECHNIQUE

Method	Power consumption (mv)	Time Delay (S)
Clock tree	50.05	2.27
Top-level mesh	46.51	1.75
Leaf-level mesh	44.05	1.80
Multi-level mesh	47.75	1.79
Link- R	49.12	1.44
Link-MST	48.12	1.36
Link-S	48.05	1.18

6. Conclusion

A Sensitivity based link insertion algorithm is used for inserting cross links in a clock tree network. It can overcome the drawbacks of the existing algorithm of lengthy links and increased skew variations. The effectiveness of the entire algorithm is validated using VHDL in Quartus ll. The new algorithm is able to achieve comparable or better skew variability reduction with significantly lower wire length and power. The results shows that the links added by the proposed algorithm are shorter on average and thereby making the non tree routable.

7. Acknowledgement

We extend our heart-felt gratitude to the management of SNS College of Technology, and to our Principal Dr.V.P.Arunachalam for providing me with all sorts of supports in completion of this project. We are thankful to Dr. P.Ramamoorthi**,** Dean, Department of Electronics and Communication Engineering for his sheer encouragement for the completion of this project. We take immense pleasure in expressing our humble note of gratitude to Prof.S.Arumugam, Professor and Head, Department of Electronics and Communication Engineering, for his remarkable guidance and besides his positive approach had offered incessant help in all possible ways from the beginning.

8. References

[1] Zamini. M.S; Taajobian, M.; Saeedi, M; "An Efficient Non-Tree Clock Routing Algorithm For Reducing Delay Uncertainity", in *Proceedings of the ACM/IEEE DAC*, San Diego, CA, June 2008, pages 558-565

[2] Samanta, R.; Jiang Hu; Peng Li "Discrete Buffer And Wire Sizing For a Link Based Non-Tree Clock Network" ,Very Large Scale Integration (VLSI) Systems, IEEE Transactions on July 2010, pages 1025 - 1035

[3] Rajaram, J. Hu, and R. Mahapatra. "Reducing clock skew variability via cross links," in *Proceedings of the ACM/IEEE DAC*, San Diego, CA, June 2004, pages 18–23.

[4] G. Venkataraman, N. Jayakumar, J. Hu, P. Li, S. Khatri, A. Rajaram,P. McGuinness, and C. J. Alpert, "Practical techniques to reduce skew and its variations in buffered clock networks," in *Proc. IEEE/ACM Int.Conf. Comput.-Aided Des., Nov. 2005*

[5] A. Rajaram, D. Pan, and J. Hu, "Improved algorithms for link based non-tree clock network for skew variability reduction," in *Proc. ACMInt. Symp. Phys. Des., Apr. 2005*

[6] A. Rajaram and D. Z. Pan, "Fast incremental link insertion in clock networks for skew variability reduction," in *Proc. IEEE Int. Symp. Quality Electron. Des., Mar. 2006*

[7] S. R. Nassif, "Modeling and analysis of manufacturing variations," in *Proceedings of the IEEE CICC*, San Diego, CA, May 2001, pp. 223–228.

[8] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser, "Clock skew verification in the presence of IR-drop in the power distribution network," in *IEEE Transactions on CAD*, vol.19, no.6, pp.635–644, June 2000.

[9] W.-C. D. Lam, C.-K. Koh, and C.-W. A. Tsao, "Power supply noise suppression via clock skew scheduling," in *Proceedings of the IEEE ISQED*, San Jose, CA, March 2002, pp. 355–360.

[10] B. Lu, J. Hu, G. Ellis, H. Su, "Process variation aware clock tree routing," in *Proceedings of the ISPD*, Monterey, CA, April 2003, pp. 174–181.