International Journal of Advanced Science and Technology
Vol. 29, No. 6, (2020), pp. 2570 - 2778

Optimized Global Routing Techniques for Efficient VLSI Physical Layout Design

[1]Dr.T.Rajendran[*], [2]O.Vignesh

[1]*Associate Professor, Department of Information Technology, QIS College of Engineering and Technology, Ongole, India.*

[2]*Assistant Professor, Department of Electronics and Communication Engineering, QIS College of Engineering and Technology, Ongole, India.*
[1]*tlrajen@gmail.com* ,* [2] *vicky6058@gmail.com*

Abstract

Physical Layout design is the major critical task in efficient ASIC design. According to Moore's law sizing of integrator circuits (ICs) are becoming very compact. The trending of VLSI system design moves to System on Chip (SoC) concepts. Backend design steps like partitioning, placement, floor planning and routing are plays vital role in IC design. Routing is the most wanted part to design optimized SoC for VLSI implementation. There are lot of methods and techniques proposed to solve the routing problems. These methods are deals with the speed of IC routing, efficient routing and optimization of wire length. Here the study of various current algorithms in routing to understand, how they are able to meet those challenges.

Keywords: *VLSI, SoC, ASIC, Routing, Physical Layout Design and Optimized IC design.*

1. Introduction

Scaling of the IC design is used to fix the chip size and IC standard cells structure. In SoC design routing delay and it's inter connect complexity is producing more critical delay compare to standard cell gate delay of the SoC architecture. So the complex routing design can affects the particular part of IC design and reduce the overall performance of the correctly routed area of that IC design. Global Routing is a vital step in physical implementation of ASIC designs. It can be defined as a rough connection between the nets of the placed designs. Exact connections are established during the detailed routing phase which is not discussed in this paper. Global routing is performed after efficient placement of standard physical cells of the layout. There have been ISPD [1] routing contests such as ISPD 2007 and ISPD 2008 in which routers from academia were invited for a contest on some of the difficult benchmarks placed by state of the art placers. NTHU emerged as a winner of ISPD 2008 routing contest. Some of the state of the art routing tools and algorithms are discussed in detail [2-16]. The remaining part of the paper is organized as follows, In Section 2, we describe some of the commonly used routing techniques in modern routers. In the Section 3, we describe the algorithms involved in router NTHU [9]. In Section 4, we describe the implementation details of FastRoute [2], [3], [4]. In Section 5, we describe the tool NTUgr [15] and last section is conclusion.

2. Commonly used Routing Techniques

Based on various design methods the routings techniques are designed and optimized the timing complexity. Global routing creates the connections between the sub-blocks of SoC design. In this section, detailed survey taken to solve the routing delay issues for SoC design. Static power of IC design is depends the routing delay and its timing complexity.

2.1. Decomposition of Multiple-pin Net

This technique is used by a number of pin-net in routing tools. In this technique, several two-pin nets are decomposed from the multi-pin net of main block or sub-block pins. After the two pin division the processing of two-pin nets becomes easier during global routing. Decomposition methods are classifies two techniques, which are Minimum Spanning tree (MST) and Rectilinear Steiner minimal tree (RSMT) techniques.

2.2. Maze Routing

Maze Routing [18] is one the earliest routing methods which explore all possible routing paths between sources and sink by using shortest path algorithms such as Dijkstra's algorithm or breadth first search algorithm. Using shortest path algorithm, maze routing finds the route with the lowest cost. To improve the efficiency of shortest path search, bounds to the search are applied using A*-search [19]. By doing this, some of the grid points are excluded from the shortest path search.

2.3. Multi-sink and Multi Source Routing

This method [3] is an improvement over maze routing. In maze routing, since it obtains paths from source to sink, it happens that some of the better routes, which are from the subtrees of the multi-source net, are not considered. In multi-sink and multi-source maze routing, some of the congested areas can be avoided and better paths can be obtained.

2.4. Pattern Routing

Pattern routing is the best method [20] and it routes a net with two pins with already defined patterns. It is two different shapes which are named as Z-shaped and L-shaped patterns. This method is more optimized and fast compared to maze routing. However, it has drawback that solution quality may degrade as all the possible routes for the two pin nets may not be considered.

2.5. Routing Monotonically

This method, [3], the routing monotonically is used based on the source of the design. It is optimized design when the (source pin net) two pin nets are placed left and bottom of the sink. Bounding box is used to interconnect between complex pin packed blocks. The search space of the area between two blocks is based on wire length of a two- pin net and it has to search for (m+ní2)! / (mí1)!(ní1)! paths on a m × n grid graph. This method is used to improve the search space of blocks, thereby achieving better quality. Runtime complexity of these methods remains the same as that of pattern routing for a Z-shape.

2.6. RIP and Re-routing

This method is used by Chi Dispersion [21], Labyrinth [20], Fashion [8], FastRoute [2], [4] and BoxRouter [10]. This method iteratively improves the quality of solution. In this method routing is performed without considering the congestion of the region. Once all the nets are routed, a congestion map is obtained. The routes passing through most congested regions are ripped to be re-routed again with better paths. In this method, the solution quality is affected by the sequence of routes processed.

2.7. Negotiated Congestion Routing

This method [22], was proposed to even the two objectives of minimization of the IC timing performance issue problems which is affected from the critical paths in the routed area and elimination of congestion. PathFinder [22], was the first router which implemented this technique. In this method, cost function of routing an edge is defined as Equation 1. We have taken this equation from [9].

$$cost_e = (b_e + h_e) \times p_e \qquad (1)$$

where be is the basic cost of routing an edge e, pe denotes the number of other nets presently going through e in the current count of iteration, and he denotes the history of congestion for e during previous iterations. If overflows are present in e during previous iterations, he discourages router tool to route through e, and preserves the routing resources of e for nets with less routing choices. he is updated as shown in Equation 4.

$$h_e^{i+1} = \begin{cases} h_e^i + k & \text{if e has overflow} \\ h_e^i & \text{otherwise} \end{cases} \qquad (2)$$

where k is a parameter defined by user and i is the iteration count and. Equation 1 and 4 are the basic equations for negotiating congestion driven routing and are used by most of the modern global routing tools.

2.8. Assignment of Layer

Layer assignment is an important step as it maps 2D routing topology to multilayer solution space. This is performed by ILP or dynamic programming [4]. The objective of assignment of layers is to minimize the number of vias used and also for preserving the overflow obtained from 2D routing.

3. NTHU Routing

Routing plane consists of multiple layer formation which is used to project on IC routed area and then FLUTE [23] is used to break down the major multiple-pin net into of two-pin nets set. Then, NTHU-Route performs the routing of every two pin net using two probabilistic L-shaped patterns. If the edges are along a probabilistic route, half demand is added to it, and if the edge is along a straight route, full demand (1) is added to it. Then, by using edge shifting technique, NTHU-Route updates the topology of every multi-pin net and routes every two pin net with the L-shaped pattern with least cost.

3.1. Congestion Region Identification based Rip-up and Reroute

When rip-up and re-route methods are used, the ordering of nets to be ripped and re-routed is important because it affects the quality of re-routed nets. In practice, small nets (nets having small bounding box) are routed before large nets (nets having large bounding box) as they have lesser flexibility. Overflow free paths are present in the routes that have more flexibility. As this strategy does not have consideration on the current congested regions, there are still ways to improve its solution quality. For example, assuming that there is a set Ns of smaller nets and a set Nl of larger nets situated in a region which is congested of routes. Based on the above method, the nets in Ns will be considered for routing first. After that, it will

not be easy to estimate the paths free from overflow in the region for the set Nl. This is due to the fact that the region will be occupied with most edges located in the nearby region which were routed earlier. To prevent this from happening, NTHU-Route at first makes use of a different method to identify congested regions, and perform rip-up and re-route of the edges that are situated in the congested region. The method begins with an initial solution and it works in the following manner.

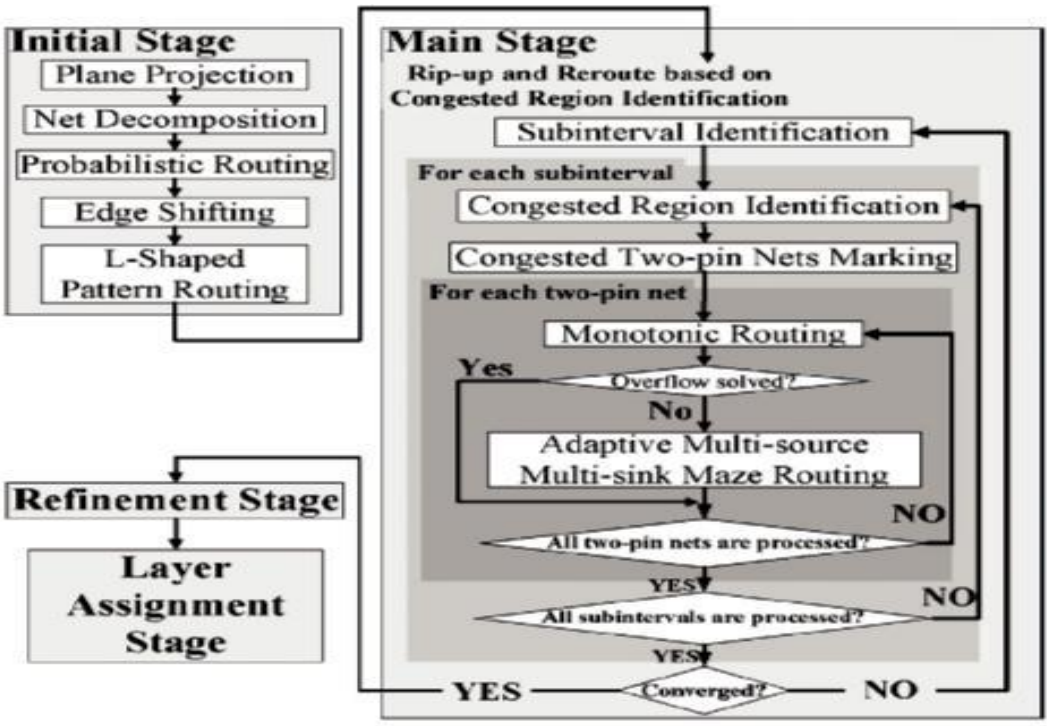

Fig. 1. Flow of NTHU Route

At first, NTHU-Route calculates the congested regions of all the routing edges and creates a definition for an auxiliary distance between maximum value of congestion and unity. After that, it performs partitioning of the auxiliary distance into m sub-distance I1 , I2 , ..., Im , where value of m is fixed to 10. For example, if the maximum value of congestion is 2, the sub-distances are in the following ranges [2, 1.9), [1.9, 1.8), [1.8, 1.7),..., [1.1, 1). Following this, starting from the most congested edge, an overflowing routing edge e is chosen which has values of congestion lying in I1. After I1 is processed, it iteratively processes all the sub-distances by rip up and rerouting all the marked two pin nets. Finally, before starting the next iteration of the algorithm, reroute and rip-up of the remaining overflowing two-pin nets is performed if there are any remaining nets which are not marked.

3.2. History based Cost function

In the main stage of global routing, NTHU Route works on a cost function based on history of routing, which is used to estimate the cost of routing an edge e. This is inspired by the concept of negotiated congestion, and it is defined as follows:

$$cost_e = b_e + h_e \times p_e + vc_e(6) \qquad (3)$$

Type where be is the basic cost of routing for an edge e and is assigned to one unit of wire length. Cost of congestion of the edge e is defined by the term hepe., and via cost function of edge is is defined by vce.. The manner in which historical term he is updated during the subsequent iterations is given by the following equation:

$$h_e^{i+1} = \begin{cases} h_e^i + k & \text{if e has overflow} \\ h_e^i & \text{otherwise} \end{cases} \quad (4)$$

where hei = 1 and i is the count of iteration. The definition of penalty term pe is as follows:

$$p_e = \begin{pmatrix} d_e + 1 \\ c_e \end{pmatrix}^{k_1} \quad (5)$$

where value of k1 is set to 5 which is a user-defined parameter.

The vce is defined as follows:

$$vc_e = \begin{cases} 1 & \text{if passing } e \text{ makes a bend} \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

A reduction in wire length and routing demand is obtained due to equation (6). This is due to the fact that if an edge e, which is considered for routing, is visited by another route which is present in the same multi pin net, it is very likely to share the edge e with some other two-pin routing instances.

4. Fast Route

This section presents an overview of a fast global router, FastRoute [8]. FastRoute has a different approach from the traditional global routers. FastRoute is a global router whose objective is to be applied at the application in both placement and routing. Global router is required to be invoked multiple times to estimate the interconnect estimation for placement. Here, runtime of the global router becomes critical factor. FastRoute avoids doing a maze routing algorithm in its implementation as maze routing is known to consume much of runtime of global routers. FastRoute makes use of Steiner tree construction to avoid maze routing. Due to the good quality of Steiner tree structures obtained during global routing, FastRoute performs maze routing only once during the whole global routing flow.

4.1. Generation of Congestion

Using minimal Steiner tree algorithm FLUTE [13], in this phase, Steiner tree are created for all the nets of the design. After this, all the generated Steiner trees are broken into two pin nets and routing is performed for these nets using L-shaped routing patter. Following this, a congestion map is obtained from this rough routing procedure.

4.2. Construction of Congestion-driven Steiner Tree

This phase consists of breaking up of Steiner trees into smaller nets with two pins. Then, routing using Z-shaped pattern is performed on the two pins nets which is accompanied by rip-up and re-route. Finally, using maze routing, the two pin nets around the congested region is ripped-up and re-routed. In order to find less congested regions, a cost function is introduce which is based on logistic function [14]. Congested regions are identified using the maze routing method but it is find less congestion on the route. FastRoute is used to find extremely fast method. At early designing stage of IC route

using the FastRoute perform the high speed operation and it is possible to integrated the sub blocks.

5. FGR

This section describes the architecture and key algorithms of Routing in modern routing systems. Initially the nets are FGR. We also present the mathematical foundation and detail decomposed into sub-nets by using minimum spanning tree of implementation of FGR in this section. After this, two methods of stages are (1) Pre-increment of historical cost for congestion hotspots and (2) area routing for small bounding box.

5.1. Algorithmic Frame Work

Congestion and wire-length minimization should be carefully balanced in a routing algorithm implementation. It may be step is used mainly for the designs that are difficult to route, necessary to take detour paths in cases of overcapacity of some example newblue3 circuit of ISPD07 benchmarks, which has GCells and also due to some routing violations, but detouring been proved to be un-routable as it contain high pin density should not be performed in excess as it may cause tiles which has great routing demands than the capacities overconsumption of routing resources, which may further available for routing around the tiles. Hence, a new method is increase congestion. From ISPD07 routing contest [13], it is required to be developed to manage the cases with high density observed that some routers have the capability of minimizing of pins. The negotiation based iterative rip-up and re-route wirelength and some are able to find solutions free of approach uses a history based cost function which is important violations, but very few of the routers are good at both. Most function in spreading of the congested edges. Here, the solution of the routers use predetermined rules which are applied in all and is used to optimize a multivariate function under a number of constraints [16]:

The constrained optimization is reduced to an unconstrained optimization formula of the Lagrangian function F

$$min_{x \in X} W(x) \text{ subject to } C_e(x) = 0, \quad 1 \leq e \leq n \qquad (9)$$

here, real-valued Lagrange multipliers are utilized. In the case of routing, overflow of routing edge e is represented by Ce(x) , the total wirel ength of the solution of routing is represented by W(x), solution x and is usually defined as a sum over nets or routing edges.

$$F(x, \lambda) = W(x) + \sum_{e=1}^{n} \lambda_e C_e(x) \qquad (10)$$

$$W(x) = \sum_{i=1}^{m} R_i(x) = \sum_{e=1}^{n} B_e(x) \qquad (11)$$

where Ri(x) is the number of segments used by net i and Be(x) is the number of nets passing through edge e. Thus (4) can be rewritten as:

$$F(x, \lambda) = \sum_{e=1}^{n} (B_e(x) + \lambda_e C_e(x)) \qquad (12)$$

In the above optimization there are two variables: 1) unknowns x 2) lagrange multipliers. Iterative techniques are used to optimize the form of function given above. Newton's method, steepest descent etc. can be used for optimization of such functions. Lagrange multipliers are solved additively using the equation which follows:

$$\lambda^{k+1} = \lambda^k + \alpha C(x^k) \qquad (13)$$

where parameter for line search is Į which is greater than 0. From Formula 2 we see that there is similarity between how he is computed and how Lagrange multipliers are computed. Here, they are also solving large problems which are sparse in nature. Hence a different optimizer in the form iterative procedure needs to be adopted. This could be hill-climbing or greedy search or rip-up and re-route. However, as the Lagrange multipliers are continuous function s, the same updation procedure can be utilized. Interpreting Formula 6 for a given net i in terms of NCR yields

$$c_e = b_e + h_e p_e(8) \qquad (14)$$

which is different than Formula 1 [21], but also is more reveland since it is preserving the basic cost.

Hence FGR uses DLM mechanism (Discrete Lagrange Multiplier) instead of NCR. NCR was used in the earlier verison of FGR (during ISP 2007 global routing contest). A new penalty function is used to calculate the value of pe. FGR achieves improved results after using DLM formulation.

5.2. Penalty for Congestion

Suppose re and ue are representing the resources and current usage for a routing edge e and the relative overflow we is defined by we = ue/re. Thee the congestion penalty term pe for edge e is computed as a function of pe as follows:

$$p_e = \begin{cases} exp(k(\omega_e - 1)) & if\ \ \omega_e > 1 \\ \omega_e & otherwise \end{cases} \qquad (15)$$

Note that here, for routing edges having overflow, we have exponential form of this cost function, which often increase the congestion estimate due to which overfull edges are avoided by maze routing algorithm while it is re-routing the nets. They have studied that 0 < k _ ln10 and found that higher values of k reduces runtime, but increases detouring and routed length. By default, the value of k is set to ln5 in FGR. In NCR it is common to route all the nets by their shortest paths, whereas, in FGR, to create an initial solution, weight of the edge is used as be+pe.

TABLE I. RESULTS OF ISPD 2008 ROUTING CONTEST

Benchmarks	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Rank
NTHU-Route	2	2	1	1	1	1	2	1	1	1	2	1	1	3	1	1	1.38
NTUgr	1	1	4	3	3	3	5	4	2	2	3	3	3	2	3	3	2.81
FastRoute 3.0	3	6	3	2	4	2	7	3	3	6	1	2	2	1	2	2	3.06
BoxRouter	4	4	2	4	2	4	1	2	6	5	4	4	4	4	4	4	3.63
FGR	5	7	5	5	5	6	3	5	5	3	5	5	6	6	5	6	5.13
NCTU	6	3	6	6	6	7	6	6	4	4	6	7	5	5	6	5	5.5
AMGR	7	5	7	7	7	5	4	7	7	7	7	6	7	7	7	7	6.5
IGOR	9	8	8	8	8	8	8	8	8	8	9	8	8	8	8	8	8.13
Simple Router	9	9	9	9	9	9	9	9	9	10	8	9	10	9	9	9	9.13
HSR	8	10	9	9	9	10	9	9	9	9	10	10	9	10	10	10	9.31

6. Conclusion

Global routing is important and relevant in the present IC design scenario due to the reducing transistor sizes and increasing proportion of routing delays. Thus research in the area of global routing is imperative. In this paper we summarize state-of-the-art techniques for global routing. Table 1 shows the performance of various routing methods and its bench mark circuit rank. We have studied and the presented the functionality of state-of-the-art routers and the key algorithms they have contributed. We have presented cutting edges implementation of four global routing tools. 1) NTHU-Route 2) FastRoute 3) NTUgr and 4) FGR. NTHU-Route performed well in the ISPD 2008 routing contest and emerged the winner. We presented a brief survey of the techniques used in NTHU-Route. FastRoute is extremely fast router which is targeted to be used in placement framework as well. We have presented an insight into its algorithms. NTUgr is another router which performed well in ISPD 2008 routing contest and emerged as runner-up. And, finally we present an overview of methods used in FGR which are novel (based on analytical optimization) and solves many problems in global routing space. This survey will benefit researchers as this gives an introduction to cutting-edge global routers. section presents an overview of a fast global router, FastRoute [8]. FastRoute has a different approach from the traditional global routers. FastRoute is a global router whose objective is to be applied at the application in both placement and routing. Global router is required to be invoked multiple times to estimate the interconnect estimation for placement.

References

[1] International Symposium on Physical Design, http://www.ispd.cc

[2] M. Pan and C. Chu, "Fastroute: A step to integrate global routing into placement," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design, ICCAD '06, San Jose California (**2006**) November 5 – 9.

[3] Xu, Y. Zhang, and C. Chu, "Fastroute 2.0: A high-quality and efficient global router," in Proceedings of Asia and South Pacific Design Automation Conference, ASP-DAC '07 (**2007**) January 2-7.

[4] M. Ozdal and M. Wong, "Archer: a history-driven global routing algorithm," in Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on, **(2007**)Nov, pp. 488–495.

[5] M. Pan and C. Chu, "Ipr: An integrated placement and routing algorithm," in Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE, (**2007**) June, pp. 59–62.

[6] Y. Zhang and C. Chu, "Gdrouter: Interleaved global routing and detailed routing for ultimate routability," in Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE, (**2012**) June, pp. 597–602.

[7] Z. Cao, T. T. Jing, J. Xiong, Y. Hu, Z. Feng, L. He, and X.-L. Hong, "Fashion: A fast and accurate solution to global routing problem," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 27, no. 4, (**2008)** April, pp. 726–737.

[8] J.-R. Gao, P.-C. Wu, and T.-C. Wang, "A new global router for modern designs," in Design Automation Conference, 2008. ASPDAC 2008. Asia and south Pacific, (2008) March, pp. 232–237.

[9] M. Cho and D. Pan, "Boxrouter: a new global router based on box expansion and progressive ilp," in Design Automation Conference, 2006 43rd ACM/IEEE, (**2006**), pp. 373–378.

[10] M. Cho, K. Lu, K. Yuan, and D. Pan, "Boxrouter 2.0: architecture and implementation of a hybrid and robust global router," in Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on, (**2007)** Nov, pp. 503–508.

[11] C.-C. Ho, H.-P. Tsai, L.-C. Lai, and R.-B. Lin, "A router for via configurable structured asic with standard cells and relocatable ips," in Quality Electronic Design (ISQED), 2015 16th International Symposium on, (**2015**) March, pp. 51–56

[12] K.-R. Dai, W.-H. Liu, and Y.-L. Li, "Nctu-gr: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-d global routing," Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol. 20, no. 3, (**2012**) March , pp. 459–472.

[13] W.-H. Liu, W.-C. Kao, Y.-L. Li, and K.-Y. Chao, "Nctu-gr 2.0: Multithreaded collision-aware global routing with bounded-length maze routing," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, vol. 32, no. 5, (**2013**) pp. 709–722,.

[14] H.-Y. Chen, C.-H. Hsu, and Y.-W. Chang, "High-performance global routing with fast overflow reduction," in Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific, (**2009**) Jan, pp. 582–587.

[15] J. Roy and I. Markov, "High-performance routing at the nanometer scale," in Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on, (**2007**) Nov, pp. 496–502.

[16] M. D. Moffitt, "Maizerouter: Engineering an effective global router," in Proceedings of the 2008 Asia and South Pacific Design Automation Conference, ser. ASP-DAC '08. Los Alamitos, CA, USA: IEEE Computer Society Press, (**2008**), pp. 226–231.

[17] C. Lee, "An algorithm for path connections and its applications," Electronic Computers, IRE Transactions on, vol. EC-10, no. 3, (**1961**) Sept., pp. 346– 365.

[18] G. Clow, "A global routing algorithm for general cells," in Design Automation, 1984. 21st Conference on, (**1984**) June, pp. 45–51.

[19] R. Kastner, E. Bozorgzadeh, and M. Sarrafzadeh, "Predictable routing," in Computer Aided Design, 2000. ICCAD-2000. IEEE/ACM International Conference on, (**2000**) Nov, pp. 110–113.

[20] R. Hadsell and P. Madden, "Improved global routing through congestion estimation," in Design Automation Conference, 2003. Proceedings, (2003) June, pp. 28–31.

[21] L. McMurchie and C. Ebeling, "Pathfinder: A negotiation-based performance-driven router for fpgas," in Field-Programmable Gate Arrays, 1995. FPGA '95. Proceedings of the Third International ACM Symposium on, (**1995**), pp. 111–117.

[22] C. Chu and Y.-C. Wong, "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, no. 1, (**2008)** Jan pp. 70–83.