



**INDIAN INSTITUTE OF MANAGEMENT CALCUTTA**

**WORKING PAPER SERIES**

**WPS No. 619/ January 2008**

**On some selected issues in VLSI Interconnect Layouts in the nanometer range**

**by**

**Partha Sarathi Dasgupta**

Professor, IIM Calcutta, Diamond Harbour Road, Joka P.O., Kolkata 700104 India

# On some selected issues in VLSI Interconnect Layouts in the nanometer range

Parthasarathi Dasgupta  
Indian Institute of Management Calcutta  
Kolkata 700104, WB, India  
partha@iimcal.ac.in

## ABSTRACT

The advent of deep sub-micron and nanometric regime for CMOS semiconductor technology has resulted in several restrictions in the physical design of VLSI circuits primarily through constraints imposed by interconnects. These constraints typically include the interconnect delay, congestion, cross-talk, power dissipation and others. These issues have to be considered in the physical design of VLSI circuits. For a specific set of design goals, faster design convergence is often achieved by considering estimates of some or all of these parameters in the physical synthesis and logic synthesis stages. Thus, accurate estimation of these parameters have direct impact on issues such as convergence, performance, yield and manufacturability of chips, and there is immense scope of research in design and performance of interconnects. In addition to these, efforts are on for exploration and use of routing architectures which are different from the traditional Manhattan architecture. In this survey, we attempt to provide a brief overview of some of the select areas of the state-of-the art research on interconnect routing in the deep sub-micron and nanometer range.

## 1. INTRODUCTION

The VLSI layout problem is usually solved in a hierarchical framework. Each stage of the hierarchy needs to be optimized, while the problem becomes manageable for the subsequent stages. Each stage of the hierarchical design, in turn, is divided into a number of phases. The different phases of VLSI physical (layout) design are:

- Partitioning: It is the task of dividing a circuit into smaller parts (components or modules). Objective is to partition a circuit into parts, so that the size of each component is within prescribed ranges and the number of connections between the components is minimized.
- Floorplanning: It is the determination of the approximate location of each module in a rectangular chip

area, given a circuit represented by a hypergraph. This phase may also be used to determine the shape of each module, and the locations of the pins on the boundary of the modules. Typically, each module has a set of implementations, each implementation given by an area, aspect ratio, delay, power consumption, and so on. The objective is to find the set of best combination of implementations of the modules.

- Placement: It is the determination of the best position for each module, assuming each module to have a fixed shape, and a fixed set of terminals. Some of the modules, such as I/O pads, have fixed positions. Cost functions in this phase may include the area, approximate wire-length, delay satisfaction, congestion minimization, and total (or maximum) power dissipation.
- Global Routing: It decomposes a large routing problem into smaller, manageable sub-problems to be solved in subsequent detailed routing phase. The objectives in this phase may include chip size minimization, minimizing the wire-length, evenly distributing congestion, ensuring signal integrity, and so on. Traditional routing architectures are Manhattan, but in recent times, the non-Manhattan  $X$ - and  $Y$ -routing architectures are gradually becoming the industry standards.
- Detailed Routing: This phase follows the global routing phase, and is used to effectively realize interconnections in a VLSI circuit. Traditional routing model of detailed routing is the two-layer Manhattan routing. Recently, unreserved layer models are also in use, where both vertical and horizontal wires can run in both layers. In recent routing methodologies, multi-layer and over-the-cell routing are allowed. The number of layers in a multi-layered routing style can typically go up to 7 or 8.

Global routing is one of the most critical steps in physical design flow, and involves a coarse connection of the signal nets for a given placement. Thus, wires and vias are assigned to each signal net in this phase. The quality of the global routing solution directly affects chip area, speed, routing congestion, power consumption and the number of iterations required to complete the design cycle. Thus, this step has a large influence on the performance of a circuit. However, global routing is a very hard problem: even the most simple

version of the problem, where a set of two-pin nets is to be routed under congestion constraints, is *NP*-complete [35].

Great amount of research efforts have been carried out on global routing during the last two decades, covering a variety of design styles including gate arrays, sea of gates, standard cell-based designs and custom circuits. These also considered various other objectives based on area of the layout, circuit performance, cost and ease of fabrication, and time to market. The shrinking feature sizes, and the increased use of several IP cores in the VLSI circuits have yielded new types of complexities in the routing problem.

In this survey, we provide a comprehensive overview of research in global routing, with emphasis on the recent advances in performance-driven routing as reported in literature. We restrict the survey within multi-net, multi-pin routing for custom-built integrated circuits.

Several prior surveys in similar areas complement the material presented here. Two early surveys on global routing are presented in [10], and in Lengauer [29]. The books by Sherwani [28], Sait and Youssef [30] and Sarrafzadeh and Wong [16], present a more updated coverage of progress in global routing. The book by Kahng and Robins [33] and the survey paper by Cong et al. [7] focus on global routing issues for a single net. The survey on multi-net global routing [31] discusses the recent global routing methods with emphasis on performance-driven multi-net routing.

This paper is organized as follows. Section 2 introduces the background of the problem, and a brief summary of the basic techniques used in global routing is provided in Section 3. Sections 4, 5, and 6 respectively discuss the timing-driven, crosstalk-driven and congestion-driven routing techniques. Section 7 summarizes some of the recent works in *X*- and *Y*-routing architectures. Finally, Section 8 provides the concluding remarks and possible future scopes of work.

## 2. PROBLEM BACKGROUND

A custom-built integrated circuit layout may be considered to comprise a number of macro blocks or modules, having pins around the boundary of each module. Each set of electrically equivalent pins, called a net, must be interconnected through routing. Due to constraints imposed by technology or methodology, in some cases, certain areas of the layout are prohibited from allowing wire routes. These areas are often referred to as obstacles. Moreover, in the multi-layer routing, the wires in the different layers may have to be interconnected using vias. The fundamental goal of the routing step is to connect every net successfully and to resolve resource contentions. In the recent VLSI design scenario, due to the presence of millions of basic circuit components and nets integrated on a single chip, it is usually very difficult to complete the routing in a single step. As such, the routing procedure is divided into two stages: global routing and detailed routing.

## 3. BASIC TECHNIQUES

In this section, we summarize some of the basic techniques that are used to obtain global routing. These include maze routing, Steiner tree construction, and 0-1 integer linear programming.

### 3.1 Maze routing

A fundamental problem often encountered in global routing is that of finding a shortest path connecting two pins in the presence of wiring blockages. A well known solution to this problem is the maze routing [32] algorithm, which works on a routing graph similar to a grid graph. Each edge  $e$  has an associated cost  $c(e)$ , and this cost may be different for different routing directions. If an edge  $e$  is within the wiring blockage area, then its cost  $c(e) = \infty$ , otherwise it has a finite specified cost. A common cost metric is the rectilinear edge length, counted in terms of the number of grid cells, which corresponds to the Manhattan distance. This algorithm involves a cost labeling step followed by a path tracing phase. During cost labeling, starting from the source vertex  $s$ , the accumulated cost from the source to each vertex is labeled one by one in the form of a *wave expansion* until the target vertex  $t$  is reached. The minimum cost path is then traced back from  $t$  to  $s$  by retrieving the information maintained in the cost labeling phase. The runtime for this algorithm is  $O(|E| + |V| \log |V|)$  through a Fibonacci heap implementation [14], where  $|E|$  is the number of edges and  $|V|$  is the number of vertices. If there is an shortest path between  $s$  and  $t$ , maze routing is guaranteed to find it. In practice, however, maze routing has been found to be slow and has large memory requirements. Another well known class of methods is the category of line probe-based algorithms [9], which do not rely on a grid graph. While these methods are faster than maze routing, they do not guarantee to find a solution, even if one exists. The maze routing algorithm has been extended to find a path connecting two pins in such a way that it favors a path that passes through less congested areas. It is important to note that Maze routing inherently can consider one net at a time, as such any extension to multi-net domain requires nets to be considered one at a time, in a particular order.

### 3.2 Steiner tree construction

A limitation of the maze routing and line probe algorithms is that they are applicable to only two pin nets. However, in practice, nets with more than two pins are usually encountered in the routing problem. A common approach in dealing with a multi-pin net is to decompose it into a set of two-pin nets. It can be easily seen that such a procedure may not lead to the best routing quality. The wire length of a routing tree can often be reduced by carefully introducing a select set of additional nodes along with the given pins, and constructing an *MST* over all of these nodes. The added nodes are called Steiner nodes, and the tree is called a Steiner tree [16]. Traditional VLSI routing problems used horizontal and vertical wires, and hence only rectilinear Steiner trees were considered. However, in recent times, the use of non-rectilinear wire geometries are quite predominant. The *X*-architecture has already been adopted in industry, and the *Y*-architecture is being explored. Construction of rectilinear Steiner minimum tree (*RSMT*) is a well-known *NP*-complete problem and several heuristic and approximation algorithms have been developed for this problem. Several of these are based on results that relate the wire length of the optimal *RSMT* to that of the optimal Minimum Spanning Tree (*MST*). However, in practice, merely minimizing wire length or delay may not be adequate. In such cases, we need to use an appropriate delay estimator, and need to consider the required arrival times (*RAT*) at the

terminals. For a given source pin and set of sink pins, and an associated set of slacks for the sink pins, the objective is to construct a Steiner tree that maximizes the slack at the source pin. Though there is lack of accurate but elegant delay estimators, Elmore delay estimate is usually the preferred choice due to its high fidelity. Several early works [1, 8] are dedicated to finding a good compromise between wire length and the maximum source-sink path length (radius). A review on the concepts of fidelity of delay estimators and the construction of Linear *RAT* trees are discussed in [12].

Many of the Steiner tree construction algorithms that have been proposed in the literature focus on the optimization of a single net, and do not consider wire congestion issues explicitly. Nevertheless, these algorithms can be applied to serially route the nets, with the most critical nets being routed in advance of other non-critical nets. When the edge cost is defined according to congestions, the Steiner minimum tree algorithms may be applied directly to even out congestion while simultaneously restraining the wire length [6].

### 3.3 0-1 integer linear programming

Global routing may be formulated as a special type of optimization problem, called a zero-one integer linear programming (0-1 ILP) problem. For a set of candidate routing trees  $T_i = T_{i,1}, T_{i,2}, \dots$  for net  $N_i$ , we use variable  $x_{i,j}$  to indicate if tree  $T_{i,j}$  is selected for net  $N_i$ . The global routing problem can then be formulated as

$$\begin{aligned} & \text{Minimize} && \lambda \\ & \text{Subject to:} && \sum_{T_{i,j} \in \mathcal{T}_i} x_{i,j} = 1, \forall N_i \in N \\ & && \sum_{i,j:e \in T_{i,j}} x_{i,j} \leq \lambda u(e), \forall e \in E \\ & && x_{i,j} = 0, 1, \quad \forall N_i \in N, \forall T_{i,j} \in \mathcal{T}_i \end{aligned} \quad (1)$$

The first constraint, along with the restriction of the  $x_{i,j}$ 's to 0, 1, requires that one tree be chosen for each net. The second constraint and the objective together ensure that the maximum congestion is minimized. In practice, the global routing problem is seldom solved entirely using the 0-1 *ILP* formulation since the problem size can grow to be very large. More often, the *ILP* technique is embedded into a larger overall global routing strategy, such as solving a subproblem at one hierarchical level of a hierarchical routing procedure [4], where the complexity of the computing the optimal solution to a 0-1 *ILP* is manageable.

## 4. TIMING-DRIVEN GLOBAL ROUTING

In this section, we summarize some of the recent ideas on the relative accuracy of the delay estimators, and the concepts of required arrival times and associated trees. The signal delay measures the performance of the circuits. Delay estimators are used for predicting the performance; the simplest estimate being the Manhattan distance between circuit nodes. Accuracy of these estimators, as already mentioned, is of utmost importance for several reasons such as faster design convergence. Empirically, Elmore Delay has been observed to be the best estimator. A convenient measure of the relative performances of the estimators is their *Fidelity*, or relative accuracy. We discuss here several existing fidelity metrics, their applications, and some of the possible ways of improving them.

Use of buffers is one of the well-known ways of reducing the interconnect delay, and their use tend to linearize the quadratic Elmore delay estimates. However, fidelity for delay estimators are likely to be different in the presence of buffers. In practice, the construction of global routing trees depend on the *Required Arrival Times (RAT)*: typically, for a given set of RAT values at the sink terminals, the objective is to maximize the RAT at the source of a routing tree. [15] introduces the concept of Linear *RAT* trees, which emerges from the fact that Elmore delay with buffers are found to be no better than Linear delay for source *RAT* maximization. The work in [12] discusses the notion of *LRAT* trees and describes a method of construction of such a tree.

Interestingly, it is observed ([26, 27]) that precise accuracy is often not required of the delay estimates used to construct the routing trees. As such, the measure of relative accuracy, or *fidelity* is introduced in [26]. Conceptually, it is the degree to which an optimal or near-optimal solution according to an estimator will also be nearly optimal according to the actual delay.

For Routing Trees, fidelity of a delay estimator is measured as follows:

- Enumerate all possible routing solutions.
- Rank all tree topologies using the estimator.
- Rank all tree topologies by SPICE delay model (actual).
- Find the average difference between the two sets of ranks for each topology.

### 4.1 Fidelity metrics

In [27], fidelity is obtained in three computations: (a) the average difference in ranks over all topologies, (b) the average rank difference for the topology with minimum delay with the estimator; and (c) the average difference of ranks for the five topologies which have the smallest delay according to the estimator. Results indicate highest fidelity of Elmore against SPICE.

Fidelity of a heuristic is defined ([34] as the portion of the pairwise inequality relations among the optimal solutions that are correctly determined by the heuristic. Formally, for  $m$  instances, if  $h_i(s_i)$  is the objective value of the heuristic (optimal) solution to instance  $i$ , then fidelity is defined as:

$$f = \frac{|\{(i, j) : 0 \leq i < j < m, ((h_i - h_j)(s_i - s_j) > 0) \text{ or } (s_i = s_j)\}|}{\binom{m}{2}} \quad (2)$$

### 4.2 Statistical correlation

Correlation, well-known in statistics, is used to measure the degree of inter-dependence of two statistical variables.

However, it may not always be convenient to give actual values to variables, but only to assign a *rank* to the instances. Rank correlation [41, 42] is used in such cases. Two well-known rank correlation measures are summarized below.

### 4.2.1 Spearman's Rank Correlation coefficient

Consider two variables  $x$  and  $y$ . Suppose that the paired observations on  $x$  and  $y$  are ordered in magnitude. Define the *rank* of the ordered observation  $x_j$  as  $\text{rank}(x_j) = j$ .

The rank correlation coefficient is given by

$$R = 1 - \frac{6 \sum_{i=1}^n d_i^2}{n(n^2 - 1)} \quad (3)$$

where  $d_i = |x_i - y_i|$ .

Higher the value of  $R$ , stronger is the correlation.

### 4.2.2 Kendall's tau

It is an elegant way of computing the rank correlation [43]. Fundamental notion of *tau* is the disarray. Let  $(x, y)$  be two variables on each member of a sample, where the  $x$  values are arranged in increasing order. Then the extent to which their corresponding  $y$  values depart from increasing order will indicate the weakness of correlation of  $x$  and  $y$ . A simple measure of the amount of disarray is the number of interchanges of  $y$  values that will put them in the same order as the  $x$  values. Let  $Q$  = the required number of inversions. Total number of distinct pairs in  $n$  observations is  $\frac{1}{2}n(n-1)$ , so that  $0 \leq Q \leq \frac{1}{2}n(n-1)$ .  $Q = 0$  implies all  $y$ 's are in order, and  $Q = \frac{1}{2}n(n-1)$  implies  $y$  totally out of order. Kendall's coefficient is defined as

$$\tau = 1 - \frac{4Q}{n(n-1)} \quad (4)$$

### 4.3 Fidelity vs Rank Correlation

Section 4.1 provides three definitions of fidelity as summarized below.

- The first definition uses the average difference of the ranks of the values obtained by SPICE and the estimators as a measure of the fidelity.
- Equation 2 count the number of matching ranks, and find its ratio to the total number of ranks. However, tie resolutions are different in them.

If  $d_{sd}$  is the standard deviation of the rank differences (without considering any weight), then the fidelity will be measured by the tuple  $(d_{av}, d_{sd})$ .

If  $E$  and  $E'$  are two estimators, with tuples  $(d_{av}, d_{sd})$  and  $(d'_{av}, d'_{sd})$  respectively, and  $f$  and  $f'$  their respective fidelities, then

$$f > f' \text{ if either } d_{av} < d'_{av} \\ \text{or } d_{av} = d'_{av} \text{ and } d_{sd} < d'_{sd}.$$

In all the definitions, distributions of the optimal solutions, and those of the estimated objectives have been ignored.

Research activities on delay traditionally relied on simple geometric abstractions, like tree cost or tree radius. The problem is usually to construct the optimal routing tree for

a net, to minimize the maximum delay in the tree. Other objectives include critical dink delay, average sink delay, and the maximization of source *RAT* [37]. The notion of *Required departure times* appeared in [38]. [39] addressed the more intuitive formulation of area minimization subject to given timing constraints on the sinks.

In an attempt to construct performance-driven routing trees based on the *RAT*s, Lillis et. al. introduced the notion of *P-trees* in [36]. However, almost all of these approaches used Elmore model for delay estimation.

### 4.4 The LRAT Tree Problem

Required Arrival Times (*RAT*) are successfully used as a measure of performance of global routing trees. For a routing tree, the *RAT* values at the sinks of a net are usually given, and the *RAT* at the driver gate (source) is to be obtained subject to certain constraints. Let  $q_0$  denote the *RAT* at the source  $s_0$ , and  $q_i$  denote the *RAT* at a sink  $s_i$ , for  $i = 1, n$ . If  $\Delta(s_0, s_i)$  denotes the delay of the signal between source (driver) and the  $i^{\text{th}}$  sink, the constraint is

$$q_0 \leq q_i - \Delta(s_0, s_i) \quad (5)$$

For Manhattan (*M-*) routing, the minimum delay estimate between  $s_0$  and  $s_i$  is the M-distance  $d_M(s_0, s_i)$  between them. Additional delay in routing may result due to detours.

**DEFINITION 1.** *Given a source at the origin, and a set of sinks in a plane, slack at sink  $s_i$  is given by*

$$\sigma_i = q_i - q_0 - d_M(s_0, s_i) \quad (6)$$

where the source and sink terminals are assumed to be in a plane, and w.l.o.g., the source is considered to be at origin with all the sinks in the first quadrant. Thus, slack at a sink defines an upper bound on the detour encountered in connecting the source and the sink using rectilinear path. For practical purposes, when the sink terminal slacks are given, the problem is to construct a routing tree on source and the sinks such that the delay from source to each of the sinks does not violate the slack at that sink. The *LRAT* tree problem is then defined as follows:

*To find the shortest rectilinear tree  $T$  connecting all the sinks to the source such that for any sink,*

$$d_T(s_0, s_i) \leq d_M(s_0, s_i) + \sigma_i \quad (7)$$

*where  $d_T(s_0, s_i)$  is the length of the path from  $s_0$  to  $s_i$  in the rectilinear tree.*

It is clear to see that the Rectilinear Steiner Arborescence (*RSA*) and Rectilinear Steiner Minimum Tree (*RSMT*) are both *LRATT*s with zero slack and infinite slack respectively for all the terminals. Thus, we have,

**OBSERVATION 1.** *The Linear RAT tree is a generalization of the Rectilinear Steiner Minimum Tree and the Rectilinear Steiner Arborescence.*

From the range of the allowed slacks for the three types of routing trees, if  $l(RSA)$ ,  $l(LRATT)$ ,  $l(RSMT)$  and  $l(RMST)$

respectively denote the lengths of the optimum *RSA*, *RSMT*, *LRATT* and the Rectilinear Minimum Spanning Tree (*RMST*), following observation is clear [16]:

$$\text{OBSERVATION 2. } l(RSA) > l(LRATT) > l(RSMT) > \frac{l(RMST)}{1.5}.$$

## 5. CROSSTALK DRIVEN GLOBAL ROUTING

One of the major concerns of VLSI design in the deep sub-micron regime is to maintain signal integrity. Signal integrity is often affected by crosstalk noise due to the greater proximity between wire tracks, and the consequently increased contribution of coupling capacitances. The crosstalk noise from the coupling capacitance is usually considered and restrained in the detailed routing stage since the wire neighborhood information is required to have a reasonable estimation on coupling capacitance. Flexibility of the wiring topology during detailed routing is limited and any change in route is limited to a local small region, such as a routing channel or a switch box. Crosstalk avoidance in global routing is considered in [92, 93] in order to utilize the inherent flexibility.

The crosstalk driven global routing problem may be formulated as: *Crosstalk-Constrained Global Routing (CCGR)* Given a set of  $n$  nets, a routing graph  $G = (V, E)$  and the crosstalk constraints  $C_1, C_2, \dots, C_n$ , find an extended global routing solution  $S$  such that  $X_i(S) \leq C_i$ , for all  $1 \leq i \leq n$ .

The *CCGR* problem is solved using a two stage heuristic in [92]. The first stage is a net-by-net sequential routing step that minimizes the total crosstalk over all of the nets. The routing procedure for each net consists of a Steiner tree construction and layer/track assignment process. In [92], it is shown that the problem of assigning only layers/tracks to minimize crosstalk in one region is *NP*-hard. This motivated the design of a heuristic that ensures that that the routed nets are kept in their layers and the order of the track assignment in one layer is not changed. For a new net to be routed in the sequential routing, a layer is chosen and the net is inserted into a track among the routed nets so that the increase of crosstalk is minimized. Then, the increase in the crosstalk  $xtalk(e)$  on an edge  $e$  can be obtained according to this layer/track assignment information. The minimum cost Steiner tree is heuristically constructed according to the following cost definition.

$$cost(e) = \alpha \times l(e) + \beta \times overflow(e)^2 + \gamma \times xtalk(e) \quad (8)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are constant weighting factors.

In the second stage, each net with a crosstalk violation is ripped up and rerouted to satisfy the crosstalk constraint using a Lagrangian relaxation based method.

## 6. CONGESTION-DRIVEN GLOBAL ROUTING

A design is said to have routing congestion when the demand for the routing resources in some region within the design exceeds their supply. In order to measure the congestion in a region of the layout, the layout is typically divided into a number of rectangular bins. Congestion metrics include

the track overflow in bins, number of congested bins, and the maximum congestion among all the bins. The impact of congestion in a layout design is many-fold: it may worsen the performance of the design, it may add up uncertainty to design closure, and so on. A significant work on congestion-driven routing appears in [5]. It proposes a hybrid and robust global router which has strong ability to improve routability and minimize the number of vias with blockages, while minimizing wirelength. The proposed router can perform multi-layer routing with 2D global routing and layer assignment. It is based on two ideas: robust negotiation-based  $A^*$  search for routing stability, and topology-aware wire ripup for flexibility. After 2D global routing, a 2D-to-3D mapping (layer assignment) is done by the layer assignment which is powered by progressive via/blockage-aware integer linear programming.

A significant work on placement of rectangular modules that is amenable to *Y*-routing with minimum congestion appears in [24].

## 7. INTERCONNECT ROUTING IN THE X- AND Y-ARCHITECTURES

A commonly used technique for reducing the interconnect delay involves decreasing the wire length of the interconnects. Moreover, traditionally, the interconnect wires are restricted to only horizontal and vertical directions, viz., the Manhattan architecture. Clearly, such restriction adds redundant wire length over the Euclidean optimum. While lithographic considerations allow the use of wiring with arbitrary orientations, combination of  $45^\circ$  and  $135^\circ$  wires with horizontal and vertical wires have been shown to yield considerable reduction in wire length. Thus, most of the current manufacturing technologies support  $45^\circ$  wires [11].

*X*-routing is now well appreciated in chip manufacturing circle; however, the research community has recently started investigating the *Y*-architecture. This includes the LSI Logic patents [2, 3] and the work reported in [40]. This refers to a new kind of wiring with  $0^\circ$ ,  $60^\circ$ , and  $120^\circ$  oriented wires for on-chip interconnect, along with supporting methodologies including hexagonal die shapes, hexagonal power and clock distribution, etc. [18]. Though *Y*-routing may not be capable of reducing wire lengths to the extent as for *X*-routing, the *Y*-architecture does have certain specific advantages over the *X*-architecture: *Y*-routing has been observed to support regular routing grid, enabling simplified manufacturing processes and routing and design rule checking algorithms. Also, *Y*-architecture provides a throughput and wirelength improvement close to the *X*-architecture with one less routing direction.

The use of diagonal wires was suggested decades ago [19], and were exploited on printed-circuit boards and integrated circuits for more than a decade [21]. [20] presents some of the challenges and opportunities of *X*-architecture, and demonstrates a highly promising future for the same. Algorithms for *X*-architectures are proposed in [13] and [17]. The former proposes new algorithms for construction of Octilinear Steiner trees. On the other hand, *Y*-routing and *Y*-architecture for integrated circuits were introduced through series of works of two different groups in [40] and [3, 2]. The work in [18] gives an in-depth analysis of *Y*-architecture,

and highlights the potential advantages of their use vis-a-vis the  $X$ -architecture. [22] introduces some new concepts and algorithms related to Steiner trees with uniformly oriented edges. Algorithms for construction of  $Y$ -routing trees appear in [23, 25].

In Deep sub-micron regime, interconnect delays dominate VLSI circuit design. Thus, construction of cost-effective global routing trees is key to such designs. In order to reduce the interconnect delay, traditional Manhattan ( $M$ -) routing architectures are currently being replaced by the diagonal  $X$  architectures. A recent routing architecture is based on  $Y$  interconnects, involving the pervasive use of  $0^\circ$ ,  $60^\circ$ , and  $120^\circ$  oriented global and semi-global wirings. Unlike the  $X$ -routing,  $Y$ -routing is observed to support regular routing grid, which is important for simplifying manufacturing processes and routing and design rule checking algorithms.

## 8. CONCLUSION

The paper presents a brief survey of some issues in global routing in the deep sub-micron range. We consider multi-pin multi-net situations, and concentrate on some of the performance-driven problems of global routing. The performance parameters considered by us include congestion in a layout, occurrence of crosstalk, and timing closure. In addition to these, we provide a very brief summary of some latest works in  $Y$ -routing architectures. It is interesting to note that many of the parameters of performance are inter-dependent. For instance, congestion is likely to have an effect on crosstalk, interconnect delay is highly dependent on its length, and congestion is also related to the interconnect length, and so on. As a consequence, there is a need to consider a multi-objective scenario instead of the traditional single objective functions.

Another interesting area of research is the non-Manhattan routing. Both  $X$ - and  $Y$ -routing architectures have their inherent pros and cons, though both of them appear to be better than the traditional Manhattan architecture. Some research is currently being carried on in the area of general  $\lambda$ -geometry as well, but as of now, these have had very little practical impact.

The extreme difficulty of the global routing problems have drawn attention of researchers from both academia and industry, and there is immense scope in years to come.

=

## 9. REFERENCES

- [1] C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger, "Prim-dijkstra tradeoffs for improved performance-driven routing tree design", *IEEE Transactions on Computer-Aided Design*, Vol. 14, No. 7, pp. 890-896, July 1995.
- [2] M. D. Rostoker et al., "Hexagonal Architecture", U.S. Patent, No. US6407434B1, June 2002.
- [3] M. D. Rostoker et al., "CAD for Hexagonal Architecture", U.S. Patent, No. US5822214, Oct. 1998.
- [4] M. Burstein and R. Pelavin, "Hierarchical wire routing", *IEEE Transactions on Computer-Aided Design*, Vol. 2, No. 4, pp. 223-234, October 1983.
- [5] M. Cho, K. Lu, K. Yuan and D. Z. Pan, "BoxRouter 2.0 architecture and implementation of a hybrid and robust global router", *International Conference on Computer-aided Design*, 2007, pp. 503-508.
- [6] C. Chiang, C. K. Wong, and M. Sarrafzadeh, "A weighted Steiner tree-based global router with simultaneous length and density minimization", *IEEE Transactions on Computer-Aided Design*, Vol 13, No. 12, pp. 1461-1469, December 1994.
- [7] J. Cong, L. He, C.-K. Koh, and P. H. Madden, "Performance optimization of VLSI interconnect layout", *Integration: the VLSI Journal*, Vol. 21, pp. 1-94, 1996.
- [8] J. Cong, A. B. Kahng, G. Robins, M. Sarrafzadeh, and C. K. Wong, "Provably good performance-driven global routing", *IEEE Transactions on Computer-Aided Design*, Vol. 11, No. 6, pp. 739-752, June 1992.
- [9] D. W. Hightower. A solution to line routing problems on the continuous plane. In *The Sixth Design Automation Workshop*, pages 1-24, 1969.
- [10] E. S. Kuh and M. Marek-Sadowska. Global routing. In T. Ohtsuki, editor, *Layout design and verification*, pages 169-198. Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1986.
- [11] X Initiative Homepage, <http://www.xinitiative.org/>.
- [12] P. Dasgupta and P. Yadava, "Linear Required-Arrival-Time trees and their construction", *Proc. of the International Conference on VLSI Design*, 2006, pp. 790-793.
- [13] Q. Zhu, H. Zhou, T. Jing, X-L Hong and Y. Yang, "Spanning Graph-based Nonrectilinear Steiner Tree Algorithms", *IEEE Trans. on CAD/ICAS*, Vol 24, No 7, pp. 1066-1074, 2005.
- [14] M. L. Fredman and R. E. Tarjan, "Fibonacci heaps and their use in improved network optimization algorithms", *Journal of the ACM*, Vo. 34, 1987, pp. 596-615.
- [15] P. Dasgupta, "Revisiting VLSI Interconnects in Deep Sub-Micron: Some Open Questions", *Proc. of the 18<sup>th</sup> International Conf. on VLSI Design*, Jan 3-7, 2005, Kolkata, pp. 615-620.
- [16] M. Sarrafzadeh and C. K. Wong. An introduction to VLSI physical design. McGraw-Hill, New York, NY, 1996.
- [17] Chris S. Coulston, "Constructing Exact Octagonal Steiner Minimal Tree". Great Lakes Symposium on VLSI'03, April 28-29, 2003, Washington, DC, USA.
- [18] H. Chen, Chung-Kuan Cheng, A. B. Kahng, I. Mandoiu, Q. Wang and Bo Yao, "The Y-Architecture for On-Chip Interconnect: Analysis and Methodology", *IEEE Trans. on Computer-aided design*, 2005.
- [19] SDA Systems Edge Product, C. 1986.
- [20] S. Teig, "The X Architecture: Not Your Father's Diagonal Wiring", *ACM International Workshop on System-Level Interconnect Prediction*, pp. 33-37, 2002.
- [21] E. Lodi, "Routing multiterminal nets in a diagonal model," *Proceedings of the 1988 Conference on Information Sciences and Systems*, Dept of EE, Princeton University, pp. 899-902.
- [22] M. Brazil, D. A. Thomas, J. F. Weng, and M.

- Zachariassen, "Canonical Forms and Algorithms for Steiner Trees in Uniform Orientation Metrics", Tech. Rep. 02-22, Department of Computer Science, University of Copenhagen, 2002.
- [23] T. Samanta, P. Ghosal, H. Rahaman and P. Dasgupta, "A heuristic method for constructing hexagonal Steiner minimal trees for routing in VLSI", *Proc. of IEEE International Symposium on Circuits & Systems*, 2006.
- [24] T. Samanta, P. Ghosal, H. Rahaman and P. Dasgupta, "Minimum Congestion Placement for Y-interconnects: SOme studies and observations", *Proc. of IEEE CS International Symposium on VLSI*, pp. 73-80, 2007.
- [25] A. Thurber and G. Xue, "Computing Hexagonal Steiner Trees using PCS", *IEEE International Conference on Electronics, Circuits & Systems*, pp. 381-384, 1999.
- [26] K D Boese, A B Kahng, B A McCoy, and G Robins, "Fidelity and Near-Optimality of Elmore-Based Routing Constructions", *Proc. of the IEEE International Conference on Computer Design*, Cambridge, October 1993, pp. 81-84.
- [27] K D Boese, A B Kahng, B A McCoy and G Robins, "Near-Optimal Critical Sink Routing Tree Constructions", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol. 14, No. 12, December 1995, pp. 1417-1436.
- [28] N. A. Sherwani. Algorithms for VLSI physical design automation. Kluwer Academic Publishers, Norwell, MA, 3rd edition, 1999.
- [29] T. Lengauer. Combinatorial algorithms for integrated circuit layout. John Wiley and Sons, New York, NY, 1990.
- [30] S. M. Sait and H. Youssef. VLSI physical design automation: theory and practice. McGraw-Hill, New York, NY, 1995.
- [31] J. Hu and S. S. Sapatnekar, "A survey on multi-net global routing for integrated circuits", *Integration, The VLSI Journal*, Vol. 31, No.1, pp. 1-49, 2001.
- [32] C. Y. Lee, "An algorithm for path connection and its applications", *IRE Transactions on Electronic Computers*, Vol. 10, No. 3, pp. 346-365, 1961.
- [33] A. B. Kahng and G. Robins. On optimal interconnections for VLSI. Kluwer Academic Publishers, Boston, MA, 1995.
- [34] Joseph L Ganley, "Accuracy and Fidelity of Fast net length estimates", *Integration, The VLSI Journal*, 1997, Vol 23, pp. 151-155.
- [35] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-Completeness", W. H. Freeman and Co., SFO, 1979.
- [36] J Lillis, C.-K. Cheng, T.-T.Y. Lin and C.-Y. Ho, "New Performance driven routing techniques witexplicit area/delay tradeoff and simultaneous wire sizing", *33<sup>rd</sup> Design Automation Conference*, 1996, pp. 396-400.
- [37] J. Cong, L. He, C. Koh and P. Madden, "Performance optimization of VLSI interconnect layout", *Integration, the VLSI Journal*, Vol. 21, 1996, pp. 1-94.
- [38] L.P.P.P. van Ginneken, "Buffer Placement in Distributed RC-tree Networks for Minimal Elmore Delay", *Proc. IEEE International Symposium on Circuits & Systems*, 1990, pp. 865-868.
- [39] S. S. Sapatnekar, "RC Interconnect Optimization under the Elmore Delay Model", *Proc. ACM/IEEE Design Automation Conference*, 1994, pp. 387-391.
- [40] H. Chen, B. Yao, F. Zhou and Chung-Kuan Cheng, "The Y-Architecture: Yet another On-Chip Interconnect solution", *Proc. of the Asia-South Pacific Design Automation Conf.*, 2003, pp. 840-846.
- [41] S. Kotz and N L Johnson, *Encyclopedia of Statistical Sciences*, J. Wiley & Sons, Inc. 1981.
- [42] Karl V Bury, *Statistical Models in Applied Science*. John Wiley & Sons, Inc., 1975.
- [43] M.G.Kendall, *Rank correlation methods*, 4<sup>th</sup> Edition, Charles Griffin, London.