GLOBAL ROUTING FOR VLSI STANDARD CELLS

Hao Sun and Shawki Areibi

School of Engineering
University of Guelph
CANADA N1G 2W1
E-mail: {hsun, sareibi}@uoguelph.ca

ABSTRACT

Global routing is an important and time consuming step in the VLSI physical design cycle. In order to effectively solve this problem we present two standard cells global routing approaches: a heuristic approach and an Integer Linear program (ILP) based approach. The objective of the heuristic approach is to find the minimum cost path for each net by enumerating a set of possible 2-bend routes. It is also parallelized by using the Message Passing Interface (MPI) library to speed up the running time and achieve better performance. The ILP based approach formulates the global routing problem as an ILP problem and then relaxes it as a Linear Programming (LP) problem to reduce the computation complexity. All the algorithms are evaluated on Sun-blade 2000, The experimental results show high performance from both global routers. Good speedups and solution qualities are obtained from the parallel implementation. The average speedup achieved is five on six processors, and the total density is reduced by an average of 2.45% as well.

1. INTRODUCTION

The routing problem determines interconnections among cells that have been assigned positions by the placement CAD tool. Routing is normally performed in two stages. The first stage, called global or loose routing, determines through which channels a connection will run. The second stage, called local or detailed routing, fixes the precise paths that a wire will take. Global routing decomposes a large routing problem into small, manageable problems for detailed routing. Global routing is an extremely important and time consuming phase in the VLSI physical design cycle. For a circuit containing 100,000 cells and nets, global routers can easily take several hours to perform their operation.

This paper presents two approaches: a heuristic approach and an ILP based approach. The major contributions are: (i) Both approaches effectively solve the global routing problem, (ii) Parallel algorithms are implemented, which balance the work loads, shorten the running time and yield better solutions, (iii) Take advantage of the optimization solution package CPLEX and parallel library MPI to solve the global routing problem.

The remainder of this paper is organized as follows: Section 2 briefly reviews the previous work on global routing problem. Section 3 introduces a sequential global router and its parallel implementation. Section 4 presents an effective ILP formulation for global routing. The comparative results are presented in section 5. Finally, the paper concludes in section 6.

2. BACKGROUND

Several approaches have been proposed to solve the global routing problem in the past decade. There are two classes of solution methodologies: $Sequential\ Tech- niques$ and $Integer\ Programming\ Techniques$.

In sequential global routing, nets are first ordered according to certain priority, and then each net is separately routed. Some widely used sequential algorithms for two-terminal nets include maze routing algorithms [1], line-probe algorithms and shortest path algorithms [2]. Rectilinear Steiner Tree (RST) based approaches are proposed to route the multi-terminal nets. In [3], a multi-source routing algorithm based on constructing minimum cost, minimum diameter arborescence was presented. In [4], a weighted Steiner tree based global router was introduced to minimize the length and density of the routes.

Sequential global routers route the nets one by one and their qualities degrade if nets are not sorted appropriately. Integer Programming based approaches avoid the net-ordering problem. They have a global view of the whole circuit and route all the nets simultaneously. Integer Programming approaches formulate the global routing problem as a 0/1 integer programming (IP) problem. The objective is to connect each net simultaneously without violating the constraints. In [5], a hierarchical approach was proposed, the circuit layout is divided into sub-regions, the IP problem is then developed and solved for each sub-region. In [6], an integer solution for the problem was obtained by choosing the Steiner tree with the highest fractional value for each tree. In [7], a linear relaxation of the routing problem was formulated as a multi-commodity network flow problem. In [8], an adaption of Kar-

markar's interior point method was used to solve the linear relaxation of the global routing problem.

3. SEQUENTIAL HEURISTIC APPROACH

The sequential heuristic global router finds the minimum cost path for each net. It constructs the minimum spanning trees (MST) by enumerating a subset of all possible 2-bend routes [9]. In Figure 1, the standard cells layout is represented as a $Cost\ Array$ routing model. Each possible routing position in a channel is an element in the $Cost\ Array$. Its vertical dimension is set to the number of rows plus 1 and its horizontal dimension is set to the width of the routing grid. There are two constraint values for each element in the Cost Array: H_{ij} and V_{ij}, where position ij is at channel i and routing cell j. H_{ij} stands for the number of wires horizontally passing through position ij and V_{ij} is the cost of vertically traversing the position ij. The total cost of a path P is $Cost(P) = \sum_P H_{ij} + v \times C$, where C is the number channels being traversed in the path.

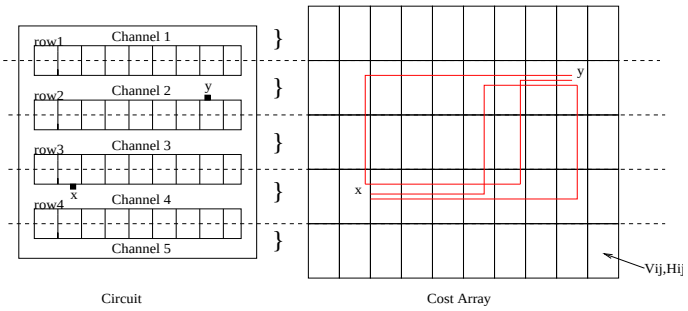

Fig. 1. Cost array model

3.1. Sequential Algorithm Implementation

The sequential Heuristic approach can be summarized by the following: (i) A multi-point wire is decomposed into two-point segments using a minimum spanning tree (MST) algorithm. (ii) Explore the electrically equivalent pin [10] and find a set of all possible 2-bend routes for each 2-point wire. (3) Evaluate the 2-bend routes set and find the minimum cost path for each two-point wire. (4) Combine the two-point minimum cost paths back together. (5) The new wire is determined and placed.

The sequential heuristic approach considers only 2-bend routes, which is simple and takes advantage of the standard cell layout feature. Therefore it reduces the running time and produces good results as well. An outline of the sequential heuristic approach is illustrated in Figure 2.

Sequential Heuristic Algorithm

```
1.  START
2.  Read Circuit information and create cost array;
3.  While( wire queue not empty)
4       If (2−terminal wire)
5.        Explore all possible 2−bend routes;
6.        Find the minimum cost path;
7       Else
8.        Decompose wire into 2−point segments;
9.        FORALL segments
10.         Explore all possible 2−bend routes;
11.         Find the the minimum cost path;
12.       End FORALL
13.       Combine the 2−point paths back together;
14      End if
15. End While
16. END
```

Fig. 2. Sequential implementation of the heuristic global router

3.2. Parallel Algorithm Implementation

A parallel algorithm is implemented by using the Message Passing Interface (MPI), which is a collection of message-passing libraries designed to provide access to advanced parallel hardware and is available on a range of distributed architectures.

In order to parallelize the sequential algorithm, the cost array is first divided into several regions. The division considers the standard cell row based layout feature. Each region may consist of one or several rows and channels. Each region is owned by a processor, which is also called $owner\ processor$.

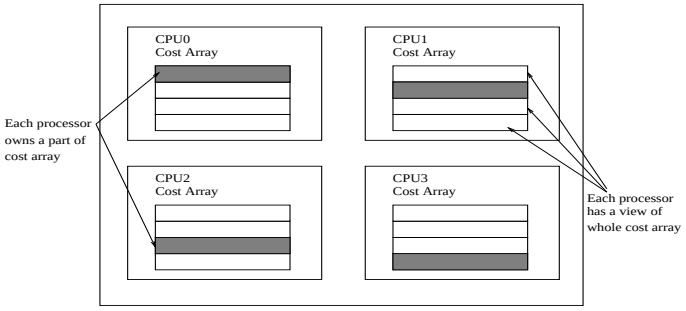

Fig. 3. Cost array is divided into 4 regions

However, each $owner\ processor$ not only owns its assigned region but also has a global view of the whole array, which is illustrated in Figure 3. A $ThresholdCost$ parameter is used in the wire assignment phase to balance the work loads whose value is based on the length of wires. A wire is assigned to a processor according to its up-most pin location

if its wire length is lower than $ThresholdCost$, otherwise, it will be put into a wire queue and be held until the final step of the wire assignment. A new data structure $delta\ array$ is added as well, which is used to keep track of the changes of the cost array. Finally each processor starts to route the wires in its own region. If a wire extends to other regions, the routing task will not be passed to the other processors since each has a global view of the array. This approach can effectively reduce the network traffic caused by passing the extended wires, it also balances the work loads very well. An outline of the parallel routing algorithm is illustrated in Figure 4.

```
Heuristic Parallel Algorithm

1.  START
2.  Read Circuit information and create cost array;
3.  Partition cost array into regions;
4   If (wirelength<ThresholdCost)
5.      Assign wire according to the upmost pin;
6.  Else
7       Hold wire into the wire waiting queue;
8.  End If
8.  FORALL processors
9.    While (not done)
10.     While (current wire queue not empty)
11.       Pick up a wire and route the wire;
12.       Compute the delta array;
13.       Send/Receive update information;
14        Update cost array;
15.     End While
16.     If (current wire queue is empty)
17        Pick up a wire from the wire waiting queue;
18.   End While
19. End FORALL
20. END
```

Fig. 4. Parallel implementation of the heuristic global router

4. ILP BASED APPROACH

The heuristic approach still has the net-ordering problem, thus the ILP based approach is proposed to overcome this drawback. Furthermore, The ILP based approach connects the nets by constructing Minimum Rectilinear Steiner Trees (MRST), which produces shorter wire-length rather than relying on 2-bend routes.

The ILP based approach is implemented in three steps: First, a set of MRST are generated and a binary variable is assigned to each tree. Next, an Integer Linear Programming (ILP) formulation is constructed with all the necessary constraints. The ILP is further relaxed as a Linear Programming (LP) problem to reduce the computational complexity. Finally, the problem is solved by using an ILP solver from ILOG (CPLEX).

4.1. ILP Formulation

The ILP formulation of the global routing problem is based on Vannelli's work [8]. The objective of this formulation model is to maximize the net connection benefit. For each net $i \in N$, there are several possible connections, and each of them is a tree in a set of rectilinear Steiner trees $T_i^1, \ldots, T_i^k$. A variable y_i is associated with each tree which connects a net. This variable is set to 1 if that particular tree is used otherwise it is set to 0 otherwise. However, only one tree is finally selected to connect this net.

The constant b_j is the benefit of connecting a net by tree y_j and is defined as:

$$b_j = (MaxWirelength + 1) - (Wirelength\ of\ tree\ j) \quad (1)$$

Assume a_{ij} is an element in a (0,1) matrix $[a_{ij}]$ which records all possible trees to connect nets, c_i is the capacity constraint, the ILP formulation can be expressed as:

$$Maximize \qquad \sum_{j=1}^{n} b_j y_j. \quad (2)$$

subject to

$$\sum_{y_j \in T_k} y_j = 1, \qquad k = 1, 2, \ldots |N|$$

$$\sum a_{ij} y_j \leq c_i, \qquad 1 = 1, 2, \ldots m$$

$$y_j \in \{0, 1\}$$

4.2. LP Relaxation

The CPU time required to solve the ILP problem increases exponentially when the problem size increases. Therefore, the formulated ILP problem is relaxed as a Linear Programming (LP) problem to reduce the computation complexity. The LP relaxation is formulated by replacing the integer constraints imposed on the variables with linear boundary constraints.

5. RESULTS

The proposed global routers are evaluated on Sun-blade 2000 workstations and tested on the MCNC benchmarks. These benchmarks are widely used in verifying VLSI physical design subproblems such as placement and routing. Table 1 compares the sequential heuristic results with the ILP based approach results before they are parallelized. It is clear that the ILP based approach produces shorter wire-lengths than those produced by the sequential heuristic approach but requires longer processing time. Table 2 lists the speedup results after the parallel algorithm is implemented on the heuristic approach. Let $T(N)$ be the time required to complete the task on N processors, speedup can be defined as the ratio: $S(N) = T(1)/T(N)$. The result shows the parallelism is

Circuit	Heuristic		Stein_MIP	
	Wirelength(u)	time(sec)	Wirelength(u)	time(sec)
prim1	891798	4.50	830365	9.09
ind1	2130137	130.90	1509809	169.09
prim2	6192874	150.31	4151385	197.93
bio	6327872	326.22	5266555	488.04
ind2	50216723	1411.99	45032892	2175.79
ind3	101136107	1734.53	92599062	3042.07
avq.small	8064612	3302.56	7085894	5510.95
avq.large	8736272	3783.28	7762927	6391.25

Table 1. Comparison of two global routers

Circuit	Speedup Obtained		
	on 2 procs	on 4 procs	on 6 procs
prim1	1.94	3.27	3.60
ind1	1.91	3.71	5.20
bio	1.95	3.87	5.28
prim2	1.89	3.85	5.16
ind2	1.94	3.89	5.49
ind3	1.98	3.79	5.31
avq.small	1.97	3.75	5.27
avq.large	1.97	3.77	5.28

Table 2. Speedup results of the parallel implementation

very effective and achieves good speedups. Distributed processing in fact improves the routing quality as well as shown in Table 3. The total channel density of the heuristic sequential router is reduced by an average of 2.45%, which eventually leads to a reduction in chip area.

Circuit	Total Channel Density		
	Before Parallel	After Parallel	Improve %
ind1	450	439	2.44%
prim2	582	573	1.54%
bio	1082	1039	3.97%
ind2	3014	2912	3.38%
ind3	5174	5021	2.95%
avq.small	8023	7926	1.20%
avq.large	8765	8625	1.59%

Table 3. Parallelism improves the total density

6. CONCLUSIONS

In this paper, two global routing algorithms for VLSI standard cells and a parallel implementation are proposed. The heuristic approach is simple but efficient. Through parallel implementation, it obtains good speedups and better routing qualities. This shows our algorithm is very effective in the parallelization. The Integer Linear Programming (ILP) based approach avoids the net-ordering problem and uses more efficient trees in the form of MRST to connect nets. The ILP approach therefore achieves better performance than the sequential heuristic approach but takes longer running time. Our future work will focus on parallel implementation of the ILP based approach to further reduce the running time and solve larger size problems.

7. REFERENCES

[1] C.Y. Lee, "An Algorithm for Path connection and its application," *IRE Transactions on Elctronic Computers*, pp. 346–365, 1961.

[2] D. W. Hightower, "A Solution to the Line Routing Program on a Continuous Plane," in *Proceedings of 6th Design Automation Conference*. 1969, IEEE.

[3] J. Cong and P. Madden, "Performance Driven Global Routing for Standard Cell Design," in *Intl Symposium on Physical Design*. 1997, pp. 73–80, IEEE.

[4] C. Chiang M. Sarrafzadeh and C.X. Wong, "A Powerful Global Router: Based on Steiner Min-Max Trees," *Trans. on CADICS*, pp. 1461–1469, 1994.

[5] M. Burstein and R. Pelavin, "Hierarchical Wire Routing," in *IEEE trans. Computer-Aided Design, Vol. CAD-2*, Oct. 1983, pp. 223–234.

[6] T.C. Hu and M. T. Shing, "A Decompsition Algorithm for Circuit Routing," *in VLSI Circuit Layout: Theory and Design,T.C. Hu and E.S. Kuh*, pp. 144–152, 1985.

[7] E. Sharagowitz and S. Keel, "A Global Router on a Multi-commodity Flow Model," *Interatction*, pp. 3–16, 1987.

[8] A. Vannelli, "An Adaptation of the Interior Point Method for Solving the Global Routing Problem," *IEEE trans on Computer-Aided Design*, pp. 193–203, February 1991.

[9] J.Rose, "LocusRoute: A Parallel Global Router for Standard Cells," in *Proc. of 25th DAC*, Jun. 1988, pp. 189–195.

[10] N. Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic, Boston, MA, 1999.