The Grid-Based Two-Layer Routing Algorithm Suitable for Cell/IP-Based Circuit Design*

Chia-Jung Liu, Yi-Chen Lin, and Jiann-Chyi Rau
Department of Electrical Engineering, Tamkang University
151 Ying-Chuan Road Tamsui, Taipei County Taiwan 25137, R. O. C
Email: {cjliu, yclin, jcrau}@ee.tku.edu.tw

***Abstract* — In this work, we employ gridded model for channel routing and place the terminals which are horizontally aligned. We have developed a two-layer channel router that can eliminate the constraints due to overlap. The proposed approach is suitable for cell/IP-based channel-less circuit with a few channels. Our developed tool can route the nets in nearly linear time achieving to the advantage of time to market, and lead to the area overhead of 6.34% increase in average. The area overhead results from the space insertion, and we also have shown that the proposed algorithm can achieve 100% routing on most ISCAS'85 benchmarks. In addition, the number of channel tracks can be minimized by our algorithm.**

I. Introduction

Integrated circuit technology has developed in the 1960s from the integration of a few transistors currently in use and the integrated circuits contain more than 10^5 transistors. Nowadays, the number of transistors in circuits grows rapidly and the circuit design becomes a very complex and difficult task. Due to large number of components, the physical design must be helped with computers. The phases of physical design extensively use computer-aided design tools [2], and many phases have already been partially or fully automated. There are different targets that one would like to optimize. Our work focuses on how to carry out routing completely within reasonable time.

Routability is a key factor of digital integration circuits design and a positive aspect of the former standard cell design style [3] based on routing channels. The standard cell architecture (figure 1) consists of rectangular cells with the same height, which are placed in several rows, and the space between two rows being called channel.

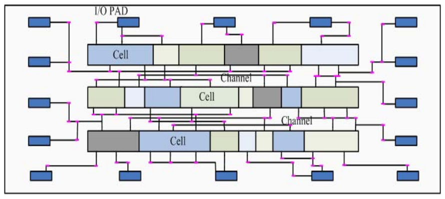

Figure 1. Standard cell architecture.

*This work was supported by the National Science Council of R.O.C. under contract NSC-96-2221-E-032-058.

Routing model can divide into two models, one is called grid-based model (figure 2(a).) and the other is called gridless-based model (figure 2(b).) [4].

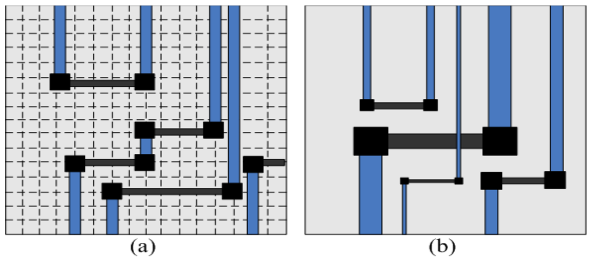

Figure 2. Routing model (a) Grid-based model. (b) Gridless-based model.

In this work, we employ grid-based routing model for channel routing. Using grid-based routing model, we can make placement and routing more easily. In the placement phase, we have to place the cells on the intersection points of the grids and then route the nets on the edges of the grids.

II. Preliminary

Detailed routing [5] is one of the most fundamental steps in physical design cycle. The channel routing is the important part of detailed routing, which is executed after placement.

A. *Channel routing model*

The channel routing model [6] (figure 3) consists of several parts of components. The nets of terminals are placed on upper and lower boundaries. The channel to be routed is defined by a rectangular region with two rows of terminals along its top and bottom sides. The horizontal metal layer is called trunk and the vertical metal layer is called branch. When horizontal metal layer and vertical metal layer are needed to connect and we must penetrate two layers with via. Combining channel routing model with grid-based model ca can be able to make the physical design easier and the design time shorter.

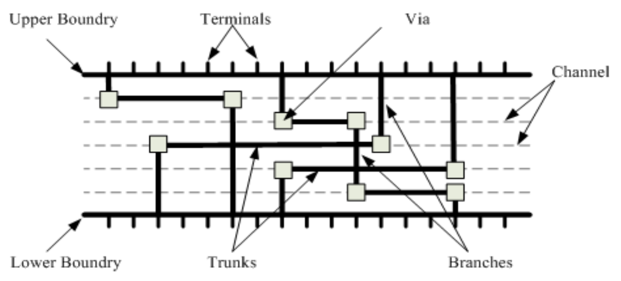

Figure 3. Channel routing model.

B. *Channel routing problem*

The main target of the channel routing is to minimize the channel height [7][8]. In real designs, each channel is assigned fixed height by the floorplanner and the task of channel router is to complete the routing within the assigned height. If channel router can not achieve 100% routing in the assigned height, channel router must to expand the channel, which changes the floorplan. This requires routing the channels in a predefined order, so that such expansions can be accommodated, without impact on the floorplan.

C. *Left-edge algorithm*

The left-edge algorithm [9] is applicable to channel routing problems which do not allow doglegs [10] and vertical constraints. In other words, it does not appear cycles in vertical constraint graph [6][9]. The left-edge algorithm sorts the intervals that are the horizontal line segments of the nets, in ascending order, relative to the *x* coordinate of the left end point of intervals. And then it allocates a track to each of the intervals. To allocate an interval to a track, left-edge will scan through the tracks from the top to the bottom and assigns the net to the first track that can adapt the net. The allocation process is restricted to one layer since the other layer is used for the vertical line segments of the nets. The figure 4 is the left-edge algorithm.

Left-edge Algorithm

```
00  Begin
01      Sort all nets on their leftmost positions;
02      Select the next net n with the lowest left-end position;
03          If (n has no descendents in VCG)
04              then begin
05                  Place n on the lowermost available track;
06                  Delete n from the sorted list;
07                  Delete n from VCG;
08              end
09          Else Goto 02
10          End If
11      Continue scanning the sorted list and from it select those nets
        which do not overlap with nets assigned to this track and have
        no descendents in VCG;
12      Remove all selected nets from the list;
13      If list ≠ Ø then Goto 02;
14      Exit
15  End
```

Figure 4. The left-edge algorithm.

III. The Extensible Channel Router

The terminals of cells are needed to place before routing. Therefore, beginning of routing we must know the cells that are placed in certain positions. In our work, we placed the terminals of cells with two rows and used the region between the two rows to route the nets of circuit.

We use a simple placement algorithm [11] to place the terminals of cells. First, we placed the input pins of circuit on the left upper row and placed the output pins of circuit on right lower row possibility.

A. *Horizontal constraint graph*

After placement the terminals of circuit, we can know the positions of each terminal. Therefore, we can compute every horizontal line segments. The first step, we will find the leading terminal of each net. Then, we will find the last terminal of each net. And we can record the two data of each net and store it in the data structure. We will take the data structure to operate the following steps.

According to the data structure, the horizontal constraint graph will be set up by the information of each net. The figure 5 is an example of horizontal constraint graph [6][9]. We do not consider the order of the horizontal line segments. Therefore, we can set up the horizontal constraint graph using undirected graph.

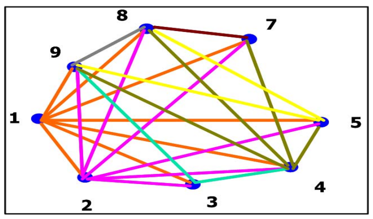

Figure 5. Example of the horizontal constraint graph (HCG).

B. *Vertical constraint graph*

In order to avoid the vertical constraints and we will set up the vertical constraint graph before routing. Consider the terminals which include upper and lower rows, we can regard them as many pairs of columns. The zero numbers represent the space of the placement. We will find the upper and lower terminals of one pair that are not contained zero numbers. Then, we will consider these pairs of terminals to set up the vertical constraint graph.

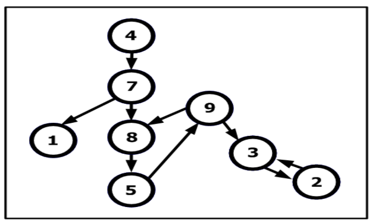

Figure 6. Example of the vertical constraint graph (VCG).

After finding these terminals, we will set up the vertical constraint graph (figure 6). We need to use directed graph to represent the vertical constraint graph. Because of the upper terminals must be above the lower terminals. We use the directed graph to represent the vertical constraint graph different from horizontal constraint graph.

C. *The extensible router*

After left-edge algorithm and have been set up the constraint graphs we will get the preliminary routing result which is including the violations of vertical constraints. We will look table up in vertical constraint graph and try to break the cycles in the vertical graph.

1) *Insertion of spaces*

Now, we face the problem that is violations of vertical constraints. We will insert pairs of spaces near by the terminals which are violations of vertical constraints. Then we will try to find the remnant tracks to route the unrouted terminals. If there are not the remnant tracks to route the terminals, we will add extra track in the most top of the channel to route the unrouted terminals. The figure 7(a) is an example of preliminary routing result with vertical constraints. There are two violations of vertical constraints in the example. Now, we face the problem that is violations of vertical constraints. In the figure 7(b), we will insert pairs of spaces near by the terminals which are violations of vertical constraints. Take figure 7(b) for example, we will find the second track to route the two terminals which are violations of vertical constraints.

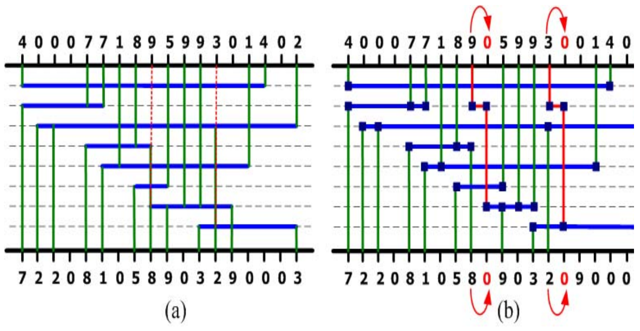

Figure 7. The example of after left-edge algorithm (a) Preliminary routing with the vertical constraints, (b) Solution of vertical constraint violations.

2) *Update the data structure*

After insert the spaces, we need to update the data structures, which are the sites of terminals and horizontal line segments. The first, we must to recompute the sites of terminals and according to these sites to change the data structure of horizontal line segments. Now, we get a new solution of routing and it can achieve 100% routing [12]. The figure 8 shows the insert space algorithm.

Insert Space Algorithm

```
Begin
  Find all the vertical constraints from VCG;
  Record the terminals which are violated vertical constrain;
    For (i=0 to i= count of terminals -1 ) Do
        insert pair of spaces near by the terminal;
        update the site of terminal;
      If (exist redundant tracks to route the unrouted terminals)
        then begin
          re-route the terminal on the redundant;
          update the date structure of tracks;
        end
      Else
        then begin
          insert a new track at top;
          re-route the unrouted terminal at top track;
          update the date structure of tracks;
        end
      End IF
    End For
End
```

Figure 8. (a) Two cases that we will optimize. (b) After optimizing the horizontal line segments.

D. *Channel tracks optimization*

In our experiments, we will divide the condition into two cases. One case is end of the upper line segment corresponding with head of the lower line segment and the other case which is head of the upper line segment corresponding with end of the lower line segment. Figure 9(a) shows the two cases. Before optimizing the segments, we must to know data structure of each horizontal line segments that record coordinates of all segments. Then our algorithm will find the two cases to reduce the channel tracks. Figure 9(b) shows the horizontal line segments after optimizing and then we reduce the channel tracks from two to one.

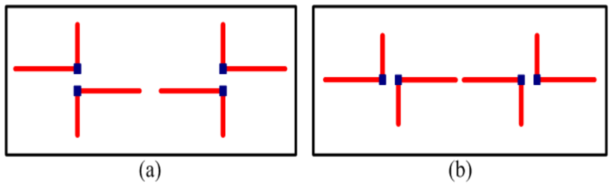

Figure 9. (a) Two cases that we will optimize. (b) After optimizing the horizontal line segments.

Further, we will extend more cases to reduce the channel tracks. The two cases that are the upper horizontal line segment overlapped the lower horizontal line segment, if the two line segments in the same channel track. The figure 9 shows the extension of the idea. The figure 10(a) is more one contact than original case and the figure 10(b) is more two contacts than original case.

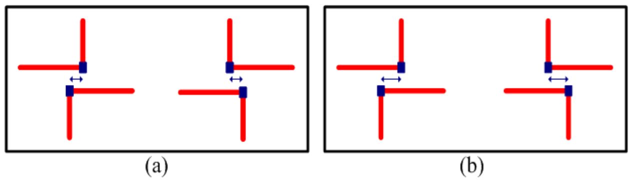

Figure 10. The extended cases (a) Overlap one contact. (b) Overlap two contacts.

In the second and third cases, we will insert two and three pairs of zero terminals to shift the horizontal line segment respectively.

IV. Experimental Results

The extensible router was compared to the ChAOS router [13]. We performed our experiments on some iscas85 benchmarks [14], which are combinatorial logic circuits.

We can observe the sixth column of table 1, the extra terminals that we need to add to solve the vertical constraints. Extensible router is better than ChAOS router that we can reduce the area overhead and the two routers utilized the same channel tracks. The percentages of area overhead are compared with the original terminal numbers. We gather statistics the difference of area overhead between the two routers in the last column of Table I. Our approach can reduce 3.34% in average of the total area than ChAOS router.

TABLE I. .

Comparison of area overhead.

Circuits		ChAOS router [13]			Our router			Difference in overhead between two router
Name	Pairs of terminals	Add extra terminals	Utilize tracks	Area overhead (%)	Add extra terminals	Utilize tracks	Area overhead (%)	
c17	9	2	11	22.22%	1	11	11.11%	(-) 11.11%
c432	253	28	196	11.06%	21	196	8.30%	(-) 2.76%
c499	300	40	243	13.33%	29	243	9.66%	(-) 3.67%
c880	514	60	441	11.67%	45	441	8.75%	(-) 2.92%
c1355	802	106	587	13.20%	72	587	8.97%	(-) 4.23%
c1908	1189	73	912	6.13%	51	912	4.28%	(-) 1.85%
c3540	2140	88	1717	4.11%	60	1718	2.80%	(-) 1.31%
c5315	2929	160	2453	5.40%	92	2453	3.14%	(-) 2.26%
c6288	3608	3	2447	0.083%	2	2447	0.05%	(-) 0.033%

The table II shows the extra terminals that we need to add after horizontal segment optimization. Besides, the Table II also shows the channel tracks that we need to route the circuits. In several cases we can reduce the channel tracks effectively and the results are showed in each case of tracks column.

V. Conclusions and Future Work

In recent years, the advance of COMS technology has led to a great development, especially on the complexity of the digital circuits. In this work, we have investigated the grid-based model and then used this model to solve the channel routing problem. We apply the presented router on some ISCAS'85 benchmark circuits in which the associated algorithm is implemented in C++ language. The results show that our router can achieve 100% routing and reduce a lot of routing area than before. In the future work, we will use the mentioned concept to deal with the crosstalk between multi-metal layers, and also extend channel routing to switchbox routing.

TABLE II.

The results of horizontal segment optimization.

Circuits	Without Horizontal segment optimization		Horizontal segment optimization								
			Case 1			Case 2			Case 3		
Name	Terminals	Tracks	Add extra terminals	Tracks	Reduction of tracks(%)	Add extra terminals	Tracks	Reduction of tracks(%)	Add extra terminals	Tracks	Reduction of tracks(%)
c17	10	11	1	11	0%	1	11	0%	14	7	(-) 36.36%
c432	274	196	18	196	0%	34	196	0%	51	196	0%
c499	329	243	14	243	0%	66	243	0%	132	239	(-) 1.64%
c880	559	441	40	440	(-) 0.22%	80	440	(-) 0.22%	189	422	(-) 4.3%
c1355	874	587	14	586	(-) 0.17%	133	586	(-) 0.17%	295	577	(-) 1.7%
c1908	1240	912	33	912	0%	121	912	0%	186	912	0%
c3540	2200	1718	38	1718	0%	138	1718	0%	399	1640	(-) 4.54%
c5315	3021	2453	60	2452	(-) 0.04%	201	2452	(-) 0.04%	595	2349	(-) 4.23%
c6288	3610	2447	1	2447	0%	5	2447	0%	5	2447	0%

References

[1] M. J. Bass and C. M. Christensen, "The future of the microprocessor business", *IEEE Spectrum*, Vol.39, April 2002, pp. 34-39.

[2] M. E. Daniel and C. W. Gwyn, "CAD Systems for IC Design", *Computer-Aided Design of Integrated Circuits and Systems*, Vol.1, January 1982, pp. 2-12.

[3] H. Fischer and W. Pschunder, "Low-cost solar cells based on large-area unconventional silicon", *Electron Devices,* Vol.24, April 1977, pp. 438-442.

[4] H. H. Chen and E. S. Kuh, "Glitter: A Gridless Variable-Width Channel Router", *International Journal of Science and Technology*, Vol.5, October 1986, pp. 59-465.

[5] E. S. Kuh and T. Ohtsuki, "Recent advances in VLSI layout", *Proceedings of the IEEE*, Vol.78, February 1990, pp. 237-263.

[6] H. W. Leong and C. L. Liu, "Discretionary channel routing", *IEE Proceedings-Circuits, Devices, and Systems*, Vol.135, April 1988, pp. 45-57.

[7] C. Yang and D. F. Wong, "Optimal via-shifting in channel compaction", *Design Automation Conference*, March 1990, pp. 186-190.

[8] U. Choudhury and A. Sangiovanni-Vincentelli, "Constraint-based channel routing for analog and mixed analog/digital circuits", *ICCAD-90. Digest of Technical Papers*., November 1990, pp. 198-201.

[9] T. Yoshimura, "An Efficient Channel Router", *21st Design Automation*, June 1984, pp. 38-44.

[10] M. M. Wada, "A Dogleg Optimal Channel Router with Completion Enhancements", *18th Design Automation*, June 1981, pp. 762-768.

[11] C. P. Hsu, "General River Routing Algorithm", *20th Design Automation*, June 1983, pp. 578-583.

[12] R. L. Rivest and C. M. Fiduccia, "A Greedy Channel Router", *19th Design Automation*, June 1982 6, pp. 418-424.

[13] G. B. V. dos Santos, M. de Oliveira Johann, and R. A. da Luz Reis, "Channel based routing in channel-less circuits", *Circuits and Systems*, May 2006, pp. 4.

[14] http://www.fm.vslib.cz/~kes/asic/iscas/