Journal of Physical Sciences, Vol. 11, 2007, 172-184

A Revisit to YACRIT: Yet Another Channel Router with Interchangeable Terminals

Rajat K. Pal

Department of Computer Science and Engineering
University of Calcutta
92, A. P. C. Road, Kolkata – 700 009, India

Received December 12, 2007; accepted December 15, 2007

ABSTRACT

This paper presents new algorithms for routing two rows of interchangeable terminals across a two-layer channel. In this case, the number of horizontal tracks required for routing is reduced significantly by simply interchanging the terminals in each cell. It has been found that on the average approximately 40% channel area or often more is saved just by interchanging the terminals. In practice, actually, on the design table of forming the final net list from a net list just given in the form of rows of terminals necessary for routing, or partially constructed channel instances are given for rearranging their terminals so that area required for routing is minimized. The number of horizontal tracks per net is assumed to be one, i.e., no-dogleg routing is performed, and subsequently with the help of each of the algorithms developed in this paper at least one feasible solution is always computed without doglegging. A generalized study is also encountered considering existence of more than two cells on a side of the given channel where cells are fixed at their relative positions though the terminals within a cell are interchangeable; intercell interchanges of terminals are not allowed.

Key words: *VLSI design, Channel routing problem, Net list, Two-layer routing, No-doglegging, Interchangeable terminals, Area minimization, Algorithm.*

1. Introduction

In the process of VLSI physical design, channel routing is an important area, and actually, it is difficult also in computing a routing solution of minimum area. Various algorithms have been developed in computing a feasible channel routing solution [1, 2, 6-9, 12]. However, almost all the routing algorithms assume that the terminals are fixed at their pin locations along the length of the channel in each cell. For which researchers had no scope to change the terminal positions, and they were compelled to work with the given net list. With this initial consideration they tried to follow different techniques such that the number of horizontal tracks is minimized, as much as possible. But it was impossible to minimize the number of horizontal tracks below the maximum density of the channel. Moreover, the interval graph (or the horizontal constraint graph (HCG)) and the vertical constraint graph (VCG) were considered simultaneously for the whole net list. For a given net list the minimum number of horizontal tracks required is greater than or equal to the maximum of the clique number of the interval graph (i.e., the HCG) and the longest path length of the VCG. Furthermore, for the presence of cyclic constraints in the VCG, the vertical constraint violation (VCV) is occurred, and to resolve it doglegging is introduced, for which the number of horizontal tracks required is increased.

In short, all of these complex situations make the channel routing problem to be a very hard problem; in general, beyond polynomial time computable [5, 6, 10, 11]. So, on the design table of forming a final net list from the net list just given in the form of terminals necessary for routing, or partially constructed channel instances whose terminals are need to be rearranged in such a way that the area required for routing from the final net list is significantly reduced. Of course, in the case of interchangeable terminals the number of vertical columns remains same, and the channel area is reduced by minimizing the number of horizontal tracks only (for a modified final net list). Again, programmable logic cells (e.g., ROMs and PLAs) are widely used in VLSI design by reason of their structural regularity and design flexibility [4]. Since their geometries are programmable, the terminals of these cells are interchangeable.

This paper describes efficient algorithms for determining the reference side, and aligning the terminals to get optimal results following *merging of nets* [12]. These algorithms are integrated into a channel router for interchangeable terminals (see Section 5). We have implemented our algorithms using the example channel instances considered in [3, 4, 9, 12]. In all these cases our algorithms yield better results.

The paper is organized as follows. In Section 2, we define and explain some relevant terms. The reference side of a channel is determined in Section 3, and in Section 4, we develop the algorithm for aligning the terminals along the length of the channel. All the algorithms developed in this paper are included in Section 5. We illustrate the newly developed algorithms with the help of some example channel instances, in Section 6. The paper is concluded in Section 7, with some remarks relevant to the work and probable open problems.

2. Some Definitions

In this section, we define some important terms and explain, whenever necessary that are associated to the problem under consideration and the algorithms developed in this paper.

(i) ***Net list*****:** This is a list of net numbers that explicitly gives two sets of terminals of different nets that are belonging to two opposite sides of a channel. A net list contains a set of top terminals and a set of bottom terminals of a channel that are aligned vertically. A given net list is the net list that is considered as input to the algorithms developed in this paper for the alignment of net terminals, and a final net list is the net list that is generated from the given net list for computing a routing solution of reduced channel area.

(ii) ***Channel area*****:** This is a rectangular routing region that contains two sets of terminals located on the top and at the bottom sides of it and that is used for routing the nets available in the given net list. One of the main objectives of channel routing problem is to minimize its area required for routing. If the number of rows (or tracks) required and the number of columns present in a routing solution are p and q, respectively, then the channel area required for routing is $p \times q$.

(iii) ***Channel density*****:** The *local density*, d of column j is the number of nets crossing that column. The *channel density*, k of a given channel is defined as the maximum of d, i.e., $k = max(d)$, where $1 \leq j \leq q$, where q is the number of columns belonging to the channel.

(iv) ***Interval graph*****:** In an interval graph there are n nodes for n nets in a channel, and two nodes i and j are connected by an edge, if and only if the two horizontal spans (or intervals) of these two nets overlap in a column, where $1 \leq i, j \leq n$. For these two nets two different tracks are required for their assignment in the reserved two-layer channel [6, 12]. This interval graph is also known as the *Horizontal Constraint Graph* (*HCG*) of the channel.

(v) ***Vertical Constraint Graph*** **(*VCG*):** Vertical constraints specify the ordering of nets for their assignments to tracks from top to bottom along the height of the channel. Initially, in the VCG, n isolated nodes are introduced. A directed edge (i, j) is introduced into the VCG, only if there is a column in the given channel with a terminal of net i on the top and a terminal of net j at the bottom, where $j \neq i$, and $1 \leq i, j \leq n$. It implies that the horizontal span of net i must be placed above to that of net j in any feasible two-layer channel routing solution [6, 12].

(vi) ***Doglegging*****:** If the horizontal span (or interval) of a net is split into two or more parts for their assignment to different tracks in a feasible routing solution, then it is called a *dogleg routing solution*, or *doglegging*. A *no-dogleg routing solution* does not contain any net of that sort, and the horizontal span of each of the nets belonging to a feasible routing solution is assigned to a track only. In general, doglegging introduces more via holes, though sometimes doglegging helps in computing a feasible solution that is not possible in no-dogleg routing or often doglegging helps in computing a routing solution of reduced channel area [6].

(vii) ***Vertical Constraint Violation*** **(*VCV*):** In the case of a cyclic vertical constraint graph (or cyclic VCG) for some channel instance, the *vertical constraint violation* occurs. It implies that as if the horizontal span (or interval) of a net corresponding to the node present in the cyclic loop have to be placed above or below to that of itself. In the case of a channel with cyclic vertical constraint in the VCG (i.e., with vertical constraint violation), often doglegging helps in computing a feasible routing solution; without doglegging no solution is feasible in the reserved two-layer channel routing.

(viii) ***Active column*****:** An *active column* is defined as a column that contains the terminals of two different nets, or a terminal and a non-terminal on both the sides of the channel. If either the terminals present along a column are of the same net or both are non-terminals, then the column is known as a *non-active column*.

3. Determination of Reference Side

The *reference side* of a channel is that side, which remains unaltered, that means the relative positions of the terminals are kept unchanged during the algorithm is processed. With respect to the reference side the terminals present on the other side of the channel, called the *opposite side*, the interchanging of terminal positions are made. For the determination of the reference side a terminal list is generated. It contains the number of terminals present separately on either side of the channel for each of the nets present in the given net list. If for a particular net all of the terminals are present on one side only, initially that net is not to be considered in order to determine the reference side. Then all the terminals of the remaining nets on each side of the channel are added separately. If the summations of number of terminals for both the sides of the channel are equal, the top one is assumed to be the reference side with respect to the other; else the side consisting of lesser number of terminals is considered as the reference side.

For example, consider a net list given below whose terminal list is computed after the net list.

0 1 4 5 1 6 7 0 4 9 10 10
2 3 5 3 5 2 6 8 9 8 7 9

Nets	1	2	3	4	5	6	7	8	9	10
Total number of top terminals	2	0	0	2	1	1	1	0	1	2
Total number of bottom terminals	0	2	2	0	2	1	1	2	2	0

Now, for the determination of the reference side, the total number of top terminals is equal to four, and the total number of bottom terminals is equal to six (for nets 5, 6, 7, and 9 only). Therefore, according to this algorithm the top side is determined as the reference side.

4. Alignment of Terminals

In order to generate a new net list whose area requirement is less from a given net list that usually requires more area, it is necessary to have as many *straight nets* (nets that are having a terminal on the top and a terminal at the bottom along a column of the channel) as possible, since they result the problem simpler by making more non-active columns. In addition, if a net contains equal number of top and bottom terminals and if it is equal to one, then after processing this algorithm no horizontal span for that particular net is required.

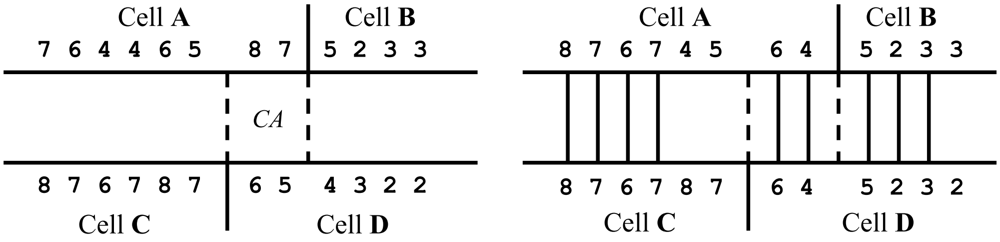
By interchanging terminals in each cell, the number of straight nets can be increased, and by this means the number of horizontal spans can be reduced a lot. Figure 1.(a) shows a bit more general case of routing a channel, where Cells **A**, **B**, **C**, and **D** are the cells with interchangeable terminals. Here in the figure, *CA* is the critical area, which is defined in [3]. Cell **A** in the upper row shares a common routing region with Cell **D** in the lower row. The common region below the right side of Cell **A** and above the left side of Cell **D** is defined as a *critical area* (*CA*). Assuming that A, B, C, and D are the sets of terminals in Cells **A**, **B**, **C**, and **D**, respectively. The set of terminals that should be aligned in *CA* is then given by the expression

$$F = A \times (D - C)$$

or

$$F = D \times (A - B),$$

where, "×" and "–" represent intersection and subtraction set operators, respectively. According to the Figure 1.(b), terminals of nets 4 and 6 are aligned in *CA*, since F is equal to the set of net numbers $\{4, 6\}$.



(a) **(b)**

Figure 1. Two rows of interchangeable terminals. **(a)** Before alignment of nets, and **(b)** After alignment of nets.

As mentioned earlier, for alignment of terminals, net terminals on the opposite side of the channel are only interchanged. One terminal on the reference side is pointed to, and if it is on an active column and there is any terminal of the pointed net on the opposite side, then the referenced column is made non-active with the help of an active column only. Subsequently, the obvious lemmas are as follows.

Lemma 1: The algorithm behind the *Alignment of Terminals* helps in obtaining an HCG computed from the final net list, which is smaller in terms of constraints present among the nets in comparison to the HCG from the given net list of the channel. In the worst case, these two HCGs might be the same but never worse in terms of amount of horizontal constraints among the nets.

Lemma 2: The algorithm behind the *Alignment of Terminals* helps in obtaining a VCG computed from the final net list, which is smaller in terms of the length of the longest path among the nets in comparison to the VCG from the given net list of the channel. In the worst case, these two VCGs may have the same structure but never a final VCG contains a longest path length, which is longer than the longest path length in the VCG of the given channel.

Lemma 3: The algorithm behind the *Alignment of Terminals* helps in obtaining a channel, which is free from any vertical constraint violation (VCV), even if the given net list may result a cyclic vertical constraint graph (VCG).

5. The Algorithm

5.1. Basic Ideas

Since the channel routing problem is known to be NP-complete [5, 6, 10, 11], heuristic algorithms are necessary to develop for computing a routing solution. As it has been stated in the previous sections that the main goal of a channel router is area minimization, it is achieved by minimizing the number of rows (or tracks) required for the layout.

Here we present a new heuristic method that always uses not more than *k* rows in computing a routing solution, where *k* is the channel density, without adding any extra column beyond the length of the channel.

Note that a router based on the ideas mentioned above has the following properties.

i) The channel length remains unaltered.

ii) The channel density of the resulting net list is less than (or at most equal to) the channel density of a given net list.

iii) A solution without any VCV is always possible, even if a cyclic vertical constraint exists in the VCG of a given net list.

iv) The reference side of a given channel remains unchanged during the algorithm being processed.

v) All changes of terminal positions are performed on the opposite side of a given channel only.

vi) The operation behind the algorithm of *Alignment of Terminals* is performed for the presence of respective active columns only.

vii) The algorithm usually follows *left-to-right* (*LR*) approach, unless and otherwise, it is mentioned.

5.2. The *Net Alignment Algorithm* (*NAA*)

Input: Net list of the given channel.

Output: Modified net list of the given channel after interchanging terminals.

Phase 1: ***Algorithm for Determining the Reference Side*** **(*ADRS*)**

Step 1: Generate the terminal list.

Step 2: Primarily consider the nets having terminals on both the sides of the given channel.

Step 3: Add all the terminals in the terminal list for both the sides separately, obeying *Step 2*.

Step 4: Compare the added numbers. Identify the side having smaller number of terminals as the reference side, or assume the top one as the reference side if the summations are equal.

Phase 2: ***Algorithm for Aligning the Terminals*** **(*AAT*)**

Step 5: **Procedure** ***AAT*()**;

```
        var     i: integer;
                j: integer;
begin
  for i := 1 to m do {m is the number of columns of the given channel}
     begin
        if (the reference side has a terminal of some net at column i)
         and
         (the top and the bottom terminals at column i are not the same) then
           begin
              for j := 1 to m do
                 begin
                 if (the terminal at column j on the opposite side is same as the terminal at
                 column
                    i on the reference side)
                    and
                    (the top and the bottom terminals at column j are not the same) then
                    interchange terminals at column i and at column j on the opposite side;
                 end;
          end;
     end;
end;
```

6. A Few Examples

Different example channel instances are taken from literature [3, 4, 9, 12] as standard problems to show the results of the algorithms developed in this paper.

Example 1. This is an example for which 100% reduction of the channel area is achieved, as each of the nets present in the net list given below is a two-terminal net and for each of them one terminal is in the top net list and the other in the bottom net list. The net list is as follows where 0 is a non-terminal, not to be connected.

1 2 5 3 0 4 0 6 0 7 8 9 A B C
9 4 2 8 6 1 0 7 0 5 B 3 C A 0

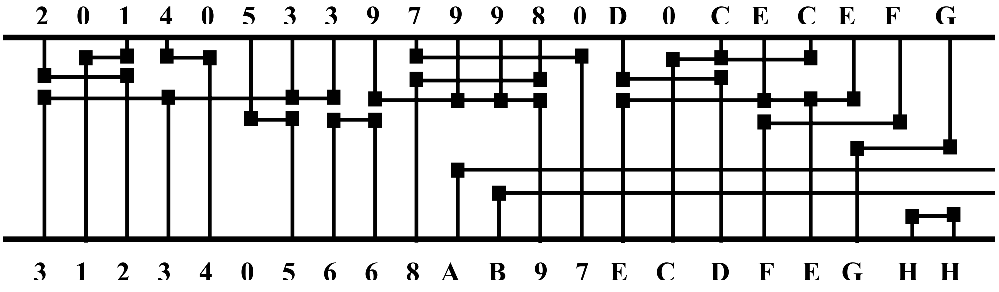
In this case the top one is the reference side, and the possible interchanges of terminals are made on the bottom side (i.e., on the opposite side) only that are shown below.

1	**2**	**5**	**3**	**0**	**4**	**0**	**6**	**0**	**7**	**8**	**9**	**A**	**B**	**C**
~~9~~	~~4~~	~~2~~	~~8~~	~~6~~	~~1~~	0	~~7~~	0	~~5~~	~~B~~	~~3~~	~~C~~	~~A~~	~~0~~
1	2	~~4~~	3	~~7~~	~~9~~		6		~~4~~	8	~~8~~	A	~~C~~	C
		5		~~9~~	4				~~9~~		~~B~~		B	
				~~B~~					7		9			
				~~C~~										
				0										

Example 2. Consider the following net list where a net may contain more than one top terminals and more than one bottom terminals; zeros are non-terminals, not to be connected.

2 0 1 4 0 5 3 3 9 7 9 9 8 0 D 0 C E C E F G
3 1 2 3 4 0 5 6 6 8 A B 9 7 E C D F E G H H

The channel (with one feasible solution) and its VCG are shown in Figures 2.(a) and 2.(b), respectively. The *left-to-right* (*LR*) approach and the *right-to-left* (*RL*) approach of alignment of terminals are presented in the following two subsections. In this case the total number of top terminals is 18, whereas the total number of bottom terminals is 15 only, identifying the bottom side of the channel as the reference one.



(a)

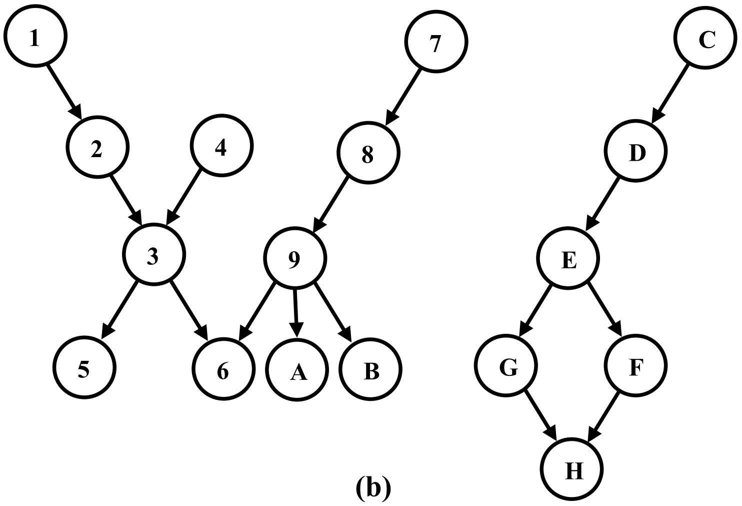
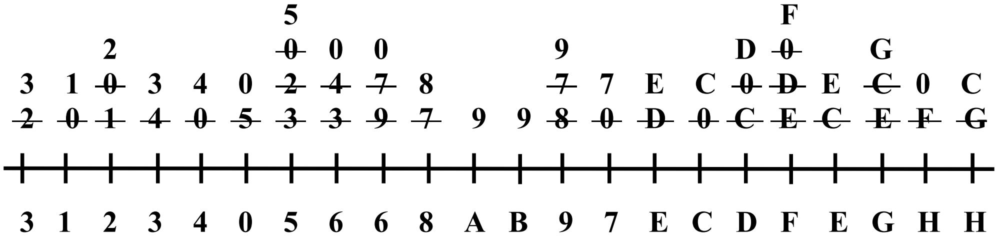


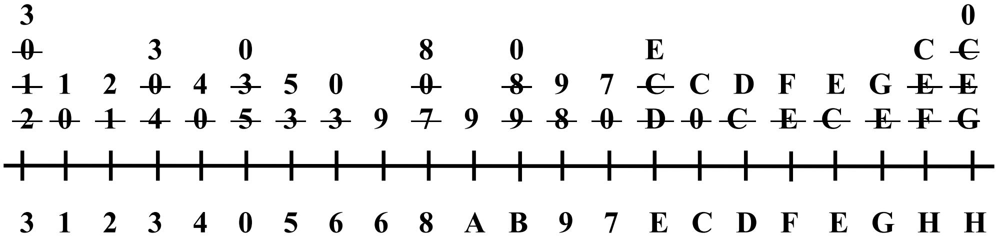
Figure 2. A many-to-many-net routing example. **(a)** The channel with a feasible routing solution of eight tracks, and **(b)** The VCG of the channel.

6.1. The *Left-to-Right (LR)* Approach

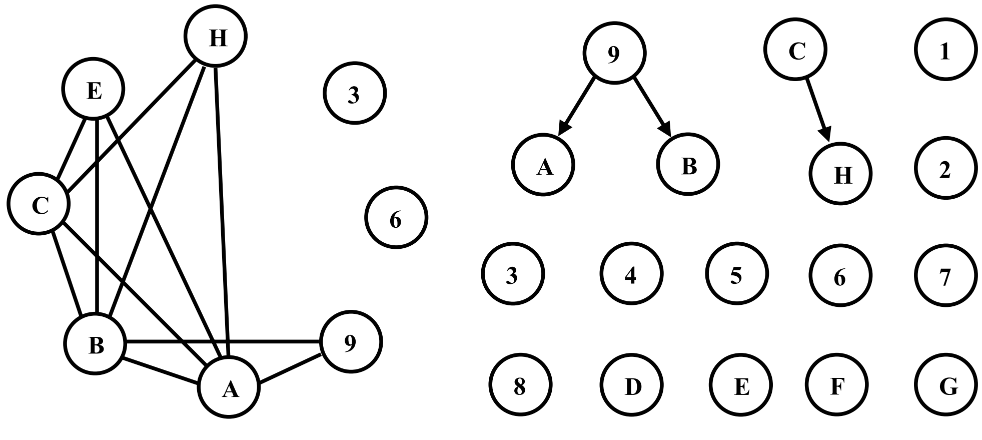


The HCG and the VCG of the modified channel are shown in Figures 3.(a) and 3.(b), respectively. Figure 4 shows an optimal routing solution with 50% reduction in channel area, after alignment of terminals scanning from left to right along the length of the channel.

6.2. The *Right-to-Left (RL)* Approach



The HCG and the VCG of the modified channel are shown in Figures 5.(a) and 5.(b), respectively. Figure 6 shows an optimal routing solution with 50% reduction in channel area, after alignment of terminals scanning from right to left along the length of the channel. Note that in both the cases, the number of comparisons and interchanges is 30 for this channel of length 22.



(a) **(b)**

Figure 3. (a) The HCG and **(b)** the VCG of the modified net list obtained using *LR* approach.

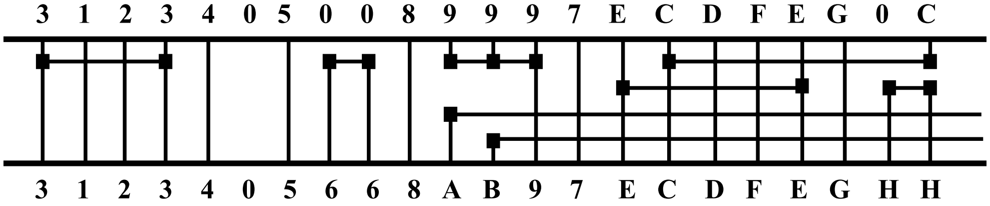
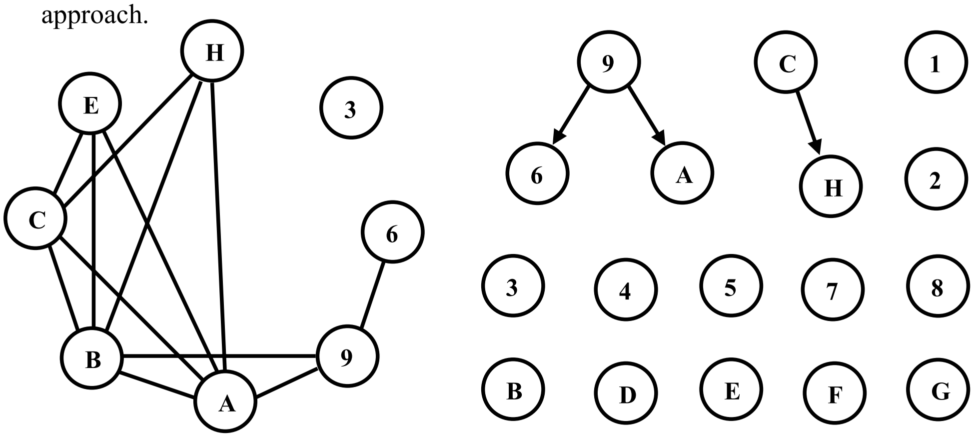


Figure 4. An optimal routing solution of the modified net list obtained using *LR* approach.



(a) **(b)**

Figure 5. (a) The HCG and **(b)** the VCG of the modified net list obtained using *RL* approach.

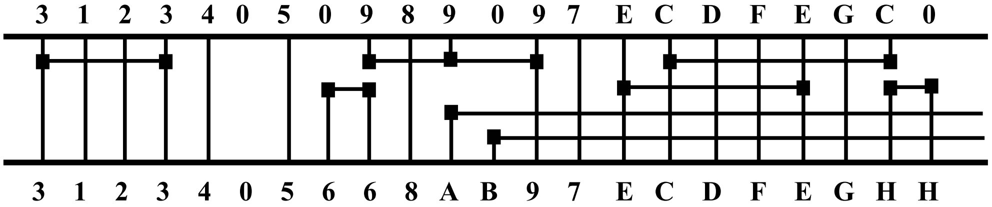
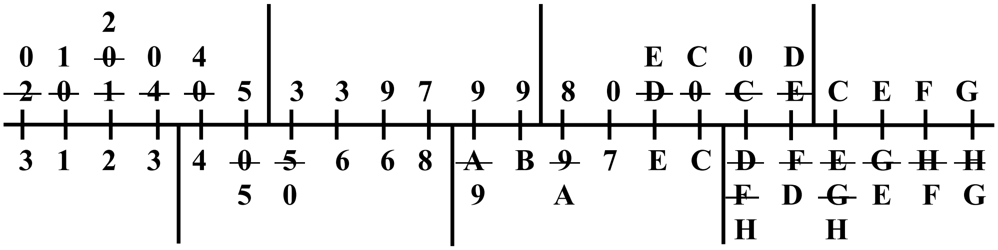
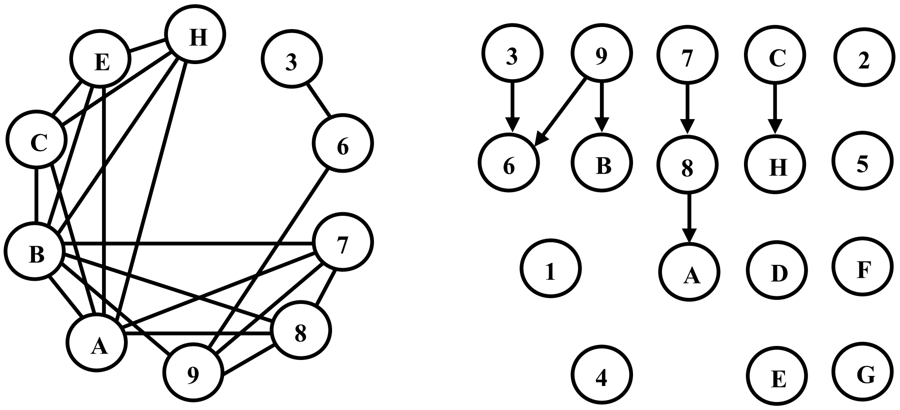


Figure 6. An optimal routing solution of the modified net list obtained using *RL* approach.

Examples 3. This example consists of a net list that is actually obtained from a number of cells placed side-by-side for a channel routing problem, and in this case, the terminals are interchangeable only within the span of terminals of a cell. That means, intracell interchanges of terminals are allowed but intercell interchanges of terminals are not allowed.





(a) **(b)**

Figure 7. (a) The HCG and **(b)** the VCG of the finally obtained net list (using *LR* approach), where only intracell interchanges of terminals are allowed.

Primarily, the above problem is no longer different so far the channel routing problem is concerned with respect to the given net list considered in the second example. The Figures 7.(a) and 7.(b) show the HCG and the VCG, respectively, for the finally computed net list. Figure 8 shows a routing solution of this modified net list that requires 37.5% less channel area in comparison to the channel area required for the given net list, as shown in Figure 2.(a). Note that Lemma 3 is applicable for a general case like this as well, though removal of VCV in a general case is still a problem to be investigated further.

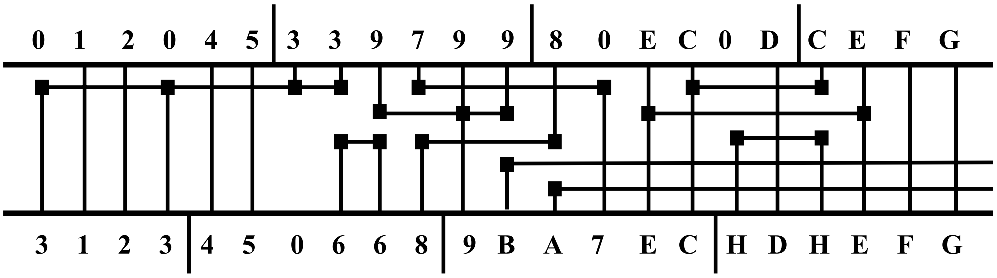


Figure 8. An optimal routing solution of a generalized case of net list where intercell interchanges of terminals are not allowed.

7. Conclusion

This paper presents new algorithms for routing two rows of interchangeable terminals across a reserved two-layer channel. In this case, the number of horizontal tracks required for routing is significantly reduced by simply interchanging the terminals belonging to either on the top or at the bottom net list of the given channel. Through experimentation, it has been found that approximately 40% channel area or more is often saved just by interchanging the terminals.

Programmable logic cells like ROMs, PLAs, etc. are widely used in VLSI design by reason of their structural regularity and design flexibility. As their geometries are programmable, the terminals of these cells are interchangeable. On the other hand, actually in practice, on the design table of forming the final net list from a net list just given in the form of rows of terminals necessary for routing, or partially constructed channel instances are given for rearranging their terminals so that area required for routing is minimized. A generalized study is also encountered considering existence of more than two cells on a side of the given channel where cells are fixed at their relative positions though the terminals within a cell are interchangeable; intercell interchanges of terminals are not allowed. We strongly feel that the generalized study still demands an in-depth research, as vertical constraint violations are not completely removed in this case.

REFERENCES

[1] Hashimoto A. and J. Stevens, Wire Routing by Optimizing Channel Assignment within Large Apertures, Proc. of 8th ACM Design Automation Workshop, pp 155-169, 1971.

[2] Ho T.-T., S. S. Iyengar and S.-Q. Zheng, A General Greedy Channel Routing Algorithm, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol 10, pp 204-211, 1991.

[3] Kobayashi H. and C. E. Drozd, Efficient Algorithms for Routing Interchangeable Terminals, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol 4, pp. 204-207, 1985.

[4] Kobayashi H. and Y. P. Foo, Programmable Logic for Parallel Convolution, IEEE Intl. Conf. on Computer Design: VLSI in Computers, Port Chester, NY, pp. 700-704, 1984.

[5] LaPaugh A. S., *Algorithms for Integrated Circuit Layout: An Analytic Approach*, Ph.D. Thesis, Lab. for Computer Sc., Massachusetts Inst. of Tech., Cambridge, 1980.

[6] Pal R. K., *Multi-Layer Channel Routing: Complexity and Algorithms*, Narosa Publishing House, New Delhi (also published from CRC Press, Boca Raton, USA and Alpha Sc. Intl. Ltd., UK), 2000.

[7] Pal R. K., A. K. Datta, S. P. Pal, M. M. Das and A. Pal, A General Graph Theoretic Framework for Multi-Layer Channel Routing, Proc. of 8th VSI/IEEE Intl. Conf. on VLSI Design, pp 202-207, 1995.

[8] Pal A., D. Kundu, A. K. Datta, T. N. Mandal and R. K. Pal, Algorithms for Reducing Crosstalk in Two-Layer Channel Routing, Journal of Physical Sciences, Vol. 10, pp. 167-177, 2006.

[9] Reed J., A. Sangiovanni-Vincentelli and H. Santomauro, A New Symbolic Channel Router: YACR2, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol 4, pp. 208-219, 1985.

[10] Schaper G. A., *Multi-Layer Channel Routing*, Ph.D. Thesis, Dept. of Computer Sc., Univ. of Central Florida, Orlando, 1989.

[11] Szymanski T. G., Dogleg Channel Routing is NP-Complete, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol 4, pp 31-41, 1985.

[12] Yoshimura T. and E. S. Kuh, Efficient Algorithms for Channel Routing, IEEE Trans. on CAD of Integrated Circuits and Systems, Vol 1, pp 25-35, 1982.