A Unified Approach to Multilayer Over-the-Cell Routing

Sreekrishna Madhwapathy, Naveed Sherwani
Dept. of Computer Science
Western Michigan University
Kalamazoo, MI 49008

Siddharth Bhingarde, Anand Panyam
Intel Corporation
Hillsboro, OR 97124

Abstract

Several Over-the-Cell (OTC) routing algorithms have been proposed for two and three layer processes. All the existing OTC routers can be used only on the cell models for which they were developed for. In this paper, we develop a unified approach to multi-layer routing, motivated by over-the-cell routing, which can be used for full-custom layouts. Our approach can also be directly applied to standard cell layouts, irrespective of the cell model used in the design. Our router has been implemented in C and tested on industrial benchmarks, such as PRIMARY I and PRIMARY II, for which it obtained channel-less layouts.

1 Introduction

Over the last four years, several researchers have investigated the use of the area over the cells, to reduce the height of the channels, which in turn, reduces the overall area of the layout [2, 3, 5]. This technique is referred to as "Over-the-Cell" routing.

Several over-the-cell (OTC) routers have been presented for standard cell layouts designed with two layers [2, 7, 15], in which OTC routing is done in M2. Although OTC routing minimized the layout heights, channelless layouts were not possible, even for low density designs. With the introduction of three metal process, two layers were made available for OTC routing, and thus allowing more nets to be routed in OTC areas. When the fabrication process does not permit the usage of vias in OTC areas, the OTC routing in M2 and M3 layers is planar. When vias are allowed in OTC areas, the router gets an additional flexibility of switching layers in OTC areas, which leads to significant reductions in the layout heights. Several three layer OTC routers have been developed [1, 2, 8].

In [2], it is assumed that the terminals are on the boundary (BTM or Boundary Terminal Model), and due to the placement of feed-throughs and power and ground lines, three cell layout styles, namely HDVC, HCVC and HCVD are used for OTC routing. In [8], the terminals are assumed to be in the center (CTM or Center Terminal Model), while in [1], it is assumed that the terminals are located at a specific offset from the boundary (MTM or Middle Terminal Model), the primary objective being the minimization of the channel heights or the elimination of the channels, thus minimizing the layout height. In [9], a new cell model called Target Based Cell model (TBC) was presented, which allows flexibility in the terminal locations. In this cell model, long vertical columns, called *targets* are provided in the M1 layer, instead of terminals located at fixed positions. Cells designed using this methodology, have smaller widths, compared to other cell models.

The advent of multi-layer processes led to the availability of more OTC area for routing, and hence, it is possible to acheive further reduction in the layout height. In [11], a four layer OTC routing algorithm was presented, for a cell model similar to TBC, which obtained channelless layouts. In this paper, we present a novel approach for multi-layer processes, which is a generalization of the approach presented in [11]. Unlike the existing approaches [13, 14], which use a net-by-net routing technique or area routing, our approach is based on channel routing. However, the channels are located on top of the active areas. This enables the use of well developed channel routing algorithms. Our approach is fast, performance oriented, and can also be applied to standard cell layouts, irrespective of the cell model used in the design.

We have implemented the proposed router in C on a SUN SPARC station 1+ and have tested this router on several benchmarks including PRIMARY I and PRIMARY II from MCNC. The router generated channelless layouts for these benchmarks.

The rest of this paper is organized as follows. In section 2, we describe the basic methodology of our approach. We present our multi-layer routing algorithm in Section 3. In Section 4, we present the summary of our experimental results and conclude with Section 5.

2 Basic Methodology

In this section, we describe the basic methodology of our approach to multilayer routing in full-custom layouts. A full custom layout may consist of several arbitrarily shaped rectilinear blocks. We assume that the terminals are located in M1 and poly layers. This imposes a restriction on the block design, due to which, M2 cannot be used for routing within the blocks. Therefore, it may not be possible to route all the *intra-block nets*. However, in our approach, we route the unrouted intra-block nets, over the blocks, along with the *inter-block nets*. By doing so, we also give the block/cell designer, the flexibility of leaving the terminals of the unrouted intra-block nets, at arbitrary positions in the block.

Figure 1 shows a full custom layout, with inter-block nets and intra-block nets, which could not be routed in the blocks using M1 and poly layers. All these nets are routed over the blocks. The *over-the-block (OTB) routing* reduces the area of the layout, since it decreases the area of the channels between the blocks, where the inter-block nets would have been routed otherwise. Infact, our router does not use the channels between the blocks in a layout, for routing.

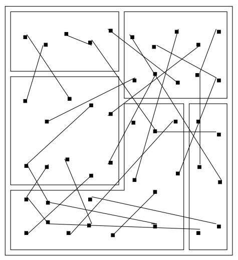

Figure 1: A Full Custom Layout with net connections for Over-the-Block Routing

3 Multi-layer Routing Algorithm

In this section, we present our algorithm, for multilayer routing, in full custom layouts. Our algorithm has the following steps. Some of the steps in this algorithm, are similar to those in the algorithm presented in [11], and we do not go into the details of these steps.

1. **Pseudo-row generation :** In this phase we partition the entire layout into *Pseudo-rows*, as shown in Figure 2(b), such that, any vertical column in a pseudo-row can have at most one terminal. In order to accomplish this, we use the M2 layer to position the terminals such that, we have at most one terminal, in each column of a pseudo-row. However, the terminal can be located anywhere in the column. After generating the pseudo-rows, the nets are classified into *same-row nets*, which have terminals in the same pseudo-row,and *multi-row nets*, which have terminals in more than one pseudo-row. Some of the vacant terminal positions in pseudo-rows are assigned to multi-row nets, so that, each multi-row net spans contiguous pseudo-rows. Hereafter, we will refer to a pseudo-row, simply as a *row*.

2. **Net Decomposition and Connection Assignment :** In this phase, each multi-row net is decomposed into *same-row nets* and *adjacent row nets*. Same-row nets are equivalent to those defined in the previous step. Each adjacent-row net conencts two same-row nets of a net, in two adjacent rows. The selection of one terminal each, from two adjacent rows, for establishing the connectivity between two same-row nets, belonging to the same net, is called *Connection Assignment*. The problem of selecting the terminals for connection assignment as described above, in all the rows, for all the nets, such that, the sum of the densities in all the channels is minimum, is called the Multi-row net Connection Assignment Problem (MCAP). MCAP was proved to be computationally hard in [10]. We use the heuristic algorithm presented in [11], to solve this problem. In this approach, each net $N_i, 1 \leq i \leq n$ (where n is the number of nets in the layout), is assigned a weight $w(N_i)$, based on the criticality of the net. Connection assignment for the nets is made, in the non-increasing order of weights.

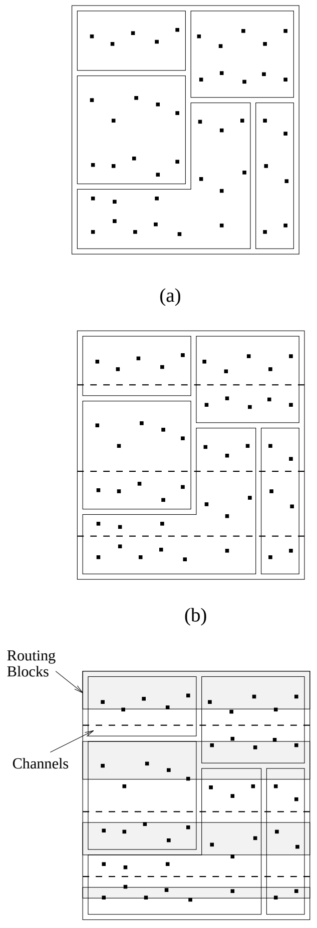

Figure 2: (a) A Full-Custom Layout (b) Pseudo-Row partitioning (c) Routing Block assignment

3. **Interval Generation :** In this step, intervals are generated for horizontal segments of nets. This procedure is described in [11]. Some of the adjacent row nets can be routed in two different ways, depending on the existence of vacant terminals. For these nets, horizontal segments are generated for both the routing choices, and only one of them can be used in the actual routing.

4. **Interval Selection :** The previous step generates two types of intervals; intervals corresponding to nets which have only one routing choice, and intervals corresponding to nets which have

two routing choices. Let $R_1, R_2, \dots R_K$ be the rows in a layout, where K be the number of rows in the layout. Let $V_i, 1 \le i \le K$, be the set of intervals in R_i, which correspond to the nets that have only one routing choice. Let $V_{i\ i+1}, 1 \le i < K$ be the set of intervals which correspond to the nets with two routing choices, that can be routed in either R_i or R_{i+1}. The main objective of this step is to select the maximum number of intervals that can be assigned to contiguous tracks in each row, such that the intervals in each set $V_{i\ i+1}, 1 \le i < K$, are assigned to only one of the cell rows R_i and R_{i+1}. We call this as the Interval Selection Problem (ISP). It is quite difficult to solve ISP optimally. At present, the complexity of IFP is unknown. We present a 0.5 approximation algorithm to solve this problem.

Let us consider two adjacent rows. Let V_i be the set of intervals of the nets that can be routed in row R_i and let V_{i+1} be the set intervals of the nets which can be routed in the row R_{i+1}. Let $V_{i\ i+1}$ be the set of nets which can be routed either in the R_i or R_{i+1}.

The approximation algorithm for interval selection is based on maximum k-independent set algorithms in interval graphs [6]. Although the algorithm is greedy in nature, we prove that it has a performance bound of 0.5. The details of our algorithm ALGO_IS are shown in Figure 3. In the algorithm, MIS refers to maximum k-independent set in interval graphs.

Algorithm *ALGO-IS*( )

Input: Set of Intervals,
$\mathcal{V} = \{V_1, V_{12}, V_2, \dots V_{K-1\ K}, V_K\}$
Output: Interval Assignment for all rows

```
Begin
  $S_1 = \phi$;
  $S_1 = MIS(V_1 \cup V_{12})$;
  For $i = 2$ to $K-1$
    $S_1 \cup MIS(V_i \cup V_{i\ i+1})$;
  $S_1 = S_1 \cup MIS(V_K)$;
  $S_2 = \phi$;
  $S_2 = MIS(V_1)$;
  For $i = 2$ to $K$
    $S_2 = S_2 \cup MIS(V_i \cup V_{i-1\ i})$;
  $S = MAX(S_1, S_2)$;
End;
```

Figure 3: Algorithm ALGO-IS

Theorem 1 *Let ρ be the approximation ratio of the above algorithm. Then $\rho \ge 0.50$.*

Proof: Let $W_i{}^*$ be the subset of V_i which is in the optimal solution. Similarly assume $W^*_{i,i+1}$ be a subset of $V_{i,i+1}$, which is in the optimal solution. Each $W^*_{i,i+1}$ can be partitioned into $U^*_{i,i+1}, D^*_{i,i+1}$, where $U^*_{i,i+1}$ is a subset of $W^*_{i,i+1}$ assigned row i, and $D_{i,i+1}$ is a subset of $W^*_{i,i+1}$ assigned to row $i+1$.

The algorithm is based on two strategies as shown in Figure 3. The first strategy guarantees that

$$| S_1 | \ge | W_1{}^* | + | W_2{}^* | + \dots | W_K{}^* | + | U_{12}{}^* | + | U_{23}{}^* | + \dots + | U_{K-1,K}{}^* |$$

Similarly the second strategy guarantees that

$$| S_2 | \ge | W_1{}^* | + | W_2 * \dots + | W_K{}^* | + | D_{12}{}^* | + | D_{23}{}^* | + \dots + | D_{K-1,K}{}^* |$$

Let α be the ratio of nets in the optimal solution which are from sets $V_{i,i+1}, 1 \le i \le K-1$. Obviously $1-\alpha$ is the ratio of the nets, which belong to sets $V_i, 1 \le i \le K$, which are in the optimal set.

It is clear that both strategies select $1-\alpha$ subset of nets. To see what fraction of α nets are chosen, notice that in the worst case, $\alpha = 1$, i.e., the optimal solution may consist of nets, which are only from $V_{12}, V_{23}, \dots V_{K-1\ K}$. By taking the maximum between S_1 and S_2, we guarantee that we will always select atleast 0.5α. Therefore the complete solution is

$$\begin{aligned} \rho &= 1 - \alpha + 0.5\alpha \\ &= 1 - 0.5\alpha \end{aligned}$$

This ensures that in the worst case, where $\alpha = 1$, $\rho = 0.5$. Therefore, the algorithm produces a solution which is atleast 50% of the optimal.□

5. **Track Assignment :** In order to simplify the routing, all the terminals in each row, are brought to a vacant track in that row, so that, all the terminal points in the net intervals are connected to their respective terminals on this vacant track, by a vertical strip in M2. Now, the problem is to determine the appropriate track in a row, to be left vacant, for these terminals, so that the total length of the vertical strips in M2 is minimized. This problem can be formally stated as follows.

INSTANCE: Given n terminals $t_1, t_2, \dots, t_n$ in k tracks, $T_1, T_2, \dots, T_k$, assume that any two consecutive tracks T_i and T_{i+1} have a virtual track T'_i in between. Let $d(t_j, T'_i)$ be the vertical distance between t_j in track T_m and T'_i, which is given by

$$d(t_j, T'_i) = | i - m | + 1$$

PROBLEM: Find a track T'_p such that

$$\sum_{q=1}^{n} d(t_q, T'_p)$$

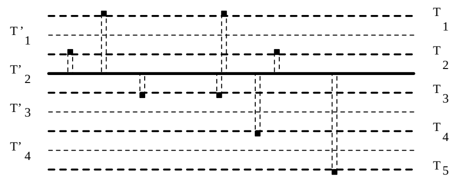

Figure 4: An Example of the Track Assignment probem

is minimized. We call this as the *Track Assignment Problem* This problem is similar to the *Single Trunk Steiner Tree Problem* [12], and can be solved in linear time. Therefore, we have the following result.

Theorem 2 *The Track Assignment Problem can be solved in $O(n)$ time.*

6. **Interval assignment:** The main objective of this step is to compute the total number of contiguous tracks required in each row, so as to to assign all the intervals (selected in the previous step). A contiguous set of tracks in a row used for interval assignment of same-row nets is called a *routing block*. The tracks in a routing block are not permutable. However, the actual position of the routing block is not fixed, and may be located anywhere in the row.

7. **Routing Block Assignment (RBA):** The routing blocks generated in the previous step are assigned to tracks in each row. First, the densities between routing blocks belonging to two adjacent rows are computed. Based on the densities, the routing blocks are assigned to tracks in each row (Figure 2(c)). An optimal algorithm which runs in $O(K)$ time to solve the Routing Block Assignment Problem (RBAP), was presented in [11].

8. **Channel Routing :** After the routing block assignment, the nets that are not completely routed in the routing blocks, are routed in the areas between the routing blocks, which we will refer to as *channels*. the terminals of the nets that are not completely routed in the routing blocks, are routed to the boundaries of the routing blocks, in M2 layer. A VHV router can be used to route these nets in the channels.

If during any of the above phases it is found that the given layout is unroutable, then we have to go back to the Placement phase, rearrange the blocks and the above procedure is repeated. This process is repeated until the placement makes the layout routable.

The above approach can also be directly applied to standard cell layouts. In this case, the pseudo-row generation step is not necessary, since standard cell layouts already have well defined cell rows. The rest of the steps are similar to the the steps described above.

4 Experimental results

We have implemented our router in C on a SUN SPARC station 1+ and tested it on several industrial benchmarks, including PRIMARY I and PRIMARY II. For all the benchmarks it has been tested on, the router generated channelless layouts.

As explained earlier, we do not have the pseudo-row generation step in standard cell layouts. After the initial phases, the routing algorithm generates the routing blocks. The routing blocks are then assigned to the tracks in each cell, based on the net densities. The height of the routing blocks and their assignment for each channel of PRIMARY I is shown in Table 1. Notice that, the multi-row nets are split into same row and adjacent row nets and the adjacent row nets are routed using vertical wire segments. Hereafter, we shall refer to the routing space between the routing blocks as a *channel* for simplicity, though it is over the cell area. The maximum height of the routing blocks is between the channels 3 through 7. In particular, channel 3 has 21 tracks. Also notice that, PRIMARY I is dense only in the top right corner of the layout and very sparse towards the left and bottom of the layout. The location of terminals for vertical tracks between two routing blocks is considered to be positioned at the topmost and bottom most tracks of a routing block. From these locations, M2 segments are used to connect to the actual terminals. The entire routing solution generated by the router for PRIMARY I is shown in Figure 5. Notice that the interconnections in each routing block are accomplished using horizontal tracks. Also notice that PRIMARY I benchmark is not dense enough to utilize all the OTC routing resources. A total of 657 net segments have been assigned to the routing blocks. Table 2 shows the number of nets assigned to each routing block. The remaining nets are routed in OTC areas between the routing blocks.

Our router takes 13.65 seconds to generate the channelless solution for PRIMARY I. This shows that our router is fast, as well as performance oriented.

5 Conclusion

In this paper, we have developed a unified routing approach, for multilayer routing, in full-custom designs. This approach can also be applied to standard cell layouts, irrespective of the cell model used in the design. The router we implemented, based on this approach, is fast and performance oriented. Using this router, we obtained channel-less layouts for industrial benchmarks like PRIMARY I and PRIMARY II.

References

[1] S. Bhingarde, A. Panyam, N. Sherwani, " Efficient Over-the-Cell Routing for General Middle Terminal Models" *IEEE Transactions on VLSI Systems*, December 1993, pp. 462-472.

[2] J. Cong, B. Preas, and C. L. Liu, "General Models and Algorithms for Over-the-Cell Routing in Standard Cell Design," *Proc. of DAC,* June 1990, pp. 709-715.

[3] N. Holmes, N. Sherwani, and M. Sarrafzadeh, "A Three Layer Over-the-Cell Channel Router," *Proc. of ICCAD*, Nov., 1991, pp. 428-431.

Table 1: Routing Block Assignments in PRIMARY I

Row No.	Routing Block		
	Height	TOP track	BOT track
1	12	1	12
2	14	1	14
3	16	1	16
4	21	1	21
5	15	4	18
6	15	4	18
7	15	3	17
8	11	6	16
9	9	2	10
10	10	1	10
11	8	1	8
12	8	1	8
13	7	1	7
14	11	1	11
15	7	1	7
16	12	1	12
17	7	1	7

Table 2: Routing Block Net Assignment in PRIMARY I

Row No.	Routing Block Height	Number of nets
1	12	35
2	14	40
3	16	43
4	21	44
5	15	49
6	15	38
7	15	41
8	11	42
9	9	43
10	10	37
11	8	36
12	8	37
13	7	36
14	11	35
15	7	38
16	12	37
17	7	26

[4] N. Holmes, N. Sherwani, and M. Sarrafzadeh, "Utilization of Vacant Terminals for Improved Over-the-Cell Channel Routing," *IEEE Transactions on CAD*, June, 1993, pp. 780-792.

[5] S. Das, S. C. Nandy,B. B. Bhattacharya, "An improved heuristic algorithm for over-the-cell channel routing", *Proc. of ISCAS*, vol. 5, 1991, pp. 3106,3109.

[6] M. Sarrafzadeh, R. D. Lou, "Maximum k-covering of Weighted Transitive Graphs with Applications", *Algorithmica*, Vol. 9, 1993.

[7] Y. Shiraishi, Y. Sakemi, "A Permeation Router", *IEEE Transactions on CAD*, May 1987, pp. 462-471.

[8] B. Wu, N. Sherwani, N. Holmes, M. Sarrafzadeh, "Over-the-Cell Routers for New Cell Model", *Proc. of DAC*, June 1992, pp. 604-607.

[9] S. Bhingarde, R. Khawaja, A. Panyam and N. Sherwani "Over-the-Cell Routing Algorithms for Industrial Cell Models", *Proc. of 7th International Conference on VLSI Design*, Jan. 1994, pp.143-148.

[10] S. Bhingarde, S. Madhwapathy, A. Panyam, N. Sherwani, " A Unified Approach to Multilayer Over-the-Cell Routing" *Tech. Report TR/93-17*, Department of Computer Science, Western Michigan University, November 1993.

[11] S. Bhingarde, S. Madhwapathy, A. Panyam, N. Sherwani, "An Efficient Four Layer Over-the-Cell Router", Proc. of ISCAS, 1994.

[12] T. Corman, C. Leiserson, R. Rivest, "Introduction to Algorithms" MIT Press, 1990.

[13] W. W. Dai, T. Dayan, "Topological routing in SURF: Generating a Rubber-band Sketch", *Proc. of DAC*, 1991, pp. 39-44.

[14] K. Khoo and J. Cong, "A Fast Multilayer General Area Router for MCM Designs", *Proc. of EURO-DAC*, Sept. 1992, pp. 292-297.

[15] S. Danda, S. Madhwapathy, A. Panyam, N. Sherwani, "An Optimal Algorithm for Maximum Two Planar Subset Problem", Proc. of GLSVLSI March 1994.

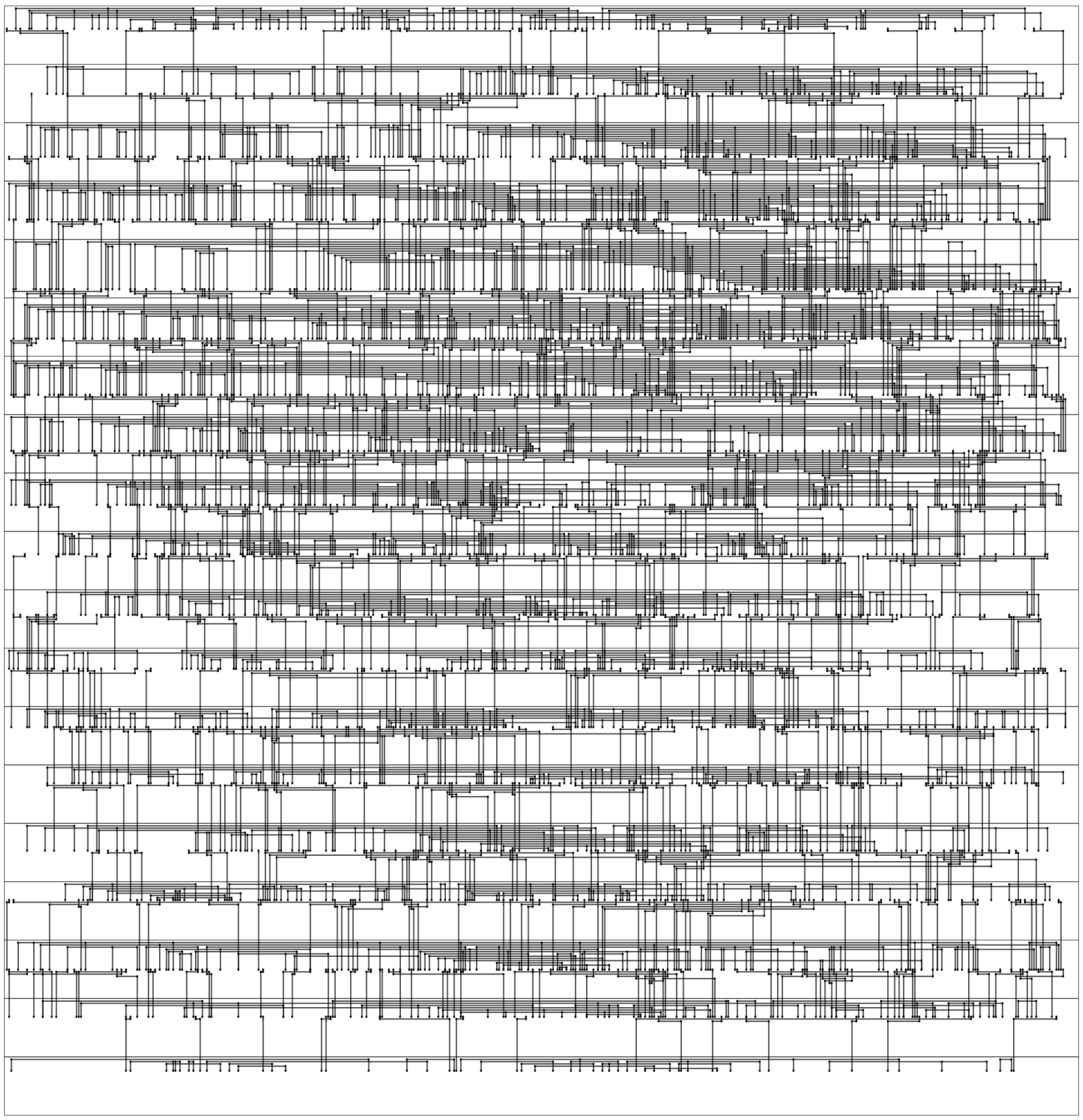

Figure 5. Routing of PRIMARY I