A Unified Approach to Multilayer Routing

Sreekrishna Madhwapathy

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses

Part of the Computer Sciences Commons

Recommended Citation
https://scholarworks.wmich.edu/masters_theses/5043
A UNIFIED APPROACH TO MULTILAYER ROUTING

by

Sreekrishna Madhwapathy

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science
Department of Computer Science

Western Michigan University
Kalamazoo, Michigan
April 1995
To
My Parents
Vidyullatha and Jayaram Rao Madhwapathy
And
My Aunt and Uncle
Neela and Naganand Sripathi
ACKNOWLEDGEMENTS

This thesis, and the degree it represents, could never have been completed without the help and support of my thesis advisor, Professor Naveed Sherwani, who has provided the starting point and constant motivation for this research. Naveed has also taught me, the art of technical writing and speaking on technical matters. I wish to express my gratitude and heartfelt appreciation to him.

Many other people have contributed in the completion of this thesis. I especially thank Professor Alfred Boals and Professor Ajay Gupta, for their willingness to serve on my thesis committee. I owe a debt of gratitude to Professor Donald Nelson, the Chairperson of our Department, for his support in providing the research facilities. I wish to extend special thanks to my academic advisor, Professor Donna Kaminski, for her constant encouragement and advice.

I wish to thank Anand, Siddharth and Danda for their valuable comments and suggestions. I would like to extend special thanks to Pramod and Aditya, for generating a standard cell library, in conventional and minimum width designs, and the other members of our ‘NiteGroup’, for their help and co-operation.

Finally, I thank my parents Vidyullatha and Jayaram Rao Madhwapathy, and my aunt and uncle, Neela and Naganand Sripathi, to whom I dedicate this thesis, for their encouragement, guidance and support which has always helped me in accomplishing my goals.

Sreekrishna Madhwapathy
A UNIFIED APPROACH TO MULTILAYER ROUTING

Sreekrishna Madhwapathy M.S.
Western Michigan University, 1995

Several Over-the-Cell (OTC) routing algorithms have been proposed for two and three layer processes. All the existing OTC routers assume that the terminals are laid out in a specific predetermined fashion. These restrictions on the terminals complicate the task of cell design and increase the width of the cells.

In this thesis, we report two results. First, we present a new layout methodology, for obtaining minimum width cells, with arbitrarily located terminals. Next, we develop a multilayer routing approach, which is independent of the cell model used in the layout design. Freed from fixed terminal placement restrictions, cell designers can aim to design with minimum width. We have also extended our approach to multilayer routing in Full Custom designs. Our router has been implemented and tested on industrial benchmarks such as PRIMARY I and PRIMARY II, for which it obtained channel-less layouts.
# TABLE OF CONTENTS

ACKNOWLEDGEMENTS ........................................ iii
LIST OF TABLES ........................................ iv
LIST OF FIGURES ........................................ v

CHAPTER

I. INTRODUCTION ........................................... 1
II. MINIMUM-WIDTH CELL LAYOUTS ......................... 6
III. THE MULTILAYER ROUTING ALGORITHM ................. 11
   Multi-row Net Connection Assignment (MCA) ........... 13
   Net Classification and Interval Generation (IGC) .... 14
   Interval Selection ...................................... 17
   Track Assignment ...................................... 20
   Interval Assignment / Same-row Routing (IAS) ........ 21
   Routing Block Assignment (RBA) ....................... 22
IV. MULTILAYER ROUTING IN FULL CUSTOM LAYOUTS ....... 26
V. EXPERIMENTAL RESULTS ................................ 29
VI. CONCLUSIONS ........................................... 33

BIBLIOGRAPHY ............................................... 34
LIST OF TABLES

1. Cell Width Comparison ........................................... 9
2. Routing Block Assignments in PRIMARY I ............... 30
3. Routing Block Net Assignment in PRIMARY I .......... 31
LIST OF FIGURES

1. Layout Area Minimization. .................................................. 4
2. Layout of a 3-input NAND (a) Conventional (b) Minimum Width. 7
3. Layout of a DELAY Gate (a) Conventional (b) Minimum Cell-Width. 8
4. An Example for Multi-Row Net Connection Assignment. .......... 14
5. Classification of Nets. ....................................................... 15
6. Algorithm ALGO-IS. .......................................................... 18
7. An Example of the Track Assignment Problem. ...................... 20
8. Possible Cases That Occur During Interval Assignment. .......... 22
9. Routing Block Assignment. .................................................. 23
10. Algorithm for Routing Block Assignment. ............................ 24
11. A Full Custom Layout With Net Connections. ....................... 26
12. Over-The-Block Routing. .................................................. 27
13. Routing of PRIMARY I. ..................................................... 32
CHAPTER I
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, or if possible, eliminate the channels in a standard cell or mixed block and cell layouts [3, 5, 6, 9]. This technique is referred to as "Over-the-Cell" routing. The effectiveness of OTC routing in reducing the height of a layout, mainly depends on two key factors; the fabrication process and the cell layout style. The fabrication process specifies the number of metal layers available for routing, and whether or not vias may be used in OTC areas, while the cell layout style specifies the locations of the terminals and the power and ground lines. The intra-cell connections are accomplished in poly and M1 layers.

In two metal layer process, OTC routing is accomplished in the M2 layer, in a planar fashion. Traditionally, the terminals were conveniently located at the boundary, and a planar subset of nets could be routed in the OTC area, while the remaining nets were routed in the channel. Several routers have been presented for two layer processes [11, 3]. Though OTC routing minimized the layout heights, channel-less layouts were not possible, even for low density designs. With the introduction of three metal process, two layers were made available for OTC routing, 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 the layers in OTC areas, which
leads to a significant reduction in the layout heights. Several three layer OTC routers have been developed [7, 3, 1].

The effectiveness of OTC routing in two and three layer processes [7, 3, 11, 1] depends on the locations of the terminals and the power and ground lines, which are specified by the cell model. Therefore, the cell model plays a significant role in minimizing the layout heights for two and three layer processes. Several cell models and corresponding routers were developed for the minimization of the layout heights [3, 6, 1]. All the existing OTC routers assume that certain restrictions have been placed on terminal locations. Usually it is assumed that all the terminals are aligned and located in one or two rows on the boundary, in the center or at a specific offset from the boundary.

In [3], 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 [1], the terminals are assumed to be in the center (CTM or Center Terminal Model), while in [7], 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.

With the advent of multi-layer processes more OTC area is now available for routing and hence, further reduction in the layout height can be accomplished. The total height in a standard cell layout is given by,

\[ H = K \cdot h_{cell} + \sum_{i=1}^{K-1} h_i \]

where \( K \) is the total number of cell rows, \( h_{cell} \) is the height of the cells and \( h_i \) is the
channel height in the \(i^{th}\) channel. The minimum height of the layout is achieved when all the channels are eliminated. The total layout area for a channel-less layout is given by \(A_T = W.K.h_{cell}\), where \(W\) is the width of the layout. The layout area can be further minimized only by minimizing the layout width \(W\). This can be accomplished by designing cells with minimum cell widths. For a given functionality, the cell width basically depends on the terminal alignment and intra-cell routing.

In [8], 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.

In this thesis, we report two results. First, present a new cell layout methodology, similar to the TBC model, for obtaining minimum width cells. Our experimental results show that, when compared to the conventional cell designs, cells designed using our methodology are on an average, 9.43\% smaller in width. Next, we present a new multilayer routing approach which is fast and performance oriented. Our algorithm assumes that the terminals are arbitrarily located in M1 or poly layers, such that each column in a cell row consists of at most one terminal. The routing in OTC areas is primarily accomplished in M2, M3 and M4 layers. In order to make a contact with a terminal, a via is dropped from M3 to M2 in the required column and a vertical segment in M2 is used for connecting the via to the terminal. As shown in Figure 1(a), though a channel-less layout is obtained for the existing cell designs, the width remains the same. However, in our approach, we adopt a two dimensional area minimization technique, wherein minimum layout areas are obtained by minimizing both the layout height as well
as the layout width, as shown in Figure 1(b). We have also extended this approach to multilayer routing in full custom designs. Unlike the existing approaches [2, 4], 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.

Figure 1. Layout Area Minimization.

We have implemented the proposed routers in C on a SUN SPARCstation 1+ and have tested these routers on several benchmarks including PRIMARY I and PRIMARY II from MCNC. The router generated channel-less layouts for these benchmarks. The overall reduction in layout area is 10% compared to the best available three layer routers [7].
The rest of this thesis is organized as follows. In Chapter II we present the minimum cell width layout methodology. In Chapter III, we present an overview of our algorithm and describe our algorithm in detail. In Chapter IV, we extend our approach to full custom designs. In Chapter V, we present our experimental results, and conclude with Chapter VI.
CHAPTER II
MINIMUM-WIDTH CELL LAYOUTS

The total area of a standard cell layout is dependent on the layout height and the layout width. While the effective utilization of OTC area leads to smaller layout heights, the layout width is determined by the cell widths. Since current process technologies lead to channelless layouts, the further die size minimization can only be achieved by minimizing the layout widths. The layout width depends on the cell widths. Therefore, it is important to minimize cell widths. In this chapter, we focus on the development of minimum-width cell layouts.

Conventional cell layouts require the terminals to be located at the cell boundaries. This was required to ease the task of channel routing. Since all the interconnections within a cell are accomplished in poly and M1 layers, the terminals have to be "brought" to the boundaries either in poly or in M1 layers. However, for circuits with dense intra-cell routing, a pre-defined position for the terminals and also "bringing" the terminals to the boundaries leads to increase in cell widths. Therefore, imposing a rigidity on the locations of terminals leads to increase in cell width.

In two and three layer processes, the total available OTC area is "limited". Therefore, in order to efficiently utilize the "limited" OTC area for routing, the terminals were required to be aligned. But, with the advent of multi-layer process technology, the OTC areas are not "limited" anymore and channelless designs can be achieved, irrespective of where the terminals are located. Therefore, terminal alignment is not necessary and the cell designer is given the flexibility to locate
the terminals at a most convenient place with a single criterion of cell-width minimization. The intra-cell routing is accomplished in poly and M1 layers while the power and ground lines are located in M1 layer. In order to compute the reduction in cell widths, several leaf cells were laid out in both the conventional and the new minimum cell-width style.

A 3-input NAND gate and a DELAY cell gate, laid out using MAGIC Ver 6.3 are shown in Figures 2 and 3 respectively. As shown in Figure 2(a), the gate poly is routed from the top cell boundary to the bottom cell boundary, forming p-type and n-type transistors. These poly routes can be directly extended beyond the power and ground lines to form the input terminals. However, the cell outputs
Figure 3. Layout of a DELAY Gate (a) Conventional (b) Minimum Cell-Width.

are usually available in M1 layer and have to be routed to the boundary in poly layer, outside the diffusion areas, which may tend to increase the cell width in conventional cells. On the other hand, minimum-width designs have flexibility in the location of the terminals. As shown in Figure 3, the DELAY cell has two terminals T1 and T2. In the conventional layout, since the terminals are located at the boundary in poly, the routing to the boundary is accomplished outside of the diffusion areas.

For analysis, a cell library comprising of over twenty leaf cells has been generated in conventional and minimum-width designs. Table 1 gives a comparison between the conventional cell layouts and the cells designed with minimum cell widths. The variation in cell widths between the cell layout styles can be attributed to intra-cell routing and terminal alignment. In the conventional design style, the intra-cell routing over the diffusion areas is accomplished in M1 layer, and the routing between the diffusions is accomplished in both poly and
M1 layers. For circuits with dense intra-cell routing, a pre-defined location for the terminals may lead to an increase in the cell width. The minimum-cell width designs have an advantage over the conventional designs, as there is no fixed location for terminals. Hence, the cell widths in minimum-width designs are always smaller than the corresponding widths in the other cell models. The conventional cell style required the terminals to be brought to the boundary while the terminals

Table 1

<table>
<thead>
<tr>
<th>Cell Description</th>
<th>Cell Width in λ</th>
<th>% Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two input NOR gate</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Three input NOR gate</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Four input NOR gate</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Two input NAND gate</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Three input NAND gate</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Inverter</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Non-Inverter</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Transmission gate</td>
<td>72</td>
<td>0.00</td>
</tr>
<tr>
<td>Nor-Latch</td>
<td>60</td>
<td>0.00</td>
</tr>
<tr>
<td>Pull-up</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Pull-Down</td>
<td>36</td>
<td>0.00</td>
</tr>
<tr>
<td>Hi-Impedance Buffer</td>
<td>72</td>
<td>0.00</td>
</tr>
<tr>
<td>Delay gate</td>
<td>84</td>
<td>0.00</td>
</tr>
<tr>
<td>D Flip-flop</td>
<td>156</td>
<td>0.00</td>
</tr>
<tr>
<td>Two input NAND and AND gate</td>
<td>49</td>
<td>0.00</td>
</tr>
<tr>
<td>Three input NAND and AND gate</td>
<td>60</td>
<td>0.00</td>
</tr>
<tr>
<td>Four input NAND and AND gate</td>
<td>60</td>
<td>0.00</td>
</tr>
<tr>
<td>Four input OR gate</td>
<td>60</td>
<td>0.00</td>
</tr>
<tr>
<td>Two input OR and NOR gate</td>
<td>48</td>
<td>0.00</td>
</tr>
<tr>
<td>Three input OR and NOR gate</td>
<td>60</td>
<td>0.00</td>
</tr>
<tr>
<td>Two bit Full Adder</td>
<td>168</td>
<td>0.00</td>
</tr>
<tr>
<td>Exclusive-OR gate</td>
<td>74</td>
<td>0.00</td>
</tr>
<tr>
<td>Exclusive-NOR gate</td>
<td>72</td>
<td>0.00</td>
</tr>
</tbody>
</table>
in the minimum cell-width designs can be conveniently located. This flexibility in the terminal locations led to 10.71% decrease in cell width for the DELAY gate and 14.58% reduction in cell width for the 3-input NAND gate. Our experimental results on the twenty-three randomly selected leaf cells laid out in both the layout styles, show that this flexibility leads to significant decrease in cell widths, when compared with the corresponding cells in the conventional cell layouts. As seen from the table, only three cells have similar widths, while the rest were wider in the conventional layout style. The minimization in cell widths range upto 25% when compared with the boundary terminal model, with pull-up cell being exceptional, with a reduction of 50% in cell width. On an average, it is seen that the cell widths in the new “minimum width” layouts are 9.43% smaller than the corresponding cells in the conventional layout style.
CHAPTER III

THE MULTILAYER ROUTING ALGORITHM

In this chapter, we present a unified routing approach to multilayer OTC routing. We assume that global routing is completed and feedthrough assignment has been done such that, multi-terminal nets (nets with more than two terminals) span contiguous cell rows. The routing algorithm has the following steps:

1. **Multi-row Net Connection Assignment (MCA):** In this phase, the multi-row nets are decomposed into same-row nets and adjacent-row nets. Same-row nets are used to connect all the terminals belonging to the same net in a cell row. Adjacent-row nets are used to connect two same-row nets belonging to the same net, in two adjacent-cell rows. The problem of choosing two terminals, from each same-row nets, of a particular net, in two adjacent-cell rows, is called the Multi-row Net Connection Assignment Problem (MCAP), which is formally defined in Section III. We conjecture that MCAP is NP-Complete and propose an efficient heuristic algorithm called ALGO-MCAP to solve it.

2. **Interval Generation for Critical and Same Row Nets (IGC):** In this step, all the decomposed segments of the nets, are classified into five categories, which intuitively indicates the difficulty involved in routing this net. After the net categorization, intervals are generated between the leftmost and rightmost terminals of a net in each cell row for all the critical and same row nets.

3. **Interval Selection (IS):** In this step we select the intervals to be routed in contiguous tracks in each cell row. However, some of the nets can be
routed in two different ways, and these nets have to be routed in one way or the other. The main objective of this step is to select a set of intervals from each cell row, such that, on the whole, the number of intervals selected, from all the cell rows, is maximum. The complexity of the interval selection problem (ISP) is unknown, and we develop an approximate algorithm ALGO-IS to solve this problem.

4. 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. We call this as the Track Assignment Problem, and in Section III, we present an optimal algorithm, to solve this problem.

5. Interval assignment / Same-row routing (IAS): The main objective of this step is to compute the total number of contiguous tracks required in each cell row to assign all the intervals (selected in the previous step). A contiguous set of tracks in a cell 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 cell row.

6. Routing Block Assignment (RBA): The routing blocks generated in the previous step are assigned to tracks in each cell row. First, the densities between routing blocks belonging to two adjacent cell rows are computed. Based on the densities, the routing blocks are assigned to tracks in each cell row. We present an optimal algorithm ALGO-RBA which runs in $O(K)$ time to solve the
Routing Block Assignment Problem (RBAP).

7. Composite VHV/HVHV OTC/Channel Router: The previous step specifies the routing areas between two adjacent routing blocks. A composite OTC/Channel router with VHV routing in OTC areas and HVHV routing in the channels is used to complete the routing between the routing blocks.

Multi-row Net Connection Assignment (MCA)

In this phase, the nets spanning adjacent cell rows are decomposed as same-row nets and adjacent-row nets. A multi-terminal net is decomposed into three segments. The first segment is used to connect all the terminals belonging to the same net in the upper cell row. The second segment is used to connect all the terminals belonging to the same net in the lower cell row. The third segment is used to establish a connection between the first two segments.

The selection of one terminal each from the upper and lower cell rows for establishing the connectivity between the two horizontal segments that belong to the same net, but located in adjacent rows is called Multi-row net Connection Assignment Problem (MCAP). An instance of MCAP is shown in Figure 4. Each time two terminals (of a net) from two adjacent cell rows are selected, they form a net in that channel. The objective of MCAP is to select terminals in all the rows for all the nets, such that sum of densities of all the channels is minimum. We conjecture that MCAP is computationally hard, and we propose the following heuristic for this problem. Each net is assigned a weight \( w(N_i) \), \( 1 \leq i \leq n \), which refers to the criticality of a net. If a net is highly critical, it is assigned a very high weight. The nets are considered in a non-increasing order of weights, For each net, sequentially, two adjacent rows are considered and the connections are selected as follows. If the horizontal segments of a net in the adjacent rows overlap, then
a single vertical segment in the overlap region is used to connect the horizontal
segments in two rows, if the net has terminals in the corresponding column in both
the cell rows. Otherwise, we select one terminal from each row such that they are
the closest pair of terminals in the overlap region. If the horizontal segments of
the nets do not overlap, then the nearest end points of the segments are connected.

![Diagram of net connection assignment](image)

Figure 4. An Example for Multi-Row Net Connection Assignment.

Net Classification and Interval Generation (IGC)

In this step, we initially classify each net as one of the five categories
explained below, which intuitively indicates the difficulty involved in routing this
net. After the net categorization, intervals are generated between the leftmost
and rightmost terminals of a net in each cell row for all the critical and same row
nets. Let $t_1$ and $t_2$ be the terminals of a net. The nets are considered in the
non-decreasing order of their weights. The two terminal net connections can be
classified as shown in Figure 5. In the following discussion $ROW(t)$($resp. COL(t)$)
refers to row (resp. column) which the terminal $t$ belongs. Type 0 nets are those nets in which $ROW(t_1) = ROW(t_2)$, or $COL(t_1) = COL(t_2)$. All the other types of nets do not satisfy the above two conditions. A net $(t_1, t_2)$ is called a Type 1 net if either the terminal position $(ROW(t_2), COL(t_1))$, or $(ROW(t_1), COL(t_2))$, or both, are vacant. To distinguish between the three types of Type 1 nets, these nets are further classified as Type 1(a), Type 1(b) and Type 1(c) nets depending on the condition they satisfy. They are routed in Metal 2 and Metal 3 layers.
Type 1(c) nets can be routed in two different ways as shown in Figure 5.

If the terminal positions \( (ROW(t_2), COL(t_1)) \) and \( (ROW(t_1), COL(t_2)) \) are not vacant, then two vacant terminal positions \( t_3 \) and \( t_4 \) in \( ROW(t_1) \) and \( ROW(t_2) \) respectively, are used to route the net as a Type 2(a) net as shown in Figure 5, where the vertical strip is in Metal 2. Even if \( t_3 \) and \( t_4 \) have already been used for a Type 2(a) net connecting two terminals in \( ROW(t_1) \) and \( ROW(t_2) \), net under consideration can still be routed using these vacant terminal positions, so that the vertical strip is in Metal 4. This type of net is called a Type 2(b) net.

If a net cannot be routed as a Type 1 or Type 2 nets, then an attempt is made to route it as a Type 3 net, as shown in Figure 5. For a Type 3 net the terminal positions \( (ROW(t_2), COL(t_1)) \) and \( (ROW(t_1), COL(t_2)) \) are not vacant. It can use any one of these terminal positions, if they were not already used for a similar net between \( ROW(t_1) \) and \( ROW(t_2) \). Like Type 1 nets, Type 3 nets can also be classified into three types. However Figure 5 shows only a Type 3(a) net. If Type 3 is also not possible then it can be routed as a Type 4 net, which is similar to Type 2, except that the terminal positions \( (ROW(t_1), COL(t_3)) \) and \( (ROW(t_2), COL(t_4)) \) are not vacant, and are not already used for a similar type of net between \( ROW(t_1) \) and \( ROW(t_2) \). If a net cannot be routed as any of the above types, then it has to be routed in the channel between \( ROW(t_1) \) and \( ROW(t_2) \).

After net classification, intervals are generated for the nets, in each cell row. These intervals correspond to the horizontal wire segments (Metal 3) in each row. Each interval has a via at both the end points. Depending on the type of via, the end points are classified into the following types. An end point corresponding to a via \( t \) between a horizontal segment in Metal 3 and a vertical segment in Metal 2, where the Metal 2 segment spans from \( ROW(t) \) to \( ROW(t) - 1 \), is called a
Type $a_1$ end point. An end point corresponding to a similar type of via, but with the Metal 2 strip that spans from $ROW(t)$ to $ROW(t) + 1$ is called a Type $b_1$ end point. In a similar way the end points corresponding to vias between Metal 3 and Metal 4 and Metal 2 to Metal 4 (Stacked via) are classified as Type $a_2$ and Type $b_2$ end points. An end point corresponding to a via between Metal 2 and Metal 3, with no vertical segment going to another cell row is called a Type $c$ end point. From Figure 5 it is clear that a vacant column in a cell row can have only one via each of types $a_1, b_1, a_2$ and $b_2$, and no Type $c$ vias. A column in a cell row having a terminal, can have only one via each of types $a_2$ and $b_2$, and any number of Type $c$ vias limited by the density at that column.

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 $R_1, R_2, \ldots, R_K$ be the cell rows in a layout, where $K$ be the number of cell rows in the layout. Let $V_i, 1 \leq i \leq K$, be the set of intervals in $R_i$, which correspond to the nets that have only one routing choice. Let $V_{i+1}, 1 \leq i < K$ 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 cell row, such that the intervals in each set $V_{i+1}, 1 \leq i < K$, 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 shown in Figure 6, to solve this problem.
\begin{algorithm}
\textbf{Algorithm }ALGO-IS( )
\begin{itemize}
\item \textbf{Input:} Set of Intervals, \\
\quad \mathcal{V} = \{V_1, V_{12}, V_2, \ldots V_{K-1}, V_K\}
\item \textbf{Output:} Interval Assignment for all cell rows
\end{itemize}
\begin{enumerate}
\item \textbf{Begin}
\item \quad \textbf{S}_1 = \phi;
\item \quad \textbf{S}_1 = MIS(V_1 \cup V_{12});
\item \quad \text{For } i = 2 \text{ to } K - 1
\item \quad \quad \textbf{S}_1 \cup MIS(V_i \cup V_{i+1});
\item \quad \textbf{S}_1 = S_1 \cup MIS(V_K);
\item \quad \textbf{S}_2 = \phi;
\item \quad \textbf{S}_2 = MIS(V_i);
\item \quad \text{For } i = 2 \text{ to } K
\item \quad \quad \textbf{S}_2 = S_2 \cup MIS(V_i \cup V_{i-1});
\item \quad S = MAX(S_1, S_2);
\item \textbf{End;}
\end{enumerate}
\end{algorithm}

Figure 6. Algorithm ALGO-IS.

Let us consider two adjacent cell rows. Let $V_i$ be the set of intervals of the nets that can be routed in cell row $R_i$ and let $V_{i+1}$ be the set intervals of the nets which can be routed in the cell row $R_{i+1}$. Let $V_{i+1}$ 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 [10]. 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 6. In the algorithm, MIS refers to maximum $k$-independent set in interval graphs.
Theorem 1 Let $\rho$ be the approximation ratio of the above algorithm. Then $\rho \geq 0.50$.

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 cell row $i$, and $D_{i,i+1}^*$ is a subset of $W_{i,i+1}^*$ assigned to cell row $i + 1$.

The algorithm is based on two strategies as shown in Figure 6. The first strategy guarantees that

$$| S_1 | \geq | W_1^* | + | W_2^* | + \ldots | W_K^* |$$

$$+ | U_{12}^* | + | U_{23}^* | + \ldots + | U_{K-1,K}^* |$$

Similarly the second strategy guarantees that

$$| S_2 | \geq | W_1^* | + | W_2^* | + \ldots + | W_K^* | +$$

$$| D_{12}^* | + | D_{23}^* | + \ldots + | D_{K-1,K}^* |$$

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

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

This ensures that in the worst case, where \( \alpha = 1 \), \( \rho = 0.5 \). Therefore, the algorithm produces a solution which is at least 50\% of the optimal. \( \Box \)

Track Assignment

In order to simplify the routing, all the terminals in each cell row, are brought to a vacant track in that cell row as shown in Figure 7, 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 cell 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.

\[
\text{INSTANCE: Given } n \text{ terminals } t_1, t_2, \ldots, t_n \text{ in } k \text{ tracks, } T_1, T_2, \ldots, T_k, \text{ assume that any two consecutive tracks } T_i \text{ and } T_{i+1} \text{ have a virtual track } T'_i \text{ in between.}
\]
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

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

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.

**Interval Assignment / Same-row Routing (IAS)**

In this phase, the intervals generated in the previous phase for each row, are sorted on the left end point and assigned using minimum number of tracks $k'$, such that $k' \leq k$, where $k$ is the number of tracks in a cell row. Let the tracks in a cell row be denoted by $T_1, T_2, \ldots, T_k$. The intervals should be assigned in such a way that at any column in a row, the end points of Type $a_1$ and Type $b_1$ are in tracks $T_i$ and $T_j$ such that, $T_i < T_j$. The same rule applies to Type $a_2$ and Type $b_2$ end points in a column. Type $c$ end points can be assigned to any track.

Figure 8 shows all the possible cases that can occur when assigning intervals with end points of Type $a_1$ and $b_1$. Cases I(a), II(a), III(a) and IV(a) are legal assignments and the rest are not legal. If the assignment is not legal, then, let $T_i$ and $T_j$ be the tracks assigned to intervals $I_1$ and $I_2$ with end points of types
\(a_1\) and \(b_1\) respectively, at a particular column, such that \(T_i < T_j\). First, we try to reassign \(I_1\) to a track \(T_i\) such that \(T_i > T_j\). If this is not the case then, an attempt is made to assign \(I_2\) to a track \(T_m\), such that \(T_m < T_i\). If this is also not possible, then the corresponding net \((ROW(i_1), COL(j_1), ROW(i_2), COL(j_2))\) is added to the list of nets, to be routed in the channel between \(ROW(i_1)\) and \(ROW(i_2)\). We define a routing block \(B_i\), of a cell row \(R_i\), as the set of tracks to which the intervals are assigned in a cell row.

![Diagram of possible cases](image)

**Figure 8. Possible Cases That Occur During Interval Assignment.**

**Routing Block Assignment (RBA)**

The routing blocks generated in the previous step are assigned to tracks in each cell row. First, the densities between routing blocks belonging to two adjacent cell rows are computed. Based on the densities, the routing blocks are assigned to tracks in each cell row. We present an optimal algorithm ALGO-
RBA which runs in $O(K)$ time to solve the Routing Block Assignment Problem (RBAP).

After interval assignment, routing-blocks are generated for each cell row. Each routing-block $B_i$ defines a fixed set of intervals to be assigned to a contiguous set of tracks over a cell row $R_i$. In this step, we develop an optimal algorithm to locate the routing-blocks in each cell row such that minimum layout height is obtained. In Figure 9 the routing blocks are shown with dotted lines. The height of the routing block $B_i$ is based on the intervals generated in IAS.

![Figure 9. Routing Block Assignment.](image)

First, we present the formal statement of the Routing-Block Assignment
Problem (RBAP). A greedy algorithm called ALGO-RBAP, shown in Figure 10, was proposed to solve RBAP. Let us start with some terminology.

**Algorithm ALGO-RBAP.**

**Input:** $R, D, B$
**Output:** Routing-Block Assignment on each Row.

```
begin Algorithm
    TOP($B_1$) = 1;
    for $i = 1$ to $(K - 1)$ do
        if $d_i > (2k - TOP(B_i) - |B_i| + 1 - |B_{i+1}|)$
            then
                $h_i = \frac{d_i - (2k - TOP(B_i) - |B_i| + 1 - |B_{i+1}|)}{2}$
                TOP($B_{i+1}$) = $k - |B_{i+1}| + 1$;
            else if $d_i > k - TOP(B_i) + 1$
                $h_i = 0$;
                TOP($B_{i+1}$) = $d_i + TOP(B_i) + |B_i| - k$
            else
                $h_i = 0$;
                TOP($B_{i+1}$) = 1;
        end
end Algorithm ALGO-RBAP
```

Figure 10. Algorithm for Routing Block Assignment.

Let $R = \{R_1, R_2, \ldots, R_K\}$ be the set of cell rows. Let $B = \{B_1, B_2, \ldots, B_K\}$ be the set of routing-blocks, such that $B_i$ lies on cell row $R_i$ and $|B_i|$ gives the number of tracks in $B_i$. Let $D = \{d_1, d_2, \ldots, d_{K-1}\}$ where $d_i$ is the density of nets between the routing blocks $B_i$ and $B_{i+1}$. Let $k$ be the total number of tracks in each cell row. The tracks in each cell row are numbered from top to bottom, in an ascending order, from 1 to $k$. For each routing block $B_i \in B$, let $TOP(B_i)$ ($BOT(B_i)$) represent the topmost (bottom-most) track assigned to $B_i$. Let $h_i$ be
the height of the $i^{th}$ channel. $h_i$ depends on the density not accommodated in OTC area between the blocks $B_i$ and $B_{i+1}$.

**INSTANCE:** Given $\mathcal{R}, \mathcal{B}, \mathcal{D}$ as described above.

**PROBLEM:** Assign a Routing-block in $R_i$ i.e $TOP(B_i)$ to a track in $R_i$ from $\{1, 2, \ldots (k- |B_i|)\}$, for $i = 1$ to $K$ such that,

$$H = \sum_{i=1}^{K-1} h_i$$

is minimized.

A top-down approach is considered for the routing-block assignment. The uppermost locations are considered in each cell row for placing the routing-block, depending on the net densities between the blocks. The greedy algorithm called ALGO-RBAP is formally stated in Figure 10. From Figure 10, it can be clearly seen that ALGO-RBAP optimally solves RBAP in $O(K)$ time. Hence, we have the following theorem.

**Theorem 3** Algorithm ALGO-RBAP generates an optimal solution for RBAP in $O(K)$ time, where $K$ is the number of cell rows in a layout.
CHAPTER IV

MULTILAYER ROUTING IN FULL CUSTOM LAYOUTS

In this chapter, 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 \textit{intra-block nets}. However, in our approach, we route the unrouted intra-block nets, over the blocks, along with the \textit{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 11 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.

![Figure 11. A Full Custom Layout With Net Connections.](image-url)
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. In fact, our router does not use the channels between the blocks in a layout, for routing. The channels in this case, are located on top of the active areas. The routing of such layouts using our approach involves the following steps: (a) Pseudo-row generation, (b) Global Routing, (c) Net Classification, (d) Interval Generation, (e) Interval Assignment, (f) Routing Block Assignment and (g) Inter-row routing.

![Figure 12. Over-The-Block Routing.](image)

In the Pseudo-row generation phase we partition the entire layout into *Pseudo-rows* 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. The terminal can be located anywhere in the column. After generating the pseudo-rows, the rest of the steps followed are similar to those in our standard cell routing approach. However, the M2 layer is not used any further, and the remaining routing is done in the the layers M3 and above.
Figure 12(a) shows an example of a general cell layout. The partitioning of the layout into pseudo-rows is shown in Figure 12(b). Figure 12(c) shows the layout after the routing block assignment phase. The inter-row routing is done in the space between two adjacent routing blocks called the channel.

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

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 2. 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 13. Notice that the interconnections in each routing block are accomplished using horizontal tracks. Also notice that PRIMARY I benchmark is not dense enough
Table 2

Routing Block Assignments in PRIMARY I

<table>
<thead>
<tr>
<th>Row No.</th>
<th>Routing Block</th>
<th>Height</th>
<th>TOP track</th>
<th>BOT track</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>12</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>14</td>
<td>1</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>16</td>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>21</td>
<td>1</td>
<td>21</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>15</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>15</td>
<td>4</td>
<td>18</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>15</td>
<td>3</td>
<td>17</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>11</td>
<td>6</td>
<td>16</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9</td>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>10</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>12</td>
<td></td>
<td>8</td>
<td>1</td>
<td>8</td>
</tr>
<tr>
<td>13</td>
<td></td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>11</td>
<td>1</td>
<td>11</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>12</td>
<td>1</td>
<td>12</td>
</tr>
<tr>
<td>17</td>
<td></td>
<td>7</td>
<td>1</td>
<td>7</td>
</tr>
</tbody>
</table>

to utilize all the OTC routing resources. A total of 657 net segments have been assigned to the routing blocks. Table 3 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.
Table 3

Routing Block Net Assignment in PRIMARY I

<table>
<thead>
<tr>
<th>Row No.</th>
<th>Routing Block Height</th>
<th>Number of nets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12</td>
<td>35</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
<td>40</td>
</tr>
<tr>
<td>3</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>4</td>
<td>21</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td>49</td>
</tr>
<tr>
<td>6</td>
<td>15</td>
<td>38</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>11</td>
<td>42</td>
</tr>
<tr>
<td>9</td>
<td>9</td>
<td>43</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>37</td>
</tr>
<tr>
<td>11</td>
<td>8</td>
<td>36</td>
</tr>
<tr>
<td>12</td>
<td>8</td>
<td>37</td>
</tr>
<tr>
<td>13</td>
<td>7</td>
<td>36</td>
</tr>
<tr>
<td>14</td>
<td>11</td>
<td>35</td>
</tr>
<tr>
<td>15</td>
<td>7</td>
<td>38</td>
</tr>
<tr>
<td>16</td>
<td>12</td>
<td>37</td>
</tr>
<tr>
<td>17</td>
<td>7</td>
<td>26</td>
</tr>
</tbody>
</table>
Figure 13. Routing of PRIMARY I.
CHAPTER VI

CONCLUSIONS

In this thesis, we have developed a multilayer OTC routing approach for standard cell layouts, which is independent of the cell model. For most designs, the proposed router generates channel-less layouts. Furthermore, the proposed router allows a new methodology for cell design, which leads to minimization of the layout width. We have also proposed a new layout methodology for obtaining minimum width cells. Experiments with our cell library has shown that, the cells with minimum width are 9.43% smaller than the conventional cells. We have also extended our approach to multilayer routing in Full Custom Layouts.
BIBLIOGRAPHY


