A Three-Layer Over-the-Cell Multi-Channel Routing Method for a New Cell Model

Masahiro Tsuchiya* Tetsushi Koide*
Faculty of Engineering, Hiroshima Univ.*
4-1, Kagamiyama 1 chome,
Higashi-Hiroshima, 739 JAPAN
Tel: +81-824-24-7676, Fax: +81-824-22-7195
e-mail: koide@ecs.hiroshima-u.ac.jp

Shin’ichi Wakabayashi* Noriyoshi Yoshida†
Faculty of Information Sciences, Hiroshima City Univ.†
151-5, Ozuka, Numata-Choo,
Asa-Minami-Ku, Hiroshima 731-31, JAPAN
Tel: +81-82-830-1760, Fax: +81-82-830-1792
e-mail yoshida@ce.hiroshima-cu.ac.jp

Abstract—In this paper, we present a new cell model for over-the-cell routing and a new over-the-cell multi-channel routing method. In the proposed new cell model, terminals can be placed arbitrarily on the second layer of a cell so that each cell doesn’t require the extra routing region on the first layer of a cell to align terminals. Unlike conventional cell models, some parts of the second layer are also utilized for the intra-cell routing in order to reduce the chip area. Therefore the size of the proposed cell model can be smaller than that of conventional cell model.

The proposed method consists of three phases. In order to utilize the proposed cell model, in phase 1, we simultaneously handle all channels to determine the most effective routing patterns from the set of possible routing patterns to minimize the chip area. In phase 2, for the effective routing patterns of nets selected in phase 1, over-the-cell routing nets are selected by a new greedy algorithm considering obstacles on over-the-cells. Finally, the conventional channel routing algorithm is applied for nets unrouted on over-the-cell. From the experimental results with MCNC benchmarks, the proposed cell model and algorithm produce smaller height of layouts as compared to those produced by conventional cell models and algorithms, and the effectiveness of the proposed method and cell model was shown.

I. INTRODUCTION

Over-the-cell channel routing is one of the most effective methods in standard cell layout design to minimize the chip area, and many heuristic algorithms have been developed [8, 10, 11, 12, 14]. The basic idea of over-the-cell (in short, OTC) routing is to use the region over the cells for routing some nets. Since the number of nets to be routed in the channel is reduced by routing some nets over the cells, the number of tracks in the channel is also reduced, resulting the reduction of the chip area.

Efficient utilization of over-the-cell routing regions mainly depends on a routing model [1]. A cell model specifies the locations of terminals and Vdd/Gnd busses, the number of metal layers, and permissibility of vias on OTC regions. Normally, either two metal layers (M1, M2) or three metal layers (M1, M2, M3) are available for routing. Vdd/Gnd busses are located either in M1 or M2, either inside or outside of a cell. Usually, the cell models can be classified by the terminal location. Based on the terminal location five classes of cell models have been proposed and associated OTC routers have been developed [3, 4, 7, 9, 16, 17, 19, 20, 21]. Those five classes of cell models are: the Boundary Terminal Models (BTM) [7, 9], the Center Terminal Models (CTM) [21], the Middle Terminal Models (MTM) [3], the Target Based Cells (TBC) [4], and the cell model that we have proposed in Ref. [16] (we call this model the Column-dependent Capacity Model (CCM)). Table I shows characteristics of cell models and examples of OTC routing of these cell models, respectively.

In most cell models except the TBC, terminals on the second layer of a cell row need to be assigned in line either at the boundary or the center of a cell row. Then, an extra routing region on the first layer is generally needed to align terminals. The extra routing region hinders the minimization of the chip area. Cell regions with multi-layers have been available for the OTC routing by the advance of semiconductor technology. It is desirable to increase the number of intra-cell routing layers without extending the cell area. This means that we need to use a part of the second layer as well as on the first layer for intra-cell routing. In the TBC, although terminals are placed on the second layer of a cell row arbitrarily, no part of the second layer is allowed to be used for intra-cell routing.

In this paper, we introduce a new cell model in which terminals are placed on the second layer of a cell row arbitrarily and a part of the second layer of a cell row is also used for the intra-cell routing. Since terminals are placed arbitrarily on the second layer of a cell row, no extra routing region is needed to align terminals on the first layer for the intra-cell routing. Since conventional routing algorithms are developed for the cell models in which terminals are aligned either at the bound-

---

Table I: Characteristics of cell models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Terminal</th>
<th>Vdd/Gnd</th>
<th>Feedthrough</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTM</td>
<td>boundary(M2)</td>
<td>center(M2)</td>
<td>M1</td>
</tr>
<tr>
<td>CTM</td>
<td>center(M2)</td>
<td>boundary(M1)</td>
<td>M2</td>
</tr>
<tr>
<td>MTM</td>
<td>middle(M2)</td>
<td>boundary(M1)</td>
<td>M2</td>
</tr>
<tr>
<td>TBC</td>
<td>middle(M2)</td>
<td>boundary(M1)</td>
<td>M2</td>
</tr>
<tr>
<td>CCM</td>
<td>boundary(M2)</td>
<td>center(M1)</td>
<td>M2</td>
</tr>
</tbody>
</table>

---

---

---
ary or the center of the second layer, those algorithms can’t be applied effectively for a new cell model.

In order to utilize the proposed cell model, we develop a new OTC routing algorithm for the new cell model. The proposed routing algorithm consists of three phases: (1) Multi-Channel Net Assignment, (2) Over-the-Cell Routing Considering Obstacles, and (3) Terminal Reassignment and Channel Routing. In phase 1, we simultaneously handle all channels to determine the most effective routing patterns from the set of possible routing patterns to minimize the chip area. We formulate the problem as the 0-1 integer programming problem and solve it. In phase 2, for the effective routing patterns selected in phase 1, OTC routing nets are greedily selected for each track of OTC. Since the new routing algorithm performs not only routing for an entire net but also routing for a part of a net, the OTC region is utilized effectively. In phase 3, a set of nets that aren’t selected for OTC routing nets in phase 2 is routed in a channel by using the conventional channel routing method [5].

The proposed algorithm has been implemented in C language on a SPARC server 1000 (135MIPS), and has been tested with several benchmarks. Experimental results show that the proposed algorithm can achieve results which are in a number of tracks of 55.1% better than an OTC H-V routing method with the BTM, 20.2% better than the conventional routing method in Ref. [21] with the CTM.

The rest of this paper is organized as follows. In Section II, we introduce a new cell model and formulate the OTC channel routing problem. In the Section III, we present a new OTC algorithm and show experimental results in Section IV. Finally, in Section V, we conclude this paper.

II OVER-THE-CELL CHANNEL ROUTING

A. A New Cell Model

In conventional cell models, since terminals are aligned either at the boundary or at the center of the cell row, an intra-cell routing region on M1 layer is needed as the extra routing region to align terminals.

In order to eliminate this extra routing region, we propose a new cell model. Figure 1 shows a new cell model and an associated OTC routing, respectively. In the proposed cell model, it is assumed that three metal layers are available for routing and that there is a grid for each layer. A column and a track represent a vertical and a horizontal line on a grid, respectively. The routing is performed on a grid. The connection of the same nets between adjacent layers can be realized by using vias that can be inserted at any grid point. As the routing model, we assume H-V-H routing in a channel and V-H routing in an OTC region. In the OTC region, it is assumed that vias are available. Vdd/Gnd busses are placed in the boundary of M1 layer of the OTC region. Feedthroughs are placed in M2 layer of the OTC region. In the conventional cell models, only the M1 layer is used for intra-cell routing. In our model, however, in order to reduce the chip area further, some parts of M2 layer are allowed to be used for intra-cell routing. A routing pattern for intra-cell routing on M2 layer is represented by a rectilinear polygon. Such a rectilinear polygon is regarded as obstacles for the OTC routing.

Terminals can be basically placed arbitrarily on M2 layer of an OTC. We assume that there exists at most one terminal for each column and a terminal can be routed vertically by using the column in which no obstacles exist from the terminal position to at least one side of boundaries of the cell row. Since the proposed cell model doesn’t place Vdd/Gnd busses at the middle of the cell row in M2 layer, a net can cross the OTC regions and can be routed adjacent cell rows. Therefore there are much degree of the OTC routability in the proposed cell model than conventional cell models. In other words, since a net has many OTC routing patterns, we can route a net effectively by utilizing OTC of the adjacent cell rows in the new cell model.

B. Problem Formulation

We define some terminologies to explain the proposed algorithm. A set of cells forms a cell row in a chip. A set of cell rows, denoted as \( R = \{ r_1, r_2, \ldots, r_k \} \), is numbered from the bottom to the top cell rows. Similarly, a channel is a region between adjacent cell rows, denoted by \( C = \{ c_0, c_1, \ldots, c_k \} \), which is numbered from bottom to top of a chip. Let the extended channel \( c_{c_j} \in EC \) be the routing region extended from channel \( c_j \) to obstacles and/or terminals on the OTCs of cell rows \( r_j \) and \( r_{j+1} \). Tracks on a cell row \( r_j \) are numbered from 1 to \( \# \) according to track starting from the bottommost track, and is denoted by \( T_j = \{ t_1, t_2, \ldots, t_{\#} \} \). Columns on a cell row \( r_j \) are numbered from 1 to \( \# \) according to column by column starting from the leftmost column. Then \( L_{\text{max}} \) is the maximum column number of cell rows, that is, \( L_{\text{max}} = \max \{ \# \} \).

It is assumed that a multi-terminal net is decomposed into a set of two terminal nets during global routing and a net already has the channel number to be routed with a conventional chan-
nel routing. The netlist is defined as \( N = \{ n_1, n_2, \ldots, n_{|N|}\} \). A set of nets to be routed in a channel \( c_j \) is defined as \( N_j = \{ n_1, n_2, \ldots, n_{|N|}\} \). The location of a terminal \( p_l \in P \) on a cell row \( r_j \) is denoted by triple \((x_l, t_l, j)\) where \( x_l, t_l, \) and \( j \) are the column, the track, and the row numbers, respectively. A net \( n_i \) can be denoted by a pair of terminals. The interval between the left and right terminal column positions of each net is defined as the segment \( s_{c_{g_j}} \in S_{c_{g}} \), and the segment list corresponding to the netlist is defined as \( S_{c_{g}} = \{ s_{g_1}, s_{g_2}, \ldots, s_{g_{|N|}}\} \). The horizontal length of the segment \( s_{c_{g_j}} \) is defined as, \( x \cap r_{c_{g_l}} h(s_{c_{g_j}}) = r(s_{c_{g_j}}) - l(s_{c_{g_j}}) \), where \( l(s_{c_{g_j}}) \) and \( r(s_{c_{g_j}}) \) are the column numbers of the left and right terminals of a segment \( s_{c_{g_j}} \), respectively.

Let the top (resp. bottom) OTC routing capacity at the column \( x \) in the cell row \( r_j \), denoted as \( cap^t_j(x) \) (resp. \( cap^b_j(x) \)), be the number of routable tracks between the top channel boundary \( c_j \) and a terminal/obstacle that exist at the column \( x \). If no terminals or obstacles exist at the column \( x \) in the cell row \( r_j \), \( cap^t_j(x) \) and \( cap^b_j(x) \) are the half height of the cell, that is, \( cap^t_j(x) = [H/2] \), \( cap^b_j(x) = [H/2] \). Let \( cross_{n\ell}(x) \) be the number of segments crossing the column \( x \). Then, the local extended channel density \( d_j(x) \) at the column \( x \) of a local extended channel \( c_j \) is defined as, \( d_j(x) = \max\{0, cross_{n\ell}(x) - (cap^t_j(x) + cap^b_j(x))\} \). The extended channel density \( D_j \) is defined as, \( D_j = \max_{1 \leq x \leq L_j} d_j(x) \). The total extended channel density \( AD \) in a chip is defined as, \( AD = \sum_{j=0}^k D_j \). Now, we formulate the OTC multi-channel routing problem as follows.

[The Over-the-Cell Multi-Channel Routing Problem]

Inputs: (1) Netlist \( N \), (2) Obstacles on OTCs, (3) A set of extended channels \( EC \)

Outputs: Routing patterns on the channels and the OTCs

Objective: Minimization of the total number of tracks used for channel routing

Constraints: Do not violate the layout constraints

III An Over-the-Cell Multi-Channel Routing Algorithm

A. Outline of the Algorithm

The proposed OTC multi-channel routing algorithm consists of three phases as shown below.

Phase 1: Multi-Channel Net Assignment: Since a terminal can be located arbitrarily on a cell, the OTC routing capacity at each column, which is derived from terminal positions, is not always uniform. In order to utilize all OTC regions of all cell rows effectively, we perform the multi-channel net assignment that selects the most effective routing pattern from possible routing patterns utilizing vacant columns, on which there exist no obstacles/terminals. Since the algorithm takes not only one routing pattern obtained from a result of global routing but also some routing pattern obtained by utilizing some channels (multi-channel) into consideration, whole OTC of all cell rows can be used effectively. We formulate the problem as the 0-1 integer programming problem and find the most effective routing patterns of nets with the minimum total extended channel density.

Phase 2: Over-the-Cell Routing Considering Obstacles: After phase 1, we obtained a net set \( N_j \) to be routed in an extended channel \( c_j \). The OTC routing algorithm is performed for each set of nets \( N_j \) independently. The algorithm constructs horizontal constraint graphs, and then performs selections of OTC routing nets greedily for each track \( t_k \) of an extended channel \( c_j \). Since the objective of OTC routing is the minimization of the extended channel density \( D_j \), it is preferable to select nets as OTC routing nets such that the extended channel density \( D_j \) is reduced as much as possible.

Phase 3: Terminal Reassignment and Channel Routing: Phase 3 is performed for nets that have not been routed as OTC routing nets in phase 2. Firstly, terminals of unrouted nets are routed to the boundary of a cell row with vertical segments. Then, since all terminals of unrouted nets are located at the boundary of a cell row, we can assume that this model is like the BTM. Next, we apply the conventional greedy three layer channel routing method[5]. After we have performed the proposed algorithm from phases 1 to 3, OTC channel routing has been finished. In the following subsections, we describe each phase in detail.

B. Multi-Channel Net Assignment (Phase 1)

1. The Multi-Channel Net Assignment Problem

We proposed a new cell model in the previous section. If either terminals or obstacles don’t exist at a column \( x \) of a cell row \( r_j \), a net \( n_i \) can be routed at the adjacent extended channel \( c_{j-1} \) or \( c_{j-1} \) by passing the column \( x \) of the cell row \( r_j \). In other word, a net can have another routing patterns in addition to the routing pattern to be routed in \( c_j \) which is determined by global routing. Although a net can be routed several extended channels far from \( c_{j+1} \), it may make the total wire length of the net longer. Therefore, if terminals of a net exist in the adjacent cell rows \( r_j \) and \( r_{j+1} \), it is assumed that the three routing patterns (within \( c_{j-1}, c_j, \) and \( c_{j+1} \)) are available for the net \( n_i \) in maximum (Figure 2 (a)). On the other hand, if terminals of a net exist in the same cell row \( r_j \), it is assumed that the net can have the four routing patterns (within \( c_{j-2}, c_{j-1}, c_j, \) and \( c_{j+1} \)) in maximum (Figure 2 (b)). We should select the most effective routing pattern to minimize the total extended channel density.

![Fig. 2. Restriction of routing patterns.](image-url)
density $AD = \sum_{j=1}^k D_j$. We define the selection of the most effective routing patterns in given possible routing patterns as the Multi-Channel Net Assignment Problem (MCNAP). In phase 1, the routing pattern of a net $n_i$ for the MCNAP is determined so as to minimize the total extended channel density $AD$.

The MCNAP has been shown NP-hard [9]. Therefore, in the proposed algorithm, the MCNAP is solved by applying the 0-1 integer linear programming (ILP). In order to formulate the MCNAP as the 0-1 ILP problem, we need to generate constraints concerning with an extended channel density $D_j$ for each column $x$ of a cell row $r_j$. However, in this case, the number of constraints will become large. Therefore, in order to figure out the critical area of an extended channel, we borrow an idea of the zone representation [22]. An advantage of the zone representation is that the number of constraints and variables of the 0-1 ILP can be reduced.

2. 0-1 Integer Programming Problem Formulation

In the MCNAP, we generate the following three constraints, (1) the pattern incompatible constraint, (2) the vacant column constraint, and (3) the zone density constraint.

(1) The Pattern Incompatible Constraint: First, we explain the pattern incompatible constraint. Figure 3 (a) shows an example of the pattern incompatible constraint. It is assumed that a net $n_i$ has $m_i$ (in our assumption, $m_i \leq 4$) routing patterns denoted by $PT_i = \{p_{i1}, p_{i2}, \ldots, p_{im_i}\}$. To route a net without having inconsistency, for each net $n_i$ only one routing pattern $p_{ik} \in PT_i$ should be selected. We call this constraint as the pattern incompatible constraint. It is assumed that a routing pattern $p_{ik}$ of a net $n_i$ corresponds to a variable $y_{ik} (y_{ik} \in \{0, 1\})$. If $y_{ik} = 1$, the net $n_i$ is routed by the routing pattern $p_{ik}$. Otherwise, i.e., $y_{ik} = 0$, the net $n_i$ isn’t routed by the routing pattern $p_{ik}$. Then the pattern incompatible constraint is denoted by,

$$\sum_{k=1}^{m_i} y_{ik} = 1, \quad \forall n_i \in \mathbb{N}. \quad (1)$$

(2) The Vacant Column Constraint: Next, we explain the vacant column constraint. We define a column $x^j_t$ of cell row $r_j$ with no obstacles/terminals as a vacant column $x^j_t$ of a cell row $r_j$. Now, we introduce a constraint for the vacant column $x^j_t$. As shown in Figure 3 (b), if more than one routing patterns of nets can be considered at the vacant column $x^j_t$, we must select one of these patterns to route without causing a wire-short. We call this constraint the vacant column constraint and denoted by,

$$\sum_{n_i \in \mathbb{N}, pt_{i,k} \text{ which passes } x^j_t} y_{ik} \leq 1, \quad \forall r_j \forall x^j_t. \quad (2)$$

This constraint allows only one net to pass through a vacant column $x^j_t$. Otherwise, more than one net are routed, making a wire-short.

(3) The Zone Density Constraint: Finally, we explain the zone density constraint (Figure 3 (c)). Considering the OTC routing capacities $cap^i_t(x)$ and $cap^j_t(x)$ of a cell row $r_j$ by a terminal at a column $x$ of a cell row $r_j$, we want to select the most effective OTC routing patterns so as to minimize the extended channel density $D_j$. To formulate the degree of the contribution of a routing pattern to minimize the extended channel density $D_j$, we consider the zone density constraint for each zone $z^j_l \in Z_j$ in each $cc_j$. Let $H$ be the number of tracks on an OTC region. Let $PT_j(z^j_l) \subset PT_j$ be the set of routing patterns of a net $n_i$ in a zone $z^j_l \in Z_j$ of a $cc_j$. Let $Cap^i_b(z^j_l)$ (resp. $Cap^j_b(z^j_l)$) be the maximum of $cap^i_t(x)$ (resp. $cap^j_t(x)$) in the zone $z^j_l$ of the $cc_j$. We have already discussed in Section II that the OTC routing capacity at a column $x$ of $cc_j$ is determined from $cap^i_t(x)$ and $cap^j_t(x)$. The OTC routing capacity of the zone $z^j_l$ is determined by $Cap^i_b(z^j_l)$ and $Cap^j_b(z^j_l)$. Therefore, the zone density constraint is denoted by,

$$\sum_{n_i \in \mathbb{N}, pt_{i,k} \text{ which passes } x^j_t} y_{ik} + (H - Cap^i_b(z^j_l)) + (H - Cap^j_b(z^j_l)) \leq D_j, \quad \forall c_j \in EC, \forall z^j_l \in Z_j. \quad (3)$$

The reason of subtraction $Cap_b(z_l^j)$ from $H$ is to prevent that the left hand side of the inequality becomes negative. Consequently, the MCNAP is formulated with the 0-1 ILP as shown below.

\[ \text{minimize : } AD = \sum_{j=1}^k D_j \]
\[ \text{subject to : } \sum_{k=1}^{m_i} y_{ik} = 1, \quad \forall n_i \in \mathbb{N} \]
\[ \sum_{n_i \in \mathbb{N}, pt_{i,k} \text{ which passes } x^j_t} y_{ik} \leq 1, \quad \forall r_j \forall x^j_t \]

[ The 0-1 ILP Formulation of the MCNAP ]
\[ \sum_{n_i \in N} \sum_{p_i \in Pr(t_1)} u_{k1} + (H - C a p_h(z_1^i)) + (H - C a p_h(z_2^i)) \leq D_j, \\]
\[ \forall c_{j1} \in EC, \forall z_1^i \in Z_j, \]

**C. Over-the-Cell Routing Considering Obstacles (Phase 2)**

In phase 2, we update a set \( N_j = \{n_1, n_2, \cdots, n_{t_j}\} \) of nets for the OTC routing from the net set of the extended channel \( c_{j1} \). We can solve the OTC routing problem, which generates over-the-cell routing patterns to minimize the extended channel density \( D_j \), by treating independently each extended channel \( c_{j1} \). Therefore, we only discuss the OTC routing of an extended channel \( c_{j1} \) in the following.

1. **The Horizontal Constraint Graphs (HCGs)**

Before describing the details of the phase 2, we define the **Horizontal Constraint Graphs (HCGs)** \( G_j^h(t_k) = (V_j^h(t_k), E_j^h(t_k)) \) used in the algorithm. Given a net set of a \( c_{j1} \), \( N_j = \{n_1, n_2, \cdots, n_{t_j}\} \), and the corresponding segment set \( Sc_{j1} = \{sc_{j1}(1), sc_{j1}(2), \cdots, sc_{j1}(s_{t_j})\} \), the horizontal constraint graphs are constructed at each track \( t_k \) of a top and bottom cell rows in an extended channel \( c_{j1} \) and is defined \( G_j^h(t_k) \) and \( G_j^h(t_{k+1}) \), respectively. The horizontal constraint graphs \( G_j^h(t_k)(s = t, b) \) for a track \( t_k \in T_j \) on an OTC region are directed weighted graphs. Vertices \( v_i \in V_j^h(t_k) \) in \( G_j^h(t_k) \) consist of a segment set \( Sc_{j1}(t_k) \subset Sc_{j1} \) where a segment in \( Sc_{j1}(t_k) \) can be H-V routable at the track \( t_k \) on an OTC region of the cell row \( r_{j+1} \) (resp. \( r_j \)). A directed edge \( e_{lm} = (v_i, v_m) \in E_j^h(t_k)(s = t, b) \) is inserted from a vertex \( v_i \) to a vertex \( v_m \) if the two vertices \( v_i, v_m \in V_j^h(t_k) \) corresponding to the segments \( sc_{j1}, sc_{j1} \) satisfy the condition that horizontal segments \( sc_{j1} \) and \( sc_{j1} \) don’t overlap each other and satisfy the condition \( r(sc_{j1}) < l(sc_{j1}) \). Figure 4 shows an example of \( G_j^h(t_k) \). The horizontal constraint graph \( G_j^h(t_k) \) in Figure 4 is constructed for a track \( t_k \) on the upper cell row \( r_{j+1} \). Since the set of vertices \( V_j^h(t_k) \) in \( G_j^h(t_k) \) consists of the nets that can be routed at the track \( t_k \) on the OTC region of the cell row \( r_{j+1} \), that is, the segments \( sc_{j1}, sc_{j2}, \cdots, sc_{j6} \) are selected in the example. For the directed edge, since the segments \( sc_{j1} \) and \( sc_{j2} \) overlap each other, the directed edge \( e_{12} = (v_1, v_2) \) is not inserted between \( v_1 \) and \( v_2 \). On the other hand, the segments \( sc_{j1} \) and \( sc_{j2} \) do not overlap each other and satisfy \( r(sc_{j1}) < l(sc_{j2}) \), then the directed edge \( e_{13} = (v_1, v_3) \) is inserted from \( v_1 \) to \( v_3 \).

Next, we define the weights of the vertex \( v_i \) for the horizontal constraint graph \( G_j^h(t_k)(s = t, b) \). The phase 2 consists of two stages, and the horizontal constraint graph \( G_j^h(t_k) \) is employed for each stage. We define the weight of a vertex \( v_i \) in stages 1 and 2 as \( w(v_i) \). The weight \( w(v_i) \) corresponding to the segment \( sc_{j1} \) is defined in the following expression.

\[
w(v_i) = \alpha \times \max_{l(sc_{j1}) < x < l(sc_{j2})} \frac{d_j(x)}{D_j} + x \log w(t_k) \log \delta(sc_{j1}),
\]

where \( \alpha \) is a constant. In our experiment, we set \( \alpha = 30 \). From the definition of this weight, a segment \( sc_{j1} \) with a bigger value \( w(v_i) \) is more effective to reduce the extended channel density \( D_j \) and easier to be selected as an OTC routing net.

2. **An Over-the-Cell Routing Algorithm**

The proposed algorithm consists of two stages. In stage 1, a segment \( sc_{j1} \) of a net \( n_i \in N_j \), which contributes to reduce the extended channel density \( D_j \), is routed on the OTC region. In order to utilize the OTC regions which are not used for routing of stage 1, in stage 2, a segment \( sc_{j1} \) of a net \( n_i \in N_j \) which could not be routed in stage 1 is divided into subsegments. And then the subsegments are routed in an unrouter interval on the OTC region. An unrouter interval is an interval that is not be used for routing at a track \( t_k \) on an OTC region (Figure 5). For example in Figure 5, the segment \( sc_{j1} \) and the segment \( sc_{j2} \) are routed at the track \( t_k \), but the center region of the track \( t_k \) isn’t used for routing. But the segment \( sc_{j3} \) can not be routed in the unrouter interval without dividing it. Dividing the segment \( sc_{j1} \) to two sub-segments, the right subsegment can be routable at the track \( t_k \). This dividing procedure can be applied to the segment satisfying the condition that a subsegment can be routed at the track \( t_k \) on the OTC region.

The proposed algorithm selects OTC routing nets from finding the path \( \text{path}(t_k) \) that has the maximum total weight on \( G_j^h(t_k)(s = t, b) \) for stages 1 and 2. The total weight of vertices
in a path $\text{path}(t_k)$ on $G_j^2(t_k)$ is denoted by,
\[
    w(\text{path}(t_k)) = \sum_{v_i \in \text{path}(t_k)} w(v_i).
\]

Finding the path $\text{path}(t_k)$ with the maximum total weights on $G_j^2(t_k)$ means to find a set of nets that do not overlap each other and that reduce an extended channel density $D_j$. The detailed description of the proposed algorithm of phase 2 is as follows.

[Phase 2: Over-the-Cell Routing Algorithm]

(Stage 1)

**Step 1**: Set the starting tracks, $bt_j$ and $tt_j$ at cell rows $r_j$ and $r_{j+1}$ to $bt_j = H - \max_{1 \leq j \leq L} \text{cap}_j(x_j)$, $tt_{j+1} = \max_{1 \leq j \leq L} \text{cap}_j(x_{j+1})$, respectively;

**Step 2**: Calculate the weight $w(v_i)$ of $G_j^2(tt_{j+1})$ and $G_j^3(bt_j)$ for a segment $seq_i \in Sseq_j$ of net set $N_j$, which can be routed at a track $bt_j$ and $tt_{j+1}$ on an OTC region, respectively;

**Step 3**: Construct graphs $G_j^2(tt_{j+1})$ and $G_j^3(bt_j)$, and find the paths $\text{path}(bt_j)$ and $\text{path}(tt_{j+1})$ that have the maximum total weights, respectively;

**Step 4**: If $w(\text{path}(bt_j)) > w(\text{path}(tt_{j+1}))$, then assign a set of segments on $\text{path}(bt_j)$ at the track $bt_j$;

Otherwise, assign a set of segments on $\text{path}(tt_{j+1})$ at the track $tt_{j+1}$;

(Stage 2)

**Step 5**: Perform the following Steps 5.1 - 5.4 to unrouter intervals at the track $st = \{bt_j \text{ or } tt_{j+1}\}$ on the selected cell row from the leftmost positioned unrouter interval to the rightmost positioned unrouter interval;

**Step 5.1**: Divide segments into subsegments to be routed on the current track $st$ of the OTC region and calculate the weight $w(v_i)$ for a subsegment $seq_i$ that can be routed within an unrouter interval;

**Step 5.2**: Construct the graphs $G_j^2(st) = (s, t)$ and find the path $\text{path}(st)$ with the maximum total weight;

**Step 5.3**: Route subsegments of nets in the path $\text{path}(st)$ on the OTC region within the unrouter interval;

**Step 5.4**: If this process has been performed for all unrouter intervals at the track $st$, then go to Step 6;

Otherwise, go to Step 5.1 to perform the next unrouter intervals;

**Step 6**: If $bt_j = H$, $tt_{j+1} = 0$, then terminate;

Otherwise, if a cell row $r_j$ was selected in Step 4, then $bt_j := bt_j - 1$ and go to Step 2, else $tt_{j+1} := tt_{j+1} - 1$ and go to Step 2.

Next, we analyze the time complexity of the OTC routing algorithm described above. Let $|N_j|$ be the number of nets routed within the channel and $H$ be the height of the cell row. In the OTC routing algorithm, the time complexity to construct the HCG and to find a path for each track is $O(|N_j|^2)$. This process is performed for the $H$ tracks of the cell row. Therefore, the time complexity is $O(H \cdot |N_j|^2)$.

Figure 6 shows an example for an execution of the OTC routing algorithm. In this example, the OTC routing pattern of a cell row $r_j$ is considered. Segments under the cell row in Figure 6 show the net set $N_j$. Figure 6 (a) shows an input of the OTC routing. It is assumed that the location with a column $x$ and a track $t$ is denoted by $(x, t)$ and that a horizontal interval between $(x, t)$ and $(x', t)$ is denoted by $[x, x']$. The algorithm is performed from the track 4 far from the channel. In Figure 6 (b), the segments $seq_2$ and $seq_3$ are selected as the candidates for the OTC routing net. In this example, the segment $seq_2$ is selected as an OTC routing net and routed by using on M3 layer of the OTC. Furthermore, since the right terminal of the segment $seq_2$ is within the unrouted interval $[8, 11]$. The segment $seq_3$ is divided into sub-segments $seq_{g31}, seq_{g32}$ on the track 4. Then the segment $seq_{g31}$ is routed from the right terminal to the point $(8, 4)$ where the segment $seq_{g31}$ doesn’t intersect the segment $seq_2$ within the unrouted interval $[8, 11]$. The right terminal position of the segment $seq_3$ is newly defined as $(8, 4)$, which is the location of the column at which the terminal exist. From this operation, the horizontal length of the segment $seq_3, x length(seq_3)$ is changed from $x length(seq_3) = 5$ into $x length(seq_3) = 2$. Similarly, in Figure 6 (c), the algorithm is performed for the next track 3. However, in this case, since no net is selected in stage 2, only stage 1 is performed. Note that the candidates to be routed at this track 3 for the OTC routing net are not only the the segments $seq_1$ and $seq_2$ but also the segment $seq_3$. Similarly, the algorithm is performed as shown in Figures 6 (d) and (e). When the algorithm has been performed for all tracks on the OTC region, the OTC routing algorithm has been finished.
TABLE III

<table>
<thead>
<tr>
<th>Data</th>
<th>(D)</th>
<th>(\sum T_{CR})</th>
<th>(\sum T_{PR})</th>
<th>(\sum I_{PB})</th>
<th>(% R_{TPB})</th>
<th>(\sum I_{HV})</th>
<th>(% R_{THV})</th>
<th>(\sum I_{TC})</th>
<th>(% R_{RTC})</th>
</tr>
</thead>
<tbody>
<tr>
<td>mult16</td>
<td>284</td>
<td>153</td>
<td>66</td>
<td>95</td>
<td>30.2</td>
<td>88</td>
<td>25.0</td>
<td>89</td>
<td>25.8</td>
</tr>
<tr>
<td>struct</td>
<td>188</td>
<td>99</td>
<td>40</td>
<td>52</td>
<td>25.1</td>
<td>58</td>
<td>31.0</td>
<td>53</td>
<td>22.5</td>
</tr>
<tr>
<td>primarySC1</td>
<td>419</td>
<td>221</td>
<td>4</td>
<td>127</td>
<td>41.7</td>
<td>107</td>
<td>15.7</td>
<td>120</td>
<td>38.5</td>
</tr>
</tbody>
</table>

IV. Experimental Results

We have implemented the proposed algorithm described in Section III on a SUN SPARCserver 1000 (135MIPS) in C programming language. Table II shows experimental data. We used the PRIMARY benchmark in Ref. [21], and the MCNC benchmarks of mult16, struct, and primarySC1. The placement and the global routing of the MCNC benchmarks were obtained from TimberWolfSC4.2c [18] and GLORIA [15], respectively. In benchmarks shown in Table II, terminals are placed at the boundary of a cell row and equi-potential terminals are on the vertical column, and hence the cell model is different from the proposed cell model. For the proposed algorithm, we transformed these data of the conventional cell model to the proposed cell model by randomly assigning track numbers of terminals and placing obstacles without changing column numbers of terminals(Figure 7). For the benchmark in Ref. [21], track numbers of terminals are randomly generated from the track \(t_1\) to the track \(t_9\) without changing original column number of terminals Capacities on the OTC region set to 6. Obstacles are placed at positions satisfying the condition that terminals can be routed by the routing patterns obtained from the global routing. For these experimental data, we have performed various experiments to make sure of the effectiveness of the proposed cell model and method. For phase 1 of the proposed algorithm, we have used the Ip_solve1.5 [2] as the 0-1 ILP package.

In order to show the effectiveness of the proposed cell model, we firstly have compared four conventional methods with the proposed method, in which the multi-channel net assignment of phase 1 does not apply. The four conventional methods are (1) channel routing method for the BTM (CR) [5], (2) planar routing method for the BTM (PR) [7], (3) H-V routing method for the BTM (HV) [19] and (4) Ref.16’s routing method for the CCM (CC) [16]. Table III shows the comparative results for conventional routing methods. In Table III, \(D\) denotes the original channel density with two layers. \(\sum T_{PR}\) denotes the total number of tracks in the proposed method. Similarly, \(\sum T_{CR}\), \(\sum T_{PR}\), \(\sum T_{HV}\), and \(\sum T_{CC}\) are defined as the total number of tracks for each routing method, respectively. \(\% R_{TPB}\), \(\% R_{THV}\), and \(\% R_{RTC}\) denote the improvement ratio of the proposed method to each routing method, respectively. From the results of Table III, the proposed method can route nets in a number of tracks of 41.7% in maximum better than the PR for the BTM, and we can show the effectiveness of the proposed cell model and routing method.

Next, in order to show the effectiveness of the proposed method, we compared the proposed method with the H-V routing method for the BTM. In this case, we apply the multi-channel net assignment of phase 1. Table IV shows the experimental results. In Table IV, \(\text{diff}\) denotes \(\text{diff} = T_{HV} - T_{PR}\). The conventional routing method in Table IV is to perform H-V routing on the OTC region with allowing vias for the BTM. Then, the OTC routing capacities on M2 and M3 layers are uniform, i.e., 6. From the results of Table IV, the proposed method can route nets in a total number of tracks of 55.1% better than the HV for the BTM.

Finally, we have compared the proposed method with the CTM routing method (CTMR) for the CTM in Ref. [21]. Table V shows the experimental results after the OTC routing. The experimental results of the CTM routing method are referred from Ref. [21]. We also compared with the planar and H-V routings. From the results of Table V, the number of tracks of the proposed method are locally bigger than the number of tracks of the CTM routing method. For example, in the channel...
number 1, the number of tracks of the proposed method are one more than the number of tracks of the CTM routing method. However, when we consider the total number of tracks in all channels, the total number of tracks of the proposed method are much smaller than the total number of tracks of the CTM routing method. Therefore, the proposed method utilizes OTCs of all cell rows more effectively than the CTM routing method. Table VI shows improvement ratio of the proposed method to the conventional methods. The proposed method can route nets in a number of tracks of 51.7% better than the CR for the BTM, 12.6% better than the PR for the BTM, 1.2% better than the HV for the BTM, and 20.2% better than the CCMR for the CTM, respectively. Therefore, we can conclude that the proposed method is effective in reducing the chip area.

V Conclusion

In this paper, we have introduced a new three-layer standard cell model in which terminals and obstacles are placed arbitrarily on M2 layer, and proposed an associated over-the-cell multi-channel routing algorithm considering terminal positions and vacant columns. In the proposed cell model, since terminals need not to be aligned and obstacles can be placed arbitrarily on M2 layer of a cell row, a routing region on M1 layer for an intra-cell connection is reduced. Therefore, the chip size with the proposed cell model has decreased. The proposed algorithm selects the most effective routing pattern from some routing patterns obtained by utilizing a vacant column of a cell row. Since the routing pattern is routed with not only an entire segment but also a part of a segment, the proposed algorithm can utilize an over-the-cell region effectively. We have performed experiments on benchmarks for this cell model. Experimental results show that the proposed cell model and algorithm are both effective in reducing the chip area.

ACKNOWLEDGEMENTS

The authors thank Mr. Y. Negoro who has helped in the implementation of programs.

REFERENCES


#### TABLE V

<table>
<thead>
<tr>
<th>channel</th>
<th>D</th>
<th>$t_{CR}$</th>
<th>$t_{PR}$</th>
<th>$t_{HV}$</th>
<th>$t_{CTM}$</th>
<th>$t_{PR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11</td>
<td>6</td>
<td>5</td>
<td>0</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td>9</td>
<td>6</td>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>21</td>
<td>11</td>
<td>7</td>
<td>7</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>24</td>
<td>15</td>
<td>9</td>
<td>9</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>5</td>
<td>20</td>
<td>11</td>
<td>9</td>
<td>8</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>32</td>
<td>17</td>
<td>12</td>
<td>8</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>7</td>
<td>22</td>
<td>14</td>
<td>7</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>21</td>
<td>16</td>
<td>8</td>
<td>7</td>
<td>9</td>
<td>7</td>
</tr>
<tr>
<td>9</td>
<td>21</td>
<td>13</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>10</td>
<td>14</td>
<td>8</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>11</td>
<td>16</td>
<td>12</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>8</td>
<td>5</td>
<td>4</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>13</td>
<td>13</td>
<td>8</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>13</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>6</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>16</td>
<td>11</td>
<td>6</td>
<td>3</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>17</td>
<td>13</td>
<td>7</td>
<td>4</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>18</td>
<td>6</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Total</td>
<td>289</td>
<td>172</td>
<td>95</td>
<td>84</td>
<td>104</td>
<td>82</td>
</tr>
</tbody>
</table>

#### TABLE VI

<table>
<thead>
<tr>
<th></th>
<th>$%t_{CR}$</th>
<th>$%t_{PR}$</th>
<th>$%t_{HV}$</th>
<th>$%t_{CTM}$</th>
<th>$%t_{PR}$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>51.7</td>
<td>12.6</td>
<td>1.2</td>
<td>20.2</td>
<td>20.2</td>
</tr>
</tbody>
</table>