Chapter 4

Exact Minimum-Width Transistor Placement for General CMOS Cells

4.1 Introduction

This chapter shows a minimum-width transistor placement method which is applicable to CMOS cells in presence of non-dual P and N type transistors, whereas the cell layout synthesis methods explained in the previous chapters are only for dual cells. This chapter only targets the minimum-width transistor placement, and does not take the intra-cell routings into consideration. The proposed method are the first exact method which can be applied to CMOS cells with any types of structure, whereas almost all of the conventional exact transistor placement method[19, 22, 24] are applicable only to dual CMOS cells. All of these methods make pairs of complementary P and N type transistors and align them in minimum width. Therefore, all these methods can not be applied to some cells that have non-dual P and N type transistors. Even for the dual circuits, these methods can not always generate the minimum-width layouts, since the width depends on pairing of P and N type transistors. Since non-dual CMOS cells occupy a major part of an industrial standard-cell library, the exact minimum-width transistor placement should be applied even to non-dual CMOS cells.

Zhang *et al.*[31] proposed the novel transistor pairing algorithm which is applicable to the cells including non-series-parallel networks. This algorithm decides pairs of P and N type transistors for general complex gates, in which there are more than two transistors with a common input signal in their gates, so that the resulting cell width is minimized. This method, however, is not an exact method and generates large placement when a given cell has a transistor which does not have a pair transistor with the common gate input signal.

We have proposed a cell layout synthesis method using Boolean Satisfiability (SAT) in Chapter 2. This method can generate an exact minimum-width transistor placement for non-

Table 4.1 The numbers of non-dual cells in commercial standard-cell libraries.

cell library	*total number of cells*	*the number of non-dual cells*	*ratio*
130nm A	527	274	52%
130nm B	578	294	51%
90nm A	462	90	19%
90nm B	340	176	52%

series-parallel networks if there is no P and N type transistors which can not be paired by the common gate input signal. However, this method has the pairing restriction same as [31] and is not applicable to the cells including the P and N type transistors which can not be paired by the common gate input signal. This chapter proposes a minimum-width transistor placement method for CMOS cells in presence of non-dual P and N type transistors using SAT.

There are three main contributions of this work. First, an exact transistor placement problem of non-dual CMOS cells is defined for the first time. Table 4.1 lists the numbers of cells including non-dual P and N type transistors in several industrial standard-cell libraries. In a standard-cell library, not only flip flops or three-state buffers but also intrinsically dual CMOS circuits possibly have non-dual P and N type transistors as a result of transistor folding to meet height requirement, as shown in Figure 4.1. As a practical matter, non-dual CMOS cells occupy about a half of an industrial standard-cell library. We propose a Pseudo-Boolean[30] formulation of an exact minimum-width transistor placement problem of non-dual CMOS cells. As explained in Chapter 3, Pseudo-Boolean formulation can handle Conjunctive Normal Form (CNF), Pseudo-Boolean Form (PBF), and optimization constraints. PBF constraints are expressed by linear inequalities of Boolean variables. The proposed method can solve all these non-dual cells that can not be solved by the conventional exact transistor placement methods only for dual cells. In addition, P and N type transistors with common gate input signal are aligned vertically as much as possible for ease of intra-cell routing which follows the transistor placement. Using this formulation, an exact minimum-width transistor placement is generated for the cells to which the conventional exact method only for dual cells is not applicable. Moreover, this method generates a smaller width placement even for a number of dual cells by introducing the pairs of P and N type transistors with different gate input signals.

Second, the hierarchical approach of the transistor placement is also presented for practical

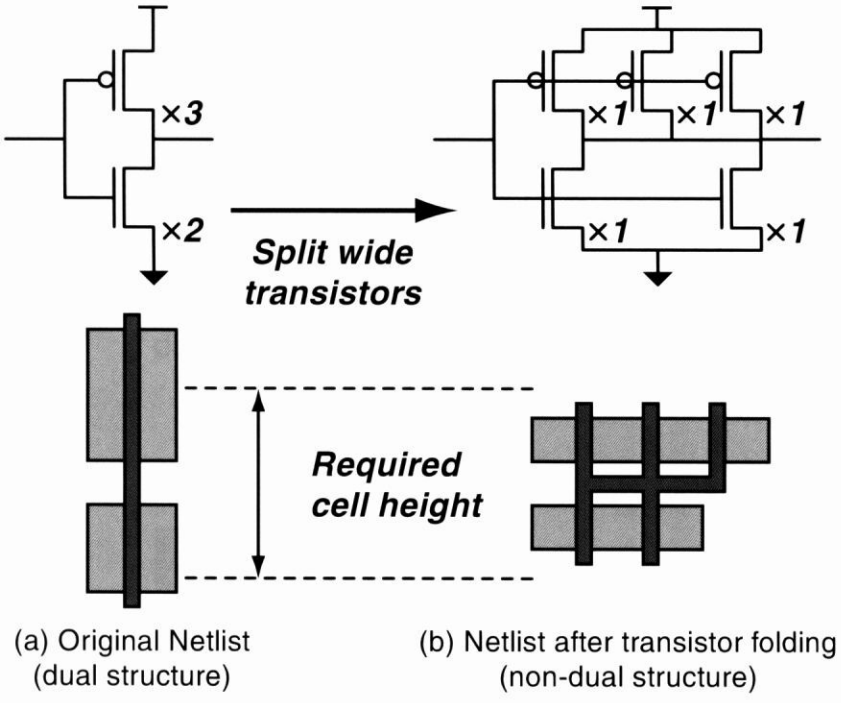

Figure 4.1 An example of a dual CMOS circuit which has a non-dual structure after transistor folding.

use. Although this approach has possibility to generate wider placement than that generated flatly, the experimental results show that it generates the transistor placements with little increase in cell width. It generates a minimum-width transistor placement hierarchically in about one second even for the cells with large number of transistors, whereas the runtimes of our flat approach is over one hour for larger cells with more than 24 transistors.

Third, we generalize the proposed single-row placement method to the multi-row placement and propose an exact minimum-width multi-row transistor placement method for dual and non-dual CMOS cells. The multi-row style layout is essential in several applications such as data paths in which the height of cells is variable while the width of cells is often predetermined. The multi-row style also has flexibility to place large P transistors and small N transistors together and vice versa. Moreover, the multi-row style has other advantages over the single-row style. It offers control over the cell shape and aspect ratio, and can also reduce the intra-cell routing and wire lengths. Therefore, the multi-row style cell layout attracts attention on recent deep sub-micron technologies and a lot of place&route and cell layout synthesis tools begin to support multi-height cells. The proposed method uses an efficient gate connection style and generates more area-efficient transistor placements than the conventional exact minimum-width multi-row style transistor placement method only for dual cells.

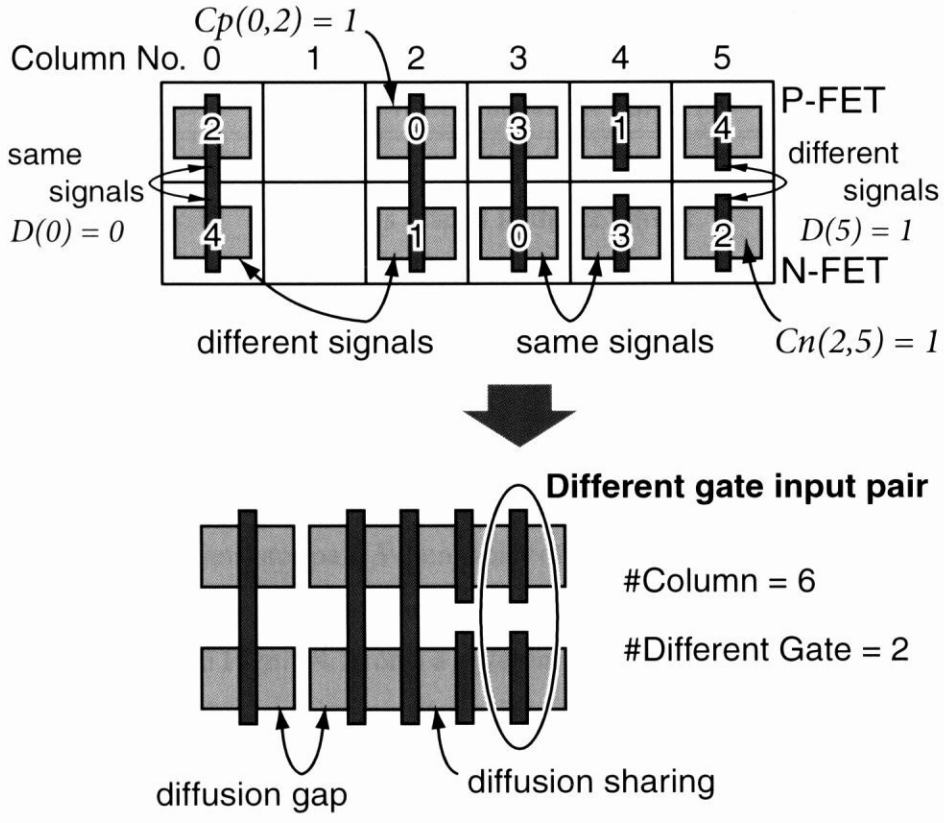

Figure 4.2 The problem definition of the single row transistor placement for non-dual CMOS cells.

Section 4.2 describes the problem definition of the single-row transistor placement method for non-dual CMOS cells. The Boolean formulation of our flat and hierarchical single-row transistor placements are explained in Section 4.3 and Section 4.4, respectively. Next, Section 4.5 generalizes the single-row placement method to multi-row placement. Then, Section 4.6 presents the experimental results and finally Section 4.7 summarizes this chapter.

4.2 Problem Definition

When X N type transistors and Y P type transistors are given, we have to place these $X + Y$ transistors in the minimum area. This problem can be transformed into the problem that places all transistors using the minimum number of columns as illustrated in Figure 4.2. The P type transistors are aligned in the upper row and the N type transistors are in the bottom. Two neighboring transistors face the diffusions that belong to the common signals each other to connect their source or drain by diffusion sharing. The empty columns result in the diffu-

Table 4.2 The variables used for the formulation of the single-row transistor placement for non-dual CMOS cells.

name	*number*	*condition which makes the value 1*
$C_n(i,k)$	$X \times W$	N-FET i is placed in the column k.
$C_p(i,k)$	$Y \times W$	P-FET i is placed in the column k.
$F_n(i)$	X	N-FET i is flipped.
$F_p(i)$	Y	P-FET i is flipped.
$D(k)$	W	Different gate input pair is placed in the column k.

sion gaps in the final layout as illustrated in Figure 4.2. The P and N type transistors placed in the same column form a transistor pair. When a pair of P and N type transistors have different gate input signals, this pair is called a *different gate input pair*. For example, P type transistor 4 and N type transistor 2 in Figure 4.2 form a *different gate input pair*. If a column has only a P or an N type transistor, this transistor is also called a *different gate input pair*. Under these layout styles, the transistor placement problem is transformed into the Conjunctive Normal Form (CNF) and Pseudo-Boolean Form (PBF) constraints. We will explain two approaches for generating a minimum-width single-row transistor placement in following sections. First, the flat approach which generates an exact minimum-width placement of any types of CMOS cells will be explained. Next, we will explain the hierarchical approach which reduces the runtime drastically with little increase in cell width.

4.3 Flat Single-Row Transistor Placement

In our formulation of the flat approach, $W + 1$ variables are introduced for each transistor to identify its location and whether it is flipped or not, where W is the number of the columns of the placement area. Additional W variables are needed to identify whether the *different gate input pair* is placed in each column or not. All variables needed for this formulation are listed in Table 4.2. Table 4.2 shows the names of the variable type, the numbers of each type of variables, and the conditions when each type of variables takes the value of 1. Some example cases of such conditions are also illustrated in Figure 4.2. We formulate the following Boolean constraints which express all the possible transistor placements in W columns under our layout style.

Transistor overlap constraint: N type transistors must not overlap in the same column. This constraint is expressed by the following PB constraints.

$$\sum_{i=0}^{X-1} C_n(i,k) \leq 1, \quad 0 \leq k < W \tag{4.1}$$

The same constraint is expressed as PB constraints for P type transistors.

$$\sum_{i=0}^{Y-1} C_p(i,k) \leq 1, \quad 0 \leq k < W \tag{4.2}$$

Transistor instantiation constraint: Each transistor must be instantiated once. The following PB constraints express this constraint.

$$\sum_{k=0}^{W-1} C_n(i,k) = 1, \quad 0 \leq i < X \tag{4.3}$$

$$\sum_{k=0}^{W-1} C_p(i,k) = 1, \quad 0 \leq i < Y \tag{4.4}$$

Neighboring transistors constraint: Two transistors facing the diffusions which belong to the different signals each other can not be placed in the neighboring columns. For example, N type transistors 4 and 1 in Figure 4.2 can not be placed in the neighboring columns. This constraint is expressed by the following logic equation.

$$GAP_n(i,j) \wedge \bigvee_{k=0}^{W-2} (C_n(i,k) \wedge C_n(j,k+1)) = 0 \tag{4.5}$$
$$i \neq j, \quad 0 \leq i < X, \quad 0 \leq j < X$$

Here, $GAP_n(i,j)$ is the logic function of $F_n(i)$ and $F_n(j)$, and takes the value of 1 if N type transistor i can not share its diffusion with N type transistor j placed to its immediate right, otherwise 0. This logic equation is expressed as CNF. The same constraint is also expressed as follows for P type transistors.

$$GAP_p(i,j) \wedge \bigvee_{k=0}^{W-2} (C_p(i,k) \wedge C_p(j,k+1)) = 0 \tag{4.6}$$
$$i \neq j, \quad 0 \leq i < Y, \quad 0 \leq j < Y$$

Objective function: The number of the *different gate input pairs* is minimized for ease of intra-cell routing which follows the transistor placement. The objective function is expressed as follows.

$$Minimize: \quad \sum_{k=0}^{W-1} D(k) \tag{4.7}$$

Definition of $D(k)$: When a *different gate input pair* is placed in the column k, the value of $D(k)$ must be 1. This definition is expressed by the following logic equation,

$$D(k) = \bigvee_{i,j}(C_n(i,k) \wedge C_p(j,k)), \quad 0 \le k < W \tag{4.8}$$

$$GATE_n(i) \neq GATE_p(j), \quad 0 \le i < X,\ 0 \le j < Y \tag{4.9}$$

where $GATE_n(i)$ and $GATE_p(j)$ mean the gate input signals of N type transistor i and P type transistor j, respectively. This logic equation is transformed into the PB constraints as follows for all combinations of i and j expressed by (4.9)

$$C_n(i,k) + C_p(j,k) \le D(k) + 1, \quad 0 \le k < W \tag{4.10}$$

under the condition that $D(k)$ is minimized. The minimization condition of each $D(k)$ value is already expressed by the objective function (4.7). Therefore, this definition is simply expressed by the PB constraints (4.10) for all combinations of i and j expressed by (4.9). When the column k has only a P or an N type transistor, $D(k)$ also takes a value of 1.

Finally, we can determine whether given transistors can be placed in W columns or not using these constraints. If no satisfiable assignment is found by the Boolean solver, it is guaranteed that there is no possible placement of W columns. Therefore, we can find an exact minimum-width transistor placement using the procedure described below.

1. For a given transistor netlist, count the number of N and P type transistors. The initial number of column W is set to max(#N-FET, #P-FET).

2. Search for a satisfiable assignment of the Boolean constraints constructed for W columns. If a satisfiable assignment is found, these transistors can be placed in W columns and this procedure terminates. Otherwise, go to step 3.

3. $W = W + 1$. Go to step 2 again.

4.4 Hierarchical Single-Row Transistor Placement

As explained in Section 4.1, the flat approach can not solve the cells with large number of transistors in reasonable time. Therefore, we also propose a hierarchical approach of our

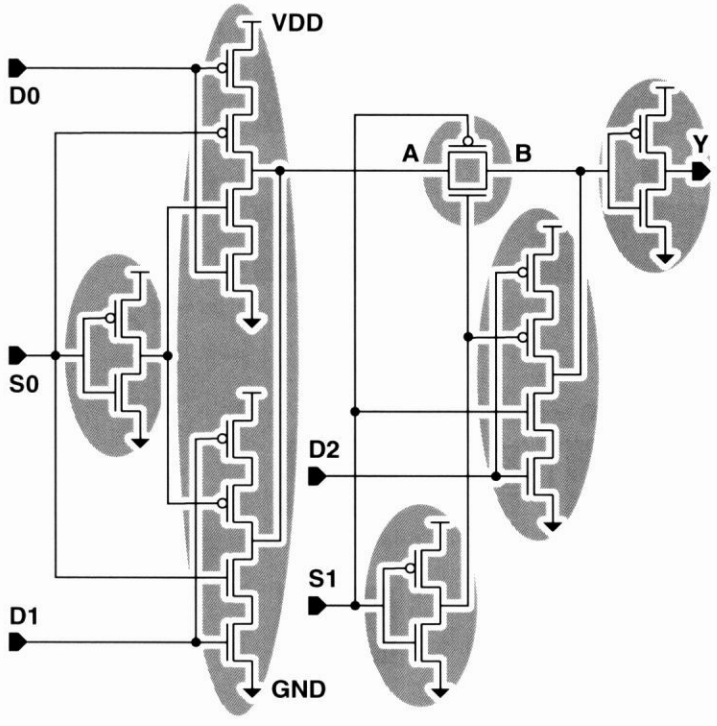

Figure 4.3 Schematic of 3-input multiplexer and its logic block partitioning.

transistor placement method based on circuit partitioning for practical use. The hierarchical transistor placement procedure used in this method is similar to that of Chapter 3 and also composed of partitioning, intra-block placement, and inter-block placement. However, the partitioning and intra-block placement procedures have some differences to handle non-dual CMOS cells effectively. The details of these procedures are explained in the following paragraphs.

Partitioning: Our approach first partitions the given cell into logic blocks. A logic block consists of transistors that are connected together by their diffusions except power and ground. We also recognize a transmission gate as a logic block in this partitioning procedure. Figure 4.3 shows an example of 3-input multiplexer circuit. A transmission gate is identified as an individual logic block. Two clocked inverters whose outputs are connected are identified as one logic block using our partitioning procedure.

Intra-block placement: Next, intra-block placement of each logic block is performed using the procedure explained in Section 4.3. In this stage, an exact minimum-width transistor placement of each block is generated. When Pseudo-Boolean constraints are generated, one new optimization cost is introduced to maximize the number of the connections by diffusion sharing between logic blocks in the inter-block placement step. The diffusions which can be

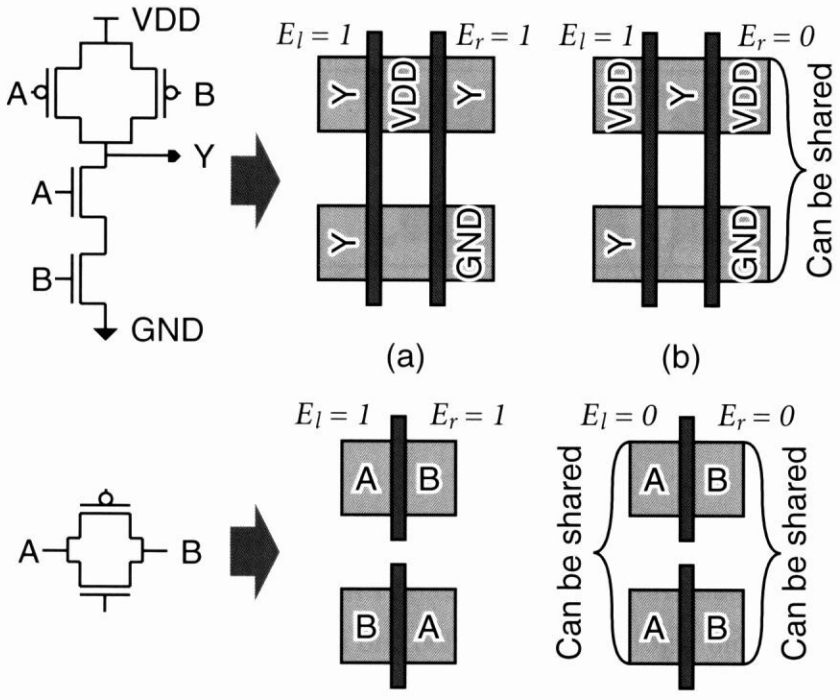

Figure 4.4 Additional variables introduced to maximize the number of the connections by diffusion sharing between logic blocks.

shared between logic blocks are power, ground, and the diffusions of the transmission gates. For example of the Figure 4.3, signal name "VDD", "GND", "A", and "B" can be shared by two blocks. These signals are called *inter-block sharing signals* hereafter. For maximizing the number of the connections by diffusion sharing in the inter-block placement step, it is better to place the diffusions that belong to the inter-block sharing signals on the edge of the placement of each block as illustrated in Figure 4.4. Therefore, we introduce new variables E_l and E_r, and a new objective function into the formulation of transistor placement to maximize the number of the connections by diffusion sharing. E_l (E_r) takes the value of 1 unless the diffusions of the left (right) edge are assigned to the pair of the inter-block sharing signals as illustrated in Figure 4.4. The new objective function is expressed as follows.

$$Minimize : (W + 1) \times (E_l + E_r) + \sum_{k=0}^{W-1} D(k) \tag{4.11}$$

In our formulation, the number of the connections by diffusion sharing between logic blocks has the higher priority than the number of the *different gate input pairs*. Therefore, E_l and E_r have the coefficient $W + 1$ to have larger cost than the *different gate input pairs*.

Inter-block placement: Inter-block placement is also based on Boolean Satisfiability. The constraints explained in Section 4.3 except "different gate constraints" and the "objective

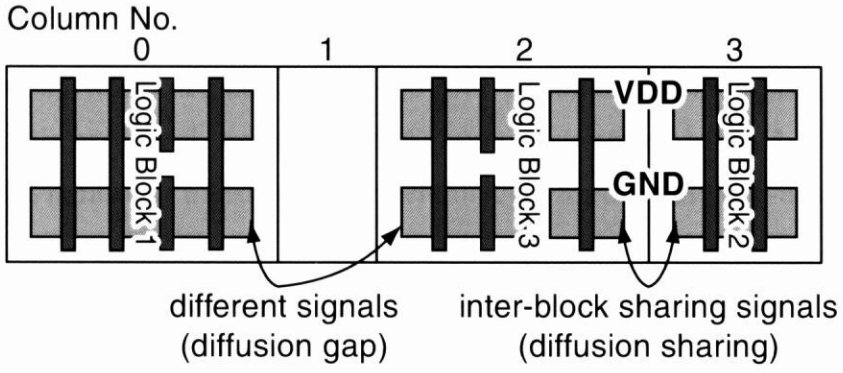

Figure 4.5 The problem definition of the inter-block placement.

function" are needed for the formulation of the inter-block placement. The problem definition of the inter-block placement is shown in Figure 4.5. In the formulation of the inter-block placement, each logic block is placed in one column and two logic blocks must not overlap in the same column. Two logic blocks facing the diffusions that can not be shared each other must not be placed in the neighboring column, as illustrated in Figure 4.5.

These constraints are expressed as CNF and PBF constraints in the same manner as explained in Section 4.3. Therefore, we obtain an exact minimum-width block placement using the same procedure as explained in Section 4.3. Finally, the flow of our hierarchical approach for transistor placement is described as follows.

1. A given transistor netlist is partitioned into logic blocks.

2. Exact minimum-width transistor placements are generated for each block using the new objective function and the same flow as explained in Section 4.3.

3. After placements of all blocks are generated, count the number of the logic blocks. The initial number of the column W is set to the number of logic blocks.

4. Search for a satisfiable assignment of the Boolean constraints for the inter-block placement of W columns. If a satisfiable assignment is found, these logic blocks can be placed in W columns and this procedure terminates. Otherwise, go to step 5.

5. $W = W + 1$. Go to step 4 again.

Although this approach has a possibility to generate wider placement than that of the flat approach, the experimental results show that it generates the transistor placements quickly with little increase in cell width.

4.5 Generalization to Multi-Row Transistor Placement

In this section, we generalize the single-row placement method for dual and non-dual CMOS cells explained previously to the multi-row placement and propose an exact minimum-width multi-row transistor placement method for dual and non-dual CMOS cells.

4.5.1 Problem Definition

When X N type transistors, Y P type transistors, and the number of the P/N row R are given, we have to place these $X + Y$ transistors in the minimum width. This problem can be transformed into the problem which places all transistors using the minimum number of columns. Figure 4.6 illustrates the problem definition of the proposed multi-row transistor placement. *Column*, *row*, and *boundary* are defined as shown in Figure 4.6 (a), respectively. Although we assume two P/N rows and these rows abut their P transistor side each other (so called NPPN style) in this figure, the number of P/N rows and their directions can be changed. Moreover, our formulation can be extended to handle other styles, *e.g.*, NPN or PNP styles, and so on. Two horizontally neighboring transistors face the diffusions that belong to the common signals each other to connect their source or drain by diffusion sharing. The empty columns result in the diffusion gaps in the final layout as illustrated in Figure 4.6 (b). When two vertically neighboring transistors have different gate input signals, these gate signals can not be connected. This point is called a *different gate* as shown in Figure 4.6 (b). *Different gate* is defined on the *boundary*. If one of the vertically neighboring grids has a transistor and the other does not, this point is also called a *different gate*. The conventional multi-row transistor placement method does not consider the gate connection between two P/N rows. Our gate connection style is a key to handling the multi-row placement of CMOS cells including non-dual P and N type transistors efficiently. Under these layout styles, the multi-row transistor placement problem is transformed into the Conjunctive Normal Form (CNF) and Pseudo-Boolean Form (PBF) constraints.

4.5.2 Placement Formulation

In our formulation of the multi-row transistor placement, $C \times R + 1$ variables are introduced for each transistor to identify its location and whether it is flipped or not, where C and R are the number of columns and rows of the placement area, respectively. The value of R is assumed to be given as an input. Additional $C \times (2 \times R - 1)$ variables are needed to identify

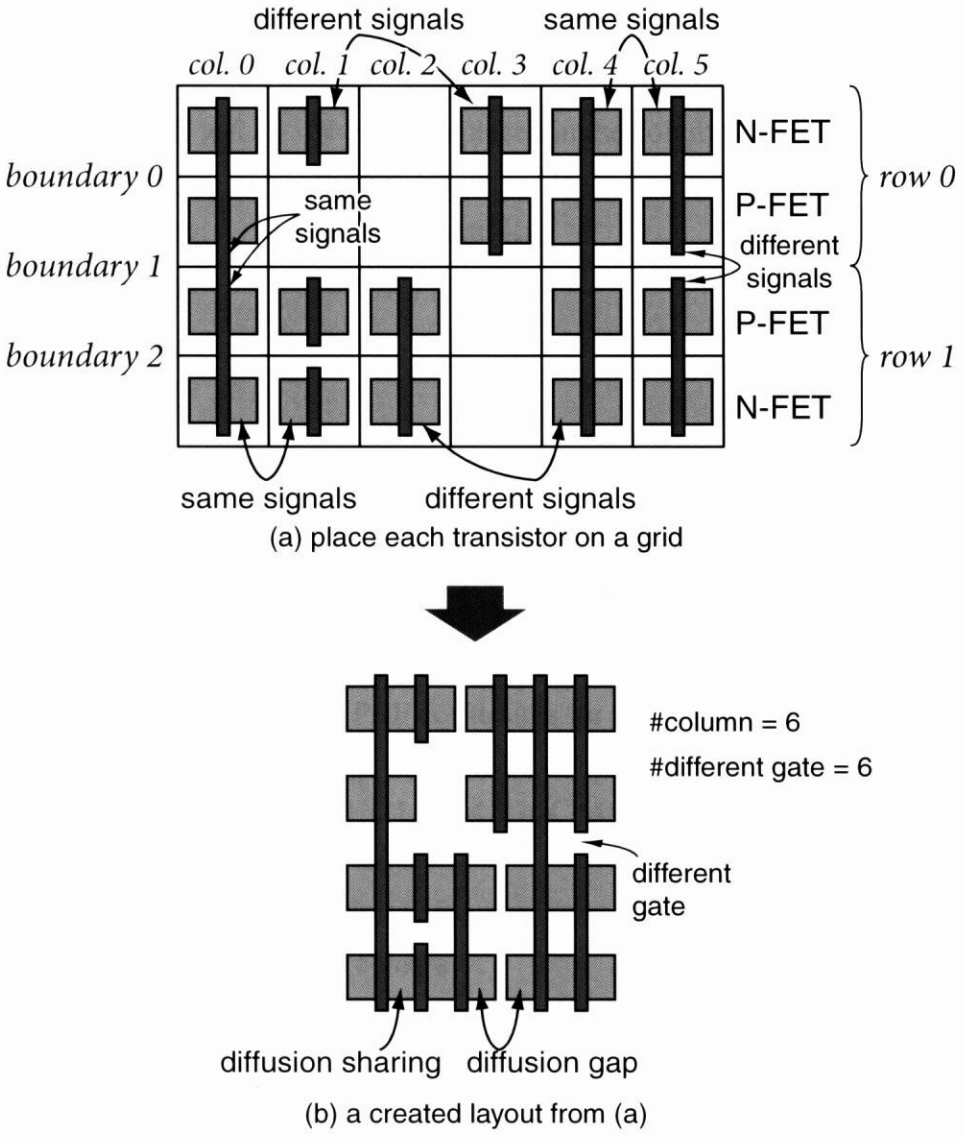

Figure 4.6 The problem definition of the multi-row transistor placement for dual and non-dual CMOS cells.

whether there is a *different gate* on each *boundary* of vertically neighboring grids or not. The number of the *different gate* is minimized in our formulation for ease of intra-cell routing which follows the transistor placement. All variables needed for this formulation are listed in Table 4.3. Table 4.3 shows the names of the variable type, the numbers of each type of variables, and the conditions when each type of variables takes the value of 1. We formulate the following Boolean constraints which express all the possible transistor placements in C columns and R rows under our layout style.

Table 4.3 The variables used for the formulation of the multi-row placement.

name	*number*	*condition which makes the value 1*
$L_n(i,k,l)$	$X \times C \times R$	N-FET i is placed in the column k, row l.
$L_p(j,k,l)$	$Y \times C \times R$	P-FET j is placed in the column k, row l.
$F_n(i)$	X	N-FET i is flipped.
$F_p(j)$	Y	P-FET j is flipped.
$D(k,l)$	$C \times (2 \times R - 1)$	Different gate is placed in the column k, boundary l.

Transistor overlap constraint: N type transistors must not overlap in the same *column* and the same *row*. This constraint is expressed by the following PBF constraints.

$$\sum_{i=0}^{X-1} L_n(i,k,l) \leq 1, \quad 0 \leq k < C, \quad 0 \leq l < R \tag{4.12}$$

The same constraint is expressed as PBF constraints for P type transistors.

$$\sum_{j=0}^{Y-1} L_p(j,k,l) \leq 1, \quad 0 \leq k < C, \quad 0 \leq l < R \tag{4.13}$$

Transistor instantiation constraint: Each transistor must be instantiated once. The following PBF constraints express this constraint.

$$\sum_{k=0}^{C-1} \sum_{l=0}^{R-1} L_n(i,k,l) = 1, \quad 0 \leq i < X \tag{4.14}$$

$$\sum_{k=0}^{C-1} \sum_{l=0}^{R-1} L_p(j,k,l) = 1, \quad 0 \leq j < Y \tag{4.15}$$

Neighboring transistors constraint: Two N type transistors facing the diffusions which belong to the different signals each other can not be placed in the horizontally neighboring grids. This constraint is expressed by the following logic equation.

$$GAP_n(i,j) \wedge \bigvee_{k=0}^{C-2} (L_n(i,k,l) \wedge L_n(j,k+1,l)) = 0 \tag{4.16}$$
$$i \neq j, \quad 0 \leq i < X, \quad 0 \leq j < X, \quad 0 \leq l < R$$

Here, $GAP_n(i,j)$ is the logic function of $F_n(i)$ and $F_n(j)$, and takes the value of 1 if N type transistor i can not share its diffusion with N type transistor j placed to its immediate right, otherwise 0. This logic equation is expressed as CNF. The same constraint is also expressed

as follows for P type transistors.

$$GAP_p(i,j) \wedge \bigvee_{k=0}^{C-2}(L_p(i,k,l) \wedge L_p(j,k+1,l)) = 0 \tag{4.17}$$
$$i \neq j,\ 0 \leq i < Y,\ 0 \leq j < Y,\ 0 \leq l < R$$

Objective function: The number of the *different gate* is minimized for ease of intra-cell routing which follows the transistor placement. The objective function is expressed as follows.

$$Minimize: \sum_{k=0}^{C-1}\sum_{l=0}^{2R-2} D(k,l) \tag{4.18}$$

Definition of $D(k,l)$: When a *different gate* is placed in the *column* k, *boundary* l, the value of $D(k,l)$ must be 1. Inside *row* $r(=l/2)$, this definition is expressed by the following logic equation,

$$D(k,l) = \bigvee_{i,j}(L_n(i,k,r) \wedge L_p(j,k,r)), \tag{4.19}$$
$$0 \leq k < C,\ 0 \leq r < R$$

$$GATE_n(i) \neq GATE_p(j),\ 0 \leq i < X,\ 0 \leq j < Y \tag{4.20}$$

where $GATE_n(i)$ and $GATE_p(j)$ mean the gate input signals of N type transistor i and P type transistor j, respectively. This logic equation is transformed into the PBF constraints as follows for all combinations of i and j expressed by (4.20)

$$L_n(i,k,r) + L_p(j,k,r) \leq D(k,l) + 1, \tag{4.21}$$
$$0 \leq k < C,\ 0 \leq r < R$$

under the condition that $D(k,l)$ is minimized. The minimization condition of each $D(k,l)$ value is already expressed by the objective function (4.18). Therefore, this definition is simply expressed by the PBF constraints (4.21) for all combinations of i and j expressed by (4.20). When the *column* k, *row* r has either a P or an N type transistor, $D(k,l)$ also takes a value of 1. $D(k,l)$ is defined not only inside each P/N row (*e.g.*, boundaries 0 and 2 in Figure 4.6 (a)), but also on the boundary between two P/N rows (*e.g.*, boundary 1 in Figure 4.6 (a)) in similar manner.

Finally, we can determine whether given transistors can be placed in C columns and R rows or not, using these constraints. If no satisfiable assignment is found by a Boolean

constraint solver, it is guaranteed that there is no possible placement of C columns and R rows. Therefore, we can find an exact minimum-width multi-row transistor placement using the procedure described below.

1. A netlist and the number of rows R are given.
2. For a given transistor netlist, count the number of N and P type transistors. The initial number of column C is set to $\lceil max(\#\text{N-FET}, \#\text{P-FET}) / R \rceil$[4.1].
3. Search for a satisfiable assignment of the Boolean constraints constructed for C columns and R rows. If a satisfiable assignment is found, these transistors can be placed in the C columns and R rows, and this procedure terminates. Otherwise, go to step 4.
4. $C = C + 1$. Go to step 3 again.

4.6 Experimental Results

The proposed flat and hierarchical single-row transistor placement methods and multi-row transistor placement method were implemented to show the effectiveness of the proposed methods. In this experiment, we used *PBS*[30] for a Boolean constraint solver. *PBS* can handle CNF constraints, PBF constraints, and optimizations. The results of each placement method are shown in the following sections.

4.6.1 Single-Row Flat Approach

The characteristics of the cells used for the experiment of single-row placement methods are shown in Table 4.4. Tables 4.5 and 4.6 show the comparison with the exact transistor placement method for dual cells explained in Chapter 2, which is applicable to the cells that can not be applied to other conventional exact methods such as [22] and [19]. Moreover, it theoretically generates the same or smaller width placement than that of other existing exact transistor placement methods for dual cells.

Table 4.5 compares the problem sizes of each formulation, and Table 4.6 compares the numbers of columns of generated placement, and the total runtimes of each method for generating the minimum-width transistor placement from a netlist. In Table 4.5, *#variable*, *#clause*, and *#inequality* mean the number of variables, clauses, and inequalities of each

[4.1] $\lceil X \rceil$ indicates a minimum integer which is equal to or larger than X.

Table 4.4 Cells used for the experiment of the single-row transistor placement.

Cell No.	*Circuit*	*#transistor*
1	Series-parallel circuit for $Y = (A \vee B) \wedge C$	10
2	Three-state buffer	12
3	2-input multiplexer	14
4	2-input AND	15
5	Series-parallel circuit for $Y = (A \wedge B) \oplus C$	18
6	3-input exclusive OR	20
7	3-input exclusive NOR	22
8	D Flip Flop	24
9	D Flip Flop (buffered)	26
10	3-input exclusive NOR (buffered)	28

formulation, respectively. Since both methods are based on the Boolean satisfiability, the problem size of each formulation is also compared in this table. In Table 4.6, *#column* and *#different* mean the number of columns and the number of the *different gate input pairs* of generated placement, respectively. The value of *#different* for the placement method only for dual cells is always 0, since it always makes pairs of P and N type transistors with a common gate input signal. The proposed method used *PBS* for a Boolean constraint solver since it creates CNF, PBF, and optimization constraints, whereas the placement method for dual cells used *Chaff*[25] for a CNF-SAT solver since it creates only CNF constraints. Since *PBS* is proposed to generalize the algorithms used in *Chaff* to solve 0-1 ILP problems that may include PBF and optimization constraints, *Chaff* finds solutions more quickly to the problems of pure CNF constraints. For some cells which have more than 20 transistors, we have found that *Chaff* can find solutions much more quickly to the problems created by the placement method for dual cells than PBS. Therefore, the placement method for dual cells used *Chaff* in this experiment.

The cell number 2 (three-state buffer) has transistors which can not be paired by the common gate input signals. Although the cell number 4 (2-input AND) is an intrinsically dual CMOS circuit, this cell also has non-dual P and N type transistors as a result of transistor folding. Therefore, the placement method only for dual cells can not be applied to these cells, whereas the proposed flat approach generates the exact minimum-width transistor placements of them as shown in Table 4.6. In the case of the cell number 10, the runtime was over 3600

Table 4.5 Problem size comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach.

Cell No.	*Problem Size*				
	Dual		*Proposed*		
	#variable	*#clause*	*#variable*	*#inequality*	*#clause*
1	40	760	65	121	488
2	N/A	N/A	103	251	1104
3	70	2522	103	251	1104
4	N/A	N/A	143	336	1960
5	90	4588	208	699	4212
6	100	6412	251	1011	6320
7	110	8478	321	1427	9120
8	144	17122	349	1163	11040
9	130	13070	404	2127	14170
10	140	15516	434	2465	15860

Table 4.6 Width and runtime comparison results between the exact single-row transistor placement method for dual cells and the proposed flat approach.

Cell No.	*Width*			*Runtime (sec.)*	
	Dual	*Proposed*		*Dual*	*Proposed*
	#column	*#column*	*#different*		
1	6	**5**	2	0.03	0.04
2	N/A	7	2	N/A	0.15
3	9	**7**	2	0.86	0.17
4	N/A	8	1	N/A	0.49
5	10	10	0	1.66	2.03
6	14	**11**	2	55.82	29.06
7	15	**13**	2	159.63	969.96
8	18	**13**	4	1382.98	1240.83
9	—	—	—	>3600	>3600
10	16	—	—	1730.13	>3600

— : the procedure does not terminate after 3600 seconds

Table 4.7 Comparison results between our flat and hierarchical approaches of the single-row transistor placement.

Cell No.	Width				Runtime (sec.)	
	Flat		Hierarchical		Flat	Hierarchical
	#column	#different	#column	#different		
1	5	2	**6**	0	0.04	0.02
2	7	2	7	2	0.15	0.02
3	7	2	7	2	0.17	0.14
4	8	1	8	1	0.49	0.03
5	10	0	10	**2**	2.03	0.08
6	11	2	11	**4**	29.06	0.29
7	13	2	13	**4**	969.96	3.39
8	13	4	13	4	1240.83	0.31
9	—	—	15	4	>3600	1.21
10	—	—	16	4	>3600	1.85

— : the procedure does not terminate after 3600 seconds

seconds for the flat approach, and that of both the flat approach and the placement method for dual cells were over 3600 seconds in the case of the cell number 9. In the other cases, the proposed method generated the same or smaller width placements by introducing *different gate input pairs*. Although the intra-cell routing may become difficult by introducing the *different gate input pairs*, it is easy to complete it if using the second metal layer. The runtimes of the proposed flat approach were comparable to the exact method for dual cells for cells with about 20 transistors, but the runtimes became over one hour for cells with about 26 transistors whereas the dual method could solve the cells with about 28 transistors in one hour under our experimental setup. The next experimental results show that these runtimes can be reduced drastically using the hierarchical approach.

We also compared the proposed transistor placement method for non-dual cells with the commercial cell generation tool *ProGenesis*[13] for the "mux2" circuit which the transistor placement method for dual cells proposed in Chapter 2 generates larger width placements than the commercial tool as shown in Table 2.4. The method in Chapter 2 generates the larger width placement because of the gate connection restriction between P and N type transistors, and generates the placement of 8 columns whereas the commercial tool generates the

6 column placement. The transistor placement proposed in this chapter has no restriction in the gate connection between P and N type transistors and the experimental result demonstrated that the proposed method generates the 6 column placement for this circuits. From this result, we can conclude that the proposed transistor placement method considers the gate connection between P and N type transistors more efficiently than the placement method for dual cells, and generates more area-efficient transistor placement.

4.6.2 Single-Row Hierarchical Approach

Table 4.7 shows the comparison result between our flat and hierarchical approaches. This table compares the numbers of the columns and the *different gate input pairs* of the generated placements, and the total runtimes. The width of the generated placement increased in the case of the cell number 1, and the numbers of different gate input pairs increased in the cases of the cells number 5, 6, and 7. This table clearly shows that the runtimes for transistor placement was drastically reduced by our hierarchical approach with little increase in cell width. Moreover, this hierarchical approach could solve the cells with larger number of transistors. The solutions for the cells number 9 and 10 were generated within two seconds by the hierarchical approach, whereas the flat approach could not generate the solutions within one hour.

Table 4.8 compares the numbers of cells that can be applied to the exact transistor placement method for dual cells explained in Chapter 2 and the proposed flat and hierarchical approaches in the case of an industrial standard-cell library of a 90nm technology including 340 cells. This table shows the numbers of the cells and coverage ratios of the cells that are applicable to each method and that can be solved in 3600 seconds. Since the proposed method is applicable to CMOS cells with any types of structures, this method was applied to all of the 340 cells, whereas the method for dual cells was about a half. The coverage ratios of the cells that can be solved in 3600 seconds by the flat approach and the method for dual cells are 43% and 32%, respectively. The number of the cells solved by the flat approach was larger than that of the dual one since the proposed method can be applied to non-dual circuits as well. Among the cells solved by both methods, the proposed flat approach generated smaller width placements for 29 out of 103 dual cells by introducing the *different gate input pairs*. Using the hierarchical approach, the coverage ratio increased to 81%, and only 3 out of 147 cells have one column wider placement. Among the 144 cells whose width are kept minimum, only 16 cells have larger number of *different gate input pairs* than the placement

Table 4.8 Comparison of the number of cells that can be applied to the proposed exact single-row transistor placement method and the method for dual cells in the case of a cell library with 340 cells.

placement method	*applicable*		*solvable*	
	#cells	*coverage*	*#cells*	*coverage*
Dual	164	48%	110	32%
Proposed (flat)	340	100%	147	43%
Proposed (hierarchical)	340	100%	274	81%

Table 4.9 Cells used for the experiment of the multi-row transistor placement.

Cell No.	*Circuit*	*#transistor*
1	Series-parallel circuit for $(A0 \vee A1) \oplus B$	12
2	Half-adder	14
3	Series-parallel circuit for $(A0 \vee A1) \oplus B$ (buffered)	18
4	D latch	19
5	3-input XNOR	20
6	2-input multiplexer	21
7	3-input XNOR (buffered)	22
8	D flip flop	24
9	4-input multiplexer	26
10	Full adder	28

generated by the flat approach. The rest of 19% cells can not be solved within one hour by the proposed hierarchical approach since these cells have logic blocks with large number of transistors even after the partitioning procedure. We expect the hierarchical approach can solve all of the cells by improving the partitioning algorithm to partition the large logic blocks into multiple blocks. Although the total time for generating 81% of 340 cells was about 7 hours, a large part of this time was consumed by several cells. Except these time-consuming cells, 76% of 340 cells were solved in 30 minutes. This result also shows that our hierarchical approach reduced the runtimes drastically with little increase in cell width.

4.6.3 Multi-Row Placement

We used another set of 10 CMOS cells from a standard-cell library of 90nm technology for the experiment of multi-row transistor placement. The characteristics of the cells used in this experiment are summarized in Table 4.9. Table 4.10 summarizes the problem size of the proposed multi-row transistor placement formulation and Table 4.11 shows the comparison results with the conventional style multi-row placement method[19]. In the conventional style, only P and N type transistors with the same gate input signal can be placed in the same column in each P/N row, and the gate connection between two adjacent P/N rows is not considered. This conventional style placement method was also formulated in the same manner as the proposed method for comparison in this experiment, although the original method[19] was solved using ILP. We assumed two P/N rows and NPPN placement style for both methods.

In Table 4.10, *#variable*, *#clause*, and *#inequality* mean the number of variables, clauses, and inequalities of the proposed formulation, respectively. Table 4.11 shows the numbers of columns of generated placement, and the total runtimes of each method for generating the minimum-width transistor placement from a netlist. In the column of width, *#column* and *#different* mean the number of columns and the number of the *different gate* of generated placement, respectively. Since the conventional style does not consider the gate connection between two adjacent rows, this table does not show the number of *different gate* for the conventional style. However, it is notable that the gates of the P/N transistor pair inside each P/N row are always connected in the conventional style.

The cells number 4 and 6 in Table 4.9 have non-dual structure and have transistors which can not be paired by the common gate input signals. Therefore, the conventional style can not be applied to these cells, whereas the proposed method generates the exact minimum-width transistor placements of them as shown in Table 4.11. In the case of the cell number 10, the runtime was over 3600 seconds for the proposed method. In the other cases, the proposed method generated the same or smaller width placements by using our gate connection style. The runtimes of the proposed method were comparable to the case of using conventional style for the cells with up to 22 transistors and much shorter for the cells number 8 and 9, but the runtime became over one hour for the cell with 28 transistors whereas the conventional style could solve this cell within one hour under our experimental setup. Although the runtime depends not only on the number of the transistors but also on the connections inside the cell, these results show that the proposed method can solve the cells with up to 26 transistors in

Table 4.10 Problem size of the proposed multi-row transistor placement formulation.

Cell No.	*#transistor*	*Problem Size*		
		#variable	*#inequality*	*#clause*
1	12	93	294	504
2	14	138	486	1200
3	18	213	1028	2496
4	19	265	1399	3460
5	20	278	1628	3560
6	21	291	1653	4700
7	22	304	1954	4480
8	24	330	2280	5360
9	26	411	3162	6816
10	28	441	7344	3796

Table 4.11 Width and runtime comparison results of the proposed multi-row transistor placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals.

Cell No.	*#transistor*	*Width*			*Runtime (sec.)*	
		Conventional	*Proposed*		*Conventional*	*Proposed*
		#column	*#column*	*#different*		
1	12	4	**3**	4	0.13	0.04
2	14	4	4	2	0.21	0.21
3	18	5	5	2	1.47	1.81
4	19	N/A	6	8	N/A	90.30
5	20	7	**6**	5	37.67	87.99
6	21	N/A	6	3	N/A	42.09
7	22	7	**6**	6	34.91	157.09
8	24	—	6	8	>3600	47.00
9	26	—	7	8	>3600	631.27
10	28	8	—	—	735.05	>3600

— : the procedure does not terminate after 3600 seconds

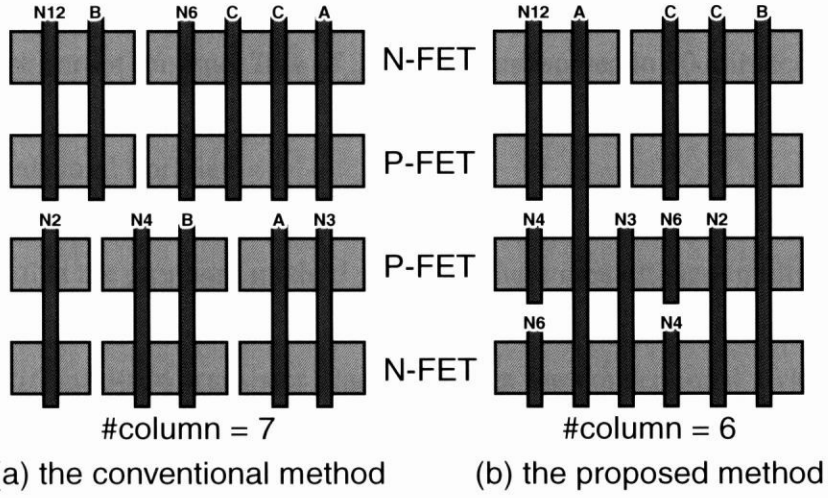

Figure 4.7 Examples of the layout of the cell number 7 in Table 4.9 created by (a)the conventional and (b)the proposed multi-row placement method. The conventional style assumes the pair of P/N transistors with the same gate input signals.

reasonable time. Figure 4.7 shows examples of the layout of the cell number 7 in Table 4.9 created by the conventional and the proposed method. This result clearly shows that the proposed method considers the gate connection more efficiently and creates more area-efficient multi-row transistor placement than the conventional one.

4.7 Summary

This chapter proposed flat and hierarchical approaches for generating a minimum-width single-row transistor placement of CMOS cells in presence of non-dual P and N type transistors, and generalized the single-row placement method to the multi-row placement method. Our approaches are the first exact minimum-width transistor placement method for non-dual CMOS cells. The flat single-row approach generated smaller width placement for 29 out of 103 dual cells than the placement method for dual cells explained in Chapter 2 which theoretically generates the smallest width placement among the existing exact methods for dual cells. The experimental results showed that it is not only applicable to CMOS cells with any types of structure, but also more effective even for dual CMOS cells compared with the transistor placement method only for dual cells.

The hierarchical single-row approach which is based on circuit partitioning reduced the runtime drastically and generated 81% of 340 cells in an industrial standard-cell library of a 90nm technology within one hour for each cell, whereas the flat approach and the exact

method only for dual cells generated 43% and 32%, respectively. Except several cells which consumed a large part of runtime, 76% of 340 cells were solved in 30 minutes. By improving the partitioning algorithm, we expect to implement a new hierarchical approach which covers 100% cells of industrial libraries.

The experimental results of the exact minimum-width multi-row transistor placement method showed that the proposed method generates more area-efficient multi-row placement than the conventional method only for dual cells by using the gate connection style which is more suitable for multi-row transistor placement than the conventional style and can solve the cells with up to 26 transistors in reasonable runtime.