

A Cell Synthesis Method for Salicide Process

Kazuhisa Okada, Takayuki Yamanouchi and Takashi Kambe
Design technology development center, Sharp corporation
2613-1, Ichinomoto Tenri Nara, 632-8567, Japan
Tel: +81-743-65-2531
Fax: +81-743-65-3814
email: {okada,t_yamau,kambe}@icg.tnr.sharp.co.jp

Abstract In this paper, we propose a cell synthesis method for a Salicide process. Our method utilizes the local interconnect between adjacent transistors, which is available in some Salicide processes, and optimizes the transistor placement of a cell considering both area and the number of local interconnects. In this way we reduce the number of metal wires and contacts. The circuit model is not restricted to conventional series-parallel CMOS logic, and our method enables us to synthesize CMOS pass-transistor circuits. Experimental results show that our method uses the local interconnect effectively, and optimizes both cell area and metal wire length.

I. INTRODUCTION

Logic synthesis and automatic place-and-route are key technologies for rapid time-to-market SoC (System on a Chip) design. When using such methods the quality of the cell library is a critical factor, because the circuit performance, area, and routability depend on the library cells. In order to shorten the period of library development, a technology for synthesizing high quality cells is almost indispensable.

Moreover, cell layout has become diversified due to process technology innovation. For example, Salicide process technology introduced a silicide material in order to reduce diffusion resistance. In some Salicide processes, silicide material can also be used to connect adjacent transistors. We call this feature *local interconnects* in this paper.

The local interconnect enables us to connect the source or drain of nMOS and pMOS transistors directly; that is to say it can connect them without metal wires. Fig.1(a) shows an example of local interconnects. In the figure, the drains of transistors P1 and N1 are connected with local interconnect, and the drains of transistors P2, N2, and the source of the transistor N1 are also connected directly.

Local interconnect can reduce contacts, wires, and area of cells, thus it can improve the speed of the circuit and routability during auto-layout. However, conventional cell synthesis algorithms focus on minimizing the diffusion gap for CMOS logic cells with series-parallel structure[1-6]. To our knowledge there are no studies on cell synthesis algorithms which can use local interconnect in recent Salicide process technolo-

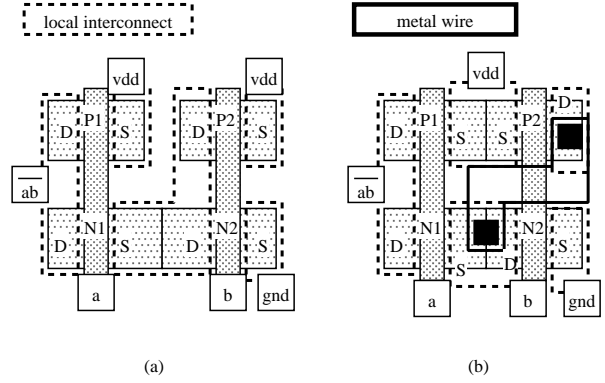


Fig. 1. Two layouts of pass-transistor NAND (a) no metal wire (b) low area

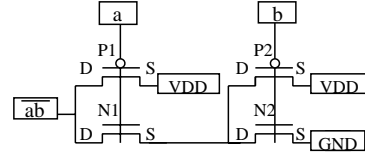


Fig. 2. Pass-transistor NAND

gies.

In this paper we propose a cell synthesis method for a Salicide process. Our method uses local interconnect between adjacent transistors, and optimizes both area and the number of local interconnects; thereby reducing the number of metal wires and contacts.

Taking the features of local interconnect into account, we can implement the same circuit in different ways. Because the local interconnect has higher degree of freedom than the conventional cell synthesis model, we have to consider the trade-off between cell area and metal wire length in our synthesis model.

Fig.1 shows this trade-off example. Fig.1(a) and Fig.1(b) realize the same circuit, which is shown in Fig.2. In Fig.1(a), there are no metal wires, but there are two contacts and a wire

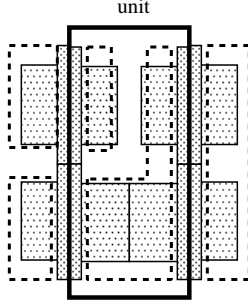


Fig. 3. Example of a placement unit

in Fig.1(b). Therefore, in the case of Fig.1(a), the resistivity is lower, and metal wires can be used for over the cell routing in vertical direction. On the other hand, the cell area of Fig.1(b) is smaller than that of Fig.1(a), because more space is required between P1 and P2 in Fig.1(a) to isolate their source and drain.

These trade-offs depend strongly on the process technology of the cell to be implemented. In our method, we can control the trade-offs by tuning the parameters of the cost function which is defined in our synthesis algorithm. Since we adopt a cost function approach in our method, we can synthesize different cell layouts with different features for use in different contexts within a single circuit.

This paper is organized as follows: Section II define the layout model of cell synthesis; section III describes the synthesis algorithm; section IV shows the effect of our method in experiments; and section V is our conclusion.

II. LAYOUT MODEL

A. Definition of units

Most conventional cell synthesis methods are based on a one-dimensional transistor row model for each type of transistor, and focus on minimizing diffusion gaps between MOS transistors of the same type (i.e. two pMOS or two nMOS). Thus a MOS transistor is the basic unit of their placement model. However, these methods cannot easily evaluate the local interconnect between complementary types of MOS transistors (i.e. between pMOS and nMOS transistors), because adjacent transistors in different rows affect the area of the cell as do adjacent transistors in same row. Moreover, our algorithm can increase the number of vertical routing tracks using the local interconnects between pMOS and nMOS transistors, but the conventional methods cannot. We introduce a new layout model in order to take into account the local interconnects between opposite types of MOS transistors. We consider pairs of pMOS and nMOS transistors with the same gate signal, and define placement units based on the local connectivity between two adjacent pairs.

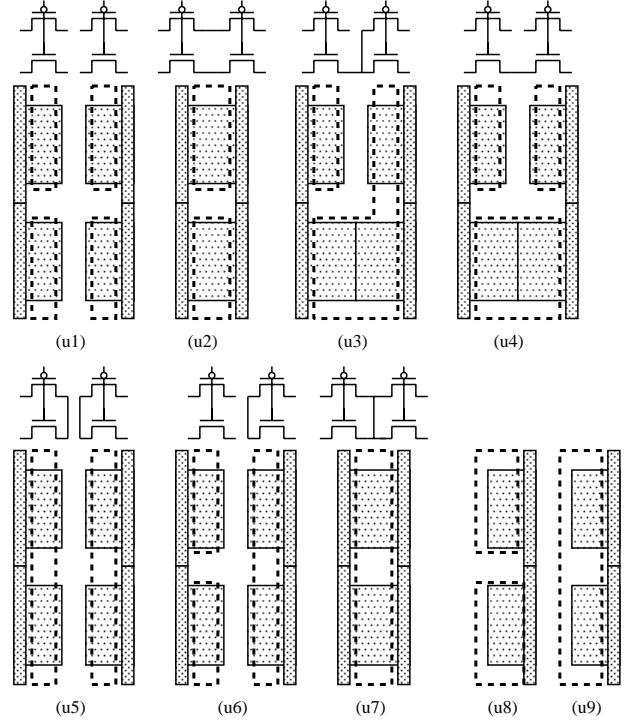


Fig. 4. Example of the unit set

Definition: A placement unit is a set of layout elements which consists of two transistor pairs facing each other and the local interconnects between them. The elements of the facing transistor pairs are the right-hand side of the left transistor pair and the left-hand side of the right transistor pair (Fig.3). Special units represent the transistor pairs at the end of the row. A unit set is a set of placement units which represents all possible patterns of interconnection between two adjacent transistor pairs. Note that a unit set includes special units.

This unit based modeling has great advantages for cell synthesis.

- We can prepare well designed units in advance, and the synthesis method need not care about complex design rules for local interconnections.
- The information for cell synthesis, namely cell width and the number of local interconnects, can be obtained very easily from the width of the units and the number of local interconnects in the units.
- Placing the unit in a one-dimensional row, we can easily complete the layout of all local interconnections in a cell.

Fig.4 shows the concept of the unit set. Units u1-u7 represent all possible patterns of interconnections between two adjacent transistor pairs, and units u8 and u9 are the special units. Table I gives information on the unit set, which is used in this paper.

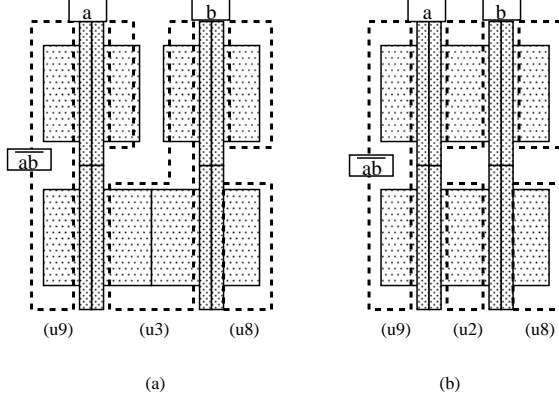


Fig. 5. Example of unit placement

TABLE I
NUMERICAL VALUE OF UNIT FEATURES
(LI : local interconnect)

unit	u1	u2	u3	u4	u5	u6	u7	u8	u9
width	1.5	1	1.5	1.5	1.5	1.5	1	1	1
# of vertical LIs	0	0	1	0	2	1	2	0	1
# of horizontal LIs	0	2	1	1	0	0	2	0	0

B. Cost function

In this section we explain the cost function for unit placement.

The following formula is the cost function for a unit:

$$f(u) = k_w \cdot w(u) - k_v \cdot v(u) - k_h \cdot h(u)$$

where u denotes a unit, $w(u)$, $v(u)$, $h(u)$ denote width and the number of vertical and horizontal local interconnection of unit u in Table I, respectively. $K = (k_w, k_v, k_h)$ is the parameter of the cost function. The cost of each unit indicates the priority during cell synthesis. The smaller $f(u)$ is, the better the unit is. The following formula is the cost function of a cell which consists of one or more units:

$$F(\text{Cell}) = \sum_{u_i \in \text{Cell}} f(u_i) \quad \dots (1)$$

To minimize the internal wire length, the parameter K is set to be $(1, 10, 1)$ for example, and we denote this cost function as $f_{\text{wire}}(u)$. Generally vertical wire lengths, which will be removed by vertical local interconnection, are longer than horizontal ones because transistor channels are relatively wide. For another example, to minimize the area K is set up to be $(10, 1, 1)$, and let $f(u)$ be $f_{\text{area}}(u)$ at this parameter setting. In practice, we must optimize the parameter for the target technology.

Table II shows the costs which are obtained by applying the two cost functions defined above to every unit in Table I. The

TABLE II
COST OF UNITS

unit	u1	u2	u3	u4	u5	u6	u7	u8	u9
$f_{\text{wire}}(u)$	1.5	-1	-9.5	0.5	-18.5	-8.5	-21	1	-9
$f_{\text{area}}(u)$	15	8	13	14	13	14	6	10	9

units including vertical local interconnects have least cost using cost function is $f_{\text{wire}}(u)$, and the narrow units have least cost using $f_{\text{area}}(u)$.

Next we show a cost evaluation example for unit placement of the cell. The cost of Fig.5(a) is calculated by formula (1) as;

$$\begin{aligned} F_{\text{area}}(\text{cell}) &= f_{\text{area}}(u9) + f_{\text{area}}(u3) + f_{\text{area}}(u8) \\ &= 9 + 13 + 10 = 32 \end{aligned}$$

and the cost of Fig.5(b) is 27, similarly.

III. ALGORITHM

A. Overview

Our cell synthesis algorithm optimizes the cost function to obtain optimal unit placement. The algorithm consists of two phases:

- unit assignment
- transistor ordering

In unit assignment a given transistor order is represented by placement of units and the cost of the unit placement is obtained. An optimum drain/source assignment is obtained for all transistors at the same time.

In the second phase the transistor order is improved iteratively to minimize cost. For each iteration, cost is obtained by unit assignment.

Our method supposes that the set of transistors can be partitioned into nMOS and pMOS transistor pairs, each pair having a common gate signal. CMOS logic and CMOS transfer gate logic satisfy this condition.

B. Unit assignment

Because each unit corresponds in our method to a connection pattern between transistors, the unit assignment of transistors is identical with the drain/source assignment of transistors when transistor order is fixed. The unit assignment and the drain/source assignment are obtained at the same time by solving an assignment graph.

Assignment graph

Definition: The assignment graph is defined as $G = \{V, E\}$. $V = \{v_s, v_t, v_{i,j} | i = 1, \dots, n, j = 1, 2, 3, 4\}$ denotes the set of nodes in the graph where n is the number of transistor pairs. Nodes v_s, v_t denote the source and sink nodes which correspond to the

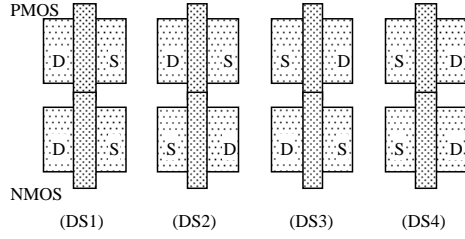


Fig. 6. Four patterns of drain/source assignment

cell boundaries. *DS1-DS4* denote four patterns of drain/source assignment of a transistor pair (Fig.6). The node $v_{i,j}$ denotes the assignment in which the i -th transistor pair use the DSj drain/source assignment pattern. $E = \{e(v_{i,j}, v_{i+1,k}) | i = 1, \dots, n-1, (j,k) = 1, 2, 3, 4\}$ denotes the set of edges of the graph. The edge $e(v_{i,j}, v_{i+1,k})$ is the edge between the nodes $v_{i,j}$ and $v_{i+1,k}$.

In an assignment graph, each node corresponds to one of the drain/source assignment patterns, each edge corresponds to a unit and a path from the source node to the sink node expresses a unit placement. Our method finds the optimum path in the assignment graph. A cell is synthesized by placing the units according to the path and by interconnecting the nets which are not connected by local interconnection.

Fig.7(a) shows an example of assignment graph. Nodes $v_{i,1}$, $v_{i,2}$, $v_{i,3}$ and $v_{i,4}$ are represented as ①, ②, ③ and ④, respectively. The path [⑤- ①- ③- ④- ①] represents drain/source assignments DS1, DS3, DS4 as shown in Fig.7(b). The same path is also expressed as the sequence of edges [$e(v_s, v_{1,1})$ - $e(v_{1,1}, v_{2,3})$ - $e(v_{2,3}, v_{3,4})$ - $e(v_{3,4}, v_t)$]. Every $e(x,y)$ will be related to one of the units by the process explained in the next paragraph. In our example the path edges are related to the nodes $u8, u3, u4, u9$.

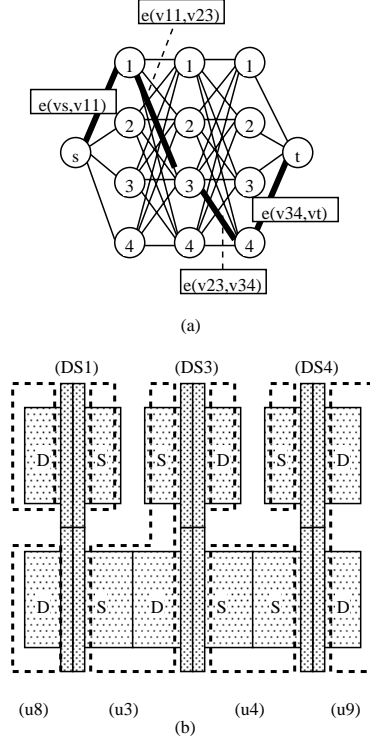


Fig. 7. Assignment graph (a)example of path (b)unit assignment

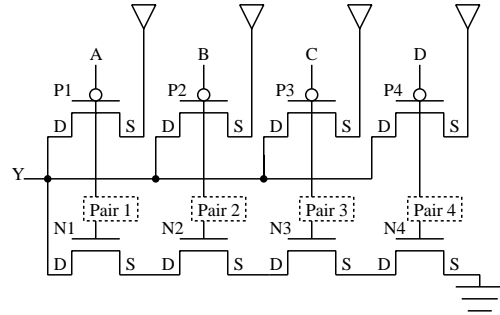


Fig. 8. CMOS 4 inputs NAND

Edge cost calculation In the following process every edge is related to a unit. The corresponding unit of $e(v_{i,j}, v_{i+1,k})$ is denoted by $u(v_{i,j}, v_{i+1,k})$ and its cost is denoted by $c(v_{i,j}, v_{i+1,k})$.

Here we describe the edge cost calculation algorithm.

1. placing transistor pairs (i) and ($i+1$) left to right with drain/source assignments DSj and DSk , respectively.
2. checking the existence of connections among the right diffusion regions of transistor pair (i) and the left diffusion regions of transistor pair ($i+1$) according to the netlist.
3. selecting the unit according to existence of connection and setting it to $u(v_{i,j}, v_{i+1,k})$.
4. evaluating the unit with the cost function and setting the cost to $c(v_{i,j}, v_{i+1,k})$

For example, consider the connection between transistor pairs 1 and 2 in Fig.8. For $e(v_{1,1}, v_{2,3})$, which corresponds to the

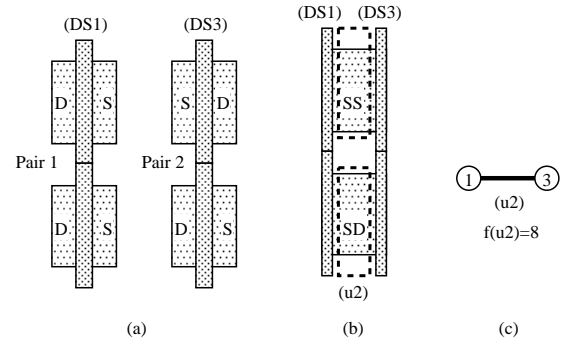


Fig. 9. Corresponding unit and cost of edge

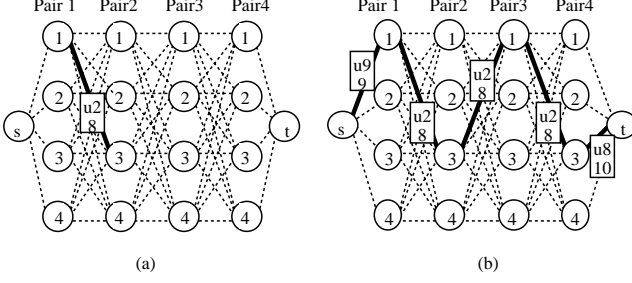


Fig. 10. (a)Attaching corresponding units and costs (b) Shortest path

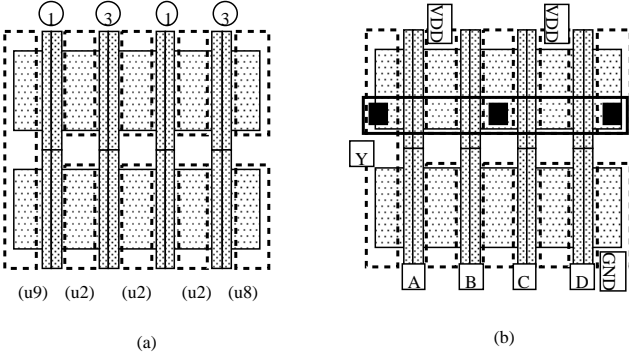


Fig. 11. (a)Optimum unit assignment (b)After metal wire routing

drain/source assignment DS1 for pair 1 and DS3 for pair 2, both of the right diffusion regions of P_1 and N_1 are sources, and the left regions of P_2 and N_2 are a source and a drain, respectively, as shown in Fig.9(a). According to the netlist (Fig.8), P_1 and P_2 , N_1 and N_2 turn out to be connected. Thus, unit $u2$ is set to $u(v_{1,1}, v_{2,3})$ which includes two horizontal local interconnects (Fig.9(b)). Cost $f_{area}(u2) = 8$ is set to $c(v_{1,1}, v_{2,3})$, as shown in Fig.9(c).

Fig.10(a) shows the assignment graph in which the corresponding unit and the cost (Fig.9(c)) are obtained. In the same way, we obtain units and costs for all edges.

The time complexity of the algorithm is $O(n)$, where n denotes the number of transistor pairs.

Shortest path in the assignment graph We can find the optimum unit assignment by solving the shortest path problem from source to sink in the assignment graph. The shortest path can be found with the time complexity $O(n)$, where n denotes the number of transistor pairs, because the assignment graph does not include any cycles.

For example, Fig.10(b) shows the shortest path when applying the cost function $f_{area}(u)$ to Fig.8. Fig.11(a) shows the unit placement in this case, and Fig.11(b) shows the synthesized cell in which metal wires have been attached.

TABLE III
EXPERIMENTAL RESULTS

circuit	nor8		nor6	
cost function	f_{wire}	f_{area}	f_{wire}	f_{area}
a. cell width	20.0	17.0	12.5	10.5
b. metal wire length	64.5	85.6	30.5	38.5
c. # of vertical LI	7	0	4	3
d. # of horizontal LI	16	22	11	14
e. # of porosities	18	8	10	6
f. processing time (sec)	3.0		1.0	

	or-and2222		or-and33		xor3	
	f_{wire}	f_{area}	f_{wire}	f_{area}	f_{wire}	f_{area}
a.	14.5	12.5	11.0	9.5	14.5	13.0
b.	47.3	48.8	25.1	24.1	43.8	43.0
c.	3	1	3	2	6	5
d.	11	14	10	12	8	13
e.	11	7	8	6	16	13
f.	1.5		1.2		1.0	

C. Transistor ordering

Since optimum unit assignment for a certain transistor order can be obtained by the algorithm in Sections III-B, we improve repeatedly the order of transistor pairs to obtain optimal cells with the following process:

1. determining initial transistor order by simple heuristic.
2. selecting two transistor pairs at random.
3. exchanging the order of these pairs.
4. applying unit assignment and obtaining the cost.
5. undoing step 3. if the cost gets worse.
6. repeating 2. to 5. a certain number of times.

The Repetition number in step 6. is a empirical parameter.

IV. EXPERIMENTAL RESULTS

Table III shows the cell synthesis results of some CMOS logic gates using both of the cost functions $f_{area}(u)$ and $f_{wire}(u)$. We set the repetition times of transistor order improvement in Section III-C to 100. We took the set of units shown in Table I.

When using the cost function $f_{wire}(u)$ the total length of metal wires in every cell is shorter and the number of vertical local interconnects is larger than when using $f_{area}(u)$. The number of porosities (tracks for vertical metal wires on the cell) is also larger. These results show that the cost function $f_{wire}(u)$ works well and vertical local interconnections take the place of some metal wires.

When using the cost function $f_{area}(u)$ the width of every cell is smaller than that of $f_{wire}(u)$. These results show that $f_{area}(u)$

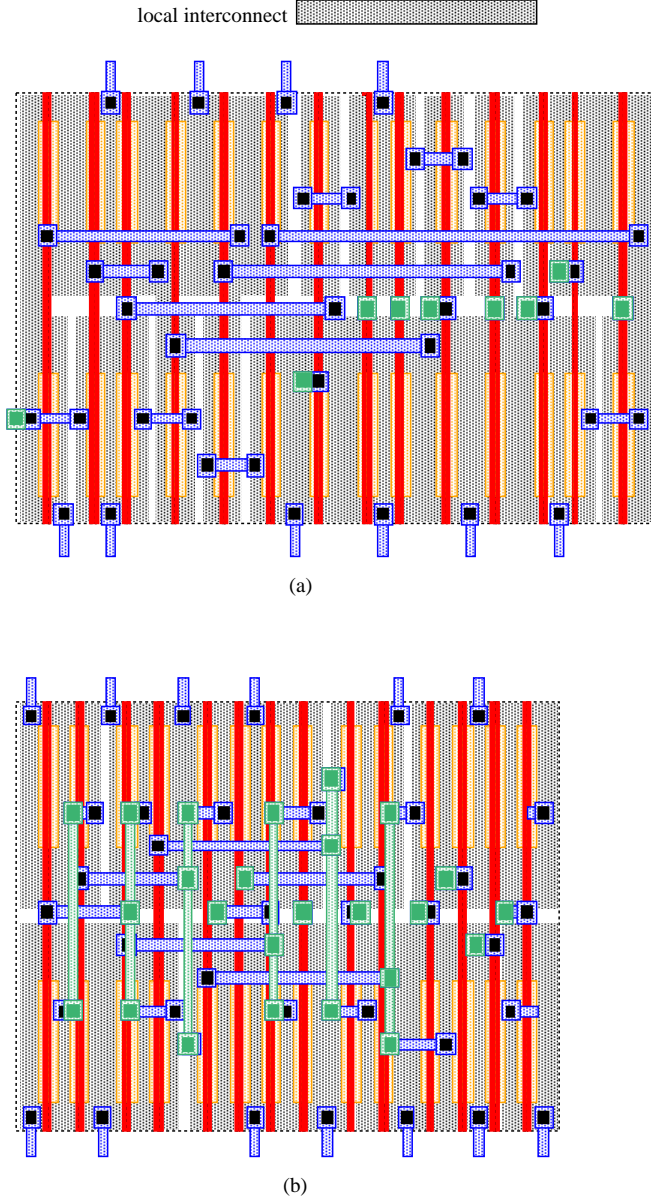


Fig. 12. Cell synthesis results with two cost functions (a) f_{wire} (b) f_{area}

also works well. The number of horizontal local interconnects is larger in this case, because narrow units have horizontal local interconnects in the unit set of Table I.

The processing time in Table III is measured with UltraSparc 30.

Fig.12(a) and (b) show the cell synthesis results for a CMOS 8 input NOR using cost functions $f_{wire}(u)$ and $f_{area}(u)$, respectively. The former includes fewer metal wires, especially vertical wires, and the latter is smaller.

These experimental results prove that our method is flexible, that is to say, it uses local interconnection to optimize various objectives such as area, metal wire length and porosity, according to a flexible cost function.

V. CONCLUSION

In this paper, we have proposed a cell synthesis method which makes use of local interconnects between adjacent transistors. The set of units, which represent all possible patterns of local interconnection, is designed in advance; our method selects suitable units and places those units to synthesize optimal cells. Since we use a parameterizable cost function, we can optimize the cells for various objectives, such as area, metal wire length and porosity.

REFERENCES

- [1] K.Tani: "Two-Dimensional Layout Synthesis for Large-Scale CMOS Circuits", IEEE, Proceedings of International Conference on Computer Aided Design '91 pp.490-493, 1991.
- [2] A.Gupta and J.P.Hayes: "A Hierarchical Technique for Minimum-Width Layout of Two-Dimensional CMOS Cells," Proc. of 10th International Conference on VLSI Design, pp.15-20,1997.
- [3] S.Saika, M.Fukui, N.Shinomiya and T.Akino: "A Two-Dimensional Transistor Placement for Cell Synthesis," Proc. of Asia South-Pacific Design Automation Conference'97, pp.557-562, 1997.
- [4] Y.Sasaki, K.Rikino and K.Yano: "ALPS: An Automatic Layouter for Pass-Transistor Cell Synthesis," Proc. of International Conference on Computer-Aided Design'98, pp.227-232, 1998.
- [5] S.Rekhi, J.D.Trotter and D.H.Linder: "Automatic Layout Synthesis of Leaf Cells," Proc. of 32nd Design Automation Conference, pp.267-272, 1995.
- [6] C.C.Chen and S.L.Chow: "The Layout Synthesiser: An Automatic Netlist-to-Layout System," Proc. of 26th Design Automation Conference, pp.232-238, 1989.