Multi-Level Placement with Circuit Schema Based Clustering in Analog IC Layouts

Takashi Nojima [†,††] Xiaoke Zhu [†] Yasuhiro Takashima [††] Shigetoshi Nakatake [††] Yoji Kajitani[††]

† System Development Dept.
SII EDA Technologies Inc.
2–5, Hibikino, Wakamatsu-ku,
Kitakyushu, Fukuoka, 808–0135 Japan
{takashi.nojima, xiaoke}@sii.co.jp

†† Dept. of Information and Media Science
The University of Kitakyushu
1–1, Hibikino, Wakamatsu-ku,
Kitakyushu, Fukuoka, 808–0135 Japan
{takasima, nakatake, kajitani}@env.kitakyu-u.ac.jp

Abstract

This paper aims at developing an automated device-level placement for analog circuit design which achieves comparable quality to manual designs by experts. It extracts a set of clusters from a circuit schema as experts do. We provide a multi-level placement based on the Sequence-Pair by relaxing the shape of clusters from rectangles and allowing boundaries of clusters to be 'jagged'. The quality of placement is evaluated by a multi-objective according to an expert's guideline. We adopt a multi-step simulated annealing to balance a trade-off between the objectives. In experiments, we tested the placement for industrial examples. Our tool attained placements better than those by manual on the average by 10.8% and 6.8% with respect to area and net-length, respectively. It also achieved 1/730 layout time compared with the time by manual.

I. Introduction

In analog circuit design, the placement phase is critical for the performance since it influences all the parasitic layout effects. Also device mismatching puts a fundamental limit on the achievable accuracy of circuits. A large portion of the effort involved in analog circuit design is spent in the layout phase. The layout of analog circuits is still a manual, time-consuming and error-prone task.

Toward automatic analog layout systems, most of ideas are based on an approach to use 'cell-based' layout style : ILAC [1], KOAN/ANAGRAM II [2], PUPPY_A [6], and LAYLA [8]. An analog circuit usually consists of subcircuits, and each subcircuit is defined by a certain function. In the layout style, a subcircuit is regarded as a 'cell' on which designated layout rules are imposed with respect to the height, power, ground, I/O terminal positions and others [8]. We call such a subcircuit 'cell'. Tasks of layout designers are (i) place-and-route in a cell and (ii) assemble-and-interconnect with cells. The target of this paper is placement in a cell.

A cell has the following features in its layout. Its power and ground lines are laid out parallel and straight. An example of a cell and that of its layout are shown in Figure 1 and 2, respectively. In general, each cell consists of a few hundreds of devices.

There are several studies for device-level placement. One of them is rectangle-packing based placement that regards each device as a rectangle [5, 7, 11]. However, they have not overcome a manual placement yet. Placement is required to be satisfied layout rules, symmetric constraints, matching constraints, etc. The size of a cell is too large to layout successfully satisfying these layout rules. Experts can craft placement of such a small size as satisfying various constraints. The automatic placement in a cell is required to seek a placement that satisfies layout rules, not allowing the size of a cell to be large.

We note that experts derive some clusters from a schema of a circuit. Such a cluster usually corresponds to a certain function. They place devices in a cluster closely to each other. Furthermore, the size of a cluster is appropriate so that layout constraints imposed on devices are restricted inside a cluster. Therefore, a reasonable automatic placement starts from the observation of the manual design. Then we abstract the features we can implement. This paper provides a multi-level placement with schema based clustering.

Given a cell whose size is a few hundreds of devices, we introduce a concept of 'schema based clustering'. By observing the schema of a cell, we can derive a set of elemental functions from the cell, such that the functions are I/Os, amplifiers, current-mirrors, differential-pairs, etc. A subcircuit corresponding to a function is called 'cluster', which consists of a few tens of devices. The size is small enough to be laid out neatly. In Figure 1, there are three clusters, input-, amplifier-, and output-cluster.

Our strategy is described by steps as follows. (1) Derive clusters from a cell by observing the schema; (2) Place devices in each cluster; (3) Place clusters in a cell; These data and optimization are handled by the Sequence-Pair [3, 4]. The Sequence-Pair is well known to be able to get a rectangle packing quickly. If the shape of clusters is regarded as a rectangle, however, it may result in much wasted area at boundaries between clusters. Therefore, we add more two steps to relax the shape of clusters from rectangles. (4) Extract topologies between clusters and impose constraints on devices such that if a cluster *A* is placed in the left of a cluster *B*, a device in *A* must not be placed in the right of any device in *B*; We call the constraints 'jagged-boundary constraints'. The Sequence-Pair can represent them naturally. (5) Place all devices in a cell satisfying the constraints; After all, we will attain a placement with effective use of area by relaxing the shape of clusters from rectangles and allowing boundaries of clusters to be 'jagged'. As shown in Figure 2, each cluster is not a rectangle nor each boundary of clusters straight.

We implemented our multi-level placement faithfully to our idea, and experimented for several industrial instances to compare the results with respective manual designs. We adopted the following criteria to evaluate the quality of placements : (a) cell area, (b) total net length, and (c) number of intersections between edges of minimum spanning trees (MSTs). Our placement demonstrated a significant performance comparable to the corresponding manual design.

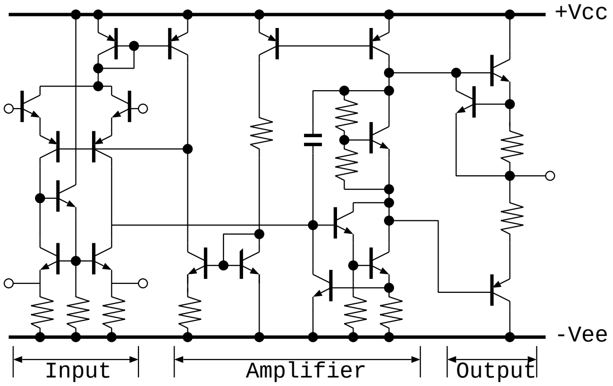

Fig. 1. Schema of cell μA741 consisting of three clusters

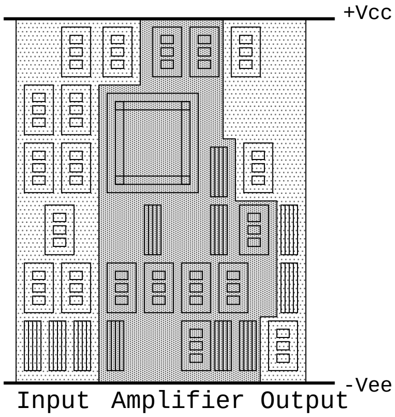

Fig. 2. Placement of the cell corresponding to schema in Figure 1

The rest of this paper is organized as follows. Section II describes our placement problem and the Sequence-Pair. Section III describes how the jagged-boundary constraints are represented by the Sequence-Pair. Section IV provides an optimization technique by simulated annealing. Section V is for experiments on industrial examples. Section VI summarizes the contribution and future works.

II. Placement Problem and Sequence-Pair

The target of this paper is placement of devices in a cell. Each device is represented as a rectangle which has width and height. The power and ground lines are laid out parallel and straight. Each device is laid out among power and ground not overlapping each other. The objectives are to minimize (a) cell area, (b) total net length, and (c) number of intersections of MSTs. An MST is defined in every net where nodes correspond to terminals and edges to connections.

We propose a hierarchical optimization algorithm for this device-level placement problem. Our approach is sketched as follows.

Step 1: Extract clusters from a cell guided by experts.

Step 2: Place devices in each cluster.

Step 3: Place clusters in a cell.

Step 4: Extract topologies between clusters and impose constraints on devices.

Step 5: Place all devices in a cell under the constraints.

In this approach, both placement and constraint are represented by the Sequence-Pair (SP hereinafter) [3, 4] which is described briefly for completeness.

A placement is an arrangement of rectangles on a plane without overlapping each other. Given n rectangles, an *SP* is an ordered pair of permutations Γ_+ and Γ_- of rectangle names. The k-th rectangle in Γ_* ($*$ is $+$ or $-$) is denoted as $\Gamma_*(k)$. While the position of rectangle x in Γ_* is denoted as $\Gamma_*^{-1}(x)$, by the inverse function.

It imposes positional relations, the *ABLR-relations*, between every pair of rectangles in the form that "one is above (below, left-of, right-of) the other".

ABLR-relations from an *SP* : For a pair of rectangles $\{a, b\}$,

- $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$, then a is left-of b (equivalently, b is right-of a).
- $\Gamma_+^{-1}(a) > \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$, then a is below b (equivalently, b is above a.)

Figure 3 shows a placement of seven rectangles derived from an *SP*. Note that the ABLR-relations derived from the *SP* are all satisfied. Once a system of ABLR-relations is given, it is easy to obtain a placement by using vertical and horizontal constraint graphs.

It is proved that any *SP* has its corresponding placement that can be obtained very quickly (in $O(n \log\log n)$ time [9]).

It is proved that any *SP* has a placement and any minimal placement has the corresponding *SP*. Hence searching the placements for optimization is almost equivalent to searching *SP*'s. This methodology of heuristic search has been established these ten years after the invention of BSG [5] and SP in 1995. This paper is also following this methodology.

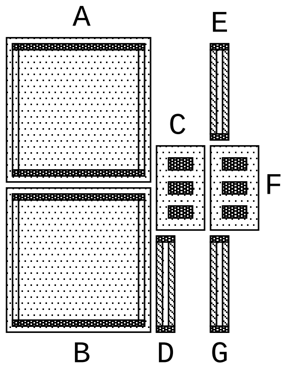

Fig. 3. Placement corresponding to ($ABCDEFG$; $BADCGFE$)

III. Jagged-Boundary Constraint

We can attain a placement of devices in each cluster by using the SP. Also, in a similar way, we can determine the relative positions of clusters. If ABLR-relations of clusters are inherited to ones of devices, the shape of cluster is the bounding box of the cluster's devices. Due to this over-estimation of area, it may result in much wasted area at boundaries between clusters. In this section, we relax the constraints on the shape of clusters to allow boundaries of clusters to be jagged. This will be empirically proved effective to limit a yield of wasted area at boundaries between clusters.

A similar study is found [10] in which free-rectangles are defined to fill notches between clusters. In analog layout design, however, it is not applicable because there are no devices corresponding to free-rectangles.

The object of using clusters is to place devices closely. It limits the ABLR-relations of devices. Consider the layout of a cell shown in Figure 2. The amplifier-cluster is placed in the right-of the input-cluster. We could guess that devices in the amplifier-cluster are right-of devices in the input-cluster since devices belonging to the same cluster are placed closely to each other. However, this is not completely satisfied : For example, several devices in the input-cluster are forced to be above the devices of the amplifier-cluster. Thus, the boundaries between clusters would be jagged in a manual design. We formulate such a jagged boundary in terms of ABLR-relations of devices.

First, we start from the observation of ABLR-relations among clusters. In an example of Figure 4, two clusters $A = \{a_1, a_2, a_3, a_4, a_5\}$ and $B = \{b_1, b_2, b_3, b_4, b_5\}$ are given. The ABLR-relation between A and B is referred to as "A is left-of B".

Assume that ABLR-relations of devices in clusters are inherited from those of clusters. Then, their ABLR-relations between a_i $\{i = 1, 2, \cdots, 5\}$, and b_j $\{j = 1, 2, \cdots, 5\}$ are as shown below.

$$\{\Gamma_+^{-1}(a_i) < \Gamma_+^{-1}(b_j)\} \wedge \{\Gamma_-^{-1}(a_i) < \Gamma_-^{-1}(b_j)\}$$

A placement satisfying this system of relations is depicted in Figure 5. This has apparent defects around the boundaries. This is due to the assumption of ABLR-relations imposed on devices in clusters. In fact there exists another placement with less area as shown in Figure 6. We observe that this has been attained by allowing jagged boundaries. We also observe that cluster A is still left-of cluster B in the figure, in the sense that A could be moved leftwards horizontally without interacting with B.

Given a relation that A is left-of B, we note there are three types of ABLR-relations between devices $a \in A$ and $b \in B$: "a is left-of b", "a is below b", and "a is above b". These are simply equivalent to "a is not right-of b". For example in Figure 6, it holds a_4 is left-of b_4, below b_5, and above b_3 as well.

Concluding the observation, given such an ABLR-relation between clusters A and B as

$$\{\Gamma_+^{-1}(A) < \Gamma_+^{-1}(B)\} \wedge \{\Gamma_-^{-1}(A) < \Gamma_-^{-1}(B)\},$$

then ABLR-relations between a_i $\{i = 1, 2, \cdots, 5\}$, and b_j $\{j = 1, 2, \cdots, 5\}$ are as follows.

$$\begin{aligned} &\overline{\{\Gamma_+^{-1}(b_j) < \Gamma_+^{-1}(a_i)\} \wedge \{\Gamma_-^{-1}(b_j) < \Gamma_-^{-1}(a_i)\}} \\ =&\overline{\{\Gamma_+^{-1}(b_j) < \Gamma_+^{-1}(a_i)\}} \vee \overline{\{\Gamma_-^{-1}(b_j) < \Gamma_-^{-1}(a_i)\}} \\ =&\{\Gamma_+^{-1}(a_i) < \Gamma_+^{-1}(b_j)\} \vee \{\Gamma_-^{-1}(a_i) < \Gamma_-^{-1}(b_j)\}. \end{aligned}$$

When we place all devices, we use these formulae as a constraint such that devices of a cluster are placed closely to each other. We call it 'jagged-boundary constraint'.

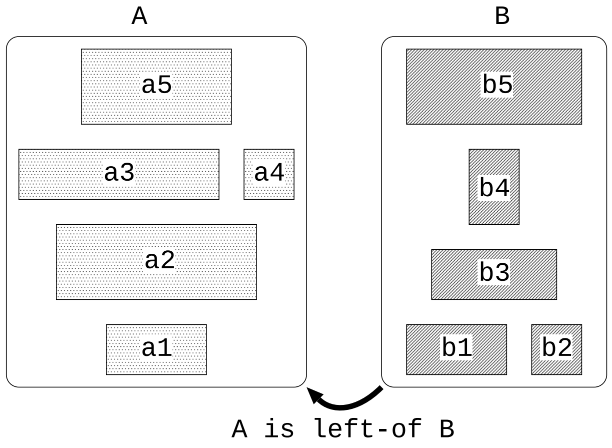

Fig. 4. ABLR-relation between A and B : (AB; AB)

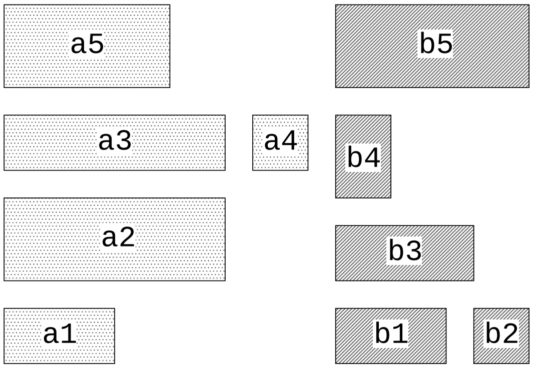

Fig. 5. Placement with straight boundary : $(a_5a_3a_4a_2a_1b_5b_4b_3b_1b_2; a_1a_2a_3a_4a_5b_1b_2b_3b_4b_5)$

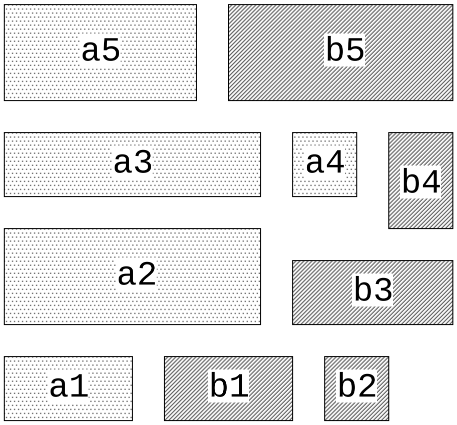

Fig. 6. Placement with jagged boundary : $(a_5b_5a_3a_4a_2b_4b_3a_1b_1b_2; a_1b_1b_2a_2b_3a_3a_4b_4a_5b_5)$

IV. Strategy for Optimization by Simulated Annealing

In rectangle-packing based placement, a simulated annealing (SA) has been widely and usefully employed. We also use SA to seek for an optimum placement at each of following three stages given in Section II.

Step 2: Place devices in each cluster.

Step 3: Place clusters in a cell.

Step 5: Place all devices in a cell under the jagged-boundary constraint.

Our target is multi-objectives to minimize (a) cell area, (b) total net length by the half perimeter of a bounding box, and (c) number of intersections between edges of MSTs. In manual design by experts, devices are so placed with few intersections between edges of MSTs. It usually results in high routability or few vias. This is why we adopt this objective.

To handle multi-objectives, we introduce a serial optimization by SA in four steps which we call the multi-step SA. At each step, an SA is invoked to optimize one objective, while other objectives are regarded as constraints. The algorithm is described as follows.

Multi-step optimization by SA

Step 1: Seek a placement with the minimum total net-length. Let the resultant total net-length be L_1.

Step 2: Seek a placement with the minimum area under the constraint that the total net-length is less than $\alpha_2 L_1$. Let the resultant total net-length and area be L_2 and A_2, respectively.

Step 3: Seek a placement with the minimum number of intersections of MSTs under the constraints that the total net-length and area are less than $\alpha_3 L_2$, $\beta_3 A_2$, respectively. Let the resultant area and the number of intersections of MSTs be A_3 and I_3, respectively.

Step 4: Seek a placement with the minimum total net-length under the constraints that the area and the number of intersections MSTs are less than $\beta_4 A_3$ and $\gamma_4 I_3$, respectively.

α_i, β_i, and γ_i are parametric constants tuned according to instances.

V. Experimental Results

We implemented a multi-level placement faithfully to our idea by C language. We tested the placement for six industrial instances of analog circuits on a computer on Celeron 1.0 GHz. We prepared a manual design by an expert for each instance. The number of devices and nets are shown in Table I. We compared each placement with the manual, evaluating them with respect to area, total net-length, and the number of intersections of MSTs.

The results by our placement and those by manual are shown in Table I. In the comparisons with respect to (a) area, (b) total net-length, and (c) number of intersections of MSTs, our placements are better than the manual ones on the average by (a) 10.8%, (b) 6.8%, and (c) 34.3%, respectively. The automated placement result and the manual placement of data4 are shown in Figure 8 and 9, respectively.

The computation time is also shown in the table. The significant relation between the computation time and the number of nets is shown in Figure 7.A reasonable approximation of the computation time is proportional to the square of the number of nets.

To demonstrate more the quality of our placement, we applied a commercial shape-based router REXSIR [12] to our result of data4. This automated result and the manual routed result are shown in Figure 10 and 11, respectively.

The manual layout time of data4 was 13 hours (7 hour placement, 6 hour routing). On the other hand, the time by the automation took only 79 seconds (77 sec. placement, 2 sec. routing). We achieved 1/730 layout time compared with the time by manual.

We note that the expert laid out taking other factors into consideration that some devices must be aligned to decrease mismatching, or a shield ring had better be inserted into the surrounding of capacitances. In the experiments, our placement did not pay so careful attention to such a performance of analog circuit. However, clusters defined by observing a schema are large enough to be imposed symmetric or matching constraints. An algorithm to handle symmetric constraints on the *SP* has been proposed [7] and we can adopt the algorithm into our placement.

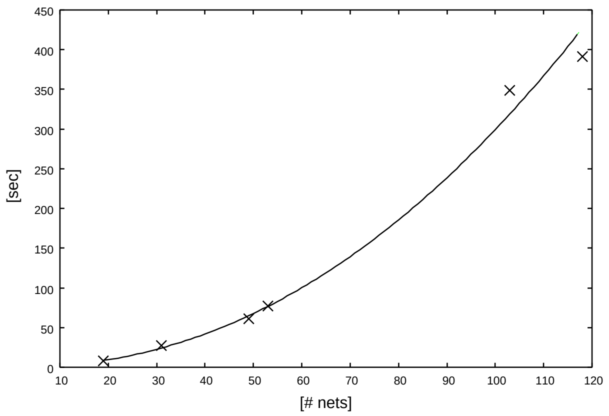

Fig. 7. Computation time v.s. number of nets

VI. Conclusions

A multi-level placement for an analog cell was proposed. It follows a guideline of how experts layout manually. The algorithm consists of steps (1) extract a set of clusters from a circuit schema, (2) place devices in each cluster, (3) place clusters in a cell, (4) extract jagged-boundary constraint, and (5) place all devices in a cell under the constraint. Our key idea is in relaxation of the shape of clusters from rectangles by allowing boundaries of clusters to be 'jagged'. These procedures are based on the Sequence-Pair. We evaluated the quality of a placement by a multi-objective. We adopted a multi-step simulated annealing to balance a trade-off between the objectives.

In experiments, we tested the placement for industrial examples. It attained placements better than those by manual on the average by 10.8%, 6.8%, and 34.3% with respect to area, total net-length and the number of intersections of MSTs, respectively. It also achieved 1/730 layout time compared with the time by manual layout.

As future works, we will develop an algorithm to involve other experts knowledge and techniques such as matching, shielding, and symmetricity.

Acknowledgments

We appreciate Mr. Masaki Mastui, Toppan Technical Design Center Co., Mr. Nobuto Ono, Mr. Yoshihiro Mochizuki, and Mr. Masakazu Endo, SII EDA Technologies Inc., for their valuable discussion to develop our algorithms. This work was supported in parts by funds from the Japanese Ministry of ECSST via Kitakyushu and Fukuoka knowledge-based cluster projects.

References

[1] J.Rijmenants, J.B.Litsios, T.R.Schwarz, and M.Degrauwe, "ILAC: An Automated Layout Tool for Analog CMOS Circuits", IEEE J. of Solid-State Circuits, Vol.SC-24, No.2, pp.417–pp.425, 1989.

[2] J.Cohn, D.Garrod, R.Rutenbar, and L.Carley, "Analog Device-Level Automation", Kluwer Academic Publishers, 1994.

[3] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "Rectangle-Packing-Based Module Placement", International Conference on Computer Aided Design (ICCAD) 1995, pp.472–479, 1995.

[4] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair", IEEE Trans. Computer-Aided Design, Vol.15, No.12, pp.1518–1524, 1996.

[5] S.Nakatake, K.Fujiyoshi, H.Murata, and Y.Kajitani, "Module Placement on BSG-Structure and IC Layout Applications", ICCAD 1996, pp.484–pp.491, 1996.

[6] E.Malavasi, E.Charbon, E.Felt, and A.Sangiovanni-Vincentelli, "Automation of IC Layout with Analog Constraints", IEEE Trans. on Computer-Aided Design, Vol.15, No.8, pp.923–942, 1996.

[7] F.Balasa and K.Lampaert, "Module Placement for Analog Layout Using the Sequence-Pair Representation", Design Automation Conference (DAC) 1999, pp.274–279, 1999.

[8] K.Lampaert, G.Gielen, and W.Sabsen "Analog Layout Generation for Performance and Manufacturability", Kluwer Academic Publishers, 1999.

[9] X.Tang, D.F.Wong, "FAST-SP : A Fast Algorithm for Block Placement based on Sequence-Pair", Asia and South Pacific Design Automation Conference (ASP-DAC) 2001, pp.521–526, 2001.

[10] X.Zhu, S.Nakatake, Y.Kajitani, and N.Ono "Floorplanning Consistent with Partial-Clustering on The Sequence-Pair", International Conference on Computer, Circuits and Systems (ICCCAS) 2002, pp.1386–1390, 2002.

[11] F.Balasa, C.S.Maruvada, and K.Krishnamoorthy, "Using Red-Black Interval Trees in Device-Level Analog Placement with Symmetry Constraints", ASPDAC 2003, pp.777–782, 2003.

[12] Seiko Instruments Inc. http://www.sii.co.jp

TABLE I
Industrial data and placement results

ratio : comparison ratio to manual results

name	#devices	#nets	manual			proposal			
			area	net length	#intersects of MSTs	area	net length	#intersects of MSTs	time [sec]
						ratio	ratio	ratio	
data1	27	19	66136	4099	37	55720	3542	11	8
						15 %	13 %	70 %	
data2	43	31	230953	9692	49	223600	7845	18	27
						3 %	19 %	63 %	
data3	64	49	108800	7241	104	89280	7443	67	61
						18 %	△ 3 %	36 %	
data4	83	53	422608	17767	165	354400	18170	119	77
						16 %	△ 2 %	27 %	
data5	151	103	91165	11147	352	90315	11306	352	349
						1 %	△ 1 %	0 %	
data6	160	118	396621	24296	304	348150	20636	272	391
						12 %	15 %	10 %	

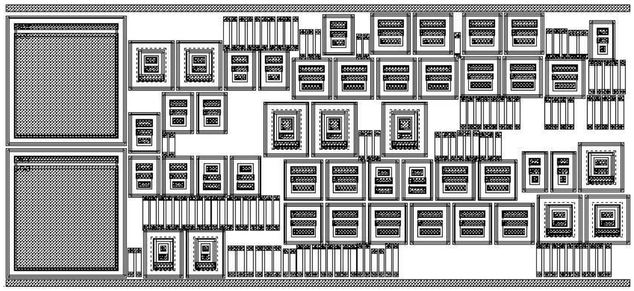

Fig. 8. Our placement of data4

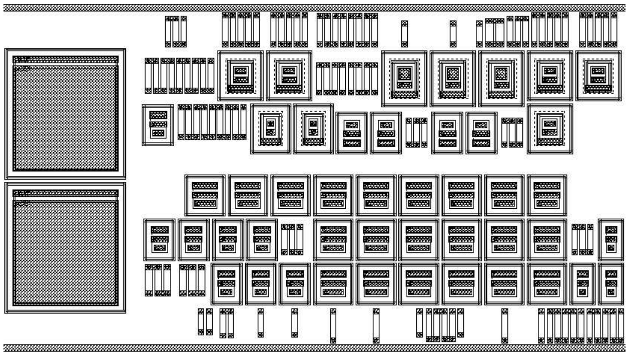

Fig. 9. Manual placement of data4

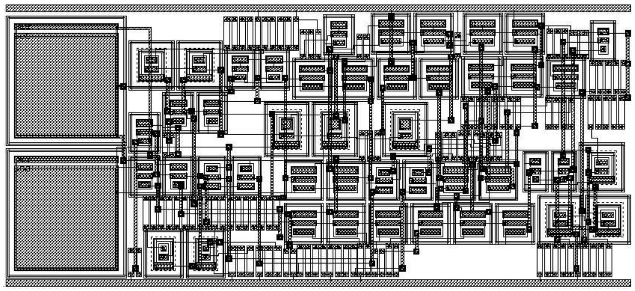

Fig. 10. Our placement and automated routing of data4

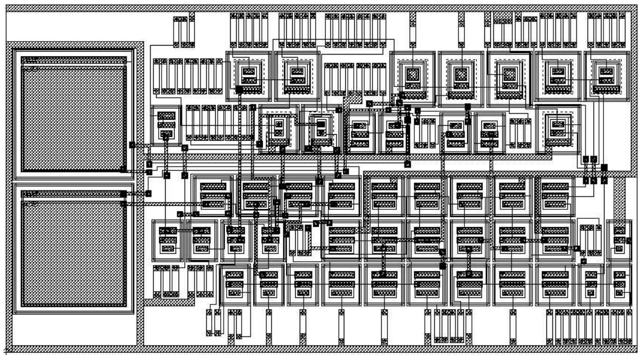

Fig. 11. Manual placement and routing of data4