Exact Wiring Fault Minimization via Comprehensive Layout Synthesis for CMOS Logic Cells

Tetsuya Iizuka[†], Makoto Ikeda[†‡], and Kunihiro Asada[†‡]

†Dept. of Electronic Engineering, University of Tokyo
‡VLSI Design and Education Center (VDEC), University of Tokyo
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{iizuka, ikeda, asada}@silicon.u-tokyo.ac.jp

Abstract

This paper proposes an exact cell layout synthesis technique to minimize the probability of wiring faults due to spot defects. We modeled the probability of faults on intra-cell routings with considering the spot defects size distribution and the end effect of critical areas. By using the model as a cost function, we comprehensively generate the minimum width layout of CMOS logic cells and select the optimum layouts. Experimental results show that our technique reduces about 15 % of the fault probabilities compared with the wire-length-minimum layouts for CMOS logic circuits which have up to 14 transistors.

1. Introduction

The recent improvement of VLSI process technologies enables us to integrate a large number of transistors on one chip, and significantly improve the circuit performance. On the other hand, the methodology of VLSI design becomes more and more complex and some new problems, such as Design For Manufacturability (DFM) have arisen. Due to the very high costs associated with the manufacturability of sub-micron integrated circuits, even a modest yield improvement can be extremely significant. In order to achieve the high yield, a standard-cell layout synthesis considering the DFM is required because standard-cells are the most basic elements of the cell-based design methodology. Although many papers have been published in standard-cell layout synthesis[1, 6], DFM was not taken into consideration.

In this paper, we propose a comprehensive cell layout synthesis technique to minimize the probability of wiring faults due to spot defects which is one of the main sources of electrical failure in VLSI integrated circuits. We modeled the probability of wiring faults on intra-cell routings with considering the spot defects size distribution and the end effect of critical areas. By using the model as a cost function, we comprehensively generate the minimum width layout of CMOS logic cells and select the optimum layouts.

2. Fault Model

Critical area is defined as the area in which the center of a spot defect must fall to cause the fault. Its correct estimation plays an important role in layout sensitivity to spot defects and yield prediction. The spot defect size distribution and the critical area are defined as $D(x)$ and $A(x)$ respectively, where x is the spot defect size. Assuming the area distribution of defect density is uniform, P_0, the fault probability P can be expressed as:

$$P = P_0 \int_{min}^{max} D(x)A(x)dx \tag{1}$$

where *min* and *max* are the minimum and the maximum defect size[2]. The spot defects size distribution function can be assumed to be

$$D(x) = \frac{{X_0}^2}{x^3} \tag{2}$$

where X_0 is the peak defect size of the distribution[5]. We consider only a bridging fault of two wire segments on the same layer due to a spot defect, which is called fault type OE (One-layer Extra-material defect). Since it is impossible to define the shape of a real spot defect, the shape of defect is assumed to be rectangular for simplicity in this paper. The critical areas of two wire segments whose width are w and spaced by d with the defect size x are illustrated in Fig. 1. The shaded area of Fig. 1 (a) illustrates the critical area between parallel wire segments. Assume that the length of overlapped section is l, the critical area is expressed as $l \times (x - d)$ when the end effects are neglected[2]. We take the end effect into account so that the model can be applicable to more complex intra-cell routings, and the back slashed areas are added to the critical area $L(x)$ which is newly expressed as

$$L(x) = (x + l) \times (x - d). \tag{3}$$

Fig. 1 (b) illustrates the critical area at the "L" type corner. Because the end section of two critical areas are overlapped here, the area of the redundant rectangle illustrated

0-7695-2093-6/04 $20.00 © 2004 IEEE

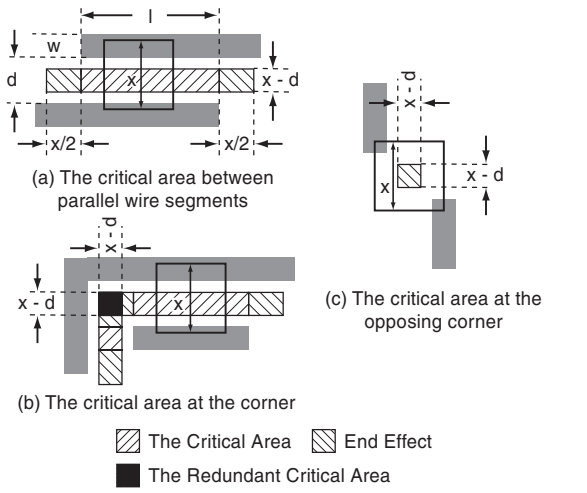

Figure 1. Critical areas of two wire segments spaced by d with the defect size is x.

as a black rectangle is subtracted once from the critical area. If the corner type is "T" or "+", the area of the rectangle is subtracted two or three times respectively. The area of the black rectangle $R(x)$ is expressed as

$$R(x) = (x - d)^2. \tag{4}$$

The critical area at the opposing corner is also illustrated in Fig. 1 (c). The area of this rectangle is also equal to $R(x)$.

The total critical area is easily calculated using these equations when the size of the spot defect is x. The probabilities of fault which results from each critical area is calculated using equation (1) and the defect size distribution function (2) as follows:

$$P_L = P_0 {X_0}^2 \int_{min}^{max} \frac{1}{x^3} L(x) dx \tag{5}$$

where P_L is the probability of fault which results from the critical area $L(x)$. Since P_0 and X_0 are process-dependent constant in this equation, these factors are excluded in the following discussion. The integral value without $P_0 {X_0}^2$ is defined as a "Sensitivity" to spot defect and written as S_L. If the defect size x is smaller than d, the defects will not cause any fault because of the zero critical area. If $d \le x < 2d+w$, the critical area $L(x)$ is expressed as equation (3). For $2d+w \le x$, the critical area is saturated to $(2d+w+l)\times(d+w)$. As a result, the sensitivity is expressed as

$$S_L = \int_d^{2d+w} \frac{(x+l)(x-d)}{x^3} dx + \int_{2d+w}^{max} \frac{(2d+w+l)(d+w)}{x^3} dx. \tag{6}$$

By setting max to ∞, we obtain

$$S_L = \ln \frac{2d+w}{d} + \frac{l-d}{2}\left(\frac{1}{d} - \frac{1}{2d+w}\right). \tag{7}$$

By calculating in the same manner, we also obtain the sensitivity S_R which results from the critical area $R(x)$:

$$S_R = \ln \frac{2d+w}{d} - \frac{d+w}{2d+w}. \tag{8}$$

Table 1. Our layout styles.

1.	Static dual CMOS logic circuits.
2.	Transistors are drawn up in two horizontal rows, the upper row for P-MOSFETs and the bottom row for N-MOSFETs.
3.	Two transistors that have the same GATE terminals are vertically aligned.
4.	All transistors are uniform-size.
5.	The intra-cell routing uses only first metal layers.
6.	VDD are connected from the top of P-diffusion to the top boundary by the vertical first metal.
7.	GND are connected from the bottom of N-diffusion to the bottom boundary by the vertical first metal.
8.	It is assumed enough to place single contact from metal to diffusion or polysilicon.

These sensitivities are calculated between two different wire segments only when these are placed on the neighboring grids. The final sensitivity is calculated by subtracting sensitivity portions which results from all redundant rectangles from sum of sensitivity portions which results from all $L(x)$ and opposing corners.

3. Layout Styles

Our layout styles are described in Table 1. The general styles for one dimensional width minimization of static dual CMOS logic cells were proposed by Uehara and vanCleemput[6]. Our layout style No. 1 through 3 are for the transistor placement based on Uehara's styles. However, our layout style No. 3 is different from Uehara's style. The complementary MOSFETs must be aligned vertically in Uehara's style whereas MOSFETs which have the same GATE terminals can be aligned vertically in ours. There are some cases that our layout style generates smaller width layouts. Although we assumed that all transistors are uniform-size for simplicity in this paper, it is easy to extend it to handle multiple sized transistors. The layout styles No. 5 through 8 in Table 1 are for intra-cell routing. We use only first metal layers for intra-cell routing because it is suitable for standard-cell layouts. Single contact hole is assumed enough to connect the metal and diffusion or polysilicon due to the commonly used silicidation.

4. Comprehensive Layout Synthesis

4.1. Transistor Placement

Our method utilizes the procedure proposed by [4] to generate the minimum width transistor placements. In [4], the minimum width placements are generated one by one until a routable placement is found via Boolean Satisfiability. We apply this method and generate all possible minimum width placement comprehensively under our layout styles.

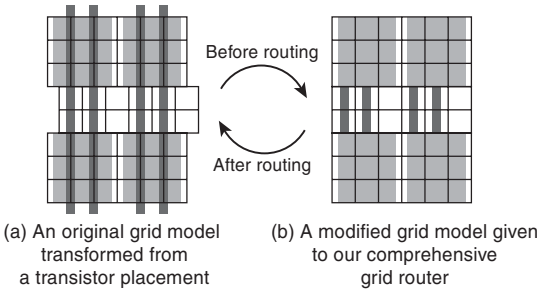

Figure 2. Grid models used in our cell layout synthesis system.

4.2. Intra-Cell Routing

A comprehensive intra-cell router has been proposed by [3], but this method assumes that the I/O placement is already fixed and the manhattan two layers model used by this method is not well suited for real standard-cell layouts. A router in the present study uses only first metal layers and considers the constraints characteristic of standard-cell layouts such as VDD/GND, silicides, and I/O port placements.

In our method, a generated placement is transformed into a grid model problem illustrated in Fig. 2 (a). When it is given to our router, the grids for GATE, between the P- and the N-diffusions, are moved half grid to the left, because our method routes the problems of complete grid model as illustrated in Fig. 2 (b). Our method routes a given grid model problem from left to right, column by column. Inside each column, all possible patterns are generated for each pattern of the previous column considering VDD/GND terminals, silicides, GATE, and I/O. VDD terminals must be connected from the top of P-diffusion to the top boundary of routing area, and GND terminals from the bottom of N-diffusion to the bottom boundary as described in No. 6 and 7 in Table 1. As described in No. 8 in Table 1, single contact hole is enough to connect the metal and the terminals on the diffusion or polysilicon because of the silicides. If a GATE terminal is an input or output port and is not connected to other terminals by metal layers, at least one grid of this terminal must be empty so that an I/O contact can be placed on it. It also considers the design rules so as not to generate the routing patterns which violate metal design rules when the grids are transformed back to the original grid position as shown in Fig. 2 (a). We use a constraint for search space reduction as described in [3], "A net can not fork when it extends to the next column". Fig. 3 describes the flow of our routing algorithm. The procedures described in boxes are implemented as functions. The details of each procedure is abbreviated in this paper.

5. Overall System

Using the procedures described in section 4, we can generate all possible layouts comprehensively. Then calculate the sensitivity using the model described in section 2 for each layout, so that we can find the DFM-optimized layouts. Overall flow of our exact DFM-optimized cell synthesis is described as follows.

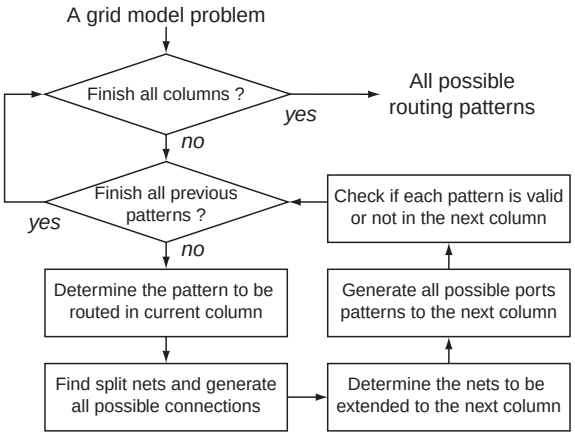

Figure 3. The flow diagram of our comprehensive router.

1. Given a transistor netlist
2. Generate a minimum width transistor placement using the procedure proposed by [4].
3. Comprehensively route a generated placement using the procedure explained in section 4.2. If this placement is routable, generate routed layouts. Otherwise no solution.
4. Find another minimum width placement using the procedure of [4]. If there is one, go to step 3 again. Otherwise go to step 5.
5. Place the I/O contact hole for each generated layout if needed. If several possible I/O placement patterns can exist for one layout, generate all possible I/O patterns.
6. Calculate the sensitivity for each layout and find the optimum solution.

6. Experimental Results

We applied our layout synthesis method to 8 CMOS logic circuits which have up to 14 transistors in a standard-cell library and Table 2 shows the results. The experiments were conducted on a 750 MHz UltraSPARC-III workstation with 2 GB of RAM. In this experiment, we assume 0.35 μm process technology and use 8 rows for routing grids; 3 rows for P- and N-diffusions respectively, and 2 rows for GATE as illustrated in Fig. 2. For each circuit the table indicates the number of transistors, the resultant number of columns of routing grids, the number of possible placement, the number of routable placement, the number of final layouts, the CPU times for layout synthesis and for selecting sensitivity-minimum layout, the minimum sensitivity

Table 2. The results of comprehensive cell layout synthesis.

Circuit name	#tr.	#col.	#place	#route	#layout	Cell synth. CPU(sec.)	Selection CPU(sec.)	Minimum sensitivity	Ave. sensitivity wire-length-min.	Reduction ratio(%)
ao222	14	9	32	8	948852	14685.04	323.70	18.04	20.55	12.19
aoi21	6	4	4	4	999	0.36	0.17	5.37	6.56	18.10
aoi211	8	5	4	4	8839	1.70	1.60	6.08	7.13	14.77
eno	10	6	2	1	19648	13.70	4.45	11.35	13.33	14.93
gen2	12	8	24	2	34120	419.98	10.38	17.78	20.90	14.91
mux2	12	9	144	4	392	6449.82	0.19	31.72	33.79	6.10
nand4	8	5	4	4	38366	3.01	6.92	5.37	6.38	15.78
xnor2	10	7	144	24	27414	2164.30	6.97	11.18	15.27	26.81

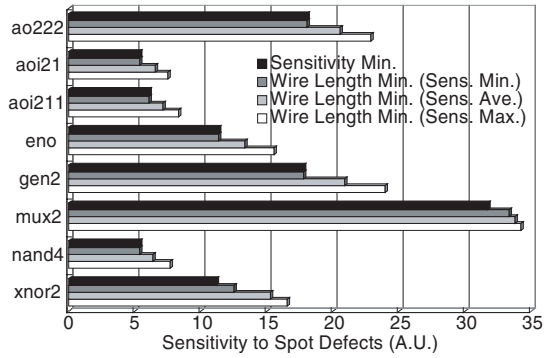

Figure 4. Changes in layout sensitivity to spot defects.

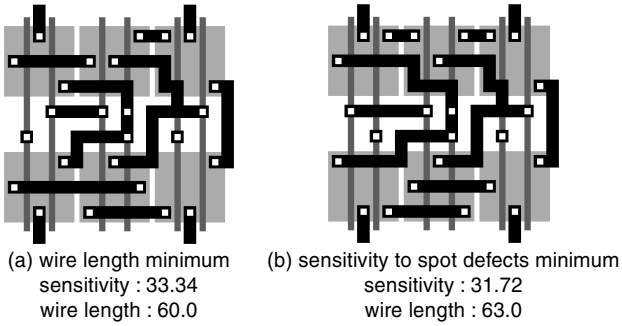

Figure 5. The optimum layouts of "mux2" generated by our method and selected by (a) wire length and (b) sensitivity to spot defects.

value and the average sensitivity value of all wire-length-minimum layouts, and the sensitivity reduction ratio. As shown in Table 2, our method generates the layout comprehensively in only a few seconds for circuits with up to 8 transistors, whereas the run time is severely high for the circuit with 14 transistors. In the case of standard-cell layout synthesis, we are allowed to consume relatively long time to obtain high quality layouts. For larger circuits, however, we need some other technique such as netlist partitioning to solve them in a reasonable time. The comparison results are illustrated in Fig. 4. The minimum sensitivity value and the minimum, average, and maximum sensitivity value of wire-length-minimum layouts are plotted in this graph for comparison. These results show that the wire-length-minimum layouts are not always the optimum layout for the sensitivity to spot defects. Compared with the average sensitivity value of all the wire-length-minimum layouts, our method can improve the sensitivity about 15 % on average for 8 circuits by picking up the optimum layout of sensitivity from all possible layouts. The snapshot of the mux2 layouts with wire-length-minimum and with sensitivity-minimum are illustrated in Fig. 5 for example. The sensitivity optimum layout has smaller sensitivity whereas its wire length are longer than the wire length optimum layout.

7. Conclusions

An exact cell layout synthesis technique to minimize the probability of wiring faults has been presented. The probability of wiring fault due to spot defects for intra-cell routings was modeled considering the spot defects size distribution and the end effect of critical areas, and used as a cost function. Our cell layout synthesis technique generates the minimum width layouts of CMOS logic cells comprehensively, and selects the optimum layouts based on the cost functions. We applied our comprehensive layout synthesis method to 8 CMOS logic circuits which have up to 14 transistors and the results show that the fault probabilities are reduced about 15 % compared with the wire-length-minimum layouts. Our layout synthesis method can be applicable for deriving the optimum cell layouts of some other cost metrics, such as power, delay, and signal integrity if reasonable cost functions are given.

References

[1] M. Guruswamy, et al., "CELLERITY: A Fully Automatic Layout Synthesis System for Standard Cell Libraries," *in Proc. ACM/IEEE 34th Design Automation Conference*, pp. 327–332, 1997.

[2] E. P. Huijbregtz, H. Xue, and J. A. G. Jess, "Routing for Reliable Manufacturing," *IEEE Trans. on Semiconductor Manufacturing*, vol. 8, No. 2, pp. 188–194, May 1995.

[3] T. Iizuka and K. Asada, "An Exact Algorithm for Practical Routing Problems," *in Proc. IEEE AP-ASIC*, pp. 343–346, 2002.

[4] T. Iizuka, M. Ikeda, and K. Asada, "High Speed Layout Synthesis for Minimum-Width CMOS Logic Cells via Boolean Satisfiability," *in Proc. IEEE ASP-DAC*, 2004 (to appear).

[5] C. H. Stapper, "Modeling of Defects in Integrated Circuit Photolithographic Patterns," *IBM J. Res. Develop.*, vol. 28, No. 4, pp. 461–475, 1984.

[6] T. Uehara and W. M. vanCleemput, "Optimal Layout of CMOS Functional Arrays," *IEEE Trans. on Computers*, vol. C-30, No. 5, pp. 305–312, May 1981.