Timing-Driven Cell Layout De-Compaction for Yield Optimization by Critical Area Minimization

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 a yield optimization method for standard-cells under timing constraints. Yield-aware logic synthesis and physical optimization require yield-enhanced standard cells and the proposed method automatically creates yield-enhanced cell layouts by de-compacting the original cell layout. However, the careless modification of the original layout may degrade its performances severely. Therefore, the proposed method de-compacts the original layout under given timing constraints using a Linear Programming (LP). We develop a new accurate linear delay model which approximates the difference from the original delay and use this model to formulate the timing constraints in the LP. Experimental results show that the proposed method can pick up the yield variants of a cell layout from the trade off curve of cell delay versus critical area and is used to create the yield-enhanced cell library which is essential to realize yield-aware VLSI design flows.

1. Introduction

The recent improvement of VLSI process technologies enables us to integrate a large number of transistors on one chip, and significantly improves the circuit performance. On the other hand, 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 deep sub-micron integrated circuits, even a small yield improvement can be extremely significant. Recently, a lot of papers related to VLSI yield improvement have been published[1, 2]. Most of them are a proactive methodology, which is not a post process. In [1], logic synthesis for manufacturability is proposed. This methodology introduces the manufacturability cost into logic synthesis and replaces the traditional area-driven technology mapping with a new manufacturability-driven one. It realizes larger reduction of the manufacturability cost when yield-optimized cells are available in the cell library. A new design flow proposed in [2] integrates manufacturability information into the timing-driven synthesis and place & route cost function. Yield-aware logic synthesis, place & route, and timing optimization are executed incrementally in this design flow. This flow uses a DFM extension library, which has variants of the basic logic functions with different manufacturability costs. They demonstrated the advantages of this methodology by applying it to several commercial ICs.

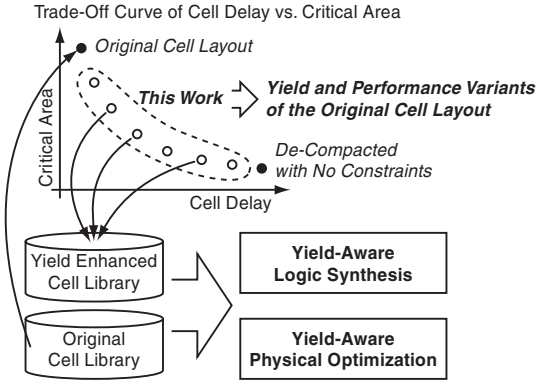


Figure 1. Overview of the proposed timing-driven cell layout de-compaction method.

As stated above, a yield-enhanced standard cell library is essential to these yield-aware VLSI design methodologies. Yield-enhanced standard cell libraries were, however, designed mainly by hand and there is no fully automated standard cell yield optimization method proposed for this purpose. This paper proposes an automatic yield-optimization technique for standard cells. Several papers have proposed the de-compaction method for yield-optimization[3, 4]. However, these methods consider only yield and area as costs, and the circuit performances are not considered. The careless modification of the original layout may degrade its performances severely and the created layout is not always acceptable for the target performances. Therefore, we propose a timing-driven cell layout de-compaction technique for yield optimization. The proposed method relaxes the width of a given cell layout under given timing constraints to optimize the yield. Since

3-9810801-0-6/DATE06 © 2006 EDAA

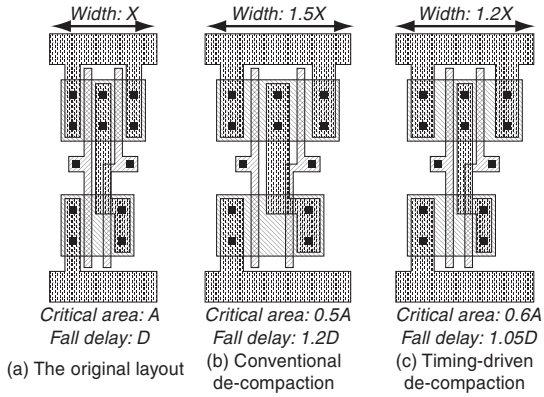


Figure 2. The conceptual layouts of 2-input NAND created by the conventional and the timing-driven de-compaction methods.

there is normally a high quality, hand-crafted original cell library, it is straightforward to optimize the yield by relaxing the width of the original layout with low computational effort, rather than creating it from scratch. Moreover, the de-compacted layout preserves the integrity and the predictability of the original layout because they also preserves the relative geometry as the original layouts. The proposed method optimizes the yield by minimizing the Critical Area (CA). CA is defined as the area in which the center of a spot defect must fall to cause a fault and its reduction plays an important role for yield enhancement. The overview of the proposed method is illustrated in Figure 1. This method creates a yield-optimized cell layout under given various timing constraints and can pick up the yield variants of a cell layout from the cell delay versus CA trade off curve. These cells are prepared as a yield-enhanced cell library and used for the yield-aware logic synthesis and physical optimization. The proposed method de-compacts the original cell layout using a Linear Programming (LP). The minimization cost of the LP is the total CA. We develop a new accurate linear delay model to formulate the timing constraints in the LP. This model approximates the delay difference from the original delay induced by the differences of the parasitic capacitances after de-compaction. Figure 2 shows the conceptual illustration of the cell layouts created by the conventional de-compaction method and the proposed timing-driven de-compaction method. The conventional timing-unaware method increases the width and space in the original layout for CA minimization, whereas the timing-driven one does not increase the width and space of the nets which have an effect on the target delay during CA minimization to meet the given timing constraint. Therefore, the proposed method creates the yield and performance variants of a cell layout depending on the given timing constraint.

2. Design Rule Constraints

The target of the proposed de-compaction method is standard cells and their height are fixed. Therefore, we

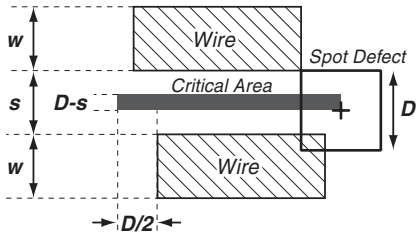


Figure 3. Schematic diagram of a short type critical area.

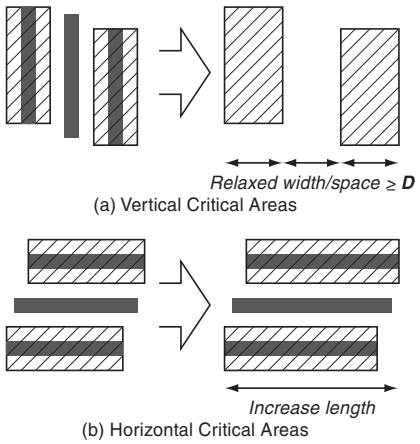


Figure 4. Variation of (a) vertical and (b) horizontal critical areas after horizontal de-compaction.

explain the de-compaction of only horizontal direction in this paper. Design rule constraints are formulated from a constraint graph constructed from a given layout, i.e., a set of polygons. Each constraint exists between the vertical edges of polygons. Each vertex of the graph corresponds to each edge of polygons and each edge of the graph has a weight value which corresponds to either the value of the minimum space or width. Once a constraint graph is constructed, it is straightforward to formulate the linear constraints. Of course, not only spacing and width design rules, but also other miscellaneous rules are formulated to create a de-compacted layout without design rule violation.

3. Critical Area Minimization

In this section, we will explain how to minimize the total CA. Figure 3 illustrates the schematic diagram of short type CA between two parallel wire segments whose width are w and spaced by s with the defect size D. In this paper, we assume that the shape of the spot defect is square for simplicity of the CA calculation. If the center of the defect falls inside the CA, these two wires are connected and cause a fault. Open type CA is also defined for each single wire segment in the same manner. The total CA is calculated if the coordinates of all edges are known. In our formulation, all these coordinates are given as variables and

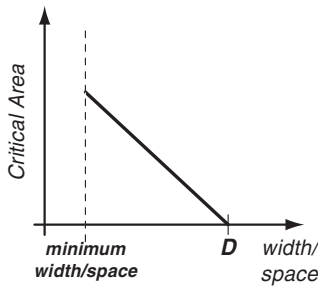


Figure 5. Change of the vertical critical area by the width or space.

the total CA should be minimized. Figure 4 shows variation of vertical and horizontal CA of both short and open type after horizontal de-compaction. The vertical CA are reduced by relaxing the width/space of polygons and finally become 0 when the width/space becomes the same value as the defect size D, whereas the horizontal CA are possibly increased since the lengths of horizontal wire segments are increased by horizontal de-compaction. The horizontal CA is easy to formulate as a linear function because it increases in proportion to the length. On the other hand, the calculation of the vertical CA is not so easy because it changes as shown in Figure 5. The area should be 0 if the width/space is larger than the defect size D. To realize this function, we use temporary variables r and l. These variables are defined as follows:

$$r \geq \frac{x_1 + x_2}{2}, \quad r \geq x_1 + \frac{D}{2} \tag{1}$$

$$l \leq \frac{x_1 + x_2}{2}, \quad l \leq x_2 - \frac{D}{2} \tag{2}$$

where x_1 and x_2 are the right and left edge of the polygons, respectively, as shown in Figure 6, and D is the defect size. Assume A is the vertical CA between these polygons, A can be written as $A \propto (r - l)$. To minimize the CA A, r should be minimized and l should be maximized. Under this condition, r and l are described as follows.

$$\begin{cases} r = x_1 + D/2, \ \ l = x_2 - D/2 & (x_2 - x_1 \leq D) \\ r = l = (x_1 + x_2)/2 & (x_2 - x_1 > D) \end{cases} \tag{3}$$

Figure 6 also illustrates these conditions. We can formulate the cost function as a sum of CAs, each of which is shown in Figure 5 using these variables.

4. Delay Model

We need a linear delay model to formulate the timing constraint as an LP. The well-known linear timing approximation is Elmore delay model. Figure 7 (a) illustrates a simple example of 2-input NAND. When the input signal A rises from logic level 0 to 1, then the output signal Y falls to logic level 0. In this situation, this transistor network is replaced by an RC network shown in Figure 7 (b) which

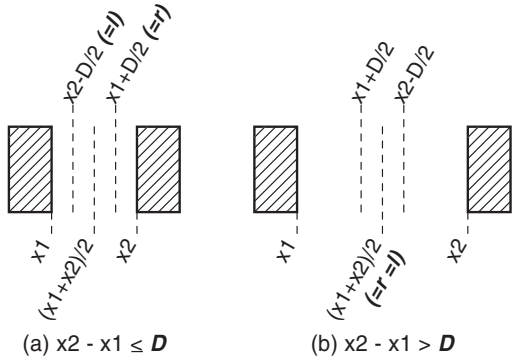


Figure 6. Calculation of the vertical critical area.

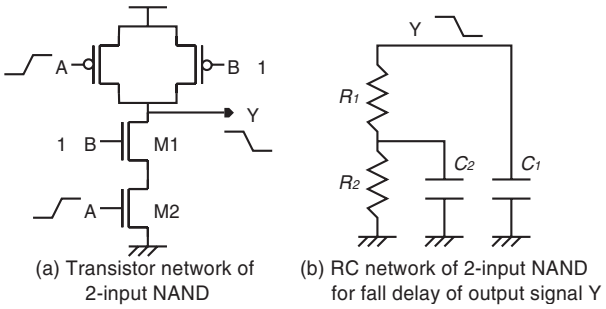
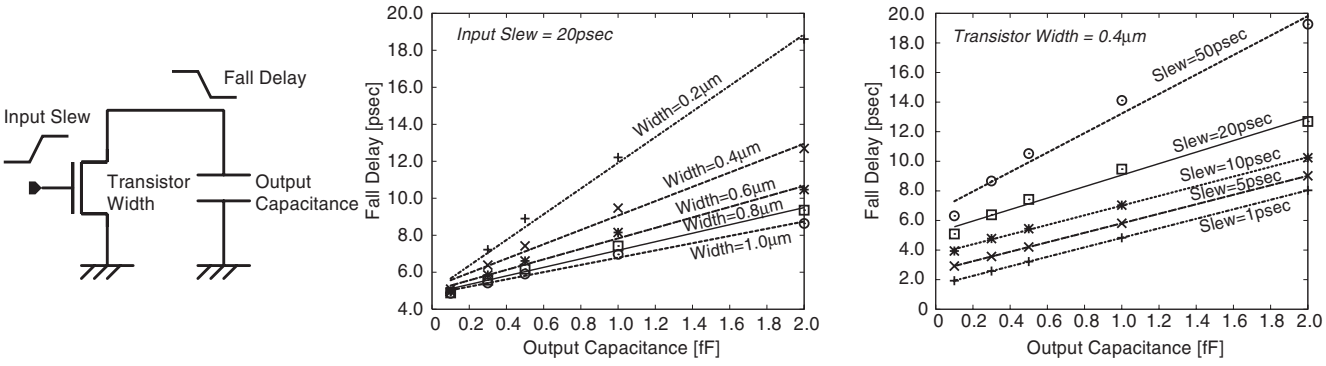


Figure 7. An example of 2-input NAND and its RC network for calculating Elmore delay.

consists of ON resistors R_1 and R_2 of the transistors M1 and M2, respectively, and parasitic capacitors C_1 and C_2. Using Elmore delay model, we can calculate the fall delay of the output signal Y as follows.

$$Delay_{A \to Y} = (R_1 + R_2) \times C_1 + R_2 \times C_2 \tag{4}$$

However, this model is not accurate enough to model the transistors of the recent deep sub-micron technologies. Therefore, we develop a new delay model which only calculates the delay difference induced by the difference of parasitic elements after de-compaction. Since the proposed method de-compacts a given original layout, we can extract the original parasitic elements and simulate the original delay values from this layout by Synopsys *HSPICE*. Once the original delay value is simulated, a delay difference by parasitic capacitances is approximated by a linear function as shown in Figure 8. This figure shows the graphs of fall delay of an N type transistor versus output capacitance. The schematic of the simulated circuit is shown in Figure 8 (a). Figure 8 (b) shows the delay variation by changing the width of the transistor and (c) shows the delay variation by changing the value of gate input slew. In both cases, the delay value is almost proportional to the capacitance, and the slope values are different from each other. Therefore, these slope values are calculated in advance and stored as a table of transistor width and input slew for P and N type transistors, respectively.



(a) Test Circuit (b) Delay variation by transistor width (c) Delay variation by input slew

Figure 8. Preliminary results of delay variation by changing the transistor width and the input slew.

Using these slope values, the procedure of delay increase calculation is written as follows.

1. Convert a given transistor network into an RC network and formulate the Elmore delay function in the same manner as shown in Figure 7.
2. Replace the values of ON resistors by the slope values that are determined by the values of the width and the input slew of each transistor.
3. Replace the values of parasitic capacitors by the difference of each parasitic capacitor after de-compaction.

For example of Figure 7, the fall delay increase $\Delta Delay_{A\to Y}$ is described as a linear function of parasitic capacitance differences as follows,

$$\Delta Delay_{A\to Y} = (k_1 + k_2) \times \Delta C_1 + k_2 \times \Delta C_2 \qquad (5)$$

where k_1 and k_2 are the slope values of the transistors M1 and M2, respectively, and ΔC_1 and ΔC_2 are the differences of the parasitic capacitors C_1 and C_2 after de-compaction, respectively. Using this model, we can describe the timing constraints by a linear inequation as

$$Delay^{initial}_{A\to Y} + \Delta Delay_{A\to Y} \le Delay^{target}_{A\to Y} \qquad (6)$$

where $Delay^{initial}_{A\to Y}$ and $Delay^{target}_{A\to Y}$ are the original and the target cell delay, respectively.

The difference of the parasitic capacitances also has to be linearly modeled by the coordinates of polygons in the original layout. The diffusion capacitance is calculated by a linear function of the area and the perimeter of the diffusion region. The parasitic capacitance of wires to ground or between overlapped layers is also calculated by a linear function of the area of the overlapped regions. For intra-layer cross coupling capacitances, it is easy to extract the capacitances between horizontal parallel wire segments because the de-compaction of the horizontal direction only increases the length of the parallel wires and this type of capacitance is mainly proportional to the length of the parallel wires. However, the coupling capacitances between vertical parallel wire segments are not so easy to calculate because the distance between these two wires are increased by the horizontal de-compaction and the capacitance value is not proportional but inversely proportional to the distance. Therefore, we approximate the value of this type of capacitance by linear function which is proportional to the distance with negative slope value. In the proposed method, the distance of these two wire segments increases to at most the same value of the defect size. In the following section, experimental results show that this approximation is accurate enough to calculate the timing constraints within this range. A capacitor between two signals is approximated by two capacitors from each signal to ground. Both of them have the same capacitance as the original capacitor. At this stage, we can describe the timing constraints as linear functions of the coordinates of the polygons in the layout and can formulate them into the LP problem.

This model describes the delay value of single-stage transistor networks. We are developing an extension of this model to approximate the output slew and this enables us to apply the proposed method to multi-stage transistor networks.

5. Overall Flow

Figure 9 shows the overall flow diagram of the proposed method. The input to the proposed method is the original cell information and design constraints. The original cell information includes the original cell layout for polygon information and the netlist of the cell for transistor connection information. Design constraints include the target cell delay for each timing arc[1] and the maximum cell width. The maximum width constraint was not explained in the previous sections, but it can be formulated as a part of the design rule constraints. Using this information, the linear constraints generator formulates the constraints explained in the previous sections and then the LP solver searches for the solution of the generated LP problem and creates the de-compacted layout from the solution.

[1] A timing arc is defined as a signal flow from an input to an output on a cell, e.g., A rise → Y fall.

Table 1. The benchmark circuits used in this experiment.

Circuit	*Explanation*	*#trans.*	$Delay_{orig}$ *[psec]*	$Area_{orig}$ $[\mu m^2]$	CA_{orig} $[\mu m^2]$
NAND3_1	3-input NAND	6	33.05	3.70	0.97
NAND3_2	3-input NAND (buffered)	12	34.56	6.53	1.97
NAND4_3	4-input NAND (buffered)	36	53.11	22.15	10.97
NOR4_1	4-input NOR	8	73.65	4.76	1.30
NOR4_2	4-input NOR (buffered)	28	65.95	17.41	8.47
ON2222_3	Series-parallel circuit for $\overline{(A0 \vee A1) \wedge (B0 \vee B1) \wedge (C0 \vee C1) \wedge (D0 \vee D1)}$	56	64.83	28.75	9.92

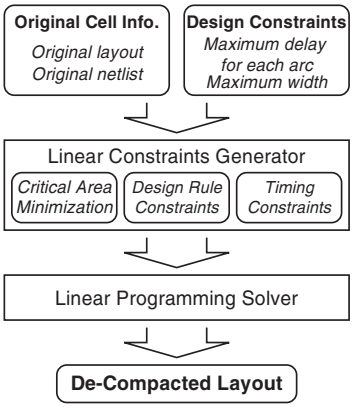


Figure 9. The overall flow diagram of the proposed method.

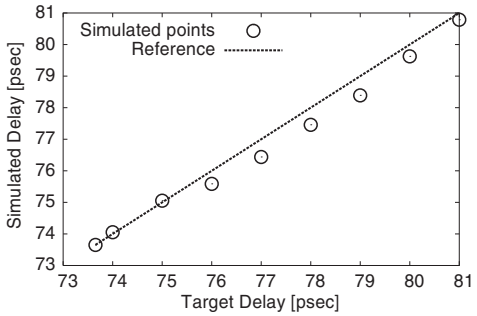


Figure 10. Accuracy of the proposed delay model in the case of NOR4_1.

6. Experimental Results

The proposed timing-driven de-compaction method was implemented to show its effectiveness. In this experiment, we used ILOG *CPLEX 9.1* for an LP solver and 6 single-stage cells from a standard-cell library of 90 nm technology were used as benchmarks. Table 1 summarizes the characteristics of these cells. This table shows the circuit name, the explanation of each circuit, the number of transistors, the original cell delay value, the original cell area, and the original critical area. $Delay_{orig}$ column shows the original delay of a timing arc. The delay value of these arcs were constrained in this experiment. To calculate the original cell delay, a netlist with parasitic capacitances is extracted using Mentor Graphics *Calibre xL* and simulated using Synopsys *HSPICE*. The tables of the delay slope value(Figure 8) for P and N type transistors are also calculated using *HSPICE* in advance. In this experiment, we did not connect an additional capacitor to the output net to clarify the effect of intra-cell parasitic elements. The defect size used in this experiment is 1.5 times larger than the minimum width/space of the first metal layer and the CAs are calculated only for the first metal layers.

Table 2 shows the results of the proposed timing-driven de-compaction method. This table shows the target and the actual delay value, the cell area, and the CA of the generated cell layouts. "No constraint" in the column of Target Delay means that no timing constraints were set in this case. Because the vertical CA decreases but the horizontal CA possibly increases by horizontal de-compaction, there must be an optimal value of the total CA. The value of CA in the no constraint case is the minimum CA value for each cell. The runtime to create the de-compacted layout is about 0.1 second even for the largest example of ON2222_3 which consists of 56 transistors. The runtime for creating the tables of slope values and the first *HSPICE* simulation for each cell is excluded because they are conducted just once in advance. The delay values of the generated layouts are also simulated by *HSPICE* using a netlist extracted by *Calibre xL*. The errors of the target and actual delay values are less than 1 % for most cases and the average error is 0.63 %. Figure 10 plots the target and the simulated delay values in the case of NOR4_1 when the input signal to the P type transistor connected to VDD falls from logic level 1 to 0. The simulated delay values show good accordance with the target delay values. These results show that the developed delay model is accurate enough for the proposed de-compaction method.

After de-compaction, the cell areas increase about 10 to 50 % and the CAs decrease about 10 to 50 % depending on the cells and the constraints. The average values of the cell area increase and the CA reduction without timing constraints are about 26 % and 24 %, respectively. As the timing constraint becomes tight, the cell area increase and CA reduction become small. In the cases of NAND3_1, NAND3_2, and NOR4_1, the proposed method creates the cells with smaller delay than the no constraint case while their area and CA is equal to that of the no constraint case. Figure 11 shows the trade-off curves of target delay versus CA and cell area versus CA in the case of NOR4_1. The conventional simple de-compaction method can not create these various yield-optimized cell layouts. On the other hand, the proposed method can pick up the yield and performance variants of a cell layout from these curves and these cells are prepared as a yield-enhanced library which is essential to realize yield-aware VLSI design flows.

Table 2. Results of the proposed timing-driven de-compaction method. The runtime of de-compaction for each cell is less than 0.1 second for all cases.

Circuit	*Target Delay [psec]*	*Actual Delay [psec]*	*error [%]*	*Area [μm^2]*	*increase [%]*	*CA [μm^2]*	*reduction [%]*
NAND3_1	35	34.98	0.06	5.70	54.05	0.47	51.55
	37	36.77	0.63	5.70	54.05	0.47	51.55
	No constraint	37.92	—	5.70	54.05	0.47	51.55
NAND3_2	35	34.95	0.14	8.11	24.20	1.56	20.81
	36	35.76	0.67	8.49	30.02	1.47	25.38
	No constraint	36.54	—	8.49	30.02	1.47	25.38
NAND4_3	54	53.81	0.35	23.87	7.77	10.29	6.20
	55	54.72	0.51	24.81	12.01	10.10	7.93
	No constraint	55.83	—	25.58	15.49	10.00	8.84
NOR4_1	76	75.59	0.54	5.71	19.96	0.92	29.23
	80	79.62	0.48	6.02	26.47	0.85	34.61
	No constraint	81.33	—	6.02	26.47	0.85	34.61
NOR4_2	67	67.47	-0.70	19.33	11.03	7.59	10.39
	70	68.87	1.64	20.25	16.31	7.30	13.81
	No constraint	70.06	—	20.76	19.24	7.27	14.17
ON2222_3	66	65.63	0.56	31.20	8.52	9.11	8.17
	67	66.18	1.25	31.29	8.83	8.94	9.88
	No constraint	67.65	—	31.68	10.19	8.86	10.69
average	—	—	0.63	—	25.91*	—	24.20*

$*$: the average of the no constraint cases

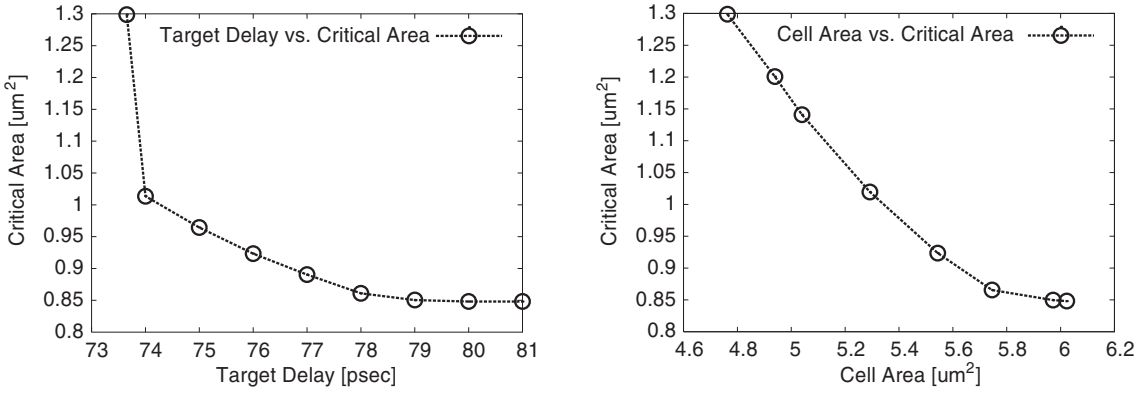


Figure 11. Trade off curves of delay versus CA and cell area versus CA in the case of NOR4_1.

7. Conclusions

This paper proposed a yield optimization method for standard-cells by CA minimization under timing constraints. The proposed method de-compacts the original layout under given timing constraints using the LP. We developed a new linear delay model which approximates the difference from the original cell delay and used this model to formulate the timing constraints as the LP. Experimental results showed that the developed delay model is accurate enough to constrain the delay during de-compaction. The maximum CA reduction was about 25 % on average of 6 cells. The proposed method can pick up the yield and performance variants of a cell layout from the cell delay versus CA trade off curve and can provide a yield-enhanced library. The proposed method is the essential technique to realize the yield-aware VLSI design methodologies.

As ongoing works, we are developing an extension of the timing model to apply the proposed method to multi-stage transistor networks and taking other manufacturability metrics, such as layout regularity[5], into consideration.

Acknowledgment

This work is supported by the Grant-in-Aid for Scientific Research of the Japan Society for the Promotion of Science (JSPS), and VLSI Design and Education Center (VDEC), University of Tokyo.

References

[1] A. Nardi and A. L. Sangiovanni-Vincentelli, "Synthesis for Manufacturability: a Sanity Check," in *Proc. IEEE/ACM Design, Automation and Test in Europe*, pp. 796–801, 2004.

[2] C. Guardiani, N. Dragone, and P. McNamara, "Proactive Design For Manufacturability (DFM) for Nanometer SoC Designs," in *Proc. IEEE Custom Integrated Circuits Conf.*, pp. 15.1.1–15.1.8, 2004.

[3] C. Bamji and E. Malavasi, "Enhanced Network Flow Algorithm for Yield Optimization," in *Proc. ACM/IEEE 33rd Design Automation Conference*, pp. 746–751, 1996.

[4] Y. Bourai and C.-J. R. Shi, "Layout Compaction for Yield Optimization via Critical Area Minimization," in *Proc. IEEE/ACM Design, Automation and Test in Europe*, pp. 122–125, 2000.

[5] X. Yuan, K. W. McCullen, F.-L. Heng, R. F. Walker, J. Hibbeler, R. J. Allen, and R. R. Narayan, "Technology Migration Technique for Designs with Strong RET-Driven Layout Restrictions," in *Proc. ACM Int. Symp. on Physical Design*, pp. 175–182, 2005.