Open critical area model and extraction algorithm based on the net flow-axis*

Wang Le(王 乐)[a)b)†], Wang Jun-Ping(王俊平)[a)], Gao Yan-Hong(高艳红)[a)], Xu Dan(许 丹)[a)], Li Bo-Bo(李玻玻)[a)], and Liu Shi-Gang(刘士钢)[b)]

[a)] *School of Communication Engineering, Xidian University, Xi'an* 710071, *China*
[b)] *Microelectronics Institute, Xidian University, Xi'an* 710071, *China*

(Received 24 January 2013; revised manuscript received 17 July 2013)

In the integrated circuit manufacturing process, the critical area extraction is a bottleneck to the layout optimization and the integrated circuit yield estimation. In this paper, we study the problem that the missing material defects may result in the open circuit fault. Combining the mathematical morphology theory, we present a new computation model and a novel extraction algorithm for the open critical area based on the net flow-axis. Firstly, we find the net flow-axis for different nets. Then, the net flow-edges based on the net flow-axis are obtained. Finally, we can extract the open critical area by the mathematical morphology. Compared with the existing methods, the nets need not to divide into the horizontal nets and the vertical nets, and the experimental results show that our model and algorithm can accurately extract the size of the open critical area and obtain the location information of the open circuit critical area.

Keywords: critical area, mathematical morphology, layout optimization, yield

PACS: 85.40.Bh, 61.72.–y **DOI:** 10.1088/1674-1056/22/12/128504

1. Introduction

With the constant development of the integrated circuit (IC) technology, the IC feature size is decreasing, and therefore the IC yield loss due to manufacturing defects[1–5] becomes more and more serious, resulting in failing to obtain the expected yield by designing integrated circuits just based on the IC design rules.[6–9] The very large scale integrated circuit (VLSI) is sensitive to defects in the IC manufacturing process, and the rate of the IC yield loss which is caused by defects in the manufacturing process is more than 80%.[10] The important approach to raising IC yield related to random defects is to optimize the IC layout in the IC design stage, especially in the layout design stage, which depends on the information about the random defects causing yield loss.[11–14] The existing experimental results show that the real defects are of irregular shape,[15–18] and that these defects have a certain space distribution.[19–21] In order to describe the yield loss in the IC manufacturing stage, a concept of the critical area in an IC layout and its extraction method are presented.[22–26] The critical area plays an important role in bridging between the IC design stage and the IC manufacturing stage, and its reduction can effectively improve the yield. The main method of reducing the critical area is to extract the critical area and then to adjust the parameters of nets such as the net routing path in the IC layout. Therefore how to accurately extract the critical area and to find out the nets which will be adjusted and its position information in the IC layout becomes a hot topic of improving layout optimization accuracy and efficiency.

The critical area plays a crucial role in IC yield estimation and reliability prediction. The main methods can be roughly grouped into the three categories. The first kind is based on Monte Carlo calculations,[27] with its advantage of easy yield estimation and its disadvantage of low speed. The second one is the geometric method by means of polygon operator and the method based on Voronoi diagram,[28–31] the former is better in calculation efficiency, and the latter is better in calculation accuracy. It is a pity that defects are assumed to be approximately square.[32] The last one is the model method based on the regular defect shape and those regular nets,[33] which is high in speed but low in accuracy by considering defects and nets as a regular shape during extraction of the critical area.[34] In Ref. [35], a method of extracting critical areas based on the mathematical morphology is presented, with its advantage that this method takes the irregular nets and the irregular defect into consideration based on the image processing technology, and its disadvantage that this method reduces the critical area extraction precision because the nets in the IC layout are divided into the vertical nets and the horizontal nets. This paper is the extension of Ref. [35], and it does not need to divide the nets into the vertical nets and the horizontal nets. Therefore this algorithm improves further the extraction precision of the critical areas.

*Project supported by the National Natural Science Foundation of China (Grant Nos. 61173088 and 61070143) and the 111 Project (Grant No. B08038).
†Corresponding author. E-mail: xianwangle@163.com
© 2013 Chinese Physical Society and IOP Publishing Ltd

2. Open critical area model based on the net flow-axis

The open critical area is the region in which the center of a real defect must fall in order to cause the conductor N to be completely broken in the IC layout, which is equal to the region in which the center of a real defect must fall in order to cause the open circuit fault in the IC manufacturing process.[35]

Definition 1 (net flow-axis) In the IC layout, net flow-axis is defined as the axis that represents the routing path of the net, which can describe the topological structure and the geometric characteristics of the layout net (See Fig. 1 for the illustration of the net flow-axis).

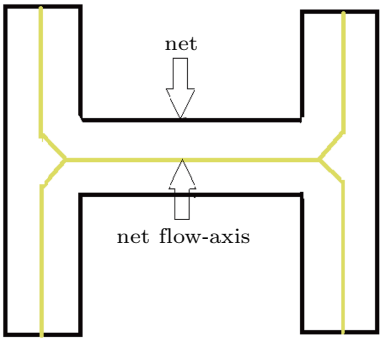


Fig. 1. (color online) Net flow-axis.

Definition 2 (net flow-edge) In the IC layout, net flow-edge is defined as the net contour boundary located on both sides of the net flow-axis, which can describe the contour of the net and the width of the path in the net routing. (See Fig. 2, where the net flow-edge is illustrated with NE_1, NE_2, NE_3, and NE_4.)

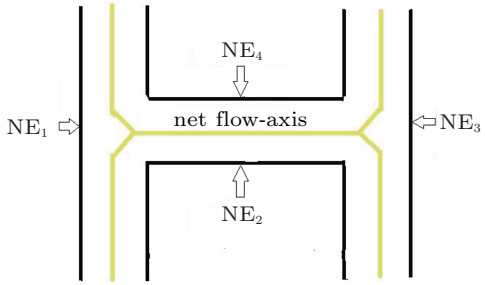


Fig. 2. (color online) Net flow-edge.

The open critical area model for a single net based on the net flow-axis: according to the dilation operation in mathematical morphology and the definition of the open critical area, we choose the real defect $d(X_c,Y_c)$ discretionarily as the structuring element, in which both the core of the defect and the origin of the structuring element are (X_c,Y_c), the net flow-edge is NE_k, $k=1,2,3,\ldots,n$. Dilate the net flow-edge NE_k respectively by the defect $d(X_c,Y_c)$, then we can find the intersections of all these dilation regions, and the intersection region is the open critical area formed of the real defect $d(X_c,Y_c)$ and the net N, named as $A_o(X_c,Y_c,N)$. The region of dilating the net flow-edge NE_k by the real defect $d(X_c,Y_c)$ can be written as

$$D_k = \mathrm{DILATE}(\mathrm{NE}_k,X_c,Y_c),\quad k=1,2,\ldots,n. \tag{1}$$

Thus, the open critical area $A_o(X_c,Y_c,N)$ of the single net N resulting from the set of defect centers is expressed as

$$A_o(X_c,Y_c,N)=\mathrm{Area}(P),\quad P=\left\{p|p\in\sum_{i=1}^{n}D_k,p\geq 2\right\}. \tag{2}$$

In Eqs. (1) and (2), $\mathrm{DILATE}(\mathrm{NE}_k,X_c,Y_c)$ is the dilation of the defect $d(X_c,Y_c)$ to NE_k, p is the pixel value, P is the critical area, and $\mathrm{Area}(P)$ is the open critical area. If the open critical area $A_o(X_c,Y_c,N)$ is 0, the corresponding defect which is the minimum size will not cause the open circuit fault in the net. If the open critical area $A_o(X_c,Y_c,N)$ takes a smaller value, the open critical area related to the smaller defect is the discontinuous region in the layout. While the defect is a big enough one, the open critical area can obtain a maximum value, and it is equal to the area of the layout.

3. Open critical area extraction algorithm based on the net flow-axis

In this section, according to the given open critical area model, we first present the detailed procedure of the core algorithm for extracting the open critical areas based on the net flow-axis. Then we explain the design philosophy of this algorithm according to each step. Finally, we analyze the performance of this algorithm.

For a particular level of the layout I, the defect is $d(X_c,Y_c)$, a set which consists of the defects is M, the flow-axis in the net N is L, the flow-edge in the net N is NE_j, where $j=1,2,3,\ldots,N_{\mathrm{num}}$, i is the sequence number of a net, and m is the total number of nets. A large layout is separated into some small layouts in advance. Combining the data information of the known defect and the open critical area model of the single net in Eqs. (1) and (2), the detailed procedures of the core algorithm for each layout to extract the open critical area related to a single defect are given as follows.

Algorithm 1 Algorithm to extract the open critical area with a single defect

Step 1 Input a binary layout I, save the layout I in a corresponding matrix $L_{N_1\times N_2}$.

Step 2 Separate each net in the layout I based on the connectivity, mark the single net $Q(i)$, $i=1,2,3,\ldots,m$, m is the number of the nets, and the initial value of i is 1.

Step 3 Choose the single candidate net $N=Q(i)$.

Step 4 Extract the flow-axis of the net N, and record it as L.

Step 5 Extract the flow-edge of the net N, and record it as NE_j, $j = 1,2,3,\ldots$, N_{num}, N_{num} is the number of the net flow-edge, and the initial value of j is 1.

Step 6 Choose the candidate net flow-edge NE_j.

Step 7 According to Eq. (1), dilate the net flow-edge NE_j by the defect $d(X_c, Y_c)$, and obtain D_j.

Step 8 Let $j = j+1$. If $j \le N_{\mathrm{num}}$, then go back to Step 6.

Step 9 According to the open critical area model, calculate the superposition result $\sum_{j=1}^{N_{\mathrm{num}}} D_j$ of all dilations of the net flow-edges.

Step 10 According to Eq. (2), extract the open circuit critical region P, calculate and store the corresponding open critical area $A_o(X_c, Y_c, N)$.

Step 11 Let $i = i+1$. If $i \le m$, then go back to Step 3.

Step 12 Store and output the open critical areas in the layout I.

The design philosophy of this algorithm above is given as follows. The number of the nets can be obtained by inputting the corresponding matrix of the layout in Steps 1 and 2 above. As shown in Fig. 3, the total number of the nets is 13, i.e., $m = 13$. The number in Fig. 3 is the sequence number of the corresponding net. In order to reduce the execution time, Steps 3–5 extract the net flow-axis and the net flow-edges. Steps 6–8 calculate the dilation of each flow-edge through the cycle operation. In Step 9, we obtain the superposition result of all dilations of the net flow-edges. Due to the fact that the layout is a binary layout, the region where the pixel value of the superposition result is bigger than 1 is the open critical area. Steps 10–12 calculate and output the open critical area for each net.

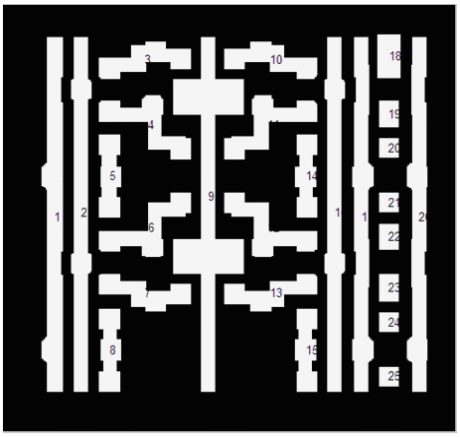


Fig. 3. $L1$ in MUSB.

The following is the analysis of the time complexity of this algorithm. The execution time of the algorithm mainly depends on the double-cycle from Steps 3–11. Hence, the time complexity of the algorithm is $O(mn)$, where m is the number of the nets in the layout I and n is the average value of the numbers of net flow-edges. So we can find that the execution time of this algorithm increases with the increasing of the number of the nets when the shape of the defect remains constant. As a result, we can reduce the execution time by adopting the divide-and-conquer strategy when dealing with the large scale layout. The core idea of the divide-and-conquer strategy is to divide the large scale layout into smaller blocks and then to perform corresponding processing. To validate that the execution time is acceptable, we conduct the experiments on extracting the open critical area by our method and divide-and-conquer strategy for interface chip MUSB-$L61$ layer. Figures 3 and 4 are two layouts $L1$ and $L2$ randomly selected in MUSB-$L61$ layer, and the execution results as shown in Table 1. Due to limited space of this paper, we only give the results of nets 1–17. This method is implemented on a dual core 2.33-GHz PC machine with 4-GB memory. The unit of critical area is μ^2, μ is the unit of grid measure in layout, the defect is $20\mu \times 26\mu$. In the following tables, "Net No." shows the net number in the corresponding layout. "OCA $L1$" is the open critical area (OCA) which is extracted by our method for $L1$, "OCA $L2$" is the open critical area which is extracted for $L2$. Besides, the "Time1 (s)" and "Time2 (s)" are the corresponding execution times in our algorithm, and they are in units of s. In Table 1, the maximum and the minimum execution times of $L1$ are 2.53 s and 0.98 s, and the maximum and the minimum execution times of $L2$ are 2.19 s and 0.53 s. The average execution times are 1.15 s and 0.78 s. Total time of $L1$ is 19.59 s, and total time of $L2$ is 13.33. In Ref. [35], this method needs to divide nets into the horizontal nets and the vertical nets and chooses interactively the size of the structural element. The completion of this algorithm is by a semi-automatic method. To our automatic algorithm, we can easily find that the execution time of our algorithm is at an acceptable level.

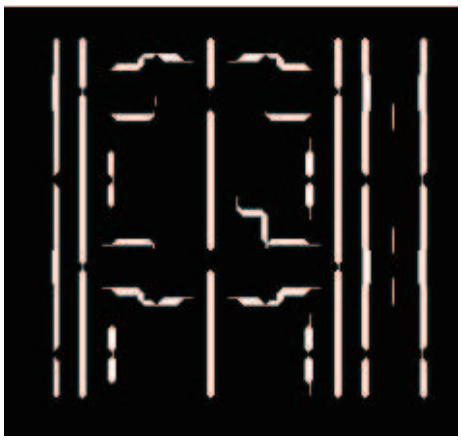


Fig. 4. OCA of $L1$ in our method.

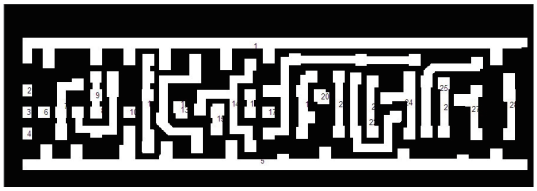


Fig. 5. $L2$ in MUSB.

Table 1. Execution results of $L1$ and $L2$.

Net No.	OCA $L1$	Time1/s	OCA $L2$	Time2/s
1	266.0	1.24	8810.3	2.19
2	1749.0	0.97	260.0	0.53
3	250.1	0.99	275.0	0.56
4	189.0	0.98	260.0	0.54
5	0.0	1.03	7957.5	2.07
6	63.0	1.06	182.0	0.60
7	250.1	1.14	1359.5	0.72
8	261.0	0.99	2319.9	0.57
9	1068.0	2.53	1018.6	0.58
10	294.1	1.03	182.0	0.58
11	195.0	1.03	1947.5	0.59
12	200.1	1.42	2975.6	0.59
13	294.1	1.02	270.3	0.60
14	261.0	1.02	3443.3	0.81
15	261.0	1.07	429.4	0.59
16	1749.0	1.03	637.5	0.60
17	1302.0	1.04	189.0	0.61

In addition, the execution time can decrease further in this paper. Firstly, the characteristic of the mathematical morphology operation is that it is easy to realize the hardware parallel implementation, so the overall execution time can be shortened by N times, where N is the number of the nets in a block. For the $L1$ and $L2$, the net maximum execution time is the whole layout execution time, so the total execution time decreases to 2.58 s and 2.19 s. Secondly, the extraction of the open critical area in allusion to the different defects in shape can be distributed, so the whole execution time of the algorithm is the maximum of the open critical area extraction time for each defect. Thus in the permission of hardware investment, this algorithm has a good application prospect as the execution time can be reduced greatly.

Compared with the existing algorithms, this algorithm has the advantages which are listed in the following 4 aspects. Firstly, the random defects are used in this algorithm, so the characterizations of the defects can better satisfy their natural forms. Secondly, it uses the theory of mathematical morphology to extract the open critical area, which makes the parallel implementation of the algorithm possible. Thirdly, it does not need to divide nets into the horizontal nets and the vertical nets so that the precision and execution speed of the open critical area are improved. There is an SRAM layout in Fig. 6, and the sequence numbers of the nets are 1–13. As shown in Fig. 7, the open critical area is extracted by the method given in Ref. [35]. As shown in Fig. 8, the open critical area is extracted by our presented algorithm. Comparing Fig. 7 with Fig. 8, we can find that according to the definition of open circuit critical area, some errors can be seen which are marked with some ellipses in Fig. 7, and the areas corresponding to these errors have been modified in Fig. 8. Table 2 further presents the performance comparison between our algorithm and the existing methods for $L1$ and $L2$. In Table 2, "OCA $L1$" and "OCA $L2$" are the execution results in $L1$ and $L2$ by our algorithm, "OCA $L1$o" and "OCA $L2$o" are the execution results obtained by the method in Ref. [35]. In $L1$, we can see that the OCAs (net 2, 9, 16, and 17) by the existing method are larger than those by our algorithm, and the results of other nets are reversed. In $L2$, the OCAs (net 7, 8, 9, 11, 12, 14, 15, and 16) by the existing method are larger than results by our algorithm. Finally, our method gives the detailed position to optimize the IC layout, so it makes layout optimization design intuitive. Figure 5 shows the OCA of $L1$ extracted by our algorithm, and it can give the detailed OCA position to optimize the IC layout shown in Table 3. As shown in Table 3, in the "$N(c)$", "N" is the net number and "c" is the number of OCAs in a corresponding net. The 2nd–5th columns in Table 3 are the detailed OCA positions for the corresponding net. For net 1, we can see three open critical areas, of which the minimum rows are 53, 295, and 585, maximum rows are 53, 295, and 585, minimum columns are 82, 82, and 84, and maximum columns are 91, 91, and 91 respectively. Likewise, we can obtain the detailed OCA positions of other nets. That position information is important for the net automatic optimization in the further process.

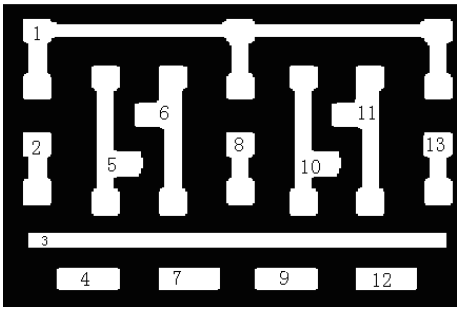


Fig. 6. SRAM layout with the net numbers.

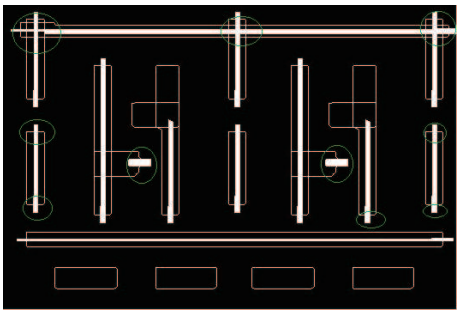


Fig. 7. OCA extraction in Ref. [35].

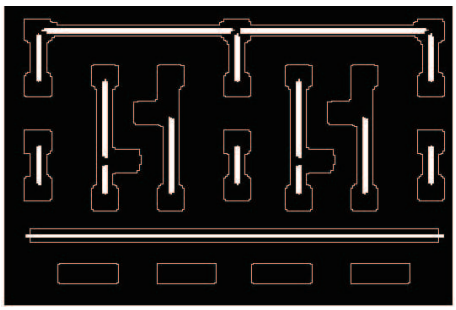


Fig. 8. OCA extraction by our algorithm.

Table 2. Performances comparison of $L1$.

Net No.	OCA $L1$	OCA $L1o$	OCA $L2$	OCA $L2o$
1	266.0	276.0	8810.3	8663.1
2	1749.0	1265.5	260.0	0.0
3	250.1	366.6	275.0	0.0
4	189.0	937.0	260.0	0.0
5	0.0	25.0	7957.5	7142.1
6	63.0	444.0	182.0	0.0
7	250.1	350.6	1359.5	1771.8
8	261.0	355.5	2319.9	2739.8
9	1068.0	551.3	1018.6	1328.6
10	294.1	494.4	182.0	0.0
11	195.0	586.3	1947.5	2065.9
12	200.1	424.0	2975.6	3705.6
13	294.1	492.5	270.3	0.0
14	261.0	355.5	3443.3	4316.3
15	261.0	355.5	429.4	649.5
16	1749.0	1265.5	637.5	999.5
17	1302.0	940.8	189.0	0.0

Table 3. OCA detailed positions of $L1$.

$N(c)$	Minimum row	Maximum row	Minimum column	Maximum column
1(3)	53,295,585	282,573,651	82, 82, 84	91,91,91
2(3)	51,146,438	138,430,653	126,126,126	136,136,136
3(2)	80,80	106,90	180,257	252,297
4(1)	177	190	180	250
5(2)	241,293	285,334	175,175	182,182
6(1)	388	402	180	250
7(2)	470,486	496,496	180,257	252,297
8(2)	531,583	575,628	175,175	185,185
9(3)	52,173,439	137,406,652	339,339,339	348,348,348
10(2)	80, 80	90,106	390,437	432,509
11(1)	177	190	437	509
12(1)	334	402	390	509
13(2)	486,470	496,496	390,437	432,509
14(2)	239,293	285,336	504,504	514,514
15(2)	529,583	575,630	502,502	529,512
16(3)	51,146,438	138,430,653	551,551,551	561,561,561
17(3)	52,295,585	282,573,652	596,596,596	608,608,605

4. Conclusions

The study of the critical area is a significant content of the design for manufacturability, and the critical area plays an important role in predicting the IC yield and reliability. By means of mathematical morphology, in this paper we present a new computation model and novel extraction algorithm of the open critical area based on the net flow-axis. The net shape and the defect shape in this algorithm are unconstrained, i.e., the algorithm is applicable to any shape of the nets and does not need to assume the defect model. Therefore, this algorithm is widely used. Then, in this algorithm the defect size and spatial distribution in practical application should be taken into account. For the different spatial distributions of defects, we can use the algorithm in this paper to extract the critical area. So this algorithm is suited to the different spatial distributions of defects. Besides, this algorithm is insensitive to the defect shape. Therefore in the algorithm there is no need to take into consideration the defect shape such as ellipse and circle, thereby overcoming the disadvantage that for diverse shape defects different methods have to be adopted. In addition, this algorithm has lower time complexity and high precision, and it can be implemented in parallel and find the specific position information of the optimized layout. Due to the advantages of the algorithm above, it shows a good application prospect. Based on the image processing, in this paper the defects of missing material in the IC manufacturing process are combined with the geometric shape of the layout to obtain the

position and the size of the open circuit critical area. This effective information provides premise condition for the layout optimization and the yield assessment.

References

[1] Muller D 2006 *IEEE/ACM International Conference on Computer-Aided Design*, November 5–9, 2006, San Jose, CA, USA, p. 480

[2] Minsik C, Hua X, Puri R and Pan D Z 2008 *IEEE Trans. Comput. Aid. Des. Integr. Circ. Syst.* **27** 872

[3] Liu S G, Wang J P, Su Y B and Wang L 2012 *Chin. Phys. B* **21** 098503

[4] Zhang C X, Ding X H, Wang R, Zhou Y Q and Kong L M 2012 *Chin. Phys. B* **21** 034202

[5] Sinha S, Su Q, Wen L N, Lee F, Chiang C, Cheng Y K, Lin J L and Harn Y C 2008 *IEEE Trans. Semicond. Manuf.* **21** 14

[6] Pan D Z, Cho M and Kun Y 2010 *Foundations and Trends in Electronic Design Automation* **4** 1

[7] Xue X Y, Xu S R, Zhang J C, Lin Z Y, Ma J C, Liu Z Y, Xue J S and Hao Y 2012 *Chin. Phys. B* **21** 027803

[8] Cong J, Fan Y P and Jiang W 2006 *IEEE/ACM International Conference on Computer Aided Design*, November 5–9, 2006, San Jose, CA, USA, p. 7

[9] Cross D, Nequist E and Scheffer L 2007 *2007 International Symposium on Physical Design*, March 18–21, 2007, Austin, TX, USA, p. 171

[10] Zhu Z M, Hao B T, Li R and Yang Y T 2010 *Acta Phys. Sin.* **59** 2003 (in Chinese)

[11] Ling X, Hu M B and Ding J X 2012 *Chin. Phys. B* **21** 098902

[12] Barnett T S, Bickford J and Weger A J 2007 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, June 11–12, 2007, Stresa, Italy, p. 351

[13] Ducotey G, Couvrat A, Audran V, Pepper D, Couturier L and David D 2008 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, May 5–7, 2008, Cambridge, MA, USA, p. 138

[14] Izuka T, Ikeda M and Asada K 2007 *IEEE Transactions on VLSI Systems* **15** 716

[15] Chong S, Rying E, Perry A, Lam S and St. Lawrence M A 2007 *2007 IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, June 11–12, 2007, Stresa, Italy, p. 69

[16] Ronnie P, Eshwege H, Valfer E, David D, Pepper D, Cricchio F, Hinschberger B and Kolar D 2008 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, May 5–7, 2008, Cambridge, MA, USA, p. 53

[17] Porat R, Dotan K, Hemar S, Levin L, Li K, Sung G, Lin C T, Lin S K and Wang H I 2008 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, May 5–7, 2008, Cambridge, MA, USA, p. 11

[18] Jansen S, Florence G, Perry A and Fox S 2008 *IEEE/SEMI Advanced Semiconductor Manufacturing Conference*, May 5–7, 2008, Cambridge, MA, USA, p. 69

[19] Wang J P and Hao Y 2005 *Chin. J. Semicond.* **26** 1514

[20] Dong G, Liu J, Xue M and Yang Y T 2011 *Acta Phys. Sin.* **60** 046602 (in Chinese)

[21] Zhou S Y, Wang K, Zhang Y F, Pei W J, Pu C L and Li W 2011 *Chin. Phys. B* **20** 080501

[22] Liu F, Zhao H, Li M, Ren F Y and Zhu Y B 2010 *Chin. Phys. B* **19** 040513

[23] Deng S X and Liang S D 2012 *Chin. Phys. B* **21** 047306

[24] Li D M, Wang Z H, Huang L Y and Gou Q J 2007 *Chin. Phys.* **16** 3760

[25] Ao B Y, Xia J X, Chen P H, Hu W Y and Wang X L 2012 *Chin. Phys. B* **21** 026103

[26] Wang J P and Hao Y 2009 *Acta Phys. Sin.* **58** 4267 (in Chinese)

[27] Walker H and Director S W 1986 *IEEE Trans. Comput. Aid. Des. Integr. Circ. Syst.* **5** 541

[28] Deng S X and Liang S D 2012 *Chin. Phys. B* **21** 047306

[29] Feng Q, Xing T, Wang Q, Feng Q, Li Q, Bi Z W, Zhang J C and Hao Y 2012 *Chin. Phys. B* **21** 017304

[30] Wang J P, Hao Y and Zhang J M 2007 *Chin. Phys.* **16** 1796

[31] Meng Y J, Liu Y W and Tang Y H 2012 *Chin. Phys. B* **21** 074206

[32] Sun D P and Hao Z X 2010 *Journal of Computer Research and Development* **47** 1244 (in Chinese)

[33] Sun X L, Hao Y and Song G X 2007 *Journal of Electronics & Information Technology* **29** 496 (in Chinese)

[34] Evanthia P 2011 *IEEE Tran. Comput. Aid. Des. Integr. Circ. Syst.* **30** 704

[35] Wang J P and Hao Y 2006 *Chin. Phys.* **15** 1621