CASE STUDIES ON LITHOGRAPHY-FRIENDLY VLSI CIRCUIT LAYOUT

A Thesis

by

PRATIK JITENDRA SHAH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2008

Major Subject: Computer Engineering

CASE STUDIES ON LITHOGRAPHY-FRIENDLY VLSI CIRCUIT LAYOUT

A Thesis

by

PRATIK JITENDRA SHAH

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Approved by:

Chair of Committee,	Jiang Hu
Committee Members,	Weiping Shi
	Duncan Moore Walker
Head of Department,	Costas Georghiades

December 2008

Major Subject: Computer Engineering

ABSTRACT

Case Studies on Lithography-Friendly VLSI Circuit Layout.
(December 2008)

Pratik Jitendra Shah, B.E., R.V.College of Engineering
Chair of Advisory Committee: Dr. Jiang Hu

Moore's Law has driven a continuous demand for decreasing feature sizes used in Very Large Scale Integrated (VLSI) technology which has outpaced the solutions offered by lithography hardware. Currently, a light wavelength of 193nm is being used to print sub-65nm features. This introduces process variations which cause mismatches between desired and actual wafer feature sizes. However, the layout which affects the printability of a circuit can be modified in a manner which can make it more lithography-friendly.

In this work, we intend to implement these modifications as a series of perturbations on the initial layout generated by the CAD tool for the circuit. To implement these changes we first calculate the feature variations offline on the boundaries of all possible standard cell pairs used in the circuit layout and record them in a Look-Up Table (LUT). After the CAD tool generates the initial placement of the circuit, we use the LUT to estimate the variations on the boundaries of all the standard cells. Depending on the features which may have the highest feature variations we assign a cost to the layout and our aim is now to reduce the cost of the layout after implementing perturbations which could be a simple cell flip or swap with a neighboring cell. The algorithm used to generate a circuit placement with a low cost is Simulated Annealing which allows a high probability for a solution with a higher cost to be selected during the initial iterations and as time goes on it tends closer to the greedy algorithm. The idea here is to avoid a locally optimum solution. It is also essential to minimize the impact of the iterations performed on the initial solution in terms of wirelength, vias and routing congestion.

We validate our procedure on ISCAS85 benchmark circuits by simulating dose and defocus variations using the Mentor tool Calibre LFD. We obtain a reduction of greater 20% in the number of instances with the highest cell boundary feature variations. The wirelength and the number of vias showed an increase of roughly 2.2-8.8% and 1.2-7.8% respectively for different circuits. The routing congestion by and large remains unaffected.

ACKNOWLEDGEMENTS

This work is a result of the constant help I received from faculty, friends and family. I want to extend my heart-felt gratitude to all of them for having provided all the technical and motivational support throughout my Master's program.

First of all, I thank my advisor, Dr. Jiang Hu. He has been a great advisor and a pleasure to work with and always the first person I would approach for any help over the past couple of years. He has always extended complete support to me and I will always appreciate him for this.

I would also like to thank my committee members, Dr. Weiping Shi, and Prof. Hank Walker, for their support throughout the course of this research.

I thank all the staff at the Electrical and Computer Engineering Department for taking care of all the administrative issues and tolerating sometimes nagging questions.

I cannot mention names of all my friends here, for the space will be too short and I might be taken to task if I skip any. But my sincere thanks to all of them here in the US and back home in India.

Finally, thanks to my parents and family for all their patience and love.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

FIGURE Page

LIST OF TABLES

1. INTRODUCTION

In sub-65nm VLSI technology, lithography hardware solutions have been unable to match the increasing demands for feature-size reduction. This has led to undesired mismatches between mask layout feature sizes and those actually printed on the wafer. As these variations approach the tolerance limit, they make a significant impact on the timing and power consumption of a circuit.

The printability of a layout can be measured in terms of the accuracy with which we are able to print the smallest feature sizes in the circuit. These are often referred to as the critical dimension (CD) of the circuit layout. Current chip fabrication technologies use optical lithography to print the circuits on a silicon wafer. The critical dimension in optical lithography depends on the parameters involved in the following relation.

$$CD = \frac{k\lambda}{NA} \tag{1}$$

To print the smallest feature size we need to use the best possible combination of all the parameters involved in the above equation. This would include:

- a low wavelength (lambda) light source,
- projection system with a high NA (numerical aperture) and
- a low Rayleigh factor k.

Over the years the improvement in resolution (CD) was driven to a large extent by the reduction in the wavelength of the light source used. But the reduction in wavelength has not been able to match the demand in feature size reduction to improve circuit performance. This has largely been due to the limited number of light sources which can deliver the required spectral power to satisfy the lithography throughput requirement and also due to attenuation of light at wavelengths below 193nm in which case lithography would have to be performed in the absence of oxygen and water [1].

This thesis follows the style of *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems.*

The theoretical limit for the NA of a projection system is 1, which refers to the case when the projection lens is able to capture all the light emitted from the light source. However controlling parameters such as aberrations, polarization effects and also the inverse quadratic dependence of the Depth-of-focus (DOF) on the NA (eqn 2) makes it difficult to introduce projection systems with high NA [2].

$$\mathrm{DOF} = \frac{\lambda}{2\mathrm{NA}^2} \tag{2}$$

The DOF is the maximum amount of focus change that can be tolerated before the printed pattern size falls out of the desired specification [1]. Higher values of NA would mean a low DOF which would require a very tight control of focus in the exposure system and also of the wafer flatness.

The third parameter in eqn (1) k has a theoretical lower limit of 0.25[1]. But due to yield issues it is difficult to obtain a k of less than 0.75. To reduce the value of k less than this value requires the implementation of Resolution Enhancement Techniques (RETs) to improve the quality of image. Current RET's used are Optical Proximity Correction (OPC), Off-axis illumination , Sub-resolution Assist features, attenuated and alternating phase-shifting masks among other techniques. This can help reduce the value of k closer to its theoretical limit.

Current lithography techniques use light wavelengths of 193nm, NA of close to 0.75 and a value of k close to 0.3. They use various complex RET's which make the mask making process considerably expensive. This combined with the increasing demand to make smaller feature sizes introduces heavy constraints on the manufacturing process and also forces designers to set aggressive timing targets [3]. This introduces a lot of process variations in the die. The polysilicon features forming the transistor gates are affected the most due to these variations since they have the smallest feature sizes. This makes it essential to control the variations by making the layout as litho-friendly as possible. This would not only reduce the burden on the lithographic process, ensure a more accurate design in terms of timing and power consumption and also an improved yield.

In [4, 5] methods are introduced to predict the electrical behavior of wires and transistors using contour-based lithography simulations and they implement the methodology for full-chip timing and power analysis. However it does not attempt to improve the printability of the circuit. [6, 7] deal with RET-aware physical design for routing problems. These works however do not improve the variations on the polysilicon gates which are generally implemented using the smallest feature size. CD variations on these features directly affect the timing and performance of the circuit. One method to improve these variations is to implement circuit fabrics using a regular physical geometry and has been implemented in [8, 9]. The disadvantage of such an implementation is a loss in circuit performance.

To draw a lithography-friendly layout it is essential that the library of standard cells used can achieve a high printability for all the involved transistors. However, this cannot guarantee the printability of the boundary transistors which are significantly affected by the neighboring cells especially in a dense layout which indicates that the placement of the standard cells in the layout can affect the printability of the circuit. This leads us to investigate the possibility of performing perturbations on the placed circuit generated by a CAD tool such that we may be able to obtain a better placement solution in terms of printability. A perturbation with respect to a standard cell may refer to a simple cell flip or can also be extended to cell exchange with a neighboring cell. Using such perturbations it may be possible to reduce the boundary variations in standard cells which would help us achieve a much more lithography-friendly layout.

Problem Statement: Given a circuit generated by a placer, implement perturbations to improve its printability on the basis of prior knowledge of standard cell behavior such that the final placement is more lithography-friendly.

In this work we restrict the analysis on the polysilicon features forming transistor gates at the boundaries of standard cells. To the author’s knowledge the closest related works are [10, 11]. In [10] perturbations are performed to the circuit placement but these are restricted to spacing optimization between the standard cells. This restricts the quality of lithography-friendly cell placement. [11] implements a dynamic programming based approach to cell perturbations which includes cell flipping and swapping but is implemented with standard cell with feature sizes of 130nm which show lesser variation compared to the sub-100nm feature sizes and it also does not verify the results by measuring the variations generated on boundary cell variations in the actual circuits.

In this work the required prior information regarding the behavior of standard cells is calculated offline and maintained in the form of a Look-Up table (LUT). Each record in the LUT corresponds to the boundary polysilicon variations when any two standard cells in the concerned library are placed adjacently. The cases corresponding to different cell orientations are also taken into account.

Next we perform a series of perturbation iterations based on the data in the LUT beginning with the circuit generated by the placer 'S0', such that the final circuit generated 'Sn' will be more lithography-friendly. Each iteration involves perturbations performed on the current solution 'Sm' to generate a new intermediate solution 'S (m+1)'. The perturbations are performed on the basis of the data stored in the LUT. Whether the solution 'S (m+1)' is more lithography-friendly than 'Sm' depends on the algorithm used to perform the perturbations involved in each iteration.

The iterations may be performed in a "greedy" fashion which would try to obtain a better solution (i.e. a more lithography-friendly solution) than the current solution. However, this method may return a result which is only locally optimum. Hence an approach using Simulated Annealing is implemented which assigns a probability to select a solution with a "higher cost" (which is not as lithography-friendly as the current solution) on each iteration. This probability reduces with the number of iterations performed and towards the end the approach becomes similar to the greedy method since iterations will be performed only to obtain a "better solution". This method improves the chances of obtaining a better optimum solution than the greedy approach.

The initial placed circuit (S0) is implemented using a CAD tool which although not lithography-driven generates a high quality solution (for e.g. in terms of wire length). Hence it is important that the final solution generated does not degrade the quality of the initial solution in terms of total wire length, total number of vias used and routing congestion. Also in an actual circuit the standard cells may have different CD variations when compared to the corresponding value of standard cell boundary variations placed at the same distance in the LUT. Hence it is essential to verify the improvement in printability between the final and initial solution by generating the actual value of standard cell CD variations.

The remainder of the thesis is organized as follows. Section 2 explains the formulation of the Look-Up table which records the standard cell behavior. In section 3, definitions of the various layout parameters such as the estimation of CD variations, layout cost and routing parameters have been defined. Section 4 discusses the methodology used to improve a feature variation by describing the methods of measurement. The implementation of the perturbation iterations using Simulated Annealing are described in Section 5. We discuss the tools and experimental details in Section 6 and present the results obtained on ISCAS benchmark circuits in Section 7. This work is concluded in Section 8 along with some ideas for future work.

2. FORMULATION OF THE LOOK-UP TABLE

The printability of a boundary polysilicon feature in a standard cell depends to a large extent on its neighboring cell in the layout. We use the CD variations of a feature as a measure of its printability. The higher the variation, the lesser the quality of printability and vice versa. To estimate the variations of the boundary polysilicon features in a layout we place two standard cells adjacently at a distance which is equal to the closest distance between two standard cells in a dense layout. We then estimate the CD variations (using a tool called Calibre LFD) on the boundary polysilicon features of both the standard cells which face each other. The CD variation is calculated on the basis of the PV (Process Variation) Bands generated by simulating dose and defocus variations in the lithography process. The details involved in the calculation are explained in section 6. Consider two standard cells A and B placed adjacently as shown in Figure 1. The rectangles within each standard cells indicate boundary polysilicon features.

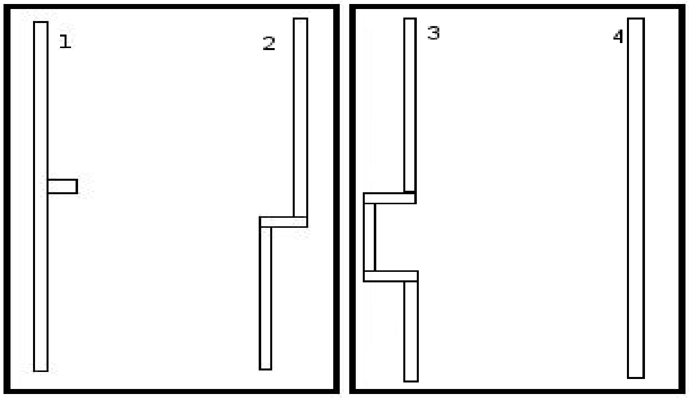
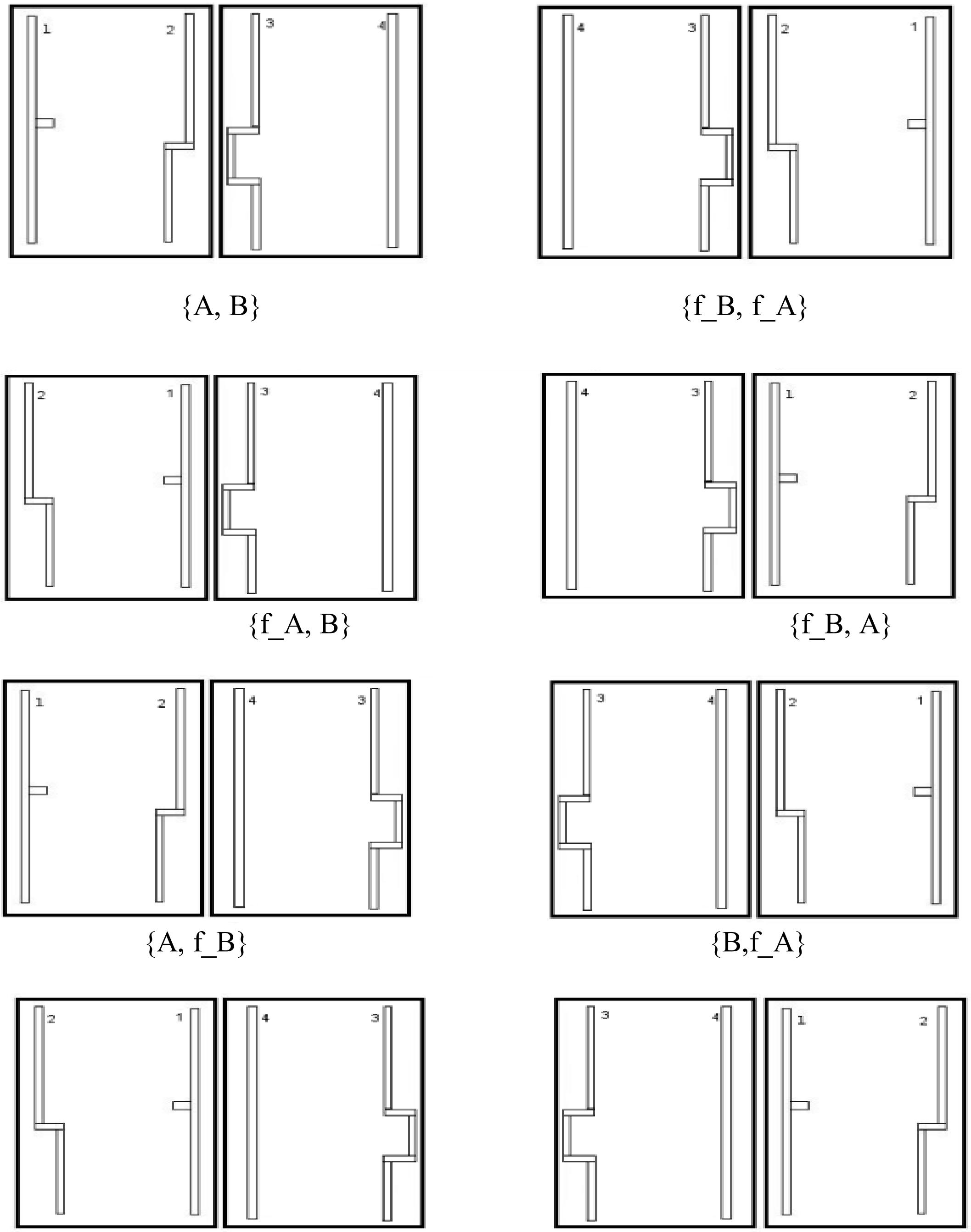


Figure. 1. Standard cell pair {A, B}

We intend to measure the CD variations on the boundary polysilicon features which in this case would be the polysilicon features numbered 2 and 3 in Figure 1 which belong to standard cells A and B respectively. This set of variations is recorded in the LUT. We repeat the process of taking similar boundary polysilicon measurements with all possible orientations of standard cells A and B. The possible 8 cases are shown in the Figure 2.



{A, B}

{f_B, f_A}

{f_A, B}

{f_B, A}

{A, f_B}

{B,f_A}

{f_A, f_B}

{B,A}

Figure. 2. Possible orientations of standard cells placed adjacently

In Figure 2, f_A and f_B stand for the flipped versions of standard cells A and B respectively. As can be seen in the figure the pairs of standard cells placed on the same row (for example {A, B} and {f_B, f_A}) have the same polysilicon features on the boundary (in this case 2 and 3) but on the opposite sides. Hence we may perform the measurements only for 4 which is half of the above required 8 cases for a given pair of standard cells and enter the corresponding variation measurements in the LUT. We can obtain the measurements for the other 4 cases simply by exchanging the values of the obtained variations in the corresponding similar case.

Table I shows an example of how the boundary variations information is recorded in the LUT. fand2x1 and fand2x2 stand for the flipped versions of the and2x1 and and2x2 standard cells.

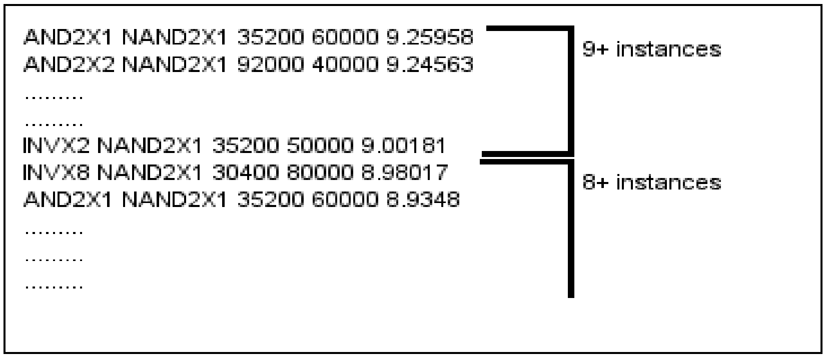
Table I. LUT for 2 standard cells

	and2x1	fand2x1	and2x2	fand2x2
and2x1	8.95,7.24	8.86,9.06	8.71,7.89	8.88,6.46
fand2x1	6.78,7.7	7.11,8.42	6.52,7.65	7.13,5.6
and2x2	5.6,7.13	6.46,8.88	5.46,7.9	5.68,6.45
fand2x2	7.65,6.51	7.9,8.7	7.46,7.66	7.98,5.23

In this work, a total of 23 standard cells are considered and the entire LUT is put up in the appendix.

3. LAYOUT INFORMATION AND COST

The aim of this work is to reduce the variations of polysilicon features on the boundaries of standard cells. To improve the printability of the circuit we begin with trying to improve the printability of the standard cells which would have the highest CD variations. To obtain this information we traverse the layout in a row wise fashion and make a record of the variations on the boundaries of the standard cells by locating the corresponding record in the Look-Up table that was created previously on the basis of the standard cells involved. The boundary cell variations obtained are then sorted in a non-increasing order along with the details of the coordinates of the standard cell in which the variation occurs. This information is used to carry out the perturbations on the current layout of the circuit. A sample of the information is shown in Figure 3. The exact method of perturbations performed to improve the printability of each variation is explained in the following section.



```
AND2X1 NAND2X1 35200 60000 9.25958
AND2X2 NAND2X1 92000 40000 9.24563
.........
.........
INVX2 NAND2X1 35200 50000 9.00181
INVX8 NAND2X1 30400 80000 8.98017
AND2X1 NAND2X1 35200 60000 8.9348
.........
.........
.........
```

9+ instances
8+ instances

Figure 3. Estimated layout variations information

In the above figure we see that the first two columns of a single line store the information of the pair of standard cells at whose boundary the variation occurs. The next two columns indicate the lower left coordinate of the left standard cell and the final column indicates the variation (obtained from the LUT) of the concerned boundary polysilicon feature in percentage (%).

Layout Cost: To measure the lithography-friendliness of the layout generated by the perturbation we define a cost associated with the placement of the standard cells in the layout. After obtaining the variations information in the format above it is possible for us to associate a cost with each layout. To calculate the cost we first obtain the integer values of the variations involved in the layout. This gives us a set of numbers between 1 and the 'floor' of the highest CD variation observed. The highest CD variation is generally 9 for the process parameters in the standard cells considered in this work. We use only the top two values involved in the set of numbers obtained since the features having these variations affect the printability of the circuit the most. For example in figure 3 the two values are 9 and 8. We then obtain the number of instances in the layout that have variations greater than the two integer values obtained previously. For example if 8 and 9 are the integer parts of the top most CD variations then we calculate the number of instances which have variations greater than 9 and the number of instances which have variations greater than 8 but less than 9. We then calculate the cost of the layout with the following relation.

$$\text{Cost} = (\text{highest integer})^3 * (\text{no. of instances with value} >= \text{highest integer}) + (\text{highest integer-1})^3 * (\text{no. of instances with value} >= \text{higest integer -1}) \quad (3)$$

The cubes of the integer values as shown in the relation above are introduced to assign a higher cost to the variations with the highest values. Hence if a layout has higher number of instances with variations having a value greater than the highest integer then the cost associated with the layout will also be higher.

Our intention in this work is to reduce this cost by reducing the number of instances having high CD variations. To do this we use the Simulated Annealing algorithm details of which will be discussed in the following sections.

Routing Information: Since we perform perturbations on the placement of the circuit in an attempt to improve the printability of the circuit there will be a change in the routing parameters involved. We record the following information to calculate the degree of change that was brought in between the initial layout generated by the CAD tool and the final layout obtained after the required perturbations are performed.

- Total Wirelength.
- Vias used.
- Congestion: The routing congestion can be measured in different areas of the circuit and is generally represented in terms of a ratio called the congestion ratio.

  This ratio can be calculated for each region using the following relation.

  $$\text{Congestion ratio} = \text{Utilized Tracks}/\text{Total Tracks} \tag{4}$$

  where the Total tracks represent the total number of routing tracks available to be used by the nets and the utilized tracks represent the number of tracks that were actually used for routing. Since it is a ratio it has a value always less than 1. The congestion for the entire circuit is calculated and represented in the form of a histogram which gives us a fair idea of the routing density.

4. PERTURBATION METHODOLOGY TO IMPROVE A BOUNDARY VARIATION

As mentioned in the previous section we estimate the variations at the standard cell boundaries using the LUT. We try to improve the printability by trying to reduce the highest CD variations obtained. We do this by considering each of the features (with the highest CD variations) individually. Hereon the polysilicon feature whose CD variation we attempt to improve will be addressed as the 'affected poly' and the standard cell in which it occurs will be referred to as the 'hurting cell'. The basic idea is to measure a few possible permutations (or moves) which would change the environment of the 'affected poly' and implement a perturbation based on the best measurement obtained. The method of measurement and perturbation is explained as we go on.

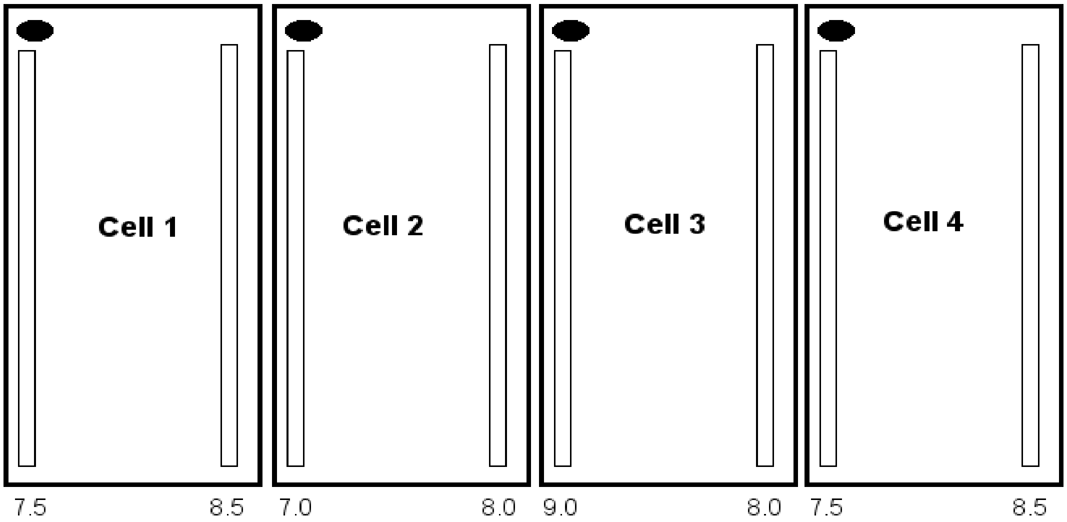


Figure 4. Standard cells in a layout

Consider four standard cells placed adjacently in a layout as shown in Figure 4. The dark spotted region on the top left hand side of each cell indicates the initial unflipped position of each of the standard cells. The CD variations on the boundary polysilicon features are mentioned at the bottom of the figure below the respective features. In the above example the 'affected poly' is on the left boundary of 'Cell 3' which is the 'hurting cell'. To reduce the variation on the 'affected poly' we can try to alter the placement of the considered four cells by measuring all possible permutations with the cells keeping in mind that each cell can be placed in its normal and flipped version. But this leads to a large set of permutations and these only increase when we consider more standard cells. Hence we restrict ourselves to measuring those possible options which affect the immediate environment of the 'affected poly'. The following examples show two of the options.

- Flipping the 'hurting cell' as shown in Figure 5.

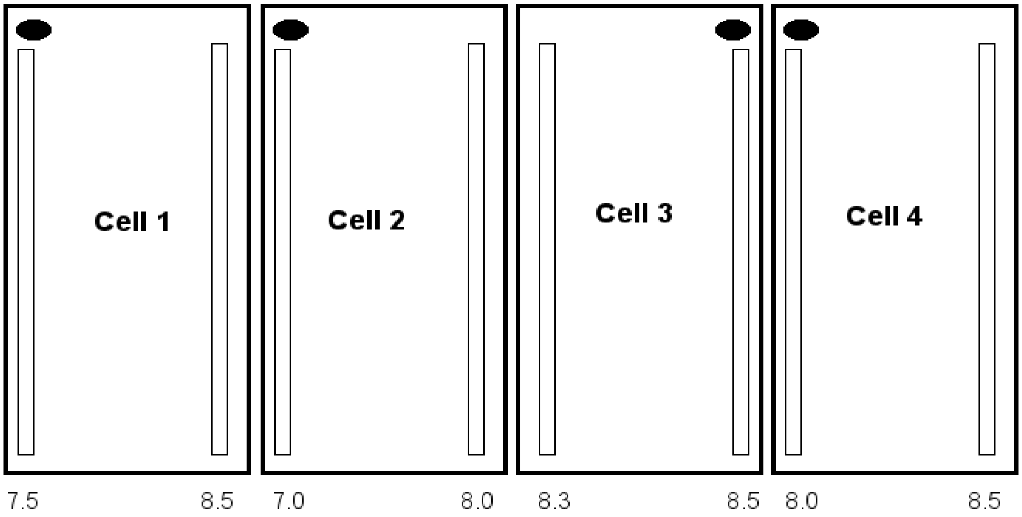


Figure 5. Cell placement after flipping Cell 3

- Swapping Cell 1 and Cell 2 as shown in Figure 6.

  Flipping standard cells is used in [12] for wirelength reduction but in this case we use it to improve printability. The first option with just a cell flip may be seen as more effective of the two possible cases shown in the Figures 5 and 6. This is because the 'affected poly' has its CD variation reduced from 9.0 to 8.3 and none of the other boundary features that could have been affected by the move, for example the right boundary of Cell 2 and the left boundary of Cell 4 (in Figure 5) record a variation greater than that of the initial 'affected poly' whose value is 9.0 although there is an increase in their values compared to the initial case in Figure 4.

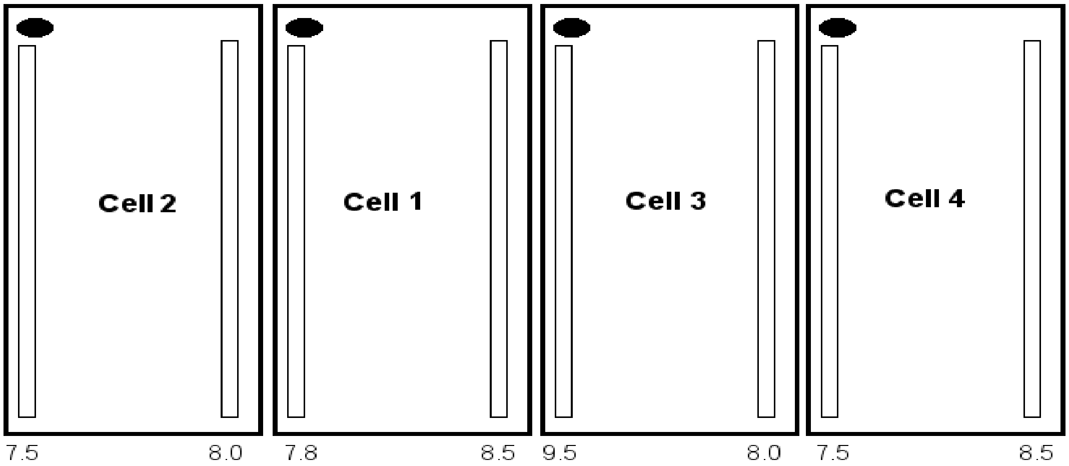


Figure 6. Cell placement after swapping Cell 1 and Cell 2.

This is a trade-off that may be observed on implementing these perturbations i.e. to reduce the CD variation on a particular boundary feature we may end up increasing the variation on a neighboring feature. We hence need to decide a limit for its increase since this would decide the nature of perturbations that would be implemented as discussed later.

The second option as shown in Figure 6 is not that attractive since the 'affected poly' has its CD variation increased from 9.0 to 9.5. Hence if given a choice one may choose to go ahead with the first option.

It is also observed that a perturbation can affect the variations of cells which are placed away from the 'hurting cell' such as Cell 4 in Figure 5 and Cell 1 in Figure 6. We need to make sure that the changed CD variations are taken into account to measure the quality of a 'move' (one of the possible permutations) and only then should a perturbation be actually implemented.

In actual implementation we also consider a permutation which would allow us to swap standard cells which have another standard cell in between them. For example we can consider swapping cells 2 and 4 in Figure 7.

The figure shows the placement of seven standard cells. Cell 4 is the hurting cell in this case. We restrict all the possible perturbations to the standard cells within the bracketed region shown. For example we may try a perturbation which swaps cells 5 and 6 (on the right boundary of the hurting cell) or swaps cells 2 and 3 (on the left boundary of the hurting cell), but do not consider cells 1 and 7 to implement a perturbation. These cells will be used only to measure a perturbation. The idea is to only cause a small perturbation so that the change in wirelength is not large.

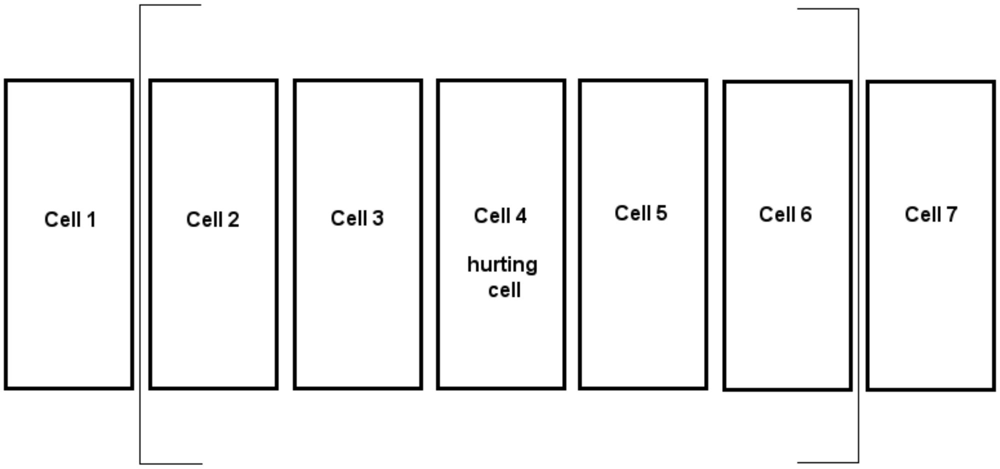


Figure 7. Typical placement of cells considered for measurement (5plus2)

The brackets are imaginary lines which enclose those standard cells which are considered for perturbation to improve the CD variation of the 'affected poly' without significantly increasing the variations on the cells which may be affected by the perturbation. Since there are 5 standard cells in the perturbation region and 2 outside it which are used apart from the 5 standard cells to measure the value of a 'move' we refer to such an arrangement as '5plus2'. The other cases which occur on boundaries of the layout will be discussed later.

4.1. METHOD OF MEASUREMENT

As mentioned earlier, we intend to measure a variety of ‘moves’ which would change the environment of the ‘affected poly’ and implement a perturbation based on the best one. To measure the value of a ‘move’ we:

- maintain an upper limit to the value of a CD variation. When we consider a possible ‘move’ we estimate the new boundary CD variations of all the boundary features within the bracketed region and the ones immediately before and after the region using the LUT. We have to ensure that none of these CD variations exceed an upper limit which allows us to control the kind of perturbations to be introduced in the circuit placement. We shall now refer to this ‘upper limit’ as just ‘limit’. The value for ‘limit’ is very close to the value of the CD variation that we are trying to reduce. We may choose to keep the value for the ‘limit’ higher than the CD variation being considered (variation of the ‘affected poly’), in which case we will also be considering moves that would potentially introduce higher CD variations than the CD variation of the ‘affected poly’ or we can choose to keep it a certain amount lower than the CD variation of the ‘affected poly’ in which case we would choose only those moves which would reduce the values of the CD variations. The choice of ‘limit’ will be determined in the Simulated Annealing algorithm which will be explained in the following sections. If any of the boundary features we consider for measurement purpose has a variation greater than the ‘limit’ then we assign a value of ∞ (a very high value) to the ‘move’.
- sum up the CD variations of the boundary polysilicon features within the bracketed region based on the values in the LUT. We should also include the CD variations of polysilicon features which face the imaginary brackets but which are not within the bracketed region. For example in Figure 7 this would indicate the CD variations of the polysilicon feature on the right boundary of cell 1 and the left boundary of cell 7. Hence we choose to use the standard cells placed

immediately outside the imaginary bracket region only for the purpose of measurement and not for perturbation.

The 'move' corresponding to the lowest value of the measurement is chosen to be implemented. If all the considered 'moves' have a value of ∞, then none of the moves is chosen for perturbation and we leave the environment of the 'affected poly' unchanged.

4.2. GAPS IN CELL PLACEMENT

So far, we discussed the typical case where we may have a continuous adjacent placement of standard cells in a layout. In this case the distance between the standard cells is equal to the distance between the cells when forming the LUT. However, this may not be the case all the time since there may be larger gaps between the cells or boundary cases where we lesser cells to deal with compared to the 5plus2 case. The latter is discussed in the next subsection and here we discuss the possibilities when there is a gap larger than the typical gap between the cells. This can be illustrated as shown in Figure 8.

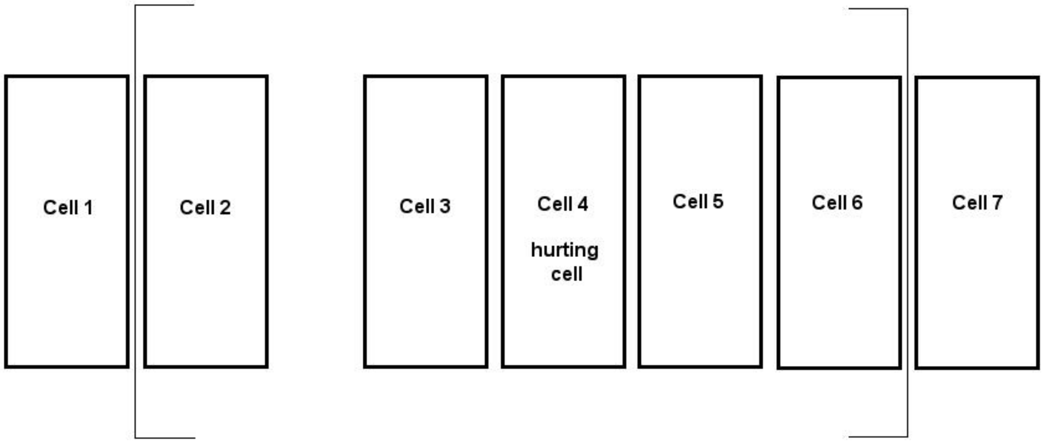


Figure 8. Typical placement 5plus2 but with a large gap between cells 2 and 3

As shown in the figure there is a larger distance between the cells 2 and 3 than the typical distance between the cells. Due to the larger gap the effect of the neighboring cell on the boundary feature variations can be considered to be reduced. Hence we can utilize this gap in such a way that the 'affected poly' is to made to face the gap so that the impact of the neighboring cell on it will be reduced and we may be able to reduce the CD variation. For the example shown cell 3 and cell 4 can be swapped in such a way that the 'affected poly' faces the large gap. Again there are a list of possible permutations that can be made to best improve the CD variation of the 'affected poly'. All the possible permutations (moves) are discussed in the appendix.

4.3. SPECIAL CASES IN CELL PLACEMENT

If the 'hurting cell' is very close to the boundary limits of the core in which the layout is placed then the possible 'moves' we consider to make a perturbation will differ from the '5plus2' case. The various positions in which the 'hurting cell' can be is shown below. Also each of these placements is named depending on the number of standard cells we can use to perform perturbation (i.e. the bracketed region) and the number of cells immediately adjacent to the bracketed region (either 1 or 2).

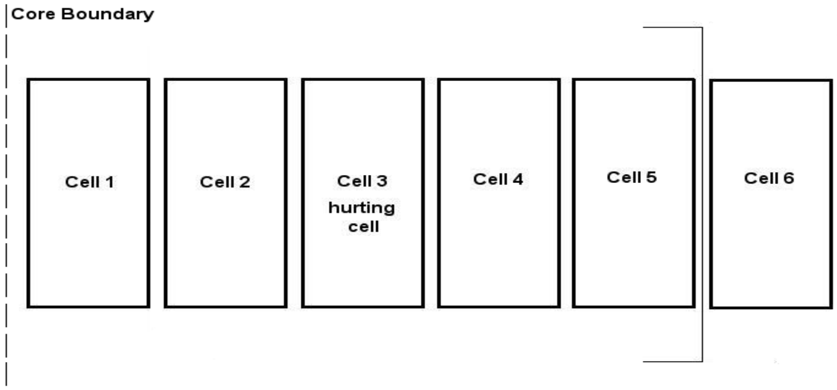


Figure 9. 5plus1, Case 1

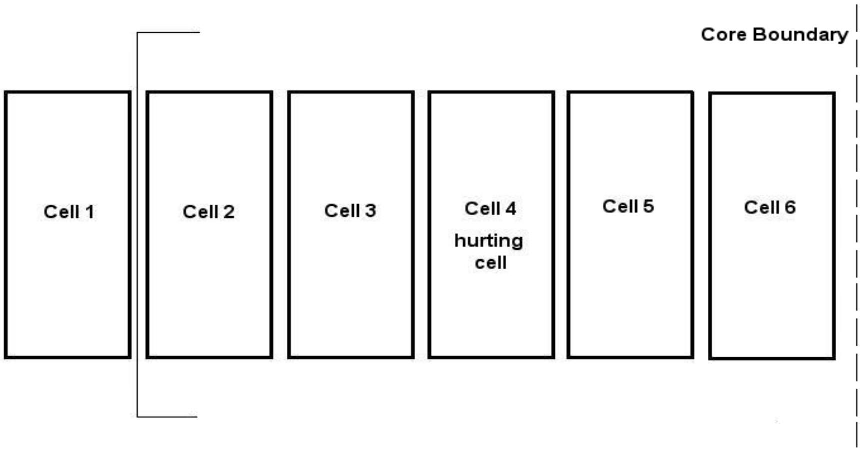


Figure 10. 5plus1, Case 2

Figures 9-14 show the special cases which occur on the boundary of the core. Figures 9, 11 and13 show the hurting cell in different positions close to the left boundary of the core whereas Figures 10, 12 and14 show the hurting cell in different positions close to the right boundary of the core. Again in each of these cases the 'affected poly' may be on the right or the left boundary of the 'hurting cell'. For each of these cases including the typical case of '5plus2' the permutations tried to implement a perturbation are different.

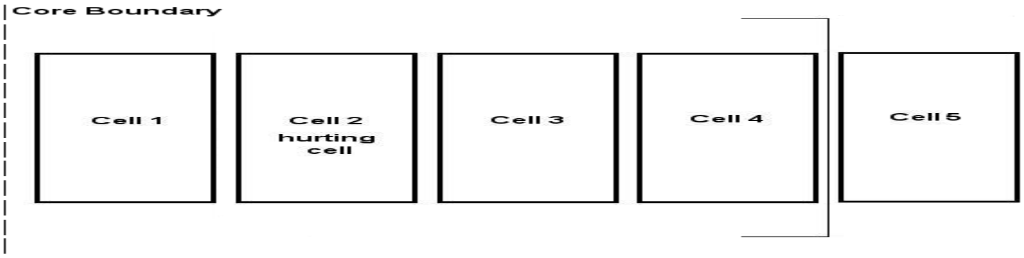


Figure 11. 4plus1, Case 1

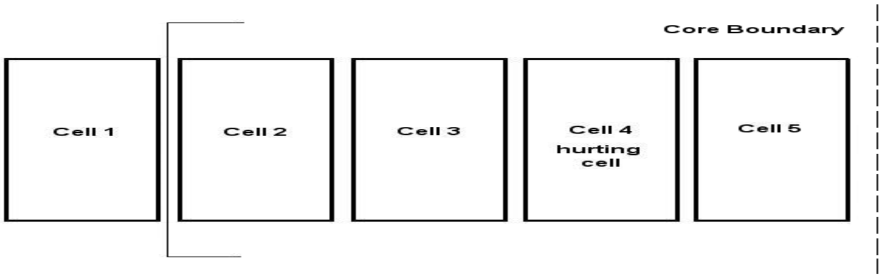


Figure 12. 4plus1, Case 2

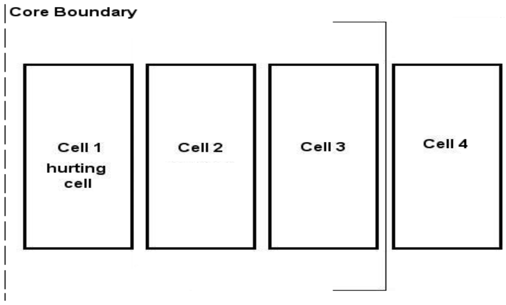


Figure13. 3plus1, Case 1

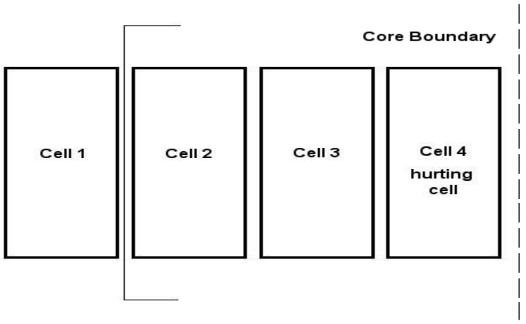


Figure14. 3plus1, Case 2

4.4. 'MOVES' FOR EACH POSSIBLE CELL PLACEMENT

The permutations that are considered for each of the placement cases are mentioned below. We consider the cells in each possible placement to be in increasing order for example {c1, c2, c3, c4, c5, c6, c7} in the case of 5plus2 and corresponding to each case, mention all the permutations that were tried out. The main idea of choosing these particular permutations is to change the environment of the 'affected poly'.

a) 5plus2 with 'affected poly' on the left boundary: {c1,c2,c3,fc4,c5,c6,c7}, {c1,c2,fc3,fc4,c5,c6,c7}, {c1,fc2,fc3,fc4,c5,c6,c7}, {c1,c2,c3,c4,fc5,c6,c7}, {c1,c2,c3,c4,fc5,fc6,c7}, {c1,c2,c3,c5,c4,c6,c7}, {c1,c2,c3,c5,c4,fc6,c7}, {c1,c2,fc3,c5,c4,c6,c7}, {c1,c2,fc3,c5,c4,fc6,c7}, {c1,fc2,fc3,c5,c4,c6,c7},{c1,fc2,fc3,c5,c4,fc6,c7}, {c1,c2,c3,c6,c5,c4,c7}, {c1,c2,c3,c6,c5,fc4,c7}, {c1,c2,fc3,c6,c5,c4,c7}, {c1,c2,fc3,c6,c5,fc4,c7}, {c1,fc2,fc3,c6,c5,c4,c7}, {c1,fc2,fc3,c6,c5,fc4,c7}, {c1,c2,c3,c4,c6,c5,c7}, {c1,c2,c3,c4,fc6,c5,c7}, {c1,c2,c3,c4,fc6,fc5,c7}, {c1,fc2,c3,c4,fc6,fc5,c7}

b) 5plus2 with ‘affected poly’ on the right boundary: {c1,c2,c3,fc4,c5,c6,c7}, {c1,c2,c3,fc4,fc5,c6,c7 }, {c1,c2,c3,fc4,fc5,fc6,c7}, {c1,c2,fc3,c4,c5,c6,c7}, {c1,fc2,fc3,c4,c5,c6,c7}, {c1,c2,c4,c3,c5,c6,c7}, {c1,fc2,c4,c3,c5,c6,c7}, {c1,c2,c4,c3,fc5,c6,c7}, {c1,fc2,c4,c3,fc5,c6,c7}, {c1,c2,c4,c3,fc5,fc6,c7},{c1,fc2,c4,c3,fc5,fc6,c7 }, {c1,c4,c3,c2,c5,c6,c7}, {c1,fc4,c3,c2,c5,c6,c7}, {c1,c4,c3,c2,fc5,c6,c7}, {c1,fc4,c3,c2,fc5,c6,c7}, {c1,c4,c3,c2,fc5,fc6,c7}, {c1,fc4,c3,c2,fc5,fc6,c7 }, {c1,c3,c2,c4,c5,c6,c7}, {c1,c3,fc2,c4,c5,c6,c7}, {c1,fc3,fc2,c4,c5,c6,c7}, {c1,fc3,fc2,c4,c5,fc6,c7}

c) 5plus1 with cell3 being the ‘hurting cell’ with the ‘affected poly’ on the left boundary: {c1,c2,fc3,c4,c5,c6}, {c1,fc2,fc3,c4,c5,c6}, {fc1,fc2,fc3,c4,c5,c6}, {c1,c2,c3,fc4,c5,c6}, {c1,c2,c3,fc4,fc5,c6}, {c1,c2,c4,c3,c5,c6}, {c1,c2,c4,c3,fc5,c6}, {c1,fc2,c4,c3,c5,c6}, {c1,fc2,c4,c3,fc5,c6}, {fc1,fc2,c4,c3,c5,c6},{fc1,fc2,c4,c3,fc5,c6}, {c1,c2,c4,c5,c4,c3,c6}, {c1,c2,c5,c4,fc3,c6}, {c1,fc2,c5,c4,c3,c6}, {c1,fc2,c5,c4,fc3,c6}, {fc1,fc2,c5,c4,c3,c6}, {fc1,fc2,c5,c4,fc3,c6}, {c1,c2,c3,c5,c4,c6}, {c1,c2,c3,fc5,c4,c6}, {c1,c2,c3,fc5,fc4,c6}, {fc1,c2,c3,fc5,fc4,c6}

d) 5plus1 with cell3 being the ‘hurting cell’ with the ‘affected poly’ on the right boundary: {c1,c2,fc3,c4,c5,c6}, {c1,c2,fc3,fc4,c5,c6}, {c1,c2,fc3,fc4,fc5,c6}, {c1,fc2,c3,c4,c5,c6}, {fc1,fc2,c3,c4,c5,c6}, {c1,c3,c2,c4,c5,c6}, {fc1,c3,c2,c4,c5,c6}, {c1,c3,c2,fc4,c5,c6}, {fc1,c3,c2,fc4,c5,c6}, {c1,c3,c2,fc4,fc5,c6},{fc1,c3,c2,fc4,fc5,c6}, {c3,c2,c1,c4,c5,c6}, {fc3,c2,c1,c4,c5,c6}, {c3,c2,c1,fc4,c5,c6}, {fc3,c2,c1,fc4,c5,c6}, {c3,c2,c1,fc4,fc5,c6}, {fc3,c2,c1,fc4,fc5,c6}, {c2,c1,c3,c4,c5,c6}, {c2,fc1,c3,c4,c5,c6}, {fc2,fc1,c3,c4,c5,c6}, {fc2,fc1,c3,c4,fc5,c6}

e) 5plus1 with cell4 being the ‘hurting cell’ with the ‘affected poly’ on the left boundary: {c1,c2,c3,fc4,c5,c6}, {c1,c2,fc3,fc4,c5,c6}, {c1,fc2,fc3,fc4,c5,c6}, {c1,c2,c3,c4,fc5,c6}, {c1,c2,c3,c4,fc5,fc6}, {c1,c2,c3,c5,c4,c6}, {c1,c2,c3,c5,c4,fc6}, {c1,c2,fc3,c5,c4,c6}, {c1,c2,fc3,c5,c4,fc6}, {c1,fc2,fc3,c5,c4,c6},{c1,fc2,fc3,c5,c4,fc6}, {c1,c2,c3,c6,c5,c4}, {c1,c2,c3,c6,c5,fc4}, {c1,c2,fc3,c6,c5,c4}, {c1,c2,fc3,c6,c5,fc4}, {c1,fc2,fc3,c6,c5,c4}, {c1,fc2,fc3,c6,c5,fc4}, {c1,c2,c3,c4,c6,c5}, {c1,c2,c3,c4,fc6,c5}, {c1,c2,c3,c4,fc6,fc5}, {c1,fc2,c3,c4,fc6,fc5}.

f) 5plus1 with cell4 being the ‘hurting cell’ with the ‘affected poly’ on the right boundary: {c1,c2,c3,fc4,c5,c6}, {c1,c2,c3,fc4,fc5,c6}, {c1,c2,c3,fc4,fc5,fc6}, {c1,c2,fc3,c4,c5,c6}, {c1,fc2,fc3,c4,c5,c6}, {c1,c2,c4,c3,c5,c6}, { c1,fc2,c4,c3,c5,c6}, {c1,c2,c4,c3,fc5,c6}, {c1,fc2,c4,c3,fc5,c6}, {c1,c2,c4,c3,fc5,fc6},{c1,fc2,c4,c3,fc5,fc6}, {c1,c4,c3,c2,c5,c6}, {c1,fc4,c3,c2,c5,c6}, {c1,c4,c3,c2,fc5,c6}, {c1,fc4,c3,c2,fc5,c6}, {c1,c4,c3,c2,fc5,fc6}, {c1,fc4,c3,c2,fc5,fc6}, {c1,c3,c2,c4,c5,c6}, {c1,c3,fc2,c4,c5,c6}, {c1,fc3,fc2,c4,c5,c6}, {c1,fc3,fc2,c4,c5,fc6}

h) 4plus1 with cell2 being the ‘hurting cell’ with the ‘affected poly’ on the left boundary: {c1,fc2,c3,c4,c5}, {fc1,fc2,c3,c4,c5}, {c1,fc2,fc3,c4,c5}, {fc1,fc2,fc3,c4,c5}, {c1,fc2,fc3,fc4,c5}, {fc1,fc2,fc3,fc4,c5}, {c1,c2,fc3,c4,c5}, {c1,c2,fc3,fc4,c5}, {c1,c3,c2,c4,c5}, {c1,c3,c2,fc4,c5},{fc1,c3,c2,fc4,c5}, {c1,c2,c4,c3,c5}, {c1,c2,fc4,c3,c5}, {c1,c2,c4,fc3,c5}, {fc2,c1,c3,c4,c5}.

i) 4plus1 with cell2 being the ‘hurting cell’ with the ‘affected poly’ on the right boundary: {c1,fc2,c3,c4,c5}, {fc1,fc2,c3,c4,c5}, {c1,fc2,fc3,c4,c5}, {fc1,fc2,fc3,c4,c5}, {c1,fc2,fc3,fc4,c5}, {fc1,fc2,fc3,fc4,c5}, {fc1,c2,c3,c4,c5}, {c2,c1,c3,c4,c5}, {c2,fc1,c3,c4,c5}, {c2,fc1,fc3,c4,c5},{c2,fc1,fc3,fc4,c5}, {c1,c3,c2,c4,c5}, {c1,fc3,c2,c4,c5}, {fc1,fc3,c2,c4,c5}, {c1,c4,c3,c2,c5}, {c1,c4,c3,fc2,c5}, {c1,c4,fc3,fc2,c5}, {c1,fc4,fc3,c2,c5}, {fc1,fc4,fc3,c2,c5}.

j) 4plus1 with cell4 being the ‘hurting cell’ with the ‘affected poly’ on the left boundary: {c1,c2,c3,fc4,c5}, {c1,c2,fc3,fc4,c5}, {c1,fc2,fc3,fc4,c5}, {c1,fc2,fc3,fc4,fc5}, {c1,c2,c3,c4,fc5}, {c1,c4,c3,c2,c5}, {c1,c4,fc3,c2,c5}, {c1,fc4,fc3,c2,c5}, {c1,c4,fc3,fc2,c5}, {c1,c4,fc3,fc2,fc5},{c1,fc4,fc3,fc2,c5}, {c1,fc4,fc3,fc2,fc5}, {c1,c2,c4,c3,c5}, {c1,c2,c4,fc3,c5}, {c1,c2,fc4,fc3,c5}, {c1,c2,c4,fc3,fc5}, {c1,c2,fc4,fc3,fc5}, {c1,fc2,c4,fc3,fc5}, {c1,fc2,fc4,fc3,fc5}, {c1,c2,c3,c5,c4}, {c1,c2,fc3,c5,c4}.

k) 4plus1 with cell4 being the 'hurting cell' with the 'affected poly' on the right boundary: {c1,c2,c3,fc4,c5}, {c1,c2,c3,fc4,fc5}, {c1,c2,fc3,fc4,fc5}, {c1,fc2,fc3,fc4,fc5}, {c1,c2,fc3,c4,c5}, {c1,fc2,fc3,c4,c5}, {c1,c4,c3,c2,c5}, {c1,fc4,c3,c2,c5}, {c1,fc4,fc3,fc2,fc5}, {c1,c2,c4,c3,fc5},{c1,fc2,c4,c3,c5}, {c1,fc2,fc4,c3,c5}, {c1,c2,c3,c5,c4}, {c1,c2,c3,c5,fc4}, {c1,c3,c2,c4,c5}, {c1,c3,fc2,c4,c5}, {c1,fc3,fc2,c4,c5}.

l) 3plus1 with cell1 being the 'hurting cell' with the 'affected poly' on the right boundary: {fc1,c2,c3,c4}, {c1,fc2,c3,c4}, {c1,fc2,fc3,c4}, {c3,c2,c1,c4}, {c3,c2,fc1,c4}, {c2,c1,c3,c4}, {c2,c1,fc3,c4}, {fc2,c1,fc3,c4}, {fc2,fc1,fc3,c4}, {c1,c3,c2,c4},{c1,fc3,c2,c4}, {c1,fc3,fc2,c4}, {fc1,fc3,c2,c4}, {fc1,fc3,fc2,c4}.

m) 3plus1 with cell4 being the 'hurting cell' with the 'affected poly' on the left boundary: {c1,c2,c3,fc4}, {c1,c2,fc3,c4}, {c1,fc2,fc3,c4}, {c1,fc2,fc3,fc4}, {c1,c4,c3,c2}, {c1,fc4,c3,c2}, {c1,fc4,fc3,c2}, {c1,fc4,fc3,fc2}, {c1,c2,c4,c3}, {c1,fc2,c4,c3},{c1,c2,fc4,c3}, {c1,fc2,fc4,c3}, {c1,c2,fc4,fc3}, {c1,fc2,fc4,fc3}, {c1,c3,c2,c4}, {c1,c3,fc2,c4}, {c1,fc3,fc2,c4}.

5. PERTURBATION ITERATIONS USING SIMULATED ANNEALING

Annealing is a heat treatment technique in which materials are heated to high temperatures which releases atoms from their current positions and then cooled slowly in order to obtain a stress-free structure. In [13] we learn that a similar approach can be used to achieve combinatorial optimization. This is referred to as Simulated Annealing in which we increase the temperature of the system to be optimized to a high value which means that we allow the system to make. In the current context we use this approach to improve the printability of the circuit. This is done by implementing a series of perturbation iterations on the circuit placement. Each iteration involves a set of perturbations that are performed on the current circuit placement to generate a modified placement structure as shown in Figure 15 below.

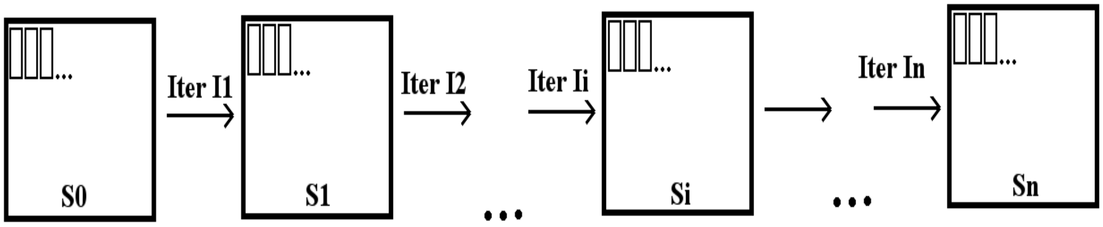


Figure 15. Perturbation iterations on circuit placement

The initial placement of the circuit is generated by the CAD tool. We refer to this as the initial solution S0. On the basis of the highest CD variations in each solution each iteration generates a new solution whose highest feature variations in turn become the basis of the next iteration. The final solution 'Sn' generated by the Simulated Annealing algorithm is a stable one and is more lithography-friendly compared to the initial solution S0. The following pseudo-code explains the procedure involved.

1. Initialize a control variable T (Temperature) to a high value.
2. Generate an initial solution S0.
3. current_solution = S0.

```
4. Loop While (T > Final_T)
     C.  new_solution = PERTURB (current_solution)
     ii. ΔCost = Cost(new_solution) -  Cost(current_solution)
     iii. if (△Cost < 0)
     iv.        current_solution = new_solution ;  T = cooling_schedule(T);
     v.  else if ( exp(-△Cost/T) > rand(0,1) )
     vi.        current_solution = new_solution ; T = cooling_schedule(T);
5. End.
```

The algorithm uses a control variable Temperature which is initialized to a high value. This is analogous to allowing the system to be optimized to be heated to a high temperature just as in the case of annealing. Mathematically, the control variable Temperature is used to control the probability of choosing a neighboring solution which may have a higher cost (analogous to energy in the case of annealing) compared to the current solution. The cost of the system that is used in the algorithm is related to the lithography-friendliness of the placement structure and is defined in Section 2 and equation 3.

The initial solution S0 is generated by the CAD placement tool. We then assign the current solution to be S0. Iterations are performed on the current solution to obtain the next solution and then a decision is made based on the cost of the solution if it could be used as the next 'current_solution' (Si). This is done within the loop which is executed until the control variable T is reduced to zero or a very low value. To find a 'new_solution' we randomly select one of the neighboring solutions of the current solution. A random neighboring solution can be obtained by perturbing the current solution on the basis of its highest feature variations and by randomly choosing a value to decide the 'limit' which is explained in Section 3.1. If the value of 'limit' is greater than the highest feature variations we allow a possibility of selecting a neighboring solution which may have a higher cost compared to the current solution and if the value of ‘limit’ is less than the value of the highest feature variations then the neighboring solution selected will have a lesser cost compared to the current solution.

Next the difference between the cost of the ‘new_solution’ and the ‘current_solution’ is calculated. This is analogous to the two different energy states an atom may occupy and whether the atom moves to a state of higher energy depends on its current temperature and the energy difference between the states. Similarly we choose to convert the ‘new_solution’ to the ‘current_solution’ only depending on cost difference and the value of the control variable T. If the cost difference $\triangle C$ is negative indicating that the ‘new_solution’ is more lithography-friendly, we straight away convert the generated ‘new_solution’ to the ‘current_solution’. If $\triangle C$ is positive then we choose to convert the ‘new_solution’ to the ‘current_solution’ only on the basis of a probability calculated by the expression $\exp[-(\triangle C / T)]$. If the calculated probability is greater than a random number generated which has its value between 0 and 1, then we let the conversion of the ‘new_solution’ to the ‘current_solution’ take place. From the expression it can be inferred that the probability that a solution with a higher cost may be chosen as the next ‘current_solution’ when we have a high T (analogous to a system which is at a high temperature) and a low $\triangle C$.

Each time when a randomly chosen neighboring solution is converted to the ‘current_solution’ the control variable T is lowered or in other words we allow the system to cool down. The rate at which we lower its value determines the quality of solution we obtain. If the rate of cooling is very low then we get a higher chance to obtain a more stable state at each temperature but it also increases the run-time of the algorithm.

In this context we reduce the value of T by a factor 0.9 every time a ‘new_solution’ is converted to a ‘current_solution’. Also if the loop tries 30 different neighboring solutions without obtaining a suitable new ‘current_solution’ from any of them then we exit the loop and thereby end the flow of the program. At the end of the algorithm we have the final placement structure ‘Sn’ as shown in figure 15, which is more lithography-friendly, compared to the initial solution S0.

It must also be noted that so far the cost of a particular solution ‘Si’ has been totally based on the boundary polysilicon feature variations as derived from the LUT. The cost does not verify the actual feature variations that may be different from the values recorded in the LUT. Hence it is essential to perform an actual lithography simulation on the initial and final solutions ‘S0’ and ‘Sn’ respectively and find out if there has been an actual improvement in the lithography-friendliness of the layout. The reason we do not perform lithography simulations within the algorithm and restrict ourselves to the estimated values of variations from the LUT is simply because of run-time issues and that was the whole purpose of formulating the LUT.

6. EXPERIMENTAL SETUP

6.1. STANDARD CELL LIBRARY

The standard cell library used is downloaded from the VLSI Computer Architecture Research Group at the Oklahoma State University [14]. The library supports the TSMC 180nm technology, provides cell layouts, geometry files (in LEF – Library Exchange File format) and timing library files in LIB (liberty), DB (Synopsys compatible) and tlf (Timing Library Format compatible with Cadence) formats. We import all the standard cell layouts from the gds files in the library into Cadence Virtuoso.

6.2. LUT FORMULATION

The procedure involved in the formulation of the LUT requires finding the CD variations of boundary features in standard cell pairs placed adjacently. A total of 23 unique standard cells are considered for this procedure. We automate the layout generation of the required standard cell pairs using Cadence Skill and simultaneously stream out their respective gds files.

These gds files are imported into Mentor Graphics Calibre Wokbench which can be used to scale down the layouts to 65nm technology. Another tool in the Calibre suite called the Calibre LFD (Litho-Friendly Design) is used to generate the PV (Process Variation) bands on the scaled down layouts. The PV bands are generated over 3 subwindows. The concept of a process window can be explained through Figure 16.

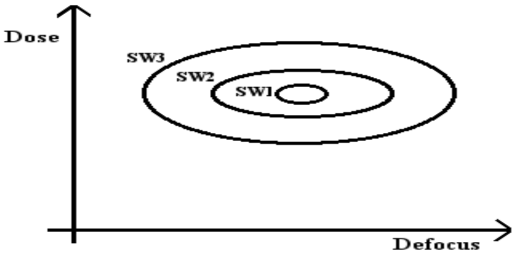


Figure 16. 3 Process Window’s SW1, SW2 and SW3

Figure 16 is obtained from [15]. Each process window (subwindow) can be seen as covering an area of possible dose and defocus levels during production. A larger subwindow indicates greater process variations. The following set of sample points (each sample point indicates (dose,defocus)) are used to calculate the PV Bands in each subwindow. Here D50 indicates a defocus value of 50.

Subwindow1: (0.98, D0), (1.00, D0), (1.02, D0)
(0.98, D50), (1.00, D50), (1.02, D50)

Subwindow2: (0.95, D0), (1.00, D0), (1.05, D0)
(0.95, D50), (1.00, D50), (1.05, D50)

Subwindow3: (0.95, D0), (1.00, D0), (1.05, D0)
(0.95, D100), (1.00, D100), (1.05, D100)

Since the PV Bands are generated over 3 subwindows we have 6 readings associated with each polysilicon feature in the concerned layout. This is because we have a 'Max PV Band' and a 'Min PV Band' associated with each feature (Figure 17) in each subwindow and since there are 3 subwindows we consider, we have a total of 3*2 readings associated with each poly. However, in this work we keep the values of generated for each of the subwindows segregated. The LUT can be generated from the readings of each of the subwindows but we base the work on the LUT generated by the readings of subwindow1 which has the least possible dose and defocus variations and hence the production would need to spend maximum amount of time with such process parameters.

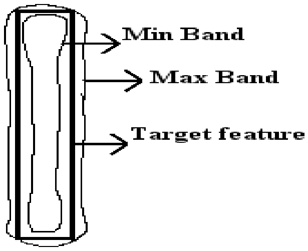


Figure 17. Max and min bands for a feature

The ‘Max PV Band’ indicates the outer edge of the PV Band generated and indicates the thickest feature size that can be printed on the wafer. The ‘Min PV Band’ indicates the thinnest feature that can be printed on the wafer. It can happen that the ‘Max PV Band’ is printed within the boundaries of the target feature (in this case the ‘Min PV Band’ will also be within the target feature). It is also possible that the ‘Min PV Band’ is printed outside the boundaries of the feature (in this case the ‘Max PV Band’ will also be outside the boundaries of the target feature).

The generated PV Bands are used to calculate the CD variations. We focus the calculation only on the gate region of a particular boundary transistor (intersecting region between the polysilicon and the diffusion regions) and not the entire boundary polysilicon feature. In Figure 18 we approximate a PV Band to a rectangular shape to determine the formula used to calculate the CD variation.

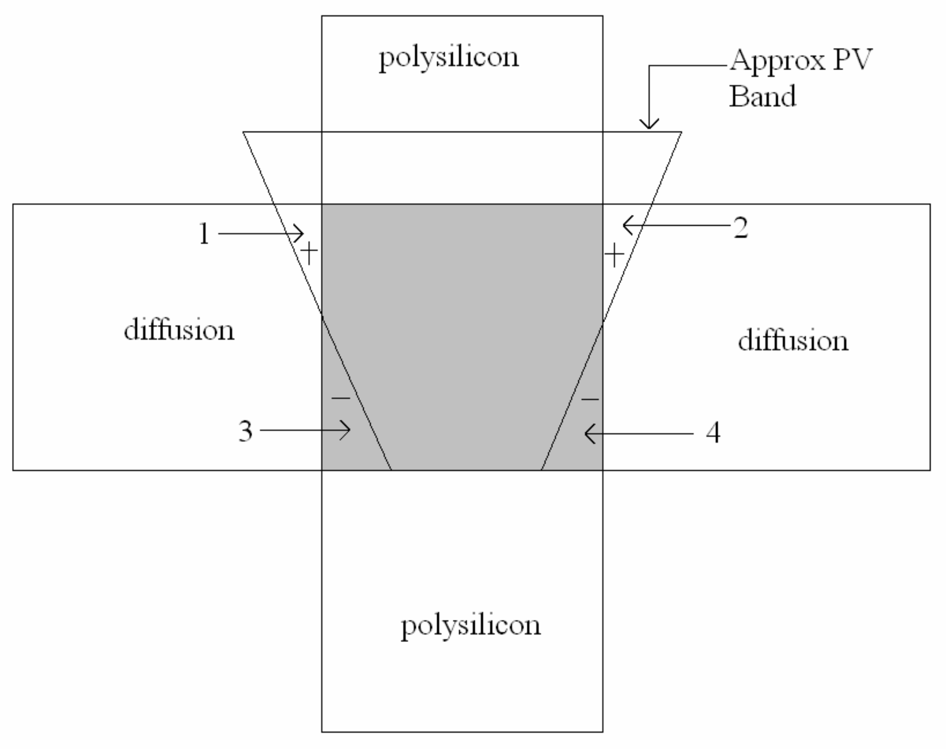


Figure 18. Calculation of CD variation

In the above figure it can be observed that the PV Band generated causes the CD variation to be positive (i.e. the CD exceeds the designed value) in regions 1 and 2 of the figure. On the other hand the CD variation is negative in regions 3 and 4 of the figure. The average CD variation can hence be given by

$$\% \text{ CD variation} = \frac{[\text{Area}(1) + \text{Area}(2)] - [\text{Area}(3) + \text{Area}(4)]}{\text{Area}(\text{gate})} \tag{5}$$

where

Area (gate) = Area ((diffusion) AND (active))

Area (1) + Area (2) = Area ((PV) AND (diffusion)) – Area ((PV) AND (gate))

Area (3) + Area (4) = Area (gate) – Area ((PV) AND (gate))

In the above relations the operator Area (x) indicates calculates the area enclosed by the parameter ‘x’. The AND is a logical operator and represents the intersection of its operands.

After the generation of all the PV Bands for all the required layouts, we use a perl script to create the LUT recording only the boundary feature variations in each layout as explained in Section 2. Since we have 2 values of CD variations for each feature, one each for the Max and Min PV Bands, we have to choose the variation with the higher absolute value to be recorded into the LUT.

6.3. TEST CIRCUITS

ISCAS85 benchmark circuits [16, 17] are used as test circuits to implement the algorithm. The verilog netlists of the circuits are synthesized using Synopsys Design Analyzer. It uses the geometry (LEF) and timing library (lib, tlf) files based on the same standard cell library that was used to create the LUT. The results for each of the test circuits are discussed in the next section.

6.4. SIMULATED ANNEALING

We use Cadence First Encounter to generate the initial solution S0. A Tcl script implementing the algorithm as explained in Section 4, is executed in the First Encounter shell which performs the perturbation iterations, records the placement structure of the intermediate iterations and generates the final solution ‘Sn’.

6.5. VERIFICATION

After generating the required solution ‘Sn’ for a test circuit, we perform lithography simulations using Mentor Graphics Calibre LFD to generate PV Bands on the both ‘S0’ and ‘Sn’. We then use a perl script to record all the standard cell boundary feature variations separately for each case. This helps us verify if there is actual improvement in the printability of the circuit, since the algorithm is executed based on the readings in the LUT which may or may not match with the actual feature variation values.

7. RESULTS AND ANALYSIS

The results of the algorithm on each of the ISCAS85 benchmark circuits are discussed individually as case studies below. For each of the test circuits we:

- plot the variation of the cost function (Section 2)as the solutions iterate from 'S0' to 'Sn'.
- present the estimated standard cell boundary feature variations obtained from the LUT (based on subwindow1) in a histogram format and make a comparison between the solutions 'S0' and 'Sn'.
- present the values of standard cell boundary feature variations from whole-circuit simulations in a histogram format and make a comparison between solutions 'S0' and 'Sn'.
- the impact on routing parameters such as total wirelength, number of via's and congestion.

7.1. c432

We use the verilog netlist for c432, a 27-channel interrupt controller to synthesize a circuit with 194 standard cells. After generating the initial solution 'S0' we use the perturbation algorithm to generate the final solution 'Sn'.

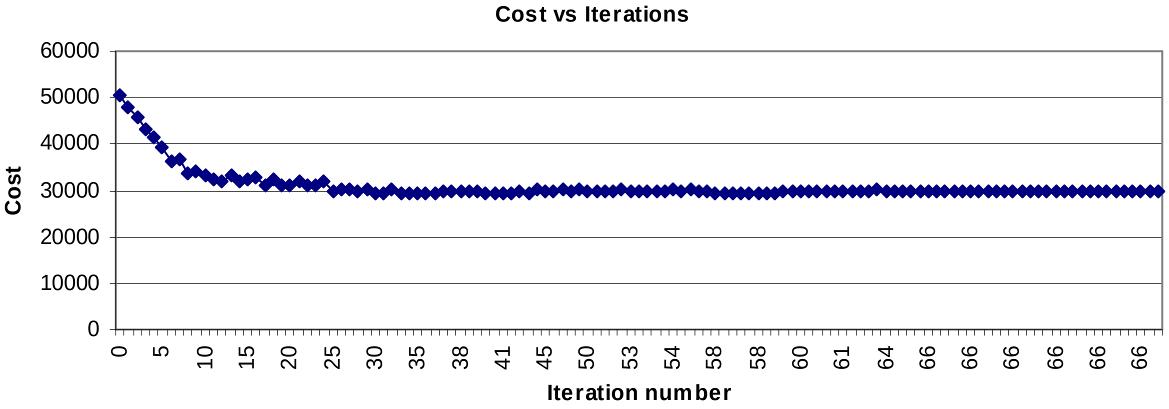


Figure 19. Variation of the cost function with iterations, c432

The variation of the cost function as the solutions iterate is plotted Figure 19. We observe a gradual decrease in the cost of a solution as the iterations proceed, although there are intermediate cases where a solution with a cost greater than the current solution 'Si' is generated. This happens when the cost difference between the new solution and the current solution is very small and the algorithm is still in its initial stages which allows for a high probability to select such a solution.

The estimated boundary feature variations are represented in histogram format as shown in Figure 20 for both the initial solution 'S0' and the final solution 'Sn'. The readings are based on values obtained from the LUT, subwindow 1 (SW1). All the iterations are performed based on these readings, but we shall also analyze the changes caused in subwindows 2 and 3.

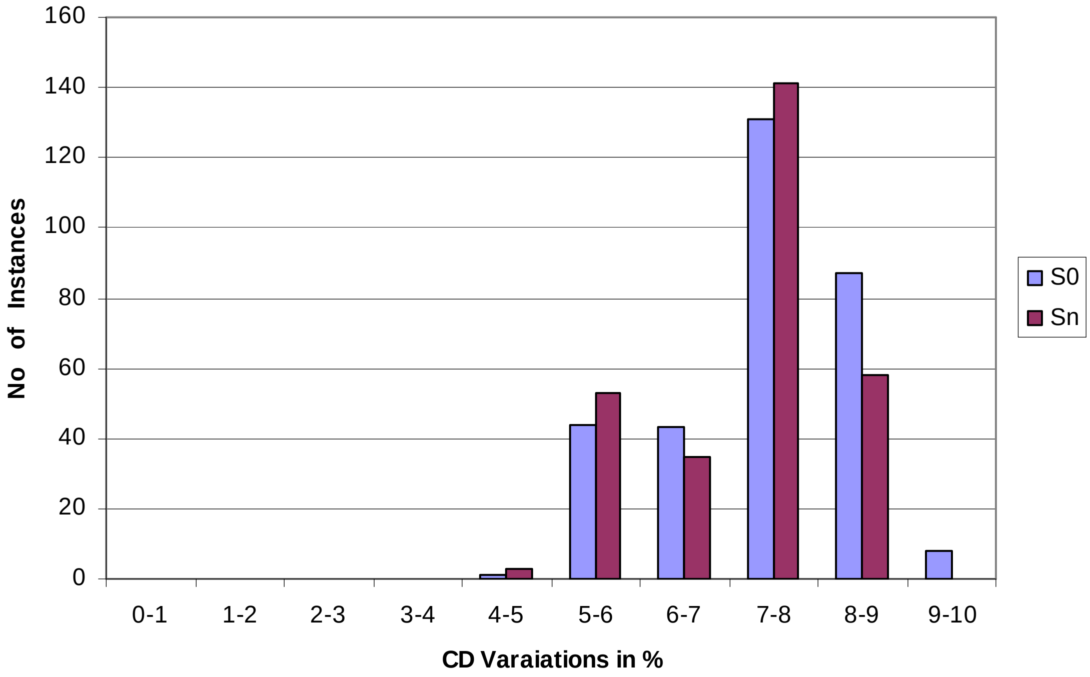


Figure 20. Estimated feature variations for 'S0' (initial solution) and 'Sn' (final solution), c432

As we can see from the above figure the number of instances having the highest CD variations (8-9 and 9-10) are lesser in 'Sn' as compared to the initial solution 'S0'.

This is a result of the series of perturbations that are performed in order to improve the environment of the polysilicon features with the highest CD variations.

Next, to verify the actual impact of the algorithm we perform whole-circuit simulations on both 'S0' and 'Sn' over the three subwindows. The calculated standard cell boundary feature variations for each of the subwindows are represented in histogram format as shown in Figures 21, 22 and 23 respectively.

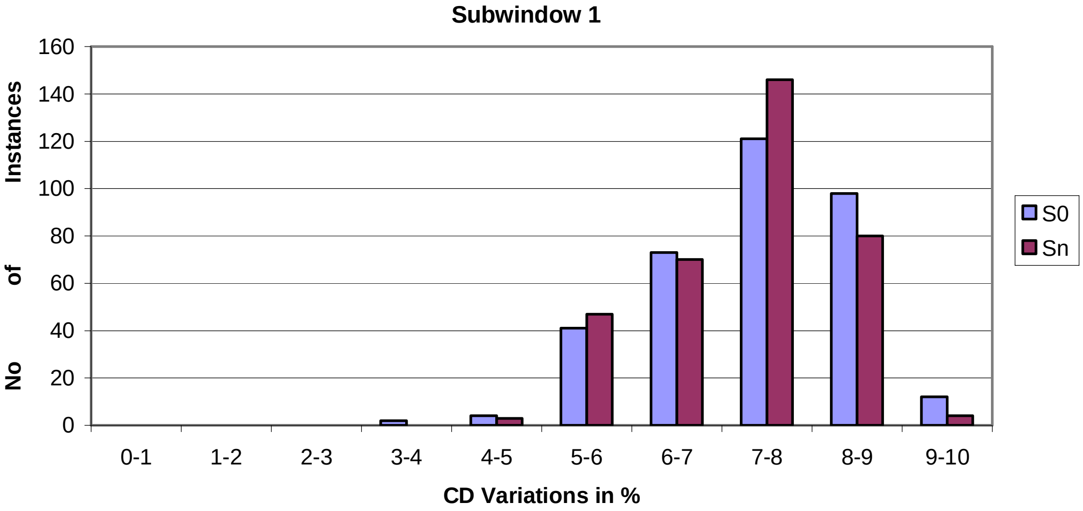


Figure 21. CD Variations in histogram format from c432 whole-circuit simulations, SW1

We observe a similar reduction in the number of instances with high CD variations in subwindow1 (Figure 21). Although, the improvement is not as high as anticipated from the estimated reduction in figure 20. This could be understood from following example. The boundary variations estimated from the LUT for a pair of 'NAND' cells placed adjacently are {7.1, 8.8} where the first value represents the feature variation for the standard cell on the left and the right value is for the standard cell on the right. However different values are calculated from the actual whole-circuit simulations when the same pair of cells happened to be placed adjacently. For instance, in c432 we obtain {8.1, 8.6}, {8.2, 8.6} as the boundary variations for the 'NAND' cells placed adjacently in different parts of the layout. In general, the boundary variations obtained from the whole-circuit simulations have a value slightly different from the estimated values and this value tends to be on the higher side. Even if the value is lesser than the estimated boundary feature variation, the difference is not much.

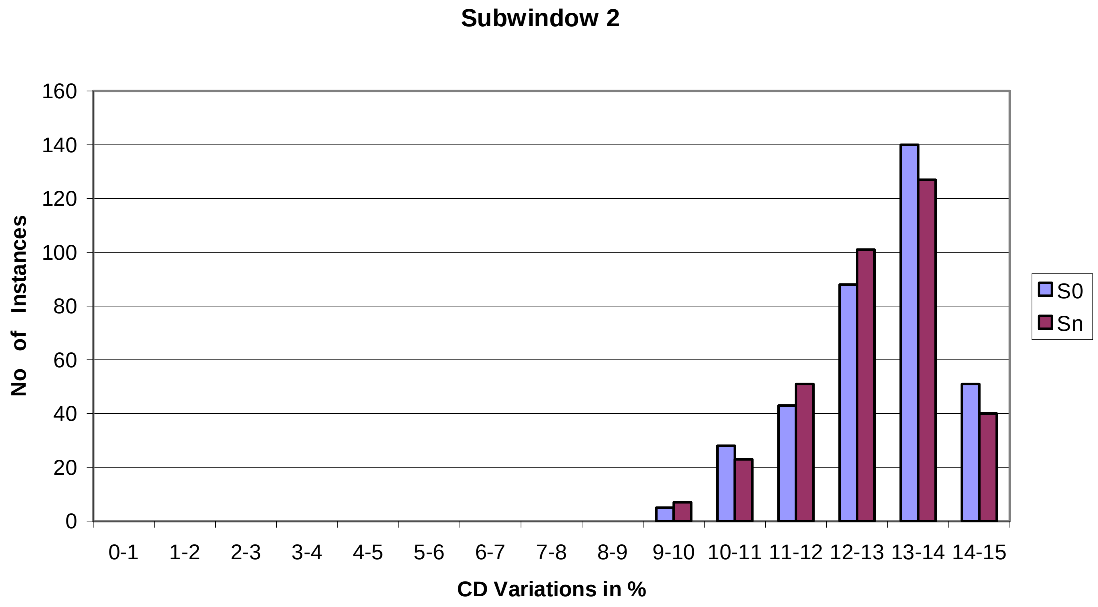


Figure 22. CD Variations in histogram format from c432 whole-circuit simulations, SW2

From Figure 22, a reduction in the number of instances with high CD variations is observed in subwindow2 as well. However, we do not observe any such improvement in subwindow3, as observed from Figure 23 below. This could be attributed to the large process window used and hence the variations produced are large irrespective of the relative placement of cells. From the results obtained it is essential that the process spends the least possible time outside subwindows 1 and 2. Hence, in the following test circuits results have been presented only for subwindows 1 and 2.

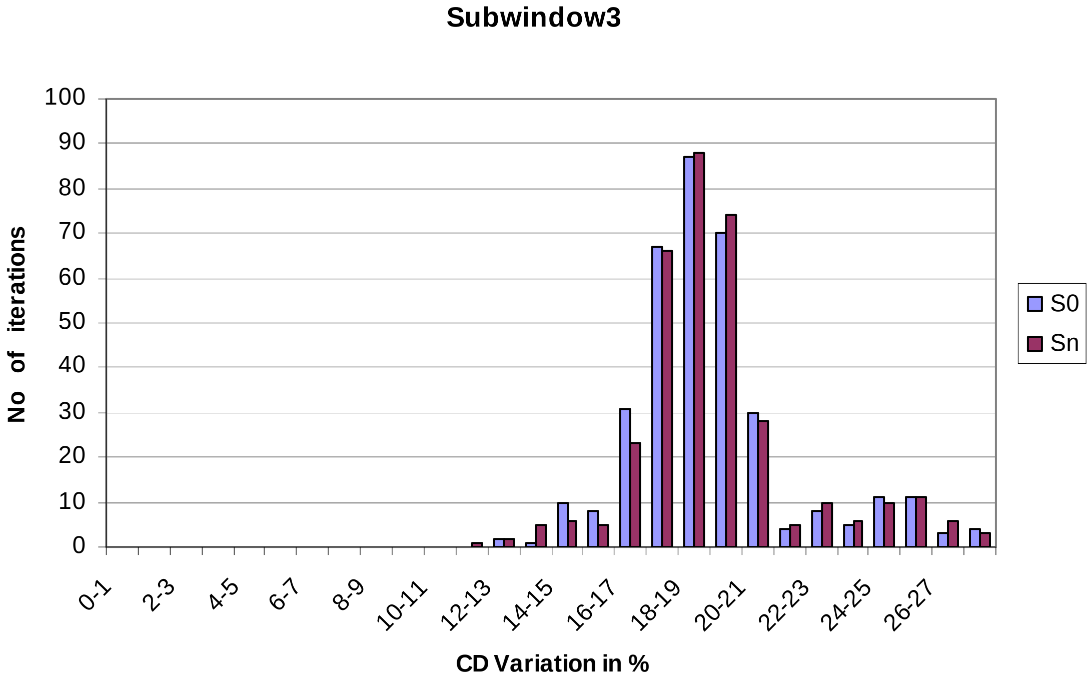


Figure 23. CD Variations in histogram format from c432 whole-circuit simulations, SW3

To measure the impact on routing parameters in the circuit we measure the percentage change in wirelength, number of vias between solutions ‘S0’ and ‘Sn’. This is tabulated in Table II.

Table II. Impact on wirelength and number of vias, c432

	S0	Sn	% change
Wirelength (um)	6263	6667	6.4
Vias	1263	1284	1.66

We then compare the routing congestion ratio totaled over both the horizontal and vertical directions (calculated by equation (4)) for the solutions ‘S0’ and ‘Sn’ in histogram format as shown in below in Figure 24.

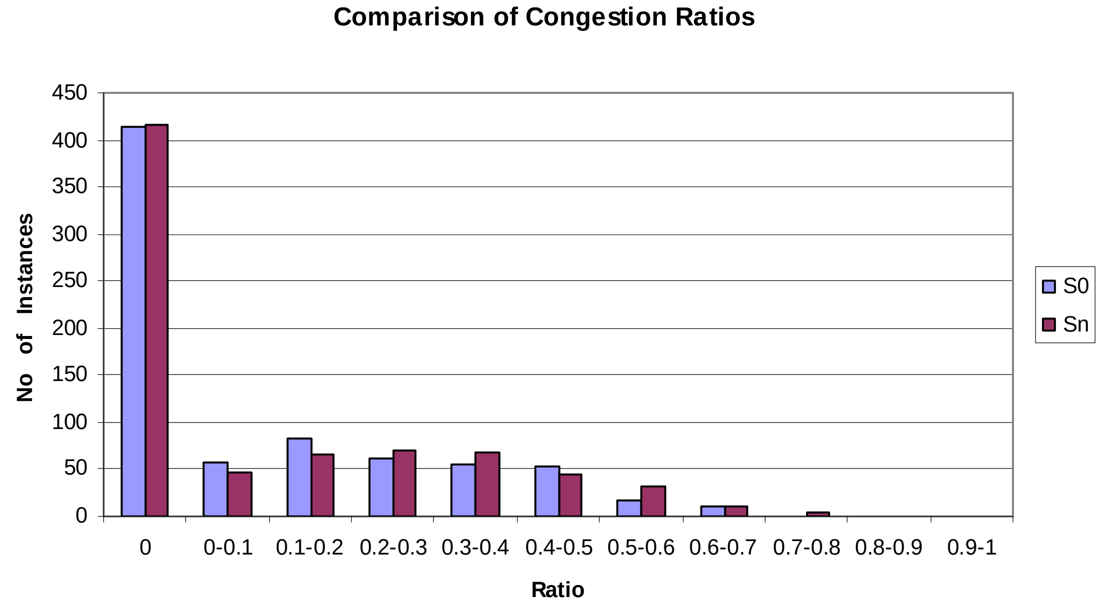


Figure 24. Comparison of congestion ratios, c432

The number of vias and the congestion are hardly affected, but the percentage change in wirelength could be considered to be on the higher side.

7.2. c880

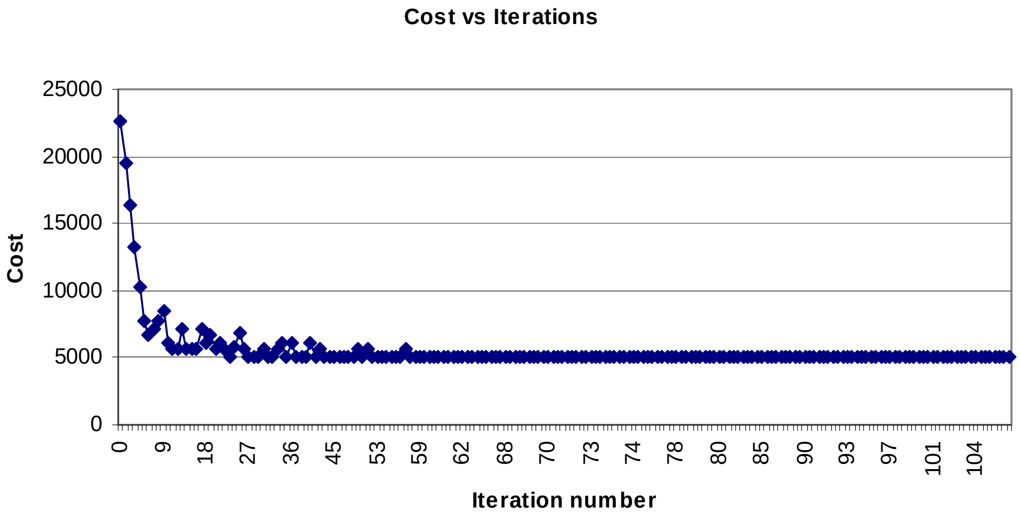


Figure 25. Variation of the cost function with iterations, c880

This ISCAS85 benchmark circuit is for an 8-bit ALU. The verilog netlist is synthesized to produce a circuit using 205 standard cells. The variation of the cost function, as the solutions iterate towards the final solution 'Sn' is shown in Figure 25.The estimated standard cell boundary feature variations for the initial and final solution 'S0' and 'Sn' are represented in histogram format in Figure 26.

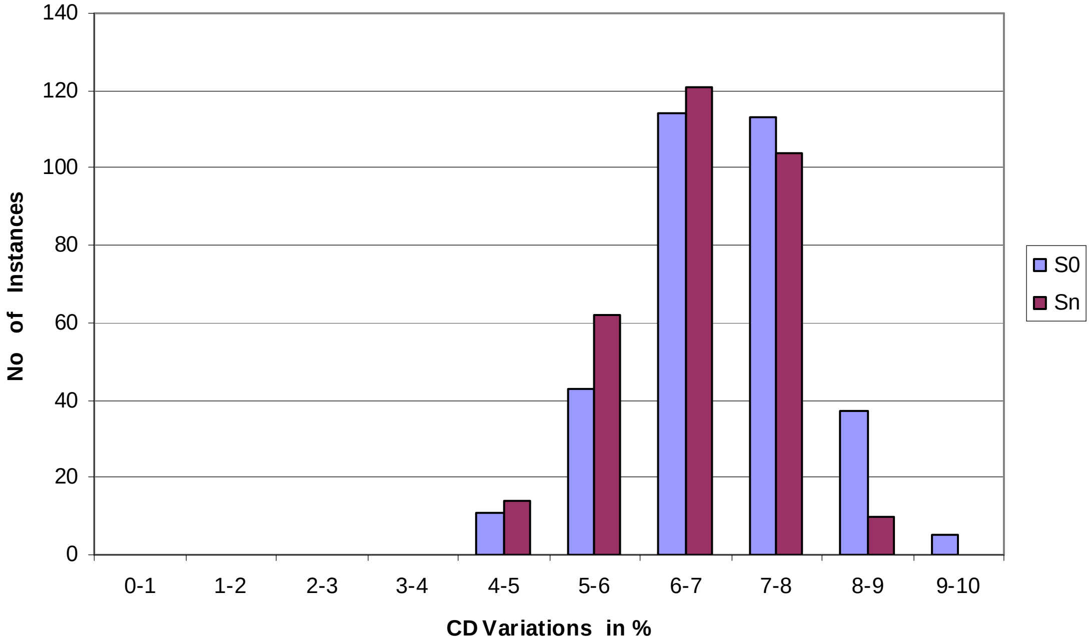


Figure 26. Estimated feature variations for 'S0' and 'Sn', c880

As in the case with c432, we observe a reduction in the number of instances with high CD variations from 'S0' to 'Sn'.

For verification, the results from the whole-circuit simulations on 'S0' and 'Sn' on subwindow1 and subwindow2 are shown in Figure 27 and Figure 28. The final solution generated 'Sn' shows a considerable reduction in the number of instances with high CD variations in both the subwindows.

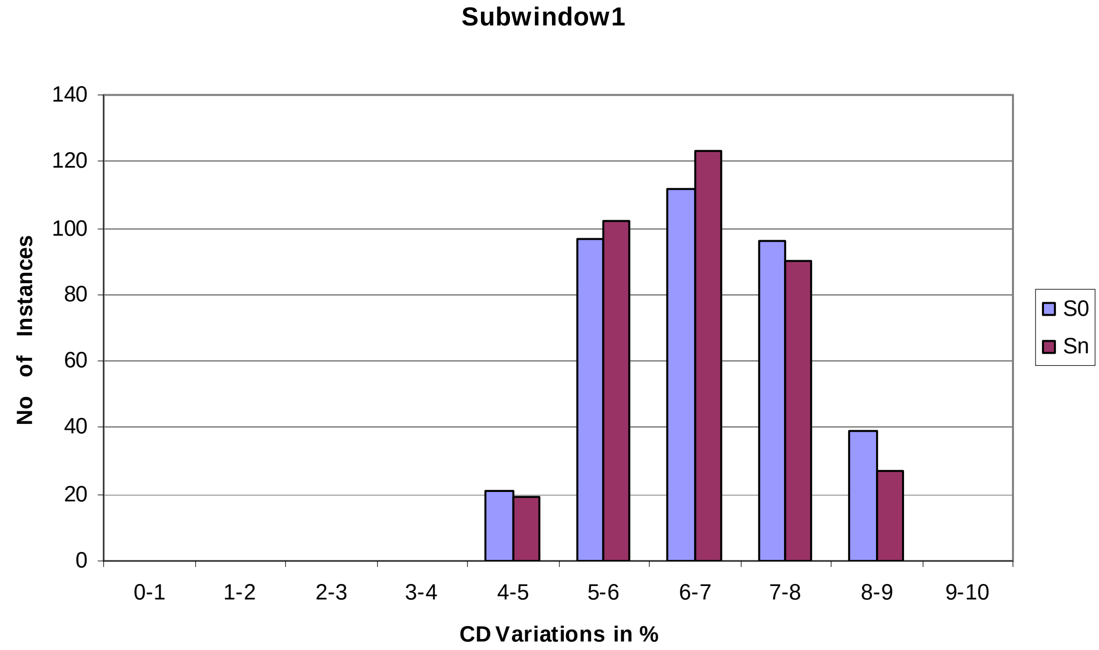


Figure 27.CD Variations histogram format from c880 whole-circuit simulations, SW1

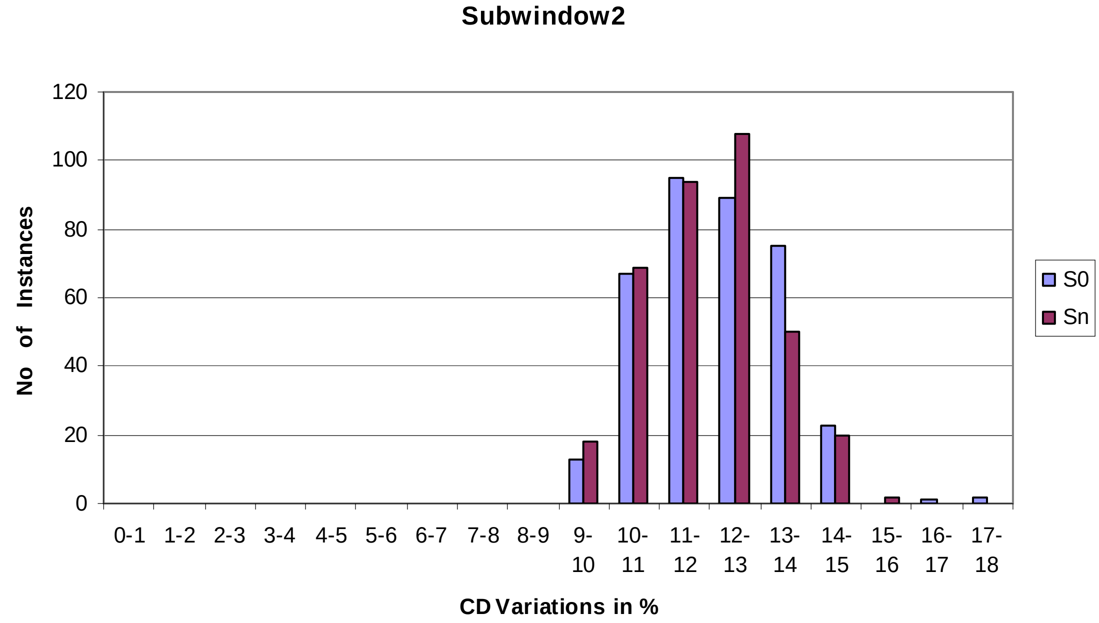


Figure 28.CD Variations histogram format from c880 whole-circuit simulations, SW2

The impact on routing parameters wirelength and the number of vias due to the iterations of perturbations are presented in Table III.

Table III. Impact on wirelength and number of vias, c880

	S0	Sn	% change
Wirelength (um)	9205	9952	8.1
Vias	1593	1717	7.7

The congestion ratios for both ‘S0’ and ‘Sn’ totaled over both the horizontal and vertical directions are compared in the histogram in Figure 29.

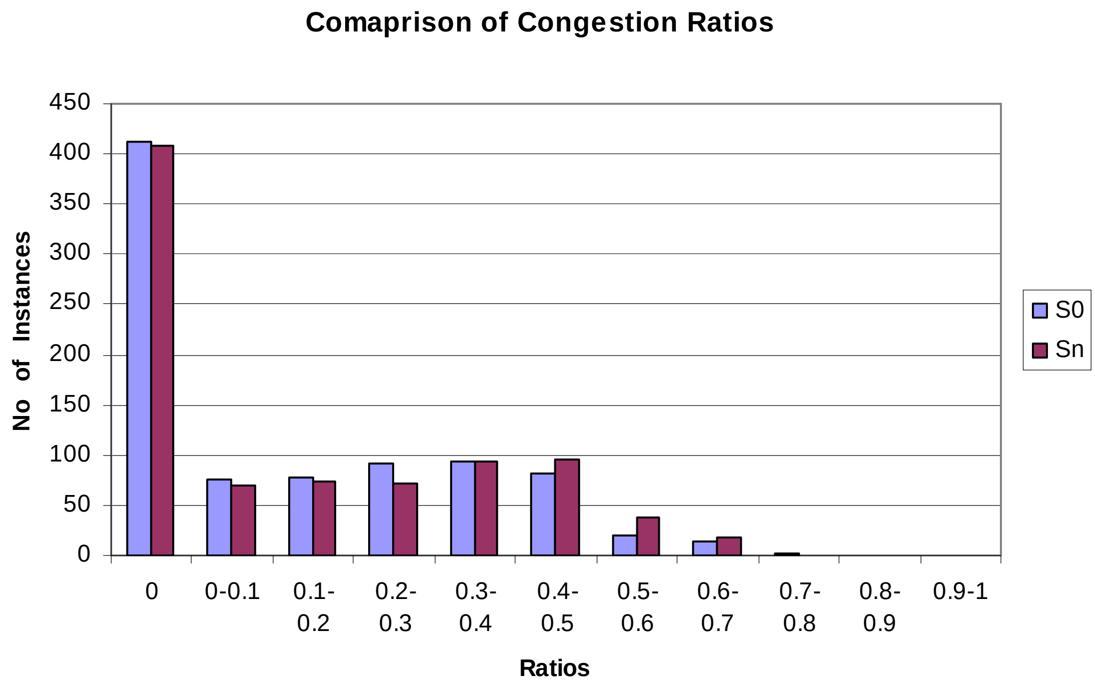


Figure 29. Comparison of congestion ratios, c880

The wirelength and the number of vias show a considerable increase in this case, but the congestion is by and large unaffected.

7.3. c3540

The c3540 is synthesized using 574 standard cells. The implementation of the algorithm using Simulated Annealing generates the solution ‘Sn’ from the initial solution ‘S0’ with a variation in cost function as shown in Figure 30.

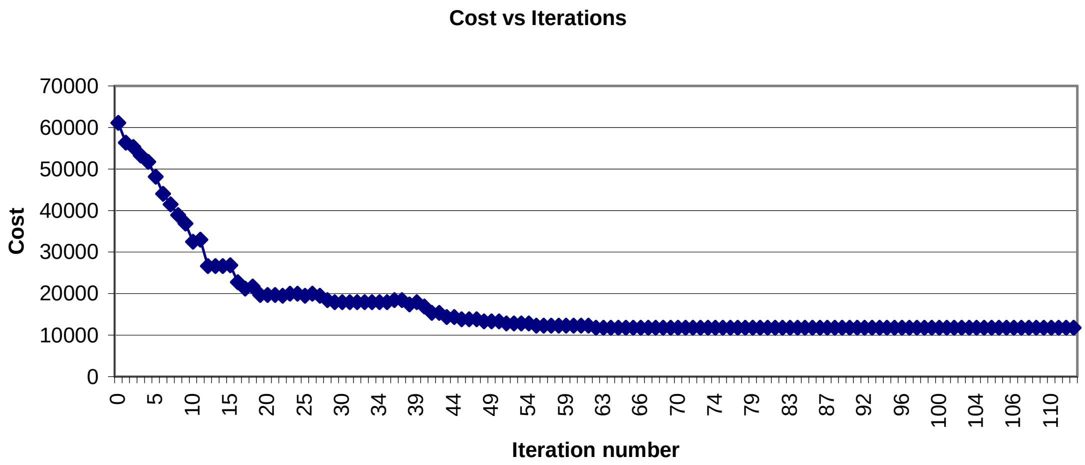


Figure 30. Variation of the cost function with iterations, c3540

The estimated standard cell boundary feature variations show a considerable reduction in the number of instances having high CD variations. This is represented in the histogram below in Figure 31, which compares the initial and final solutions.

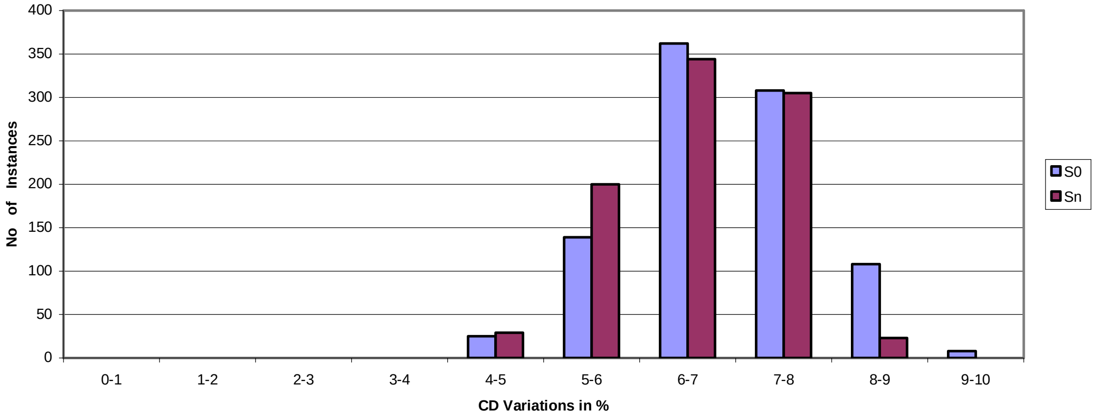


Figure 31. Estimated feature variations for ‘S0’ and ‘Sn’, c3540

We verify the actual impact of the algorithm with whole-circuit simulations on both ‘S0’ and ‘Sn’. Just as in the previous two circuits we present the results of the simulations in histogram format for both the subwindows 1 and 2 in Figures 32 and 33 respectively and observe an improvement in printability due to the reduction in the number of instances with high CD variations.

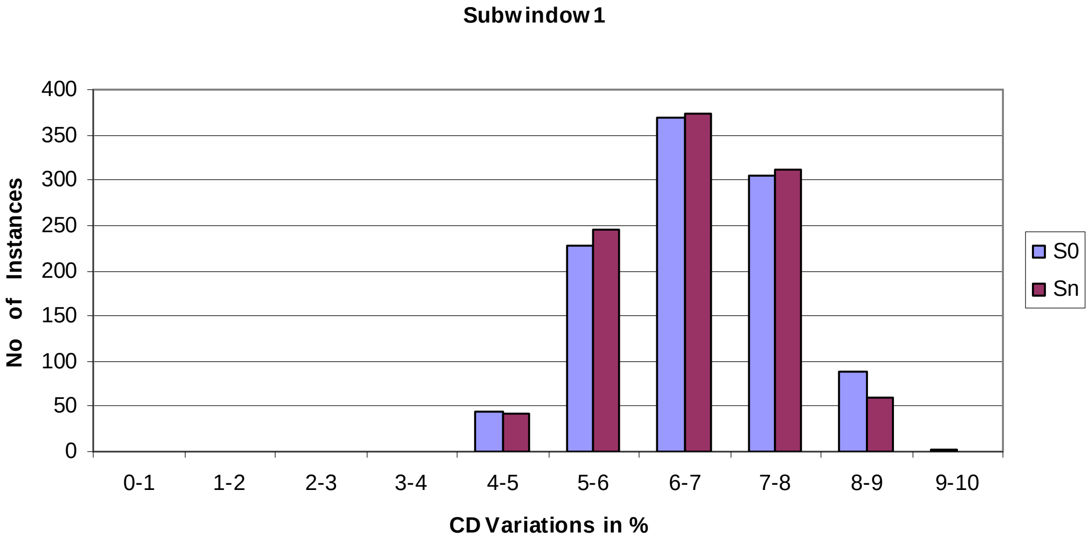


Figure 32. CD Variations histogram format from c3540 whole-circuit simulations, SW1

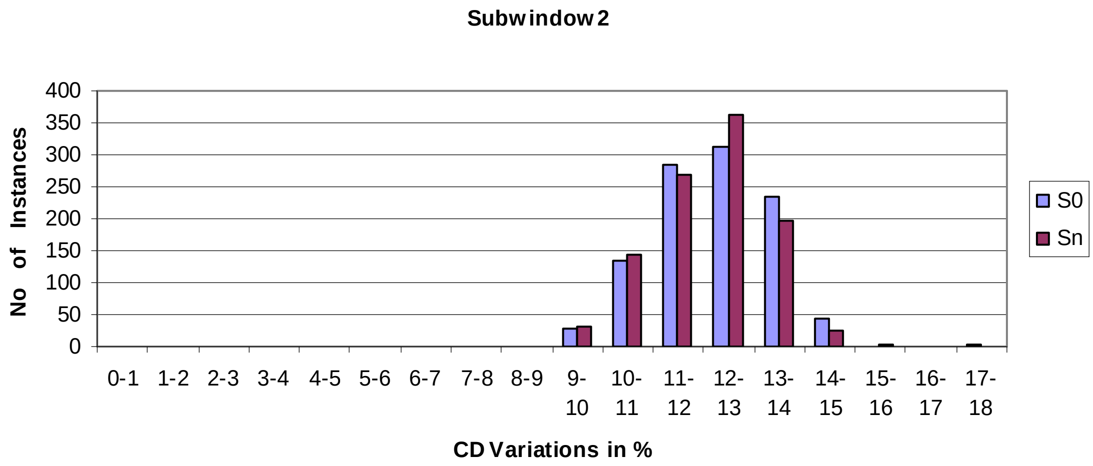


Figure 33. CD Variations histogram format from c3540 whole-circuit simulations, SW2

The changes in wirelength and number of vias are shown in Table IV.

Table IV. Impact on wirelength and number of vias, c3540

	S0	Sn	% change
Wirelength (um)	30076	32713	8.8
Vias	5104	5503	7.8

As in the case of c880, the increase in both the wirelength and the number of vias is high although the routing congestion shows only a small change. The routing congestion is compared for both ‘S0’ and ‘Sn’ as shown in the histogram in Figure 34.

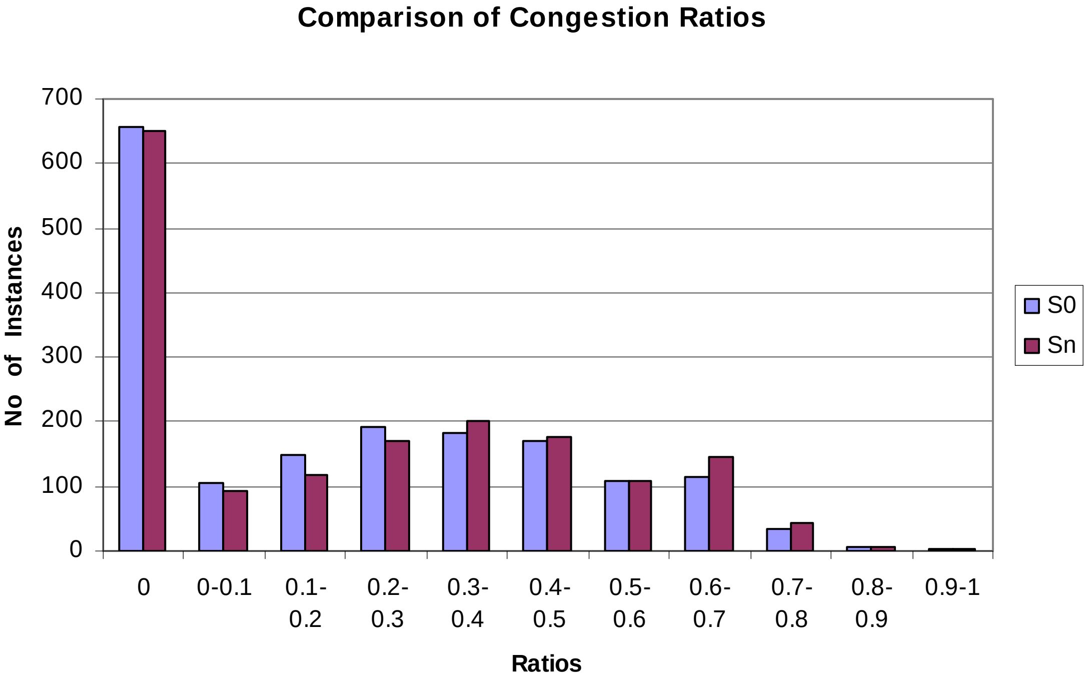


Figure 34. Comparison of Congestion Ratios, c3540

7.4. c5315

The c5315 is a 9-bit ALU and is synthesized using 705 standard cells. As in the previous test cases, we present the variation of the cost function on implementing the algorithm with simulated annealing, estimated CD variations in histogram format, the CD variations from whole-circuit simulations in subwindows1 and 2 and also the impact on the routing parameters in the Figures 35-39 and table V respectively.

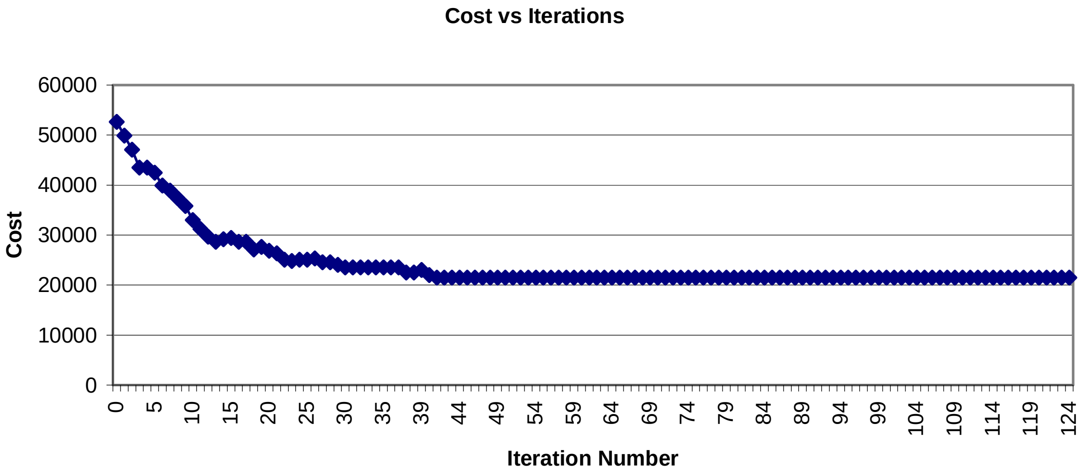


Figure 35. Variation of the cost function with iterations, c5315

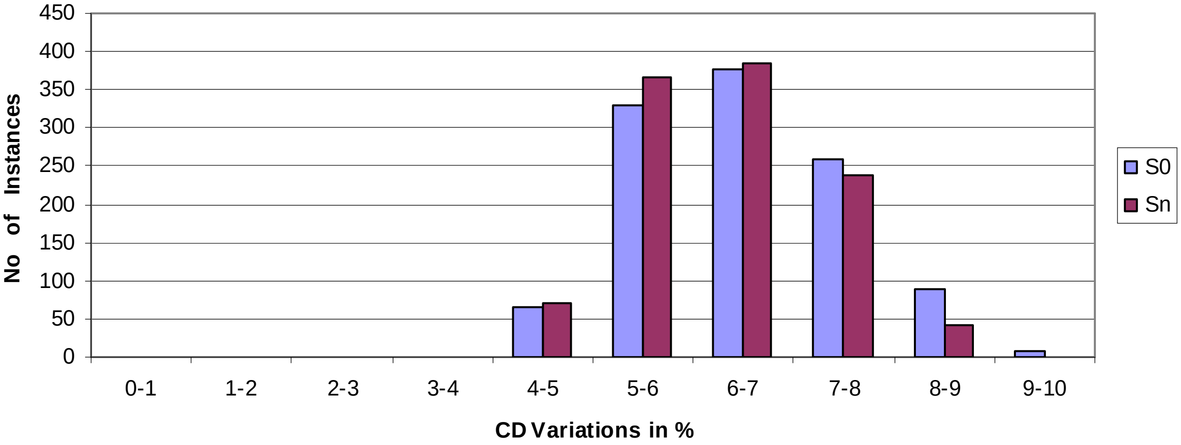


Figure 36. Estimated feature variations for ‘S0’ and ‘Sn’, c5315

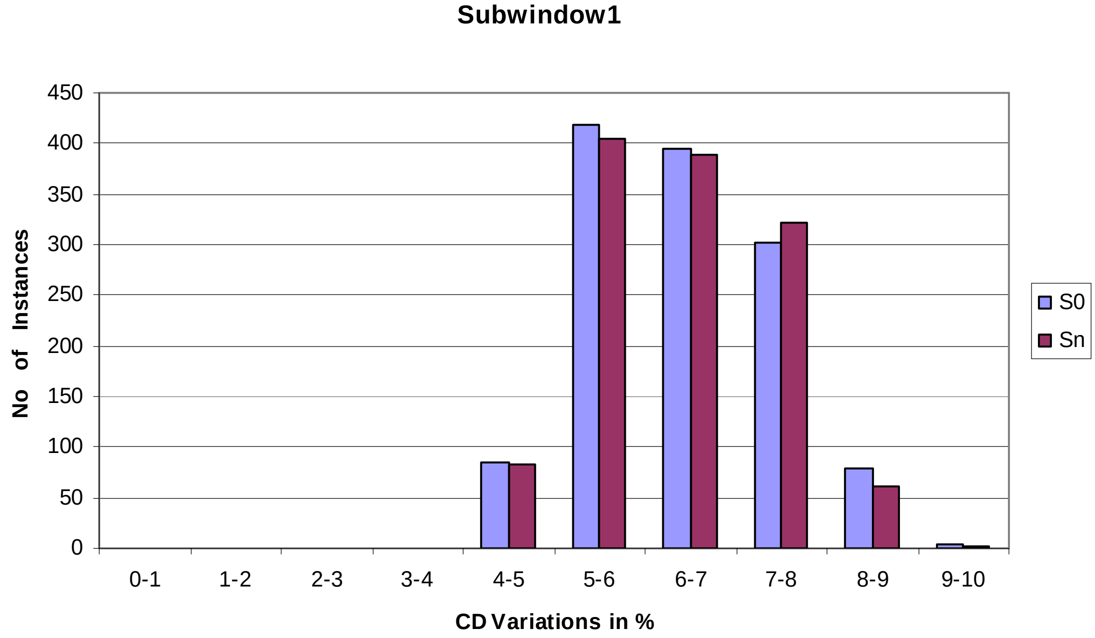


Figure 37. CD Variations histogram format from c5315 whole-circuit simulations, SW1

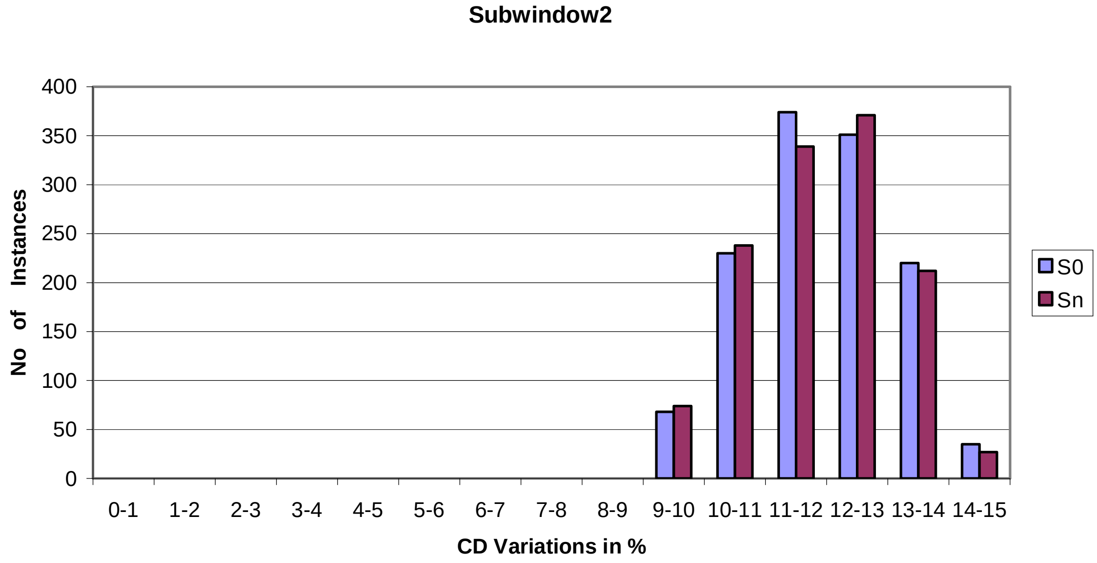


Figure 38. CD Variations histogram format from c5315 whole-circuit simulations, SW2

Table V. Impact on wirelength and number of vias, c5315

	S0	Sn	%change
Wirelength (um)	40251	41136	2.2
Vias	6251	6324	1.2

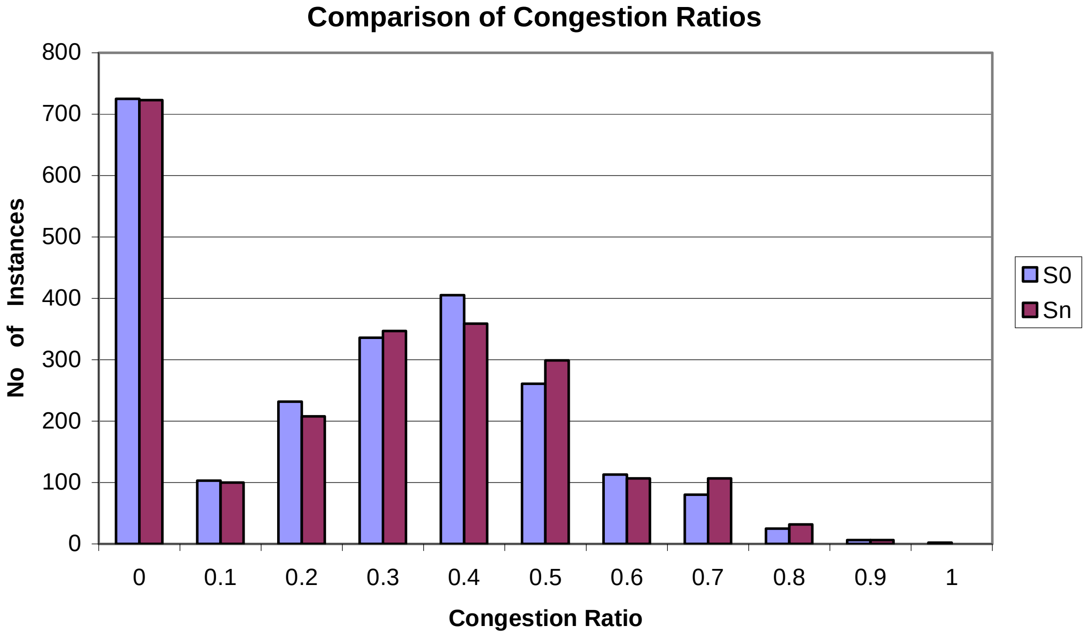


Figure 39. Comparison of Congestion Ratios, c5315

In this case, we observe an improvement in printability in both the subwindows and also the impact on the routing parameters is very small.

7.5. RESULTS SUMMARY

Table VI. Summary of Results

Circuit	No. of standard cells	Estimated Reduction % (SW1)	Actual Reduction in % SW1	Actual Reduction in % SW2	Wirelength increase in %	Increase in no. of vias in %
c432	194	42.1	20.91	12.56	6.4	1.66
c880	205	76.19	30.77	28.57	8.1	7.7
c3540	574	80.17	33.33	19.49	8.8	7.8
c5315	705	57.57	21.95	6.27	2.2	1.2

In table VI we represent the estimated results and those verified from the whole-circuit simulations. The reduction in percentage in columns 3, 4 and 5 represent the decrease in number of instances in the top two percentage points for the respective subwindows.

8. CONCLUSIONS AND FUTURE WORK

In this work, we intend to improve the printability of the circuit layout by trying to reduce CD variations in the design stage itself. We specifically try to reduce variations caused on standard cell boundary features in a dense circuit, by modifying the placement solution generated by a CAD tool. For this, we implement a series of perturbation iterations using simulated annealing which randomly chooses a neighboring solution at each iteration and based on the cost difference between the current and the neighboring solution uses the latter as the new intermediate solution. This procedure continues until a stable solution is reached. The above procedure was implemented on four ISCAS benchmark circuits and all of them showed a reduction in the number of instances with high CD variations on the boundaries of standard cells from 20-33% in subwindow1 and 6-28% in subwindow2. The wirelength and the number of vias showed an increase between 2.2 – 8.8% and 1.2 – 7.8% respectively due to the perturbations. The routing congestion was not greatly affected.

In this work the algorithm is implemented on a very dense circuit layout where the standard cells are placed at the closest possible distance. The assumption here is that if the distance between the standard cells is large then the neighboring cell will not affect the printability of a standard cell. However, this is usually not the case and the neighboring cell does affect the printability of a feature depending on the pitch between the boundary features and not the absolute distance [3]. Hence an extension to this work would be to create a LUT for all the possible pitches between standard cell boundary features and perturbations are made after matching the pitch between two standard cells to one of the LUT's and hence not restrict the perturbations to just one pitch (which is the least possible pitch). As mentioned earlier, the estimated variations for a circuit layout obtained from the LUT need not necessarily match with the actual CD variations calculated from whole-circuit simulations. Hence it would be good exercise to find the correlation between the estimated CD variation for a feature and its actual CD variation. Also since the LUT is calculated offline, it should be possible to include a 'lithography-friendliness' cost within a placement algorithm itself, which would provide a good quality solution not only in terms of wirelength and congestion but also produce a 'lithography-friendly' layout.

REFERENCES

[1] A.-K. Wong, “Resolution enhancement techniques in optical lithography,” *SPIE Press*, 2001.

[2] L.Liebmann, “Layout impact of resolution enhancement techniques: Impediment or opportunity?” in *Proceedings of International Symposium on Physical Design*, 2003, pp.110-117.

[3] C. Visweswariah, “Death, Taxes and Failing Chips,” in *Proceedings of ACM / IEEE Design Automation Conference*, 2003, pp. 343-347.

[4] D. Tsien, C.K Wang, Y. Ran, N. Verghese, “Context-specific leakage and delay analysis of a 65-nm standard cell library for lithography-induced variability,” *SPIE Press,* 2007.

[5] K. Cao, J.Hu, and S. Dobre, “Standard cell characterization considering lithography induced variations,” in *Proceedings of ACM / IEEE Design Automation Conference*, 2006, pp. 801-804.

[6] L.-D. Huang and D. Wong, “Optical proximity correction (opc)-friendly maze routing,” in *Proceedings of ACM / IEEE Design Automation Conference*, 2004, pp. 186-191.

[7] J.Mitra, P.Yu and D. Pan, “Radar: Ret-aware detailed routing using fast lithography simulations,” in *Proceedings of ACM / IEEE Design Automation Conference*, 2005, pp. 369-372.

[8] V. Khetarpal, T. Hersan, V. Rovner, D. Motiani, Y. Takagawa, L. Pileggi and A. Strojwas, “Design methodology for ic manufacturability based on regular logic

Bricks," in *Proceedings of ACM / IEEE Design Automation Conference*, 2005.

[9] L. Pileggi, H. Schmit, A. Strojwas, P. Golpalakrishnan, V. Khetarpal, A. Koorapaty, C. Patel, V. Rovner and K. Tong, "Exploring regular fabrics to optimize the performance-cost trade-off," in *Proceedings of ACM / IEEE Design Automation Conference*, 2003.

[10] P. Gupta, A. Kahng, and C.-H. Park, "Detailed placement for improved depth of focus and cd control," in *Proceedings of Asia and South Pacific Design Automation Conference*, 2005, pp. 343-348.

[11] S.Hu and J.Hu, "Pattern sensitive placement for manufacturability," *ACM International Symposium on Physical Design*, 2007,pp. 27-34.

[12] C.-W. Sham, E. Young, and C. Chu, "Optimal cell flipping in placement and floorplanning," in *Proceedings of ACM / IEEE Design Automation Conference*, 2006, pp. 1109 – 1114.

[13] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vechhi, "Optimization by simulated annealing," *Science*, 1983, pp. 671-680.

[14] J.E.Stine, J.Grad, I.Castellanos, J.Blank, V.Dave, M.Prakash, N.Iliev, and N.Jachimiec. (2008, Feb.)A framework for high-level synthesis of system-on-chip designs. [Online]. Available:http://vcag.ecen.okstate.edu/project/scells/

[15] Mentor Graphics Corporation, "Calibre Litho-Friendly Design User's Manual", Software version, 2006.

[16] J.P.Hayes.(2008, Jun.) ISCAS high level models. [Online]. Available:http://www.eecs.umich.edu/~jhayes/iscas.restore/benchmark.html.

[17] X. Lu, and W.Shi.(2008, Jun.) Layout and parasitic information for iscas circuits.[Online]. Available:http://dropzone.tamu.edu/~xiang/iscas.html.

APPENDIX

The Look-Up table generated for the parameters specified in section 6 for SW1 is presented below. The cells beginning with ‘f’ as prefix represent the flipped version of the standard cell.

Table	and2x1	fand2x1	and2x2	fand2x2	aoi21x1
and2x1	8.95211,7.24261	8.86106,9.06561	8.71101,7.89281	8.87966,6.46705	8.86106,7.63761
fand2x1	6.78105,7.70329	7.11025,8.42046	6.51865,7.65186	7.1386,5.60202	8.42945,7.77989
and2x2	5.60202,7.1386	6.46705,8.87966	5.46162,7.90772	5.68733,6.45776	5.70304,7.6328
fand2x2	7.65186,6.51865	7.89281,8.71101	7.46155,7.66772	7.98731,5.23093	7.86092,7.78502
aoi21x1	6.70566,6.60996	6.49546,8.97327	6.45457,5.54066	6.51854,5.99768	6.81979,7.15669
faoi21x1	7.77989,8.42945	7.63761,8.86106	7.78502,7.86092	7.6328,5.70304	7.75012,7.71324
aoi22x1	7.27644,8.24875	7.47313,5.90759	7.17293,7.37024	7.47377,5.50714	7.3274,7.57308
faoi22x1	7.8552,8.43426	7.72545,8.84182	6.45189,7.85083	7.71584,5.64629	7.73439,7.70259
bufx2	5.46262,6.67003	5.05989,8.95404	5.03032,7.36351	5.05636,5.83898	4.8781,7.76734
fbufx2	7.41211,6.68108	7.11116,8.95404	7.10909,7.33564	7.11052,5.69374	7.1219,7.7061
bufx4	7.22068,6.69887	6.67385,7.40035	7.18864,7.35198	6.66872,5.84827	6.94156,7.79063
fbufx4	6.87568,6.70944	6.51982,8.95404	8.0442,7.3664	6.51982,5.85789	6.41614,7.74374
dffposx1	7.055,6.66955	6.56516,8.9162	6.14104,7.35294	6.57478,5.83898	6.45189,7.76128
fdffposx1	5.89813,6.66619	5.79537,8.9162	-4.97872,7.35294	5.78543,5.83898	5.56325,7.77117
invx1	6.30573,6.7128	5.86847,8.9348	6.08151,7.45098	5.88771,5.84859	5.85857,7.75777
finvx1	6.21669,6.47587	6.62832,8.93672	5.97197,5.26865	6.62832,5.92266	6.58185,7.69589
invx2	5.85775,6.69502	-5.53985,8.9348	-5.29854,7.40677	-5.53985,5.83898	-5.68983,7.76702
finvx2	-5.85263,7.95079	-5.55876,8.9348	-6.03689,7.06747	-5.55876,5.86623	-5.73758,7.75745
invx4	5.95421,6.67003	5.82968,7.40035	5.8874,7.3491	5.82808,5.84859	5.76049,7.78106
finvx4	5.55796,8.03681	6.11614,8.95404	5.47672,7.02422	6.10765,5.85661	6.00517,7.76734
invx8	6.83276,6.69887	6.52617,7.40035	6.79463,7.35006	6.53318,5.83898	6.25921,7.78106
finvx8	6.84198,8.06565	6.39682,8.95404	6.80889,7.0396	6.39865,5.87585	6.81458,7.76734
latch	5.45493,6.72386	5.23366,8.95404	5.32803,7.43849	5.23174,5.87713	4.97552,7.79573
flatch	4.72235,6.75269	-4.71748,8.95404	4.97552,7.40965	-4.7271,5.85789	-4.91783,7.79191
mux2x1	-5.7407,6.69502	6.37824,8.95981	-7.41595,7.42647	6.36574,5.70817	-5.19712,7.72651
fmux2x1	6.35542,6.83199	5.91336,8.77193	6.36562,5.53777	5.91336,-5.48534	6.29484,6.87003
nand2x1	7.57785,7.08141	7.67255,8.86106	7.4481,7.9431	7.67159,5.01565	7.57785,7.26271
fnand2x1	8.73078,8.03681	9.25958,8.9348	8.34198,7.11842	9.24563,5.85661	8.42128,7.77596
nand3x1	8.80552,7.05402	7.80507,8.87004	8.64249,7.82584	7.76653,5.65943	8.73093,7.67739

fnand3x1	7.24091,6.6758	8.48992,8.93224	7.73425,7.38322	8.48992,5.48791	-5.33676,7.51821
nor2x1	6.89138,8.12812	7.29967,7.07333	7.31666,7.32266	7.28624,4.96339	7.24629,7.13128
fnor2x1	-6.60145,6.47636	-6.22615,8.91813	5.32907,5.24077	-6.27308,5.92201	-6.22616,7.69621
nor3x1	5.77705,5.84967	5.99161,8.87453	5.4514,7.89072	5.98585,5.65847	5.7482,5.88819
fnor3x1	6.04927,7.02807	6.03318,8.86106	5.82521,7.83593	6.02438,5.62994	5.99792,7.40465
oai21x1	8.2079,6.76711	7.50597,8.59495	7.11313,7.36832	7.53224,-5.0102	6.7772,7.26143
foai21x1	7.66741,6.75942	7.59657,8.95404	7.93596,7.438	7.57797,5.77806	6.87843,7.60976
oai22x1	7.34534,6.60083	6.75849,7.40035	7.35368,7.06171	6.71905,5.49881	6.73422,7.53863
foai22x1	7.93051,6.75942	6.8473,8.9162	7.85104,7.40965	6.83896,5.77806	7.24921,7.60019
or2x1	7.96113,6.53979	7.48625,6.98548	7.85991,7.23712	7.50156,5.04162	8.47171,6.68853
for2x1	-8.64018,6.49029	-7.24646,8.93672	5.28985,5.26913	-7.24646,5.94125	-6.85562,7.70546
or2x2	-5.04313,6.78008	-5.05061,8.58918	6.11733,7.2501	-5.05061,5.38531	5.03352,7.54278
for2x2	7.0771,6.44752	7.30494,7.40228	6.66797,5.24077	7.2707,5.90278	7.54468,7.723
xnor2x1	6.25,6.76711	6.48821,8.95404	6.21443,7.39523	6.49479,5.86751	6.25849,7.81073
fxnor2x1	4.95373,6.75269	4.78929,7.40035	5.10979,7.39523	4.78416,5.78095	-4.69094,7.78297
xor2x1	6.19216,6.76711	6.39203,8.95404	6.16572,7.39523	6.39059,5.86751	6.19168,7.80147
fxor2x1	4.69159,6.70944	-4.76685,8.95404	4.93706,7.39523	-4.76685,5.86751	-4.93578,7.79605

Table	faoi21x1	aoi22x1	faoi22x1	bufx2	fbufx2
and2x1	8.97327,6.49546	8.84182,7.72545	5.90759,7.47313	8.95404,7.11116	8.95404,5.05989
fand2x1	6.60996,6.70566	8.43426,7.8552	8.24875,7.27644	6.68108,7.41211	6.67003,5.46262
and2x2	5.99768,6.51854	5.64629,7.71584	5.50714,7.47377	5.69374,7.11052	5.83898,5.05636
fand2x2	5.54066,6.45457	7.85083,6.45189	7.37024,7.17293	7.33564,7.10909	7.36351,5.03032
aoi21x1	6.93329,6.51701	6.81594,7.25351	7.49231,6.95855	7.79394,7.29648	7.08333,5.12337
faoi21x1	7.1635,6.38074	7.70259,7.73439	7.57308,7.3274	7.7061,7.1219	7.76734,4.8781
aoi22x1	6.95855,7.49231	-5.19738,7.52026	5.37447,7.02684	5.46712,7.97938	5.34443,5.75017
faoi22x1	7.25351,6.81594	7.69401,7.65139	7.49244,7.39438	7.78694,7.18596	7.81963,4.95565
bufx2	5.12337,7.08333	4.95565,7.81963	5.75017,5.34443	5.60653,7.63825	5.69182,5.64308
fbufx2	7.29648,7.79394	7.18596,7.78694	7.97938,5.46712	7.62159,7.62672	7.62159,5.54754
bufx4	7.02893,7.82856	7.17662,7.81707	6.8308,-5.45605	7.16669,7.62736	7.42023,5.72067
fbufx4	6.06057,7.11373	6.53145,7.81578	6.98079,5.46143	6.90686,7.62159	6.94359,5.7133
dffposx1	6.74983,7.07949	6.51118,7.80617	6.98196,5.41937	7.03289,7.65555	7.12078,5.72163
fdffposx1	5.90214,7.06025	6.09778,7.81546	-4.68638,5.47091	-5.44948,7.63633	6.03359,5.70207
invx1	6.09547,7.0791	5.92136,7.78694	6.29296,5.49463	6.27434,7.64017	6.31605,5.70175
finvx1	6.08585,6.84903	6.67794,7.05148	6.85531,-5.72737	6.20195,7.33843	5.71201,5.23623
invx2	-6.10348,7.07179	-5.67156,7.79656	5.5453,5.47566	5.86801,7.64017	5.87873,5.69214

finvx2	-5.90823,7.0791	-5.71034,7.78438	-6.12111,5.36719	-5.91319,7.64017	-5.86238,5.45552
invx4	5.6099,7.80548	5.46695,7.81707	5.44337,-5.50665	6.00052,7.62736	5.75177,5.70175
finvx4	5.81012,7.09064	6.06093,7.79399	5.03136,5.33968	5.56405,7.62736	5.68236,5.43661
invx8	6.44689,7.80548	6.24872,7.81707	6.88541,-5.48957	6.77804,7.62736	6.46361,5.70175
finvx8	6.32908,7.09064	6.82051,7.79399	6.8718,5.33968	6.7758,7.62736	6.14922,5.43661
latch	5.14004,7.11334	5.1742,7.81963	5.16633,5.38712	5.37834,7.65362	5.57864,5.79858
flatch	-4.65047,7.10219	-4.82554,7.80617	4.70626,5.48989	4.71274,7.65362	4.76493,5.74151
mux2x1	-5.87008,7.74508	5.68304,7.6892	6.15702,5.40735	5.61449,7.64017	5.87328,5.68765
fmux2x1	6.28174,6.6936	5.65485,7.29022	6.45171,-5.74634	6.3162,7.46336	5.28077,5.16473
nand2x1	7.06564,6.71591	7.73693,7.32676	7.92503,-5.1256	7.6312,7.21607	7.8149,4.94383
fnand2x1	7.76056,7.09064	8.60102,7.76547	8.85561,5.38111	8.90186,7.67348	8.88591,5.452
nand3x1	7.76119,6.79594	8.85244,7.73022	8.3745,-5.1977	8.68171,7.77919	7.92984,5.6745
fnand3x1	7.12335,7.35457	-5.26755,7.53602	7.9309,5.27581	6.99559,7.65298	7.90666,5.44366
nor2x1	6.54691,7.19067	7.27667,7.78085	6.61704,-4.96148	7.0871,7.60622	6.94823,5.19904
fnor2x1	-5.74254,6.87211	-5.88009,7.05148	-6.89481,-5.83899	-6.5903,7.31921	-5.41667,5.21699
nor3x1	5.86106,7.24685	5.8933,6.07759	6.35535,-5.10853	5.86939,7.73114	6.013,5.03809
fnor3x1	5.9598,7.23646	6.10035,7.53922	5.7212,5.1139	6.01797,7.73434	6.0645,5.15864
oai21x1	6.64056,6.64512	7.56536,8.07055	7.0137,5.43044	6.90792,7.56522	7.17881,5.23462
foai21x1	7.13393,7.09026	7.02873,7.77252	7.85915,5.42032	7.62511,7.62159	7.78669,5.63122
oai22x1	6.2625,7.45999	6.89157,7.50942	7.16118,5.22521	7.304,7.64594	7.46993,5.22981
foai22x1	6.37087,7.09026	7.36233,7.77252	7.01338,5.41083	7.90776,7.62159	8.12109,5.6216
or2x1	7.50412,6.50739	8.43583,8.30898	7.81587,5.22584	8.49285,7.27885	7.37774,5.29073
for2x1	-7.35226,6.87211	-6.65721,7.05148	-6.8094,5.26	-8.62172,7.33843	-6.93983,5.21699
or2x2	-4.46628,6.96291	5.00949,8.20034	-4.63059,5.29162	4.5743,7.64465	-4.79116,4.90215
for2x2	7.41228,6.86019	7.54853,7.04058	6.54932,5.31312	7.06211,7.28078	6.98907,5.17884
xnor2x1	6.12945,7.11373	6.05404,7.81578	6.08489,5.49558	6.21619,7.64465	6.27549,5.76075
fxnor2x1	4.7319,7.11373	4.74286,7.80617	4.82905,5.51487	4.97071,7.65362	5.06694,5.70753
xor2x1	6.03824,7.11373	6.00356,7.81578	6.00569,5.49558	6.10211,7.64465	6.18139,5.76075
fxor2x1	-4.52703,7.10219	-4.86528,7.80617	4.57481,5.49937	4.68422,7.65362	4.94607,5.75113

Table	bufx4	fbufx4	dffposx1	fdffposx1	invx1
and2x1	8.95404,6.51982	7.40035,6.67385	8.9162,5.79537	8.9162,6.56516	8.93672,6.62832
fand2x1	6.70944,6.87568	6.69887,7.22068	6.66619,5.89813	6.66955,7.055	6.47587,6.21669

and2x2	5.85789,6.51982	5.84827,6.66872	5.83898,5.78543	5.83898,6.57478	5.92266,6.62832
fand2x2	7.3664,8.0442	7.35198,7.18864	7.35294,-4.97872	7.35294,6.14104	5.26865,5.97197
aoi21x1	7.11373,6.06057	7.82856,7.02893	7.06025,5.90214	7.07949,6.74983	6.84903,6.08585
faoi21x1	7.74374,6.41614	7.79063,6.94156	7.77117,5.56325	7.76128,6.45189	7.69589,6.58185
aoi22x1	5.46143,6.98079	-5.45605,6.8308	5.47091,-4.68638	5.41937,6.98196	-5.72737,6.85531
faoi22x1	7.81578,6.53145	7.81707,7.17662	7.81546,6.09778	7.80617,6.51118	7.05148,6.67794
bufx2	5.7133,6.94359	5.72067,7.42023	5.70207,6.03359	5.72163,7.12078	5.23623,5.71201
fbufx2	7.62159,6.90686	7.62736,7.16669	7.63633,-5.44948	7.65555,7.03289	7.33843,6.20195
bufx4	7.38064,5.42642	7.3707,7.22338	7.33078,5.42443	7.34104,6.99959	7.11613,5.58698
fbufx4	6.98246,6.91241	5.78106,7.24055	6.92565,5.88499	6.92565,7.07007	6.58953,6.23655
dffposx1	7.07007,6.92565	6.99959,7.34104	7.21344,5.9358	7.25127,7.16919	6.90565,5.72933
fdffposx1	5.88499,6.92565	5.42443,7.33078	6.03849,5.38795	6.04266,7.03257	6.31185,6.22373
invx1	6.28203,6.99527	6.25321,7.32341	6.31791,6.04394	6.3667,7.24197	6.67,6.29745
finvx1	6.23655,6.58953	5.58698,7.11613	6.22373,6.31185	5.72933,6.90565	6.72471,6.69588
invx2	5.85455,6.96964	5.77646,7.30898	5.86,6.00581	5.88514,7.23235	-6.03048,6.33008
finvx2	-5.88627,6.31022	-5.88194,7.25977	-5.87185,6.62622	-5.8316,6.96977	-5.85935,6.3493
invx4	6.03962,5.6414	5.65559,7.35162	5.94828,5.97793	5.74343,7.15925	5.92361,5.74518
finvx4	5.59545,-5.11651	5.61102,7.3319	5.54786,-4.50379	5.662,6.95054	5.76145,5.76376
invx8	6.84558,5.80668	6.51638,7.37102	6.79943,-5.17388	6.50365,7.15925	6.50733,5.74518
finvx8	6.85055,5.12163	6.56295,7.34136	6.79823,-4.49418	6.54114,6.96015	6.57239,5.7785
latch	5.41391,6.97007	5.25418,7.36637	5.34693,6.06157	5.62673,7.26057	5.28957,5.78362
flatch	4.73453,6.97007	4.74569,7.36637	4.71274,6.06157	5.19102,7.25127	-4.58231,5.78362
mux2x1	5.47388,7.00723	6.37824,7.36797	5.53057,-5.27451	6.83672,7.14322	6.13244,6.38645
fmux2x1	6.35542,7.54023	5.45684,6.74182	6.29356,-5.26329	5.33996,4.92203	6.22087,6.90601

nand2x1	7.56151,7.77727	7.64322,5.82086	7.59227,-5.29053	7.81154,6.26763	7.0521,6.6773
fnand2x1	8.7327,6.31107	8.65266,7.33319	8.73174,6.62526	8.76327,6.96015	8.0892,6.35763
nand3x1	8.77361,6.40418	7.74135,7.42552	8.63672,6.92682	7.79477,7.21119	8.40255,6.6158
fnand3x1	7.36511,6.94573	6.31502,7.31972	7.51161,-5.47576	7.93629,7.02043	7.38549,6.31791
nor2x1	7.16093,6.13639	6.83313,6.87343	7.09979,-5.61965	6.916,6.69597	7.23245,6.22053
fnor2x1	-6.63913,6.57672	-6.75939,7.1639	-5.86317,-4.94091	-5.75677,6.85788	-5.49749,6.70549
nor3x1	5.83366,7.10759	6.00395,7.4967	5.79684,6.6118	6.02891,5.0538	5.91885,6.49344
fnor3x1	6.02483,5.84299	6.00047,5.94638	6.00506,6.50028	6.05461,5.11888	6.05422,6.53508
oai21x1	6.91273,7.33566	7.23039,6.61598	7.68935,-4.75279	7.20126,4.84572	6.53739,6.21541
foai21x1	7.6594,6.98246	7.93193,7.38032	7.57736,-5.26457	8.06851,7.16695	6.78614,5.74518
oai22x1	7.27997,6.30979	7.22466,7.30032	7.30497,-4.72588	7.30834,6.76394	6.55156,6.36788
foai22x1	7.93756,6.96964	7.12303,7.37102	7.91929,-5.26457	7.39971,7.14803	7.08737,5.74518
or2x1	8.48132,7.03157	7.1703,6.91751	7.6722,4.81016	7.38923,6.19518	7.25131,5.89637
for2x1	-8.7167,6.57672	-6.78593,7.17785	-7.09364,-4.93129	-7.04101,6.8768	-7.2363,6.72407
or2x2	4.6464,-5.07807	-4.75988,6.95182	5.12902,5.81994	-4.83328,7.05377	4.87778,6.22309
for2x2	7.05211,6.05012	6.95368,7.15493	7.01673,-5.04602	6.45699,6.79183	7.62889,6.05269
xnor2x1	6.23846,5.89424	6.18283,7.38064	6.20001,-5.16715	6.38321,7.22274	6.10467,5.78362
fxnor2x1	4.93866,6.97007	4.92972,7.36156	4.92713,-5.25208	5.29169,7.22883	4.86944,5.78362
xor2x1	6.11685,5.89424	6.09739,7.38064	6.13736,-5.17549	6.29088,7.22274	6.0284,5.78362
fxor2x1	4.69287,6.97007	4.6758,7.37118	4.67236,6.06157	4.95345,7.25127	4.85374,5.78362

Table	finvx1	invx2	finvx2	invx4	finvx4
and2x1	8.9348,5.86847	8.9348,-5.55876	8.9348,-5.53985	8.95404,6.11614	7.40035,5.82968
fand2x1	6.7128,6.30573	7.95079,-5.85263	6.69502,5.85775	8.03681,5.55796	6.67003,5.95421

and2x2	5.84859,5.88771	5.86623,-5.55876	5.83898,-5.53985	5.85661,6.10765	5.84859,5.82808
fand2x2	7.45098,6.08151	7.06747,-6.03689	7.40677,-5.29854	7.02422,5.47672	7.3491,5.8874
aoi21x1	7.0791,6.09547	7.0791,-5.90823	7.07179,-6.10348	7.09064,5.81012	7.80548,5.6099
faoi21x1	7.75777,5.85857	7.75745,-5.73758	7.76702,-5.68983	7.76734,6.00517	7.78106,5.76049
aoi22x1	5.49463,6.29296	5.36719,-6.12111	5.47566,5.5453	5.33968,5.03136	-5.50665,5.44337
faoi22x1	7.78694,5.92136	7.78438,-5.71034	7.79656,-5.67156	7.79399,6.06093	7.81707,5.46695
bufx2	5.70175,6.3160	5.45552,-5.86238	5.69214,5.87873	5.43661,5.68236	5.70175,5.75177
fbufx2	7.64017,6.27434	7.64017,-5.91319	7.64017,5.86801	7.62736,5.56405	7.62736,6.00052
bufx4	7.32341,6.25321	7.25977,-5.88194	7.30898,5.77646	7.3319,5.61102	7.35162,5.65559
fbufx4	6.99527,6.28203	6.31022,-5.88627	6.96964,5.85455	-5.11651,5.59545	5.6414,6.03962
dffposx1	7.24197,6.3667	6.96977,-5.8316	7.23235,5.88514	6.95054,5.662	7.15925,5.74343
fdffposx1	6.04394,6.31791	6.62622,-5.87185	6.00581,5.86	-4.50379,5.54786	5.97793,5.94828
invx1	6.29937,6.39363	6.29937,-5.80082	6.28014,5.9451	6.29937,5.68412	6.28014,5.77726
finvx1	6.3871,6.68883	6.3493,-5.85935	6.33008,-6.03048	5.76376,5.76145	5.74518,5.92361
invx2	5.9451,6.28014	5.69182,-5.81044	5.872,5.90727	5.69182,5.68412	5.872,5.77245
finvx2	-5.80082,6.29937	-5.9289,-5.96799	-5.79815,5.71547	-5.93947,5.53969	-5.8507,6.01045
invx4	5.77726,6.28014	6.01045,-5.8507	5.77245,5.872	6.1006,5.71586	6.12753,5.7803
finvx4	5.68412,6.29937	5.53969,-5.93947	5.68412,5.69182	5.5977,5.52159	5.71611,5.98049
invx8	6.70952,6.29937	6.72629,-5.8318	6.7023,5.88162	6.75345,5.71595	6.49196,6.12737
finvx8	6.90052,6.31861	6.76683,-5.93947	6.89571,5.70143	6.77244,5.60298	6.18268,6.10043
latch	5.74824,6.28334	5.50012,-5.81513	5.75754,5.84892	5.37513,5.73694	5.67194,6.12304
flatch	4.86047,6.28334	4.69896,-5.80552	4.84123,5.86815	4.61019,5.73646	4.9371,6.12368
mux2x1	6.06469,6.32631	6.35602,-5.82442	-5.48759,5.87617	6.01072,5.67957	6.89507,6.09947

fmux2x1	5.37302,6.27757	6.24,-5.777	5.3486,-6.01082	6.30695,5.78661	5.28077,5.27518
nand2x1	7.83558,5.79601	7.81622,-5.92024	7.83558,-5.95408	7.65571,6.00324	7.78268,5.67546
fnand2x1	9.03066,6.28591	8.70146,-5.95549	9.00181,5.68027	8.67887,5.6251	8.94602,6.04272
nand3x1	8.82195,6.49815	8.80245,-5.52479	8.82195,-5.56485	8.62249,6.14921	7.95159,6.17995
fnand3x1	7.97707,6.32951	7.54507,-6.11508	7.97707,5.56742	7.41856,5.7238	6.43621,5.92009
nor2x1	6.98561,6.29809	7.20207,-6.12983	6.98099,-6.19261	7.12901,5.40109	6.96319,6.17754
fnor2x1	5.46899,6.68924	-6.06197,-5.84013	-5.40743,-6.12721	-6.15118,5.80952	-6.0384,5.98389
nor3x1	5.98179,6.40582	5.91885,-5.57991	5.98179,-5.62833	5.88588,6.16668	6.1046,6.17594
fnor3x1	6.10765,6.45134	6.02978,-5.49723	6.10765,-5.57319	6.01001,6.0699	6.1376,6.02718
oai21x1	7.10555,6.28976	6.50807,-6.03433	6.99061,-6.08841	6.43887,5.4466	7.25618,6.2282
foai21x1	7.84825,6.29937	6.81722,-5.8318	7.82901,5.88162	6.72076,5.81289	7.80689,6.12063
oai22x1	7.78829,6.31861	7.26587,-6.14649	7.69916,-5.46643	7.20178,5.5296	7.54784,5.92891
foai22x1	8.14161,6.29937	7.83341,-5.8318	8.12269,5.872	7.82155,5.80808	8.09416,6.12063
or2x1	8.21214,5.98197	7.96881,-5.69047	8.09736,-5.74472	7.83941,5.49531	7.41476,6.22627
for2x1	-6.94675,6.68924	-6.65067,-5.84013	-6.95791,-6.11791	-6.76257,5.81449	-6.36965,5.9887
or2x2	4.75884,6.28976	-4.65537,-5.72636	4.70145,-5.76043	-4.71242,5.56549	-4.75127,6.39283
for2x2	7.14297,5.99223	7.09056,-5.85935	7.11719,-6.11759	7.08633,5.79847	7.08179,5.93532
xnor2x1	6.24391,6.28334	6.24439,-5.80584	6.23926,5.86815	6.15354,5.74158	6.27725,6.13859
fxnor2x1	5.10991,6.28334	5.0739,-5.80584	5.08105,5.85853	4.8531,5.72957	5.22629,6.10781
xor2x1	6.17177,6.28334	6.16075,-5.80584	6.16231,5.86815	6.09121,5.73678	6.18075,6.14292
fxor2x1	5.03584,6.28334	4.90694,-5.81513	5.01661,5.86815	4.54225,5.72684	5.04354,6.11839

Table	invx8	finvx8	latch	flatch	mux2x1
and2x1	8.95404,6.39682	7.40035,6.52617	8.95404,-	8.95404,5.23366	8.77193,5.91336

			4.71748		
fand2x1	8.06565,6.84198	6.69887,6.83276	6.75269,4.72235	6.72386,5.45493	6.83199,6.35542
and2x2	5.87585,6.39865	5.83898,6.53318	5.85789,-4.7271	5.87713,5.23174	-5.48534,5.91336
fand2x2	7.0396,6.80889	7.35006,6.79463	7.40965,4.97552	7.43849,5.32803	5.53777,6.36562
aoi21x1	7.09064,6.32908	7.80548,6.44689	7.10219,-4.65047	7.11334,5.14004	6.6936,6.28174
faoi21x1	7.76734,6.81458	7.78106,6.25921	7.79191,-4.91783	7.79573,4.97552	6.87003,6.29484
aoi22x1	5.33968,6.8718	-5.48957,6.88541	5.48989,4.70626	5.38712,5.16633	-5.74634,6.45171
faoi22x1	7.79399,6.82051	7.81707,6.24872	7.80617,-4.82554	7.81963,5.1742	7.29022,5.65485
bufx2	5.43661,6.14922	5.70175,6.46361	5.74151,4.76493	5.79858,5.57864	5.16473,5.28077
fbufx2	7.62736,6.7758	7.62736,6.77804	7.65362,4.71274	7.65362,5.37834	7.46336,6.3162
bufx4	7.34136,6.56295	7.37102,6.51638	7.36637,4.74569	7.36637,5.25418	6.74182,5.45684
fbufx4	5.12163,6.85055	5.80668,6.84558	6.97007,4.73453	6.97007,5.41391	7.54023,6.35542
dffposx1	6.96015,6.54114	7.15925,6.50365	7.25127,5.19102	7.26057,5.62673	4.92203,5.33996
fdffposx1	-4.49418,6.79823	-5.17388,6.79943	6.06157,4.71274	6.06157,5.34693	-5.26329,6.29356
invx1	6.31861,6.90052	6.29937,6.70952	6.28334,4.86047	6.28334,5.74824	6.27757,5.37302
finvx1	5.7785,6.57239	5.74518,6.50733	5.78362,-4.58231	5.78362,5.28957	6.90601,6.22087
invx2	5.70143,6.89571	5.88162,6.7023	5.86815,4.84123	5.84892,5.75754	-6.01082,5.3486
finvx2	-5.93947,6.76683	-5.8318,6.72629	-5.80552,4.69896	-5.81513,5.50012	-5.777,6.24
invx4	6.10043,6.18268	6.12737,6.49196	6.12368,4.9371	6.12304,5.67194	5.27518,5.28077
finvx4	5.60298,6.77244	5.71595,6.75345	5.73646,4.61019	5.73694,5.37513	5.78661,6.30695
invx8	6.1362,6.94957	6.15377,6.91884	6.07996,4.76557	6.07749,5.32151	5.81805,6.98077
finvx8	6.84598,6.88219	6.8846,6.86817	6.8604,4.6621	6.86561,5.39917	6.57615,5.46414
latch	5.39917,6.86561	5.32151,6.07749	5.3462,-4.82361	5.36511,5.09303	5.1798,6.61532
flatch	4.6621,6.8604	4.76557,6.07996	5.20593,5.14504	5.2149,5.81417	5.53056,6.1724
mux2x1	6.1065,6.86449	6.48052,6.06718	6.10265,5.16106	-6.02208,5.27726	-5.88194,5.89487
fmux2x1	5.46414,6.57615	6.98077,5.81805	6.1724,5.53056	6.61532,5.1798	6.14819,6.1791
nand2x1	7.57641,6.63672	7.9284,5.92756	7.88687,-4.98321	8.58057,-4.79507	7.11649,6.14502
fnand2x1	8.62601,6.83124	8.98017,6.10824	9.03162,5.00564	6.89153,5.28977	7.93205,5.84184
nand3x1	8.70209,6.88588	8.40759,5.97086	8.88166,5.4216	6.94587,5.13459	7.46547,6.11093
fnand3x1	7.40202,6.81329	6.34349,6.04282	7.91805,4.95917	6.65427,5.10189	7.80539,5.84269

nor2x1	7.23783,6.51871	6.99908,5.96854	7.27859,-5.25849	7.45069,-4.81815	6.77718,6.08002
fnor2x1	-6.22193,6.67157	-6.63281,6.25537	-5.34599,4.90341	-5.35043,5.0522	8.41293,6.1484
nor3x1	6.02438,6.90775	6.35781,5.98523	6.00762,5.34918	6.41681,4.87105	5.58606,6.12448
fnor3x1	6.09238,6.53762	5.71002,5.91926	6.16131,5.15594	4.81951,4.89638	6.0501,6.13104
oai21x1	6.53691,6.51847	7.00648,6.0036	7.85755,-5.56004	7.14872,-4.78705	6.80123,6.0254
foai21x1	6.77844,6.89846	8.00374,6.11495	7.80906,5.2024	6.59223,5.32343	6.24808,5.85476
oai22x1	7.33028,6.79831	7.16919,6.07222	7.5937,4.90373	6.32724,5.12914	6.66981,5.86894
foai22x1	7.8411,6.89854	7.18041,6.11495	8.12952,5.2024	5.83046,5.33305	7.19537,5.85476
or2x1	7.96113,6.6102	8.41092,6.00776	7.97458,-5.6549	6.828,-5.28945	7.9701,5.87995
for2x1	-6.80909,6.66436	-6.68129,6.25361	-6.86831,4.93193	6.66603,5.07111	5.9197,6.15475
or2x2	-4.66563,6.64024	4.61071,6.06527	5.2524,-4.91655	5.25617,4.67836	-4.7967,6.07367
for2x2	7.11402,6.63664	6.77824,6.24171	7.17516,4.87457	6.30744,5.04322	-5.32561,6.13845
xnor2x1	6.19985,6.86337	6.1753,6.11671	6.35495,5.20593	5.46459,5.41288	6.27628,5.85539
fxnor2x1	4.85855,6.85536	4.92523,6.07645	5.33155,5.20593	-4.57823,5.35549	5.146,5.86534
xor2x1	6.12726,6.86096	6.09434,6.11671	6.27836,5.20593	5.32751,5.40327	6.21731,5.84078
fxor2x1	4.60506,6.85864	4.6479,6.07989	5.15209,5.20593	6.0907,5.34588	4.92937,5.85327

Table	fmux2x1	nand2x1	fnand2x1	nand3x1	fnand3x1
and2x1	8.95981,6.37824	8.9348,9.25958	8.86106,7.67255	8.93224,8.48992	8.87004,7.80507
fand2x1	6.69502,-5.7407	8.03681,8.73078	7.08141,7.57785	6.6758,7.24091	7.05402,8.80552
and2x2	5.70817,6.36574	5.85661,9.24563	5.01565,7.67159	5.48791,8.48992	5.65943,7.76653
fand2x2	7.42647,-7.41595	7.11842,8.34198	7.9431,7.4481	7.38322,7.73425	7.82584,8.64249
aoi21x1	7.74508,-5.87008	7.09064,7.76056	6.71591,7.06564	7.35457,7.12335	6.79594,7.76119
faoi21x1	7.72651,-5.19712	7.77596,8.42128	7.26271,7.57785	7.51821,-5.33676	7.67739,8.73093
aoi22x1	5.40735,6.15702	5.38111,8.85561	-5.1256,7.92503	5.27581,7.9309	-5.1977,8.3745
faoi22x1	7.6892,5.68304	7.76547,8.60102	7.32676,7.73693	7.53602,-5.26755	7.73022,8.85244
bufx2	5.68765,5.87328	5.452,8.88591	4.94383,7.8149	5.44366,7.90666	5.6745,7.92984
fbufx2	7.64017,5.61449	7.67348,8.90186	7.21607,7.6312	7.65298,6.99559	7.77919,8.68171
bufx4	7.36797,6.37824	7.33319,8.65266	5.82086,7.64322	7.31972,6.31502	7.42552,7.74135
fbufx4	7.00723,5.47388	6.31107,8.7327	7.77727,7.56151	6.94573,7.36511	6.40418,8.77361
dffposx1	7.14322,6.83672	6.96015,8.76327	6.26763,7.81154	7.02043,7.93629	7.21119,7.79477

fdffposx1	-5.27451,5.53057	6.62526,8.73174	-5.29053,7.59227	-5.47576,7.51161	6.92682,8.63672
invx1	6.32631,6.06469	6.28591,9.03066	5.79601,7.83558	6.32951,7.97707	6.49815,8.82195
finvx1	6.38645,6.13244	6.35763,8.0892	6.6773,7.0521	6.31791,7.38549	6.6158,8.40255
invx2	5.87617,-5.48759	5.68027,9.00181	-5.95408,7.83558	5.56742,7.97707	-5.56485,8.82195
finvx2	-5.82442,6.35602	-5.95549,8.70146	-5.92024,7.81622	-6.11508,7.54507	-5.52479,8.80245
invx4	6.09947,6.89507	6.04272,8.94602	5.67546,7.78268	5.92009,6.43621	6.17995,7.95159
finvx4	5.67957,6.01072	5.6251,8.67887	6.00324,7.65571	5.7238,7.41856	6.14921,8.62249
invx8	6.06718,6.48052	6.10824,8.98017	5.92756,7.9284	6.04282,6.34349	5.97086,8.40759
finvx8	6.86449,6.1065	6.83124,8.62601	6.63672,7.57641	6.81329,7.40202	6.88588,8.70209
latch	5.27726,-6.02208	5.28977,6.89153	-4.79507,8.58057	5.10189,6.65427	5.13459,6.94587
flatch	5.16106,6.10265	5.00564,9.03162	-4.98321,7.88687	4.95917,7.91805	5.4216,8.88166
mux2x1	5.34235,-5.82936	6.0323,8.69979	-6.98131,7.59993	-5.26091,5.99492	-5.55043,7.79248
fmux2x1	6.34212,6.77308	5.84184,7.93205	6.14502,7.11649	5.84269,7.80539	6.11093,7.46547
nand2x1	7.59993,-6.98131	7.11004,8.81425	7.16401,7.76297	7.16783,8.56263	7.0298,7.95273
fnand2x1	8.69979,6.0323	8.35592,8.70483	8.85429,7.82343	8.51836,7.49316	8.812,8.78937
nand3x1	7.79248,-5.55043	8.78937,8.812	7.95273,7.0298	8.43183,7.97322	8.44298,7.7238
fnand3x1	5.99492,-5.26091	7.49316,8.51836	8.56263,7.16783	7.39241,7.48662	7.90652,8.76438
nor2x1	7.09671,-5.11504	7.36972,8.37466	7.23943,6.91518	6.72373,7.13248	7.40633,7.23992
fnor2x1	-6.37081,5.74183	-5.69244,8.11995	-6.15805,6.46767	-6.92829,6.91253	-6.45506,7.87588
nor3x1	5.95239,-6.26827	5.9609,8.78172	5.82019,7.06514	5.7383,7.8946	5.96528,8.43952
fnor3x1	6.02191,-6.09034	6.116,8.78989	6.01641,7.079	6.04625,7.90613	6.00514,8.42028
oai21x1	7.60913,7.25864	6.76423,8.28095	7.27051,6.88748	7.0247,7.81615	7.63158,6.31156
foai21x1	7.66839,5.29169	6.83581,8.49433	6.56195,7.20317	7.40559,7.31705	8.26344,8.45953
oai22x1	7.31187,5.27662	7.3274,8.37322	6.57414,7.1664	7.14826,7.65389	7.42536,8.493
foai22x1	7.89089,5.28207	7.93243,8.49433	7.3699,7.20317	7.71388,7.29397	7.35258,8.47107
or2x1	7.47603,5.94221	8.02647,8.00942	7.50284,6.65298	7.86183,7.7454	7.64454,7.06294
for2x1	-6.71053,5.75113	-6.62029,8.13437	5.49554,6.48199	-6.87908,6.9356	-8.92813,7.87588
or2x2	-4.7401,5.37665	-4.51084,8.37466	5.56027,6.89464	-4.47303,7.13786	5.8593,7.25146
for2x2	6.97407,5.70881	7.02519,8.11275	6.31887,6.55363	6.69028,-6.03082	7.31187,7.86396
xnor2x1	6.17642,5.52574	6.24535,8.56209	6.22098,7.20269	6.18495,7.50854	6.32775,8.44298

fxnor2x1	4.87971,5.48374	4.90918,8.54815	-5.06791,7.20269	4.79734,7.37165	5.25899,8.44298
xor2x1	6.11951,5.52574	6.15787,8.56209	6.28543,7.20269	6.05243,7.50854	6.2205,8.44298
fxor2x1	4.66618,5.46707	4.82971,8.54815	-5.19391,7.20269	4.58455,7.3828	4.9605,8.44298

Table	nor2x1	fnor2x1	nor3x1	fnor3x1	oai21x1
and2x1	8.91813,-6.22615	7.07333,7.29967	8.86106,6.03318	8.87453,5.99161	8.95404,7.59657
fand2x1	6.47636,-6.60145	8.12812,6.89138	7.02807,6.04927	5.84967,5.77705	6.75942,7.66741
and2x2	5.92201,-6.27308	4.96339,7.28624	5.62994,6.02438	5.65847,5.98585	5.77806,7.57797
fand2x2	5.24077,5.32907	7.32266,7.31666	7.83593,5.82521	7.89072,5.4514	7.438,7.93596
aoi21x1	6.87211,-5.74254	7.19067,6.54691	7.23646,5.9598	7.24685,5.86106	7.09026,7.13393
faoi21x1	7.69621,-6.22616	7.13128,7.24629	7.40465,5.99792	5.88819,5.7482	7.60976,6.87843
aoi22x1	-5.83899,-6.89481	-4.96148,6.61704	5.1139,5.7212	-5.10853,6.35535	5.42032,7.85915
faoi22x1	7.05148,-5.88009	7.78085,7.27667	7.53922,6.10035	6.07759,5.8933	7.77252,7.02873
bufx2	5.21699,-5.41667	5.19904,6.94823	5.15864,6.0645	5.03809,6.013	5.63122,7.78669
fbufx2	7.31921,-6.5903	7.60622,7.0871	7.73434,6.01797	7.73114,5.86939	7.62159,7.62511
bufx4	7.1639,-6.75939	6.87343,6.83313	5.94638,6.00047	7.4967,6.00395	7.38032,7.93193
fbufx4	6.57672,-6.63913	6.13639,7.16093	5.84299,6.02483	7.10759,5.83366	6.98246,7.6594
dffposx1	6.85788,-5.75677	6.69597,6.916	5.11888,6.05461	5.0538,6.02891	7.16695,8.06851
fdffposx1	-4.94091,-5.86317	-5.61965,7.09979	6.50028,6.00506	6.6118,5.79684	-5.26457,7.57736
invx1	6.68924,5.46899	6.29809,6.98561	6.45134,6.10765	6.40582,5.98179	6.29937,7.84825
finvx1	6.70549,-5.49749	6.22053,7.23245	6.53508,6.05422	6.49344,5.91885	5.74518,6.78614
invx2	-6.12721,-5.40743	-6.19261,6.98099	-5.57319,6.10765	-5.62833,5.98179	5.88162,7.82901
finvx2	-5.84013,-6.06197	-6.12983,7.20207	-5.49723,6.02978	-5.57991,5.91885	-5.8318,6.81722
invx4	5.98389,-6.0384	6.17754,6.96319	6.02718,6.1376	6.17594,6.1046	6.12063,7.80689
finvx4	5.80952,-6.15118	5.40109,7.12901	6.0699,6.01001	6.16668,5.88588	5.81289,6.72076
invx8	6.25537,-6.63281	5.96854,6.99908	5.91926,5.71002	5.98523,6.35781	6.11495,8.00374
finvx8	6.67157,-6.22193	6.51871,7.23783	6.53762,6.09238	6.90775,6.02438	6.89846,6.77844
latch	5.0522,-5.35043	-4.81815,7.45069	4.89638,4.81951	4.87105,6.41681	5.32343,6.59223
flatch	4.90341,-5.34599	-5.25849,7.27859	5.15594,6.16131	5.34918,6.00762	5.2024,7.80906
mux2x1	5.74183,-6.37081	-5.11504,7.09671	-6.09034,6.02191	-6.26827,5.95239	5.29169,7.66839
fmux2x1	6.1484,8.41293	6.08002,6.77718	6.13104,6.0501	6.12448,5.58606	5.85476,6.24808
nand2x1	6.46767,-6.15805	6.91518,7.23943	7.079,6.01641	7.06514,5.82019	7.20317,6.56195
fnand2x1	8.11995,-5.69244	8.37466,7.36972	8.78989,6.116	8.78172,5.9609	8.49433,6.83581

nand3x1	7.87588,-6.45506	7.23992,7.40633	8.42028,6.00514	8.43952,5.96528	8.45953,8.26344
fnand3x1	6.91253,-6.92829	7.13248,6.72373	7.90613,6.04625	7.8946,5.7383	7.31705,7.40559
nor2x1	7.56446,-5.91128	7.1674,-5.7333	6.25932,5.25936	5.90926,6.73755	7.49901,6.32186
fnor2x1	6.30114,5.21679	5.42405,7.33127	6.34459,6.13	6.22155,6.00459	6.09619,6.83196
nor3x1	6.00459,6.22155	6.73755,5.90926	5.94772,6.13898	5.9757,5.99223	5.98228,7.90147
fnor3x1	6.13,6.34459	5.25936,6.25932	6.01715,6.15472	5.9993,6.0296	6.00836,7.65844
oai21x1	6.71328,7.02212	8.60206,6.99979	8.41167,5.84141	8.40499,5.44938	5.98213,6.37625
foai21x1	6.83196,6.09619	6.32186,7.49901	7.65844,6.00836	7.90147,5.98228	8.16637,7.70843
oai22x1	6.62526,5.86932	5.22837,7.77145	7.74817,6.01742	7.31251,5.98119	7.35944,7.86482
foai22x1	7.16332,6.08466	5.40356,7.49901	8.56695,6.0166	7.81266,5.97405	7.97057,8.15708
or2x1	7.41595,-5.93315	6.4379,7.35099	8.32949,5.58851	8.30127,5.49656	7.75612,6.67878
for2x1	-7.15093,6.31268	6.26722,7.57625	-6.98828,6.03252	5.74523,5.95924	-6.56223,7.28542
or2x2	4.92424,5.9197	5.00712,7.20222	-5.0364,6.02483	4.89381,6.00422	4.56212,7.36233
for2x2	7.62889,5.85779	-6.81782,7.48379	7.36088,6.02483	-5.85026,5.97981	7.07941,7.22902
xnor2x1	6.09554,6.13503	4.98568,7.50205	6.26314,6.09183	6.21746,5.94608	6.19408,8.21957
fxnor2x1	5.06428,6.12349	-4.91279,7.47656	5.24631,6.0976	5.1798,5.95979	4.94892,8.17182
xor2x1	6.001,6.13503	4.88575,7.50205	6.21379,6.09183	6.07094,5.94608	6.13063,8.20995
fxor2x1	4.82842,6.13503	-5.23849,7.49064	5.07486,6.09156	5.13684,5.96226	4.68389,8.19073

Table	foai21x1	oai22x1	foai22x1	or2x1	for2x1
and2x1	8.59495,7.50597	8.9162,6.8473	7.40035,6.75849	8.93672,-7.24646	6.98548,7.48625
fand2x1	6.76711,8.2079	6.75942,7.93051	6.60083,7.34534	6.49029,-8.64018	6.53979,7.96113
and2x2	-5.0102,7.53224	5.77806,6.83896	5.49881,6.71905	5.94125,-7.24646	5.04162,7.50156
fand2x2	7.36832,7.11313	7.40965,7.85104	7.06171,7.35368	5.26913,5.28985	7.23712,7.85991
aoi21x1	6.64512,6.64056	7.09026,6.37087	7.45999,6.2625	6.87211,-7.35226	6.50739,7.50412
faoi21x1	7.26143,6.7772	7.60019,7.24921	7.53863,6.73422	7.70546,-6.85562	6.68853,8.47171
aoi22x1	5.43044,7.0137	5.41083,7.01338	5.22521,7.16118	5.26,-6.8094	5.22584,7.81587
faoi22x1	8.07055,7.56536	7.77252,7.36233	7.50942,6.89157	7.05148,-6.65721	8.30898,8.43583
bufx2	5.23462,7.17881	5.6216,8.12109	5.22981,7.46993	5.21699,-6.93983	5.29073,7.37774
fbufx2	7.56522,6.90792	7.62159,7.90776	7.64594,7.304	7.33843,-8.62172	7.27885,8.49285
bufx4	6.61598,7.23039	7.37102,7.12303	7.30032,7.22466	7.17785,-6.78593	6.91751,7.1703
fbufx4	7.33566,6.91273	6.96964,7.93756	6.30979,7.27997	6.57672,-8.7167	7.03157,8.48132
dffposx1	4.84572,7.20126	7.14803,7.39971	6.76394,7.30834	6.8768,-7.04101	6.19518,7.38923
fdffposx1	-4.75279,7.68935	-5.26457,7.91929	-4.72588,7.30497	-4.93129,-7.09364	4.81016,7.6722
invx1	6.28976,7.10555	6.29937,8.14161	6.31861,7.78829	6.68924,-6.94675	5.98197,8.21214
finvx1	6.21541,6.53739	5.74518,7.08737	6.36788,6.55156	6.72407,-7.2363	5.89637,7.25131
invx2	-6.08841,6.99061	5.872,8.12269	-5.46643,7.69916	-6.11791,-6.95791	-5.74472,8.09736

finvx2	-6.03433,6.50807	-5.8318,7.83341	-6.14649,7.26587	-5.84013,-6.65067	-5.69047,7.96881
invx4	6.2282,7.25618	6.12063,8.09416	5.92891,7.54784	5.9887,-6.36965	6.22627,7.41476
finvx4	5.4466,6.43887	5.80808,7.82155	5.5296,7.20178	5.81449,-6.76257	5.49531,7.83941
invx8	6.0036,7.00648	6.11495,7.18041	6.07222,7.16919	6.25361,-6.68129	6.00776,8.41092
finvx8	6.51847,6.53691	6.89854,7.8411	6.79831,7.33028	6.66436,-6.80909	6.6102,7.96113
latch	-4.78705,7.14872	5.33305,5.83046	5.12914,6.32724	5.07111,6.66603	-5.28945,6.828
flatch	-5.56004,7.85755	5.2024,8.12952	4.90373,7.5937	4.93193,-6.86831	-5.6549,7.97458
mux2x1	7.25864,7.60913	5.28207,7.89089	5.27662,7.31187	5.75113,-6.71053	5.94221,7.47603
fmux2x1	6.0254,6.80123	5.85476,7.19537	5.86894,6.66981	6.15475,5.9197	5.87995,7.9701
nand2x1	6.88748,7.27051	7.20317,7.3699	7.1664,6.57414	6.48199,5.49554	6.65298,7.50284
fnand2x1	8.28095,6.76423	8.49433,7.93243	8.37322,7.3274	8.13437,-6.62029	8.00942,8.02647
nand3x1	6.31156,7.63158	8.47107,7.35258	8.493,7.42536	7.87588,-8.92813	7.06294,7.64454
fand3x1	7.81615,7.0247	7.29397,7.71388	7.65389,7.14826	6.9356,-6.87908	7.7454,7.86183
nor2x1	6.99979,8.60206	7.49901,5.40356	7.77145,5.22837	7.57625,6.26722	7.35099,6.4379
fnor2x1	7.02212,6.71328	6.08466,7.16332	5.86932,6.62526	6.31268,-7.15093	-5.93315,7.41595
nor3x1	5.44938,8.40499	5.97405,7.81266	5.98119,7.31251	5.95924,5.74523	5.49656,8.30127
fnor3x1	5.84141,8.41167	6.0166,8.56695	6.01742,7.74817	6.03252,-6.98828	5.58851,8.32949
oai21x1	6.16872,6.33107	5.96818,6.06857	7.98082,-5.40894	5.48101,-6.70664	7.19067,6.50715
foai21x1	7.13384,6.83343	8.15708,7.97057	7.86482,7.35944	7.28542,-6.56223	6.67878,7.75612
oai22x1	-5.40894,7.98082	-5.43834,6.83447	-5.75172,7.08776	5.56578,-6.79363	5.50665,7.75239
foai22x1	6.06857,5.96818	7.11525,8.04908	6.81882,7.81739	6.25545,-6.31229	7.17742,7.96113
or2x1	6.50715,7.19067	7.96113,7.17742	7.75239,5.50665	7.45229,5.67791	7.34329,6.26636
for2x1	-6.70664,5.48101	-6.31229,6.25545	-6.79363,5.56578	5.24948,-7.42471	-5.86971,7.26796
or2x2	6.25143,7.42632	4.63903,6.30416	-4.89702,5.67867	4.87553,5.23756	6.55131,7.63248
for2x2	6.25128,7.28638	7.16093,6.22308	6.56086,5.52436	7.59928,5.23794	5.60278,7.37535
xnor2x1	5.25826,6.02156	6.20321,7.18287	6.05811,6.46479	6.04074,5.41867	5.45558,7.93834
fxnor2x1	4.6576,6.79103	5.02615,7.16492	4.67163,6.45784	4.71306,5.41867	-4.58708,7.9377
xor2x1	5.12355,6.80498	6.12246,7.18287	5.97363,6.46479	5.96575,5.4302	5.29873,7.93834
fxor2x1	-5.02694,6.79103	4.81689,7.1739	-4.59052,6.45499	-4.46662,5.41867	-4.81729,7.9377

Table	or2x2	for2x2	xnor2x1	fxnor2x1	xor2x1
and2x1	7.40228,7.30494	8.58918,-5.05061	7.40035,4.78929	8.95404,6.48821	8.95404,-4.76685
fand2x1	6.44752,7.0771	6.78008,-5.04313	6.75269,4.95373	6.76711,6.25	6.70944,4.69159
and2x2	5.90278,7.2707	5.38531,-5.05061	5.78095,4.78416	5.86751,6.49479	5.86751,-

					4.76685
fand2x2	5.24077,6.66797	7.2501,6.11733	7.39523,5.10979	7.39523,6.21443	7.39523,4.93706
aoi21x1	6.86019,7.41228	6.96291,-4.46628	7.11373,4.7319	7.11373,6.12945	7.10219,-4.52703
faoi21x1	7.723,7.54468	7.54278,5.03352	7.78297,-4.69094	7.81073,6.25849	7.79605,-4.93578
aoi22x1	5.31312,6.54932	5.29162,-4.63059	5.51487,4.82905	5.49558,6.08489	5.49937,4.57481
faoi22x1	7.04058,7.54853	8.20034,5.00949	7.80617,4.74286	7.81578,6.05404	7.80617,-4.86528
bufx2	5.17884,6.98907	4.90215,-4.79116	5.70753,5.06694	5.76075,6.27549	5.75113,4.94607
fbufx2	7.28078,7.06211	7.64465,4.5743	7.65362,4.97071	7.64465,6.21619	7.65362,4.68422
bufx4	7.15493,6.95368	6.95182,-4.75988	7.36156,4.92972	7.38064,6.18283	7.37118,4.6758
fbufx4	6.05012,7.05211	-5.07807,4.6464	6.97007,4.93866	5.89424,6.23846	6.97007,4.69287
dffposx1	6.79183,6.45699	7.05377,-4.83328	7.22883,5.29169	7.22274,6.38321	7.25127,4.95345
fdffposx1	-5.04602,7.01673	5.81994,5.12902	-5.25208,4.92713	-5.16715,6.20001	6.06157,4.67236
invx1	5.99223,7.14297	6.28976,4.75884	6.28334,5.10991	6.28334,6.24391	6.28334,5.03584
finvx1	6.05269,7.62889	6.22309,4.87778	5.78362,4.86944	5.78362,6.10467	5.78362,4.85374
invx2	-6.11759,7.11719	-5.76043,4.70145	5.85853,5.08105	5.86815,6.23926	5.86815,5.01661
finvx2	-5.85935,7.09056	-5.72636,-4.65537	-5.80584,5.0739	-5.80584,6.24439	-5.81513,4.90694
invx4	5.93532,7.08179	6.39283,-4.75127	6.10781,5.22629	6.13859,6.27725	6.11839,5.04354
finvx4	5.79847,7.08633	5.56549,-4.71242	5.72957,4.8531	5.74158,6.15354	5.72684,4.54225
invx8	6.24171,6.77824	6.06527,4.61071	6.07645,4.92523	6.11671,6.1753	6.07989,4.6479
finvx8	6.63664,7.11402	6.64024,-4.66563	6.85536,4.85855	6.86337,6.19985	6.85864,4.60506
latch	5.04322,6.30744	4.67836,5.25617	5.35549,-4.57823	5.41288,5.46459	5.34588,6.0907
flatch	4.87457,7.17516	-4.91655,5.2524	5.20593,5.33155	5.20593,6.35495	5.20593,5.15209
mux2x1	5.70881,6.97407	5.37665,-4.7401	5.48374,4.87971	5.52574,6.17642	5.46707,4.66618
fmux2x1	6.13845,-5.32561	6.07367,-4.7967	5.86534,5.146	5.85539,6.27628	5.85327,4.92937
nand2x1	6.55363,6.31887	6.89464,5.56027	7.20269,-5.06791	7.20269,6.22098	7.20269,-5.19391
fnand2x1	8.11275,7.02519	8.37466,-4.51084	8.54815,4.90918	8.56209,6.24535	8.54815,4.82971
nand3x1	7.86396,7.31187	7.25146,5.8593	8.44298,5.25899	8.44298,6.32775	8.44298,4.9605
fnand3x1	-6.03082,6.69028	7.13786,-4.47303	7.37165,4.79734	7.50854,6.18495	7.3828,4.58455
nor2x1	7.48379,-6.81782	7.20222,5.00712	7.47656,-4.91279	7.50205,4.98568	7.49064,-5.23849
fnor2x1	5.85779,7.62889	5.9197,4.92424	6.12349,5.06428	6.13503,6.09554	6.13503,4.82842
nor3x1	5.97981,-5.85026	6.00422,4.89381	5.95979,5.1798	5.94608,6.21746	5.96226,5.13684
fnor3x1	6.02483,7.36088	6.02483,-5.0364	6.0976,5.24631	6.09183,6.26314	6.09156,5.07486
oai21x1	7.28638,6.25128	7.42632,6.25143	6.79103,4.6576	6.02156,5.25826	6.79103,-5.02694

foai21x1	7.22902,7.07941	7.36233,4.56212	8.17182,4.94892	8.21957,6.19408	8.19073,4.68389
oai22x1	5.52436,6.56086	5.67867,-4.89702	6.45784,4.67163	6.46479,6.05811	6.45499,-4.59052
foai22x1	6.22308,7.16093	6.30416,4.63903	7.16492,5.02615	7.18287,6.20321	7.1739,4.81689
or2x1	7.37535,5.60278	7.63248,6.55131	7.9377,-4.58708	7.93834,5.45558	7.9377,-4.81729
for2x1	5.23794,7.59928	5.23756,4.87553	5.41867,4.71306	5.41867,6.04074	5.41867,-4.46662
or2x2	-4.52415,-6.2547	4.79635,5.33209	5.24681,-4.62376	5.29009,5.09841	5.25226,-4.90109
for2x2	4.94224,7.57275	4.66577,4.70985	-4.94724,4.66947	-4.84073,6.0127	-4.9111,-4.52046
xnor2x1	6.0127,-4.84073	5.09841,5.29009	6.29473,-4.5991	6.3005,5.40909	6.25753,6.05339
fxnor2x1	4.66947,-4.94724	-4.62376,5.24681	5.005,4.98128	5.02647,6.27323	5.02647,4.73196
xor2x1	5.95357,-4.85227	4.98694,5.29009	6.20658,5.02647	5.24071,6.3005	6.1131,6.03442
fxor2x1	-4.52046,-4.9111	-4.90109,5.25226	4.73196,5.02647	6.05339,6.25753	4.73068,4.73196

Table	fxor2x1
and2x1	8.95404,6.39203
fand2x1	6.76711,6.19216
and2x2	5.86751,6.39059
fand2x2	7.39523,6.16572
aoi21x1	7.11373,6.03824
faoi21x1	7.80147,6.19168
aoi22x1	5.49558,6.00569
faoi22x1	7.81578,6.00356
bufx2	5.76075,6.18139
fbufx2	7.64465,6.10211
bufx4	7.38064,6.09739
fbufx4	5.89424,6.11685
dffposx1	7.22274,6.29088

fdffposx1	-5.17549,6.13736
invx1	6.28334,6.17177
finvx1	5.78362,6.0284
invx2	5.86815,6.16231
finvx2	-5.80584,6.16075
invx4	6.14292,6.18075
finvx4	5.73678,6.09121
invx8	6.11671,6.09434
finvx8	6.86096,6.12726
latch	5.40327,5.32751
flatch	5.20593,6.27836
mux2x1	5.52574,6.11951
fmux2x1	5.84078,6.21731
nand2x1	7.20269,6.28543
fnand2x1	8.56209,6.15787
nand3x1	8.44298,6.2205
fnand3x1	7.50854,6.05243
nor2x1	7.50205,4.88575
fnor2x1	6.13503,6.001
nor3x1	5.94608,6.07094
fnor3x1	6.09183,6.21379
oai21x1	6.80498,5.12355
foai21x1	8.20995,6.13063
oai22x1	6.46479,5.97363

foai22x1	7.18287,6.12246
or2x1	7.93834,5.29873
for2x1	5.4302,5.96575
or2x2	5.29009,4.98694
for2x2	-4.85227,5.95357
xnor2x1	6.3005,5.24071
fxnor2x1	5.02647,6.20658
xor2x1	6.12272,5.29004
fxor2x1	4.73068,6.20658

VITA

Pratik Jitendra Shah received his Bachelor of Engineering degree in electronics and communication from R.V. College of Engineering in 2005. He entered the computer engineering program at Texas A&M University in September 2006 and received his Master of Science degree in December 2008. His research interests include physical design to improve circuit printability.

He may be reached at the Department of Electrical and Computer Engineering, 111 WERC, Texas A&M University, College Station, TX 77843. His email address is pratiksh@ece.tamu.edu.