Yield-aware Placement Optimization

P. Azzoni (univr), M. Bertoletti (pdf), N. Dragone (pdf), F. Fummi (univr), C. Guardiani (pdf), W. Vendraminetto (univr)

Affiliations: pdf – PDF Solutions (IT); univr – University of Verona (IT)

1 Abstract

In this paper we describe a methodology addressing the issue of avoiding yield hazardous cell abutments during placement. This is made possible by accurate characterization of the yield penalty associated with particular cell-to-cell interactions. Of course characterizing all possible cell abutments in a library of 600+ cells is impractical. We will describe some simple heuristics that attempt to resolve the cell abutment pre-characterization complexity. Finally we will show a possible implementation of the proposed yield-aware placement optimization methodology and demonstrate the potential of cell interaction penalty characterization for a 90nm design test case.

2 Introduction and Motivations

Maximum packing density and performance × power trade-off optimization have always been the key objectives of physical design. As technology scales in the nanometer regime yield and variability are also becoming critical. At 65nm and below, complex lithography and *RET* (Resolution Enhancement Technique) methods are required in order to ensure that the drawn layout patterns are printed on silicon without significant errors. In spite of such highly complex tools and methodologies it is not possible to ensure that all critical circuit layouts will be printed as drawn. Imperfections due to lithography process and *CMP* in conjunction with new materials and device architectures significantly increase the probability of faults and the circuit parameters variability [1]. As a consequence the yield of IC products is no longer dominated by random defects (Random Mechanisms Limited Yield or *RMLY*) but at 90nm and below most yield detractors can be characterized as Systematic Mechanism Limited Yield (*SMLY*) effects, i.e. such that their probability is strongly impacted by one or more layout attributes. Different type of *SMLY* may affect both interconnect layers or front-end layers (i.e. Active/Poly/M1/Contact/Via1) and can therefore be found inside standard cell layouts or in the interconnects. A significant fraction of *SMLY* is actually caused by cell-to-cell interactions. This is because standard cell designers are becoming increasingly aware of yield issues and try to eliminate as much as possible those features that may be litho "un-friendly". The place&route tools however have no notion of the possible yield penalties that may be caused by particular abutments of cells. Cell placement in particular is only driven by timing and routability constraints.

In this paper we describe a methodology that allows to model hazardous cell-to-cell interactions due to litho and other *SMLY* effects. As the yield penalty associated with particular cell abutments is characterized, the place&route tools can be made aware of such interactions and thus driven to avoid them as much as possible, *compatibly* with the other design constraints.

The problem of IC yield modeling, analysis and optimization has been thoroughly explored in the literature [4]-[7]. In these works the yield-layout attributes are extracted from the final full chip layout (flat). A hierarchical cell-level pre-characterization of such attributes is necessary in order to enable yield-aware design flow and proactive *DFM* [9]-[10] (Figure 1). In this paper we describe the extension of such hierarchical methodology in order to capture cell-to-cell interactions and demonstrate a prototype cell placement algorithm with yield and lithography hot spot minimization considerations.

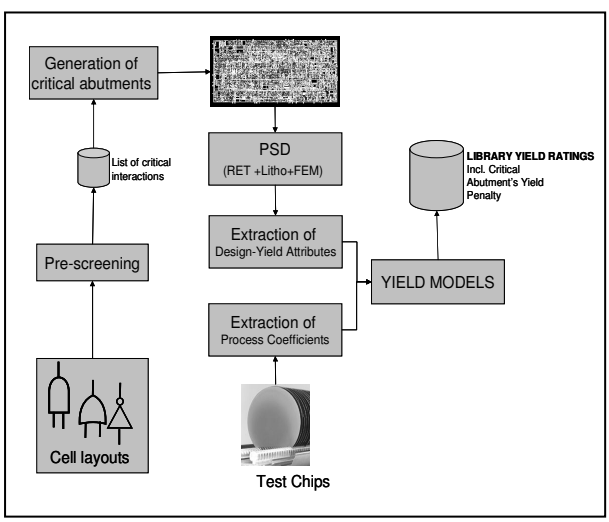

Figure 1 – Standard Cell DFM View Creation Flow

978-3-9810801-2-4/DATE07 © 2007 EDAA

3 Methodology Flow

The optimization methodology described in this work concerns the block place and route step of ASIC/SoC logic design flow. As previously discussed the main idea is to add a new yield term to the cost function of the placement optimization that is a function of the relative location of the cells. In this way, the placer is aware of the yield cost of any possible transformation and can try to optimize yield, while at the same time maintaining all the other hard/soft constraints. In fact, it is conceivable that other design constraints, such as block area and timing, will be considered as primary goals and the optimizer will never be allowed to increase yield at the cost of increasing area or violating the timing specs. The proposed yield-aware placement optimization algorithm will however be able to exploit any available design slack in order to optimize for yield. It will also be able to select the most litho and yield friendly configuration of standard cell placement among all legal equivalent timing/area placement implementations. It is indeed possible that some of these alternative solutions will cause some other secondary cost functions to be worse, such as for example total wire length and/or power. The availability of accurate yield models is required in order to quantify the secondary objectives trade-off for a particular implementation.

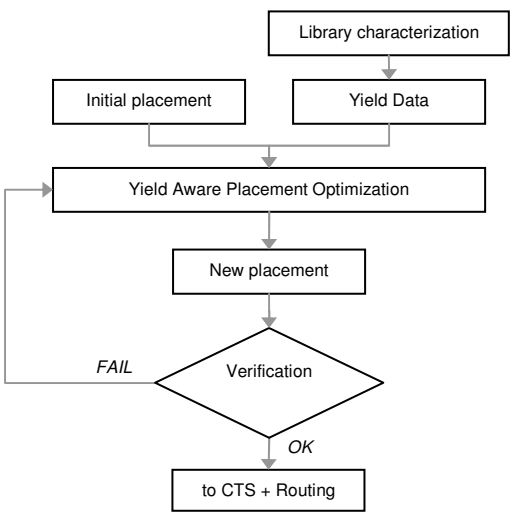

Figure 2 – Diagram of the methodology workflow

The diagram in Figure 2 shows the high level view of the proposed methodology flow. A short description of the particular implementation of the method that was chosen to demonstrate its feasibility and impact is provided in the next paragraphs.

3.1 Initial placement

An initial legal block/chip placement is first obtained by using any of the popular placement algorithms, such as those described in [11]-[14]. After this is done the results can be exported in a standard physical design representation format such as for example Design Exchange Format (*DEF*, v. 5.6 - *.def* file extension) for yield-aware optimization. The cells are organized in rows that occupy most of the available block/die area. Filler cells separators are used to spread out cells in order to ensure that the resulting placed data-base is routable. In our implementation filler cells are designed to be "yield-neutral", i.e. in order to create negligible yield interactions with any of the other library cells.

3.2 Yield Data

The yield attributes of each standard cell in the library is extracted by using yield simulation as described in [8] and [9]. This step requires the availability of a set of process coefficients that describe the process sensitivity of each yield loss mechanism for the particular technology of choice[10].
In addition to this the penalty associated with a set of possibly critical cell abutments is also extracted and quantified by simulation. This entails evaluating the yield loss that is caused by the variability of lithographic equipment and process parameters, such as focus and exposure dose.

3.3 Yield-aware Placement Optimization

In this step the initial placement is first scanned by rows and critical cell clusters (also called "frames") are determined. Next, each frame is optimized by exhaustively applying a predefined set of yield driven transformations and a new chip placement is produced. The cell position and neighborhood in the placement is modified while also trying to minimize the impact on the primary design objectives.

3.4 Verification

The yield optimized placement is verified in order to check that all primary design objectives are still met and that the final resulting trade-off between yield and the other secondary design objectives is acceptable.

4 Library Characterization

The need to include a yield objective along with other traditional design objectives in the design optimization cost function (e.g. for physical synthesis) was presented in [9]. In that work a new cell library view format, called .pdfm, was first introduced to represent standard cell manufacturability [9]. In order to support yield aware placement flows, such a format must be extended to account for cell-to-cell interactions. In fact, not only isolated cells but also couples or triplets of cells should be characterized. In fact several failure mechanisms can be caused by cell-to-cell interactions. They may eventually represent a relevant fraction of the total chip's yield loss. We can classify such mechanisms in the following three main types:
Critical Area (CA): cell-to-cell interactions must be included in order to accurately extract bottom-up critical area yield loss without underestimating the impact of material shorts. An example of the impact of neglecting cell-to-cell interaction penalty on random cell fail rate is shown in Figure 3.

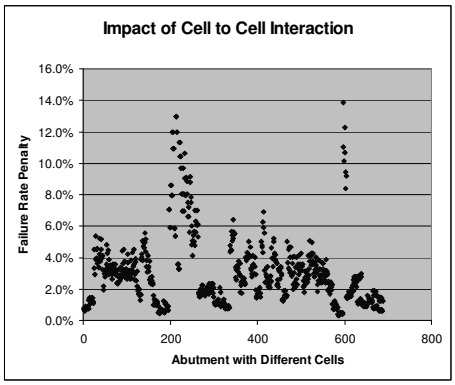

Figure 3 – Impact of CA cell-to-cell interaction

Systematic effects: these are yield loss mechanisms that are correlated to particular layout patterns and that can occur within cells but also at the cell boundaries. An example of an effect which is quite common at 90nm and below is density dependent contact failure rate (see Figure 4).

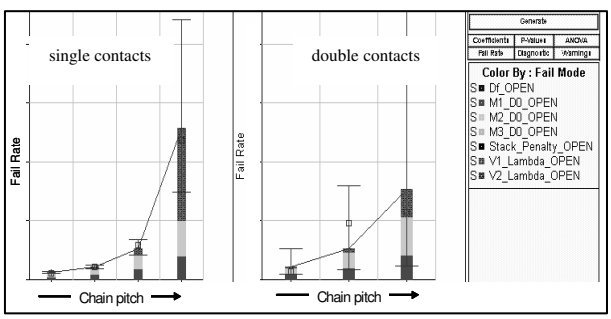

Figure 4 – Single vs. double contact fail rates as a function of density

Lithography: lithography related hot spots are strongly influenced by the layout environment. Identical patterns with varying neighborhoods may lead to strong differences of the printed shapes. In Figure 5 we show an example of a cell-to-cell interaction that causes a lithography hot spots.

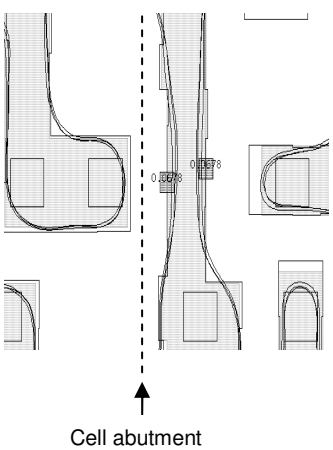

Figure 5 – Example of litho hot spot generated by cell-to-cell interaction

In theory, all possible cell-to-cell interactions should be characterized and this would represent a problem of complexity at least $2*N^2$. Given the typical size of modern standard cell libraries, which can contain up to a few thousands cells, an exhaustive characterization is impractical. In order to alleviate the cell yield characterization complexity issue we explored a set of heuristics that allow to drastically reducing the number of abutments relevant for yield loss estimation. This has been implemented in a linear complexity cell-library pre-screening procedure consisting of the following steps.

1) *Critical Area based cell interaction pruning*: by analyzing the critical area by layer of certain cells (in particular that of larger cells) it is possible to exclude a-priori a certain number of abutments because their contribution to the total CA will be *always* negligible compared to isolated CA extraction.
2) *Lithography simulation based cell boundary pre-screening*: similarly to the previous case it is possible to simulate the litho hot spots of every cell in isolation with a set of worst case dummy environments. All cells that do not create any hot-spot with the worst-case environment can be pruned from the candidate list for abutment characterization.
3) *Systematic pre-screening*: similarly to the previous case it is possible to identify certain cells that, due to their periphery's layout attribute will never be able to sensitize a specific systematic effect and can therefore be also pruned during pre-screening.

5 Optimization Algorithm

The optimization algorithm requires the yield information obtained in the library characterization step and the initial chip placement (*.def* file). The actual optimization procedure consists of a preliminary phase when cell rows are scanned to detect critical cell abutments and partition the layout in frames based on that, followed by a yield maximization step where the cluster obtained in the first step are optimized separately.

5.1 Definition of yield-aware cell placement transformations

We decided to set some limits to the list of allowed cell placement modifications, because we assume that the initial placement is already optimized for timing, area *DRC* and so on. We applied two types of transformations: *local cell movements* and *decoupling*. The former refers to changes performed on the cell's orientation and position within the selected frame. The latter, instead consists of a filler cell insertion between two adjacent cells.

Local cell movements

A cell placement can be modified locally by changing either its orientation (i.e., by flipping a cell around its vertical axis within a row) and/or its position. In order to reduce the probability that such moves can impact

routability and timing we decided to only allow a cell to be moved within its frame.

Decoupling (filler insertion)

The insertion of fillers minimizes the cell abutment penalty, because they were designed in order to be yield neutral in the target library. Since the area of the block/die is fixed, the maximum number of filler cells that can be inserted is also fixed and defines the so called block utilization ratio. Moreover since filler cells are used to decrease local congestion and make sure that the block is routable only limited re-positioning of filler cells is possible in order to eliminate nasty cell abutments. When new filler is inserted in a row, most of the row cells (e.g., all the cells to the right of the insertion point) must be shifted by at least the width of the filler cell. This shift must be coherent with the maximum offset allowed for any displaced cell.

5.2 Algorithm implementation description

In this section we describe a possible implementation of the yield-aware placement algorithm that is enabled by the availability of the cell-to-cell interaction penalty. This simple placement optimization algorithm is comprised of just a few simple steps and it is provided to show the feasibility of improving logic design yields by carefully optimizing cell neighborhood for manufacturability.

Constraints definition

In our implementation of the yield aware placement optimization algorithm, we start by introducing a set of additional design constraints that bound the extent to which the initial block placement can be modified. In fact we assume that an initial legal placement already exists and that the cell interaction penalty information can be used to improve yield by post-processing the existing placement in a sort of ECO mode. The most important constraint is therefore represented by the maximum offset that can be applied to a cell when moving it from its original location in order to avoid a critical abutment. This offset bound can vary from cell to cell and, in particular, we used the results of a static timing analysis to set tighter bounds on the registers and clock buffers as well as on all the cells on the critical path. This is in order to minimize the impact on the block timing as well as to reduce undesired clock skew. These bounds are translated into soft "don't touch" constraints for the placement optimization process.

Furthermore we also disallow moving a cell to a different row in order to minimize the probability of a significant increase of the routing congestion due to cell rewiring across rows.

Identification of frames

Cells in a row are typically separated by filler cells. In our algorithm implementation we first identify cluster of cells separated by fillers as shown in Figure 6. In fact as in our implementation filler cells (of different sizes) are specifically designed in order to be "yield-neutral" we can safely assume that no yield penalty is associated with the abutment of an actual cell with a filler cell. This assumption is rather strong as filler cells are required to provide or maintain an "ideal" neighbor in terms of layer density and feature pitch, which is not typically the case in most libraries. In fact, this requires filler cells to be very carefully designed having this particular constraint in mind and not just be defined as pure white space or de-coupling capacitors. If however a library of yield-neutral filler cell is available it is possible to partition the row in clusters (also called frames) of cells separated by fillers and limit the scope of the cell-to-cell interaction optimization to each frame which greatly simplifies the complexity of the proposed algorithm.

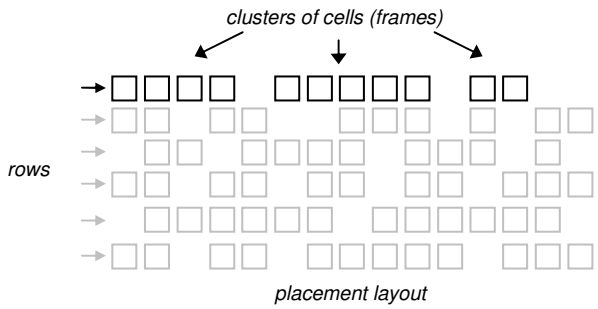

Figure 6 – Chip placement represented as rows of clustered cells separated by yield-neutral filler cells

Frame optimization

After each row is broken down into individual frames a certain number of different operations on frames can be defined that optimize yield by changing the abutment context of each cell. In our implementation we considered three types of operations:

- Cell flip: a cell is flipped along its vertical axis.
- Permutation: two cell's positions are swapped.
- Filler cell insertion.

All these operations are only executed within a frame, e.g. the permutation of two cells belonging to different frames is not allowed. This is in order to limit the perturbation of the initial placement. Nonetheless the complexity of exhaustively applying any of these operations is extremely high. For example, it is 2^N for flip operations, it is $2^N \times N!$ for combinations of *permutations + flips*. In order to achieve reasonable run time execution of our method we decided to further partition frames into sub-frames of equal length and to set a limit to the maximum length of a sub-frame. Since the total number of operations (i.e., *move+cell abutment penalty evaluation*) increases sharply when sub-cluster size > 4, as shown in Figure 7, we set the max sub-cluster size to 4 and then applied exhaustive *flip + permutation* transformations to every sub-cluster.

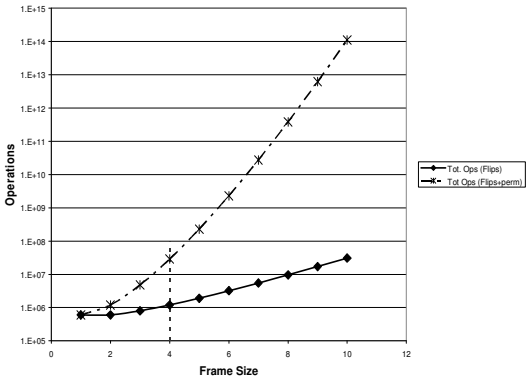

Figure 7 – Total no. of operations (flips only and flips + perm) as a function of frame size for a 1MEG design

The total number of possible operations to be considered is actually diminished by considering that some cells are actually fixed because of the timing constraints.

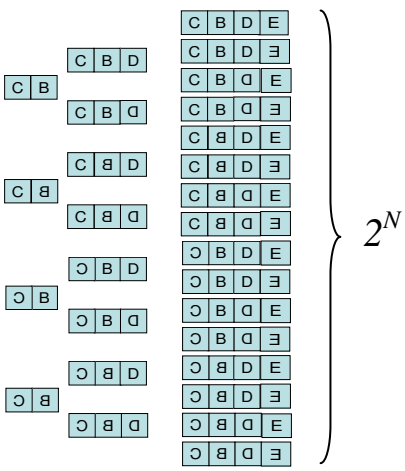

Figure 8 – All possible flips for frames of 2, 3 and 4 cells.

After sub-cluster optimization is completed there are still:

$$\sum_{i=1}^{N_L} MOD_4(length(C_i)) - 1 \quad (1)$$

cell-to-cell interactions that remain unoptimized, where N_L is the number of frames of length > 4 and C_i represents the i-th frame of length > 4. Therefore, the last step in our implementation consists of a linear time sorting of all such un-optimized interactions based on their abutment penalty followed by insertion of a certain number of yield-neutral filler cells starting from most critical interactions to least critical; As the budget of available filler cells for this final operation as well as the maximum cell displacement offset are limited, we also consider breaking down big filler cells into smaller ones (if available in the library). This creates additional available yield-neutral cells to remove further critical cell-to-cell interactions as shown in Figure 9.

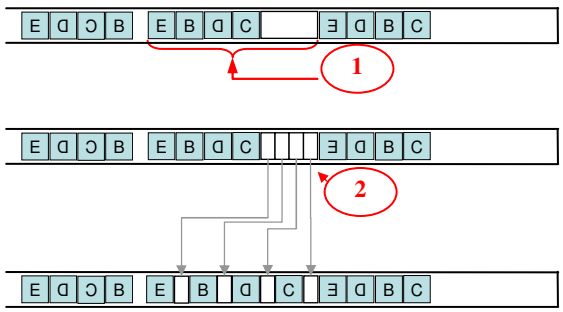

Figure 9 – Filler cell insertion to remove all remaining critical cell-to-cell interactions in the final step

6 Application Example

The described optimization algorithm was implemented as a stand-alone C-language program running on Linux We used a cell library characterization environment with lithography/etch hot spot extraction capability based on [8] to extract all cell-to-cell interactions as described in Section 4

6.1 Block layout model

The data loaded from the initial placement file are used to map the standard cell to a data matrix. Each entry in the data matrix contains all the required data for a single cell, including cell topology, yield information as well as the cell specific optimization constraints.

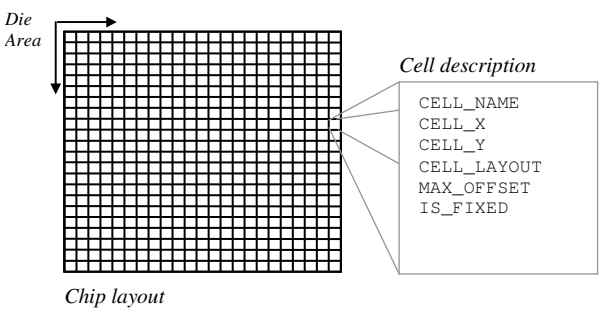

Figure 10 – Matrix representation of the chip layout

The matrix representation of the block layout is created by parsing the DEF file exported from Cadence Encounter GPS [14]. All the optimizations described in the previous sections are performed on the chip data model and a new DEF is recreated with the optimized placement at the end of the optimization. The optimized DEF can be re-imported in Encounter for final routing and timing verification.

6.2 Results

The yield driven placement optimization algorithm has been tested on an Intel 8051 synthesizable microcontroller with a 4 clock-bus cycle, and parameterizable ROM and RAM address ranges. The design has been mapped on a

90nm standard cell library that was previously characterized for yield thus generating a .pdfm file which included all the cell-to-cell interactions yield penalties. The total count of equivalent gates after technology mapping and initial placement is about 20,000. The layout snapshot in Figure 11 shows the synthesized logic portion of the design. The cells highlighted in yellow were moved with respect to the initial placement. in order to reduce the yield loss caused by critical cell-to-cell interactions. The yield improvement per Million Equivalent Gate is 0.7% starting from a baseline yield of ~98%. This means that 30% of the total functional yield loss of this block can be actually recovered by the proposed yield improvement algorithm.

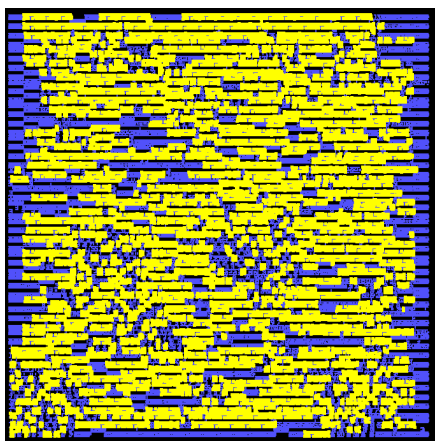

Figure 11 – Layout snapshot of the cells involved in the optimization process (yellow areas)

After re-importing the optimized placed data-base in Encounter the design has been successfully routed and the design has been successfully verified to meet all timing and DRC/LVS constraints.

```
Block features:
die
area: 334880 335790
chip rows: 91
total cells instance count: 5418

Memory usage (KBytes):
library: 348.66
chip: 149.21
Tot.: 497.87

Optimizing chip with algorithm n 5...
Optimizing cell frames ...
  Total cells modified: 2599
  Optimized frames: 1198
  Average frame size: 2.169449
  Max frame size: 11
  Min frame size: 2
  Critical path (frozen)cell count : 32
  Critical path considered 96 times during optimization
End frame optimization.

...optimization terminated.

---- Chip data --------------------------------
Chip yield before optimization:          99.927%
Ideal yield on placement:                99.959%
Max potential yield improvement:          0.031%
Chip yield after optimization:           99.950%
Total Yield improvement:                 0.023%
-----------------------------------------------
Saving chip description to file 'new_placement.def'...
...new description correctly saved.
```

Figure 12 – Optimizer Log File

7 Conclusions

A yield driven methodology to optimize synthesized logic block placement in order to avoid systematic yield loss caused by lithography related hot spots has been presented. An example placement optimization algorithm has been implemented in order to verify the feasibility and effectiveness of the proposed method. The results obtained on an application example demonstrate the feasibility of the method and showed also very encouraging results in terms of potential yield loss recovery, as above 30% of the total functional yield loss of the sample block could be successfully recovered with the proposed technique.

8 References

[1] A. J. Strojwas, “Conquering Process Variability: A Key Enabler for Profitable Manufacturing in Advanced Technology Nodes”, Keynote Speech, IEEE International Symposium on Semiconductor Manufacturing, ISSM 2006, Sep. 2006

[2] K. Keutzer, K. Kolwicz, and M. Lega, "Impact of library size on the quality of automated synthesis," Proc. of ICCAD, 120-123, 1987.

[3] Ken Scott and Kurt Keutzer, "Improving cell libraries for synthesis," Proc. of CICC, 128-131, 1994.

[4] A.J. Strojwas, S.W. Director, "VLSI: linking design and manufacturing," IEEE Spectrum, 24-28, Issue 10, Oct. 1988.

[5] P.K. Nag, W. Maly, "Hierarchical extraction of critical area for shorts in very large ICs," Proceedings of International Workshop on Defect and Fault Tolerance in VLSI Systems, 19-27, Nov. 1995.

[6] H.T. Heineken, W. Maly, "Manufacturability analysis environment - MAPEX," Proceedings of Custom Integrated Circuits Conference, May 1994.

[7] D.J. Ciplickas, X. Li, A.J. Strojwas, "Predictive yield modeling of VLSIC’s," 5th International Workshop on Statistical Metrology, 28-37, 2000.

[8] °YRS User’s manual”, PDF Solutions, Inc.

[9] Guardiani, C.; Dragone, N.; McNamara, P.; “Proactive design for manufacturing (DFM) for nanometer SoC designs” Custom Integrated Circuits Conference, 2004. Proceedings of the IEEE 2004, 3-6 Oct. 2004 Page(s):309 - 316.

[10] Kibarian, J.; Guardiani, C.; Strojwas, A.J., “Design for manufacturability in nanometer era: system implementation and silicon results”, Solid-State Circuits Conference, 2005. Digest of Technical Papers. ISSCC. 2005 IEEE International, 6-10 Feb. 2005 Page(s):268 - 269 Vol. 1

[11] Sechen, C.; Sangiovanni-Vincentelli, A., “The TimberWolf placement and routing package” IEEE Journal of Solid-State Circuits, Volume 20, Issue 2, Apr 1985 Page(s):510 - 522

[12] H. Eisenmann and F. M. Johannes, “Generic global placement and floorplanning,” in Proc. ACM/IEEE Design Automation Conf., San Francisco,CA, 1998, pp. 269–274.

[13] Gi-Joon Nam; Reda, S.; Alpert, C.J.; Villarrubia, P.G.; Kahng, A.B., “A fast hierarchical quadratic placement algorithm” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 25, Issue 4, April 2006 Page(s):678 – 691

[14] “Cadence SOC Encounter User Guide”, Cadence Design Systems.