# Enhanced Layout Optimization of Sub-45nm Standard, Memory Cells and Its Effects

*Seung Weon Paek*, Dae Hyun Jang*, Joo Hyun Park*,*
*Naya Ha*, Byung-Moo Kim*, Hyo Sig Won*, Kyu-Myung Choi*,*
*Kuang-Kuo Lin[‡]*
*Simon Klaver[¤], Shobhit Malik[¤], Michiel Oostindie[¤], Frank Driessen[¤]*

**Samsung Electronics, Korea*
*[‡]Chartered Semiconductor Manufacturing, Inc*
*[¤]Takumi Technology Corp., Santa Clara, CA*

## ABSTRACT

Automatic layout optimization is becoming an important component of the DfM work flow, as the number of recommended rules and the increasing complexity of trade-offs between them makes manual optimization increasingly difficult and time-consuming. Automation is rapidly becoming the best consistent way to get quantifiable DfM improvements, with their inherent yield and performance benefits for standard cells and memory blocks. Takumi auto-fixer optimization of Common Platform layouts resulted in improved parametric tolerance and improved DfM metrics, while the cell architecture (size and routability) and the electrical characteristics (speed/power) of the layouts remained intact. Optimization was performed on both GDS-style layouts for standard cells, and on CDBA (Cadence Data Base Architecture)-style layout for memory blocks. This paper will show how trade-offs between various DfM requirements (CAA, recommended rules, and litho) were implemented, and how optimization for memories generated by a compiler was accomplished. Results from this optimization work were verified on 45nm design by model and rule based DfM checking and by wafer yields.

**Keywords:** DfM kit(s), DfM validation, Trade-off, CAA, Recommended rule, litho, primitive cell(s), leaf-cell(s), Memory compilers.

## 1. INTRODUCTION

Design for manufacturability (DfM) is one of the foremost concerns in the semiconductor industry these days. There is an increasing need for layout optimization which is mainly focused on reducing systematic and random defect vulnerability of the design layout. Some attempts to improve the index of DfM-friendly design have been made with decreasing systematic variation of the design by analysis of optical proximity correction (OPC) and OPC rule checking (ORC) result or by avoiding particle defects using critical area analysis (CAA) [1]. It is difficult to combine two or more fixing methods because it is hard to determine the most effective way of achieving process friendly layout when multiple rules conflict during the optimization.

In this paper, we present an automated DfM-aware optimization method for standard cell and hierarchical hard-IP blocks such as the Memory blocks. The proposed method finds the most optimized layout of a standard cell or Memory blocks by considering the analysis results from CAA, litho simulation kit (LFD) and manufacturing checking deck (MCD) at the same time [2]. We are showing early results in Silicon that confirm improved Memory blocks yield as measured by functional performance build-in self tests (BIST).

## 1.1. STATUS of LIB DfM VALIDATION & Its LIMITATIONS.

### 1.1.1. Case of Standard Cells

In general, since the start of the 90nm technology node, it was common to perform library validation using MCD kits (DRC-type preferred rule counting) that were able to detect violations of preferred rules in the layout [3]. Subsequently, sign-off was done based on LFD kits (litho simulation, using OPCed layout) [4]. From the 65nm technology node onwards, CAA scoring was added, to detect hotspots caused by random particles, in addition to use MCD and LFD kits. From 90nm/65nm technology a design methodology addressing DfM considerations was available as a manual flow.

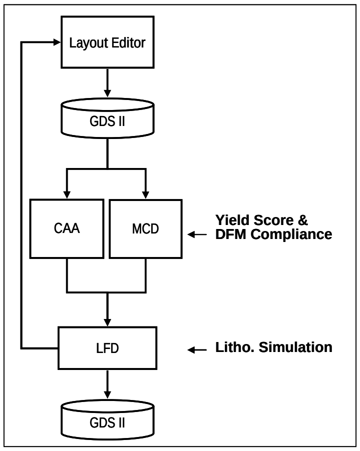


Figure 1.1.1: A general overview of Lib. validation flow manually.

† LFD : Litho Friendly Design (Enable to detect Litho hotspot using OPC model)
† MCD : Manufacturing Checking Deck (Enable to detect hotspot by preferred rule with scoring)
† Preferred rule : Recommend Rule (Guideline with priority for layout modify within design rule )
† CAA : Critical Area Analysis ( Enable to estimate yield using static particle distribution )

The flow starts with drawing layout with design rules. Then, at the Yield score & DfM stage, the layout is modified for preferred rules. A problem at this stage is that for most preferred rule violations it is hard to distinguish whether the layout is fixable or non-fixable, e.g. due to physical layout limitations (insufficient room & preferred rule priorities). Decisions also need to be made about trade-off amongst preferred rules with same priority. Even though two preferred rules may have the same priority. In such a case, a more comprehensive approach is needed.

Another issue is with CAA (include Contact Hole doubling, wire widening etc.). At cell level, a problem is that there is a very little discernment in score between before & after applying improvements for CAA. The reason is that the generic characterization of CAA kits is to focus to predict yield on the chip level. So, this kit is more suitable for a large area (chip level) than for a small area (cell).

Finally, as sign-off, litho simulation (LFD) is able to detect litho hotspots caused by low process margin. Among detected hotspots, it can be expected that severe hotspots could impact to yield by bridging or pinching of patterns, which must be removed [5], even if the layout gets worse in terms of CAA or MCD scoring.

.

In the manual method for library validation, it is hard to be consistent and comprehensive in striving for both layout correctness and DFM quality It is normally beyond the judgment abilities for average layout designers to tackle the myriad of design tradeoffs and complexity.  Realizing this, we have to resprt tp automatic fixing flow [6] in library validation.

### 1.1.2. Case of Memory compiler.

In the case of embedded Memories, excluded bit cells, the situation is that so-called leaf cells can in principle be placed in any combination of neighboring leaf cells to make a large variety of Memory blocks through the compiler. Analysis of DfM quality by CAA, MCD and LFD therefore takes place on a large variety of Memory blocks. However, a manual flow to systematically improve the DfM results has significant limitations in current memory compiler and an auto fixing flow is critically needed.

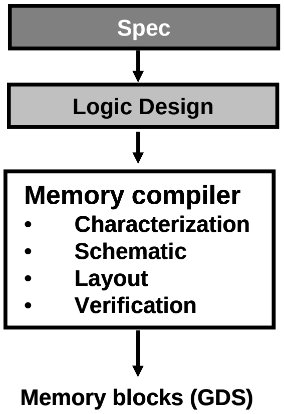


Figure 1.1.2: Flow scheme for the memory block generation

# 2. OPTIMIZATION OF AUTO FIXING FOR LIB VALIDATION.

## 2.1. Standard Cells

### 2.1.1. Optimize auto fixing flow with constraints

The following types of constraints should be considered or used in optimization by auto-fixing:

1. Architectural rules (for boundary rules and pin ports),
2. Performance rules to prevent gate length variation, and other changes in parasitic parameters for characterization,
3. Trade-offs between DfM kits (esp. on litho patterning)
4. Rules to allow future cell shrinks to  next process half-node,
5. Special requests by designers.

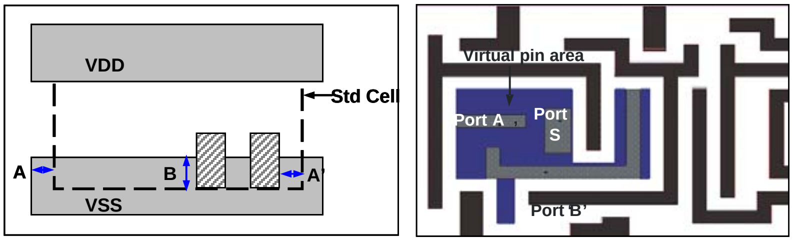


Figure 2.1.2: Constraint examples (Architectural rules).

Automatic optimization must take architectural rules into account (boundary conditions and rules for pin ports). Boundary rules were implemented by preventing movement of via landing areas in pin metals[7], as well as preventing movement of metals around virtual pin areas (see figure 2.1.2). This was done to enhance routability of cells when used in the place-and-route (P&R) step. To minimize gate length variation, ACLV rules (Across Chip gate Length Variation) were enforced for stable parametric characterization. Additional rules were also implemented to allow future migration, to the next process (half) node.

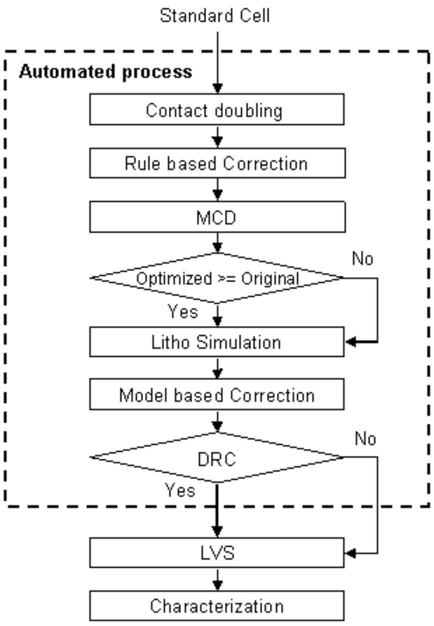


Figure 2.2.2: Optimize auto fixing flow with constraint

As described so far, the proposed method consists of contact doubling, rule based optimization and model based optimization. In order to complete such a complex job in a reasonable time, the optimization should not be done in a manual way but automatically. Figure 2.2.2 shows the flow chart for the overall layout optimization method for standard cells we developed.

The starting point for this flow assumes that the input primitive cells were designed completely by conventional design methods. The layout fixing starts with applying contact doubling first. At this stage, polygons are moved and extended to create enough area for an additional contact doubling. Second, rule based optimization is performed. In this stage, most of preferred rules are applied to the standard cells. The fixing at this point is focused largely on the development of patching layouts to meet required preferred rules as much as possible. After this step, MCD is run to confirm whether the correction brings manufacturability improvement of the cell or not and to move the cell to the next step. If the yield score of a cell improved as we expected, the cell passed through two fixing phases (Contact Hole doubling and preferred rules) would be transmitted to the next step. But if the predicted yield of the cells became worse than that before applying the correction, the original cell will be a source of the following lithography hotspot removal.

Litho simulation [8] will run on cells conveyed from the previous step. To execute lithography simulation, the model generated by the specific FAB where the product chip will be processed is needed. Hotspots are identified from extracted litho contours through process window and all of these hotspots should be fixed as part as the surrounding permits. Since there still are constraints like freezing the location of pin/ports, the correction is done within the limits of those constraints by the optimizer.

## 2.2. 45nm Memory Blocks: Hierarchy-preservering DfM Optimization

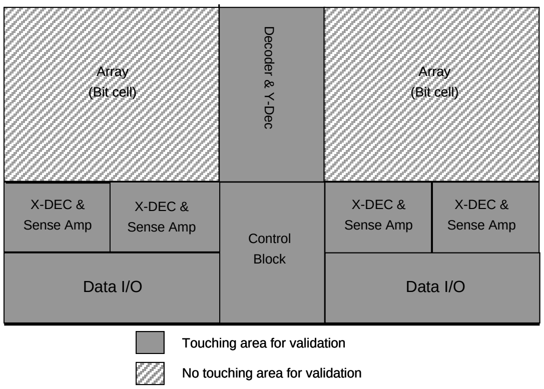


Figure 2.2.1: Modifiable area in the general memory block for the validation.

Although having a more complicated design-flow than std. cells, memory blocks are optimized for the DfM while maintaining their hierarchy. Furthermore, certain cells such as the bit-cell should not be touched because it is very sensitive in terms of electrical performance for any small modification of its layout.

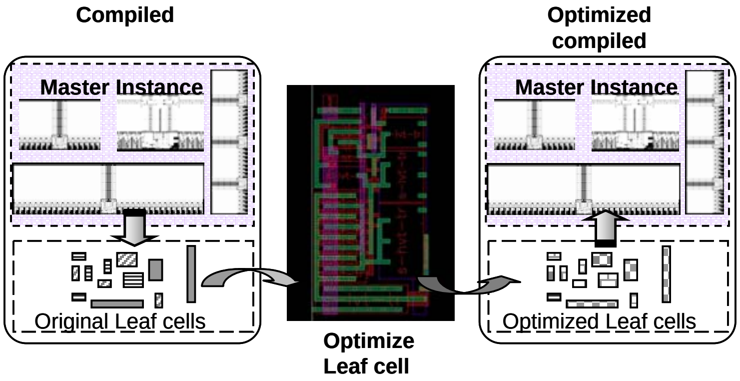


Figure 2.2.2 Concept of DfM validation for the Memory block
Extract leaf cells (left), Optimize leaf cells (center), Replace optimized cells (right).

Optimization of the ‘non-frozen’ part of the layout is done by a hierarchical flow that analyzes the hierarchy and determines which extracted leaf cells from a master instance of various generated memory blocks can be modified.

Optimization for DfM is then done similarly to what is described above. Preferred rules were enforced across all layers below M1 by the 2D optimizer with edge adjustments. The hierarchy of the layout result was identical to that of the input and the layout is compliant with usage in the commonly used Cadence Database Format (CDBA).

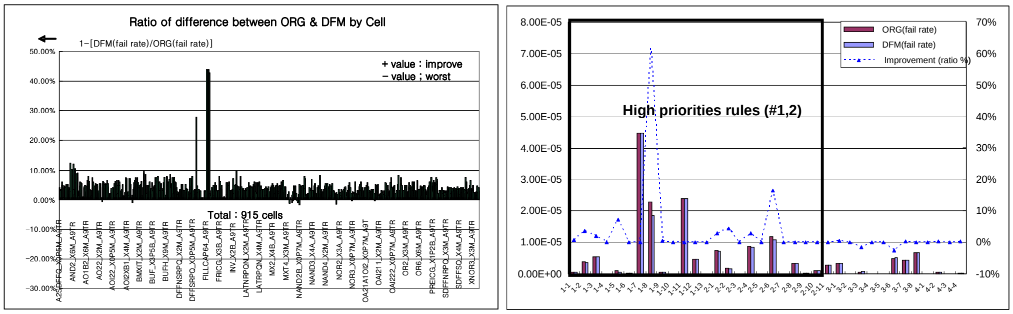
# 3. ASSESSMENT RESULTS OF DfM VALIDATION ON 45nm LIB.

## 3.1 RESULTS ON STANDARD CELLS

### 3.1.1 DfM Scoring and Electrical Verification

We verified effectiveness of the auto-fix flow on a library of 915 45nm cells. The three steps (contact doubling, preferred rule optimization, and litho based optimization) were used. Contact doubling only added 18 contacts to the whole library. This number was low mainly because for this tape-out we opted for the “fixed port” methodology which means that virtual pin port area is treated rigidly; in other auto fixing setups we support more flexible control of the virtual pin area that results in more DfM improvements. The average cell DfM score (fail rate) as calculated by MCD (preferred rule count) improved by Avg. 4.31% (range: 43.89% ~ -1.79% on all cells).

Of the 915 cells, 899 cells improved DfM score, and 16 got worse. The 16 cells got worse because of litho hotspot fixing, which resulted in less preferred rule implementation. When DfM score is sorted on preferred rule priority (figure 3.1.1, right), we could see that most preferred rules on the left (high priority) improved, at the cost of reducing some preferred rules on the right (low priority). Some rules did not improve but these relate to Contact Hole doubling and that they do not improve here relates to the choice that ports were frozen. Finally, it was confirmed that no litho hotspots were present on the standard cells under current litho process.



(a) DfM score by each cells. (b) DfM score by preferred rule

Figure 3.1.1 DfM score after auto fixing flow on 915ea cells.

We also verified the electrical behaviors and found that the maximum difference is 21.4ps on transition & max 16pW on leakage. These numbers were within our tolerances [9]. Besides that, no changes were made in ports of cells, so routability was not affected.

### 3.1.2 Example results of AUTO FIXING.

After optimization, standard cells were found LVS and DRC clean. LFD and MCD decks were run to verify DfM scoring statistically, and a lot of layouts were inspected in detail by expert designers to verify the proper trade-off between conflicting preferences. Examples of trade-off are shown in figure 3.1.2 at locations A and B. At location A, an RX spacing of priority 1 won over a preference to widen RX with priority 2. At location B, PC spacing with priority 1, pushes down an end cap (size priority 2), but to prevent this from reducing, the RX (+gate) area below is also pushed down, and leads to a reduction in PC-RX space at priority 3.

We confirmed that all high priority rules improved after auto fixing, and that worsening (trade-off) with low priority rules was minimized..

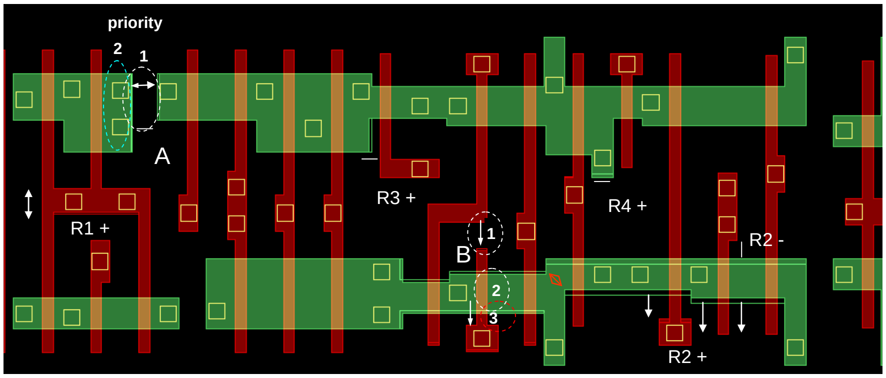


Figure 3.1.2: Layout status in the one cell after auto fixing.

## 3.2 SIMULATION & SILICON YIELD RESULTS ON MEMORY BLOCK

DfM and non-DfM variants of a 800um x 900um test design were placed on a multi-product wafer (MPW). The results were tested at the initial process stage in the manufacturing line. The reference block contained original Memory blocks (non-DfM) whereas the DfM block contained an optimized Memory blocks (DfM). Both blocks contained standard cells that were optimized and they had the same CMP densities from M1~M6. The area of the Memory blocks in this design is relatively high (almost over 40% occupied).

After chip implementation, chip-level sign-offs DRC/LVS and LFD (litho-hotspot) cleanliness were verified, MCD confirmed that scoring increased significantly by Avg. 8.44% (Fail Rate) on Memory blocks prior to performing the wafer experiment. And also, correct implementation of the highest-priority preferred rules has been confirmed improving MCD scoring on DfM-ed memory blocks.

| **DfM items** | **Reference block** | | | **DfM block** | | |
|---|---|---|---|---|---|---|
| | **Std. cell** | **Memory** | **Dummy** | **Std. cell** | **Memory** | **Dummy** |
| | **DfMed** | **NoDfMed** | **DfMed** | **DfMed** | **DfMed** | **DfMed** |
| **MCD score**<br>**(Memory blocks)** | **2.12E-04 (FR)** | | | **1.96E-04(FR)** | | |
| **Improvement** | **1.61E-05 (8.44%↑)** | | | | | |

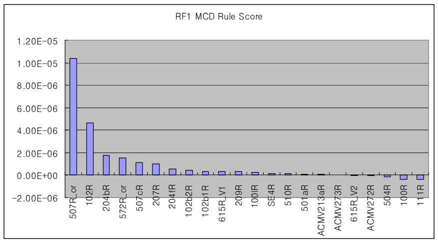


Figure 3.2.1: Differences in MCD scoring & design status between original & DfM optimized Design (left), MCD scoring improvement by rules on one of validated Memory block (right).

Two wafers each with 74 dies of both blocks were processed and measured. The chips were subject to electrical testing. One of the wafers showed almost the same yield data for the two types of blocks but the other wafer(#12) showed a significant yield difference of ~8% (from 79.7% to 87.8%) after BIST functional measurements to the advantage of the DfM-ed Memory blocks. Furthermore, electrical performance of the DfM block seemed to be slightly better (~2%) than that of the reference block.

The result discrepancy between the two wafers is under investigation but it may be caused by the fact that the DfM preferred-rule enforcement effectively improves process and yield robustness in wafer#12 through manufacturing process windows (systematic).

| WF | Chip | Yield | Good | BIST |
|---|---|---|---|---|
| #12 | Reference | 79.73 | 59 | 12 |
| #12 | DfM | 87.84 | 65 | 5 |

Table 3.2.1: Measured wafer results: Reference (12 dies failed by BIST), DfM (5 dies failed by BIST)

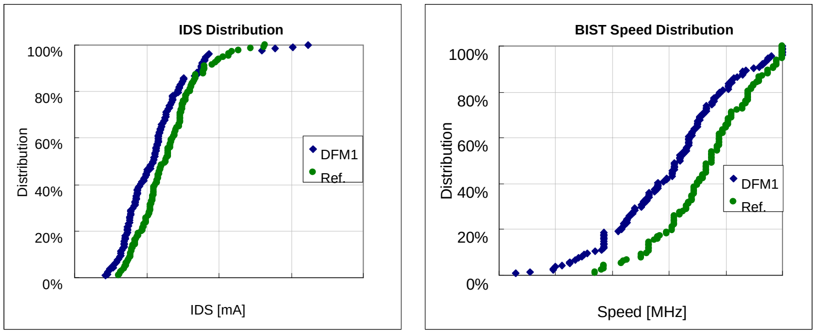


Figure 3.2.2: Results of IDS (source-drain current) distribution on whole dies (left), Results of Speed distribution on whole dies (right)

# 4. SUMMARY AND CONCLUSIONS

DfM validation includes analysis of random, systematic & parametric issues. A large amount of design effort is needed to look at preferred rules / CAA / Litho (LFD) simultaneously.

Therefore, an automatic fixing flow is a better approach than a manual method for getting a DfM optimized layout. A method was constructed with 3 stages: contact doubling (major function of CAA), preferred rule optimization with priorities, and litho-based (remove severe hotspot) correction [9]. We found this flow is best suited to address both electrical & process performance with DfM score.

We found that auto fixing improves turnaround time vs. a manual method and that it does not add any design penalty (no area increase or performance cost). Furthermore, we showed that auto fixing allowed good results with significant yield improvement of the Memory block as measured on Silicon. This yield impact is possibly related to enhanced tolerance to random and systematic process variations.

As a final remark, we realized that it is important to properly define preferred rule priority through test patterns using wafer data. Since wafer data is usually gathered at the initial stage of process setup, expectations on the time frame of process improvement need to be used in priority setting. Designer feedback must also be used to set proper tradeoff between preferred rule and (electrical) performance issues.

While we have a working auto-fixing flow that improves layouts design DfM quality, there are open questions:

* What should our target DfM score be for each cells?
* What should be the priority/weight of random/systematic/parametric factors?
* What is the contribution of yield loss or gain in standard cells to the yield loss or gain of an entire chips?.

We have shown that the auto fixing flow can improve on the long turn-around time of manual flow. In addition, it can consistently and comprehensively tackle the complex trade offs between preferred rules and litho issues. Lastly, yield improvements from our auto-fixing flow was verified with silicon wafer result..

## Acknowledgements

We would like to thank Takumi team at Takumi Technology Corporation for their hard work and effort in setting up the auto fixing flow for 45nm Lib. validation. In addition, we appreciate the helps from various Common Platform (CP) DfM teams from Samsung, Chartered Semiconductor Manufacturing, Inc. – our colleagues had contributed to the development of DFM kits used in this paper, as well as many valuable discussions and brain-storming.

## List of References

[1] Chiang, C.; Kawa, J. “Three DfM Challenges: Random Defects, Thickness Variation, and Printability Variation “ Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on, 4-7 Dec. 2006 Page(s):1099 –

[2] Robert Aitken, “ DfM Metrics for Standard Cells “ proceedings of the 7th International Symposium on quality Electronics Design (ISQED’2006).

[3] Maynard, D.N.; Runyon, S.L.; Reuter, B.B. “Yield Enhancement Using Recommended Ground Rules “ Advanced Semiconductor Manufacturing, 2004. ASMC '04. IEEE Conference and Workshop, 4-6 May 2004 Page(s):98 - 104

[4] Schellenberg, F.M.” Design for Manufacturing in the Semiconductor Industry : The Litho / Design Workshops, VLSI Design, 1999. Proceedings. Twelfth International Conference On 7-10 Jan. 1999 Page(s):111 – 119.

[5] K. Peter, R. März, and S. Gröndahl, “Litho-Friendly Design (LFD) methodologies applied to library cells,” *Proc SPIE* (2006), vol. 6349, no. 14.

[6] Riviere-Cazaux, L.; Lucas, K.; Fitch, J. , “Integration Of Design For Manufacturability (DfM) Practices In Design Flows.” Quality of Electronic Design, 2005. ISQED 2005. Sixth International Symposium on, 21-23 March 2005 Page(s):102 - 106.

[7] Shi-Hao Chen; Ke-Cheng Chu; Jiing-Yuan Lin; Cheng-Hong Tsai, “ DfM/DFY practices during physical designs for timing, signal integrity, and power” Design Automation Conference, 2007. ASP-DAC '07. Asia and South Pacific, 23-26 Jan. 2007 Page(s):232 –

[8] Mark E. Mason “ DfM EDA Technology : A Lithographic Perspective “ Symposium on VLSI Technology Digest of Technical Papers, June(2005), page : 90-91.

[9] Carlo Guardiani, Massimo Bertoletti,Nicola Dragone, Macro Malcotti, and Patrick McNamara, “An Effective DfM Strategy Accurate Process and IP Pre-Characterization” DAC 2005, June 13-17, (2005).