Journal of VLSI circuits and systems

Journal of VLSI Circuits and Systems, ISSN: 2582-1458 Vol. 6, No. 1, 2024 (pp. 51-54)

RESEARCH ARTICLE

WWW.VLSIJOURNAL.COM

Smart Ways to Catch the Abutment DRCs at IP Level

Swanand Kulkarni, Ravi J N

NXP Semiconductors, Manyata Tech Park, Nagavara, Bengaluru, Karnataka, India

KEYWORDS:
DRC(design rule check),
Abutment DRC,
Standard Libraries,
Routing,
VLSI,
Layout,
PnR,
SoC(System on Chip)

ARTICLE HISTORY:
Received 14.09.2023
Accepted 20.10.2023
Published 24.11.2023

DOI:
https://doi.org/10.31838/jvcs/06.01.08

Abstract

While developing any IP layout internal DRCs (Design Rule Check) will be taken care of with the help of Foundry decks. But it will be important to catch the DRCs when such IPs sit next to each other. Such DRCs are called Abutment/Half DRCs. These DRCs must be clean, or else during the Placement and Route(PnR) stage of the Design, SOC designers will see surprising DRC errors, even though the cells were individually DRC clean. These DRCs must be caught early in the design stage. Otherwise, errors seen during Vt change/ECO change will cause delay on the Tape-out schedule and/or area penalty. Current innovation highlights Smart ways to catch such Abutment DRCs at the IP level. The main target is Standard cell IPs, but the idea can easily be ported to any other IPs.

Author e-mail: Swanand Kulkarni,@gmail.com Ravi J N @gmail.com

How to cite this article: Kulkarni S, Ravi JN. Smart Ways to Catch the Abutment DRCs at IP Level. Journal of VLSI Circuits and System Vol. 6, No. 1, 2024 (pp. 51-54).

Introduction

Standard cell libraries[2], [3] are necessary for any semiconductor integrated circuits(IC) that primarily contain logic cells like AND, OR, NOT…etc. that can be implemented using Complementary metal oxide semiconductor(CMOS) logic. Standard cells are placed in rows and they will be abutted with other cells as shown in Fig. 1. PnR tools[4] should be able to place them next to each other without reporting any DRC violations.

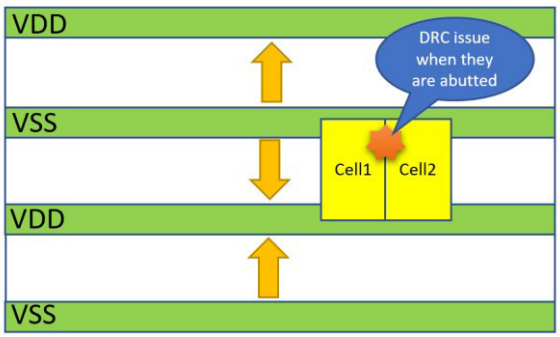

Fig 1: Standard cell placed in rows

While placing them user must make sure that they abut with each other without creating any DRC violations. For example, when Cell1 sits next to Cell2, the SoC designer may see DRC violations which will not allow us to manufacture that IC since the Metal layers were placed very close to the PR-Boundary of the cell. Such issues must be caught as early as possible.

The authors of[1] presented a way to catch the abutment DRC by just abutting 2 different Standard cells in All combination cells abutment(Brut-force) way, which will not guarantee 100% DRC Coverage.

The conventional/brut force way of catching such Boundary DRC issues using All-Combination abutment will require a huge run time and effort(all Mx, My, R180, R270, R90 flips/rotations must be considered) which makes it practically impossible to rely on keeping time to market as a critical item. For a Standard cell library comprising 1000 cells, the AllCombination run would take at least one day to complete all possible abutments. So came up with smart DRC.

In this paper, a smart way to catch such issues using Standard cell Track/Height and Technology dependent Smart DRC structures is presented. This approach will ensure that all possible DRC issues are caught in the cell level itself, inside the test area just by abutting Smart DRC structures around the Device Under Test(DUT). Such DRC issues might get missed during the All-Combaintion abutment structure creation. In Section II the basic idea behind the Smart DRC flow is explained. Section III will cover a detailed explanation of the steps involved in the Smart DRC flow. The key features of the methodology are explained in Section IV. The proof of concept and the applications of the methodology is covered in Section V and VI respectively.

Basic Concept

As shown in Fig. 2, this methodology is used to catch Abument/Boundary DRCs, not the DRC errors which might be seen within the cell. For example, Metal1(M1) may not be kept at the minimum Half

DRC[2] distance from the Left side of the PRBoundary. That will lead to DRC violation when that cell is abutted to a similar cell.

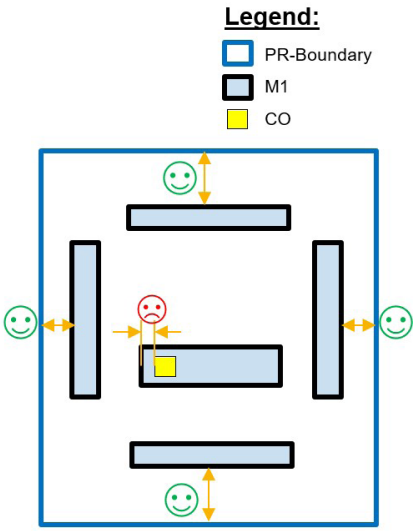

Fig. 2: Mainly Abutment DRCs are targeted

1. As shown in Fig. 3, flow starts with the following steps: 1. Identifying relevant CAD layers and associated DRC rules for Standard Cell IP.

2. Create a Dummy Cell layout that contains all the relevant layers identified in Step 1.

3. Creating Half-DRC structures on all 4 sides of the Dummy cell containing all the relevant layers identified in step 1 and which constitute all the DRC rules checks possible which are under consideration in step 1.

4. The layout of the Dummy cell (mentioned in step2) and layout of Half-DRC structures (mentioned in step3) are developed as described in the section <XX> 5. Test the Dummy cell along with abutted DRC structures.

5. Replace the Dummy cell with a real Standard cell (DUT) and check if any DRC issues are reported.

6. Correct the DUT for any DRC errors reported in step 6 and rerun step 6 until no DRC issues.

7. Repeat steps 6 and 7 for all the standard cells in the library.

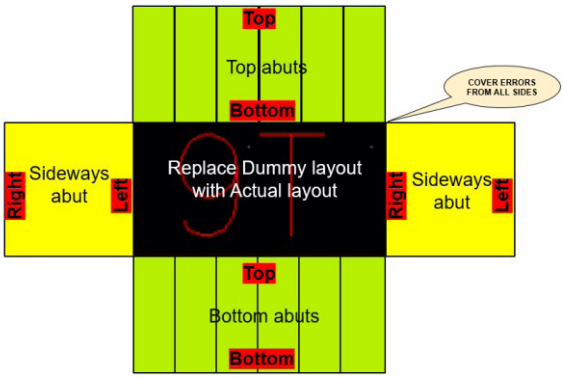

Fig. 3: Basic Concept

Detailed Explanation of the Steps involved in the Smart DRC Flow

Start with the technology-specific DRM. Pick only the Required checks/DRC rules on the relevant layers. For example, OD till M2 for Standard cells. Skip unrelated DRM checks like Contact enclosure on Metal (as it is not relevant for abutment scenarios)

Develop a Dummy layout to test the Smart DRC structures. That dummy layout will contain all the relevant layers of the Standard cell IP placed at the Half DRC distance from all 4 boundaries (Top, Down,

Right, and Left) of the cell such that it's DRC clean independently. At the same time, all the layers are placed in such a way that they just VIOLATE the HalfDRCs at All 4 boundaries when abutted with HalfDRC structures as shown in Fig. 3.

Development of 2 types of Half-DRC structures:

8. Side structures,
9. Top and Bottom structures

Again, Half-DRC structures must be DRC-clean independently. All the layers inside Half-DRC structures are placed in such a way that they are exactly at the Half-DRCs at all 4 boundaries.

Put the Half DRC structures around each of the Dummy Layouts and run DRC. Make sure that ALL the relevant DRC errors (as identified in step 1) are realized for all the relevant layers on all 4 boundaries(Expected output). Refer Fig. 4. Note that this is an absolute necessary condition for the SMART Abutment DRC flow to work to pre-create all the DRC errors targeted in the DUMMY cell layout along with the Half-DRC structure.

Once they are ready, instantiate the Targeted Standard cell layout in place of the Dummy Layout. Before instantiating, the Standard cell under the test should be DRC clean standalone. Put the Half DRC structures around each of the targeted cell. Run DRC on All cells

Abutted layout to catch any Half DRC issues within the DUT.

Below flow chart explains the same steps explained before in brief.

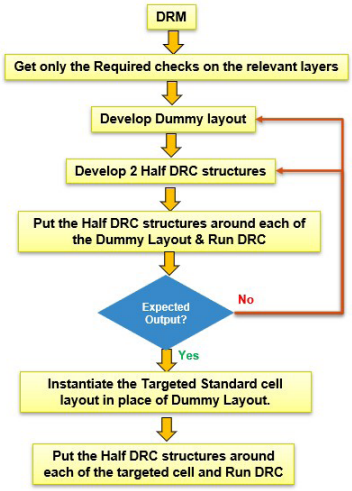

Fig 4: Flow chart of Smart DRC flow

Key Features of the Methodology

1. Smart DRC flow has 2 main steps:

   a. Develop Smart DRC structures and abut them around the targeted IP cells:

      One time preparation for any given technology(effort required: less than a week).

   b. Run DRC by enabling ONLY the relevant rules: As shown in Fig. 1, mainly Abutment DRCs are targeted to save DRC run time So, up to 92% run time-saving compared to the Brut force method of Abutment checks.

3. This flow can be easily adapted for any architecture of the library(like 7Track, 9Track, 10Track…) for any given technology. Also, Multiple height topologies are taken care of.

4. Easily portable flow for other kinds of IPs like Memory, IO

5. Dependency on the Brut force Half DRC flow creation is avoided, as the Layout designer itself can do better!!!

6. Run time reduction by removing irrelevant DRC rules w.r.t. IP:

While developing the Smart DRC structures, users can simply ignore some obvious DRC cases that can never come at the targeted IP level. This will save a lot of runtime.

An example is shown in Fig. 5 w.r.t. Metal1 DRC. At the standard cell layout level, we never see such a Metal1 shape of huge width 'X'um(almost 5 times the minimum width). So this M1.S.10 spacing error can be ignored.

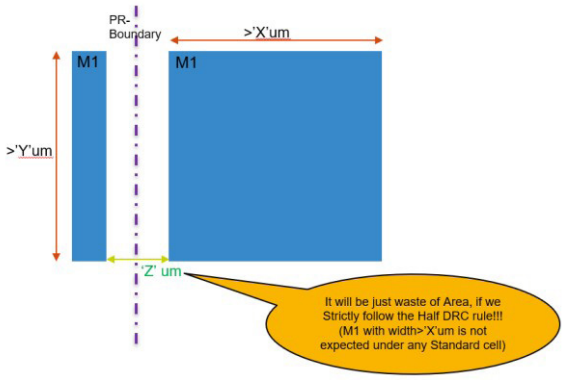

Fig. 5: DRC rule: M1.S.10: Space [at least one metal line width > 'X' um and the parallel run length > 'Y'um] > 'Z'um(minimum spacing rule for M1)

6. DRC errors caught per Structure: Each Smart DRC structure is developed keeping specific Layer-related DRC issues in mind. An example from 28nm technology is shown in Fig. 6:

Sl. No.	Smart DRC Structure Name	Layer Specific Errors Covered
1	abut_1 (top & bottom)	M1-M2(All Hor.), UHVT_N/P, PO, PO & CPO(CPO.R.2): Lg=40nm
2	abut_2 (sides)	M1(Ver.), M2(Hor.), M3(Ver.), NW, NP/PP, PO, V1-V2, CPO
3	abut_3 (top & bottom)	M1-M4(All Ver.), NP/PP, NW, CPO
4	abut_4 (sides)	M1(Ver.), M2(Hor.), M3(Ver.), OD, CO
5	abut_5 (top & bottom)	M1-M3(Hor.), OD, V1-V2
6	abut_6 (sides)	M2(Ver.), M1(Ver. & Hor.), M3(Ver.), UHVT_N/P, NP/PP, V1-V2
7	abut_7 (top & bottom)	PO & CPO(CPO.R.2): Lg=35nm
8	abut_8 (top & bottom)	PO & CPO(CPO.R.2): Lg=30nm
9	abut_CPO (sides)	CPO(CPO.R.7)
10	abut_ForCenterPWRvia (top & bottom)	V1-V2
11	abut_OD_G1 (sides)	OD(OD.S.3.1)

Fig. 6: DRC error and corresponding Smart DRC structures used

For example, 'abut_1' smart DRC structure which will be placed on the Top and Bottom of the DUT which will cover DRC errors w.r.t. layers Metal1, Metal2, UHVT_N(Ultra Hight Vt for NMOS), UHVT_P(Ultra Hight Vt for PMOS), PO(Poly), CPO(Contact Poly) layers

7. Advantages

   a. Smart way to catch the Abutment DRCs at the IP layout design stage itself.

   b. Run time advantage over Brut force method of Abutment checks

   c. Maximum possible combinations covered (Maximum accuracy): Made sure to mimic ALL(100%) the possible abutment DRC errors with a combination of Dummy cell with DRC Structures. This in turn helps in catching ALL the relevant DRC errors for the cell under consideration.

d. First-time right development, designs, and qualifications: The layout designer will get the knowledge and attitude of 1st time right while developing the layouts in the future. That will give 100% confidence to the SoC designer.

Proof of Concept

Many Standard cell DRC errors are caught during the cell development stage itself. With that successfully achieved Shift-Left over the period of time across all the technology nodes, including FinFet technology. Basic development learning from Planr technology nodes will boost confidence in FinFet technology.

Fig. 8 below shows the time comparison after using the relevant DRC rule. The result is for a library with an Area 8,00,000 um2, a total of 2000 cells targeted. By using only (63/4,000) DRC rules saved almost 92% of the Run time.

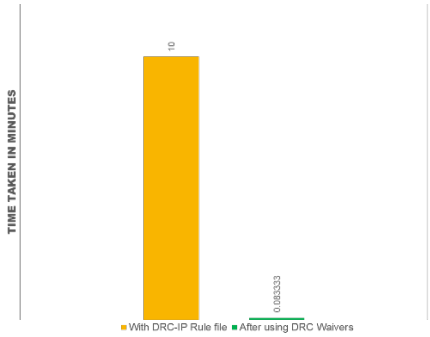

Fig 8: Time comparison after using the relevant DRC rule

Applications of the Methodology

First-time right development, designs, and qualifications for IP Layout, which will give confidence to the SoC designer. Mainly Standard cell IPs are targeted, but the idea can easily be ported to any other IPs with ease. Smart DRC flow can be used for P-cell development as well.

Conclusion

A smart way of addressing the Abutment DRC is presented which will ensure good IP layout verification coverage w.r.t. DRC rule before the PnR stage. Emphasized the Time-saving methods while developing the Smart DRC structures and also while running DRC. The proof of concept section just indicates the improvement seen in the WoW from the Standard cell delivery perspective.

References

[1] EFFICIENT STANDARD CELL ABUTMENT CHECKER - Juang-Ying Chueh, Chuck Tu-g - 2013 IEEE

[2] IC MASK DESIGN: ESSENTIAL LAYOUT TECHNIQU-S - C. Saint and J. Saint, New York:McGraw-Hill Professional, May 2002.

[3] STANDARD CELL LIBRARY DESIGN AND OPTIMIZATION METHODOLOGY FOR ASAP7 P-K - Xiaoqing Xu, Nishi Shah, Andrew Evans, Saurabh Sinha, Brian Cline, and Greg Yeric, ARM Inc., Austin, TX, USA - 2017 IEEE

[4] BLOCK LEVEL PHYSICAL DESIGN OF INTERFACING MODULE IN RISC CORE - By Siddalinga Aland, V. Venkateswarlu, Rohith B.R - IJITEE 2012