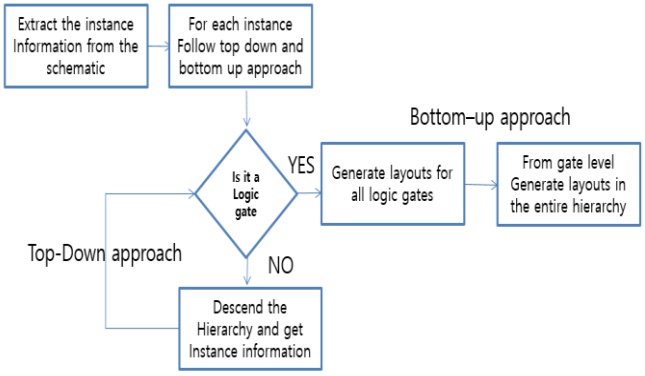


Figure Overview of automation flow for schematic with hierarchy

The layouts generated with this flow has all the pin level information and hence can preserve the connectivity information when the layouts are opened in Cadence Layout XL. Based on the connectivity information the users have the flexibility to change the placement of the blocks according to the requirement.

6. Experiments and Results

The proposed flow is implemented for both standard cell libraries and Memory compilers.

6.1 Ultra high speed Register file memory compiler

A Memory instance contains modules like Bitcell array, IO, decoder and Control block. Layout for IO and decoder is custom and smaller in terms of area. Automated floor planning techniques for memory compilers help in replicating these IO and decoder multiple times in a memory instance based on the configuration. SRAM 6T/8T Bitcell layout is often provided by the foundry PDK. Bitcell array is tiled using automated floorplan. Control block of the memory instance contains modules like Clock generator, Tracking, and other timing critical circuits for High speed memories. Since control block is critical in terms of area and timing, it often demands more time to generate layout. We have verified the proposed flow to automate the seed layout of control block of ultra-high speed memory. Time taken for handmade seed layout when all standard cell layouts are available is 2 weeks whereas, the flow can generate the seed layout of entire control block with multiple hierarchies including the standard cells in less than 10 minutes.

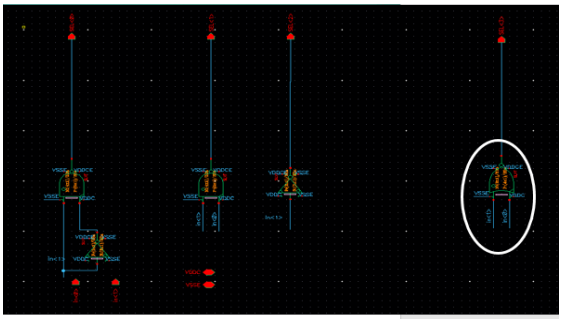


Figure 9. Schematic with multiple standard cells.

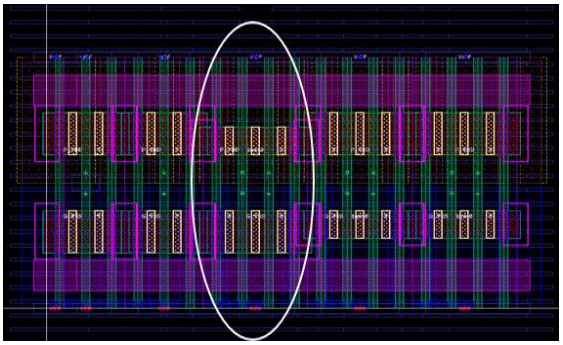


Figure 10. Optimum Layout of the schematic shown above.

The Fig. 4 shows the schematic of the sample module with the multiple standard cells at some hierarchy level of the control block of the chosen memory instance. The Fig. 5 shows the layout generated through this proposed flow.

6.2 Standard cell libraries used in SERDES and Image Sensor IPs

Standard cell libraries usually contain logic gates like NAND, NOR, INV, AND, OR and few sequential circuits like D-Flipflops with multiple variations based on drive strengths. IPs like SERDES, Image Sensors have most of their chip area occupied by standard cells. Standard cell libraries are optimized for leakage, power, performance and area depending on the specifications provided by the customer. Standard cell delivery kit usually contains netlists and layouts of cells along with their characterization data. Standard cell libraries design cycle involves multiple iterations of circuit design and Layout design to meet the required specifications. Layout automation assist to quickly develop libraries and characterize for Power, Performance and Area.

To demonstrate the impact of the flow we have chosen a Standard cell library with 800 cells. This library has

multiple variants of all the commonly used logic gates. Time taken to develop this library is very much less through this proposed flow.

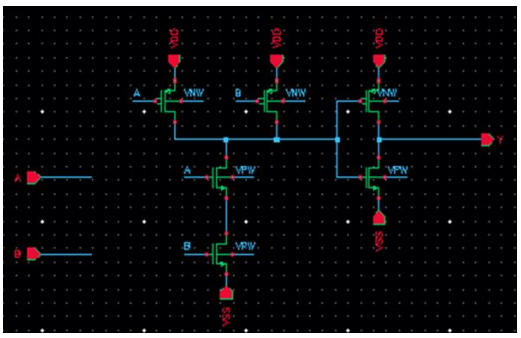


Figure 11. NAND-INV circuit schematic

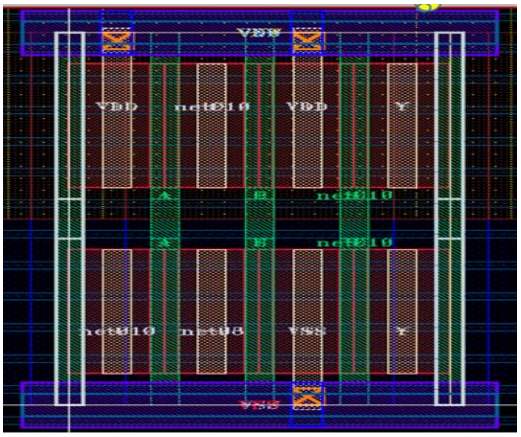


Figure 12. NAND-INV circuit layout

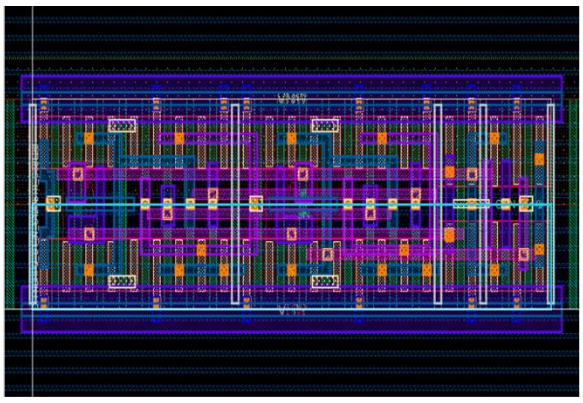


Figure 13. Handmade layout of a Flipflop with routing

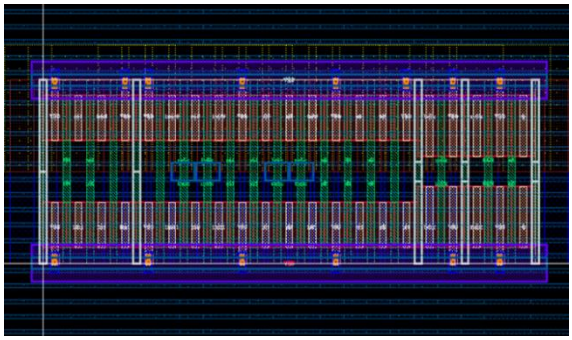
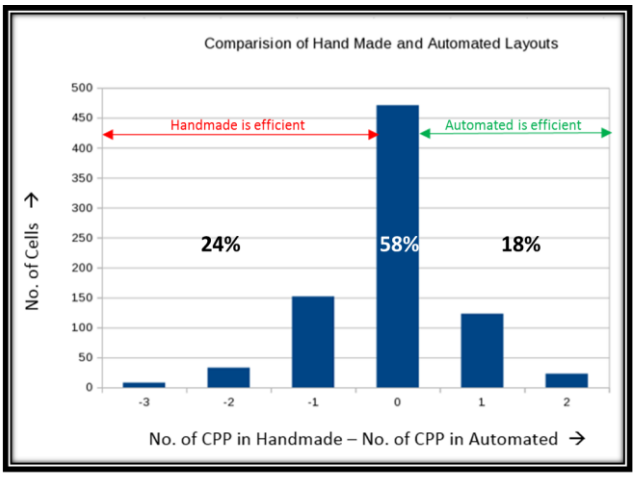


Figure 14. Automated layout of a Flipflop without routing

The Fig. 6 shows the schematic of AND gate in the chosen standard cell library. The Fig. 7 shows the layout of the corresponding AND gate generated through this proposed flow. The Fig. 8 shows the layout of D Flipflop generated manually. The Fig. 9 shows the layout of the same D Flipflop generated through the proposed flow.



- Cells considered for comparison - ~800
- Layout execution time with tool - ~10min

Figure 15. Comparison of Handmade layouts & Automated layouts

The graph in Fig. 15 indicates that 58% of the cells have same area as that of the manual layouts and 18% of the cells have lesser area when compared to that of manual layouts. Only 24% of cells have more area than that of manual layouts and since these are seed layouts, manual

intervention can optimize the area. The above stated data does not consider the routing complexity factor.

7. Advantages and Future Scope

7.1 Advantages of the flow

- User flexibility to develop the layout at any hierarchy with the desired width in standard cell pitch.
- The flow can be utilized for both CMOS and FINFET technology.
- User flexibility to customize the Pcells.
- Layout XL compatibility for custom placement of complex blocks.
- Improves the overall productivity by minimizing the cost in terms of resources and increasing the efficiency.
- This tool does not require any extra purchase of licenses

7.2 Scenarios

In this section, we will discuss few scenarios where the proposed tool becomes handy.

7.2.1 Quick area estimation of critical blocks with multiple modules

Area estimation of the critical blocks at the early stage of development cycle would provide an insight regarding the module that has to be focused to optimize the area. The proposed flow allows the user to adjust the aspect ratio of any module. Thus the proposed flow helps to optimize the layout in multiple iterations in very minimal time.

7.2.2 Standard cell Architecture analysis

Standard cell architecture definition includes identifying the maximum number of allowable metal tracks, power bus widths and positions. Automation can help in reducing the time taken for architecture evaluation by providing the user with the flexibility to change inputs and quickly characterize the cells. Once the architecture is finalized automation can tremendously reduce the time taken to develop any standard cell library.

7.2.3 First cut layouts of any schematic with complex and multiple hierarchies

The flow can descend the entire hierarchy and generates the layouts at each level. This helps the layout engineer to understand the modules to be focused to optimize the routing complexity and robustness. Eventually the user can spend more time on sign-off verifications like EM/IR estimation. Thus the flow minimizes the time taken for the iterative process of optimization of complex blocks for Power, Performance and Area (PPA).

7.2.4 PDK evaluation

For latest technology nodes the PDK from the foundry has to be evaluated for specific logic structures like NAND-NOR, NAND-INV and NOR-INV. This automation flow can help to quickly generate layouts with different standard cell architectures to analyze for required PPA. Then the best optimized cell structure is chosen.

7.3 Future scope

- Current Placement strategy of modules in schematic with complex hierarchy is random. The tool can be customized to take the user inputs to direct the placement of certain critical blocks.
- Extension of the flow to support Analog and Mixed signal layouts. This includes developing the placement strategy for Analog layouts that would requires perfect matching. Common centroid layouts are immune to cross chip gradients and provides best possible matching. Common centroid layout technique can be included in the flow.
- Concept of machine learning can be included in the flow to help in custom blocks which does not follow standard cell grid. In this procedure, the flow would be able to learn from the input handmade optimized layouts.

8. Conclusion

In this paper we have demonstrated the impact of the flow on the development time frame of the Standard cell libraries and Memory compilers. As discussed in the paper, the proposed flow minimizes the number of resources required and time-to-market. Thus minimizing the overall costs and increases the time frame available for layout optimization and validation.

In conclusion, the proposed flow can minimize the manual effort to generate seed layouts, perform analysis and Area estimation. The proposed flow can be customized according the user needs. In this paper, we have discussed the future scope of the flow to make it more generic and suitable for layout automation in various domains. Shrinking the required time to generate seed layouts at the early stages of design cycle will improve the productivity by developing more robust products in less time frame. This is most suitable in the current scenario of market which has very high demand for IPs in the domains like Mobile, 5G, Sensors and AI.

Acknowledgement

We would like to thank Parvinder Kumar Rana for his continuous support and help in this journey. We would like to thank Ravi

Kumar Gajula for his constant guidance in developing the automation technique.

References

[1] RABAEY, J. M.; CHANDRAKASAN, A. P.; NIKOLIC, B. Digital integrated circuits. [S.l.]: Prentice hall Englewood Cliffs, 2002.

[2] K. SHAHOOKAR AND P. MAZUMDER, "VLSI cell placement techniques," Journal, ACM Computing Surveys (CSUR), Volume 23 Issue 2, pp.143-220, June 1991.

[3] Tokinori Kozawa, Hidekazu Terai, "Reseach in Design Automation for VLSI Layout, " *IEEE Design & Test of Computers*, pp. 43-53, October 1985.

[4] "Cadence Virtuoso Skill Language Reference," Product Version 1CADV12.2, June 2016.

[5] "Cadence Virtuoso Parameterized Cell Reference," Product Version ICADV12.2, July 2016.