

repeater flops are required. With our solution we have designed a tool that

- Takes the floorplan information, distance from the source module to the destination and single flop drive strength as the input.
- Algorithmically finds out a tree structure with minimal number of flops.
- Generates the optimal functional RTL FUB that is ready to plug in.

The advantages of our solutions are:

- **Less Area:** This approach generates the least number of flops needed for a given source endpoint to destination endpoint. Thus area and flop delay are optimized.
- **Automated:** It needs minimal manual effort and it generates a functional RTL which is ready to plug-in. Thus the development is faster, easy to use and less prone to errors.
- **Scalable:** Adding new signals just requires adding the corresponding fields in the input to the tool and rerun the program again.
- **Fast development:** Using tool to generate automated RTL, aids quicker development & faster simulation readiness.

Optimized & automatic RTL, generated using the tool described in this document, is being intercepted in one of the Intel's IPs.

Acknowledgements.

We would like to thank Krishnamurthy B Venkataramana and Rajeev Gopalan for motivating and guiding us to create this system. We would like to thank Satheesh Pillai for letting us know the challenges back-end team is facing to close the timing on such module. We would like to thank Ryan Joseph who provided us more insights of the problems encountered in such module. We would like to thank Alex P Thomas, who is the first user of this tool, for providing us valuable feedback which made us improve our system. We would like to acknowledge the effort of Aman Aggarwal, who had helped us to develop few VBA macros for tool.

References

- [1] Eng Keong Teh; Mohamad Adzhar Md Zawawi; Mohamed Fauzi Packeer Mohamed; Nor Ashidi Mat Isa, "Access-Practical System-on-a-Chip Flop Repeater Insertion with a Meta-heuristic Technique", 2018, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems