Latch Based Designs: Advantages, Challenges and Solutions

Neelam Maniar
Design Engineer
Intel Corporation
Bangalore, India.
neelam.c.maniar@intel.com

Vasudev Anand B
Design Engineer
Intel Corporation
Bangalore, India.
vasudev.bangalore@intel.com

Shikha Subudhi
Design Engineer
Intel Corporation
Bangalore, India.
shikha.subudhi@intel.com

Hitesh Khandla
Design Engineer
Intel Corporation
Bangalore, India.
hitesh.khandla@intel.com

Gopalakrishnan Sadagopan
Design Engineer
Intel Corporation
Bangalore, India.
gopalakrishnan.sadagopan@intel.com

Abstract— **Latches provide better alternative to flip flops in high frequency design's performance due to its inherent advantages of time-borrowing. Transparency of the latches creates timing paths through the latches as if the latches are repeaters. This comes with challenges of converging the designs with huge number of cycles' paths depending on number of transparent latches in the design. The paper presents a unique way of inferring the type of latch based designs and its generic bottlenecks. It also provides the way to generate the constraints to guide the Synthesis and Place & Route (PnR) Tool which helps improve the design quality that aids faster convergence of design parameters. It further talks about the sequential loops commonly encountered in pure latch based designs, its associated challenges and solutions for Synthesis and PnR designs.**

***Keywords*—latch based designs, transparency, time-borrowing, sequential loop, high speed design, and optimization**

I. Introduction

In flip-flop (FF) based designs, the longest pipeline between the two sequential elements limits the frequency of design. Latch based designs inherently have benefits of time-borrowing and transparency. This allows unbalanced pipelines between the sequential elements without compromising on the frequency of the design. Transparency helps in propagation of data as if the latch were a repeater. This further helps in boosting up the frequency.

Dealing with transparency for design convergence is tricky. The timing critical paths of the design can have large number of cycles (this is different from multicycle paths' constraints usually encountered in FF based designs) if latches one after the other, remain transparent. It becomes essential to analyze and infer the bottleneck path segments in the critical paths and it may become time consuming. Each design block has its own set of challenges, few of which are generic and few are unique to the block. In convergence phase, all the aspects of design such as timing, power, area, circuit quality, physical verification aspects such as routing and cell congestion, pin density, etc. are targeted. Various reports specific to each aspect are analyzed to identify the design problems and thereby come up with set of user constraints. The generic user constraints can be in form of applying group paths, path margins, placement bounds, giving priority routing, applying keep out margins, applying MCMM constraints, giving one convergence aspect a priority over the other aspects for specific design portions (e.g. Multibit exclusion for timing critical sequential elements), etc. These constraints are applied for tuning and correlating the synthesis and PnR tool violations with the sign-off tool violations to the best possible extent. These solutions alone work for ~80 percent of the blocks. Often, unique solutions are needed for very high complex designs along with these generic solutions. For e.g., the latch based designs with sequential loops, the high speed multiplier design, clock domain crossing designs, etc. are unique issues and need specific addressing. Furthermore, in high frequency designs, where very small timing violation needs a fix, ECO phase could become longer if the above convergence phase is not given its due diligence.

The proposed approach aims at automating the setup timing convergence of very high speed latch based designs. Timing improvements aid in design quality improvement which in turn, helps improve power and area. It analyzes the existing set of timing reports, infers the generic issues in the design. Generic issues leading to non-optimal paths and creating timing violations are inferred through automated analysis. Once issues are identified, automated user constraints are suggested for the Synthesis and Place & Route Tool (PnR).

Sequential loops in latch based designs are commonly seen in Finite State Machines (FSMs) with control logic and arithmetic blocks such as divider. Sequential loop path is the one which starts from one sequential element, passes through at least one transparent sequential element and ends on the same sequential element it started. Loop setup margin is the difference in the arrival time at the endpoint of the loop before and after traversing the loop. Critical loop setup margin is the one which has negative loop margin. In Fig. 1, arrival time is checked at input of node 2 (X).

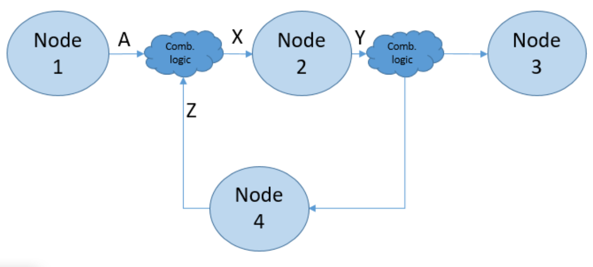

Fig. 1. Illustration of Sequential loop structure

Sequential loops (seq loops) create the run-time/accuracy trade-off as the timing models involving them, cause infinite delay propagation. There is a limitation in terms of identifying and fixing the seq loops in the design by standard implementation tools at the early stages of implementation design cycle in the default flow. Often, the fixes for such seq loops require high manual effort which is usually iterative. In this paper, we propose methods to identify seq loops (in standard implementation tools) early in the design cycle and optimize the logic, placement and route for the same. With same convergence effort in terms of work hours, traditional approach of convergence like adding group paths, placement bounds etc had non-optimal logic, placement and routing. Proposed method resulted in average improvement of approximately 95% in timing in both the blocks as compared to traditional approach.

II. Challenges and Solutions of Modelling Latch Transparency

The proposed work aims at automating the generic issues of the latch based PnR design convergence. The utility developed, samples the setup timing reports of the raw design, infers the generic issues in the design and suggests possible design solutions in form of user constraints. Various vectors of generic issues such as identification of long pole non-optimized logic impacting maximum total negative slack (TNS), identification of redundant buffers and inverters, defining right MCMM (Multi-Mode Multi-Corner) constraints, identification of scenic placements, R and C miscorrelations between the EDA tools and sign off tool, etc. are addressed. The automated utility outcome is inference of existing issues, which is otherwise a manual process. Following generic issues in the design, if any, are identified on negative slack paths through different utilities.

A. Global Identification of long pole non-optimized logic

For latch-based high-speed designs, there are often huge number of sequential elements on a single path with multiple levels of transparencies. The long-pole non-optimized logic between the two sequential elements, which contribute to maximum total negative slack of design, is identified from the timing reports. This is done by computing maximum number of negative slack paths of the setup report, impacted by specific pair of latches. Optimal number of combinational logic cells is calculated for a given PVT corner and targeted frequency. Only the sequential pairs with number of combinational levels between the two, greater than the calculated optimal threshold are identified as bottlenecks. By this, synthesis tool is enabled to further optimize the combinational levels. Solutions are suggested as user constraints in form of group paths, time borrowing, etc. Pipelines can be unbalanced in the design, but ones with huge levels of combinational logic are always the bottlenecks, where there is further scope of optimization.

B. Localized Identification of non-optimized logic and redundant repeaters

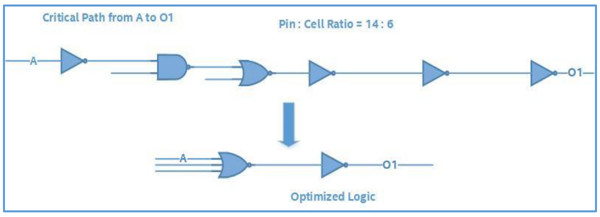

Fig. 2: Logically un-optimized circuit and possible optimized circuit solution

The Non-optimized logic between the two sequential elements of the negative slack paths is identified. All combinational logic cells placed in specified vicinity and having with a fan out of 1 between the two sequential elements are studied. If the ratio of number of pins to number of these logic levels (pins:cells ratio) is small, then these paths are identified as non-optimized. In Fig. 2, pins:cells ratio is 14:6, considering 2-input gates pin count to be 3 and inverter/buffer pin count to be 2. In such cases, it is possible to use cells with more input pins and reduce the number of cell levels along the path. Further, if number of repeaters in the path is greater than a specified threshold, path constraints are generated irrespective of the net fan-out. Synthesis tool is guided to duplicate the logic and reduce the repeaters along the path. Figure 1 shows 6 levels of combinational delay along the path from A to O1 that can be reduced to 2 levels

C. Optimal Cycle Time and MCMM constraints for synthesis

The selection of optimal cycle time for Synthesis and Place & Route plays a major role in timing convergence. The average miscorrelation of delay between the EDA tools and signoff tool on critical paths is identified. The cycle time for synthesis is scaled accordingly. For example, if the critical path delay in synthesis timing tool is 100ps and that in signoff tool is 105ps, then cycle time for synthesis is tightened by 5%. Further, optimal MCMM constraints are required for timing convergence in multiple voltage corners. This can be inferred through environment timing constraints scaling in different voltage corners on critical paths of the design. For example, the RC dominated path at the full-chip level is critical at high voltage corner over the low voltage corner. In this case, it is recommended to prioritize high voltage convergence over the low one.

D. Identification of scenic placements

A ratio of total actual distance traversed by a critical path and the minimum possible traversal distance between the end points of the path (actual:minimum ratio) is calculated. A higher ratio pinpoints a scenic placement to the designer. Further analysis and solution in this case, is left to the designer.

These utilities pinpoint the generic issues in the current design and suggests solutions independently. This process can take few iterations to fully resolve the generic issues of the design, specifically for high complex design, due to the onion-pealing nature.

III. Challenges and Solutions in Sequential Loops' Design Convergence

The standard implementation tools (Synopsys DC and ICC2 used for implementation), by default are designed such that the timing paths through transparent latches are considered as series of broken path segments between the latches. This restricts the tool to see the bottle necks caused due to transparency beyond first level of latches. It results in un-optimized design in terms of timing, power and other design parameters.

This can be overcome by switching on, advanced timing analysis (timing_enable_through_latches) which enables tool to see through multiple transparent latches. However there is a limitation to advanced timing analysis once it comes to sequential loops. Once the tool encounters sequential loop, it breaks loop by setting one of the latches in the loops as loop breaker latch. Timing analysis through this loop breaker latch is same as though advanced timing analysis is off. i.e., the timing paths through the loop breaker latch are considered as segments and transparency is disabled. This leaves the sequential loop paths un-optimized.

In this paper a methodology is proposed to identify loop paths, allowing tools to see transparent latches in the loop path, fix the loop margin early in the design cycle so that timing is better modelled resulting in better design. The proposed method is less susceptible to change in process parameters, design metrics, timing and floorplan constraints and reduces the manual effort for convergence significantly. Default flow led to 10K number of sequential loop slack violations in one of the blocks and 598 in another block from Synthesis and PnR flow, whereas, the proposed approach led to only 242 and 0 number of seq loop violations respectively.

Default flow of standard implementation tools disables the transparency in sequential loop path (by setting latch as "latch_loop_breaker") resulting in un-optimized design as discussed in earlier section. This can be resolved by guiding the tool not to consider the latches in the sequential loops for loop breaker latch using the command "set_latch_loop_breaker –avoid <all latches in seq loops>". This enables transparency through the latches forming loop to properly optimize the loops for timing and power.

IV. Results

A. Transparency

The individual convergence utilities were piloted in various high speed latch based design blocks. This led to average 52% reduction of TNS, 5.24% reduction of Cdyn and 2.38% reduction of leakage power. All the blocks piloted are complex designs in terms of performance, power and/or area convergence.

Design Block	% TNS Improve	% Cdyn Improve	% Leakage Improve	Utility
B1	22.99	15.9	-3.3	Local[1]
B2	36.82	3.1	3	Local[1]
B3	73.04	-1.5	-0.3	Local[1]
B4	82.26	1.43	1.23	Global[2]
B5	40.37	-	-	Global[2]
B8	22.99	13	2.57	Mcmm[3]
B9	36.82	1	-0.7	Mcmm[3]

Table 1: Timing, Cdyn and Leakage Gain with Individual Auto Convergence Recipes

Table 1 contains the result of auto convergence utilities. The individual utilities are short-named as below:

- Local[1]- Localized Identification of non-optimized logic and redundant repeaters
- Global[2]- Global Identification of long pole non-optimized logic
- Mcmm[3]- Optimal Cycle Time and MCMM constraints for synthesis

B. Sequential Loops

The proposed flow was run on two blocks which had high number of sequential loops with critical timing margin. The number of sequential loop paths with critical margin significantly reduced along with overall timing of the block.

a) Timing summary from timing signoff tool with default flow and with proposed flow:

Design Block	Leakage Improve %	WNS Improve %	TNS Improve %	Seq Loop converged %
B_A	8.5	79	89	97
B_B	7.5	51	99	100

Table 2: Design quality improvement with proposed method

Table 2 gives the timing, area and critical seq loops' improvement of two design blocks B_A and B_B with proposed method, as compared to default flow. The worst negative slack (WNS), total negative slack (TNS) and number of critical sequential loops were observed to be significantly less with proposed flow.

b) Comparison of a timing path with default flow and with proposed flow:

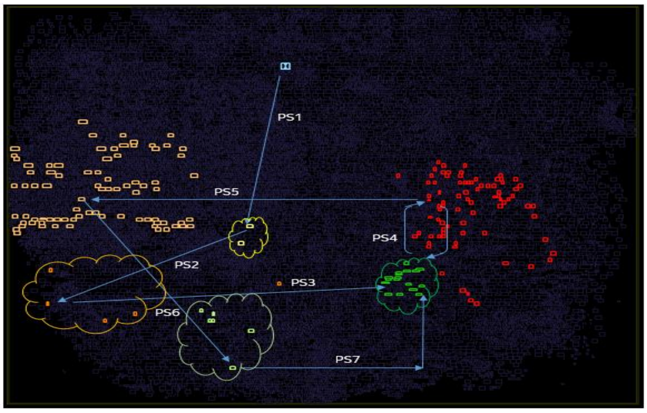

Fig 3.a) Seed placement of the latches in the loop path with distributed timing segments in default flow

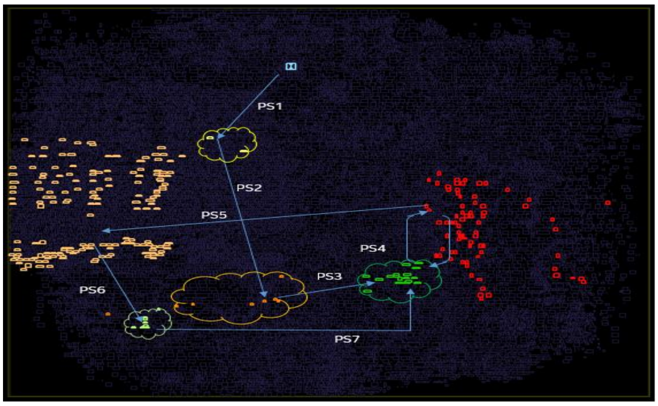

Fig 3.b) Seed placement of the latches in the loop path with distributed timing segments in proposed flow

Fig 3.a shows the seed placement of the critical path with complex loops (loops within loops) with the default flow and Fig 3.b shows the same with proposed flow. The timing path starts from a port and passes through multiple sequentials and incorporates a shorter loop path (Path segment 4 (PS4 -> PS4)) and longer loop path (PS4 -> PS5 -> PS6 -> PS7 -> PS4).

The seed placement was observed to be optimal in proposed flow as compared to default flow. Total path segment length traversed by the timing path improved by 17.5%.

Conclusion And Future Work

In this paper, various TNS reduction techniques such as Local[1], Global[2], Mcmm[3] with respect to transparency challenges and alternative flow for identifying and optimizing designs with critical sequential loops early in the design cycle are explored. The dynamic and leakage power gain in most of the designs come as a by-product of design quality improvement. These techniques attempt to fundamentally alter the optimization results from Synthesis and PnR flow. More utilities can be developed and combined together with these above ones, which target issues of a given process node design and targeted design parameters of the project. Some of these include, automation of global and localized congestion analysis of routes, pins and cells, automation of finding the priority routes, etc. The PnR flow can be evolved with the help of these utilities. Machine Learning and Deep learning algorithms which select the utility based on the Quality of results can be developed.

Acknowledgment

Authors would like to thank Sunilkumar T Bhat and Bayya Sarath Chandra for his valuable inputs in understanding sequential loops. Authors would like thank Shilpa Thakur, Shanthi Rangaswamy, Nikhil Saxena and Arpit Gandhi for a quality review of the paper.

References

[1] https://solvnet.synopsys.com

[2] Kris Tiri, Member, IEEE, and Ingrid Verbauwhede, Senior Member, IEEE, "A digital design flow for secure integrated circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (Volume: 25 , Issue: 7 , July 2006)

[3] Dimo Martev, Technische Universität München; Sven Hampel, Intel Germany, 47259 Duisburg ; Ulf Schlichtmann, Technische Universität München, "Synthesis-based methodology for high-speed multi-modulus divider", 2016 13th International Conference on Synthesis, Modeling, Analysis and Simulation Methods and Applications to Circuit Design (SMACD)