

Timing Closure of Memory Partitions for a Lower Nodes Technologies



Piyush Bhatasana

Abstract—Metal interconnects are used to make the interconnections between different part of the circuitry to realize any System on Chip (SoC) design. For the advanced process technologies, the metal interconnects affects the performance of the design. For nanometer process technologies, the coupling effect in the interconnect causes crosstalk and noise. These noise and crosstalk must be affect the operating speed of the design. This is most responsible candidate for the timing aspect of the design. Thus, the physical design and verification of the advanced process technologies should be include the effects of noise and crosstalk. If the timing of a design is not verified, then the design may not perform at the desired operating speed. The power and area are the other factors, that also to be consider with timing for a faster design. There will always be a trade-off between these three factors. Static Timing Analysis (STA) is one of the many techniques used by the designers to verify the timing of the design and also for closing the design with respect to timing, which is called as timing closure.

Keywords: Static Timing Analysis (STA), Noise and crosstalk, Setup and Hold check, Signoff, Engineering Change Order (ECO), DTA

I. INTRODUCTION

Very-large-scale integration (VLSI) is the process in which the creating an integrated circuit (IC) by combining hundreds of thousands of transistors or devices into a single chip. With the advancement in VLSI technology, there is a constant reduction in the feature size of VLSI devices (i.e. the minimum transistor size) [1-2]. The feature size decreased from about 0.25 m in 1997 to about 10 nm today [3]. Such a continual miniaturization of devices has had a strong impact on VLSI technology in several ways. One of those impacts is the timing analysis. As increase in the number of transistors per chip increases the rated speed or frequency at which a design is to be operated has to be met and the number of timing paths rises exponentially high due to multiple connections in the design [4]. The continuous decrease in feature size and corresponding increase in chip density and operating frequency have made exhaustive timing analysis a major concern in VLSI design. Hence, Static Timing Analysis has branched out as an entirely separate domain of expertise in itself in modern day System on Chip Design (SoC) [5-6].

Manuscript received on March 15, 2020.

Revised Manuscript received on March 24, 2020.

Manuscript published on March 30, 2020.

* Correspondence Author

Piyush Bhatasana*, Department of Electronics and Communication Engineering, Nirma University, Ahmedabad, Gujarat, India

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](#) article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

II. TIMING CLOSURE IMPLEMENTATION

The implementation is carried out in two stages, post IC Compiler (ICC) and ICC. In post ICC the design will be in are fixed in an incremental way. Synopsys Prime Time tool is used in this stage to analyse the timing paths and source the ECOs [2-3]. The timing path can be implemented in four different ways shown in Fig. (1). The name of the path are as below.

- 1 Flop to Flop Paths
- 2 Input to Flops Paths
- 3 Flop to Output Paths
- 4 Input to Output Paths

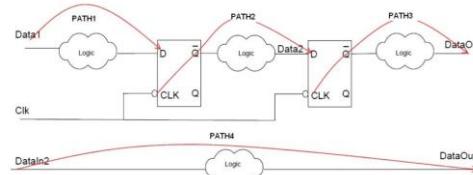


Fig. 1. Four types of Timing Paths

III. CLOCK TREE SYNTHESIS

Once the clock trees are built, it undergoes several buffer insertions and subsequent skew optimizations. An important point to be noted in the case of CTS is the use of clock buffers in clock path instead of normal buffers. Minimum pulse width is an important Design Rule Check (DRC) check performed on any design. It is required to maintain a minimum pulse width for proper functioning of the entire circuit. For the clock tree synthesis the following optimization is be done for the tree synthesis

- 1 Skew Optimization: The propagation delay is controlled by the size and location of the buffers.
- 2 Buffer sizing: A binary search algorithm is implemented to find the appropriate size of clock buffers through an iterative process since it affects downstream optimizations
- 3 Wire sizing: The size of wire is an important factor as it affects the power consumption and is prone to manufacturing issues.
- 4 Crosstalk and Noise: Any undesired or unintentional effects affecting the nominal operation of the chip is called noise. Crosstalk noise can be referred as unintentional coupling of activity between two or more signals. It is mainly caused by the capacitive coupling between two neighboring signals.

In nanometer devices, this noise can influence the functionality of the device or even the timing of the device. There are several causes for this noise in the chip.

The number of metal layers in the chip nowadays have been increased drastically, because of the technology shrinking. In addition, the wires that are used nowadays are thin and tall rather than wide and tall because of which capacitance will be more between two neighboring wires. The standard cell count is also increased in latest technologies that increases the congestion and causes lot more interactions. Higher frequency designs have faster edge rates, which will cause more current spikes and greater coupling impact. Noise margin for the designs is little because of low supply voltage shown in Fig. 2.

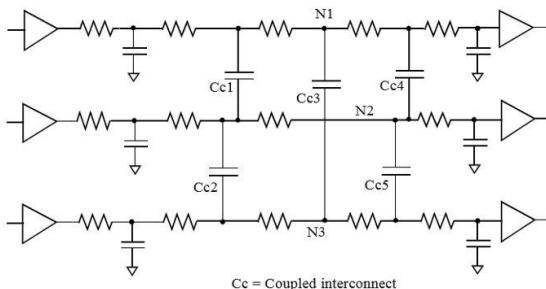


Fig. 2. A Coupled Interconnect

IV. PHYSICAL DESIGN

A. Partitioning

The process of splitting up the entire circuit or system into smaller subsystems or modules is referred to as partitioning. Floor planning the process of arranging the location of hard IPs or macros, the external ports and sub circuits or modules is referred to as Floor planning. Placement the process of finding and determining the spatial location of each cell within a particular block is referred to as Placement. Clock Tree Synthesis The process determining the buffering, gating and routing of the clock signal to meet the required clock skew and clock delay estimates is referred to as CTS. Routing the process of allocating resources for interconnections between different metal layers and routing tracks in channels is referred to as Routing. Timing Closure The optimization of performance of the SoC is carried out by specialized placement and routing techniques and implemented ECOs is collectively referred to as timing closure.

B. Inputs Required for Physical Design

The Prime Time Hyper-Scale hierarchical analysis method analyzes the block-level and top-level portions of the design using separate runs and accurately handles the timing interfaces across hierarchical boundaries. The discrepancies seen in the correlation are removed using context. Context is based on a Synopsys technology called HYPERSCALE. The context information is provided by the full chip level team. Updating the design in full chip level takes more time than updating the design in block level. If the violation is made same in block level and full chip level, then a fix in block level, will itself fix the violation in full chip level. By this, the ECO can be verified quickly and ultimately the design will be closed quickly achieving the Turn Around

Time (TAT). Creating a session requires certain input. Since it is context-based session, the context information is the first and foremost input required. The HYPERSCALE variable is enabled for this process. In the working area, netlist, spf and other timing information are saved. Session creation command is run finally.

C. Stages in synthesis

- 1 Import Design
- 2 UPF
- 3 Uniquify
- 4 Constraints
- 5 Compile
- 6 Insert DFT
- 7 Ungroup
- 8 Re-timing
- 9 Syn final

V. TIMING CLOSURE METHODOLOGY

Timing closure is done by analyzing the design with respect to the timing aspects of that design and it involves fixing all kind of timing violations. For analyzing the design STA is used.

There are two stages in the design flow in which the methodologies for timing closure are applied. Post-ICC/Signoff stage and Pre-signoff/ICC stage.

A. SETUP HOLD VIOLATION FIX

- Upsizing the standard cells (increase the drive strength) in data path.
- Pull the launch clock.
- Push the capture clock.
- Removing buffers from data path (hold margin should be checked).
- VT swap Replacing high VT cells with low VT cells.
- Replacing buffers with two inverters placing farther apart so that delay can adjust Methods to fix hold violation are given as,
- Downsizing the cells (decrease the drive strength) in data path.
- Pulling the capture clock.
- Pushed the launch clock.
- Inserting buffers/Inverter pairs/delay cells to the data path.
- By increasing the wire load model, we can also fix the hold violation.

VI. UNFIXABLE VIOLATIONS

- A: There are available library cells outside the area limit.
- B: Delay improvement is too small to fix the violation.
- C: The violation is in clock network.
- I: Buffer insertion with the given library cells cannot fix the violation.
- S: Cell sizing with alternative library cells cannot fix the violation.
- T: Timing margin is too tight to fix the violation.
- U: UPF restricts fixing the violation.
- V: Net or Cell is invalid or has do not touch attribute.
- W: Fixing the violation might degrade DRC violation.



A. MACRO PLACEMENT

B. IO PLACEMENT

The IO ports are considered pins for the blocks through which it communicates with the external world or other partitions. All the IO ports are placed on the boundary of the design only. IO port optimization is a technique through which port-pin alignment is achieved. This paves way for better timing. In IO port optimization the ports of the design and the pins inside the design are aligned in such a way that there is only a minimum amount of distance between them. Hence, routing can be made efficiently utilizing less amount of metal. The factor that ports will be talking to the neighboring partitions should also be considered during IO port optimization. Also, if the port size is same as those pin size then chances of getting DRC violations during routing is very less shown in Fig. 3.

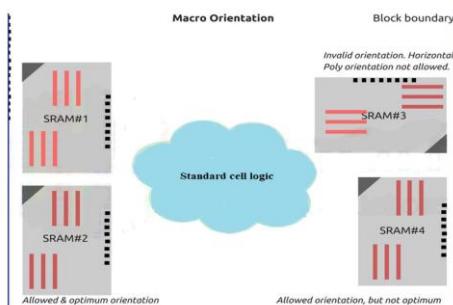


Fig. 3. Efficient macro placement and orientation

C. STANDARD CELL PLACEMENT

In modern design, the number of standard cell count in a design has increased drastically. Placing them one by one is a tedious task. Hence, some constraints are given for placing them. A core area is allocated for standard cells placement and all the cells are confined within that area. Also there is one technique called bound creation. Bound is an imaginary area in the design in which a group of cells are confined within. In this way the distance between two cells will be reduced and routing can be done with minimal metal. This ultimately improves the timing. If the placement of a cell is bad then it can be moved manually to a better place.

D. CTS

There are two important steps in the CTS. They are clock tree building and clock tree balancing. Once the CTS is done, then the skew and latency reports should be analyzed shown in Fig. 4. If they are found to be worst then the clock tree has to be rebuilt to make them better shown in Fig. 5 and 6.

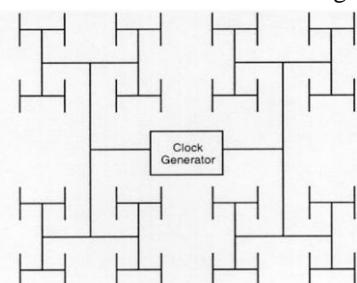


Fig. 4. Typical H-tree

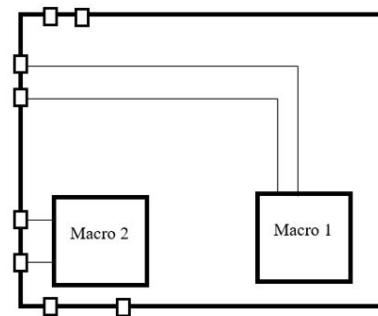


Fig. 5. Macro Placement before Optimization

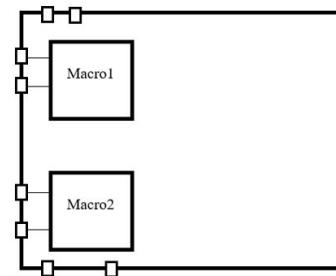


Fig. 6. Macro Placement after Optimization

E. CONTEXT BASED SESSION CREATION FOR A BLOCK

- The path is violating in both the levels with a small difference.
- The path is violating in full chip level but meeting in block level with huge margin.
- The path is violating in full chip level but in block level the same path is unconstrained.

F. IO PORT OPTIMIZATION

As a part of the layout information, in addition to boundary of the design the placement information of the IO ports will also be given. They also should be placed on the boundary of the design. These ports act as the gateway between external world and the components inside the chip. Aligning the ports along with the pins of macros or cells will reduce the routing the distance, improve the timing and helps the neighbouring partitions. In figure 5.4, it can be seen that the IO ports are randomly placed with respect to the pins of macro. This will affect the timing of the design very much. For the optimization, firstly the pins that are connected to the ports must be identified. This is done by using all-connected command. This is done in the floorplan stage so that the pin information is obtained easily. get-ports command is used to get all the ports. Then, the location of the pins is identified by getting the bounding-box information of the pins using bbox attribute of the pins in get-attribute command. After this, the ports are moved to the exact location of the pins. From there the ports are moved to the boundary by giving the specific delta value. The delta value is the difference between designs X or Y and macros X or Y shown in Fig. 7 and 8.

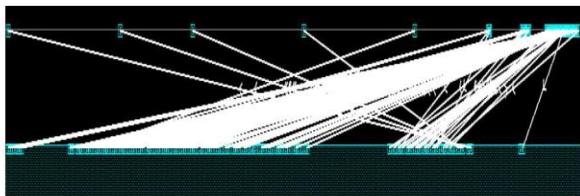


Fig. 7. IO Port Placement before Optimization

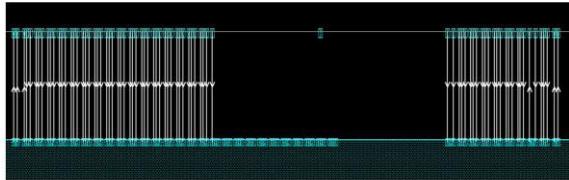


Fig. 8. IO Port Placement after Optimization

VII. RESULTS

- An Engineering Change Order (ECO) to fix the hold violations is the first output of running the script.
- After sourcing the ECO, an updated design with some of the hold violations fixed is obtained shown in Table I.
- Fixing hold violation has a negative effect on setup margin and vice versa.

Table- I: Comparison of ECO blocks before stack and after Stack

Blocks	Minimum Stack before ECO	Minimum Stack after ECO
Block 1	6756	235
Block 2	1786	278
Block 3	725	114

- Only three buffer variants are used for fixing hold violation.
 - Buffer with low drive strength (buf07)
 - Buffer with medium drive strength (buf13)
 - Buffer with high drive strength (buf27)
- The delay induced by these buffers can be found by inserting these buffers in the design using insert buffer command.
- After adding these buffers the updated timing path can be reported and the delay value shown in Table II.

Table- II: The Max and Min corner delay of the buffer

Buffers	Delay Induced (ps)	
	In max corner (Setup)	In min corner (Hold)
buf07	85.63	40.12
buf13	100.12	52.21
buf27	118.23	59.62

VIII. CONCLUSION

Various methods for achieving the timing closure of the design were discussed. Timing optimization in both the pre-signoff and signoff stage is important. Usually the design will be in the post ICC/signoff stage for long time because this stage is responsible for the signoff and closure. Once the signoff team signals, then only the design will be moved in for tape out (fabrication). That is the reason why timing team is more important in the entire design flow. Timing closure involves analyzing loads of data and deriving a solution from that analysis. From the analysis, it can be seen that hold violation is more critical than the setup violation as it does

not depend on the timing period. The hold violation occurs only after the clock tree is built. Before the clock tree is built all the clocks will be ideal, that is the latency and skew will be zero. Hence, before CTS the setup violations are fixed. The clock pulling and pushing techniques were not implemented in this project, which are to be explored in the future. Also various techniques for optimizing the timing in the pre-signoff stage were discussed. The techniques that are used are partition specific. There can be designs where IO port optimization is not possible at all. Only the shielding technique of crosstalk fixing was discussed. There are other techniques for fixing the crosstalk noise, which can be studied and explored in the future. The other design constraints, power and area should be considered and checked throughout the optimization process. Generally, those factors will also be analyzed and optimized in parallel. With increasing technology, more violations will occur in the future designs and complexity to fix the timing issue will increase. However, STA will hold good for analyzing the design and fixing the violations.

REFERENCES

1. N. Azeniard, M. Aline, D. Auvergne, "Delay bound determination for timing closure satisfaction", Proceeding of the IEEE, 5, 375-388, 2001.
2. Rahul Rithe, Sharon Chou, Jie Gu "The Effect of Random Dopant Fluctuations on Logic Timing at Low Voltage", IEEE Transactions on Very Large Scale Integration (VLSI) System, 20, 5, 911-925, 2012
3. Nithiyanantham Janakiraman, Palanisamy Nirmal Kumar "Multi-objective module partitioning design for dynamic and partial reconfigurable system-on-chip using genetic algorithm", Journal of Systems Architecture, 60, 119-139, 2014.
4. Yanqing Zhang and Benton H. Calhoun , "Hold Time Closure for Subthreshold Circuits Using a Two-Phase, Latch Based Timing Method", Proceeding of the IEEE, 2, 13-14, 2013.
5. Branimir Malnar and Goran Zelic, "Timing Closure of Clock Enable Signals on a 32 nm Intel Itanium Processor", Proceeding of the IEEE, MIPRO, Opatija Croatia, 78-83, 2018.
6. Shashank Prasad, Dongzi Liu, Oleg Levitsky, Dave Noice, Shailendra Srivastava, "Post assembly timing closure for multi million gate chips", Proceeding of the IEEE, 23rd International Conference on VLSI Design, 75-80, 2010.

AUTHORS PROFILE



Piyush Bhatasana has received B. E. degree in Electronics and Communication Engineering in 2002, M. Tech. degree in Electrical Engineering (Microelectronics, VLSI and Display Technology) in 2010, Ph. D. in RF MEMS design in 2019. Presently he is working on RF MEMS shunt switch, RF devices, VLSI design, digital convertors, Static Timing Analysis (STA), Noise and crosstalk, Setup and Hold check, Signoff, Engineering Change Order (ECO), DTA of VLSI design.