Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Slew Propagation and Optimization in VLSI Design

Pragati Agarwal[1], Rohit Kumar Gupta[2], Rajat Agarwal[3], Neelu Trivedi[4]
[1]*Research Scholar, Electronics and Communication Engineering, IFTM University, Moradabad*
[2]*Associate Professor, Kalyan Singh Government Medical College, Bulandshahr*
[3]*MBBS, Shri Ram Murti Smarak Institute of Medical Sciences, Bareilly*
[4]*Associate Professor, IFTM University, Moradabad*

ARTICLE INFO

Received: 20 Dec 2024

Revised: 12 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Propagation of slew in design is an important aspect while performing the static timing analysis (STA) of a design. Slew has a direct impact on delay of a timing path and could make the design pass or fail the timing closure. This research paper introduces the concept of slew and how it is propagated in two different ways in the design, and how this is a problem in itself to choose between the incident-slews at a slew merging point. In static timing analysis (STA), the waveforms that change are measured in terms of how fast or slow they change. The slew is measured in terms of the time it takes to transition from one level to the other. The rate of change is defined as the slew. The slew rate has inverse relationship with the transition time. If the slew is slow, then the transition time is more and if the slew is fast, the transition time is less. In a digital circuit design, there are points where timing arcs merge. Such points could be termed as slew merging points. In such conflicts, there are two approaches to move forward. The first approach is graph-based static timing analysis. In a graph- based approach, the worst case delays are taken into consideration by taking into the account the worst case slews (slow slews) along the timing paths, for setup analysis. While in case of hold analysis, the best case delays are taken into consideration by taking into the account the best case slews (fast slews) along the timing paths. The second approach is path-based static timing analysis. In this approach, the actual delays are taken into consideration by taking into account the actual slews along the timing paths, for setup as well as hold analysis. In path-based static timing analysis, delay is computed of the timing path so as to obtain the actual delay values. Such delay calculation takes some extra amount of time. There is a trade-off between this extra calculation time and accurate delay calculation. The synthesized gate-level net-lists are sourced in the tool along with the required files. These files include liberty timing files, which include the timing information for various cells present in the design. Library exchange files are also fed in which includes the information about various metal layers used for routing. The presence of this file makes the post-route timing possible as the values of resistances and capacitances could be extracted from interconnects. Graph-based and path-based approach are followed differently on the design. In the graph-based approach, initial timing reports are generated and then after optimizing the design for slack improvement, post-route timing reports are generated which are analysed.

Keywords: *Congestion, Slew, STA, GBA, PBA, Total Negative Slack, Worst Negative Slack*

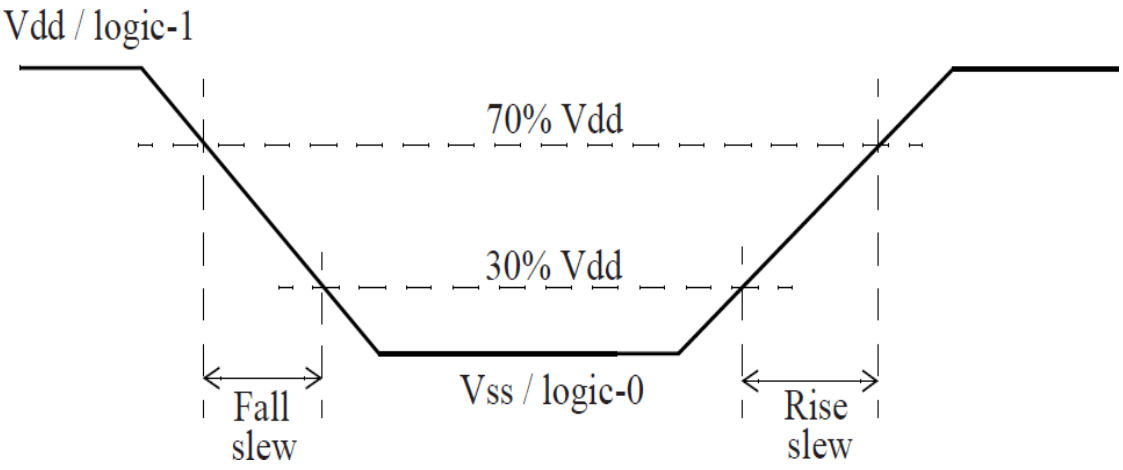
INTRODUCTION

In static timing analysis (STA), the waveforms that change are measured in terms of how fast or slow they change. The slew is measured in terms of the time it takes to transition from one level to the other. The rate of change is defined as the slew. The slew rate has inverse relationship with the transition time. If the slew is slow, then the transition time is more and if the slew is fast, the transition time is less. Figure.1 shows the rise and fall slews.

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Figure .1 Rise and Fall Slews



The values specified for slew measurements are with respect to VDD. In Figure.1, the fall slew is the time taken by the waveform to change from 70% VDD to 30% VDD. Similarly, the rise slew is the time taken by the waveform to change from 30% VDD to 70% VDD. These points which are specified for slew measurement are called the threshold points. These threshold points could be changed as well e.g. 10% VDD to 90% VDD.

In [1], an algorithm has been proposed in which timing optimization is done, which is path based, by inserting the buffers. A timing graph of a logical network is made. For thin film transistor circuits (TFT), a static timing analyzer (STAF) is presented in this paper [2] by Chao-Hsuan Hsu et al. For finding the delay of longest and shortest path, the applied STA algorithm used is block based and it is done for dissimilar regions which are under bending. SPICE simulations are used for finding gate delays. This patent [3] discusses the exhaustive path-based STA algorithm. The focus of [4] is on path based approach which is in depth of AOCV model. This path based approach helps in enhancing the capability of AOCV for the reduction of pessimism. In this paper [5] Timing Arc Based Logic Analysis, which is called as TABLA is being presented for the reduction of false noise by performing temporal logic analysis for the given circuit. The used approach is based on timing arc which helps in finding the glitches. The authors in [6] Jose Luis Guntzel et al. proposed a method without using this macro expansion. As macro expansion no doubt increases the execution time but suffers from accuracy loss. So authors proposed an algorithm for functional timing analysis of the complex circuits which has a large number of gates, which is an extension to Automatic Test Generation (ATG) based algorithm. For identifying the correct worst circuit delay, a new method for signal propagation was proposed by authors David Blaauw et al [7]. For selecting longest path delay in a dicey condition, multiple signal propagation is used in this paper. The proposed algorithm in [8] by David Blaauw et al. focusses on finding the delay and critical path of a timing graph as the traditional or earlier methods underrate the circuit delay which creates difficulties for the tools for further processing.

One of the cause of the silicon failure is the pessimism in crosstalk which if not resolved leads to violations which are difficult to identify as well as to fix. So to reduce the inherent pessimism authors Arvind NV et al [9]. proposed a solution for this. The results show a reduction of 70% violations using path based analysis and also the run time is not very large. In [10] for the purpose of timing validation, a path based technique is presented. The technique has two paths, first is ranking optimization and the other one is path filtering. So for characterizing the interdependent hold and setup times, authors proposed the methodology and algorithm [11]. Multiple pairs in STA can be utilized to decrease the unwanted pessimism and the interdependency reduces the optimism. This paper [12] deals with that moments of input waveform only by mapping the timing waveforms to the characterization waveform. This mapping is based on the moments of waveforms. It is required to verify all the process corners of a system, but due to the increase in PVT variations, the number of process corners have increased. So the designers are trying to decrease the number of process corners verification but this can lead to some unacceptable results for some cases, not all. In case of setup analysis, this could be possible, but not in case of hold. So the authors in their work [13] introduces a method which covers all the process corners in case of hold analysis. The authors Abdoul Rjoub et al. in [14] proposed

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

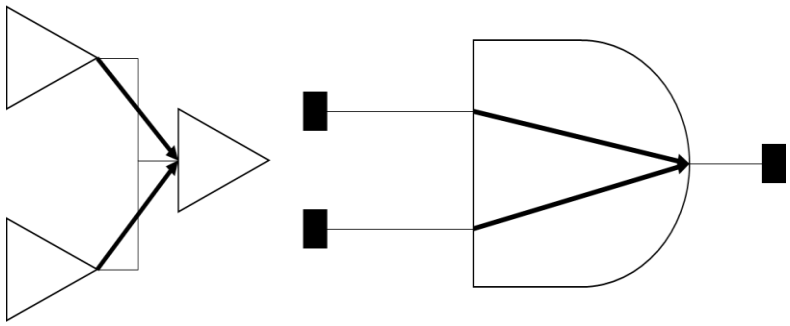
a graph model so as to observe the CMOS circuits at transistor level and to check their behavior. BSIM4 equations is the basic on which estimation of timing analysis is done for the CMOS circuits. In [15], this inaccuracy which is caused by path delay fault is considered and for that authors present a method. Google recently introduced a programming paradigm called as MapReduce for processing the big data. So in [16] authors introduced an algorithm which is path based with MapReduce, called as fast incremental path based algorithm. Using this algorithm incremental PBA problem is transformed into tasks and then they are parallel mapped to reduce and reduce the operation.

Focus of [17] is more on graph based designs and the authors are against path based designs because of the reasons like in modern designs there are large number of paths available, which are non-scalable, so as the path based techniques. The authors proposed a graph based technique, that addresses the limitations of path based techniques. For removal of pessimism, a number of approaches were developed. In this paper another approach for the same is discussed by [18] A.B. Kanng et al. in which timing model of a flexible flip-flop is being exploited for the purpose of reducing its pessimism in setup and hold critical path timing analysis. In [19] a methodology for solving timing problems at transistor level for CMOS circuits is introduced, also methods for building the tool which consists of algorithms and methods for performing the STA. In [20], authors focus on statistical static timing analysis, its aspects and applications in VLSI systems. SSTA comes as a solution to the process variation which cause uncertainties in power and process. So authors present block based SSTA model for both global and path correlations. Common-path-pessimism is one of the factor which cause problems while doing timing analysis. So, its removal is a crucial step, which is commonly called as Common-path-pessimism removal(CPPR). So the concern[21] is CPPR, the solution comes as a fast path based analysis. It is different from the others in the way that instead of searching the path explicitly. An implicit representation of path is performed which results in faster run time with the smaller search space. The timer used in this algorithm is ease of use, less complex, ease of code, simple.

GRAPH-BASED STA AND PATH-BASED STA

In a digital circuit design, there are points where timing arcs merge. Such points could be termed as slew merging points. Such slew merging points are shown in Figure.2. Now at these points, the question arises which slew should propagate forward and be considered in the delay calculation for a particular timing path.

Figure.2 Slew Merging Points



In such conflicts, there are two approaches to move forward. The first approach is graph-based static timing analysis. In a graph-based approach, the worst case delays are taken into consideration by taking into the account the worst case slews (slow slews) along the timing paths, for setup analysis. While in case of hold analysis, the best case delays are taken into consideration by taking into the account the best case slews (fast slews) along the timing paths [22]. The second approach is path-based static timing analysis. In this approach, the actual delays are taken into consideration by taking into account the actual slews along the timing paths, for setup as well as hold analysis. In path-based static timing analysis, delay is computed of the timing path so as to obtain the actual delay values. Such delay calculation takes some extra amount of time. There is a trade-off between this extra calculation time and accurate delay calculation [23].

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

PROBLEM DESCRIPTION

Most of the design engineers use graph-based static timing analysis for delay calculation in a digital design. The paths which are critical, path-based static timing analysis could be used when the tape-out is near. Generally, the runtimes of path-based static timing analysis are more as compared to the graph-based static timing analysis, for the same number of paths. Because of this only, use of path-based static timing analysis is limited. So, the design engineers need to take a call whether to fix the timing with additional run-time required for performing path-based static timing analysis and missing the deadline or to go for graph-based static timing analysis which consumes lesser amount of time [24].

There are three modes in which timing analysis could be done, which are:

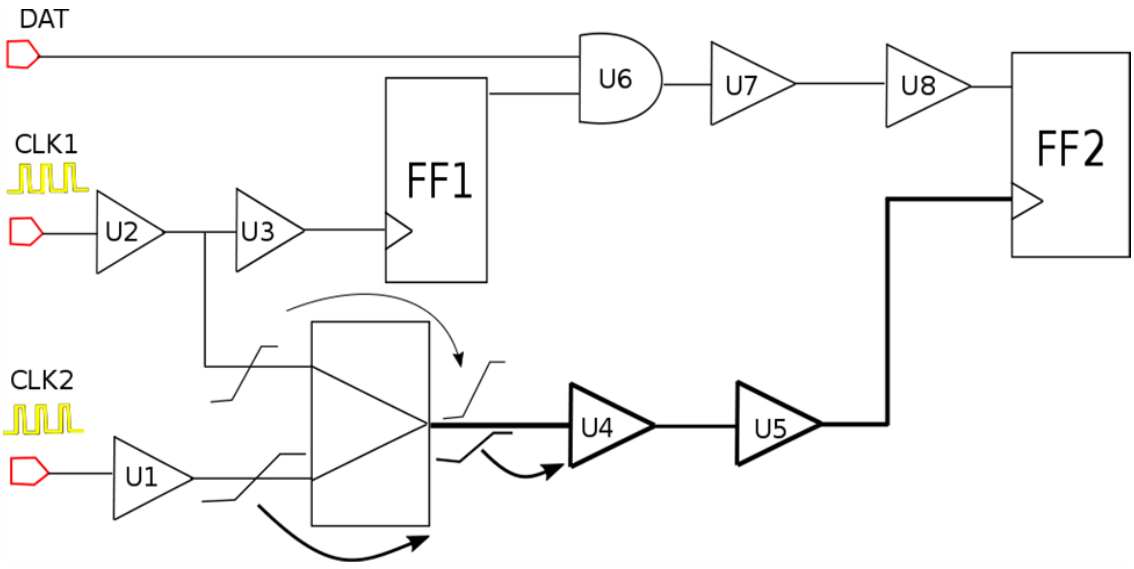
a) single
b) bc_wc
c) on_chip_variation

These analysis determines which slew (min, max or both) would propagate at various slew merging points and how setup timing paths and hold timing paths are formed by combination of max and min delay arcs.

In *single* analysis mode, a single operating corner is used for analysis. The delay values and the transition values in this corner are worst-case (or slowest) timing values. Both setup and hold are calculated for max-delay timing arcs. In *bc_wc* analysis mode, there are two PVT (Process-Voltage-Temperature) operating corners. One is the best-case (fast) corner and the other is the worst-case (slow) corner. These corner do not exist at the same time on the die. For setup paths, the longest path with max delays timing arcs is chosen for launch and the shortest path with max delay timing arcs is chosen for capture. For hold paths, the shortest path with min delay timing arcs is chosen for launch and the longest path with min delay timing arcs is chosen for capture. In *on_chip_variation* analysis mode, there are two operating corners, which could exist at the same time on a die. For setup paths, the longest path with max delays timing arcs is chosen for launch and the shortest path with min delay timing arcs is chosen for capture. For hold paths, the shortest path with min delay timing arcs is chosen for launch and the longest path with max delay timing arcs is chosen for capture.

There is an optimism in *single* and *bc_wc* analysis mode. This optimism could lead to setup violation. This could be explained with the help of an example given in Figure.3

Figure.3 Analysis of setup timing path in *single* and *bc_wc* timing analysis modes



Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In *single* or *bc_wc* analysis modes, for setup paths, max delay values are chosen for all timing information. For setup paths, the launch path should be slow whereas the capture path should be fast. In Figure.3, we have two slews at the inputs of the mux, one is fast and the other is slow. Both of these are available at the output of the mux. In these analysis modes, the one which is slower would propagate and because of which, the fastest possible timing is not achieved as compared to the actual operation on the chip.

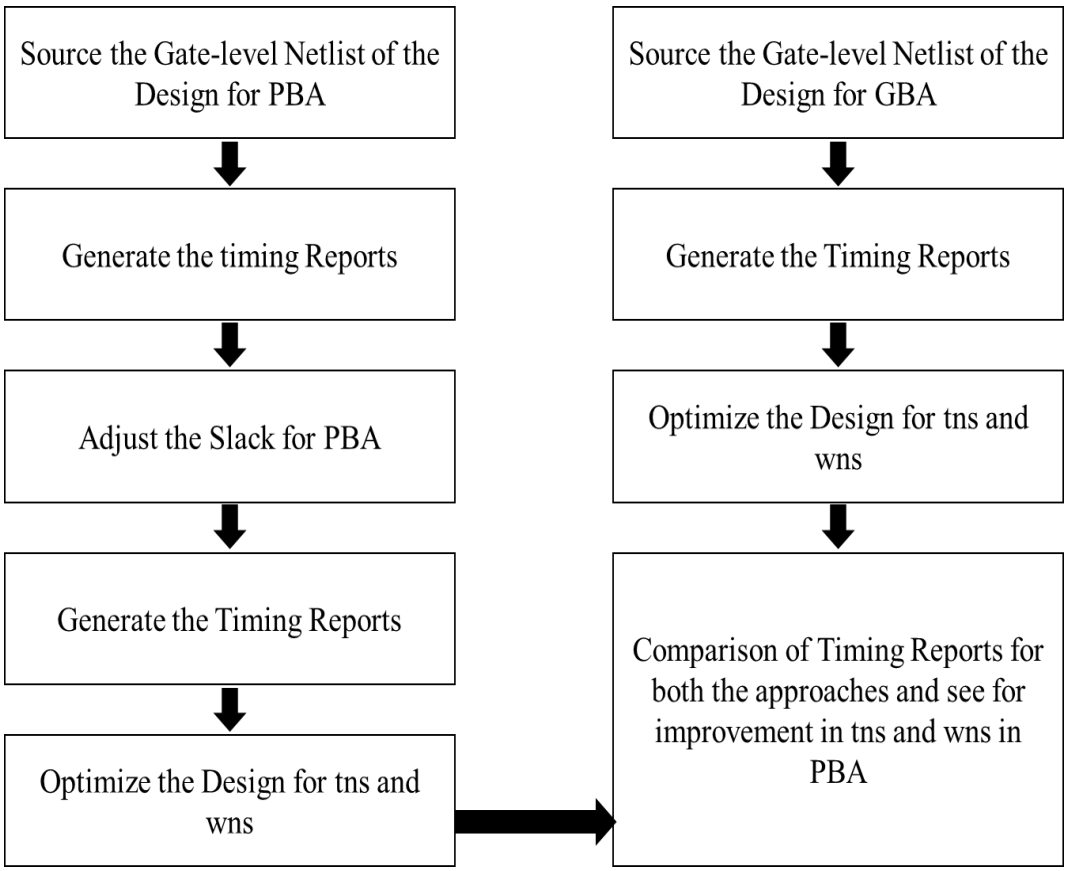
METHODOLOGY USED

The synthesized gate-level net-lists are sourced in the tool along with the required files. These files include liberty timing files, which include the timing information for various cells present in the design. Library exchange files are also fed in which includes the information about various metal layers used for routing. The presence of this file makes the post-route timing possible as the values of resistances and capacitances could be extracted from interconnects. Figure.4 shows the work flow.

Graph-based and path-based approach are followed differently on the design. In the graph-based approach, initial timing reports are generated and then after optimizing the design for slack improvement, post-route timing reports are generated which are analyzed.

In the path-based approach, after generating the initial timing reports, the PBA slack is adjusted and again timing reports are generated to see whether there is any improvement in the total negative and worst negative slack (TNS and WNS). If the improvement is seen, then the design is optimized and post-route timing reports are generated, which are analyzed and compared with the post-route timing reports obtained in the graph-based approach for improvement in TNS and WNS.

Figure.4 Work Flow



Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

ANALYSIS AND RESULTS

In Figure.5, there are various slew merging points, between input and output pins of gates g3, g4 and g5. When different slews arrive at these slew merging points, the decision has to be made which slew should propagate forward for delay calculation of the further logic gates. In case of graph-based static timing analysis, the run-times are less but the delays which are obtained are less accurate. Graph-based static timing analysis is a pessimistic approach where worst case slack is obtained for the setup and hold analysis. Moreover, in graph-based static timing analysis, a single value of delay is considered for a timing arc, which is part of a multiple timing paths.

This is not the case in path-based static timing analysis. The paths are recalculated to obtain the actual delays. This is an optimistic approach. In path-based static timing analysis, an arc could have multiple delay values for different timing paths which makes the timing analysis more realistic.

Figure.5 Design with various slew merging points

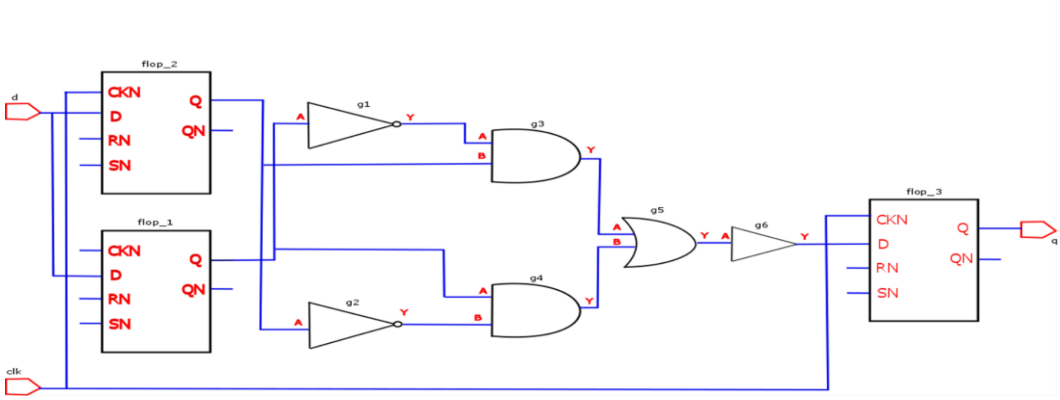
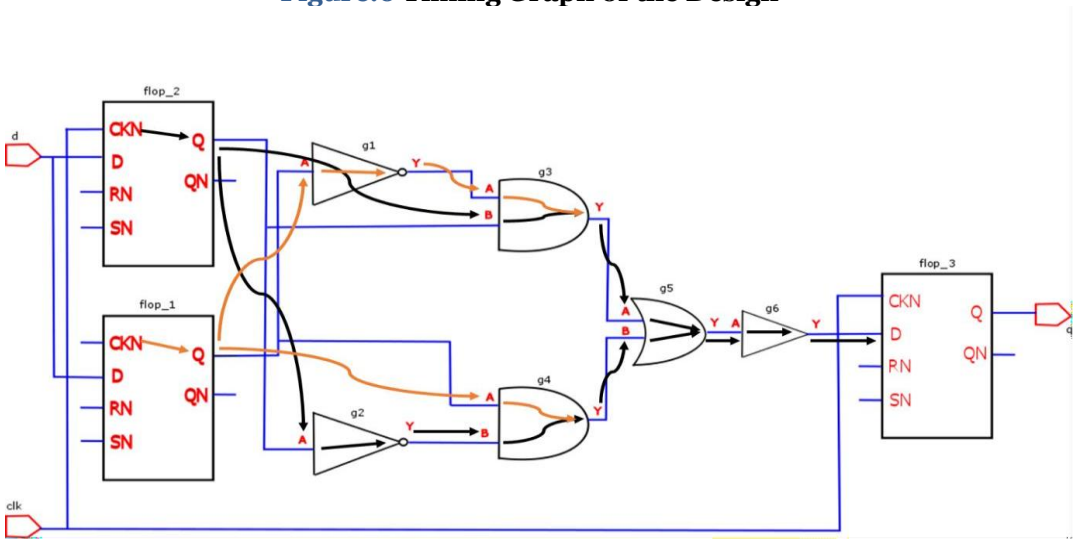


Figure.6 shows its timing graph. This figure shows all the possible timing paths in the design by showcasing all the timing arcs. The timing graph shows the timing data-base. All the input and output pins are expressed as nodes in a timing graph while the arrows which we refer to as timing arcs are used to connect these nodes.

Figure.6 Timing Graph of the Design



In path-based timing analysis, a path which is recalculated, is separated out from the graph and a more detailed timing analysis is carried on it. This process is called path recalculation because a graph path has been taken out and recalculated for a totally new timing behavior.

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

In the case of path-based static timing analysis, the slack improves a lot. This happens because the worst-case timing has already been considered in case of graph-based timing analysis. So, there is no chance of slack degradation. There are various improvements which occur in the timing of a path when a particular path is recalculated. Some the improvements are mentioned below:

a) When a path is recalculated for delays along the nets and cells, accurate timing results of slew degradation are obtained.

b) As the focus is single path, so in order to study effects like signal integrity a single edge could be propagated along the path, there is no need of any arrival window.

c) In graph-based static timing analysis, the slowest slew is propagated which is more prone to crosstalk effects. This is not the case in path-based static timing analysis where faster slews are propagated, which are less prone to crosstalk effects.

d) In graph-based static timing analysis, the values used for determination of CRPR are very pessimistic but in case of path-based static timing analysis, accurate CRPR values are obtained for a timing path.

The recalculation of the timing paths is done exhaustively. There are several factors which contribute to the total number of timing paths, which are described below:

a) Total number of start-points.
b) Rising and falling edges are considered separate.
c) Existence of paths which branch apart but later converges back together.
d) Existence of gates which have both non-inverting and inverting timing arcs like XOR gates.

For a certain endpoint, multiple timing paths could exist. So all these paths to a certain endpoint are recalculated in order to obtain the timing path with the most negative slack.
In graph-search path analysis, for a certain number of endpoints, all the timing paths are analyzed. The paths with the maximum slack checked among all the checkpoints are reported. The exhaustive path-search algorithm is a much more complex process. In exhaustive path-search algorithm, all the paths leading to a certain endpoint are recalculated with their actual delay values. The path which was earlier the worst case path could no longer be the worst one, after recalculation. If a certain path is again encountered while moving from worst case slack to best case slack, those having the worst slack are reported. Approximately twenty-five thousand such iterations are performed in an exhaustive path-search process.

Optimization on Various Designs

Various designs were taken. On them graph-based and path-based optimization were carried out, and the timing reports were obtained for improvement in the total negative slack (TNS) and the worst negative slack (WNS). The results obtained are shown in the Table 1.

TABLE 1 OPTIMIZATION ON VARIOUS DESIGNS

Design	Operating Frequency (MHz)	Number of Standard Cells	TNS After GBA Optimization (ns)	TNS After PBA Optimization (ns)	WNS After GBA Optimization (ns)	WNS After PBA Optimization (ns)
1.	333.33	2044177	-457.164	-421.998	-12.576	-12.515
2.	1000	208877	-776.712	-755.417	-0.388	-0.388
3.	1754.4	633888	-7230.315	-5895.773	-2.51	-2.509
4.	48.01	70452	-1493.779	-1451.705	-50.894	-50.894

In the Table 1, it could be seen that there is significant improvement in the TNS values when PBA based

Journal of Information Systems Engineering and Management
2025, 10(47s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

optimization is done in comparison to GBA based optimization. The improvement is there because in path based approach, the timing path is recalculated to obtain accurate slews of the arcs on that path, instead of the worst case slew which is selected in case of graph based approach.

Figure.7 Congestion Analysis Graph

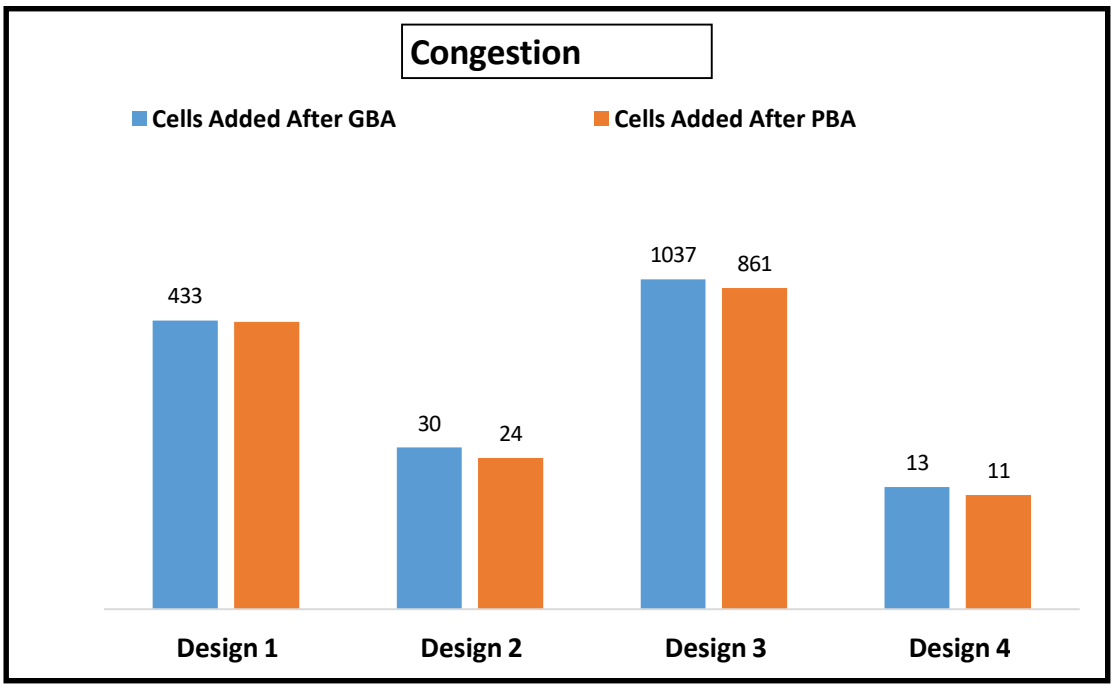


Figure.7 shows the congestion analysis graphs. The number of cells added (buffers/inverters) after optimization were obtained. As observed in the case of every design, the number of cells added is less in case of path based optimization as compared to the graph based optimization. The reason behind this is that in case of path based optimization the total negative slack is lesser as compared to that in graph based optimization, so less number of cells (buffers/inverters) are needed to remove it. Thus, lesser area is needed if path based optimization is followed in a design.

CONCLUSION

This work focusses on the slew propagation at a slew merging point. There are two aspects of looking at it. First one is the graph based approach, which is a pessimistic approach and in it the slower or the worst case slew would propagate at a slew merging point, as could be seen in the previous section. Second one is the path based approach, which is a less pessimistic approach and in it the actual slew of the timing arc for that particular path propagates forward at a slew merging point, as observed in the previous section.

When a path based approach is applied on a design, a significant improvement in the TNS (Total Negative Slack) and WNS (Worst Negative Slack) could be seen in comparison to the graph based approach. When such analysis is done on various design, a significant improvement was observed. Area needed to implement the design also gets reduced as less number of cells are added in case of path based approach as compared to graph based approach.

Currently, the semiconductor industry has become so competitive that the tape-out time has significantly decreased. With this timing closure have become impossible to achieve if a totally pessimistic approach is considered considering the worst cases, like in graph based approach. So, designer use a more

optimistic approach like the path based approach in which the actual slews for a particular timing path are recalculated in order to obtain the accurate delays for the cells.

In such an approach, there is a trade-off with time. As recalculating a path could take a significant amount of time in a design which has billions of cells and one could miss the targeted time for timing closure, there is a risk involved. So, one has to choose among them. Generally, only for critical paths path based approach is followed and for the rest of the design, graph based approach is followed.

REFRENCES

[1]. Yiqian Zhang, Qiang Zhou, Xian long Hong and Yici Cai, "Path-based Timing Optimization by Buffer Insertion With Accurate Delay Model," Proc. ASIC, 21 – 24 October, 2003.
[2]. Chao-Hsuan Hsu, Chester Liu, En-Hua Ma and James Chien-Mo Li, "Static Timing Analysis for Flexible TFT Circuits", in proc. of 47th ACM/IEEE Design Automation Conference (DAC), pp. 799-802, June 2010.
[3]. Cristian Soviani, Rachid N. Helaihel, Khalid Rahmat, Fremont, "Efficient Exhaustive Path-Based Static Timing Analysis Using a Fast Estimation Technique" United States Patent Soviani et al., Patent No. US 8,079,004 B2, December 2011.
[4]. Pua Siaw Fuang and Nor Muzlifah Mahyuddin "Implementation Study of Path-Based AOCV Model on Pessimism Reduction for 20nm Technology" IEEE International Conference on Control System, Computing and Engineering, pp. 80-83, November 2014.
[5]. Murthy Palla1, Jens Bargfrede, Stephan Eggersglüß, Walter Anheier, Rolf Drechsler, "Timing Arc Based Logic Analysis for False Noise Reduction", IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, pp. 225-230, 2009.
[6]. Jose Luis Guntzel, Ana Cristina Medina Pinto, Eduardo d'Avila and Ricardo Reis, "ATG- Based Timing Analysis of Circuits Containing Complex Gates" in proc. of 13th Symposium on Integrated Circuits and Systems Design, pp. 21-26, Sept 2000.
[7]. David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Chanhee Oh and Rajendran Panda "Slope Propagation in Static Timing Analysis" in proc. of IEEE/ACM International Conference on Computer Aided Design, ICCAD, 2000, pp. 338-343, Nov 2000.
[8]. David Blaauw, Vladimir Zolotov, and Savithri Sundareswaran, "Slope Propagation in Static Timing Analysis" IEEE transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 21, no. 10, October 2002.
[9]. Arvind NV, Rajagopal KA, Ajith HS and Suparna Das, "Path Based Approach for Crosstalk Delay Analysis", in proc. of 17th International Conference on VLSI Design, pp. 727-730, 2004.
[10]. Leonard Lee, Li-C. Wangl, T. M. Mak, and Kwang-Ting Cheng, "A Path-Based Methodology for Post-Silicon Timing Validation", in proc. of International Conference on Computer Aided Design, 2004. ICCAD-2004. IEEE/ACM, pp. 713-720, Nov 2004.
[11]. Emre Salman, Ali Dasdan, Feroze Taraporevala, Kayhan K¨uc¸ ¨ukc¸akar, and Eby G. Friedman, "Pessimism Reduction in Static Timing Analysis Using Interdependent Setup and Hold Times", Proceedings of the 7th International Symposium on Quality Electronic Design (ISQED'06) 6 pp. – 164, March 2006.
[12]. David D. Ling, Chandu Visweswariah, Peter Feldmann and Soroush Abbaspour, "A Moment-Based Effective Characterization Waveform For Static Timing Analysis", 46th ACM/IEEE Design Automation Conference, DAC '09, pp. 19-24, July 2009.
[13]. S. Onaissi, F. Taraporevala, J. Liu and F. Najm "A Fast Approach for Static Timing Analysis Covering All PVT Corners", 48th ACM/EDAC/IEEE Design Automation Conference (DAC), 2011, pp. 777-782, June 2011.
[14]. Abdoul Rjoub and Almotasem Bellah Alajlouni, "Graph Modeling for Static Timing Analysis at Transistor level in Nano-Scale CMOS Circuits", 16th IEEE Mediterranean Electrotechnical Conference, pp. 80-83, March 2012.
[15]. Bo Yao, Arani Sinha and Irith Pomeranz, "Path Selection Based On Static Timing Analysis Considering Input Necessary Assignments", 2013 IEEE 31st VLSI Test Symposium (VTS), pp.1-6, April-May 2013.
[16]. Tsung-Wei Huang and Martin D. F. Wong, "On Fast Timing Closure: Speeding Up Incremental Path-Based Timing Analysis with MapReduce", 2015 ACM/IEEE International Workshop on System Level Interconnect Prediction (SLIP), pp. 1-6, June 2015.
[17]. Hari Cherupalli and John Sartori, "Graph-based Dynamic Analysis: Efficient Characterization of

Research Article

Dynamic Timing and Activity Distributions", IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 729 – 735, Nov 2015.
[18]. A.B. Kanng, H. Lee, "Timing Margin Recovery With Flexible Flip-Flop Timing Model", Fifteenth International Symposium on Quality Electronic Design, pp. 496 – 503, March 2014.
[19]. Abdoul Rjoub, Almotasem Bellah Alajlouni, Hassan Almanasrah, "Graph Modeling for Static Timing Analysis at Transistor Level in Nano-Scale CMOS Circuits", 16th IEEE Mediterranean Electrotechnical Conference, pp. 80-83, March 2012.
[20]. Abu M. Baker and Yingtao Jiang, "Modeling and Architectural Simulations of the Statistical Static Timing Analysis of the Non-Gaussian Variation Sources for VLSI Circuits", International Journal of Scientific and Research Publications, Volume 3, Issue 1, January 2013.
[21]. Tsung-Wei Huang, Pei-Ci Wu, and Martin D. F. Wong, "Fast Path-Based Timing Analysis for CPPR", 2014 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 596 – 599, Nov 2014.
[22]. Izumi Nitta, Toshiyuki Shibuya and Katsumi Homma, "Statistical Static Timing Analysis Technology", FUJITSU Sci. Tech. J., 43, 4, October 2007.
[23]. Hamed Abrishami, Jinan Lou, Jeff Qin, Juergen Froessl, Massoud Pedram, "Post Sign-off Leakage Power Optimization", in proc. of Design Automation Conference (DAC), 2011 48th ACM/EDAC/IEEE, pp. 453 – 458, June 2011.
[24]. Aseem Agarwal, David Blaauw, Vladimir Zolotov, Savithri Sundareswaran, Min Zhao, Kaushik Gala, Rajendran Panda, "Path-Based Statistical Timing Analysis Considering Inter and Intra-Die Correlations", 48th ACM/EDAC/IEEE 48th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 453 – 458, June 2011.
[25]. http://community.cadence.com/cadence_blogs_8/b/ii/archive/2013/12/02/signoff-summit-tempus-path-based-analysis-and-finfet-timing-closure
[26]. http://electronicdesign.com/eda/eda-vendors-should-improve-runtime-performance-path- based-analysis
[27]. http://vlsi-soc.blogspot.in/2014/07/timing-analysis-graph-based-vs-path.html
[28]. J. Bhasker & Rakesh Chadha, "Static Timing Analysis for Nanometer Designs – A Practical Approach," Springer, 2009.
[29]. Himanshu Bhatnagar, "Advanced ASIC Chip Synthesis," Second Edition, Kluwer Academic Publishers, 2002.