International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7 Issue-4, September 2018

On Timing Closure: Hold-Violation Removal using Insertion of Buffers, Inverters and Delay Cells

S. Aruna Mastani, Patra Suresh Reddy

Abstract*: Timing closure plays a major role in the Physical design synthesis. In the process of timing optimization, buffers, inverters, delay cells are used as delay elements inorder to speedup the circuit. Given with the violated path proper selection of combination of delay elements is to be made to meet the hold requirement of the path. Here, in this work a standard industrial design is taken having 5241 violated paths. For these paths, hold time is computed as per the linear programming optimization. Timing closure is done by various combinations of buffers, inverters and delay cells. Discrete buffers, Complex timing constraints and accurate timing models/analysis make time consuming and problem difficult to solve. The linear programming-based methodology is presented to model the setup and hold-time constraints. Then based on the solution to the linear programming optimization, buffers, inverters and delay cells are inserted as delay elements to solve hold violations. The implemented approach where delay cells, buffers, inverters are used as delay elements for optimization and using only buffers as delay elements in optimization process are tested on industrial design together with the industrial hold optimization flow, and better results achieved in terms of minimum hold slack, hold violations and utilization are reported.¬¬ Compared to the delay insertion using buffers only, the implemented approach can obtain 31% worst hold slack reductions and better utilization for the industrial circuit level design. Analysis of timing paths and removal of hold violation problem in physical design flow is implemented using TCL and PERL scripts in cadence encounter tool.

Index Terms: Delay Insertion, Hold Violation, Physical Design, Utilization.

I. INTRODUCTION

Timing closure plays a major role in physical design synthesis. Timing constraints are mainly classified into setup and hold-time constraints. Signal transitions need to arrive too early for setup constraints leads to long path while for hold time constraints signal transitions need to arrive too late leas to short path. Techniques that are available for timing optimization are buffer insertion, cell sizing and logical restructuring. Buffers, Inverters or delay cells are used as delay elements in-order to speed up the circuit. In this paper, Timing closure is done by various combinations of buffers, inverters and delay cells.

Generally, hold violations are considered after performing setup optimization. Hold violations are solved by inserting delay elements. Existing approaches for removing hold violations are the delay insertion technique used in clock period minimization [2,3,5,7] which specifies the insertion delay as unit delay elements [3], fixing hold violations for ultra-low voltage designs [8],and recently, fixing hold violations by reducing the number of buffer insertions [1]. However, in modern industrial designs, other than unit delay elements discrete type of buffers or inverters are considered in delay insertion.

In industrial designs, cell libraries are different for setup and hold time constraints, means delay of the delay elements (buffers, inverters, delay cells)caused by setup time constraints is different from hold time constraints. So, hold optimization need to consider these challenges.

In this paper, an optimization flow is presented to fix hold violations by placing buffers, inverters and delay cells as delay elements. The optimization flow includes first linear programming based optimization that specifies different delays which are occurred between setup time and hold time constraints. Second, bottom up delay insertion algorithm which is a reliable solution to the linear programming optimization by inserting delay elements. In the experimental results, comparison of delay insertion using only buffers with buffers, inverters and delay cells as delay elements on modern industrial design that runs together with a Industrial hold optimization flow, results shows that the implemented algorithm achieves better results in terms of minimum hold slack, hold violations and utilization.

II. PRELIMINARIES

Let D be the given design that consists of combinational circuits set as C, combinational circuits pins as P, and nets N which defines the pins connectivity. Let PI be the primary inputs and the sequential cells outputs (for example, latches and flip-flops), and PO be the primary outputs and the sequential cells inputs. Furthermore, B is the set of buffers, inverters and delay cells that are defined in the standard cell library, which have different technology and area parameters for each delay element. Furthermore, the setup constraints from the standard cell library are different to the hold time constraints.

Revised Manuscript Received on 30 September 2018.
***** Correspondence Author
Dr. S.Aruna Mastani, Assistant Professor, Department of ECE, JNTUA College Of Engineering, Ananthapuramu (A.P), India.
Patra Suresh Reddy, Student, M.tech in Digital Electronics and Communication Engineering, JNTUA College of Engineering, Ananthapuramu (A.P), India.

Thus, the delay information is considered individually for the setup and hold-time constraints.

For the setup constraints, the setup slack with respect to the setup constraints at pin p is given by

Setup_slack = setup_required time – setup_actual arrival time

Similarly, for the hold time constraints, The hold slack with respect to the hold-time constraints at pin p is given by

Hold_slack = hold_actual arrival time – hold_required time

Timer engine gives the required time and arrival time.Negative values of setup slacks means setup violations and negative values of hold slacks means hold violations. For on-timing closure, the design must require no timing violations. For all the pins in PO, Let TNS be the total negative setup slack and THS be the total negative hold slack. Hold violations are fixed after optimization of setup violations. While optimizing the hold violations total negative setup slack need to be taken care. While buffers, inverters, delay cells are inserted as delay elements for fixing hold violations, the inserted delay elements can increase the power consumption and area of the design. Therefore, the delay insertion problem for removal of hold violation is given by: Given a circuit level design and a standard cell library, finding a delay solution such that total negative hold slack and the cost of insertion of delay elements which means area and power consumption both are reduced. Furthermore, total negative setup slack is to be taken care.

Here in this paper, the implementing approach uses buffers, inverters and delay cells as delay elements. Over buffering results increase in power consumption and area, so better to avoid over-buffering. Some other different techniques for fixing hold violations are cell sizing and logical restructuring. The implementing approach shows better results using buffers, inverters and delay cells as delay elements by running together with industrial hold optimization technique.

An industrial timing engine is required for providing timing related information such as delay of the cells, the required arrival times, and the actual arrival times of the pins with respect to the corresponding constraints (i.e.,setup constraints for the setup slack and the hold time constraints for the hold slack). The cell delay model being used is taken from the lookup table in which two inputs are given by slew and load capacitance, respectively, whereas the slew is also taken from the slew lookup table.

III. LINEAR PROGRAMMING OPTIMIZATION

Given with the violated path proper selection of combination of delay elements is to be made to meet the hold requirement of the path. Here, in this work a standard circuit level design is taken as the reference. For these paths, hold time is computed as per the linear programming optimization. Timing closure is done by various combinations of buffers, inverters and delay cells. Discrete buffers, Complex timing constraints and accurate timing models/analysis make time consuming and problem difficult to solve. The linear programming-based methodology is presented to model the setup and hold-time constraints. Then based on the solution to the linear programming optimization, buffers, inverters and delay cells are inserted as delay elements to solve hold violations.

The input to the linear programming model is a combinational circuit $C*$ such that for any pin p of $C*$, hold slack is less than zero and setup slack should be greater than zero. Generally, for fixing the hold violation removal problem, the pins that are having negative hold slacks which represents hold violated paths are to be considered and also the total negative setup slack has to be maintained, it is natural that insertion of delay is allowed to the pins that are having setup slack greater than zero. Therefore, the extraction of the pins with hold slacks less than zero and setup slacks greater than zero is required to get $C*$ from a combinational circuit.

The time taken by the flop to stable while taking data at active edge can be said as setup time. setup analysis done based on time period with which they are operating, combinational delay between them, transition time of launch flop, launch and capture time,when to flops are communicating each other. This can be well explained using mathematical equation shown below.

$$T_{cq} + T_{comb} + T_{launch} \leq T_{clockperiod} + T_{setup} + T_{capture} + 2T_{\Delta j}$$

$$Skew = T_{capture} - T_{launch}$$

Where, $2T_{\Delta j}$ is jitter, T_{comb} is the combinational delay and T_{cq} is the delay of the flop from clock pin to output pin.

The right side of the equation is called as required time which is the time taken by the clock pathand left side is the arrival time that is taken by the data to travel through data path.

The difference between the required arrival time and the actual arrival time of the data called as setup slack. This should be maintained positive as a indication that the data can travel without any loss. Skew is nothing but the difference between capture and launch path.

The time taken to hold the input data should not change at active edge can be hold time. For this analysis is made according the timing path and data path present between the flops. Except the time period remaining all parameters are considered for hold analysis.

$$T_{cq} + T_{comb} + T_{launch} \geq T_{hold} + T_{capture} + 2T_{\Delta j}$$

The difference between arrival time and required time is called as hold slack. The hold slack should maintain positive

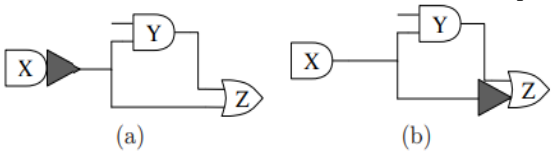

Figure 1: Delay insertion: a) place delay cell at the cell X output, b) place delay cell at one of the cell Z inputs.

Let $d_{i,b}$ be the delay of the delay element b when inserted delay element is either buffer, inverter or delay cell b across pin i. For any source/sink cell C of pin i, Δd^b_C be the change in the delays of the cell C before and after inserting delay element b at pin i. Figure 1a) shows that the delay cell b is placed at the cell X output pin i. Therefore, delays introduced by placing this delay element is $d_{i,b} + d^b_X + d^b_Y + d^b_Z$.

In Figure 1b), a delay cell b is placed at the cell Z inputs i, so the overall delay that is specified by this delay element is $d_{i,b} + d^b_X + d^b_Z$. Here, the overall delay calculation has to be computed for hold time constraints and setup constraints, respectively. Thus, for the hold time constraints hd is used and for setup constraints sd is used in the following equation. $HID_{i,b}$ represents the overall hold delay of placing delay cell b across pin i, and it is given by:

$$HID_{i.b} = hd_{i,b} + \Delta hd^b_{driver\_cell(i)} + \sum_{c:sink\_cells(i)} \Delta hd^b_c$$

Similarly for setup,

$$SID_{i.b} = sd_{i,b} + \Delta sd^b_{driver\_cell(i)} + \sum_{c:sink\_cells(i)} \Delta sd^b_c$$

Here the delay is computed from the lookup table and the output slew is computed from the slew table accordingly.

IV. BOTTOM-UP DELAY INSERTION

Here, in this paper after computing the hold delay that is required for removing the hold violations is based on the linear programming optimization. The amount of delay that is obtained from the linear programming optimization is equal to the delays introduced by the delay insertion of combination of buffers, inverters and delay cells. In bottom up delay insertion, first, a dynamic programming algorithm is implemented in order to insert combination of delay elements such as buffers, inverters and delay cells to satisfy the required hold delay that is calculated from the linear programming optimization. Second, a bottom up delay methodology is implemented to realize all the pins in a circuit level design.

For a given pin p, let DH be the hold delay and DS be the setup delay. Finding a delay solution from a standard cell library B is as follows: The hold delays that are obtained from the linear programming optimization have to be considered here

The hold delays that are introduced by the selective delay elements should be close to the pin p hold delay DH as possible.

The setup delays that are introduced by the selective delay elements should not be more than the setup delay DS.

Area of the selected delay elements is to be minimum.

To realize the above problem, an algorithm is implemented which is said to be a dynamic programming (DP) based algorithm. In this algorithm, delay candidates C (L, d_h, d_s, A) that are taken from the standard cell library. Where the list of the selective delay elements is given by L, the hold delay that is produced by placing L is given by d_h, the setup delay that is produced by placing L is given by d_s, and the area is given by A. Here comparison of two candidates is done for better realization of hold delay, for each delay element that is taken from the standard cell library B, another existing delay candidate is replaced to find out the new setup and hold delays that are introduced by the new candidate. Let the new candidate be C′ (L', d'_h, d'_s, A'), now the realization of a delay solution is done by comparing those two candidates C and C`.

First, if the setup delay of the new candidate is greater than the setup delay D_S of pin p, then the new candidate C` is removed immediately, next, if the hold delay $d_h$` of the new candidate C` is less than are equal to the hold delay $d_h$ of the original selected candidate C, then also C` is removed. The reason for removing the candidate is that the buffers, inverters and delay cells are considered as delay elements. While placing delay elements from sink to source the main criteria is to maximize the hold delay. It is better to increase the hold delay by placing more number of delay elements between sink and source; however, one must be keep in a mind that the increase in delay elements leads to increase in area and power consumption of the design which effects the utilization of the design. Over buffering should be avoided. So, if $d_h` > D_H$ + margin (20 ps), then C` is removed. Next, C` is dominated by any another existing candidate $C^*$ ($L^*$, $d_h^*$, $d_s^*$, $A^*$), if $d_h` < d_h^*$ and $A` > A^*$, then the candidate C` is removed. Decreasing d_h is not allowed. In this experiment, the process stops quickly. The delay solution for the selected candidate has to be the largest ratio of d_h/A.

At last a bottom up delay methodology is implemented to realize all the pins which specifies the chosen delay elements are placed in a bottom up topological order (i.e., from sink to source).

The linear optimization flow for an industrial level design is as shown in Figure 2. Flow starts from launch timer which is launched by the timer engine, next the pins are processed for getting a combinational circuit which contains the hold violated paths and setup met paths. Then, the hold violated paths are processed through a linear programming based optimization to fix the delays that are to be inserted for removing the hold violation problem. Here, the inserted delay makes hold slack value positive by taking care of setup slack value.

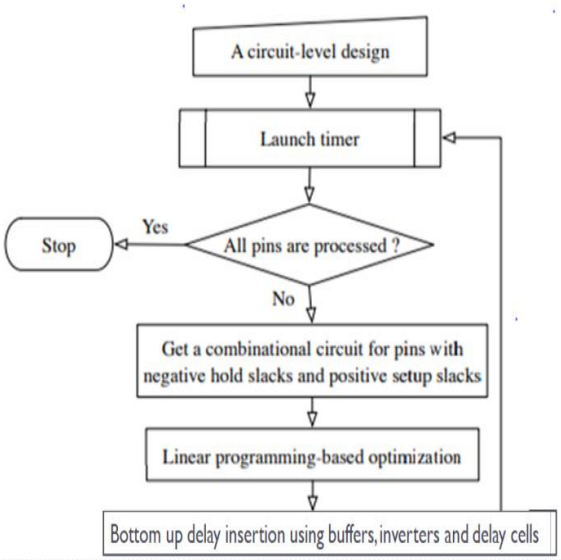

Figure 2: Implemented optimization flow

The bottom-up delay (buffers, inverters or delay cells) insertion is then applied to solve the hold violation problem. Here, the chosen delay element is placed in between sink and source of a violated hold path. The flow stops after computing the possible hold violated paths which results in reducing the minimum worst hold slack value.

V.EXPERIMENTAL RESULTS

The physical design flow is implemented using cadence encounter tool. Analysis of timing paths and removal of hold violation problem in physical design flow is implemented using TCL and PERL scripts. The results are compared based on utilization, violated paths; hold slack values i.e., WHS (Worst Hold Slack) and THS (Total Hold Slack). Utilization:

sudden jump of standard cell utilization is bad (i.e., from one stage to another stage increment in utilization is to be less as ~5%) and utilization should be less than 60% after route stage for addition of circuitry in future use. Slack: WHS should be minimum for the violating paths in-order to meet the timing closure at ECO (Engineering Change Order) stage. Our main aim is to achieve minimum utilization with minimum hold slack value.

Table 1: Design specifications:

Design type	Block (tile) Rectangular in shape
Tile dimensions	Width = 297.92 µm, Height = 1948.52µm
Standard cells count	14541
Total standard cell area	124789.52 μm^2
Number of nets	15879
Number of pins	59043
Number of primary input ports	798
Number of primary output ports	682
Clock period	4 ns
Initial utilization	21.496 %

The experiment is done on industrial design having design specifications as shown in table1. Optimization techniques are used in each step to meet the desired timing constraints in the design flow.

Table 2: Timing summary and initial statistics of the industrial design

Hold mode	All	Reg2reg	Reg2cgate	Default
WHS (ns)	-0.25	-0.14	-0.05	-0.25
THS (ns)	-374.33	-369.04	-1.69	-3.61
Violating paths	5241	5155	70	16
All paths	12763	7680	200	8143

Timing summary and initial timing statistics of the design is shown in table2. Initial standard cell utilization is 21.496%.Our main focus is done on reg2reg paths which represents register to register path. Reg2cgate path specifies the path from clock pin to the register in which additional logic circuitry is added to decrease the power consumption of the design.

Table 3: Results of the industrial hold optimization using buffers

Utilization of the design is 70.057%

Hold mode	All	Reg2reg	Reg2cgate	Default
WHS (ns)	-0.41	-0.13	0.26	-0.41
THS (ns)	-6.24	-3.13	0	-3.11
Violating paths	45	31	0	14
All paths	12763	7680	200	8143

Results of the industrial hold optimization using buffers is shown in table3 and the design is having utilization of 70.057%. Table 3 shows register to register path is having worst hold slack value of 130ps and also 31 reg2reg paths and default paths 14 are violated which results in a total hold slack value of 6.24 ns.

Table 4: Results of the implemented approach after CTS stage

Utilization of the design is 39.623%

Hold mode	All	Reg2reg	Reg2cgate	Default
WHS (ns)	-0.13	-0.13	0.25	0.03
THS (ns)	-11.84	-11.84	0	0
Violating paths	136	136	0	0
All paths	12763	7680	200	8143

Results of the implemented approach after Clock Tree Synthesis (CTS) stage using buffers, inverters and delay cells together with industrial hold optimization flow are shown in table4 and the design is having utilization of 39.623%. Results of the implemented approach after Routing stage using buffers, inverters and delay cells together with industrial hold optimization flow are shown in table5 and the design is having utilization 39.685%.

Table 5: Results of the implemented approach after routing stage

Utilization of the design is 39.685%

Hold mode	All	Reg2reg	Reg2cgate	Default
WHS (ns)	-0.09	-0.09	0.18	0.29
THS (ns)	-0.15	-0.15	0	0
Violating paths	2	2	0	0
All paths	12763	7680	200	8143

Compared to table3 in which buffers are used together with industrial hold optimization, slack has been improved about 31% as shown in table5 where buffers, inverters and delay cells are used as delay elements (i., e.130 ps to 90 ps). Violating paths in table3 are 45 whereas in the implemented approach only 2 paths are violated having a total hold slack value of 150 ps. Furthermore utilization of the design in the implemented approach is 39.685% which represents better utilization and also additional logic circuitry can be added for future purpose.

VI. CONCLUDING REMARKS

In this paper, a standard industrial circuit level design is taken to compare two methodologies. One method uses only buffers for optimization and another method uses combination of delay elements such as buffers, inverters and delay cells. The implemented method uses insertion of delay elements for timing closure. At first, a linear programming optimization is used to compute the amount of hold delay that is required to overcome

the violated hold path slack. Then the delay that is obtained is realized as a combination of delay elements with the help of a dynamic process based algorithm in bottom up delay insertion. The optimization flow for a circuit level design is also presented. The implemented approach where combination of delay elements are used for optimization and using buffers only for optimization are tested on industrial design together with the state of the art industrial hold optimization flow, better results are achieved in terms of minimal hold slack, hold violations and utilization are reported.

REFERENCES

1. Pei-Ci Wu ; Martin D. F. Wong ; Ivailo Nedelchev ; Sarvesh Bhardwaj ; Vidyamani Parkhe on timing closure: buffer insertion for hold violation removal 51st ACM/EDAC/IEEE Design Automation Conference (DAC) Pages:1 – 6,2014
2. S.-H. Huang, C.-H. Cheng, C.-M. Chang, and Y.-T. Nieh. Clock period minimization with minimum delay insertion. InDesign Automation Conference, 2007. DAC'07. 44th ACM/IEEE, pages 970–975. IEEE, 2007.
3. S.-H. Huang, G.-Y. Jhuo, and W.-L. Huang. Minimum buffer insertions for clock period minimization. In Computer Communication Control and Automation (3CA), 2010 International Symposium on, volume 1, pages 426–429.IEEE, 2010.
4. A.B. Kahng, J. Lienig, I. L. Markov, and J. Hu. VLSI Physical Design: From Graph Partitioning to Timing Closure. 2011..
5. B. Lin and H. Zhou. Clock skew scheduling with delay padding for prescribed skew domains. In Design AutomationConference, 2007. ASP-DAC'07. Asia and South Pacific,pages 541–546. IEEE, 2007
6. M. M. Ozdal, S. Burns, and J. Hu. Gate sizing and device technology selection algorithms for high-performance industrial designs. In Proceedings of the InternationalConference on Computer-Aided Design, pages 724–731.IEEE Press, 2010.
7. N. V. Shenoy, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. Minimum padding to satisfy short path constraints. In Computer-Aided Design, 1993.ICCAD-93. Digest of Technical Papers., 1993 IEEE/ACM International Conference on, pages 156–161. IEEE, 1993.
8. W.-P. Tu, C.-H. Chou, S.-H. Huang, S.-C. Chang, Y.-T. Nieh, and C.-Y. Chou. Low-power timing closure methodology for ultra-low voltage designs. In Proc. Int.Conf. on Computer-Aided Design, pages 697–704, 2013.