Minimizing Skew and Delay with Buffer Resizing and Relocation during Clock Tree Synthesis

Preeti Punia, Rouble, Neeraj Kr. Shukla, and Mandeep Singh
Department of Electrical, Electronics & Communication Engineering
ITM University, Gurgaon (Haryana) India

ABSTRACT
Rapidly increasing design complexity due to small size and higher speed, results in the problem of clock skew and insertion delay. These are the two important parameters which should be considered for successful completion of the design. In this work, a method for minimizing clock skew by buffer insertion and resize is proposed. Clock skew will be minimized during post-CTS timing analysis after placement of standard cells during physical implementation of the design. Also, buffer relocation method is used for minimizing the delay of the cells. Simulations were carried out on EDAl tools and results show that overall skew is improved by 23.95% and delay is improved by 19.50%.

Keywords
Buffer, CTS, Delay, Skew, Slack.

1. INTRODUCTION
The premium challenge in the field of VLSI design nowadays is the designing of a VLSI chip using lower process technologies. As the design complexity is rapidly increasing due to the small design size, higher clock frequency and lower voltage, higher speed, the market windows continues to shrink. As a result, insertion delay and Clock skew are the two emerging dominant factors which should be managed in order to successfully complete designs in a precise manner. Now, simply by making improvements to existing tools and methodologies will not address the full extent of these issues. There are many design variable interdependencies which are to be dealt with. The most important amongst them is the trade-off between power consumption, routability, and timing. Optimization of one design variable can cause problems with the others. Large amounts of routing resources are consumed by power and clock networks. Their construction and analysis must be started early and they should be customized for an individual chip’s demand. Clock trees are inserted after the placement of standard cells is complete. Physical design is as important as logic design in determining whether the demands are achievable or not. Design size has surpassed the limitations of gate-level design tools. The chip must be planned at high level, partitioned into smaller pieces, and then design should be completed using lower level tools. During physical implementation of the design, we come across Clock Tree Synthesis (CTS). It is the process of buffer/inverter insertion along the clock paths of the design so as to achieve minimum skew or optimized skew. The premium goal of Clock Tree Synthesis is to minimize insertion delay and skew. The concept of useful skew can also be added in the design using inverters and buffers. The key factor in physical design is to optimize the slack values and the arrival times of the clock i.e., skew and delay to meet the timing requirement of the design.

The slack can be defined as the difference between the required time and the arrival time of the clock pulse. A positive slack at a node implies that the arrival time at that node may be increased by a factor without affecting the overall delay of the circuit. Negative slack implies that a path is too slow, and the path must be speed up (or the reference signal delayed) if the whole circuit is to work at the desired speed. Clock skew is the difference between the actual and expected arrival time of a clock signal at two different flops. Clock skew results in slow system performance as it uses comparatively large time period between clock pulses. It can also cause set-up time or hold time violations. Therefore, clock skew must be minimized during clock routing. It is an elementary design principle that the timing should satisfy register’s setup and hold-time requirement. Timing violations may occur due to clocking of sequentially adjoining registers on the edge of a high skew clock i.e. the same clock edge and it can also be responsible for functional failures. Figure 3.1 represents sequentially- adjoining registers, in which a local routing structure is used for routing of the clock signal. Here, clock skew can be noticed. In Figure 1, all the three registers are clocked at the same clock edge. Here, each of the three registers has different arrival time of the clock edge. Figure 2 illustrates a case of the clock skew for the circuit shown in Figure 1.

Fig. 1: Sequentially Adjacent Registers with Clock Skew

Fig. 2: Clock Arrival Time Fluctuations in the circuit of Figure 1
Mainly skew is of two types, positive skew and negative skew. Figure 3, 4 and 5 shows the case of zero, positive and negative clock skew.

![Figure 3: Zero Clock Skew](image1)

If the clock comes late than the launch time clock then it is called positive skew. In this case, data and clock flows in same direction. In other words, positive skew is when routing of the data and the clock is in the same direction. It can lead to hold violation whereas having a positive skew improves setup time.

![Figure 4: Positive clock skew](image2)

If the clock comes early than launch time clock it is called negative skew. In this case, data and clock travel in opposite direction. It is said to have negative skew when the data and clock are routed in opposite direction. Negative skew can lead to setup violation whereas it improves hold time.

![Figure 5: Negative clock skew](image3)

There are a few well-known design techniques to make designs more robust against clock skew. Some of them are adding delay in data path, clock reversing, alternate phase clocking and adding buffers and inverters for optimized skew. In paper section 2 gives the detailed literature survey on minimization of clock skew and buffer insertion, section 3 explains the buffer resizing and relocation technique used for the minimization, section 4 discusses the simulation of the design and the results of the skew and delay and section 5 discusses the conclusions and future scope of the presented work.

### 2. LITERATURE SURVEY

Clock tree is the most commonly used structure for clock distribution networks. The methodology by [1] is responsible for fixing of the timing violations with data path optimization, by incrementally adjusting the clock arriving time of proper register candidates using buffer insertion/removal, without affecting the global clock tree synthesized before. With a slight increase in delay, the values of WNS (worst negative slack) and TNS (total negative slack) were improved with 33.16% improvement on WNS and 75.56% improvement on TNS. A technique based on linear programming (LP) which aims at reduction of skew and a delay mapping algorithm dynamically adjusting skew aberration to avoid mapping inaccuracy is discussed in [2] with an improvement of 10.30% of the maximum clock skew was observed. Despite the fact that clock networks or check matrices are utilized within some microchip plans [3], they are less favored because of the high clock power and routing cost. The as of late advertised Intel dual-core Titanium processor has re-embraced clock tree due to the power attention [4]. A clock tree is synthesized from the positions and capacitance heaps of the clock sinks. The main step is topology era through partitioning the clock sinks. It is then trailed by directing and optimization steps. Tsay [5] proposes a zero-skew clock steering algorithm for a given clock tree topology. For a wire interfacing two clock sinks, the algorithm discovers the tapping point on the wire that has the same clock delay to both clock sinks. The tapping point turns into another clock sink and the algorithm repeat in a bottom-up style until it achieves the clock root. The aggregate wire length of the final clock tree is reliant on the clock tree topology and the routing of each one wire that interfaces two clock sinks. Chao et al. [6] propose a balanced-bipartition (BB) algorithm that creates the clock tree topology and the Deferred- Merge-Embedding (DME) algorithm that decides the steering of each one wire. The joined together BB+dme algorithm accomplishes a 10% normal wire length savings contrasting with [5]. In [7, 8], the DME algorithm is stretched out to handle bounded-skew and useful-skew imperatives. The destination of these works points at minimizing the aggregate wire length of a clock tree, which thus minimizes the power consumption. In spite of the power point of interest of clock trees over clock grids, clock trees are more vulnerable to process varieties than the clock grids. Notwithstanding, process-induced clock skews are not known until the clock tree usage is carried out. To lessen the weakness of a clock tree to process varieties, a metric to anticipate process induced clock skews is required. Experimentally, it is watched that process-induced clock skews can achieve 10% of the clock delay [9]. Consequently, minimizing clock delays can minimize process induced clock skews. Interconnect delay streamlining procedures, which incorporate buffer sizing, buffer insertion and wire sizing, are pertinent to clock delay optimization.

The contrast between synthesizing a signal net and a clock tree is that for a signal net, the design target is to minimize the maximum delay from the signal source to any of its sign signal collectors, while a clock signal needs to be dispersed to
each one time sink at the recommended time. Two superb reviews of interconnect improvement systems are accessible in [1, 2]. A lot of lives up to expectations are focused around dynamic programming. Van Ginneken [10] proposes an element programming algorithm to tackle the buffer placement issue. Given a dispersed RC-tree, a buffer library, and the legal buffer positions, the issue points at discovering the buffering choice that minimizes the most extreme delay from the base of the RC-tree to any of its leaf nodes. In [11], discrete wire measuring for general routing trees is expected and a bottom-up dynamic programming methodology is utilized to spread the ideal wire estimating choices around the root node. Designers can then pick a choice by making an exchange off between delay and power utilization. In [12, 13], bottom up dynamic programming algorithms are reached out to think about concurrent buffer insertion and wire sizing procedures. The DME algorithm for zero-skew clock routing is additionally a dynamic programming algorithm, in which competitor zero-skew tapping focuses are put away as unifying sections. Developed works [14–16] permit a zero-skew clock tree to accomplish better clock delay and power by buffer insertion and wire measuring. In [17], an sensitivity based iterative algorithm performs wire sizing one portion at once and something like 1.5x to 3x enhancements on least delay are watched. In [18–20], the concurrent buffer insertion, buffer measuring and wire estimating issues are defined as optimizing issues, in which the maximum delay of each one sink node is obliged. Iterative and Lagrangian Relaxation techniques are then used to tackle the issue. In [21], iterative algorithms are utilized to diminish clock delay and clock skew through wire sizing focused around sensitivity data. Zeng et al. [22] propose a three-stage improvement algorithm, i.e., buffer insertion, skew and delay optimization, to minimize the skew and delay of a clock tree. Contrasting the initial unbuffered clock tree, a most extreme of 27x delay change is attained by buffer insertion and sizing. Chen et al. [23] plan the clock area/delay/power minimization issue for buffered clock trees as a geometric programming issue and tackle it utilizing Lagrangian Relaxation techniques. The current clock tree optimization lives up to expectations, either dynamic-programming-based or numerical programming-based, either just uses one or two accessible improvement systems, i.e., buffer insertion, buffer sizing and wire sizing, or think as of them in independent stages. The conflicting stages in the current part are given in Fig. 5 to demonstrate the possible range. Albrecht et al. [32] received the minimum balance algorithm [36] to circulate the timing edges so it yields a lexicographically greatest slack vector when the slacks are sorted in non-decreasing request. The ideal clock period could be acquired through a binary search between [0, Dmax], where Dmax is the greatest combinational path delay, by applying the Bellman-Ford algorithm [34, 35] on the timing chart. An option is to tackle the parametric shortest path issue by a path-pivoting algorithm [36]. Nonetheless, these methodologies don't mull over the factual conduct of process variations and likewise their results are not confirmed with any timing yield model.

3. BUFFER RESIZE AND RELOCATION

In this paper, we specify a method to improve skew and delay in the design during clock tree synthesis. During physical implementation, comes the CTS (Clock Tree Synthesis) which is the process of insertion of buffers or inverters along the clock paths of ASIC design in order to achieve zero/minimum skew or balanced skew. The goal of CTS is to minimize skew and insertion delay. Clock is propagated after placement because the actual physical location of modules and cells are required for the clock’s propagation which in turn results in dealing with accurate delay and operating frequency. Values of skew, WNS, TNS and delay were observed and then the clock design was optimized in iterations. While buffers are inserted in the design, we resize them to get various values for WNS and TNS. Now, we select the buffer which gives maximum positive value for Worst negative slack (WNS) and total negative slack (TNS). If we get an optimum value for these slacks before optimizing the design for setup and hold conditions, we prefer using that buffer size. Now, on the basis of buffer selection, we observe the arrival times of various registers or cells used in the design. Based on the arrival time of the various registers, skew was calculated for different cells. Further, for improvement in clock delay values, the used buffer was relocated from its position and this was carried out in several iterations to get the minimum value for the clock delay. Figure 6 show the design before buffer insertion and figure 7 show the buffer insertion in the design.

![Figure 6: Design before buffer insertion](image-url)
4. SIMULATIONS AND RESULTS
With a design concept in mind, the RTL has been written in Verilog language. Verification of design was done and, it was required to synthesize the code in order to work further in the design implementation. The synthesized code generated the gate-level Netlist and the constraint file (.sdc) was exported which was further used in the next step of physical implementation which involved floorplanning, placement, routing and finally generation of GDS-II file. The simulation and synthesis of design was carried out on EDA tools. Table 1 shows the experimental results for percentage improvement in delay. Table 2 shows the experimental results on percentage improvement in skew.

Table 1: Improvement table for skew

<table>
<thead>
<tr>
<th>Cell</th>
<th>Fan-out</th>
<th>Skew (ps) (before)</th>
<th>Skew (ps) (after)</th>
<th>Improvement in skew (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLY4X1</td>
<td>1</td>
<td>0.499</td>
<td>0.355</td>
<td>28.858</td>
</tr>
<tr>
<td>XNOR2XL</td>
<td>4</td>
<td>0.100</td>
<td>0.088</td>
<td>12.000</td>
</tr>
<tr>
<td>INVXL</td>
<td>4</td>
<td>0.241</td>
<td>0.210</td>
<td>12.863</td>
</tr>
<tr>
<td>OAI2BB1X1</td>
<td>3</td>
<td>0.186</td>
<td>0.140</td>
<td>24.731</td>
</tr>
<tr>
<td>AOI22XL</td>
<td>1</td>
<td>0.188</td>
<td>0.100</td>
<td>46.809</td>
</tr>
<tr>
<td>OAI211XL</td>
<td>1</td>
<td>0.569</td>
<td>0.423</td>
<td>25.659</td>
</tr>
<tr>
<td>DLY2X1</td>
<td>1</td>
<td>1.131</td>
<td>0.942</td>
<td>16.711</td>
</tr>
<tr>
<td>DLY4X1</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>DFFRX1</td>
<td>1</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Table 2: Improvement table for delay

<table>
<thead>
<tr>
<th>Cell</th>
<th>Delay (ps) (before)</th>
<th>Delay (ps) (after)</th>
<th>Improvement in delay (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLY4X1</td>
<td>1.438</td>
<td>1.348</td>
<td>6.259</td>
</tr>
<tr>
<td>XNOR2XL</td>
<td>0.499</td>
<td>0.389</td>
<td>22.044</td>
</tr>
<tr>
<td>INVXL</td>
<td>0.137</td>
<td>0.087</td>
<td>36.496</td>
</tr>
<tr>
<td>OAI2BB1X1</td>
<td>0.241</td>
<td>0.186</td>
<td>28.822</td>
</tr>
</tbody>
</table>

5. CONCLUSIONS AND FUTURE SCOPE
For minimizing the clock skew a method of buffer insertion and resize is proposed i.e. clock skew was minimized during post-CTS timing analysis in physical implementation of the design. Also, buffer relocation method was used for minimizing the delay of the cells. Simulations were carried out on several EDA tools. As per the observation tables, we can see that overall skew was improved by 23.95% and delay is improved by 19.50% by buffer insertion, resizing and relocation. Further, the work can be extended for useful skew optimization.

6. REFERENCES


[37] Cadence’s “SOC Encounter RTL to GDSII System Datasheet” available on URL:

[40] Video Tutorial “Place and Route with Cadence SOC Encounter (Basics)” Online available at URL: http://www.youtube.com/watch?v=Z5WKIDbtd4g.


ABOUT THE AUTHORS
Preeti Punia has done her Master of Technology (M.Tech) in VLSI Design from ITM University, Gurgaon (Haryana) India. She completed her B.Tech. in Electronics and communication Engineering from Haryana College of Technology and Management affiliated to Kurukshetra University, Kurukshetra (Haryana) India. The area of her research interest includes RTL Design and Verification, Digital VLSI Design, EDA Tools Designing and Physical Design.

Rouble has done her M.Tech. in ECE (VLSI Design) from NIT Kurukshetra in 2013. She was a gold medalist of her batch during M.Tech. at NIT, Kurukshetra (2011-2013). She completed her B.Tech. in Electronics & Communication Engineering from DCRUS&T, Murthal, Sonepat (Haryana) in 2011 and is GATE qualified. She joined ITM University as Assistant Professor in January, 2014. The broad area of her research interest is VLSI Design including Digital IC Design, Semiconductor device characterization and modeling, Low power VLSI Design and Analog IC Design.

Neeraj Kr. Shukla (IETE, IE, IACSIT, IAENG, CSI, ISTE, VSI-India), an Associate Professor in the Department of Electrical, Electronics & Communication Engineering, and Project Manager – VLSI Design at ITM University, Gurgaon, (Haryana) India. He received his PhD from UK Technical University, Dehradun in Low-Power SRAM Design and M.Tech. (Electronics Engineering) and B.Tech. Electronics & Telecommunication Engineering) Degrees from the J.K. Institute of Applied Physics & Technology, University of Allahabad, Allahabad (Uttar Pradesh) India in the year of 1998 and 2000, respectively. He has more than 50 Publications in the Journals and Conferences of National and International repute. His main research interests are in Low-Power Digital VLSI Design and its Multimedia Applications, Digital Hardware Design, Open Source EDA, Scripting and their role in VLSI Design, and RTL Design.

Mandeep Singh, Ph.D. Scholar at ITM University,Gurgaon Asst. Professor in the Department of Electrical, Electronics and Communication Engineering, ITM University, Gurgaon, (Haryana), India. He has received his M.Tech. and B.Tech. from Sant Longowal Institute of Engg. & Tech., Longowal, Sangrur, Punjab Technical University, Jalandhar (Punjab) in the year of 2008 and 2006 respectively. His main research interests are in Control Systems, Signal and Systems, Optimization techniques and its engineering application.