

A Study on Placement of Post Silicon Clock Tuning Buffers for Mitigating Impact of Process Variation

Kelageri Nagaraj, Sandip Kundu
University of Massachusetts, Amherst, USA
Email: {nagaraj, kundu}@ecs.umass.edu

Abstract

Optical shrink for process migration, manufacturing process variation, temperature and voltage changes lead to clock skew as well as path delay variations in a manufactured chip. Such variations end up degrading the performance of manufactured chips. Since, such variations are hard to predict in pre-silicon phase, tunable clock buffers have been used in several designs. These buffers are tuned to improve maximum operating clock frequency of a design. Previously, we have presented an algorithmic approach that uses delay measurements on a few selected patterns to determine which buffers should be targeted for tuning. In this paper, a study on impact of tunable buffer placement on performance is reported. Greatest benefit from tunable buffer placement is observed, when the clock tree is designed by the proposed tuning system assuming random delay perturbations during design. Accordingly, we present a clock tree synthesis procedure which offer very good protection against process variation as borne out by the results.

1 Introduction

As integrated circuit (IC) technologies scale to 45nm and beyond, process variations are becoming increasingly critical for circuit performance. A number of pre-silicon approaches have been proposed to mitigate this problem using statistical design optimization techniques. These approaches optimize the gate sizes and threshold voltage assignments voltages to maximize yield [1], [2], [3], [4], [5], [6], [7]. The basic principle behind these approaches is to use statistical models to maximize the number of chips that will meet power and delay constraints in fabricated circuits. Regardless of how a design is optimized, it is inevitable that for some population of chips, the post-silicon performance will fall short of the expectation.

To address this problem, insertion of post silicon tunable buffers have been proposed [8], [9], [10]. By controlling tuning inputs, the delay of these buffers can be varied to compensate for process variations with the aim to maximize the circuit frequency.

Typically, such buffers are inserted in the clock distribution network. By tuning these buffers (i) clock skew can be compensated and (ii) beneficial clock skew

may be introduced to make a design run faster. While the first case is well known, the second case is also of equal importance and has been explained later with an example.

The greatest challenge in post silicon tuning of clocks is not in the design but in determining what tuning settings should be applied for maximizing performance. The answer to this question is also tied to the specific problem being addressed.

In this paper, we are addressing the process variation problem. Process variation affects different dies differently. Accordingly, each die should have its unique tuning setting to maximize its performance. Also the clock tree should be structured in such way that the tuning settings should not disturb the elements which will manifest failure at the output, if tuned. The inference process for mapping failure data to tuning setting is simple and requires no delay model for the design. The only information known to the observer are the input output responses obtained from a tester. Thus, the data from the tester needs to be interpreted algorithmically to output tuning information. The tuning is specific to each die and it will mitigate the effects of process variation. Our tuning system structures the clock tree so that the tuning settings are not self-conflicting and improve the performance. High performance microprocessors are *binned* to their operating frequency. Down binning a part results in a cost penalty. Thus, a binning-yield loss may be defined. By generating optimal tuning setting our goal will be to minimize the binning yield loss.

The rest of the paper is organized as follows: in Section 2, we review the previous work. Section 3 describes preliminaries which includes basic concepts and simple explanation about our proposal. Section 4 shows results of our experiments on ISCAS 89 benchmark circuits. Finally, in section 5, we present the conclusion and propose our future extensions of this work.

2 Previous Work

In nano-CMOS designs, process variations result in a statistical spread in the achievable frequency, thereby causing some chips to fail from meeting the nominal target frequency. In [4], Borkar et. al. have suggested that as much as 30% frequency variation can be observed in high-performance microprocessors. A lot of recent work focuses on statistical techniques for considering process

variability during analysis and optimization. Statistical timing analysis has been used as a tool to predict the timing distribution of the designs [12], [13]. Many approaches have been tried to utilize this information to perform statistical optimizations like gate sizing [3], [4], [5], [6]. These techniques are used pre-silicon to counter variability.

Post-silicon tuning is another approach to reduce binning yield loss in circuits. This would allow a manufacturer to tune each chip to improve frequency. Recently, Post-Silicon Tunable (PST) clock-tree synthesis [8], [9], [10] has been proposed as one such approach that can be applied to high performance designs to correct timing violations.

Most of the work in this area focuses on design optimization. Chen et al. have proposed a timing driven post silicon tunable clock tree synthesis [14], [15]. Srivastava et al. [16] have proposed a simultaneous gate sizing and PST buffer range determination using statistical timing. These approaches rely on available timing information and need delay models. E. G. Friedmen et al. [17] have proposed a methodology to take care of the effects of process variation on the clock distribution network by calculating the clock skew range of local data path, and using this skew to reduce the clock period. Fishburn [18] has proposed optimization of clock skew using linear programming. But these schemes do not take care of the PST buffer tuning.

By contrast, our work is focused on finding appropriate settings for tuning elements based on information obtained from the tester. These settings increase the performance of the design and alleviate the effects of process variation. Our scheme does not rely on internal delays because such information is not easy to obtain from a chip without extensive and costly instrumentation.

3 Proposed Approach

A sequential design consists of a set of flip-flops and combinational logic between the flip-flops. Let us represent this with a graph $G = (V, E)$, where V is the set of flip-flops and E is the set of edges representing the combinational logic between the flip-flops. Let i and j be two flip-flops and c_{ij} be the edge representing the combinational logic between them. Let $D(c_{ij})$ be the maximum delay of the combinational logic c_{ij} . Let T_i and T_j be the clock arrival time at i and j . In order to avoid timing violations, flip-flop j should capture the data which was launched by the flip-flop i . For this to happen, following equation has to be satisfied

$$T_i + D(c_{ij}) < T_{clk} + T_j - T_{setup} \quad (1)$$

Where T_{clk} is the clock period, T_{setup} is the setup time of the flip-flop. PST buffers can control the clock arrival times T_i and T_j . Equation 1 can be satisfied by reducing T_i or increasing T_j . This means that the PST buffers can be

used to introduce beneficial clock skew to satisfy timing constraints. In other words, the timing slack is adjusted between the critical and non-critical paths. Similarly the hold time constraints can also be satisfied by using the PST buffers.

3.1 Tuning Process

The tuning process used in this paper, is described in [21]. It is an algorithmic process based on Critical Path Tracing (CPT) [19][20][21]. For the sake of brevity, the details are omitted here.

3.2 Tuning Target Selection

A critical step in tuning process is tuning target selection, where a list of target flip-flops for early or late clock adjustments are identified. The tuning target selection steps are described in [21]. It is the starting point for the clock-tree synthesis described in this paper.

3.3 Clock Tree Synthesis for PST Buffers

Clock Tree Synthesis is a design time activity, not a post-silicon activity. However, if the hardware tester is replaced by a timing simulator and what if questions are asked, sensitive flip-flops can be easily identified as above. Please also note that these flip-flops take Boolean relationships into account as well. With this assumption, following modification to critical path tracing is used to help design the clock tree.

Critical Path Tracing stops at a Flip-Flop. This does not mean that every flip-flop at the leaf level of the clock tree will have its own PST buffer. The PST buffers are placed at a higher level in the clock tree. So if we tune a particular PST buffer, then its equivalent to tuning all the flip-flops in that particular sub tree. As shown in Figure 1, by tuning the buffer PST1, we are effectively tuning flip-flops FF1, FF2 and FF3. Thus by implication, we can derive which PST buffer(s) needs to be tuned and which flip-flops are tuned as a consequence.

From Table III, in order to tune FF1 flip-flop, we tune the PST buffer PST1 in Figure 2. This results in the tuning of flip-flops FF1, FF2 and FF3. In order to tune a single flip-flop, we end up changing additional flip-flop timings which may result in creating timing failures at the other outputs. As the placement of the PST buffers move to higher levels in the clock tree, the number of flip-flops which get affected with the tuning of a single PST buffer, increases. This increases the probability of failure at the outputs, as larger number of flip-flops are disturbed.

In order to avoid this problem, we use our tuning list and reshuffle the clock tree such that the tuning element which were listed for early tuning are made to fall under the control of a single PST buffer and the flip-flops which were listed for late tunings are put under different PST buffers.

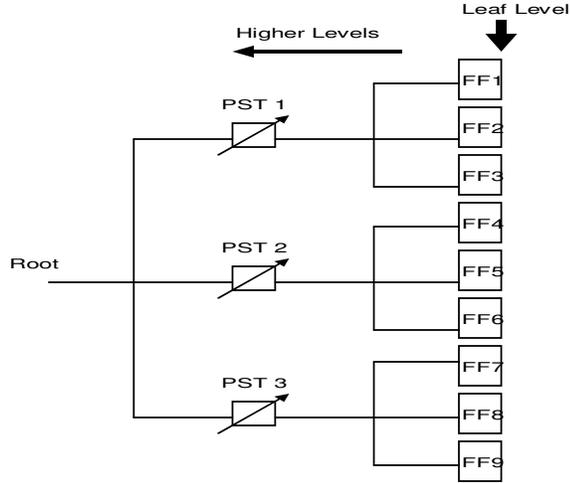


Fig. 1: Clock Tree with PST buffer

This is a greedy algorithm for clock tree synthesis. By altering the clock tree based on our tuning system, we were able to see a performance gain of around 10%, even when we tuned the PST buffers which were 2 levels above the leaf nodes of the clock tree. The important thing to note here is that the clock-tree reshuffling is done pre-silicon and this will help us in tuning the PST buffers in post silicon. In other words, the clock-tree synthesis is tuning system driven in the pre silicon phase. The results are explained in following section.

4 Experimental results

Simulation experiments were conducted on ISCAS-89 benchmark circuits for seeking performance improvement opportunities. In order to simulate the effects of process variation, multiple instances of each circuit were created. In each instance, the gate delays were modified based on sampling from a normal distribution of gate delays for 45nm process node.

The benchmark circuits are synthesized using Synopsys Design Compiler; transition test patterns are generated using Synopsys Tetramax. The resulting structural Verilog circuit is subjected to timing verification using ATPG patterns as inputs to these circuits. Timing verification is done for the targeted clock frequency and the failure patterns and the failure outputs are taken through our tuning system. The tuning system generates the tuning list consisting of the flip-flops, which have to be tuned to improve the performance.

Process variation is introduced by perturbing the gate delays in a circuit following a normal distribution. By repeating this process multiple times, multiple instances of circuits with process variation are created. The clock tree was generated as explained in section 3

Figure 2 plots the normal distribution of the minimum clock period for the different instance of the ISCAS benchmark circuit s298. The results are based on ATPG

patterns. In Figure 2, the distribution indicated by “before”, is the distribution of minimum clock periods, when no tuning settings were applied. The distribution indicated by “Leaf”, is the distribution when the tuning setting were applied, assuming that all the leaf nodes were tunable.

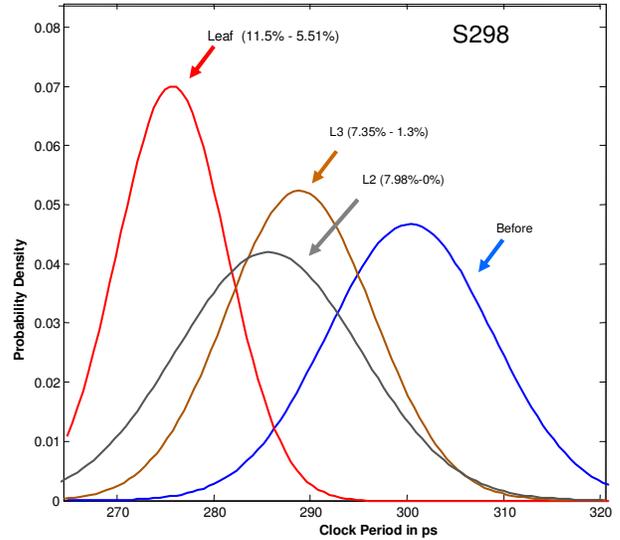


Fig. 2: Tuning settings for s298 with PST buffers at different levels of clock tree- ATPG

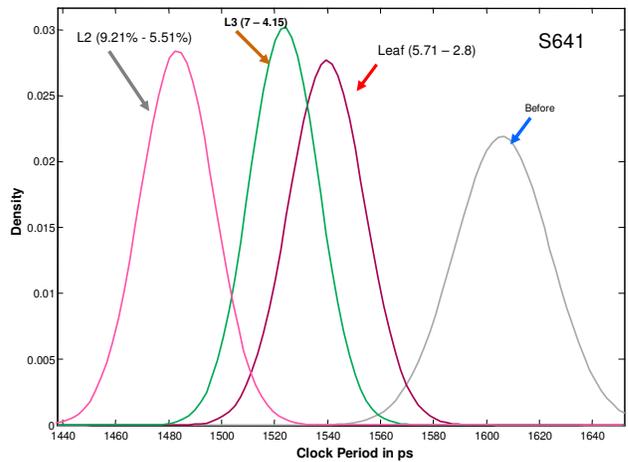


Fig. 3: Tuning settings for s641 with PST buffers different levels of clock tree- Random Patterns

The distribution indicated by “L3”, is the distribution when the tuning settings were applied to the PST buffers placed at one level above the leaf level (closer to the clock source). This showed a performance improvement in the range 7.35% to 1.3%. The decrease in the performance as compared to the leaf level, justifies our explanation in section 3, which indicated that by tuning the PST buffer at the higher level, we end up disturbing unwanted flip-flops. Similarly, the distribution indicated

Ckts	Performance Improvement – ATPG Patterns											
	Leaf				L3				L2			
	μ (mean)	σ	Min	Max	μ (mean)	σ	Min	Max	μ (mean)	σ	Min	Max
S27	1.56	2.06	0	6.25	-	-	-	-	-	-	-	-
S298	8.16	2.13	5.51	11.5	3.67	2.13	1.35	7.35	4.44	3.04	0	7.35
S444	8.8	1.37	6	10	8.82	1.4	6	10.1	8.9	.83	7.58	10
S526	7.58	.69	6.5	8.7	7.76	1.12	5.3	8.7	0.17	0.53	0	1.7
S641	9.2	0.42	8.7	9.83	9.2	0.42	8.7	9.2	9.2	0.42	8.7	9.83

Fig. 4: Performance improvement for ISCAS-89 benchmark circuits with PST buffers at different level

“L2” indicates that the tuned PST buffers were assumed to be two levels above the leaf level. This showed a performance improvement in the range of 7.35% - 0%. Again the explanation given in section 3 holds good here. Figure 3 shows the results for the random patterns and our concept holds good for random patterns as well. Figure 4 shows the performance improvement for ISCAS-89 benchmark circuits when the results were generated for ATPG patterns. In order to validate our results, we conducted our experiment using random patterns and our claim held good for random patterns as well. For brevity, we haven’t shown it here.

5. Conclusion and Future Work

Post silicon clock buffer tuning is one of the available remedies for process variation. We have presented a novel post silicon clock tuning system which uses only the external data available to generate the tuning settings. Previous approaches were focused on delays but ignored Boolean relationship between inputs and outputs. By focusing on logic functionality, we take full advantage of Boolean relations between inputs and outputs. Using the tuning system, we also create an efficient clock tree in the pre silicon phase which offers further increase in performance. Dynamic simulation on benchmark circuits with circuit delays show around 9% average improvement in performance, exceeding our initial expectations about benefits of PST buffers. We plan to extend this work further to find ideal tunable buffer locations when only a fixed budget of such buffers may be used.

Acknowledgement

This research is supported in part by a grant from Intel Corporation. We also thank mentors from our industrial sponsor for many insightful discussions on the subject.

6. References

- [1] Mani *et al.*, “An efficient algorithm for statistical minimization of total power under timing yield constraints,” Proc. DAC, 2005.
- [2] Nassif, “Delay variability: sources, impacts and trends,” Proc. ISSCC, 2000.
- [3] Singh *et al.*, “Robust gate sizing by geometric programming,” Proc. DAC, 2005.
- [4] Borkar *et al.*, “Parameter Variations and Impact on Circuits and Microarchitecture”, Proc. DAC, 2003.
- [5] Srivastava *et al.*, “A General Framework for Accurate Statistical Timing Analysis Considering Correlations”. Proc. DAC, 2005.
- [6] Agrawal *et al.*, “Circuit Optimization Using Statistical Static Timing Analysis”. Proc. DAC, 2005.
- [7] Sinha *et al.*, “Statistical Gate Sizing for Timing Yield Optimization”. ICCAD, 2005.
- [8] Desai *et al.*, “Clock generation and distribution for the first IA-64 microprocessor”. Proc. IJSSC, 2000.
- [9] Naffziger *et al.*, “Clock distribution on a dual-core multi-threaded Itanium-family processor”. Proc. ISSCC, 2005
- [10] Takahashi *et al.*, “A post-silicon clock timing adjustment using genetic algorithms”. Proc. VLSI circuits, 2003
- [11] Srivastava *et al.*, “Variability-Driven Gate Sizing for Binning Yield Optimization”. Procs. DAC, 2006
- [12] Sapatnekar *et al.*, “Statistical Timing Analysis considering Spatial Correlations Using a Single Pert-Like Traversal”. Procs. ICCAD, 2003.
- [13] Zhan *et al.*, “Correlation-aware Inbox statistical timing analysis with non-gaussian delay distributions”. Procs. DAC, 2005
- [14] Chen *et al.*, “Statistical timing analysis driven post-silicon-tunable clock-tree synthesis”, Proc ICCAD 2005
- [15] Chen *et al.*, “A yield improvement methodology using pre-and post-silicon statistical clock scheduling” Proc. ICCAD, 2004
- [16] Srivastava *et al.*, “Variability-driven formulation for simultaneous gate sizing and post-silicon tunability allocation,” Proc. ISPD, 2007
- [17] Friedman *et al.*, “Optimal clock skew scheduling tolerant to process variations” Proc. DAC, 1996
- [18] J. P. Fishburn, “Clock skew optimization” IEEE Transactions on Computers, 1990
- [19] Abramovici *et al.*, “Critical Path Tracing: An Alternative to Fault Simulation,” Proc. ITC, 1999
- [20] Girard *et al.*, “A reconvergent fanout analysis for the CPT algorithm used in delay-fault diagnosis” Proc. ETC, 1993
- [21] K Nagaraj, S Kundu, “An Automatic Post Silicon tuning system for performance gain using tester measurements” Proc. ITC 2008