# Clock Tree Synthesis Careabouts for Complex SoCs

Vishweshwara R, vishwa@ti.com
Texas Instruments India

Santhosh Thiyagaraja, santhosht@ti.com
Texas Instruments India

Rivu Das, rivudas@ti.com
Texas Instruments India

Venkatraman Ramakrishnan, rvenkat@ti.com
Texas Instruments India

Mahita Nagabhiru, mahita@ti.com
Texas Instruments India

## 1 Introduction

With increasing integration trends into System on Chips (SoC), the clock architectures are becoming more complex. Complex clock generation and shaping logic, gating architectures are common place in most designs. Synthesizing such clock trees have, thus, become very challenging and arduous. Also, the high performance, low power goals emphasize the need for clock trees that are smaller on insertion delay, area, skew and resistant to process, temperature, voltage variations. In this paper we present a few care-abouts on aspects of clock tree planning and synthesis to ensure good clock tree synthesis (CTS) quality entitlement. The first section presents aspects of clock planning. This includes creation of clock tree synthesis constraints from the various operational mode constraints, modeling the effect of CTS on placement and routing utilization early in the flow. The section also presents some placement careabouts for clock tree logic like clock gates and register placement techniques for better clock tree power and skew. The next section presents a few considerations for clock tree synthesis to ensure clock tree quality with process, temperature and voltage variations. The section also presents a few techniques to reduce the impact of clock tree switching on the dynamic voltage drop variations of the design. The paper presents some results emphasizing the clock tree quality improvements harnessed with these techniques.

## 2 Clock Tree Planning Considerations

### 2.1 Clock Tree Constraints Generation

It is very common to have multiple modes of operation of an SoC for various functional and test requirements. Creating one constraints mode to cater to all these design modes is difficult and cumbersome. While EDA solutions are now available to perform placement and timing optimization concurrently across the different modes, multi mode clock tree synthesis is still a concern. Having numerous generated clocks, case analysis at various nodes of the designs is not very conducive for good CTS quality entitlement. There is hence a need for a good constraints merging technique from a CTS standpoint.

This paper presents a simple technique to create CTS mode constraints to cater to all the various timing modes. The techniques involves, looping through the clock and generated clock definitions for all the timing modes and creating a bare minimum set of clock tree definitions that are sufficient to constrain the whole design and also make the clock tree view of the SoC simple for the CTS engine to enable good QoR. Generated clock constraints at combinational logic are all ignored to reduce the complexity of the clock constraints. The technique also derives the balancing constraints for the design that can be used during CTS.

### 2.2 Modeling Clock Trees Early in the Flow

Often, the buffers and routes that are created during CTS increase the utilization of the design to level that starts impacting placement convergence and routability. Such issues could lead to increased placement optimization iterations thus affecting the time to market of the device. It is thus essential to model the CTS overhead upfront during placement optimization.

In early CTS, clock network is synthesized soon after an initial placement is completed. This early CTS comprehends clock buffers and route resources for clock nets with necessary non default routing rules. The clock tree thus synthesized can also be effectively used to derive realistic clock gate latencies that can be used during placement optimization. The design is constrained to be in ideal clock mode for the rest of the placement optimization flow. The early clock

tree is then purged and the clock network is resynthesized after placement optimization.

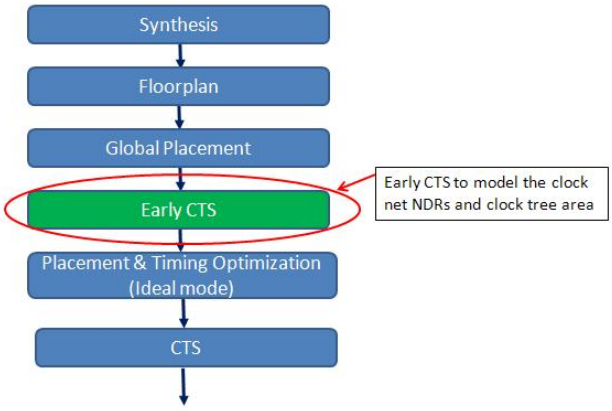


**Fig.1 Early CTS Flow**

## 2.3 Clock gate optimization

Gating clocks has been a widely adopted technique for reducing dynamic power. The clock gating strategy employed has a huge bearing on the clock tree synthesis quality along with the impact to leakage and dynamic power. The proposed clock gate optimization technique enables cloning and redistribution of the fanout among the existing equivalent clock gates. The technique is placement aware and hence reduces overall clock wire length and area. The technique involves employing the ”k-means clustering algorithm” to geographically partition the design’s registers. The paper highlights the utility of this technique by showcasing the clock tree synthesis quality of results improvement on a complex design.

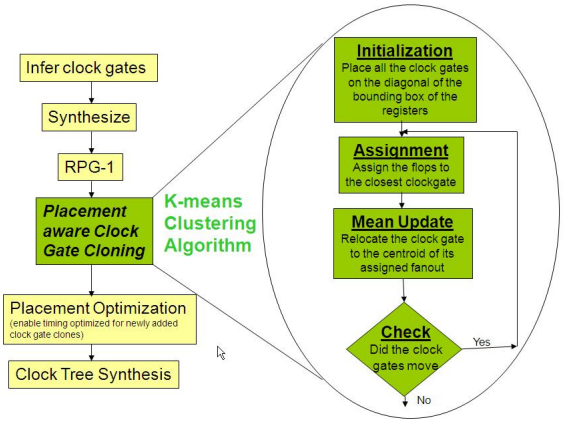


**Fig.2 Clock Gate Optimization Flow**

The related registers like the ones driven by common gates are now grouped to form a placement cluster. This enables reduction of clock leaf power and skew. There are various EDA solutions that implement this form of register clustering like regular placement, vector flops and floating clusters.

# 3 Clock Tree Synthesis Considerations

The choice repeaters selected for CTS is crucial for optimal quality of results. The various clock tree parameters like insertion delay, skew , area are all sensitive to the type of buffers used for clock network synthesis. The susceptibility of the delay of a repeater to process variations, switching current characteristics of the repeaters are some of the parameters that need to be considered.

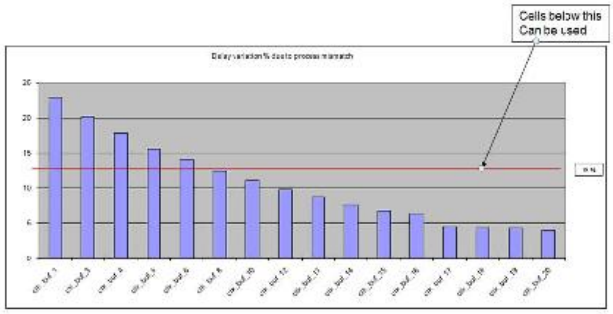


**Fig.3 Buffer delay variations due to process mismatch**

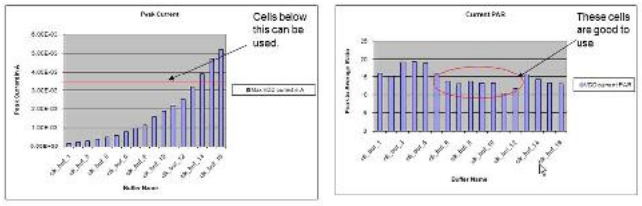


**Fig.4 Buffer switching current profiles - Average current and the peak to average ratio plots**

One of the primary considerations in building a robust clock tree is clock tree scalability. With the surge in the number of timing closure corners in deep submicron technologies, building a scalable clock tree is becoming increasingly important. A scalable clock tree has balanced insertion delays to all the clock sinks in the design. The interconnect component in the insertion delay will also need remain uniform across all sinks. This will ensure that the clock cells and interconnect will scale uniformly across timing corners (with variations in process, voltage and temperature). Fig.5 shows how a shorter insertion delay path is detoured to maintain balanced interconnect delay with the longest insertion delay path. Also, the choice of metal layers to route the clock tree nets is important to ensure scalability.

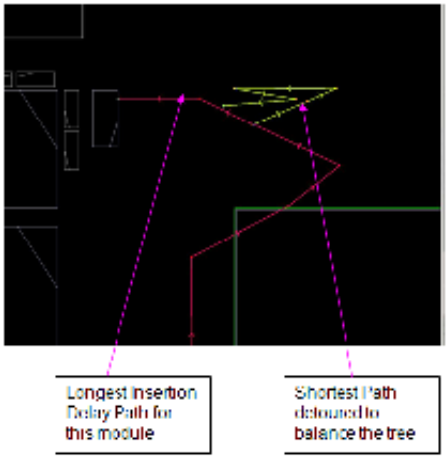


**Fig.5 Detouring clock nets to create balanced clock routing.**

## 4 Summary

The above aspects of clock tree planning and synthesis help in overcoming some of the limitations in current day EDA solutions in handling complex clock tree structures.