

Authors (Representing Synopsys Inc.,) –

- (1) Anish Keshava (Sr Engineer, ASIC Digital Design)
- (2) Aakarshak Nandwani (Engineer, ASIC Digital Design)
- (3) Nitesh M S (Sr Supervisor, ASIC Digital Design)
- (4) Satyanarayana Prasad Patnala (Sr Manager, ASIC Digital Design)
- (5) Sridharr S (Staff Engineer, ASIC Digital Design)

# Design Constraint Strategy For Dealing With Cascaded Clock MUX Structures

Providing an accurate description of cascaded clock MUX structures to static tools for improving QoR, Runtime and Turn-Around Time



**SYNOPSYS**



# Agenda

- Motivation
- Introduction
- Main Idea
- Implementation
- Evidence
- Summary

# Motivation

- Complex digital designs consist of Clock Multiplexers (clock MUXes in short), which are usually a blind spot for the static tools such as Synthesis, Static Timing Analysis (STA in short) and Clock Domain Crossing (CDC in short) Verification tools.
- There can be hiccups when it comes to clock propagation in such static tools, especially because clock multiplexers can be implemented in the design using logic that does not resemble an outright multiplexer. The motivation is to overcome the problem of clock MUXes in static tools in two parts as follows.
  - To improve QoR in Synthesis/STA tools:** Due to multiple clocks propagating through clock MUXes, post synthesis, STA tools can report a timing path (shown in Figure 1) which uses different clocks on the source and destination flops, which may not be of value to verification, this needs to be suppressed, which can also indirectly improve the runtime of Synthesis+STA.
  - To improve Efficiency & Run time in Clock Domain Crossing Verification tools:** The method of tying off the select line of a clock MUX (case analysis mode) is used for handling clock MUXes in CDC tools, and this technique has been known to be incapable of verifying all combinations of clocks through clock MUX structures thereby leading to incomplete CDC Verification, along with increased run times, added from each iteration pertaining to a certain analysis mode.

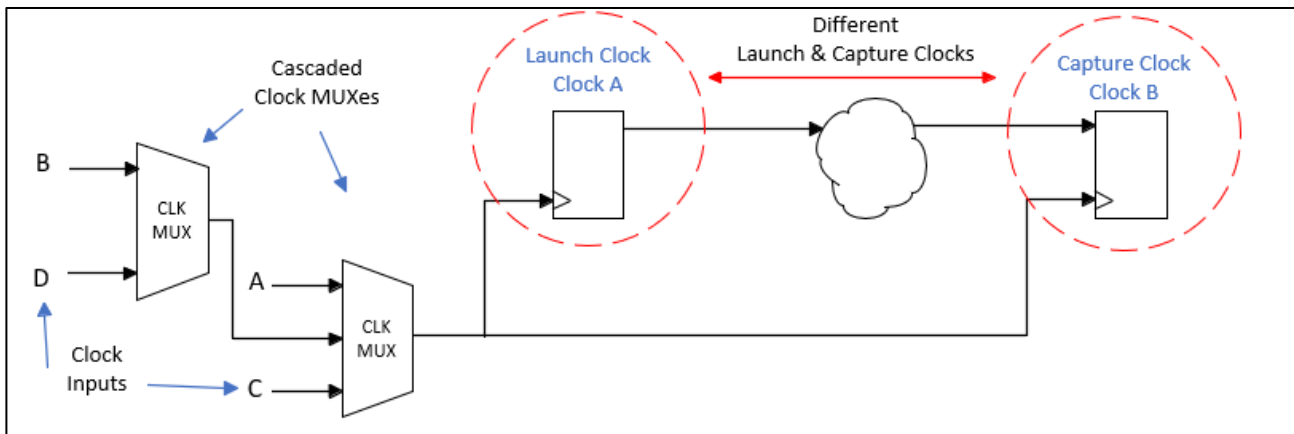


Figure 1: Without proposed methodology, tool times clock MUX path inaccurately with different clocks

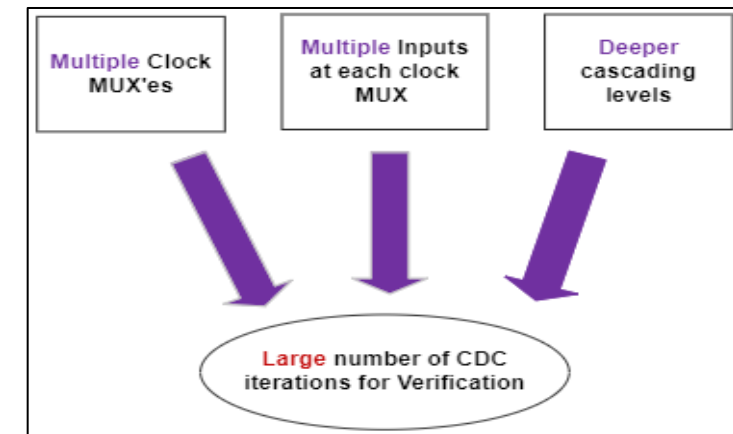


Figure 2: Sources leading to Large CDC Verification Iterations

The static verification process for clock MUXes can be optimized by using relevant synthesis design constraints (SDCs in short), using the proposed methodology. First, the idea is explained in high level, and the implementation is revealed in detail using a generic case study.

# Introduction - Background of Clock Multiplexer

- A clock MUX allows switching the clock line between different clock inputs, as shown in Figure 3
  - The switch time of a clock MUX is defined as the time elapsed between the previously chosen clock's negative edge to the currently selected clock's positive edge, as shown for a typical clock MUX, in Figure 3 (Assuming 50% duty cycle for all clocks)

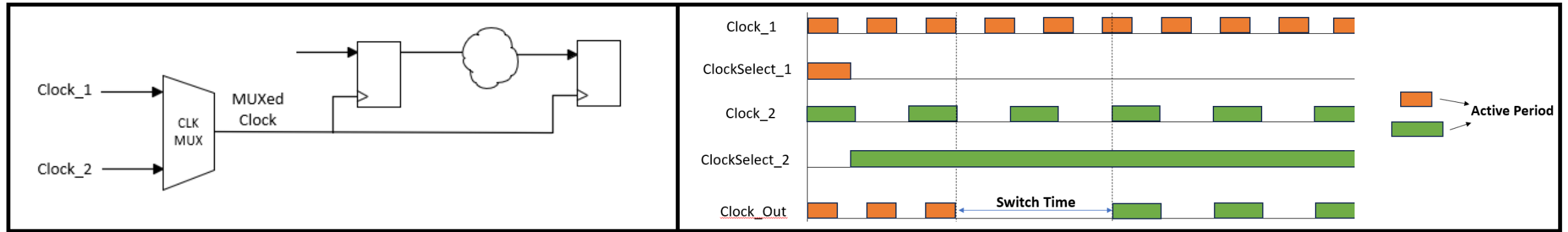


Figure 3: Typical Clock MUX & Switch Time in Clock MUX

- If the switch time is greater than “**half cycle of the fastest clock input to the clock MUX**”, then the CDC and timing verification of source and destination flops being timed by different clocks routed via the same clock MUX is redundant as the source data is expected to be stable within the switch time (exception – multicycle paths)
- Therefore, with sufficient switch time of the clock MUX (as in the case of glitch free clock MUXes), in principle - static CDC and timing verification can be reduced to the same clock reaching source and destination flop pairs at a given time
- The static tools used for verification in this regard, must be able to understand that verification must be done with only one clock passing through a clock multiplexer, or through a cascaded structure of clock multiplexers, and reach the clock pin of the source and destination flops at a given time
- A strategic set of SDC constraints involving **generated clocks** and **logically exclusive** constraints help in enforcing this nature of clock MUXes to static tools, as explained
- The proposed technique is not recommended for clock MUXes whose switch times are less than “half cycle of the fastest clock input to the clock MUX”

# Main Idea

1. Two types of SDC constraints/commands, namely, “**Create\_Generated\_Clock**” and “**Set\_Logically\_Exclusive**” are used to achieve the desired result
2. The **Create\_Generated\_Clock** constraint creates a new virtual clock at a specific pin in the design, from which only the generated clock shall propagate, as indicated in the bold line labelled “To clock MUX” in Figure 4
  - A generated clock inherits waveform, frequency and other related information from its master clock
  - The SDC command used to create a generated clock is as shown in Figure 4
3. The **Set\_Logically\_Exclusive** constraint takes any N number of clock groups as input and sets their propagation to be mutually exclusive
  - Any clock propagating in one of the N groups shall allow clocks of its own group and prevent all other N minus one group clocks from propagating at that instant
  - The verification shall cover all possible combinations of clocks with this exclusivity constraint in place
  - These logically exclusive constraints are applied to generated clocks that are created at clock MUX hierarchies in the design as shown in Figure 5

Therefore, generated clocks along with logically exclusive constraints can mimic the clock MUX functionality in terms of clock propagation. The steps of implementation for any generic clock MUX structure in the design is explained in the subsequent slides.

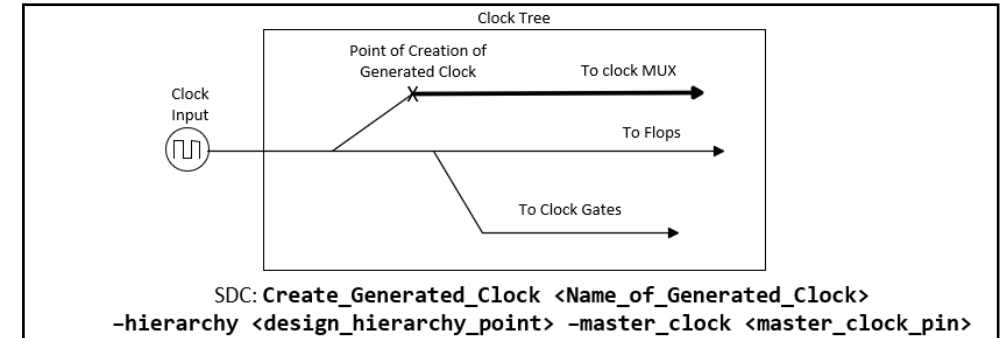


Figure 4: Creating a Generated Clock

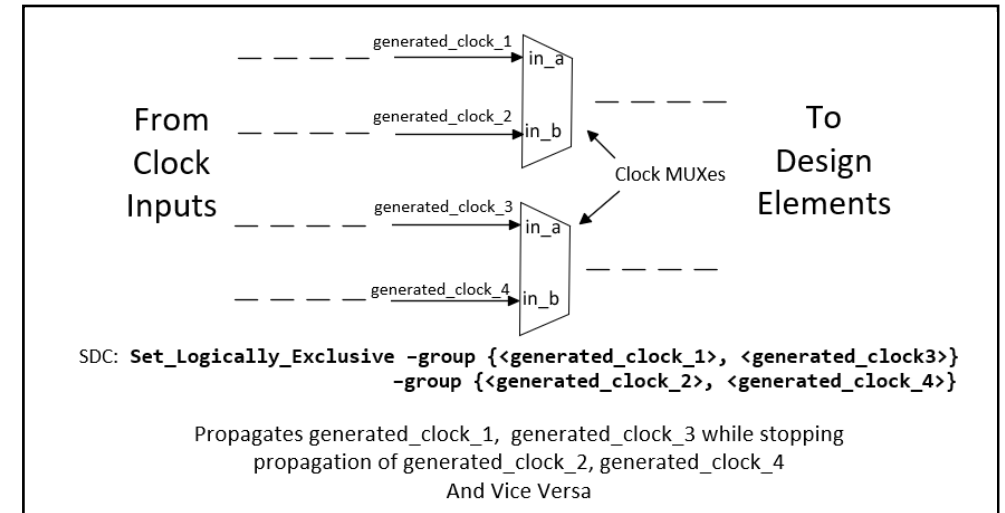


Figure 5: Setting a Logically Exclusive Constraint



# Implementation (1/2) Setting up the Generated Clocks

1. First, the clock MUX structure in the design must be identified
2. Consider a sample clock MUX structure as depicted in Figure 6, assume that the switch times are sufficiently large to consider only one clock passing through these clock MUXes as explained previously
3. Generated clocks are created at the inputs of each clock MUX as shown
4. This ensures that the branched-out parts of the master clock are unaffected by the logically exclusive constraint that is going to be applied on these generated clocks
5. Consider clock path of input clock “A”, three generated clocks are created, one at each clock MUX that A propagates to, the SDC commands to be applied are -
  1. SDC: Create\_Generated\_Clock A' -hierarchy M1.in\_a -master\_clock A
  2. The master clock for the  $(n + 1)^{th}$  stage will be the generated clock of the  $n^{th}$  stage, hence the remaining two generated clock constraints for A are as follows,
  3. -SDC: Create\_Generated\_Clock A'' -hierarchy M4.in\_a -master\_clock A'
  4. -SDC: Create\_Generated\_Clock A''' -hierarchy M5.in\_a -master\_clock A''
6. Similarly, generated clocks are created for the rest of the clock inputs (B to G)
7. The total number of generated clock constraints required would be summation of the number of cascading levels encountered by each clock input, in this case -

$$3 (A) + 3 (B) + 3 (C) + 3 (D) + 2 (E) + 2 (F) + 2 (G) = 18 \text{ Generated Clock Constraints}$$

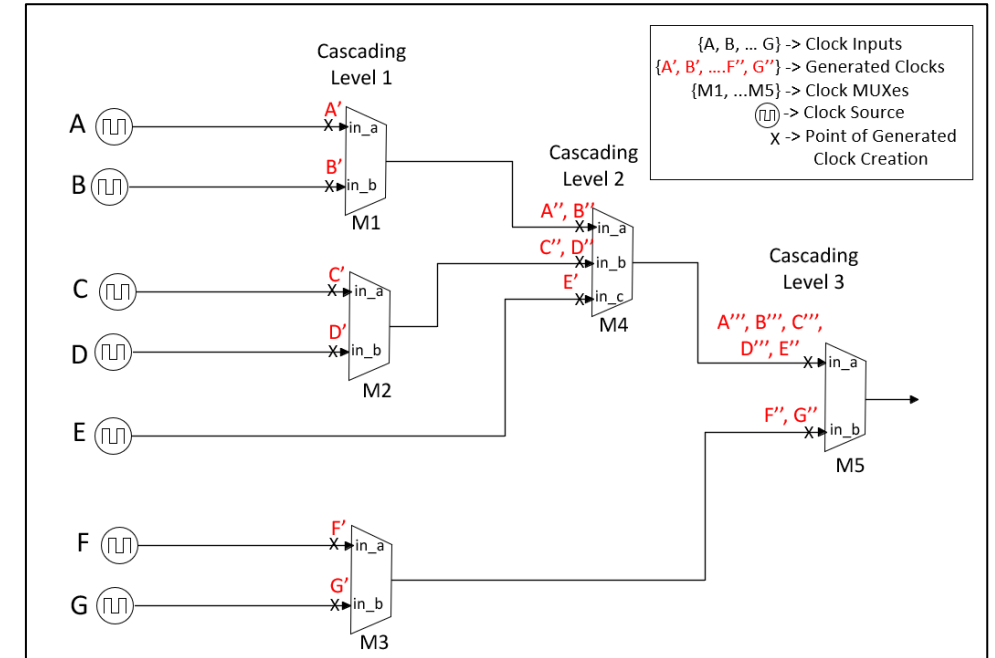


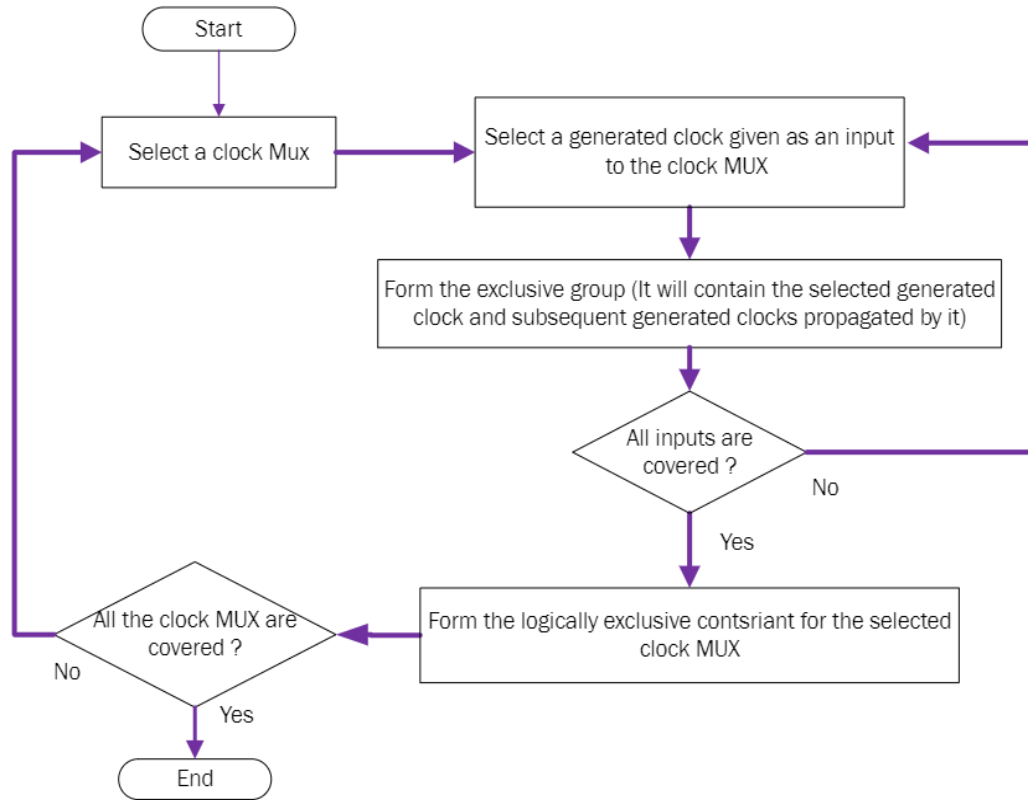
Figure 6: Sample Clock MUX Structure

Number of generated clock constraints,

$$N_G = \sum C_i$$

Where,  $C_i$  is the number of cascaded levels as seen from the clock input pin for the  $i$ 'th clock

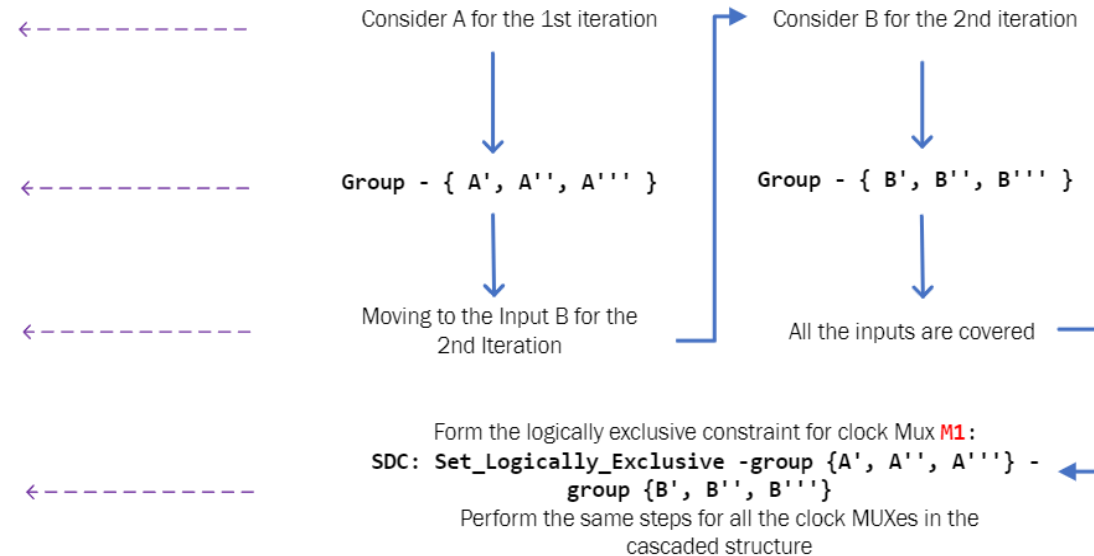
# Implementation (2/2) Adding Logically Exclusive Constraints



The logically exclusive constraints along with the generated clocks works as a translator and helps the tool to understand the relationships between the clocks getting propagated through the complex clock MUX structure. The design intent is preserved using this methodology.

At any given time, the clocks propagated will mimic the design intent of the clock multiplexer, and it shall not propagate illegal combinations of clocks

For eg., Consider clock Mux M1 in Figure 6



Remaining logically exclusive SDC constraints for structure in Figure 6:

**M2:** Set\_Logically\_Exclusive -group { C', C'', C''' } -group { D', D'', D''' }

**M3:** Set\_Logically\_Exclusive -group { F', F'' } -group { G', G''' }

**M4:** Set\_Logically\_Exclusive -group { A'', A''' } -group { B'', B''' } -group { C'', C''' } -group { D'', D''' } -group { E'', E''' }

**M5:** Set\_Logically\_Exclusive -group { A''' } -group { B''' } -group { C''' } -group { D''' } -group { E'' } -group { F'' } -group { G'' }

If the clock selects are common between different clock MUXes, then their respective logically exclusive constraints can be combined. Any additional clock MUX path added to the presented structure can still use the same approach for addition of generated clock and logically exclusive constraints.

# Evidence (1/2) Synthesis QoR Improvement

Two Synthesis+STA runs with and without the proposed technique were run on inhouse digital designs. The former reported redundant timing paths as shown in Figure 8 and the latter reported the same timing paths timed by only the fastest clock reaching the clock pin as shown in Figure 9.

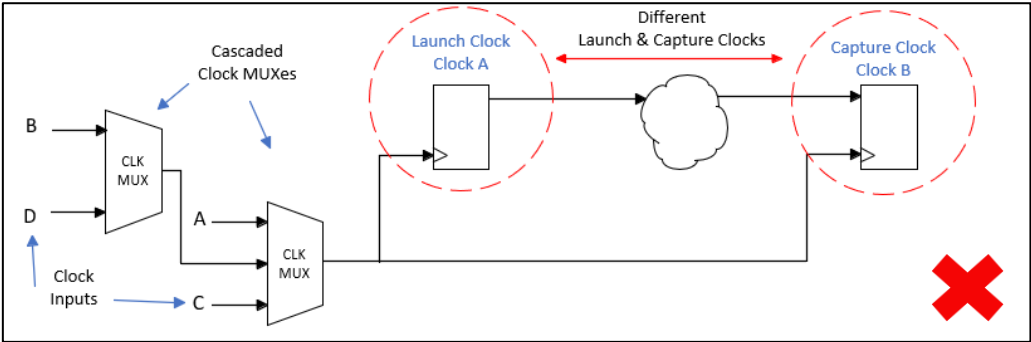


Figure 8: Without proposed technique, tool times data-path using MUXed clock with different clocks

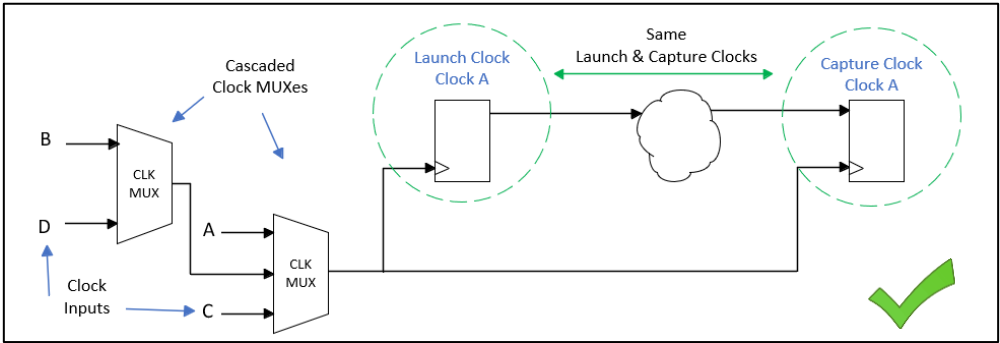


Figure 9: With proposed technique, tool times data-path using MUXed clock with same (fastest) clock A

Design	Number of clock MUXes	Levels of Cascading	Number of Generated Clocks	Number of Redundant Paths
X1	6	0	13	> 50,000
X2	10	1	15	> 150,000
Y1	9	3	42	> 40,000
Y2	18	3	95	> 60,000



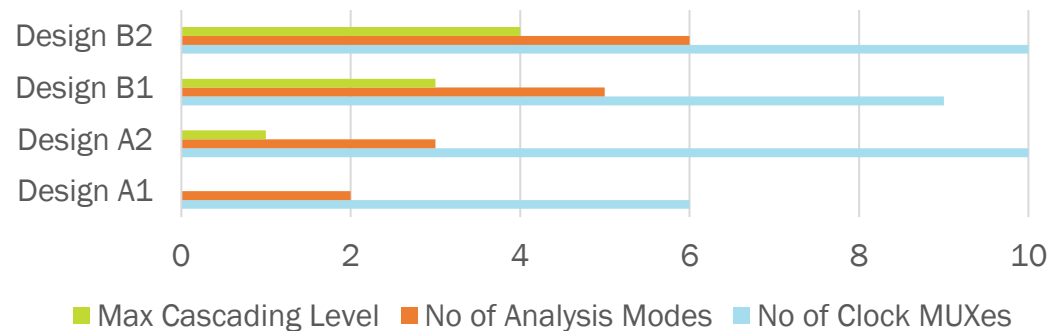
Can take up a large chunk of STA tool effort to cover all Timing Paths leading to high run times



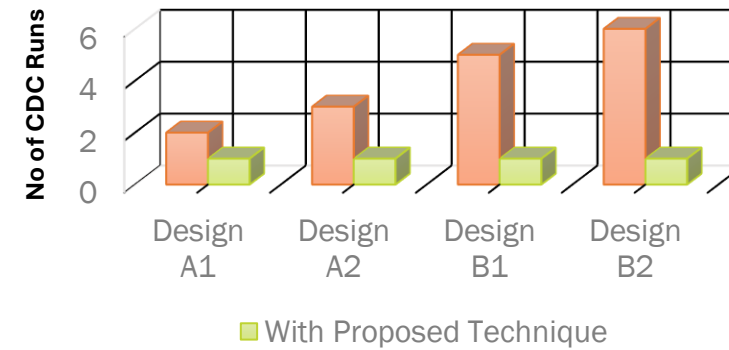
# Evidence (2/2) CDC Verification Run-Time Improvement

- Multiple in-house digital designs using the conventional case analysis modes for CDC verification previously, were run with the proposed technique using a static tool
- The relationship between all generated and master clocks were verified through the clock relationship matrix
- There were no setup violations reported with respect to clock (overlap/overwrite), thus establishing that the structure of the clock MUX is understood correctly by the tool
- The complete CDC verification of the design was done in a single run of the tool, since all possible combinations of clocks were able to propagate through clock MUXes
- The repetition of CDC violations across analysis modes was also avoided, thereby reducing the time required for debug, fix and CDC signoff

No of Clock MUXes Vs Max Cascading Level Vs No of Analysis Modes



N:1 Reduction in No of CDC Runs Per Design



# Summary

- The technique proposed was implemented on in-house designs containing glitch free clock multiplexers
- The results show that the technique can help prevent redundant timing paths from being reported after Synthesis+STA, especially in designs involving cascaded/complex clock MUX structures, thereby improving QoR of Synthesis/STA verification
- The drawback of case analysis technique such as its potential to mask CDC crossings due to its limited coverage of the design with respect to clock propagation is avoided with the proposed technique
- In turn, The Run-Time, debug time and overall time required for CDC sign-off was considerably reduced

Dependency of the applicability of this technique on various design aspects

- This technique is especially useful when design consists of deeply cascaded clock MUX paths, and can be applied to any design, irrespective of its clock MUX structure
- The number of constraints required in this technique is directly proportional to the number of clock MUXes in the design, number of clock inputs to clock MUXes, and the number of cascading levels
- This technique is applicable to any combination of clocks reaching the clock MUXes, any number of clock inputs and irrespective of the clock frequencies
- Designs that consist of clock MUXes with very small switch times have to be carefully reviewed before application of the methodology proposed in this presentation

Further Scope : The automation of this complete constraint addition process can be done, provided the information of the clock Multiplexers in the design are presented to or inferred by the tool in a suitable manner.

**SAY NO  
TO**





SHAPING THE NEXT GENERATION OF ELECTRONICS

**JUNE 23-27, 2024**

MOSCONE WEST CENTER  
SAN FRANCISCO, CA, USA



# Thank you

Email: [Anish.Keshava@synopsys.com](mailto:Anish.Keshava@synopsys.com)

LinkedIn: <https://www.linkedin.com/in/anish-keshava-5b699417a>