SYNOPSYS®

CONVERGENCE EXPERIMENT

CLOSED LOOP DATA-SENSE QUALIFIER-BASED BUS SYNCHRONIZER TO OVERCOME CONVERGENCE VIOLATIONS IN CLOCK DOMAIN CROSSINGS

Authors:

Bhargav S (ASIC Design Engineer I, Synopsys India Pvt Ltd)

Satyanarayana Prasad Patnala (Senior Manager, Synopsys India Pvt Ltd)

Agenda

- Motivation
- Main Idea
- Evidence
- Use Cases
- Summary

Motivation

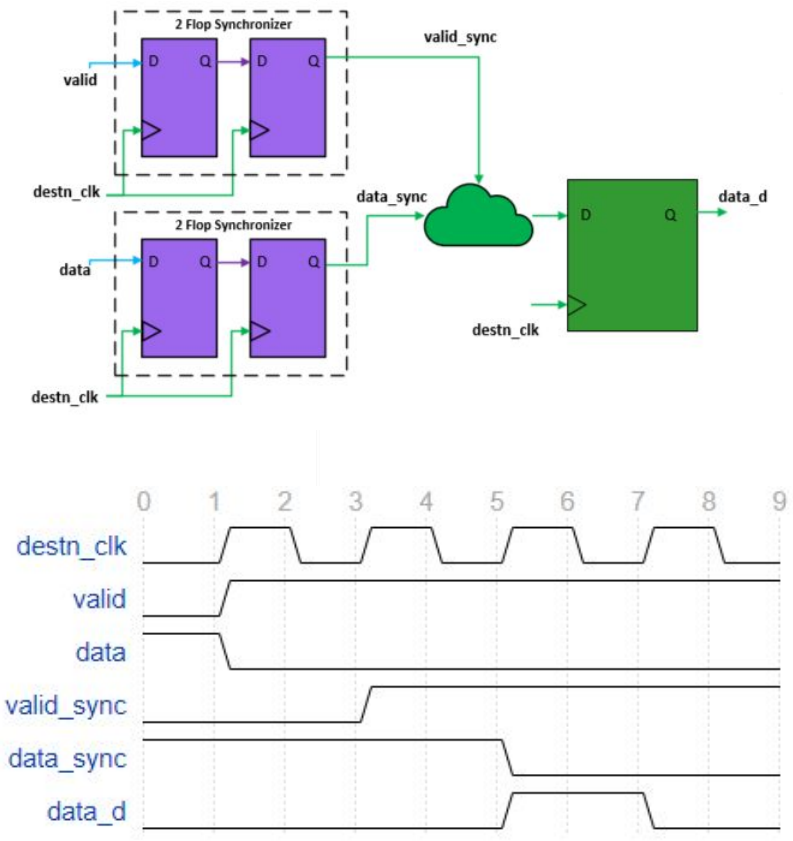

Figure 1: Example of a Convergence Violation

- Convergence is a CDC violation that occurs when different signals are synchronized to the same destination domain using individual synchronizers and they converge in the destination domain and lead to loss of data coherency and glitches being propagated.
- The conventional analysis flow for convergence violations is as follows:
  - Manually analyze the signals which are converging and determine if they are functionally related/unrelated.
  - If the converging signals are unrelated, add a constraint to the tool to not check convergence among the signals.
  - If the converging signals are related, change the RTL such that these signals converge in the source domain itself before synchronizing them or add logic to handle the sequence of occurrence of these signals in the destination domain.
- Limitations of current flow:
  - Adding constraints to not check convergence for each violation is a tedious process, difficult to maintain, and manual analysis is a risky process.
  - Added constraints might mask potential CDC issues in the future.
- Thus, we need a solution to fix convergence violations in the RTL such that there is no data coherency issue among the signals.

Main Idea (1/3)

- In any digital design, signals from different hierarchies may be synchronized to a common destination domain using individual synchronizers and then they converge in one functional block.
- Any small skew in the assertion of these signals in the source domain will lead to a loss of coherency among their synchronized versions and hence, wrong logical interpretations.
- Hence, we need to synchronize functionally related signals in such a way that the synchronized signals are always coherent with each other.

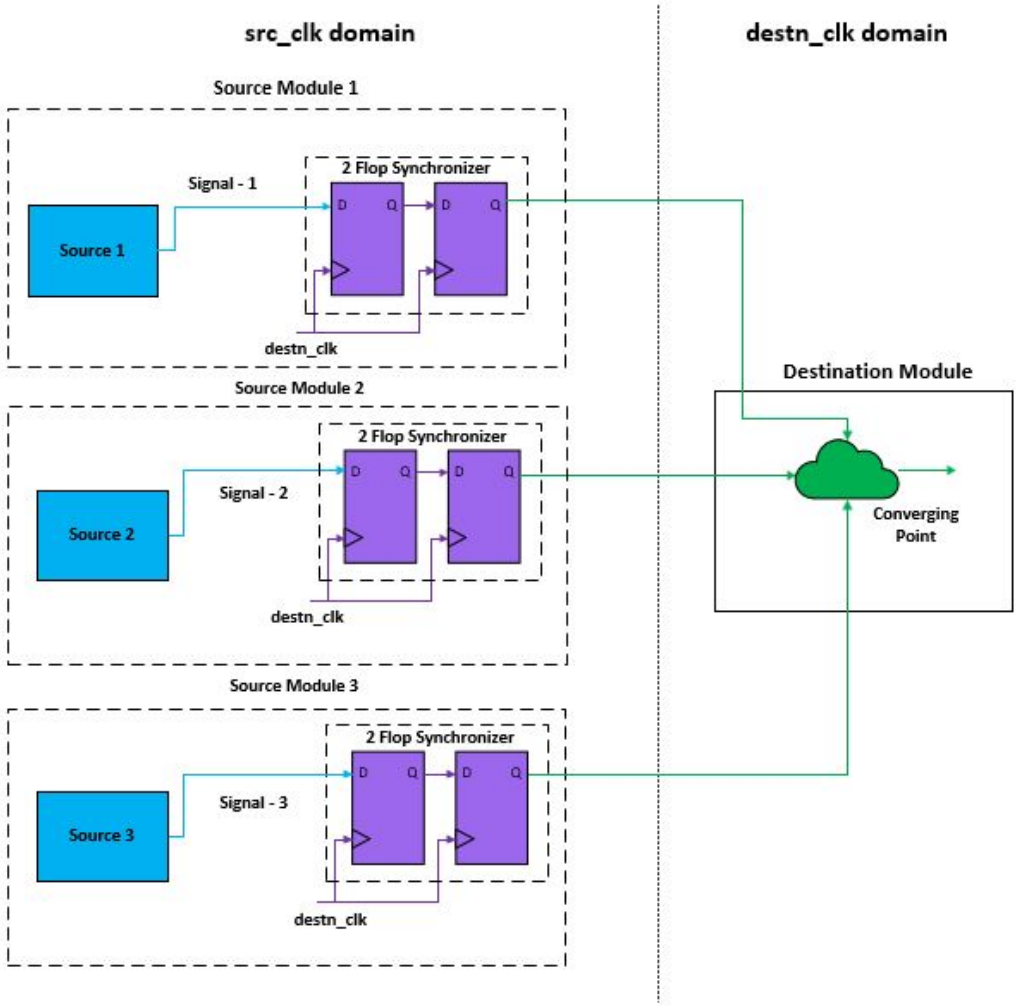

Figure 2: Conventional Synchronization Scheme

Main Idea (2/3)

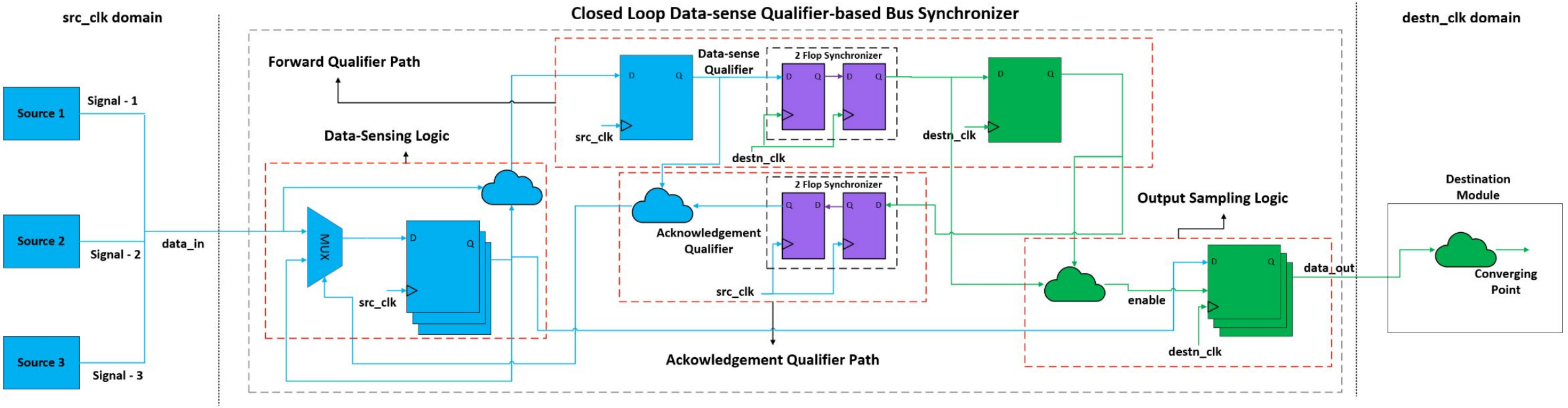

Figure - 3: Proposed Closed Loop Data sense Qualifier based Bus Synchronizer

Main Idea (3/3)

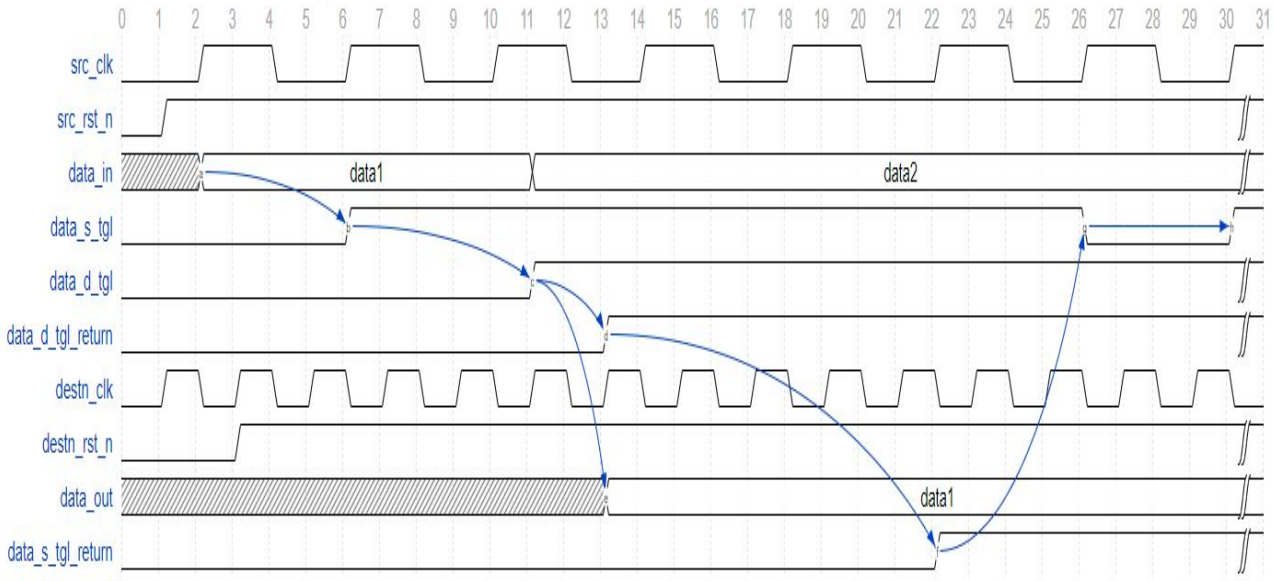

Figure - 4: Waveform explaining the sequence of actions (when the number of source and destination sync stages = 3 and mis-sampling is not enabled)

- Whenever there is a change in the bus input, the synchronizer generates a qualifier which is synchronized to the destination domain and used as an enable to sample the bus input in the destination domain.
- Once bus input is registered in the destination domain, we synchronize the qualifier back to the source domain where it acts as an acknowledgment to sample new incoming data and the cycle continues.
- Since the synchronizer uses a qualifier-based closed-loop mechanism to synchronize the bus input and does not synchronize the data input, we can ensure coherency is always met. Hence, convergence can be safely waived on the bus output bits.
- In order to ensure there is no data loss, the bus_input can only change once during the synchronization cycle i.e., until data_s_tgl == data_s_tgl_return, data_in can change only once.

Evidence

- All functionally related signals were grouped based on their source and destination clocks into clock groups. The proposed closed-loop bus synchronizer has been used to synchronize these related signals, the corresponding convergence constraints on them have been removed and the synchronizer has been proven to provide a convergence-free bus synchronization through simulation testing.

- The synchronizer has been tested with a randomized number of cycles of mis-sampling and has been proven to provide a convergence-free bus synchronization.

- Since the synchronizer uses a qualifier method to sample the source domain data in the destination domain, the synchronizer always ensures there is coherency among the output bits.

- Since the input is sampled only when there is a change in the data input, we do not add any additional dynamic power consumption even in the case of synchronization of quasi-static signals.

Table -1 : Comparison between various Synchronization Schemes

(For a 64-bit bus, when the number of source and destination sync stages = 3 and mis-sampling is not enabled)

Specification	Multi-Flop Synchronizer	Recirculation mux-based Bus Synchronizer	Dual-Handshake qualifier-based Synchronizer without Data-sensing	Proposed: Data-sense qualifier-based Bus Synchronizer
No of Flops required to synchronize 64 bits	192	68	136	136
Delay introduced on each bit	~3 destn_clk	~4 destn_clk	~4 src_clk + ~4 destn_clk	~4 src_clk + ~4 destn_clk
Merits	• Provides an open-loop synchronization scheme • Can be used when source clock is not available/not known.	• Supports open-loop synchronization of a qualified data path. • Can be used when source clock is not available/not known..	• Provides a closed-loop solution without the need for an external qualifier. • Ensures data coherency.	• Provides a closed-loop solution without the need for an external qualifier. • Ensures data coherency. • No additional dynamic power consumption.
Demerits	• Open-loop solution can lead to data loss. • Can cause loss of coherency.	• Open-loop solution can lead to data loss. • Can cause loss of coherency. • External qualifier is required.	• Additional dynamic power consumption since the synchronization loop is unconditional. • Delay introduced is more than that of an open-loop solution. • Cannot be used when source clock is not available/not known.	• Delay introduced is more than that of an open-loop solution. • Cannot be used when source clock is not available/not known.

Use Cases

The proposed synchronizer has been used in two specific use cases:

- **Register Programming**
  - Register programming is a part of every IP design. Most of the convergence violations (~70%) that came up in our IP involved programming registers.
  - Since programming is carried out initially and then the registers are quasi-static in nature, all the registers and signals converging with them can be grouped into clock groups and one bus synchronizer can be used for each clock group. This will ensure coherency among the functionally related converging signals.

- **Precision Counters**
  - In fast-running counters, although intermediate values can be missed, we need to ensure that whenever the timer value is sampled, the sampled value is accurate.
  - In this scenario, we can use the proposed bus synchronizer since it ensures coherency among the counter bits whenever it is sampled in the destination domain.

Apart from the above-mentioned use cases, the proposed synchronizer can be used in any non-timing critical data path where coherency is a requirement.

Summary

- Convergence is a commonly encountered CDC violation that can lead to data coherency issues if the convergence sources are functionally related to each other**.**

- **Problem: Addition of constraints to overcome convergence violations**
  - Can mask potential CDC violations.
  - Is difficult to update/handle in complex IPs.

- **Proposed Solution: Closed Loop Data-sense Qualifier-based Bus Synchronizer**
  - **Advantages:**
    - **Provides a closed-loop convergence-free solution that ensures data coherency among the output bits despite the differences in synchronization delay/skew among the multiple bits.**
    - **Removes the need to have convergence constraints.**
  - **Limitations:**
    - **The proposed bus synchronizer can only be used in non-timing critical paths which can support the delay caused due to the handshake mechanism and the input can only change once during a synchronization cycle.**

Thank you

Email: sbhargav@synopsys.com

LinkedIn: https://www.linkedin.com/in/bhargav-s-30b3b9195