CDC Techniques Without Synchronizers: Holistic Approach for High Quality Low Power Design

Sangeeta Raste
Circuit Technology RTL
Advanced Micro Devices, Inc.
Santa Clara, California, USA
Sangeeta.Raste@amd.com

Preethi Venkateshan
Circuit Technology RTL
Advanced Micro Devices, Inc.
Santa Clara, California, USA
Preethi.Venkateshan@amd.com

Grant Simmons
Circuit Technology RTL
Advanced Micro Devices, Inc.
Boxborough, Massachusetts, USA
Grant.Simmons@amd.com

ABSTRACT

Achieving a robust, high-quality low power design in a multiple clock domain design with effective management of clock domain crossing (CDC), is a critical milestone in the ASIC design process. In traditional CDC methodologies, the use of multi-flop synchronizers is standard for safe signal transmission across different clock domains. However, in low power designs, this approach introduces additional logic that negatively impacts power, performance, and area (PPA) - key metrics for SoC efficiency. Addressing CDC challenges in a low power environment requires innovative alternatives which maintain reliability while offering better PPA. This paper explores design techniques that address CDC violations by minimizing additional logic, enforcing architectural timing requirements, and using delay buffers with corresponding maximum delay timing constraints to ensure safe signal transfer. These design techniques include rearranging logic to remediate violations and reduce hardware overhead. The proposed methodologies make use of assertion-based verification to validate design architecture and CDC constraints. RTL Checkers are authored to match design specifications and ensure that the underlying assumptions and delay buffer constraints are continuously enforced in simulation regressions.

1 Introduction

Advancements in silicon design are driven by the demand for higher performance and lower power consumption. These improvements, however, introduce challenges in achieving comprehensive verification across multiple clock domains, particularly in mitigating clock domain crossing (CDC) violations. When dealing with asynchronous, free-running source and destination clock transfers, it's crucial to implement proper synchronization logic to avoid issues such as metastability, setup/hold violations, data loss or corruption, and spurious glitches, which can lead to misinterpretation of data. Ensuring safe transfer of signals between different clock domains is essential.

To achieve this, designers typically use synchronizers, which are circuits designed to resolve metastable nodes to a stable value before being sampled by the destination flop. Multi-flop synchronizers are often employed for clock domain crossing, as they help mitigate the risks associated with asynchronous signal transfer. However, in designs with multiple clock domains and crossings, adding numerous synchronizers can increase logic and register count which impacts power, performance, and area (PPA).

In cases where architectural relationships between source and destination clocks are well defined, delay buffers can be used to ensure robust data transfer. These buffers are applicable when data signals in the source clock (SrcClk) domain are controlled by synchronized control signals and have a well-defined architectural relationship with the destination clock (DestClk) domain. While synchronizers are still necessary, delay buffers provide a viable alternative in specific cases, as detailed in the examples within the paper. Figure 1 shows how a delay buffer is used to replace the traditional synchronizer under the scenario where A is slow data path signal with respect to synchronized control signal.

Please note that the CDC analysis tool doesn't recognize the purpose of the delay buffer. However, if you configure the tool to treat these buffers as custom synchronizers, it can resolve CDC violations.

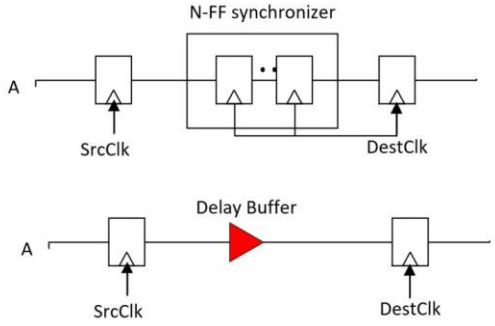

Figure 1

2 Approach I – Usage of Delay Buffer

The usage of a delay buffer cell to mitigate a "single-bit signal with no synchronizer" violation. This violation occurs when a signal transitions between different clock domains without adequate synchronization mechanisms leading to possible metastability in the design.

2.1 Example

As shown in Figure 2, there are multiple signals from SrcClk domain fanning into a combination logic expression with the end point captured in a register in DestClk domain. These fan-in signals are driven by a synchronized control signal. Figure 3, shows a delay buffer cell added to the path from SrcClk to DestClk. This buffer cell has a maximum delay parameter, to ensure safe signal transfer from SrcClk to DestClk domain; interlocked with the physical design team. For the duration of the maximum delay provided to the cell, the design injects X during logic simulation on the path to ensure that downstream logic will never capture an X during the active edge of the destination domain as shown in Figure 4

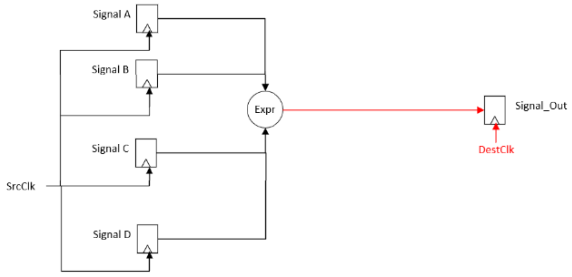

Figure 2

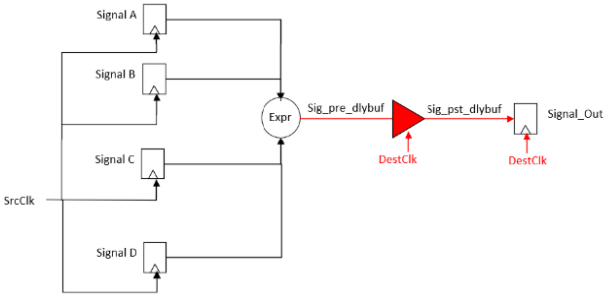

Figure 3

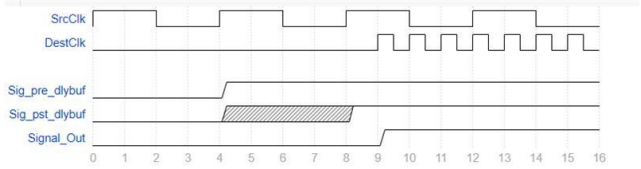

Figure 4

The robustness of this approach is enhanced by employing other methodologies such as assertion-based verification of buffer cells to ensure the clock on the destination flop is properly gated for the duration of the X injection. The duration of the modelled delay is then interlocked with the physical design team to ensure that the physical delay of the path does not exceed the modelled delay.

3 Approach II – Logic Rearrangement

Another method to address critical clock domain crossing (CDC) violations is to rearrange logic with minimal hardware overhead. One such critical violation described in this paper is a “combinational logic before synchronizer”. The presence of combinational logic within a synchronization path poses a substantial challenge for synchronized signals. This issue arises because the data utilized to establish an acceptable failure rate for synchronization paths is predicated on the assumption that the synchronizer input transitions with a frequency much lower than the capturing clock. However, the introduction of combinational logic into the path invalidates this assumption due to the potential for glitch propagation.

3.1 Example

In this example, the tool reports a CDC violation, informing the user about combinational logic before a synchronizer. In the central configuration, the tool identifies a delay buffer cell as a custom synchronizer. Consequently, placing a buffer cell in the path after combinational logic causes a violation highlighting the risk of glitches in the path.

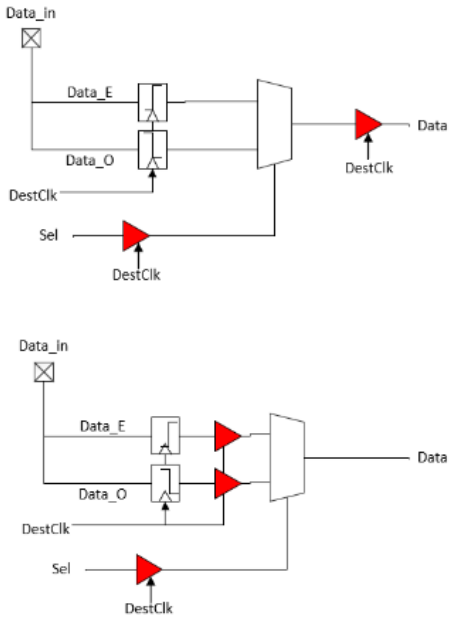

Figure 5

To resolve this violation, the logic is rearranged to move the delay buffer cell before the combinational logic. In doing so, all inputs to the combinational logic are now independently randomized, exposing potential glitch risks through simulation. Figure 5 shows how the buffer cell is moved from the output of the multiplexer to the two input ports of the multiplexer. Although this approach increases the buffer cell count from one to two, it is less compared to standard approaches like adding multi-flop synchronizers.

This approach resolves the "combinational logic before synchronizer" violation but brings up another violation reported by the clock domain crossing (CDC) tool – "reconvergence". A reconvergence violation reported by a clock domain crossing (CDC) tool occurs when signals that have been synchronized separately reconverge at a point in the design, potentially leading to data coherency issues.

3.2 Constraints

Reconvergence violations can arise from hazards in either the data or select lines, or both. To address this, it's crucial to architecturally ensure that the select line remains static when data transitions occur. When the select line is guaranteed to be stable, the multiplexer output exhibits mutually exclusive behavior, as shown in Figure 5. This behavior can be effectively communicated to the clock domain crossing (CDC) analysis tool with specific constraints, as outlined below

```
cdc signal -mutually_exclusive {Data_E}
   {Data_O} -module {data_sampler}
```

Another reconvergence issue still exists with the select line signal to the multiplexer. If the architecture guarantees that the select signal remains stable and does not change close to data transitions, a tool supported constraint can be added as shown below

```
cdc signal -stable {Sel} -module
        {data_sampler}
```

This provides additional information to the clock domain crossing (CDC) tool to analyze the path and effectively address reconvergence violations. As with any constraint used to describe architectural intent, the constrained assumption must be verified through simulation or formal methods. This is where we leverage the assertion methodology for verification purposes.

3.3 Assertions

Assertion methodology enables the verification of architectural design intent on a pre-silicon platform, ensuring the integrity and quality of the design. This applies for verification of CDC constraints as well. Custom checkers are coded in the RTL design to verify the mutual exclusivity of the signals input into the multiplexer and the stable behavior of the select signal connected to the multiplexer.

```
property mutually_excl_check;
  @(posedge DestClk);
     $stable(Data_E) | $stable(Data_O);
endproperty

property stable_check;
  @(posedge DestClk)
    $stable(Sel);
endproperty
```

The mutex assertions check that whenever a signal changes, the other signal is stable and vice versa proving their exclusivity. The stable assertion is built to verify that when the path is active, the signal is unchanged. These checkers are part of the pre-silicon regression and coverage analysis.

4 Conclusion

In conclusion, the logic rearrangement approach for resolving clock domain crossing (CDC) violations encompasses more than just altering logic. It involves logic modification, constraint updates, and thorough verification to ensure quality sign-off. This comprehensive approach addresses multiple aspects of the design, enabling a thorough verification of paths across multiple clock domains.

Not all cases can use the techniques in this paper, including where timing requirements are exceptionally stringent. But with careful design and thoughtful application, the approaches discussed offer substantial benefits in achieving reliable and efficient CDC analysis.

In summary, in low power design, it is critical to analyze the cost of every logic addition contributing to increasing power and area. By minimizing additional logic, leveraging delay buffer cells, and enforcing rigorous timing and verification strategies, these techniques achieve high quality CDC signoff without sacrificing power, performance, or area. This approach establishes a scalable framework for high-quality, low power design, ensuring both functional correctness and design efficiency.

5 References

- Clifford E. Cummings, "Clock Domain Crossing (CDC) Design & Verification Techniques Using SystemVerilog", presented at the Synopsys Users Group (SNUG) conference, Boston, MA, USA, 2008
- Srikanth Vijayaraghavan & Meyyappan Ramanathan, "Introduction to SVA," in A Practical Guide for SystemVerilog Assertions, 2005 Springer, pp 7-88
- Srikanth Vijayaraghavan & Meyyappan Ramanathan, "Checking The Checker," in A Practical Guide for SystemVerilog Assertions, 2005 Springer, pp 285-311