

Design and Implementation of a Dual Flip-Flop Synchronizer for Clock Domain Crossing with Metastability Analysis

eSim Research Migration Project

Diddekunta Nandini

Rajiv Gandhi University of Knowledge Technologies, Nuzvid

Abstract

The Design and Implementation of a Dual Flip-Flop Synchronizer for Clock Domain Crossing focuses on developing a reliable circuit for transferring a single-bit signal between two asynchronous clock domains. Clock Domain Crossing is an important issue in digital systems where signals generated by one clock domain are sampled by another clock domain. The design employs two flip-flops connected in cascade to safely synchronize the asynchronous signal with the destination clock. When the input signal changes close to the active edge of the destination clock, setup-time or hold-time violations may occur, resulting in metastability in the first flip-flop. The second flip-flop provides additional time for the metastable condition to settle before generating the synchronized output. This architecture reduces the probability of metastability propagating to the destination logic, making it suitable for reliable digital systems, FPGA designs, VLSI circuits, and communication interfaces. The proposed implementation provides a simple and effective solution for single-bit Clock Domain Crossing while maintaining low hardware complexity.

Working Principle

The operation of a Dual Flip-Flop Synchronizer is based on safely transferring an asynchronous signal from one clock domain to another while reducing the effect of metastability. The circuit follows these steps:

Asynchronous Input

- The input signal is generated from the source clock domain.
- The signal may change at any time with respect to the destination clock.

First Flip-Flop Operation

- The asynchronous input is applied to the first flip-flop.

- If the input changes close to the destination clock edge, metastability may occur.
- The first flip-flop provides the initial stage for sampling the asynchronous signal.

Second Flip-Flop Operation

- The output of the first flip-flop is connected to the input of the second flip-flop.
- The second flip-flop provides additional time for the metastability to settle.
- This reduces the probability of metastability propagating to the destination logic.

Synchronized Output

- The output of the second flip-flop is taken as the synchronized output.
- The synchronized signal can then be safely used by the destination clock domain.

Since the asynchronous signal passes through two flip-flop stages, the probability of metastability reaching the destination logic is greatly reduced. However, the dual flip-flop synchronizer introduces a small synchronization delay and is mainly suitable for single-bit Clock Domain Crossing signals.

0.1 Circuit Diagram

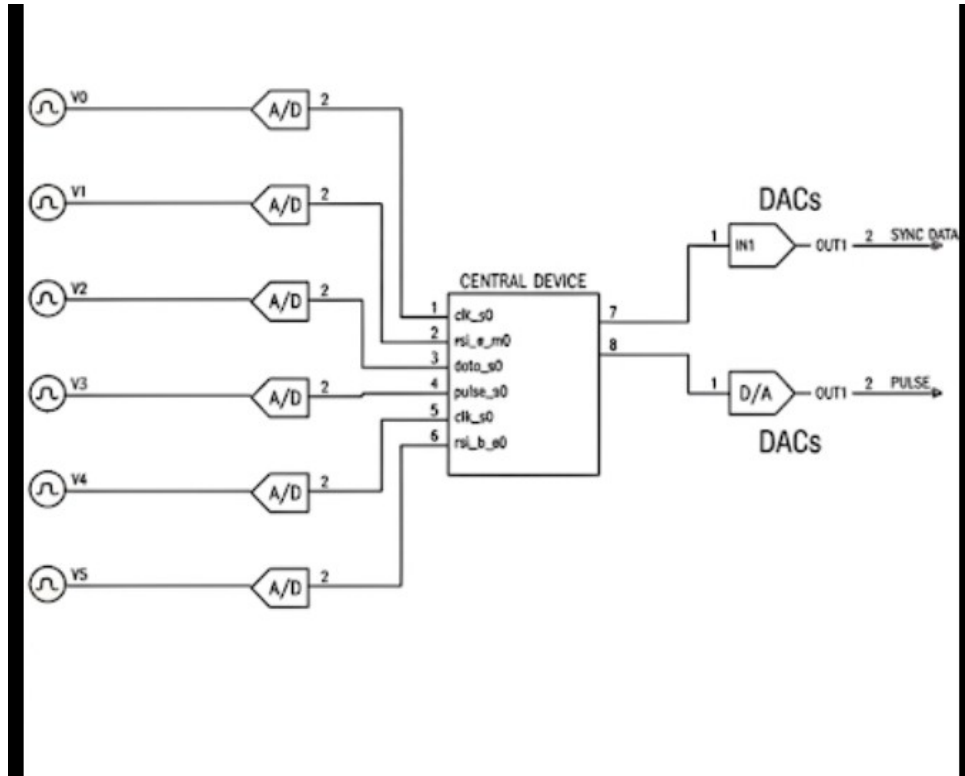


Figure 1: Dual Flip-Flop Synchronizer Circuit Diagram

0.2 Circuit Schematic in eSim

The proposed Dual Flip-Flop Synchronizer is implemented and simulated using eSim. The schematic consists of two flip-flop stages connected in cascade. The asynchronous input is applied to the first flip-flop, while the output of the first stage is connected to the input of the second flip-flop. Both flip-flops are controlled by the destination clock.

The eSim schematic is shown below.

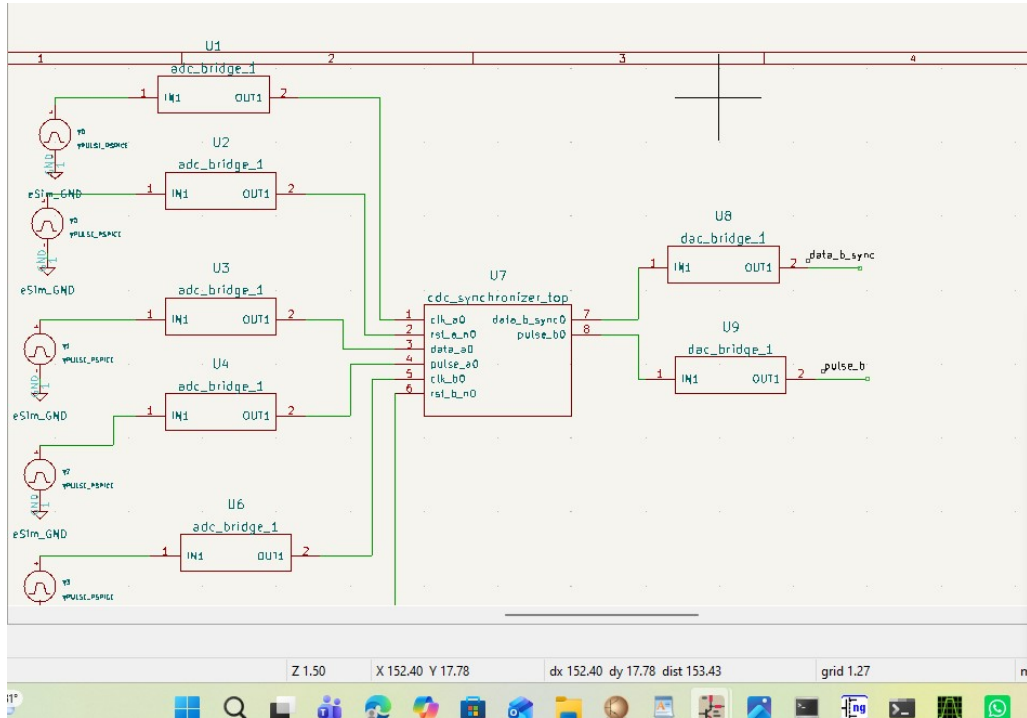


Figure 2: Dual Flip-Flop Synchronizer Circuit Schematic in eSim

The schematic shows the complete circuit implementation used for simulation. The first flip-flop samples the asynchronous input, and the second flip-flop generates the synchronized output. This implementation helps reduce the probability of metastability propagation in the destination clock domain.

0.3 Core Synchronizer Circuit

The core synchronizer circuit represents the main functional portion of the proposed Dual Flip-Flop Synchronizer. It shows the two flip-flop stages used for transferring the asynchronous signal into the destination clock domain.

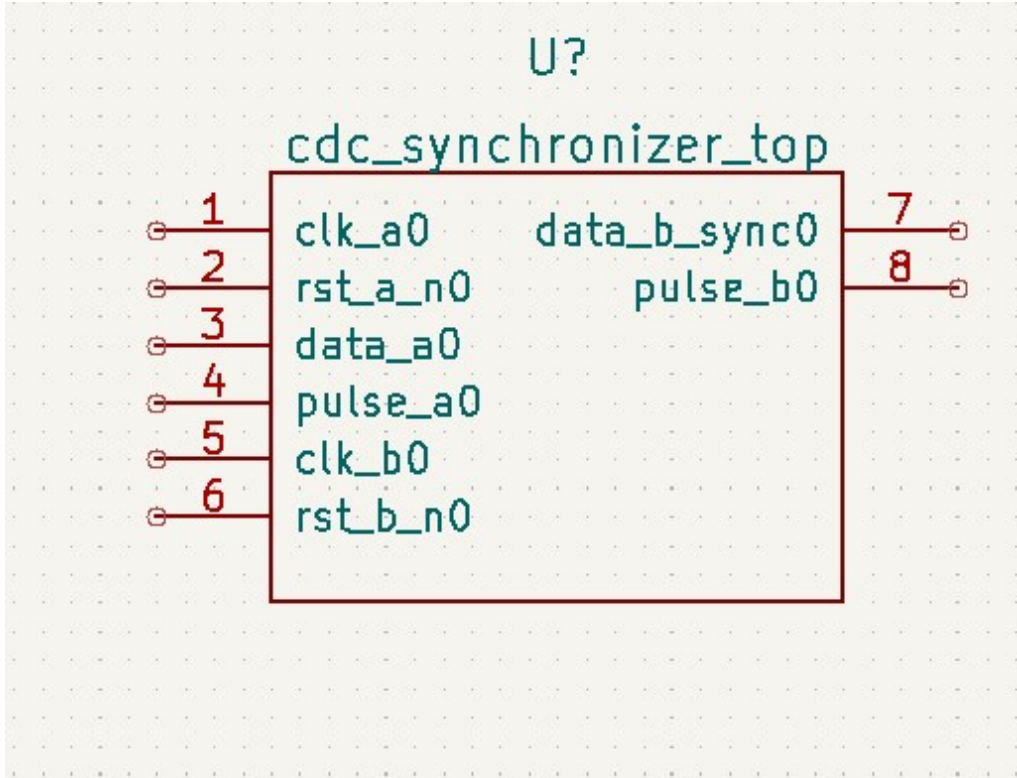


Figure 3: Core Synchronizer Circuit Diagram

The core synchronizer circuit internally consists of two flip-flops connected in series. The first flip-flop receives the asynchronous input and acts as the first synchronization stage. The second flip-flop receives the output of the first stage and produces the synchronized output. This arrangement provides additional time for metastability resolution and reduces the possibility of metastability propagating to the destination logic.

0.4 Voltage Analysis

The voltage waveform of the proposed Dual Flip-Flop Synchronizer is analyzed to observe the switching behavior of the circuit. The voltage variation with respect to time is shown below.

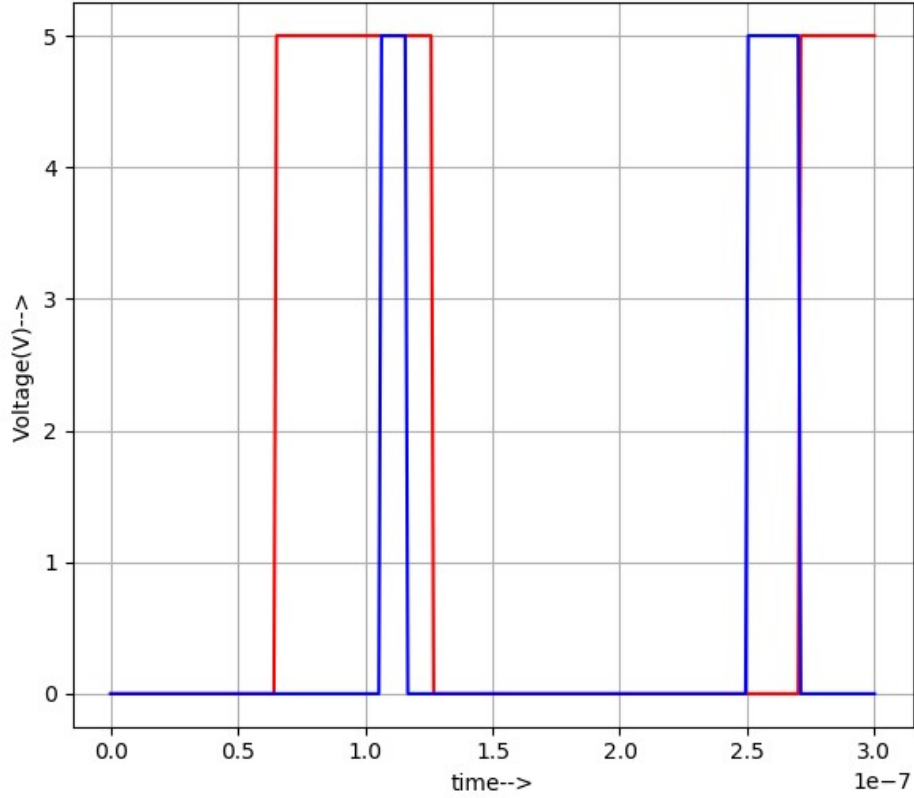


Figure 4: Voltage Analysis of the Dual Flip-Flop Synchronizer

- The waveform shows clean digital signal transitions between approximately 0V and 5V, confirming proper logic-level operation of the synchronizer circuit.
- The delayed transition of **pulse_b** with respect to **data_b_sync** demonstrates the expected synchronization delay and successful signal transfer through the flip-flop stages.

The voltage waveform demonstrates the switching behavior of the synchronizer circuit and confirms the transition between logic low and logic high voltage levels.

0.5 Current Analysis

The current waveform of the proposed Dual Flip-Flop Synchronizer is analyzed to observe the current behavior of the circuit during switching operations.

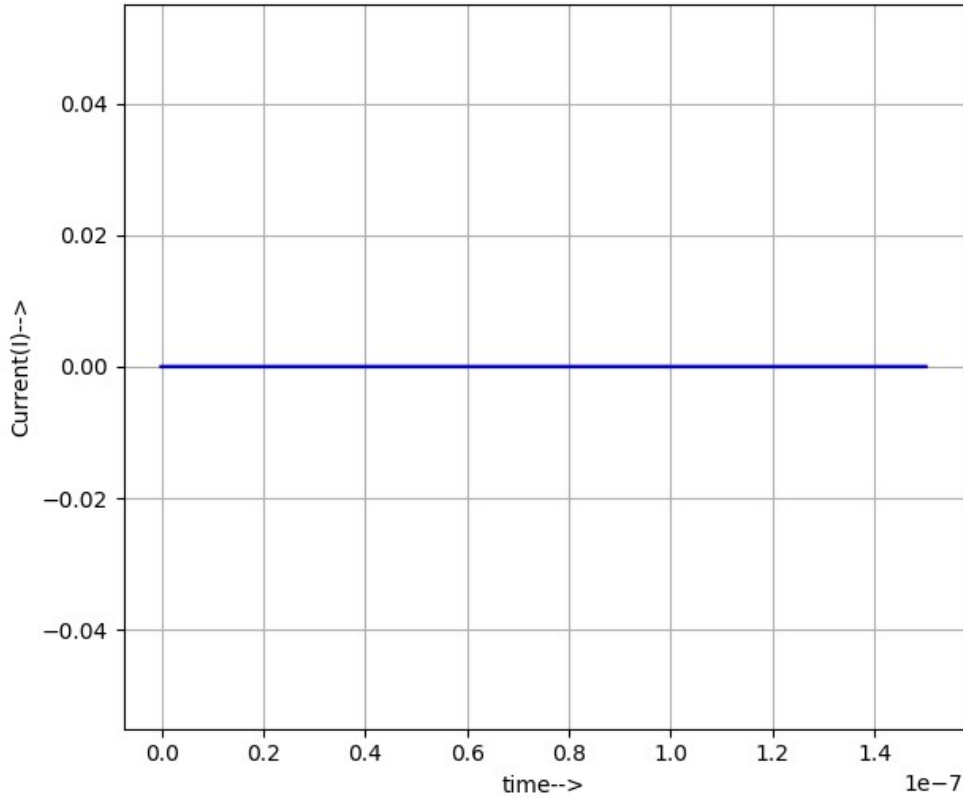


Figure 5: Current Analysis of the Dual Flip-Flop Synchronizer

- **Ideal digital behavior:** Digital blocks act like simple switches, so the tool treats them with zero internal leakage current.
- **Zero steady-state current:** When logic is steady at 0V or 5V, there is no continuous current flowing through.
- **No heavy load:** The synchronizer circuit uses digital gates, meaning it does not draw heavy power like a physical resistor.
- **Clear baseline:** Scaling keeps the current flat at the 0.00A baseline so it stays separate from the voltage graph.
- **Standard simulation result:** A flat zero line is the correct and expected output for digital blocks in eSim.

The current waveform demonstrates the expected low-power behavior of the digital synchronizer circuit under steady-state operating conditions.

0.6 Data Synchronization and Metastability Analysis

Stage 1: Level Synchronization Waveform

The `async_in` signal changes independently of `clk_b`, creating possible setup/hold violations at the sampling edge.

The `sync_stage_1` signal captures `async_in` first and is the stage most at risk of going metastable.

The `sync_out` (same signal as `data_b_sync` at the top level) samples `sync_stage_1` one `clk_b` cycle later, letting any instability settle.

The total latency from `async_in` to `data_b_sync` is 2 `clk_b` cycles.

The output waveform shows clean, glitch-free transitions, confirming successful synchronization.

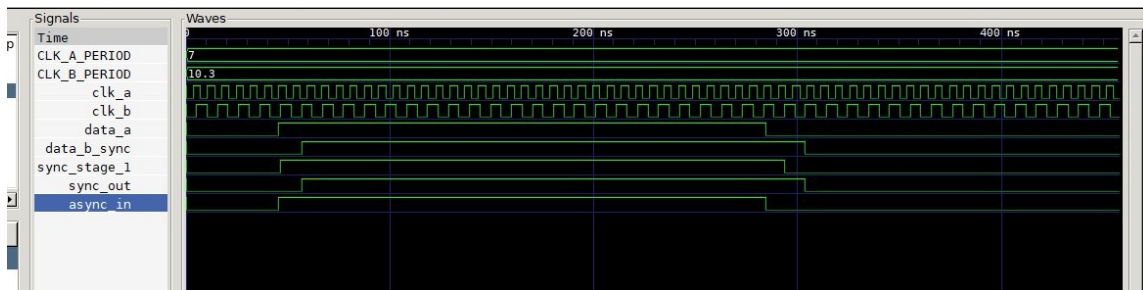


Figure 6: Level Synchronization Waveform

The waveform demonstrates the operation of the two-stage synchronizer and shows how the asynchronous input is sampled and transferred safely into the destination clock domain.

Stage 2: Pulse Synchronizer Waveform

The `pulse_a` signal generates short 1-cycle pulses in the source clock domain (`clk_a`).

The synchronizer converts each pulse into a toggle signal so it can safely cross into the `clk_b` domain without being missed.

The toggle bit is passed through the dual flip-flop stage, then edge-detected to regenerate a clean pulse (`pulse_b`) in the destination domain.

The `pulses_sent` counter increments from 0 to 5 in domain A, and `pulses_seen` increments from 0 to 5 in domain B, matching exactly.

No pulses were lost, duplicated, or distorted during the clock domain crossing.

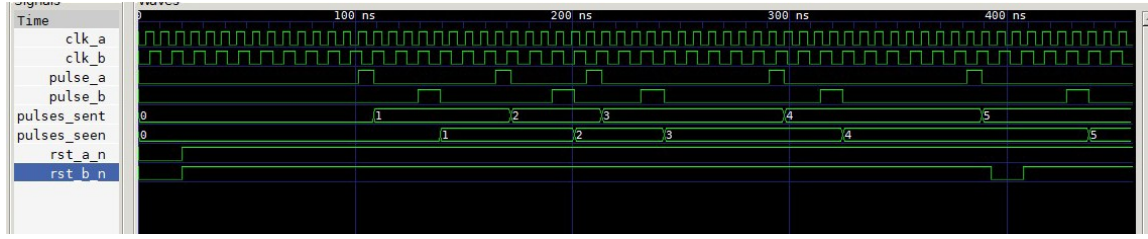


Figure 7: Pulse Synchronizer Waveform

The pulse synchronizer waveform confirms successful pulse transfer between the asynchronous clock domains and verifies the correct operation of the synchronization mechanism.

0.7 Overall Simulation Waveform

The overall GTKWave simulation verifies the complete dual flip-flop synchronizer top module. The waveform shows successful signal transfers across the asynchronous `clk_a` and `clk_b` clock domains.



Figure 8: Overall Simulation Waveform of the Dual Flip-Flop Synchronizer

The GTKWave simulation plot verifies the complete dual flip-flop synchronizer top module, showing successful signal transfers across asynchronous `clk_a` and `clk_b` clock domains. The matching pulse counters (`pulses_sent` = 5 and `pulses_seen` = 5) along with clean 2-stage synchronized outputs (`data_b_sync`) confirm zero data loss and reliable metastability mitigation.

The overall simulation results demonstrate that the proposed synchronizer successfully transfers both level/status signals and pulse signals between asynchronous clock domains while maintaining reliable synchronized outputs.

0.8 Conclusion

The Dual Flip-Flop Synchronizer for Clock Domain Crossing with metastability analysis was successfully designed and simulated to provide reliable transfer of signals between two asynchronous clock domains. The design uses two flip-flop stages to reduce the probability of metastability propagating into the destination logic.

The simulation results obtained from eSim and GTKWave verify the proper operation of the proposed synchronizer. The voltage and current waveforms demonstrate the switching and power behavior of the circuit, while the GTKWave results confirm correct synchronization between the asynchronous clock domains.

The level synchronization waveform shows that the asynchronous input is safely transferred through the two synchronization stages. The pulse synchronizer waveform further confirms successful pulse transfer, with the number of pulses sent and received matching from 0 to 5. The overall simulation waveform demonstrates correct operation of the complete synchronizer top module.

Thus, the proposed Dual Flip-Flop Synchronizer provides a simple and effective solution for single-bit Clock Domain Crossing. It significantly reduces the probability of metastability reaching the destination logic and improves the reliability of digital systems operating with asynchronous clock domains.

0.9 References

1. S. L. Harris and D. M. Harris, *Digital Design and Computer Architecture*, 2nd Edition, Morgan Kaufmann, 2012.
 2. R. Ginosar, “Fourteen Ways to Fool Your Synchronizer,” *Proceedings of the 9th International Symposium on Asynchronous Circuits and Systems*, pp. 89–96, 2003.
 3. L. Kleeman and A. Cantoni, “Metastable Behavior in Digital Systems,” *IEEE Design & Test of Computers*, Vol. 4, No. 6, pp. 4–19, 1987.
-

Software Tools and Versions

- **eSim Version:** 2.5
 - **KiCad Version:** 6.0.11
-