*International Journal of Computational and Experimental Science and ENgineering (IJCESEN)*

*Vol. 12-No.3 (2026) pp. 409-416*
*http://www.ijcesen.com*

***ISSN: 2149-9144***



**Research Article**

# Metastability in Digital Design: A Comprehensive Analysis of Clock Domain Crossings, Layout Techniques, and Timing Methodologies

**Sagar Mallik**

Independent Researcher, USA
* **Corresponding Author Email:** saga2r@gmail.com- **ORCID:** 0000-0002-5247-115X

**Article Info:**

**DOI:** 10.22399/ijcesen.5492
**Received :** 02 June 2026
**Revised :** 12 August 2025
**Accepted :** 16 August 2026

**Keywords**

Metastability
Clock Domain Crossing
Gray Code
Synchronizer
Resolution Time
Asynchronous FIFO

**Abstract:**

Metastability still remains one of the most challenging problems in modern digital designs, especially for high-speed and low-latency systems in which we have to deal with multiple clock domain crossings (CDCs). This article presents a methodology to tackle the metastability issue from the design, the layout, and the static timing analysis (STA). We derive the physical principles of setup and hold time in terms of data pulse width. We explain the operation of the very clever design of the reset synchronizer. We discuss systematic techniques for dealing with metastability in parallel data transfers using Gray coding. Our analysis shows what uncertainties metastability adds to digital designs and how, while metastability cannot be eliminated, it can be dealt with using proper synchronizing techniques, cell characterization, and layout skew matching. It briefly describes the concepts of resolution time and MTBF. Also, digs deeper into timing analysis of asynchronous FIFOs and the design of CDCs using classical theory and modern knowledge of setup/hold timing windows and metastability reduction techniques. In summary, the article provides both theoretical and practical results to the designer of asynchronous interfaces for use in System-on-Chip (SoC) designs.

## 1. Introduction

Clock domain crossings (CDC) are one of the trickier issues in digital system design. Modern SoCs (Systems on Chip) have, by necessity, multiple subsystems operating in different clock domains for power management, functional partitioning, hierarchical design abstraction, and multi-die communication. However, the fact that these are asynchronous interfaces means there is the possibility of metastability: if a flip-flop does not resolve to a logic level within a certain time, functional failure may result.There have been numerous works in the academic and industrial literature that have addressed the issue of metastability [1-3]. However, the interaction between different design techniques and layout limits and the underlying physics, as well as the behavior of the failure probability as a function of the timing margin, is ill-understood. This paper proposes a unified framework to address metastability in real designs.

Our contributions include (1) providing a perspective for setup and hold times based on data pulse width and identifying their physical cause; (2) understanding why reset synchronizer itself is safe when resets are toggled; (3) guidelines for systematic design of metastability-safe parallel data transfers using Gray coding and/or layout precision skew; (4) the first quantitative analysis of the tradeoffs between synchronizer depth, resolution time, and system MTBF; and (5) design guidelines for asynchronous FIFOs and other common CDC scenarios.

***Table 1:*** *Setup and Hold Time Physics [1-4]*

| Timing Parameter | Physical Basis | Latch Behavior | Design Impact |
|---|---|---|---|
| Setup Time | Minimum stable data window before clock edge | Cross-coupled inverters require a stable input | Constrains the maximum combinational delay |
| Hold Time | Minimum data stable window after clock edge | Latch retention loop isolation | Constrains the minimum combinational delay |
| Min Pulse Width (MPW) | Latch Characterization Points | Prevents latch contention to settle impacting clk2q | Fundamental flip-flop limit |

## 2. Metastable Devices

## 2. 1 Setup and Hold Time: The Data Pulse Width Perspective

Setup times and hold times are defined with respect to clock edges but actually describe physical phenomena that require the data signal pulses to be wider than a minimum value. When data changes long before the clock edge, a setup check is meaningless. And if the data changes long after the clock edge, it is most likely not intended to be captured at that clock edge. That is what checked-in static timing analysis is, where setup and hold checks are based on max and min arrivals of data changes respectively and it does not matter if the arrivals are outside the MPW window.

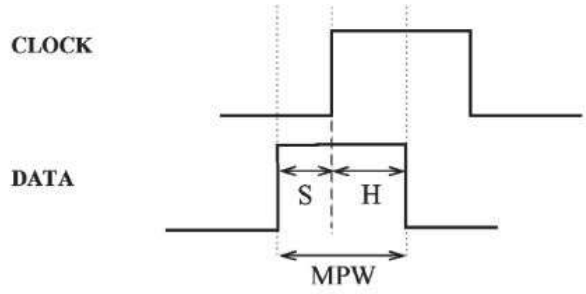


***Figure 1:*** *Setup Time (S), Hold Time (H), and Minimum Pulse Width (MPW) relationships in flip-flop timing.*

In addition, if the data signals are changing close to the clock edge, the contention within the latch retention loop makes it difficult for the cross-coupled inverters to have a stable rail-to-rail signal swing. The minimum pulse width requirement avoids the metastable equilibrium where the nodes inside the latch have not yet settled after a transition. If this condition is not met, the latch will enter a metastable state with an unknown output.As can be seen, at the cell physics level, setup time can actually be traded for hold time, so decreasing the setup time usually increases the hold time requirement. The trade-offs may increase the clock-to-Q delay, which may lead to fallout on the downstream timing path. These trade-offs are core to flip-flop characterization.

Timing diagrams showing the relationship of data arrival, clock edge, and output sampling are used to illustrate setup skew, hold skew, and associated clock-to-Q delay. These diagrams are the basis of static timing analysis. They show a failure of the circuit caused by diminishing setup or hold skew.

## 2. 2 Time windows and clock-to-Q delay of a latch

Plot shows setup and hold time characterization in relation to the clk-2-q delay of a flip-flop. The curve shows the chosen point of characterization is

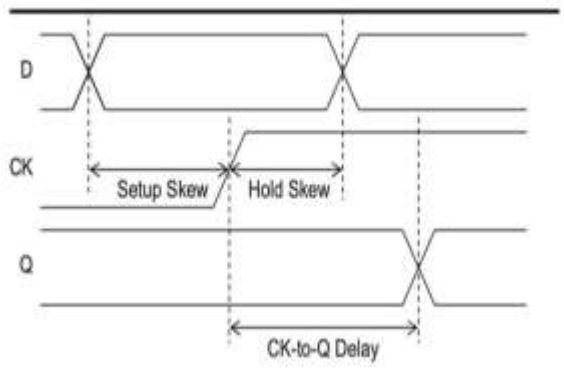


***Figure 2:*** *Timing diagram showing setup skew, hold skew, and clock-to-Q delay relationships in flip-flop operation.*

at 10% degradation of the clock-2-q delay at an infinite setup or hold time assumption. Here it is significant to see the non-linearity in the curve and how clk-2-q shoots up when the data switching approaches very close to the clock edge. In contemporary HPC designs where every bit of overmargining is proven to be expensive, it is extremely crucial to push the characterization point as close to clock edge as possible, to minimize setup and hold requirements. Also, it is important to keep in mind how setup and hold violations might cause a ripple effect of logic stabilization delay in downstream logic.

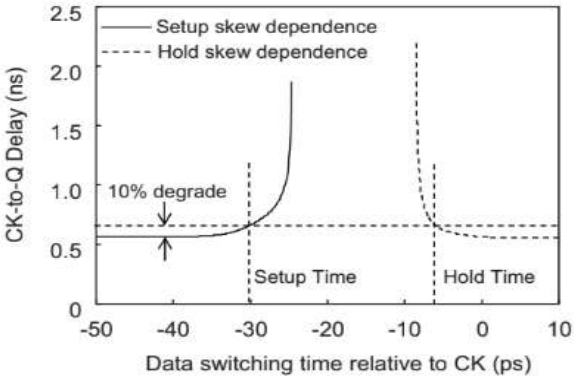


***Figure 3:*** *Relation of Setup, Hold and CLK-2-Q Characterization*

## 2.3 The myth of Metastability Elimination

The popular misconception is that good circuit design would eliminate the metastability of a latch. There is a need to look at metastability from a holistic, design usage perspective. There is a substantial amount of circuit improvement work done to create a metastability-hardened flip-flop or latch. This encompasses work to improve setup and hold time as well as resolution time (τ) [23][24]. It is not hard to realize that the point of metastability

is very unstable and can't be held for long due to natural noise in the design.

This can be understood from the Voltage Transfer Characteristic (VTC) of the forward and loopback inverters in a latch. These latches are cross-coupled, so one's output feeds into the other's input. The metastable point is the crossover point, and if one inverter slightly deviates from this point, its movement would cause the other inverter to move away, which in effect drives the first inverter to move away further. This can also be understood for a ball on a hump analogy.

But it is important to note here that, even though circuit improvement makes the stabilization process faster, as it still depends on noise in the system, the final logic of the design is unknown. Analogously, it is not known which side of the hump the ball will roll down. Inputs must be held stable across multiple capture clock edges at the receiving flip-flops; otherwise, downstream combinational logic may see illegal or conflicting logic levels. This is the real problem with metastability in the design.

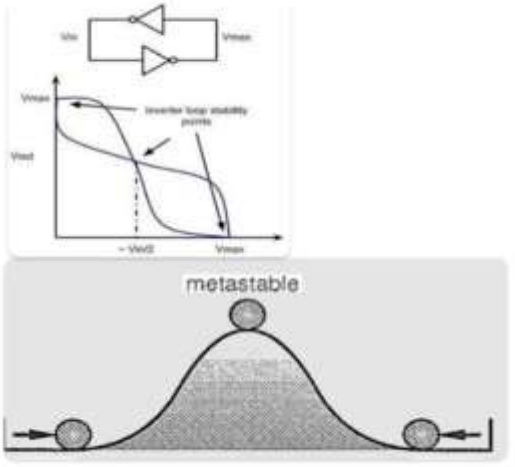


***Figure 4:*** *Voltage transfer characteristic (VTC) curves of cross-coupled inverters showing metastable point and ball-on-hump analogy for metastability behavior.*

## 2. 4 Looking at the Resolution Time

Resolution time (τ), defined by parameters of the devices as specified in standard cell characterization, is the time the latch takes to escape the metastable state.

By combining the simple physics of flip-flop operation with the basic design theory of synchronizers, metastability can be characterized by applying the voltage transfer characteristic (VTC) and the metastability settling time equations for the small-signal model of cross-coupled inverter latches [23]. This pair of inverters is a bistable device that has a metastable point in the VTC at VDD/2 (for symmetric inverters).

Resolution time (τ) is the time duration for a latch to leave the metastable state, which exhibits an exponential decay:

$$P(\text{failure}) \propto e^{(-t/\tau)}$$

Here, t represents the settling time, and τ denotes the resolution time constant. The value of τ is device-dependent, typically ranging from 10 to 100 picoseconds for many modern circuits.

The resolution time constant (τ) determining metastability escape is on the order of tens of picoseconds to hundreds of picoseconds in metastability-hardened cells of 7 nm and beyond technologies. Further resolution time beyond the minimum time needed to reach valid logic gate voltage thresholds only exponentially reduces the failure probability.

$$MTBF = 1/(2fCDC) * e^{(t/\tau)}$$

where fCDC is the number of CDC events, and t is the available settling time. For an optimized 2-stage synchronizer with $\tau \approx 30$ ps and $t \approx 100$ ps (a typical margin), the MTBF is still hundreds of years old at multi-gigahertz frequencies. Even for a single-stage flip-flop, this number is at tens of years. This is more than the usual product lifecycle.

Why is there a concern about MTBF?

The actual system reliability is governed by the sum of all system-level CDCs. A large SoC with thousands of crossings, if not properly designed, could have an unacceptable system failure rate. Though a single stage failure probability is very low (i.e., very high MTBF), the combined probability of all the CDCs passing comes to a finite number, often less than 10 years in contemporary HPC design. Also, each grade of chip (high reliability, industrial, consumer) has different MTBF targets, which in turn drive different synchronizer depths and margins.

$$MTBF_{Overall} = \frac{1}{1-\left(1-\frac{1}{MTBF_1}\right)\left(1-\frac{1}{MTBF_2}\right)\cdots\left(1-\frac{1}{MTBF_N}\right)} = \frac{1}{1-\left(1-\frac{1}{MTBF}\right)^N}$$

***Figure 5:*** *Combined MTBF Measurement in a System. If Higher Orders of the Taylor Series is ignored MTBFoverall ~ MTBF/N*

Please note that the assumption of the occurrence of a CDC event is random in the above derivation. The failure probability could be higher if the CDC clocks are somehow coherent and a CDC event occurs in every clock cycle [25]. In real designs, synchronizer modules are often implemented on CDCs of coherent clocks to mitigate clock skew or phase alignment-related timing issues. These designs are vulnerable to a lowered MTBF number.

## 3. Design Methodology and Results

The design, layout, and verification continue to involve multi-stage synchronizers and stabilizing control signals. This requires controlling data bus delay and skew during layout and controlling clock tree jitter during design and layout. Signoff quality STA (Crosstalk enabled and IR aware) should be done and probabilistic failure rates must be calculated to verify the design.

### 3.1 Data Transfer on a Control Signal

This is the simplest circuit to send data over a CDC. Though, there are many variations of handshake circuitry existing today. The transfer is gated by an AND gate, which is enabled where the controller becomes active. The controller in this case is a single bit signal and needs to be routed through a synchronizer module. The data bus does not need a synchronizer but is constrained to maximum delay which ensures arrival of the signal before the controller arrives at the receiving AND gate. For a two-stage synchronization on the controller path, due to the asynchronous relationship between clocks, the data path delay is constrained to one cycle of the destination clock. It is important to note that there is clock skew inside each of the clock domains CLKa and CLKb. As these clocks are asynchronous , the cross-domain skew is not important; however, while constraining these timing paths, it is important to budget for the skew within clock domains. In the following example 20% of the clock period is budgeted for clock skew.
set_max_delay -from [get_clocks CLKa to [get_clocks CLKb] -ignore_clock_latency [expr 0.8*max(CLKaperiod , max CLKbperiod)]
Please note that the AND gate could also be implemented as a clock gate.

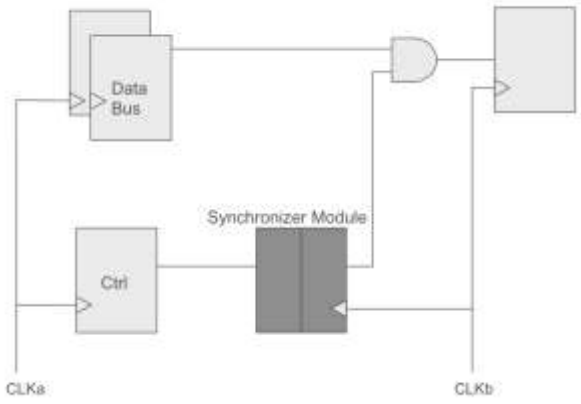


***Figure 6:*** *A Controller-Based Data Transfer from CLKa to CLKb*

### 3. 3 Reset Synchronizer Analysis

This paradox is illustrated via a reset synchronizer: the first stage of a reset synchronizer goes metastable when the reset is de-asserted by an asynchronous reset, while the second stage stays happy, although it is reset by the same asynchronous reset. This paradox is informative of the metastability phenomenon.
We all know the first FF can become metastable when RSTn is de-asserted. But why is the second FF "safe"? The second stage is fed from the same RSTn, and if it is metastable, the whole purpose of the circuit fails.
This is achieved by the design ensuring the second stage's input remains static (the reset value) at the de-assert. This is the beauty of this simple synchronizer logic. To explain with a latch circuit, with reset value of 0, as shown in the picture, a static “D” at reset value, does not cause contention, irrespective of a change in reset.

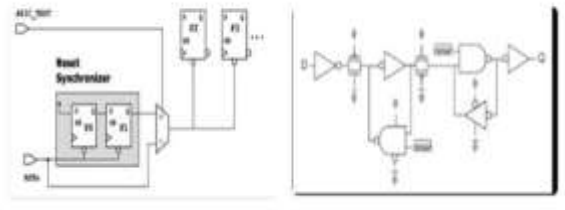

***Figure 7:*** *Reset synchronizer circuit topology showing two-stage synchronization with static reset data ensuring secondary stage safety.*

### 3. 5 Parallel Data Transfer and Gray Coding

When several bits are transferred from one asynchronous clock domain to another, multiple bits may change simultaneously (for example, from 1011 to 1100). Due to factors such as routing, buffering delay, and slew rate differentials, each transfer might arrive at different times. As alluded to earlier, in metastability, we do not know the final state of the latch; It is important to make sure the destination does not go into an illegal state.
Gray coding avoids such issues by ensuring that each code differs from its temporal neighbor only by one bit. If one bit goes metastable, the destination either settles to the new or the old value, both of which are legal states.

### 3. 6 Data Bus Skew on Parallel Transfer

Another timing constraint in parallel CDC that is often misunderstood is the skew between data bus bits. The allowable skew depends on whether the comparison is made with respect to the source or destination clock period. If the bus skew is less

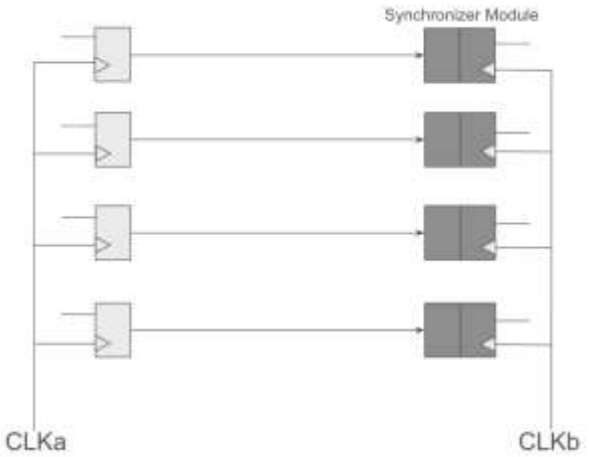


***Figure 8:*** *Parallel data transfer topology with multiple bit synchronizers between CLKa and CLKb clock domains.*

than the source clock period, simultaneous metastability cannot occur because a subsequent bit arriving after a source clock edge will not toggle the destination bus until the next clock edge.

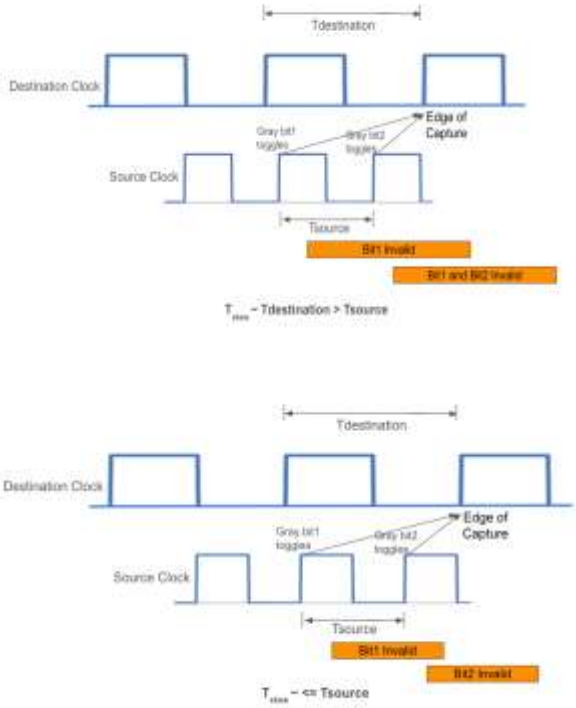


***Figure 9:*** *Data bus skew analysis showing metastability conditions relative to source and destination clock periods. Top: skew > source_period period (both bits are metastable). Bottom: skew < source period (only bit2 is metastable at destination capture).*

Layout teams also need to be aware of these limitations, and skew/clock skew analysis (maximum/minimum delay corner) should be performed. Most STA tools now contain CDC checkers that will warn of violation of the skew bounds that could lead to multiple-bit metastability. Proper specification, set_min/max_delay or set_data_check, of the skew constraint will prevent unbounded divergence of routing in the design.

## 3.7 Settling Time Simulation

A settling time simulation with various setup and hold time considerations.

***Figure 10:*** *Output Signal Stabilization at Different Setup and Hold Times.*

## 3.9 Asynchronous FIFO Implementation

Asynchronous FIFOs are a practical implementation of CDC, where unrelated clocks are used to perform transfers between the domains. The FIFO's RCLK and WCLK pointers are free-running within the domain. The pointers are compared to generate the FIFO's empty and full signals, which requires that the pointer values be Gray-coded and synchronized.

A key timing property is that data written during the current cycle cannot be read in the same cycle from the same address. The write pointer is not (and cannot be) visible to the read logic until the current cycle has ended. This is because the write pointer is synchronized (to within 2-3 logic stages) to the read side of the clock domain. This portion of the FIFO specification is important, and integration engineers should ensure they are aware of it.

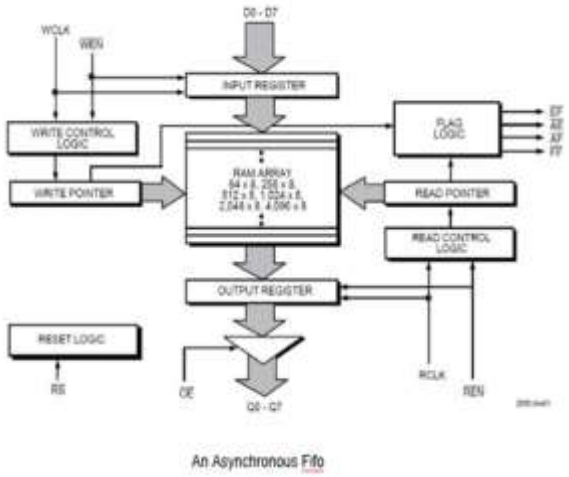


***Figure 11:*** *Asynchronous FIFO architecture showing write and read domains with pointer Gray code synchronization, RAM array, and control logic.*

The cycle is completed when the new empty/full status, determined from the Gray code pointers through the CDC synchronizers, enables the FIFO's memory cells through clock-gating and logic to avoid collisions and preserve data integrity. More advanced FIFOs can also include the threshold generation, error reporting in the clock domain, and power gating support with mode transition synchronization.

Schematic showing the memory array and read and write pointer synchronization scheme. The dotted line marks the clock domain boundary. The read pointer synchronously read through the decoder and not directly from the storage. Considerations to be taken while reading the first written data, as it will become valid on the read at best after one cycle in case of a two-stage synchronizer.

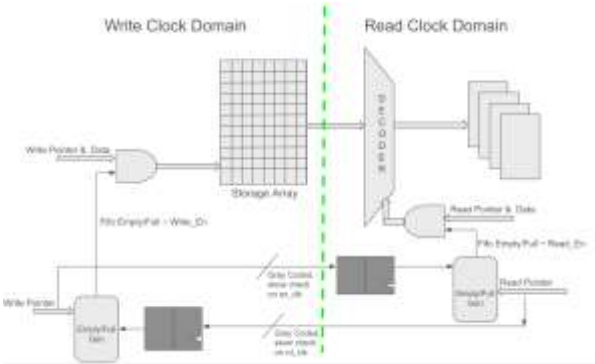


***Figure 12:*** *Asynchronous FIFO memory array showing read/write pointer synchronization scheme with clock domain boundary and decoder-based read path.*

## 4. Discussion

Our analysis puts classical CDC theory together with practical design conditions. By recasting setup/hold time in terms of data pulse width, we unify physical and mathematical timing theory models. The model also explains why standard cells exist with different setup/hold trade-offs, because the physics allows many designs for different optimization criteria [22-23].

A reset synchronizer analysis is not limited to purely asynchronous signal domains. Many designs will use an asynchronous reset signal that also must be synchronized to avoid spurious behavior. This principle that static data must avoid metastability generally holds: any signal crossing an asynchronous boundary must reach a static state before sampling [26]. Gray coding has found use in parallel communications and a small number of industrial implementations. The guarantee of only single outputs changing can be useful for multi-bit synchronization applications. Some generalized Gray code constructions allow for the transfer of non-consecutive numbers [6-8].

***Table 2:*** *Reset Synchronizer Stage Characteristics [5]*

| Stage | Input Type | Metastability Risk | Synchronization Mechanism |
|---|---|---|---|
| Primary | Asynchronous reset | High (potential metastability) | Direct clock capture with margin |
| Secondary | Static reset value | Negligible (no transitions) | Input stabilization from stage 1 |
| Tertiary+ | Synchronized reset | Negligible (exponential decay) | Multiple-stage cascading confidence |

The layout skew analysis represents a transition of CDC knowledge. Earlier architectures assumed a simpler skew matching goal (matching skew among bus bits), ignoring metastability. The current understanding is that the clock period at either side, not the symmetric skew distribution, is the physically meaningful constraint and that reliable implementations can be achieved at reasonable overhead.

***Table 3:*** *Data Bus Skew Constraint Examination*

| Skew Condition | Bit-X Arrival | Bit-Y Arrival | Simultaneous Risk |
|---|---|---|---|
| Skew > Source Period | Current source edge | Same destination edge | Both bits metastable |
| Skew < Source Period | Current source edge | Next source edge | Single-bit metastable |

A probabilistic analysis of failures shows that while MTBF calculations for individual synchronizers may be acceptable, the problem becomes meaningful at the system level and with large numbers of CDCs, and thus careful analysis and adequate margin [13-14] are required. High-reliability applications may require more stages (3+) and more settling time, while consumer applications may use shallower synchronizers with an acceptable amount of residual risk.

Asynchronous FIFO designs have existed since the early days of digital electronics, though multi-clock-domain systems, power-gated regions, and functional safety applications have been driving new FIFO designs [15-18]. Customary asynchronous FIFOs are simple and generic, but various non-standard features are now appearing, including multiple stages of synchronization, programmable thresholds for full and empty queues, and clock-domain error reporting.

More recent work on CDC verification focuses on using formal verification to verify that synchronizers are correct. These methods can exhaustively check properties of synchronizers as

well as corner-case violations that simulation methods cannot find [19-21]. Incorporating formal CDC verification into design flows is becoming more popular for safety-critical and high-reliability applications.

***Table 4:*** *Asynchronous FIFO Domain Characteristics [15-18]*

| Domain Component | Clock Signal | Synchronization Requirement | Timing Constraint |
|---|---|---|---|
| Write Pointer | WCLK | Gray code to read the side | 2-3 stage sync latency |
| Read Pointer | RCLK | Gray code to write on the side | 2-3 stage sync latency |
| Memory Array | WCLK/RCLK | No sync needed | Data available next cycle |

## 5. Conclusions

Metastability in digital circuits is a physical phenomenon that cannot be avoided but must be carefully managed through an integrated framework spanning design, layout, and verification. The setup and hold time minimums of flip-flops are dictated by the minimum pulse width of data that can be safely stored, rather than arbitrary timing windows, illustrating fundamental trade-offs inherent in cell design physics. Reset synchronizers leverage the static data property ensuring data stability during reset de-assertion guarantees no metastability in secondary stages despite potentially metastable primary stages, a principle that generalizes across all applications involving clock domain crossing. Gray coding ensures only single-bit transitions between consecutive values, eliminating illegal intermediate states and establishing it as the preferred approach for multibit transfers across asynchronous boundaries. Layout skew constraints derive from analysis, dependent on whether source or destination clock periods provide the physically meaningful constraint; symmetric skew matching alone proves insufficient for metastability elimination. While individual clock domain crossings exhibit acceptable mean time between failures, accumulation of events across large systems produces finite failure probabilities, requiring synchronizer depth to balance against total CDC count and reliability objectives. Asynchronous FIFOs exemplify practical implementations for clock domain crossing where pointer management, Gray code synchronization, and latency effects demand careful control to prevent data corruption and timing violations. Formal verification techniques increasingly supplement traditional simulation, enabling exhaustive proof of synchronizer correctness and detection of subtle corner-case violations. Coordinated integration of rigorous design methodology, stringent layout discipline, and formal verification constitutes essential requirements for implementing correct clock domain crossing. As system-on-chip designs grow increasingly complex and operate at higher frequencies, the relative importance of comprehensive metastability management escalates correspondingly, making theoretical foundations and systematic design rules critical for addressing contemporary digital system challenges.

## Author Statements:

- **Ethical approval:** The conducted research is not related to either human or animal use.
- **Conflict of interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper
- **Acknowledgement:** The authors declare that they have nobody or no-company to acknowledge.
- **Author contributions:** The authors declare that they have equal right on this paper.
- **Funding information:** The authors declare that there is no funding to be acknowledged.
- **Data availability statement:** The data that support the findings of this study are available on request from the corresponding author. The data are not publicly available due to privacy or ethical restrictions.
- **Use of AI Tools:** The author(s) declare that no generative AI or AI-assisted technologies were used in the writing process of this manuscript.

## References

[1] T.J. Chaney, C.E. Molnar, “Anomalous Behavior of Synchronizer and Arbiter Circuits.” 1973, IEEEXplore. https://ieeexplore.ieee.org/document/1672323

[2] Department of Informatics, “Metastability and Clock Domain Crossing.” University of Oslo, 2021. https://www.uio.no/studier/emner/matnat/ifi/IN3160/v22/forelesningsfoiler/in3160-clock-domains.pdf

[3] Yu Zhao et al., “Synchronization Design Method for High-Speed Signal Transmission Between FPGAs in DIGITAL Oscilloscope,” 2023. https://ieeexplore.ieee.org/document/10270086

[4] Dejan Marković et al., “Analysis and Design of Low-Energy Flip-Flops,” Academia, 2001.

https://www.academia.edu/4568265/Analysis_and_design_of_low_energy_flip_flops

[5] Peter Vanbekbergen, "A Design and Validation System for Asynchronous Circuits," ACM Digital Library, 1995. https://dl.acm.org/doi/pdf/10.1145/217474.217618

[6] Md. Mohsin Alia and Mst. Shakila Khan Rumi, "Parallel binary reflected Gray code sequence generation on multicore architectures," International Journal of Parallel, Emergent, and Distributed Systems, 2014. https://users.cecs.anu.edu.au/~mohsin/downloads/parallel-binary-reflected-gray-code-sequence-ijpeds-2013.pdf

[7] Anatoly Beletsky, Nguyen Viet Hung, "On the construction of generalized Gray code," 2006. https://www.researchgate.net/publication/301472987

[8] Dr. Tammisetti Ashok, "Design And Verification of a Parameterized Asynchronous FIFO Using Gray Code Synchronization." IJIRT, 2026. https://ijirt.org/publishedpaper/IJIRT192554_PAPER.pdf

[9] Guillaume Plassan et al., "Conclusively Verifying Clock-Domain Crossings in Very Large Hardware Designs," IEEE, 2016. https://www.researchgate.net/profile/Guillaume-Plassan/publication/310809579

[10] JASON CONG et al., "Layout optimization," SpringerNature Link, 1997. https://link.springer.com/chapter/10.1007/978-1-4615-5685-5_8

[11] EDN "Clock Domain Crossing Techniques & Synchronizers," 2014. https://www.edn.com/synchronizer-techniques-for-multi-clock-domain-socs-fpgas/

[12] David Harris et al., "Timing Analysis with Clock Skew." https://pages.hmc.edu/harris/research/tau99.pdf

[13] Suwen Yang, Mark Greenstreet, "Computing Synchronizer Failure Probabilities." https://www.cs.ubc.ca/~mrg/mypapers/date07.pdf

[14] Amit Kulkarni et al., "Evolution of CDC recipe: Learning through real case studies and methodology improvements." https://dvcon-proceedings.org/wp-content/uploads/evolution-of-cdc-recipe-learning-through-real-case-studies-and-methodology-improvements.pdf

[15] EFINIX, "Asynchronous FIFO Operation." https://www.efinixinc.com/docs-html/fifo2-core-ug/topics/fifo-core-async.html

[16] Hiranmaye Chandu, "Advancements in Asynchronous FIFO Design: A Review of Recent Innovations and Trends," ResearchGate, 2023. https://www.researchgate.net/publication/385682063

[17] Guoqi Xie et al., "Fast Functional Safety Verification for Distributed Automotive Applications During Early Design Phase," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 2018. http://esnl.hnu.edu.cn/fast_functional_safety_verification_for_distributed_automotive_applications_during_early_design_phase.pdf

[18] Aradhana Kumari, "Low power architectures for clock domain crossing in SoC design," STI. https://www.gsaglobal.org/events/wp-content/uploads/sites/10/2024/12/design-1-aradhana-kumari-stmicro-wish2025

[19] Bing Li, Chris Ka-Kei Kwok. "Automatic formal verification of clock domain crossing signals," ACM Digital Library, 2009. https://dl.acm.org/doi/abs/10.5555/1509633.1509782

[20] Bo Seung Kwon K et al., "DEVS-Based CDC Synchronizer Design for Fast Debugging of Metastability," MDPI, 2024. https://www.mdpi.com/2079-9292/13/24/5048

[21] Aman Kumar, "Pragmatic Formal Verification Methodology for Clock Domain Crossing (CDC)," DVCON, 2023. https://dvcon-proceedings.org/wp-content/uploads/91837.pdf

[22] T. Okumura and M. Hashimoto, "Setup time, hold time and clock-to-Q delay computation under dynamic supply noise," IEEE Custom Integrated Circuits Conference (CICC), 2010. https://ieeexplore.ieee.org/document/5617426

[23] David Rennie, et al."Performance, metastability and soft-error robustness tradeoffs for flip-flops in 40nm CMOS," IEEE, 2011. https://ieeexplore.ieee.org/document/6055415

[24] T. Kacprzak and A. Albicki, "Analysis of metastable operation in RS CMOS flip-flops," IEEE Journal of Solid-State Circuits, 1987. https://ieeexplore.ieee.org/document/1052671

[25] S. Beer, R. Ginosar, R. Dobkin, and Y. Weizman, "MTBF Estimation in Coherent Clock Domains," IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC), 2013. https://ieeexplore.ieee.org/document/6546191

[26] V. Patel, V. Mer, J. Patoliya, and B. Soni, "Design & Implementation of Novel Asynchronous FIFO," IEEE International Symposium on Smart Electronic Systems (iSES), 2023. https://ieeexplore.ieee.org/document/10467054