Low-Power Architectures for Clock Domain Crossing in SoC Design

Aradhana Kumari,
STMicroelectronics, Crolles, France
aradhana.kumari01@st.com

***Abstract*— Modern System-on-Chip (SoC) designs incorporate multiple clock domains to optimize dynamic power and throughput. However, clock domain crossing (CDC) introduces challenges such as metastability and data corruption. Traditional solutions like FIFOs and synchronizers often incur high area and power overheads. This paper proposes novel low-power CDC architectures that significantly reduce power, area, and latency. The proposed designs include an optimized FIFO and a power-efficient synchronizer, offering over 75% savings in Datapath registers and 50% reduction in synchronizer flip-flops. These techniques are ideal for low-power systems and scalable SoC designs.**

Keywords—Clock Domain Crossing (CDC), Low-Power SoC Design, Synchronizer Architecture, Multi-Clock Domain Systems, FIFO Optimization, Energy-Efficient Architectures.

I. INTRODUCTION

The increasing complexity of System-on-Chip (SoC) designs has led to the widespread use of multiple clock domains to balance performance and power consumption. These domains frequently exchange data and control signals, necessitating robust clock domain crossing (CDC) mechanisms. Improper handling of CDC can result in metastability, data loss, or corruption, affecting system reliability [1].

Traditional CDC solutions—such as multi-stage synchronizers [2], reset synchronizers [3], and FIFOs [4]—address these issues but at the cost of increased area, latency, and power. Synchronizers mitigate metastability but require multiple flip-flops per signal, leading to high power consumption when scaled across hundreds of signals. FIFOs, while effective for data transfer, introduce latency and consume significant areas due to deep buffering.

As low-power design becomes critical, especially in mobile and edge computing applications, there is a pressing need for more efficient CDC architecture. These innovations build upon prior work in adaptive clock gating [2], energy-quality scalable synchronizers [3], and FIFO optimization [5], and are aligned with the broader goals of scalable, fault-tolerant SoC design [6].

II. BACKGROUND AND MOTIVATION

CDC issues arise when signals traverse domains operating at different clock frequencies or phases. Synchronizers mitigate metastability but require multiple flip-flops per signal, leading to high power consumption when scaled across hundreds of signals. FIFOs, while effective for data transfer, introduce latency and consume significant areas due to deep buffering.

Metastability arises when setup or hold time requirements are violated during signal transfer between clock domains. In such cases, the receiving flip-flop may enter an unstable state that eventually resolves to a logical '0' or '1', but the outcome is unpredictable and timing dependent. This behavior is a fundamental transistor-level phenomenon and cannot be reliably captured through simulation or formal verification. As a result, metastability cannot be eliminated; instead, it must be carefully managed through robust design techniques. This concept is illustrated in Fig. 1, which shows the timing

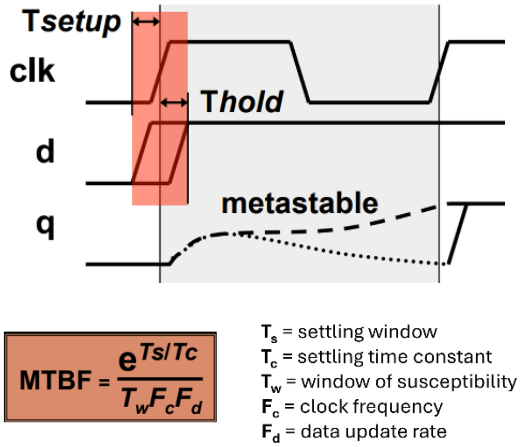

Fig. 1. Clock Domain Crossing: Metastability & MTBF

relationship between the clock signal, data input, and output of a flip-flop. The diagram highlights how violations of setup and hold times can lead to metastable behavior at the output. Mean Time Between Failures (MTBF), a reliability metric used to estimate the average time between system or component failures. The numerator $e^{Ts/Tc}$ increases MTBF as time of stress increases (assuming the system is robust). The denominator includes factors that reduce MTBF due to real-world conditions, design and process. In digital systems, reset synchronization ensures that asynchronous resets are safely and reliably aligned with clock domains to prevent metastability and unpredictable behavior. Mathematically, MTBF increases exponentially with the resolution time, meaning that adding more flip-flops or increasing the clock period can significantly improve reliability. However, this comes at the cost of latency and power. Together, MTBF and reset synchronization contribute to designing fault-tolerant systems by minimizing downtime and ensuring stable recovery from faults.

The motivation behind this work is to develop CDC and reset distribution architectures that:

1. Minimize power and area overheads.
2. Maintain or improve latency and performance.
3. Scale efficiently with increasing SoC complexity.
4. Reduce implementation cycle time.

III. Proposed Architectures

A. Synchronous FIFO

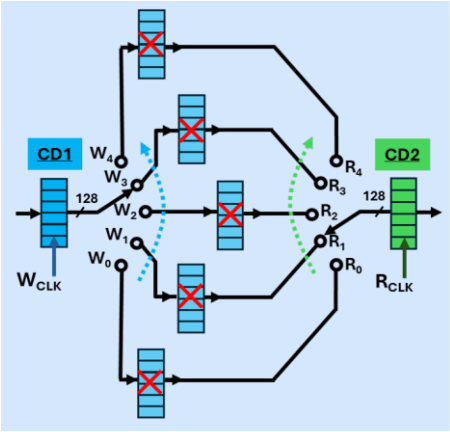

Fig. 2. Standard synchronous FIFO

Fig. 2 illustrates a conventional depth-5 synchronous FIFO used for transferring 128-bit data between two clock domains (CD1 and CD2). This design requires five 128-bit registers, contributing to substantial area and power usage. Data is written and read in the order it arrives, ensuring predictable sequencing. The FIFO consists of a series of registers, with control data blocks (CD1 and CD2) managing the write and read operations. Both the write clock (WCLK) and read clock (RCLK) are synchronized, simplifying timing and control logic. Write pointers (W0 to W3) and read pointers (R0 to R3) track the positions for data insertion and retrieval.

In contrast, Fig. 3 presents the proposed optimized FIFO architecture. By leveraging clock phase alignment between WCLK (write clock) and RCLK (read clock), the design reduces the register count to just two 128-bit registers CD1 and CD2. the FIFO depth (128 entries in CD1 and CD2) is managed using clock phase selection rather than a full memory buffer. The circular selector labeled "R/W clock phase select" with quadrants R_0/W_0 through R_3/W_3 indicates that the system uses clock phase multiplexing to determine which register to read from or write to at a given time. This approach leverages the deterministic timing of a synchronous system to reduce the FIFO to just two registers, drastically reducing the data transfer latency and throughput requirements. This technique is particularly useful in low-latency, high-speed designs.

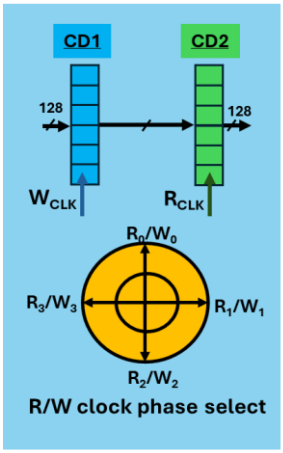

Fig. 3. Proposed synchronous FIFO

The clock phase selection is illustrated in the chronogram of Fig. 4. It shows the desired write clock Wclk and read clock Rclk relationships. The edge crossover between Wclk and Rclk can easily be detected by monitoring the register values CD1 And CD2 during an initial or in-field clock calibration. Table I summarizes the quantified advantages of the proposed 2-register FIFO solution:

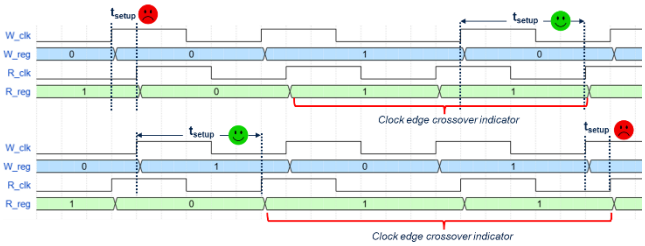

Fig. 4. Edge crossover detection

- 75% reduction in datapath register area and power.
- Lower latency due to reduced buffering.
- Simplified control logic for data transfer.

Depth-5 FIFO	Contemporary	Proposed
W_clk =250MHz	768 FFs	128 FFs
NAND gates	1536	256
R_clk@ 1GHz	128 FFs	128 FFs
Latency (UI)	3.25	1.25
DFT FFs	896	256

Table I. Result comparison

B. Control Path Synchronizer

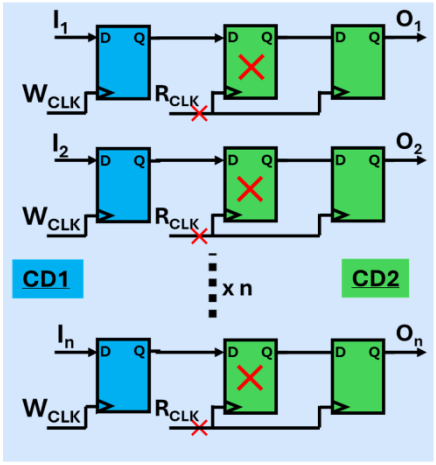

Fig. 5. Control signal CDC

Fig. 5 This image depicts a 2-stage synchronizer used for asynchronous clock domain crossing between two distinct clock domains: W_{CLK} and R_{CLK}. Each synchronizer stage consists of two D flip-flops connected in series. The input signal (I1, I2, ..., In) originates in the W_{CLK} domain, and the goal is to safely transfer it into the R_{CLK} domain. This 2-stage arrangement is designed to reduce the risk of metastability, allowing any unstable transitions in the input signal to settle before being used in the synchronous domain. It is a standard technique in clock domain crossing (CDC) design to ensure reliable data transfer between asynchronous systems. Each signal requires two flip-flops, and hundreds of such synchronizers are deployed in a typical SoC. The proposed

signal synchronizer gets rid of one of the flip-flops in the R_{CLK} domain (crossed in red).

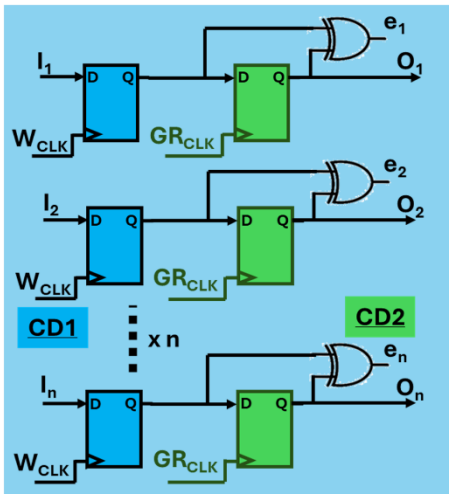

Fig. 6(a). Proposed control signal CDC

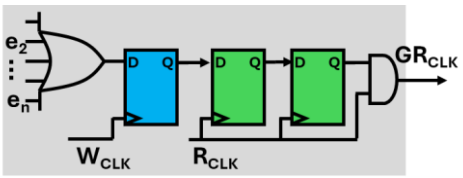

Fig. 6(b). Low power sparse clocking

A highly power optimal synchronizer is proposed in Fig. 6(a) & (b). The proposed synchronizer in introduces a clock-gating mechanism. Flip-flops in the destination domain R_{CLK} are gated GR_{CLK} and only activated when input data changes (detected by e_n) in the source domain (W_{CLK}). The proposed synchronizer ensures that all the flip-flops in the destination clock domain R_{CLK} remain clock gated (GR_{CLK}) and are clocked only when the input data changes (from source W_{CLK} domain). The proposed approach:

- Reduces flip-flop count by 50%.
- Minimizes dynamic power by avoiding unnecessary clock toggling.
- Improves scalability for large control signal sets.

Table II illustrates major power gains for an example 8 signal CDC system.

8 x 2-stage sync. @600MHz	Contemporary (µW)	Proposed (µW)	% Gain
10% data toggle	39	22	**41**
1% data toggle	30	8	**72**

Table II

C. Reset Synchronizer

Traditional reset synchronizers ensure safe de-assertion of global reset signals across clock domains but add significant routing and timing complexity. The image in Figure 7 illustrates the behavior of a reset synchronizer during both assertion and de-assertion phases of the reset signal, along with associated timing waveforms. The circuit consists of two D flip-flops in series, forming a reset synchronizer. The RESETN signal is fed asynchronously to both flip-flops, allowing immediate reset assertion without waiting for a

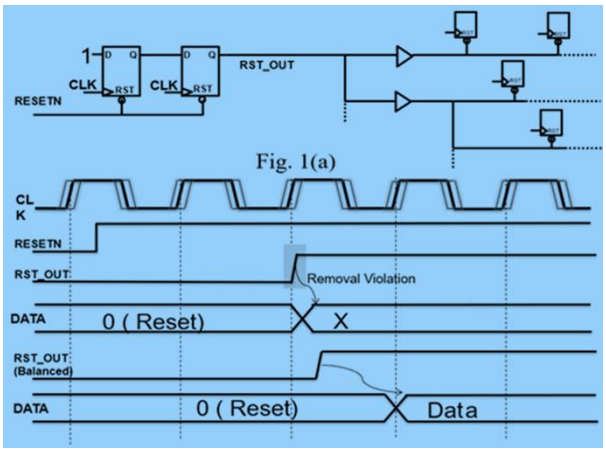

Fig.7. Reset synchronizer & network

clock edge. This ensures that all fanout registers connected to RST_OUT are promptly reset.

During reset de-assertion, the signal first reaches the synchronizer flops. The first flip-flop captures the de-asserted value (logic high) on the next clock edge, and the second flip-flop propagates it to RST_OUT on the subsequent clock edge. This makes the de-assertion synchronous, and hence, recovery and removal timing checks must be met from the second stage of the synchronizer to all domain registers.

The timing diagram shows how removal violations can occur if RST_OUT transitions at the same time as a clock edge, violating the required timing margin. To avoid this, RST_OUT must be balanced across all flops in the domain, which demands meticulous timing analysis and routing effort. This process is repetitive, area-consuming, and power-intensive, especially in large designs, making reset synchronization a critical and non-trivial aspect of digital design.

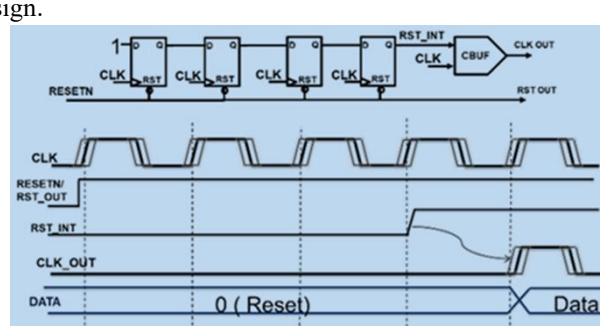

Fig.8. Proposed pre-reset de-assertion clock gating

Figure 8 presents a tunable clock delay circuit, which builds upon the conventional reset synchronizer architecture but introduces a distinct functionality. In this design, the asynchronous reset signal RESETN is directly connected to the reset tree, ensuring immediate reset assertion across all flip-flops. Upon de-assertion of RESETN, the circuit initiates a delayed activation of the clock gating control signal RST_INT, which becomes active only after a delay of approximately 4 to 5 clock cycles. This delay is achieved by cascading multiple flip-flops, each clocked by the system clock (CLK), allowing the reset de-assertion to propagate sequentially through the chain. The delayed RST_INT signal can then be used to safely enable downstream logic or clock gating, ensuring that all reset-dependent elements have stabilized before resuming normal operation. This approach

enhances robustness in reset recovery and mitigates timing hazards associated with premature activation of gated clocks.

A couple of high-speed designs were implemented using the proposed methodology and the results demonstrate its effectiveness in high frequency designs. Table III summarizes reset timing violations post place and route optimization. Both designs reported are high frequency design in 28FDSOI technology. The 4GHz design is a 2K gate design and the 2.4GHz design is a 5K gate design. WNS and TNS are abbreviations for Worst Negative Slack and Total Negative Slack in the designs.

	Recovery Violation		
Frequency	WNS (nS)	TNS (nS)	No. of Paths
4 GHz	-0.4758	-1.9773	6
2.4 GHz	-0.6702	-936.1055	3427
	Removal Violation		
Frequency	WNS (nS)	TNS (nS)	No. of Paths
4 GHz	-0.5964	-132.39	379
2.4 GHz	-0.050	-5.5164	194

Table III

IV. CONCLUSION

This paper presents a suite of innovative low-power architectures for clock domain crossing (CDC) in modern SoC designs, addressing the critical challenges of metastability, timing violations, and excessive power and area overheads. By optimizing traditional synchronizer and FIFO structures, and introducing clock-gated and delay-tuned reset synchronizers, the proposed solutions achieve significant reductions in flip-flop count, dynamic power, and latency. The 2-register FIFO architecture and sparse clocking synchronizer demonstrate over 75% savings in datapath registers and 50% reduction in synchronizer flip-flops, making them highly suitable for scalable, high-frequency, and energy-efficient SoCs. The proposed reset synchronizer approach mitigates removal violations, simplifies timing closure, and enhances robustness in high-frequency designs. These techniques not only improve robustness and timing closure but also simplify implementation and verification cycles. The results validate the effectiveness of these architectures in real-world high-speed designs, paving the way for broader adoption in next-generation low-power systems. As a conclusion, the proposed techniques enable designers to achieve significant reductions in SoC power and area without compromising on performance, making them ideal for low power systems.

REFERENCES

[1] **K. J. Kuhn**, "Reducing metastability in synchronizers for multiple clock domains," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 5, pp. 1623–1632, May 2020.

[2] **S. Das and A. Raychowdhury**, "A low-power synchronizer design using adaptive clock gating for CDC," *IEEE Transactions on VLSI Systems*, vol. 28, no. 12, pp. 2781–2790, Dec. 2020.

[3] **M. Alioto**, "Energy-quality scalable synchronizers for ultra-low power SoCs," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 1, pp. 45–56, Mar. 2020.

[4] **J. Garside**, "Metastability and synchronizers: A tutorial," *IEEE Design & Test of Computers*, vol. 34, no. 4, pp. 8–17, Aug. 2017.

[5] **Y. Kim et al.**, "Design of low-latency and low-power FIFO for CDC in SoCs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 11, pp. 3456–3468, Nov.

[6] **J. L. Hennessy and D. A. Patterson**, *Computer Architecture: A Quantitative Approach*, 6th ed., Morgan Kaufmann, 2019.