Application of Asynchronous FIFO in High-Speed Interface Data Transmission

Ruiqi Ye

Country North Cross School Shanghai,Shanghai, 200082, China
gouhohhf@gmail.com

Abstract. FIFO has been widely used in various data transmission systems due to its ability to temporarily store and manage data efficiently. In modern digital systems, ensuring efficient and reliable data transfer between components operating at different clock domains is a critical challenge. Asynchronous FIFO provides an effective solution to this problem by addressing the complexities of clock domain crossing. It minimizes data loss, mitigates metastability issues, and ensures seamless communication between components with independent clock domains. This makes asynchronous FIFO particularly valuable in high-speed data transmission systems, where maintaining data integrity and system stability is crucial. This paper explores the design, implementation, and optimization of asynchronous FIFO structures in high-speed interfaces, highlighting their role in achieving reliable, and low-latency data processing. Applications such as network communication, IoT systems, and high-speed digital interfaces benefit significantly from asynchronous FIFO's ability to handle complex data flows. The findings emphasize its importance in modern systems, particularly in scenarios requiring high-speed, efficient, and low-latency data handling.

Keywords: Data Transmission, FIFO, Highspeed Interface.

1 Introduction

With the increasing demand for high-speed data processing and communication, efficient data transmission across different clock domains has become a crucial challenge in modern digital systems. Many high-speed applications, such as network, inter-system, and digital process system, requires data transfer between components in different clock sequence. However, traditional FIFO functions struggled to process data transfer efficiently, leading to issues such as metastability, data loss, and system inefficiencies [1].

To address this problem, Asynchronous FIFO emerges as an effective solution. Asynchronous FIFO is widely used in digital systems to temporarily store and transfer data between components that operate under different clocks. Unlike synchronous FIFOs, which function within a single clock domain, asynchronous FIFOs enable communication between independent clock domains without requiring precise clock synchronization [1]. This makes it essential in highspeed interfaces where data transfer in different sub-system under different clocks, such as PCIe, USB, and Ethernet [2].

J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_90

One of the primary challenges in asynchronous data transmission is clock domain crossing, which occurs when data moves from one clock domain to another. Improper handling of CDC can result in metastability, a state where signals are neither high nor low, leading to unpredictable system behaviour [3].To reduce metastability, asynchronous FIFOs employ synchronization techniques in order to ensuring stable and reliable data transfer [4]. In addition, FIFO's depth and latency must be carefully designed to prevent data overflow and underflow, which can significantly impact system performance.

Given the increasing complexity of high-speed digital interfaces, optimizing the design and implementation of asynchronous FIFOs is essential for ensuring data integrity and minimizing latency [4]. This paper examines many applications about Asynchronous FIFO to introduce the significance and importance of Asynchronous FIFO.

2 Asynchronous Fifo In High-Speed Interface Data Transmission

2.1 Asynchronous FIFO in highspeed network interface

High-speed FIFOs have been widely utilized in network communication systems. The primary factor contributing to the cost of FIFO in such devices is the memory used in the controller. Depending on the design, the controller can employ either on-chip or off-chip FIFO, which influences both the performance of the FIFO and the complexity of the system's architecture. As systems demand higher speeds and wider, faster system buses, the complexity of FIFO design increases, necessitating improvements to its architecture. Since FIFO plays a critical role in determining the controller's operation and the latency experienced by a device, enhancements are essential.

A methodology has been proposed to determine the optimal size of FIFO for a given workload. This approach takes into account various factors, including system characteristics, internal arbitration policies governing the controller's transmit and receive operations, and the parameters of high-speed networks. By analyzing the system's latency tolerance, this method helps prevent data transfer overflows. It also demonstrates that FIFO size is influenced by system parameters such as the maximum DMA transfer size. A cut-through design for data packets is highlighted as an effective approach to minimize latency and boost system efficiency, as depicted in Fig. 1. Regarding data packet transmission and reception, the study shows that system performance is affected by both the data byte size and the system's bus transfer size [1].

2.2 Asynchronous FIFO's speed optimizing

Optimizing the FIFO design is crucial for enhancing system performance. By incorporating fight techniques, as shown in Fig. 1, along with relative delay optimization, asymmetric gates, and skewed gates, the performance of the FIFO in high-speed applications can be significantly improved. Fight optimization involves

reducing latency by balancing the effort across three components: logical effort, electrical effort, and branching effort. When the effort of each path is equal, the system achieves minimal latency, improving overall efficiency. Relative delay optimization focuses on the two main components of FIFO design, the data latches and control cells. Through the use of SRAM and enhanced logic control speed, the FIFO's performance can be greatly improved [5].

Logic gate optimization aims to enhance the micro-pipeline logic. While it cannot be directly applied to the control cell, the circuit can be divided into sub-blocks, allowing logic gates to control most components. This improves both accuracy and the FIFO's ability to shift data effectively. Though some of the methodology mentioned cannot apply to all the circumstances, it provides constructive advice for future FIFO design. Further researches can focus on the balance between the latency and power [2].

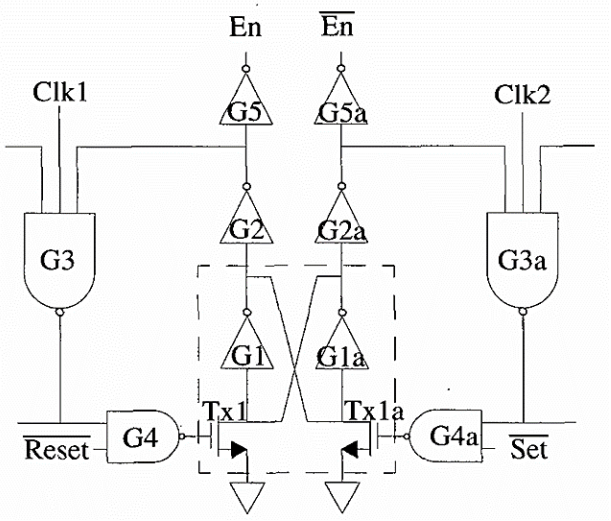

Fig. 1. Control cell optimization [2]

2.3 FIFO's application in IoT and cyber-physical system

As the industrial era advances, IoT technology is becoming increasingly significant. In typical IoT setups, numerous sensors act as data providers, while a central platform serves as the coordinator. These systems require both data storage and processing capabilities, making tools for evaluating strategic decisions essential. This article emphasizes the necessity of versatile solutions to address frequency mismatches among various nodes. Since data transmission modules operate at different frequencies depending on their specific applications, a fog computing architecture is proposed to bridge these discrepancies, as illustrated in Fig. 2. This contains three subsets: (1) IoT nodes (2) the cloud (3) Gateway layer. This paper proposes a solution that leverages a high-speed data acquisition (HSDA) algorithm combined with a reconfigurable FIFO design to enhance communication efficiency across various nodes within an IoT system. The test result indicates that with the reconfigurable FIFO design, IoT system

will reach a higher efficiency and improve the power performance. The study also points out further improvement on such design can focus on optimizing the design of the r-FIFO, allowing the system to adapt to 5G and AI IoT applications, and to integrate AI computing unit, thus better processing the sensors data [6].

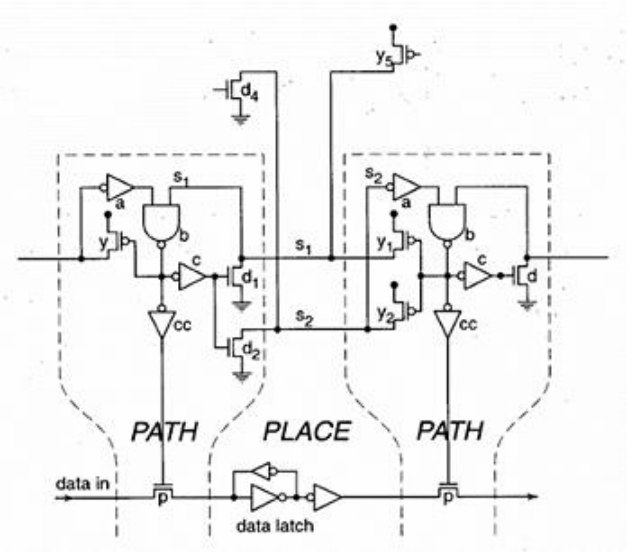

Fig. 2. Gasp with the twin state conductor [6]

2.4 Design for muti-function serial port with high-speed and large capacity of Asynchronous FIFO

With the improvement in automatic degree, more and more serial ports are applied on industrial devices and automatic equipment. By introducing fight techniques, relative delay optimization, asymmetric gates and skewed gates, the performance of the FIFO for high-speed use increases. To improve the system's performance, FPGA-controlled SDRAM with control program can be use in order to realize high-speed and large capacity FIFO. This solution can reduce the cost compare with conventional solution and has a relatively high universal and can apply in most of the system which requires a large buffer for data, as shown in Fig. 3.

Compared with random DRAM, SDRAM offers two key advantages. First, SDRAM responds to input signals by waiting for the clock, allowing it to synchronize with the system's bus clock and maintain system coherence. Second, SDRAM operates in a pipeline mode for executing commands, enabling more efficient processing and supporting its more complex operating modes compared to DRAM. All in all, the application of SDRAM in FIFO has high Portability and Typicality that allows it to provide a strong performance and a good compatibility [7].

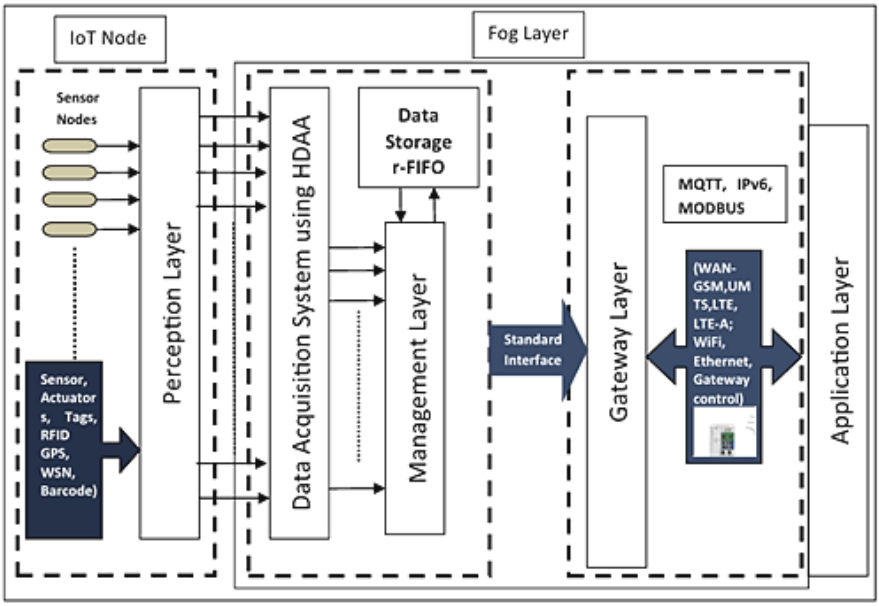

Fig. 3. Structure for system design [7]

2.5 Wrapper of PCI Express with FIFO Interfaces Based on FPGA

With the continuous progress in the electronics industry, chip performance has significantly improved. To address the growing need for higher bandwidth, PCI Express, a new bus technology, was introduced. PCI Express adopts a serial architecture that resolves several limitations of conventional parallel bus systems by utilizing clock data recovery (CDR) and differential signaling. This design is based on a packet-driven transaction model with a dedicated, non-shared data bus.

The PCI Express structure consists of three primary layers. The Transaction Layer, located at the top, is responsible for handling the transmission and reception of Transaction Layer Packets (TLPs). The Data Link Layer, positioned in the middle, ensures reliable TLP exchanges between components. Finally, the Physical Layer connects the Data Link Layer to the signaling hardware, facilitating data transmission over the link while performing tasks such as framing, deframing, scrambling, descrambling, encoding, and decoding [8].

2.6 Design of High-Speed Parallel Data Interface Based on ARM & FPGA

With the advent of the information age, embedded systems with network connectivity have been widely adopted. ARM-based platforms are extensively used in embedded systems, while FPGA technology is also noticeable due to its advantages in parallelism and reconfigurability. These features make FPGA suitable for high-speed data transmission, meeting the needs for growing challenges in modern data interface

designs. To meet the demand for high-speed digital processing, the FPGA+ARM architecture has been deployed. In the system show below, FPGA sequentially transmits six synchronized groups of parallel data to the ARM, as Fig. 4 indicate.

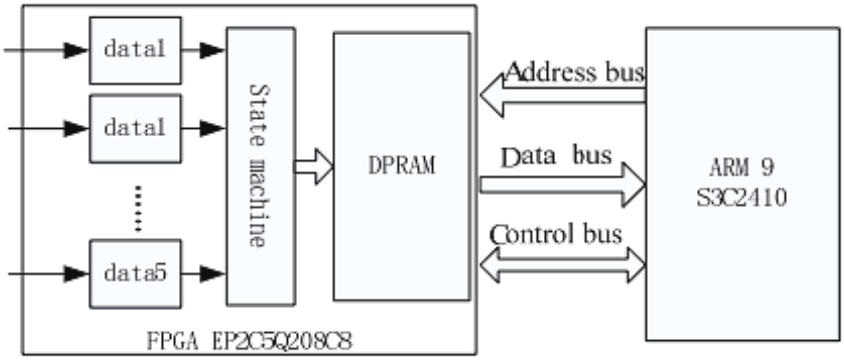

Fig. 4. FIFO design for data transmission [9]

The system adopts ping-pong control principles, along with deserialization and synchronization techniques, enabling low-speed modules to process high-speed data effectively. ARM considers the FPGA as a device connected to the external system bus of the S3C2410 to facilitate data buffering. The ARM processor accesses the buffered data by reading it through the external system bus of the FPGA.

In summary, on the ARM+FPGA hardware platform, the high-speed parallel interface addresses the data synchronization challenges encountered in asynchronous time-domain transmission. This approach ensures proper alignment of data during high-speed transfers. The communication speed between ARM and FPGA in this system reaches up to 11.25 Mbit/s. Tests confirm the system's stability and validate the accuracy and reliability of the high-speed parallel interface design between ARM and FPGA [9].

2.7 Design, Simulation and Realization of a Parametrizable, Configurable and Modular Asynchronous FIFO

A FIFO is a data structure where data is sequentially written into FIFO memory in one clock domain and sequentially read from the same memory in another clock domain. Asynchronous FIFOs are specifically designed to ensure safe data transfer between different clock domains. They are widely used in applications like video interfaces and communications with devices such as CPUs, GPUs, and off-board components. The internal status of the FIFO buffer is represented by full and empty flags, which are determined by its design. In asynchronous FIFOs, this requires comparing the write clock and read clock, which belong to separate clock domains.

A common approach for designing asynchronous FIFOs involves the use of Gray code pointers. These pointers are synchronized to the opposite clock domain to produce

reliable full and empty status signals. The paper presents a method that enables dynamic adjustment of the FIFO's width during operation. The generation of full and empty flags is adapted to support these changes, ensuring the design avoids overflow and underflow conditions. Additionally, these modifications improve pipeline latency by allowing the read/write control block to temporarily halt FIFO operations, as shown in Fig. 5. This technique makes the asynchronous FIFO a modular design, providing a versatile interface for integration with other system-level components. It has been successfully implemented in applications like rate matching, reconfigurable hardware, and data streaming [10].

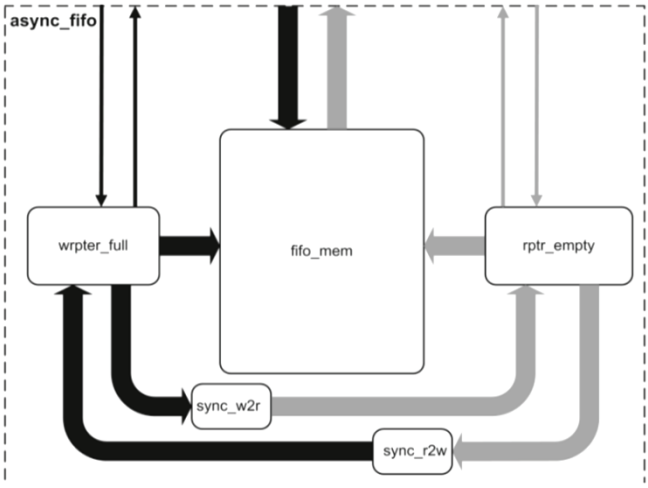

Fig. 5. System design [10]

3 Conclusion

Asynchronous FIFO plays a crucial role in modern high-speed data transmission systems, providing an effective solution for managing clock domain crossing issues and ensuring reliable and efficient communication between components operating at different clock frequencies. Its applications span various domains, including high-speed network interfaces, IoT systems, PCI Express, and ARM+FPGA architectures, demonstrating its versatility and adaptability. Through optimization techniques such as latency reduction, relative delay optimization, and modular designs, asynchronous FIFO achieves improved performance, reduced power consumption, and enhanced system reliability. The integration of reconfigurable FIFO designs further addresses the need for flexibility and scalability in complex systems, paving the way for use in emerging technologies such as 5G and AI-powered IoT applications. Overall, asynchronous FIFO remains a vital component in high-speed interface designs, with ongoing research and development focusing on further enhancing its efficiency, adaptability, and compatibility in increasingly demanding digital environments.

References

1. Sathaye, S., Ramakrishnan, K.K., Yang, H.: 'FIFO design for a high-speed network interface', Proc. 19th Conf. Local Computer Networks, Minneapolis, USA, October 1994, 1-8
2. Khan, M.A., Ansari, A.Q.: '128-Bit High-Speed FIFO Design for Network-on-Chip', Proc. Int. Conf. Information and Communication Technologies, Karachi, Pakistan, July 2011, 1-6
3. Keinert, J., Haubelt, C., Teich, J.: 'Synthesis of multi-dimensional high-speed FIFOs for out-of-order communication', Proc. Int. Conf. Architecture of Computing Systems, Dresden, Germany, February 2008, 75-86
4. Yan, Z.: 'Analysis and Design of Asynchronous FIFO', Proc. Int. Conf. Artificial Intelligence and Communication Technology, Singapore, January 2024, 1-8
5. Jones, L., Brown, D.: 'The title of the conference paper', Proc. Int. Conf. Systems Biology, Stockholm, Sweden, May 2006, 1-7
6. Gupta, S., Sharma, M., Chawla, R.: 'FPGA implementation of r-FIFO-based high-speed data acquisition IOT architecture model', SN Applied Sciences, 2020, 2, (4), 1-8
7. Wang, Y., Hu, Y., Fu, L.: 'Design and realization of multi-function serial port with high-speed/large-capacity/asynchronous FIFO', Proc. 12th IEEE Int. Conf. Control and Automation, Kathmandu, Nepal, June 2016, 879-884
8. Li, H., Zhang, X., Liu, Y., Chen, J.: 'A Wrapper of PCI Express with FIFO Interfaces based on FPGA', arXiv:1306.5501, 2013, 1-6
9. Ashour, H.: 'Design, simulation and realization of a parametrizable, configurable and modular asynchronous FIFO', Proc. Science and Information Conf., London, UK, July 2015, 449-455
10. Murali, M., Gupta, A., Durga, M.K.: 'A high-Speed Communication System is based on the Design of a Bi-NoC Router, which uses an Advanced FIFO Structure', Int. J. Advanced Research in Computer Science, 2018, 9, (2), 456-461