Proceedings of the 2nd International Conference on Functional Materials and Civil Engineering
DOI: 10.54254/2755-2721/70/20241030

A review of the design and research of asynchronous FIFOs

Baichuan Tong

Maynooth International School of Engineering, Fuzhou University, Fuzhou,350000, China

832201327@fzu.edu.cn

Abstract. Asynchronous FIFO (First In First Out) is a hardware structure for data transfer across clock domains. It is very important in the design of digital systems, especially when data needs to be exchanged between different modules or subsystems. The main function of an asynchronous FIFO is to solve the data transfer problem due to the unsynchronization of the clock domain, so as to ensure the integrity of the data and the stability of the system. Asynchronous FIFO is widely used in such fields as data acquisition systems, communication systems, and multiprocessor systems. It ensures the stability of data transmission between different modules and the overall system performance through reliable clock domain synchronization and effective read/write management. This paper provides a comprehensive review and analysis of the design principles, implementation methods, and applications of asynchronous FIFO memory. By examining the foundational concepts, various design techniques, and practical applications, this paper aims to offer valuable reference and guidance for digital system designers.

Keywords: Asynchronous FIFO, design principles, implementation methods, applications.

1. Introduction

A FIFO circuit is a storage device that enables FIFO of data and is often used as a data buffer. In digital system design, data transfer between different modules usually involves different clock domains. Since clock desynchronization may lead to data loss or misalignment, an efficient way to solve this problem is needed [1]. Asynchronous FIFOs, as a commonly used solution that enables secure data transfer and conversion between different clock domains, have important application value [2].

In China, the design and realization of asynchronous FIFO has been one of the important research directions in the field of digital system design. Asynchronous FIFOs have a wide range of applications in digital circuits and communication systems, such as in the fields of data buffering, data exchange, and data acquisition. Chinese research institutes and universities have made a lot of progress in the design and realization of asynchronous FIFOs.

Asynchronous FIFO design schemes in China are diverse, covering a wide range of aspects such as traditional asynchronous logic design, communication protocol-based design, hybrid synchronous asynchronous design, optimized design, and application-specific design. These design schemes play an important role in different application scenarios and provide rich technical support for the development of the digital system design field [3].

DOI: 10.54254/2755-2721/70/20241030

Foreign countries also have abundant scientific research progress in the design and realization of asynchronous FIFOs, covering a wide range of aspects such as high performance, low power consumption, security and reliability design, and application-specific design. These achievements not only promote the development of digital system design, but also provide important support for global scientific and technological innovation. Meanwhile, foreign countries have also adopted a variety of different design schemes in the design and realization of asynchronous FIFOs, including classical bistable design, Handshake protocol design, pipelined design, advanced asynchronous logic design, and design based on hardware description language, etc. These design schemes can be used in different application scenarios. These design schemes have their unique advantages and applicability in different application scenarios, providing a wealth of choices for the design and realization of asynchronous FIFOs.

This study aims to provide a comprehensive overview of the design principles, realization methods and their diverse applications of asynchronous FIFO technology. By systematically analyzing and discussing the fundamental theories of asynchronous FIFO and exploring its implementation strategies in various scenarios, this study seeks to provide researchers with in-depth theoretical foundations and practical guidance. By comprehensively analyzing the existing literature, this study also hopes to reveal the future development trend and potential research direction of asynchronous FIFO technology, which will provide a reference for subsequent research.

2. The use of context

2.1. Design principle of asynchronous FIFO

Asynchronous FIFO is a common data storage and transfer device used for data transfer between systems with different rates. The basic principle is to buffer and schedule data through storage units and control logic to ensure that the order of data remains the same at the input and output.

The design principle consists of three aspects. First, a suitable timing logic design is used to ensure that read and write operations can be synchronized between different clock domains. Second, the writing and reading of data are achieved through write pointers and read pointers, and synchronization and updating of these pointers are ensured [4]. In addition, storage units are used as buffers for data in order to store and transfer data efficiently. These design principles work together to ensure that the system remains efficient and reliable during data processing and transmission.

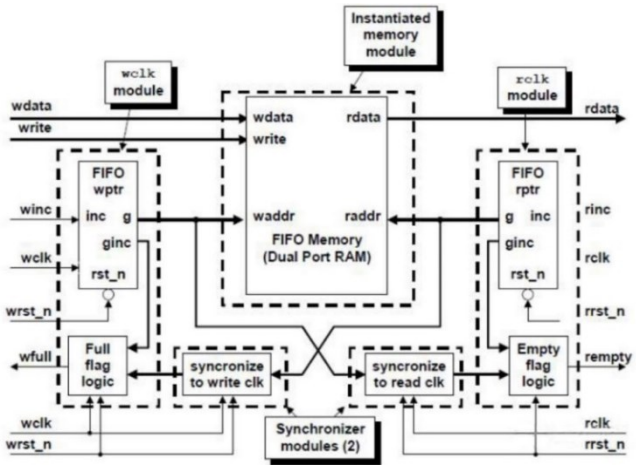

Figure 1. Block diagram of an asynchronous FIFO [4].

2.2. Asynchronous FIFO implementation method and process

The method of implementing asynchronous FIFOs consists of three main steps. First, the various clock domains involved in data transmission need to be divided and analyzed to determine the specific range of clock domains. Next, the read/write control logic is designed to ensure that data can be read/written

DOI: 10.54254/2755-2721/70/20241030

smoothly between different clock domains. Finally, the storage management module is designed, which is responsible for managing the storage and reading of data, including the functions of data writing, reading and pointer updating.

With these steps, the asynchronous FIFO implementation is able to efficiently transfer data across clock domains.

The realization process is more complex. First, a register transfer level (RTL) design is required. This step involves defining the structure, timing logic, and control logic of the FIFO. The designer needs to describe in detail the functions of writing, reading, memory management, pointer update to ensure reliable data transfer between different clock domains. Next, simulation and experimentation are performed. In this step, the RTL design is verified by simulation tools to ensure that the design is theoretically correct. The simulation process involves checking the performance of the FIFO in various situations such as full, empty, write, and read conflicts. In addition, practical experiments may also include testing on hardware platforms to verify the reliability and stability of the design in real-world applications [5]. Finally, synthesis and layout wiring is performed. The synthesis step converts the RTL design into a gate-level netlist suitable for a specific hardware implementation. The layout and wiring step places the synthesized design on the actual chip and arranges the circuit connections. This step needs to ensure that timing constraints are satisfied to ensure stable and reliable data transfer between different clock domains of the FIFO.

2.3. Asynchronous FIFOs in digital system design

Asynchronous FIFO has a wide range of applications in digital system design. First, it designs and implements data interfaces between different clock domains to ensure that data can be transferred smoothly between different clock domains, which is especially important for multi-clock domain systems because direct data transfer may lead to data errors or loss. Second, asynchronous FIFOs are used to convert data from one clock domain to another to ensure data integrity and reliability.

During clock domain conversion, asynchronous FIFO effectively solves the timing problem so that data can be stably transferred between different clock domains. In addition, the asynchronous FIFO serves as a data buffer for temporary storage and transmission of data to solve the problem of data loss and misalignment caused by clock desynchronization. In the case of uneven data traffic or the presence of bursty data transfer requirements, asynchronous FIFOs provide a reliable buffering mechanism to ensure the smooth operation of the system. These applications make asynchronous FIFOs a key component in dealing with the problem of data transmission between asynchronous clock domains and play an important role in the design of complex digital systems [6].

By collecting and analyzing data on the application of Asynchronous FIFO in the industry, we can provide a comprehensive understanding of its application in practice. The statistical analysis covers the distribution of asynchronous FIFO in different fields, design parameters, and performance data, providing an in-depth insight into the application status of asynchronous FIFO in various fields. Trend analysis of changes in the development history of asynchronous FIFOs over time reveals that as technology continues to evolve, so does the design and implementation technology of asynchronous FIFOs. For example, the use of more efficient storage units and further optimized control logic has resulted in superior performance [7].

With the wider application of digital systems, the application areas of asynchronous FIFOs are also expanding, covering a wide range of fields such as communications, embedded systems, image processing. With the increasing requirements for high data transfer rate and low latency, the performance of asynchronous FIFOs is also improving, including improvements in terms of increasing data transfer rate and reducing latency [8]. As a common data transmission device, asynchronous FIFOs play an important role in several industries, for example, in the communication industry, asynchronous FIFOs can perform data transmission between different rates and clock frequencies. Asynchronous FIFOs are often used within the computer industry to coordinate data transfer between different subsystems in a computer system. In the automotive industry, asynchronous FIFOs are used to coordinate data transfers between different electronic control units (ECUs) in automotive electronic systems.

DOI: 10.54254/2755-2721/70/20241030

2.4. Limitations of asynchronous FIFOs and directions for improvement

The performance of current asynchronous FIFOs may have bottlenecks, especially in high-speed data transfer and large-capacity storage. Traditional asynchronous FIFO designs may not be able to meet the demand for data transfer rate and storage capacity, limiting their application in high-performance systems.

In addition, the implementation of asynchronous FIFOs often requires a large number of logic gates and memory cells, resulting in large power consumption and chip area. This can be a limiting factor in some power-sensitive and area-constrained applications [9].

Improvements in asynchronous FIFOs include optimizing their design. These optimizations aim to increase their data transfer rate and storage capacity. This can be achieved by improving the structure of the memory cell. Another method is optimizing the control logic. Additionally, adopting more efficient data transfer schemes can also help. Researchers can also reduce the power consumption and chip area of asynchronous FIFOs by optimizing the circuit structure and adopting low-power design techniques to improve their applicability in power-sensitive and area-constrained applications [10].

3. Conclusion

This paper reviews the design and implementation of asynchronous FIFOs and analyzes their design principles, implementation methods, and applications. Through the review and analysis of asynchronous FIFOs, it is evident that asynchronous FIFOs play an important role in digital system design. They are widely used in related fields in real life. These applications make asynchronous FIFOs key components for dealing with data transmission problems between asynchronous clock domains. They can effectively solve timing problems. They also address data loss and misalignment issues.

However, asynchronous FIFOs are still suffering from performance bottlenecks and their power sensitivity and area constraints. These problems can be solved in the future by optimizing the design of asynchronous FIFOs to increase their data transfer rate and storage capacity, as well as by studying how to reduce the power consumption and chip area of asynchronous FIFOs to improve their applicability in power-sensitive and area-constrained applications.

References

[1] Si L Wu H Wang Y Sun F 2013 Design and realization of a large-capacity asynchronous FIFO Microelectronics vol 43 no 03 p 405-408
[2] Xie E Zhou J 2023 Analysis and Comparison of Asynchronous FIFO and Synchronous FIFO2023 Big Data and Algorithms p 260-264
[3] Li H Wang Q Yu Sh 2021 Asynchronous FIFO design based on Verilog HDL Electronic Design Engineering vol 29 no 19 p 107-111+116
[4] Chen T Lu F 2022 A delay-controllable asynchronous FIFO circuit design Microelectronics vol 52 no 01 p 42-46
[5] Ying Sh 2020 FPGA-based frame-level asynchronous FIFO design Acoustic and Electronic Engineering vol 2 p 32-34
[6] Li M Liu Y Peng Y 2023 Design of asynchronous FIFO module based on magnetically levitated knitting needle array Knitting industry vol 05 p 1-6
[7] Su G Zhang J Wu Y 2024 Design and implementation of Gigabit Ethernet communication module for motion controllers Instrumentation Technology and Sensors vol 2 p 86-92
[8] Li D Zhao Zh 2010 Design and implementation of a high-performance asynchronous FIFO Microelectronics Computer Editorial Department p 151-154
[9] Xu Q 2009 Design and implementation of high-speed asynchronous FIFO Microcomputer Information vol 25 no 20 p 153-155
[10] Patel V 2023 Design and Implementation of Novel Asynchronous FIFO International Symposium on Smart Electronic Systems p 292-295