Design and analysis of FIFO system based on basic circuit functions

Shengyu Liu

Xi'an Jiaotong University, Xi'an, China

syuliu@stu.xjtu.edu.cn

Abstract. Due to the insufficient data acquisition rate and high power consumption of sensors, this article focuses on addressing the issue of clock cycle interaction error resulting from the excessive amount of data on integrated circuit chips. Specifically, a FIFO design is proposed to achieve the transmission and transformation of data under different clock cycles. The technical challenges associated with creating an asynchronous FIFO, reducing the probability of encountering semi-stable states, and achieving delay control are analyzed in this paper. To tackle the semi-stable error, a Gray code converter and two-stage synchronizer are employed. The designed FIFO also leverages the difference and phase difference of read and write pointers to achieve high-accuracy delay control. The experiments demonstrate that the designed FIFO can successfully facilitate correct writing and reading operations. Through Modelsim simulation tests, the waveform is more precise than before, and the operation of the designed FIFO is realized.

Keywords: FIFO, asynchronous, semi-state.

1. Introduction

With the rapid development of microelectronics integration industry, there has been an explosion in algorithm improvement. Numerous high-precision chips have been integrated into hundreds or thousands of IP cores, allowing a single chip to perform multiple functions. However, the incorporation of IP cores also brings about issues of data interference [1] due to the interaction of clock domains between different data. To achieve lossless information transmission between data at two different clock frequencies, asynchronous system designs are becoming increasingly important. Engineers have employed the Tsmart-GalsBlock operational model, which can automatically generate a hybrid synchronous-asynchronous system to facilitate communication between modules in different clock domains. In the design of FIFO, engineers have realized a cross-clock domain processing and verification method for high-speed data by using dual-port SRAM as a base [3] and read-write pointers for judgment, which can be applied to both synchronous and asynchronous circuits [2]. The updated function of the courses to the next generation has resulted in the achievement of an adaptive multi-stage FIFO structure for cross-clock domain processing and sampling using the virtual channel mechanism [4]. The design method of asynchronous insertion and synchronous extraction has also facilitated cross-clock domain processing across data [5]. The use of FIFO with an empty full flag for verification and achieving communication between clock domains with configurable network on the chip [6] demonstrates significant progress in technology. It is evident that in the process of algorithm improvement, FIFO as an asynchronous device has played a key role in every iteration of technical development. Therefore, learners can better understand and realize circuit function integration only through in-depth learning and understanding of the working principle of FIFO.

2. Asynchronous FIFO system architecture

FIFO is a dual-port data cache device, and the data is first in, first out. Different from ordinary buffers, FIFO does not have external address lines. The advantage of this structure is that it reduces the input control line, which reduces the workload for later integration. However, the disadvantage is that all data can only be written and read out sequentially, and the flexibility of data is greatly limited. The diagram shows the internal structure of asynchronous FIFO, including the write data port, read out data port, write and read enabling port, read and write clock, empty and full signal, and final reset signal, RAM with dual ports, write and read controller, two beat trigger for the write address to the read clock domain, and read address to the write clock trigger. The part to check the read and write status. Write control module: the module produce the write address pointer under the combination of write enabling and write clock and reset signal. Read control module: the module produces the read address pointer under the combination of read enable and read clock and reset signal. Read and write clock synchronizer module: the read counter synchronizes to the write clock domain. The module compares the synchronized counter with the Gray code converted by the binary write counter to generate the (full) signal for marking the full write. Write reading clock synchronizer module: the write counter synchronizes to the read clock domain. The module compares the synchronized counter with the Gray code converted by the read binary counter to generate the (attribute) signal for marking the blank. Write state flag bit module: when the Gray code was judged to be full, the full (full) signal was generated by the write clock control. Read state flag bit module: when the Gray code was judged to be empty, the empty signal was generated by the read clock control. Delay judgment module: the module detects and corrects two data of the same phase due to objective factors in real time.

3. Key technical problems of FIFO

3.1. Cause of semi-stable state

In the digital circuit, to correctly generate data at the output port, it must meet the signal timing requirements defined by the register. Otherwise, the register cannot usually obtain the data at the input end to carry out the logical function operation. Under the premise of this definition, to ensure the reliable operation of the input signal, the input signal must have a stable period before the clock, commonly known as the establishment time, and maintain a period after the output. The trigger is effective after the rise of the clock. If the input signal cannot be established for a long enough time before the change and after the output, timing violations (the transistor in the trigger cannot be reliably fixed in the state of logic 0 or logic 1). Then the output generated by the trigger will enter an unsuitable condition, and the logical value of the output will change irregularly in high and low levels [7].

3.2. Solution of semi-stable state

The generation of metastable states is unavoidable in all asynchronous circuits. Still, the probability of generating metastable states can be reduced to a small and acceptable degree by changing the temporal structure of their data.

3.2.1. A two-level synchronizer. The most reliable synchronizer is one that allows the maximum amount of time for metastability resolution. The synchronizers are usually set to speed up the clock to get higher performance from the system. As a result, the designed synchronizers always work reliably with concise clock periods. Theoretical research suggests that experimental research has confirmed that when asynchronous input changes during the decision window, the duration of the metastable outputs is governed by an exponential formula [8]:

$$MTBF = \frac{exp(\frac{T_{res}}{\tau})}{T_w f_c f_d} \tag{1}$$

Therefore, if the duration of the metastable state in output can be reduced as much as possible under the condition that the input is not satisfied, a more reliable synchronizer can be constructed. Describing a couple of ways to build more reliable synchronizers, the first way is to use faster flip-flops, that is, to reduce the value of T in the MTBF equation. The second way is obvious to increase the value of τ, in the MTBF equation. The paper divides the system clock by n to obtain a slower synchronizer clock and longer τ = (n*(clock) - setup). Usually, a value of n = 2 or 3 gives adequate synchronizer reliability. However, the improvement brought by the three-stage synchronizer is not as significant as the change of the two-stage synchronizer, but the cost of the three-stage synchronizer is more expensive. Therefore, a two-stage synchronizer is selected as an essential module to reduce the metastable probability of FIFO.

3.2.2. Gray code counter. Gray code is a reliability code that can minimize the error generated by numerical changes. It can significantly reduce the error burr generated by the circuit when one state changes to the next state. This form of coding is to make two adjacent codes change only one bit, which significantly simplifies the error of operation change. Still, the disadvantage is that it takes a lot of memory, and is the unauthorized code, unable to be compared. So, the gray code must be converted into the decimal system for operation.

3.2.3. Improve clock quality with an edge-changing clock signal. The low-power consumption and high-precision clock with 32K-16 HZ on CMOS reduces the error by accurately controlling the speed of the clock [9] , which can ensure the area and power consumption.

3.2.4. Improve process accuracy. Under the influence of process error, the data of the same period will have an unavoidable delay under different transmission lines, affecting the engineer's judgment of the ideal establishment time, thus leading to the generation of a metastable state.

After the occurrence of a metastable state caused by the insufficient establishment and maintenance time when the data changes, after the two-stage synchronizer, the address pointer is encoded by Gray code, only one bit changed at a time, which can effectively reduce the probability of metastable state and increase the reliability of FIFO tremendously.

3.3. Judgment of empty and full state

Because FIFO is a memory that can only be read and written successively, to prevent FIFO from losing data due to the excessive data amount and covering the original data, FIFO needs to set the empty state and full state flag bit to prevent it. When marked with an empty state attribute = 1, FIFO cannot again read out the saved data. Otherwise, it will overwrite. When the full flag is full=1, FIFO can no longer write data to prevent errors.

Read empty conditions: the write pointer is caught up by the read pointer. Because read pointer and write pointer changes under different clocks, they cannot directly be compared, which must be converted to the same clock for comparison. So, the read pointer must convert the read clock to the write clock domain. When the read pointer and write pointer equal, attribute = 1, FIFO is empty.

Write full conditions:1) At one time, when the binary pointer is one more period than the binary pointer, full=1, and FIFO is full. 2) After multiple writing, only input a part of the data into FIFO at one time. After multiple writing, when the write pointer under the clock field only points to the n storage unit and the write pointer synchronized to the binary write pointer under the clock field means to the existing unit full=1 in the n + 1 storage unit, the FIFO is full. Compared conditions: Because the conversion module between the Gray code and the secondary system is a very occupied site, so when the circuit judges the empty full, it can use the method of sacrificing power consumption and delay to exchange for the superiority of the site. Empty state judgment: when all digits of the write pointer Gray code are the same as the read pointer, the judgment state is null. Full state judgment: when the read pointer and the write pointer, their Gray code's highest two are different, and the rest are the same. The judgment state is full.

3.4. Module of delay control

When the exciting system works, many chips will perform simultaneously. Due to the timing offset caused by the routing and layout of the clock, the data with the same phase will not be synchronized with the output data of different chips [10]. If there is a timing offset in the high-precision instrument, the error caused by this will be valuable. If the FIFO circuit can measure the delay time in real time and realize high precision control and correction to reduce the impact of some output timing offset on the whole circuit, the unnecessary differential circuits can be avoided to solve the problem. In this paper, the reading clocks and writing clocks are divided by the interpolation method. The larger the interpolation rate, the slower the clock. To simplify the correction step, the whole delay was divided into integer and decimal delays. The integer delay is the number of memory cells varying between the read pointer and the write pointer due to an error. In contrast, the decimal delay is the phase difference between the FIFO reading clock and the write clock in the same memory cell. By first adjusting the decimal delay and then adjusting the integer delay, the engineers can have higher accuracy in the measurement and modification circuit delay and better match the high-precision working chip. The adjustment delay is judged based on the previous state, so the real-time delay needs to be measured and calculated.

3.4.1. Decimal delayed. The decimal delay is mainly measuring the phase error between the read and the write clocks. The circuit obtains the phase error by sampling the writing clock, which is then output by the reading clock. The four-way sampling clock has the same clock frequency, and the phase successively shifts the clock period of 1 device clock [11]. Under this assumption, the phase difference between the reading clock and each sampling clock is inevitable. The phase difference between the reading clock and the writing clock can be analyzed and estimated based on the sampling results of the sampling clock and the writing clock. Then the circuit achieves the decimal update delay by changing the phase of the writing and reading clocks. The engineers observe the phase change of the reading and writing clocks to eliminate the phase difference between the reading clock and the writing clocks. However, if the phase change of the read and write clock is negative, the read and write pointer is not on a storage unit, so the clock phase difference of the reading clock is more significant than one writes clock. In this case, the integer delay must be updated to correct the delay accurately.

3.4.2. Integer delay. The integer delays real-time value of FIFO defined in this paper is the integer delay update that must be performed by the storage cell difference between the write pointer and the read pointer or the negative phase difference from the decimal delay. When adjusting the integer delay, if the decimal delay is not carried because the phase change is negative, the new read pointer for the current write pointer minus needs to update the integer delay value. If the decimal update carries, the latest read pointer should use the current write pointer minus the need to update the integer delay plus 1. After the update is completed, the engineers can verify the correctness of the circuit by comparing the difference between the current write pointer and the designed integer delay.

4. Circuit simulation and synthesis

Simulation of Asynchronous FIFO

FIFO is full for the first time, the following input data under the control of the read clock and read' enable can be seen that the input data is continuously updated from the graph, but after full, the data entering the ram register is no longer updated, full=1, and FIFO is full in Figure 1. In the write clock and write enable to control the next read out from FIFO, finally the data saved to ram will all read out, empty = 1, FIFO read empty in Figure 2.

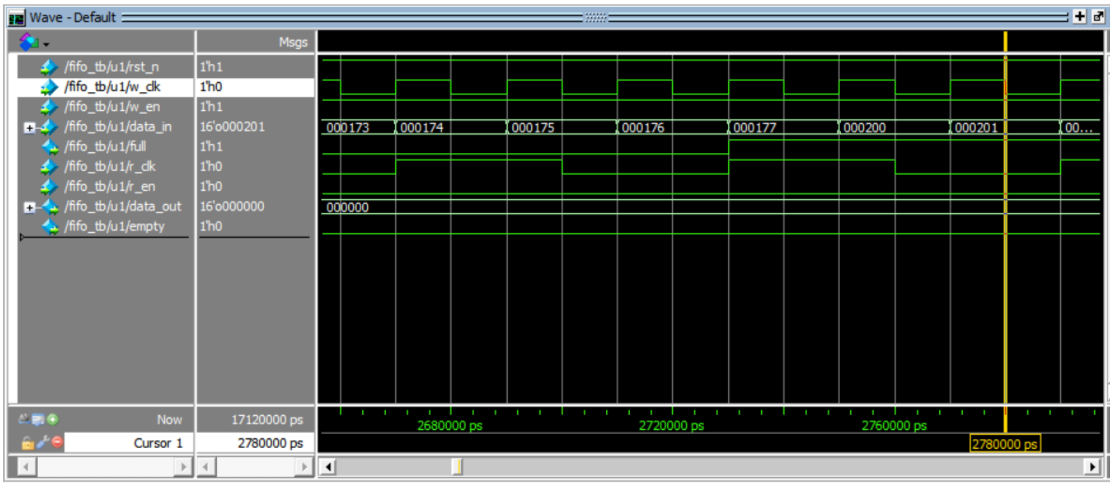

Figure 1. full=1.

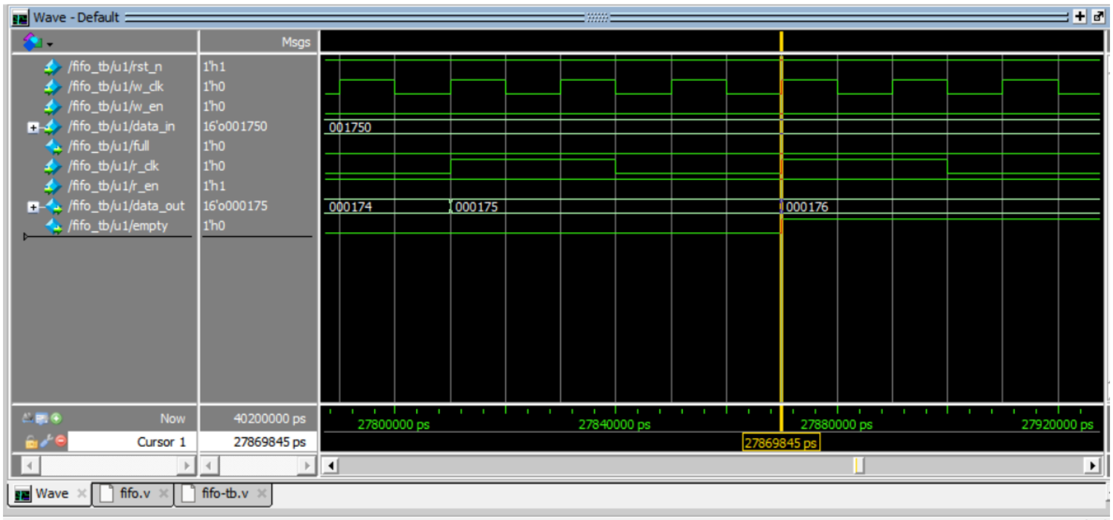

Figure 2. empty=1.

5. Conclusion

The generation of metastable states and the determination of full and empty state signals pose significant challenges in the design of asynchronous FFO memory. Asynchronous FFO memory is used to transfer data in two different clock domains, where the write address and read addresses are data that are not in the same clock domain. Therefore, synchronization of data is essential to effectively determine the status of FIFO. However, the binary number may change when the data is synchronized, and in cases where multi-digit data is metastable, the binary pointer cannot be checked for synchronization. The determination of the empty-full state signal is another challenge in the design of asynchronous FFO memory. When the read and write address pointers are equal, the FIFO memory could either be full or empty, making it challenging to determine its status accurately during chip operation due to the problem of output-input across the clock domain. This challenge can be effectively addressed through the design of synchronous FIFO, significantly improving the working efficiency of the chip while realizing the efficient use of the FIFO internal register. Additionally, the understanding of delay modules provides a unique idea for the circuit module of high-precision negative feedback. Asynchronous FIFO memory plays an essential role in the development of integrated circuits, and designing a superior performance FIFO is a difficult task for every circuit designer. The FIFO circuits need to be constantly updated and modified to keep up with the development of the IC industry. Although this design implements the basic requirements in function, there is still much room for improvement in reducing power consumption and area control and utilization. In subsequent designs, engineers can focus on the multithreading work of the FIFO chip, enabling simultaneous transmission of multiple pieces of data and improving the rate of chip operation. Moreover, as the FIFO delay constantly changes over time when a chip requiring high precision for data is working, the delay module should be designed to work again when a delay change is detected, or an interpolation rate should be generated, to improve the continuous working ability of the circuit and reduce unnecessary losses.

References

[1] Pan, Bo. Theoretical Study and Circuit Design of Ultra-low Metastable High-speed Pipelined-SAR ADC. University of Electronic Science and Technology, (2021).

[2] Yang, CX., Zhang, C., Wang, ZQ., Wang, ZJ., Xie, X., Jiang, HJ. Design of Elastic Buffer of PCS Layer Based on 10 Gbase-KR Protocol. Microelectronics and Computer 35(03),14-18(2018).

[3] Xu, Y., Lin, Y., Yang, HG. The FPGA Dual-port Memory Mapping Optimization Algorithm. Journal of Electronics and Informatics 42 (10),2549-2556(2020).

[4] Li, ZN., Li, JJ., Wang, AX., Zhong, SD. Design of High-speed Data Communication Interface Across Clock Domain. SCM and embedded system application 18(03), 13-18(2018).

[5] You, BX., Liao, YF., Ren, WL., Ma, J. FPGA in Data Rearrangements. Fire Control Radar Technology 50(03), 59-6(2021).

[6] Li, ZN., Li, JJ., Wang, J., Jin, SW. Asynchronous On-Chip Network Communication Mechanism for Dual Mode Fusion. Small Microcomputer system 40(10), 2241-2246(2019).

[7] Zhu, Yu., Dong, GT., Zhang, S. Research on The Metastable Processing Technology of FPGA Software. China Inspection and Testing 28(03), 14-17(2020).

[8] Tian, Y., Fan, YY., Li,ZW., Liu, WH., Xue, QN. Measurement of Metastable Parameters Based on FPGA. Microelectronics and Computer 33(10), 46-49(2016).

[9] Ding, H. Integration of A 32K-16 MHz Low-power, High-precision Clock Generator on A CMOS Chip. Huazhong University of Science and Technology, (2020).

[10] Chen, TT., Lu, F., Wan, SQ. et al. A Delay-controllable Asynchronous FIFO Circuit Design. Microelectronics 52(1), 42-46(2022).

[11] Lei, W., Li, JY. Design of A Broadband Signal Acquisition System Based on Alternating Sampling of Four-way ADC Chips. Electronics Technology 34(09), 30-35(2021).