Revolutionizing Digital Signal Filters: A Breakthrough in Low-Power Multipliers

Yixuan Jiang *

Department of integrated circuit design, Xidian University, Xian, China

* Corresponding Author Email: 21009102277@stu.xidian.edu.cn

Abstract. In the ever-evolving landscape of digital signal processing technology, the ubiquitous presence of digital signal filters has become a cornerstone across a myriad of domains including radar, computing, and communications. Amidst the relentless pursuit of more efficient solutions, a pivotal development emerges - the creation of a low-power multiplier tailored explicitly for digital signal filters. This innovation hinges upon the judicious selection of a carry-lookahead adder as a fundamental building block within the multiplier, ushering in enhancements that ripple through its operation. This transformative leap doesn't halt at the component level; it extends its reach further by delving into meticulous optimizations encompassing logic gate dimensions and set voltage parameters. These refinements, executed with precision, culminate in a substantial reduction in the energy consumption of the multiplier, effectively setting it apart from conventional counterparts. Remarkably, these advances don't merely remain theoretical; they find their footing in practical industry applications, thus birthing a new era in the creation of low-power digital signal filters. The profound implications of this development reverberate far and wide, notably benefiting the realm of low-power digital signal processing and its associated applications.

Keywords: Digital Signal Processing; Low-Power Multiplier; Energy Efficiency; Digital Signal Filters.

1. Introduction

With the development of computer technology, people have adopted digital signals as the foundation for computer calculations. However, in the early stages, due to the immaturity of computer technology, the development of digital signals was relatively slow. As breakthroughs were made in computer technology, the field of computing experienced rapid growth, and digital signals began to be widely applied in various fields [1]. Driven by the significant demand, the methods of digital signal processing gradually became more diverse. As a result, signal filters were developed to meet people's needs for digital signal processing.

The purpose of this paper is to design a low-power multiplier for digital signal filters in order to reduce the energy consumption required by digital signal filters. By comparing different options, this paper seeks to find a low-power adder as a constituent module of the multiplier. It also innovates the operation mode of the multiplier by introducing multiple additions achieved through the addition of adders, reducing the multiplication operation delay, and saving energy consumption. Finally, by adjusting the set voltage of the multiplier and the size of logic gates within the multiplier, further reduction in energy consumption of the multiplier is achieved. The designed multiplier in this paper can effectively reduce the energy consumption of digital signal filters, thus reducing energy consumption in digital signal processing. This has practical significance in terms of energy saving, cost control for businesses, and meeting the operational requirements of digital signal filters under specific conditions.

2. The relationship between digital signal filters and multipliers.

2.1. Application of digital signal filter

A digital signal is a signal that is represented by predefined states and is a type of discrete signal. It is typically represented using binary code, allowing the digital signal to control the state of a circuit (with 1 representing a high level and 0 representing a low level). Digital signals have advantages such

as strong immunity to interference, ease of confidentiality, high compatibility, and other advantages, making them highly promising for various applications.

In the current era, digital signals is advancing rapidly. As the primary medium for human communication, digital signals are widely used in various fields such as telecommunications, audio, images, radar, and more [2]. With the development of technology and the increasing demand for computers and artificial intelligence, there is a growing need for digital signal processing. As digital signals serve as the foundation for information storage and transmission, there are higher requirements for their processing capabilities.

Signal filters, as a processing method for digital signals, perform calculations on discrete input signals [3]. Their function is to allow the passage of most signal frequencies while blocking certain signal frequencies, thereby eliminating unnecessary signals [4]. Signal filters have the advantages of high precision, high reliability, and ease of reuse and integration [5]. They play a crucial role in addressing signal interference issues and are essential in the transmission process of signals.

Digital signal filters are based on Fourier transform and convolution calculations and can be divided into two main categories: classical filters and modern filters. Classical filters are used to remove interference signals at different frequencies when the desired signal and interference signal are at different frequency bands. They include high-pass filters, low-pass filters, band-pass filters, and band-stop filters. Different filters correspond to different filtering frequencies. However, when there is overlap between the desired signal and the interference signal, classical filters cannot effectively eliminate the interference signal, and modern filters are required. Modern filters estimate whether a signal is an interference signal using methods such as cross-correlation functions and auto-power spectra, and then exclude the identified interference signals [6]. The main types of modern filters currently used include Wiener filters and Kalman filters.

2.2. Working principle of classical digital signal filter

Convolution is a mathematical operation that combines two functions to produce a third function. It is essentially a special type of integral transformation that calculates the total sum of the overlapping areas between a function, after it has been flipped and shifted, and another function. The expression for convolution of discrete signals is:

$$x(k) * h(k) = \sum_{i=-\infty}^{\infty} x(i) \times h(n-i) \tag{1}$$

If $f_1(t) \leftrightarrow F_1(w)$ $f_2(t) \leftrightarrow F_2(w)$, F represents the Fourier transform off, then it has:

$$f_1(t) * f_2(t) \leftrightarrow F_1(w) \times F_2(w) \tag{2}$$

The Fourier transform states that any function that satisfies certain conditions can be expressed as a sum of trigonometric functions. This implies that the majority of signals can be decomposed into an infinite number of different frequency sinusoidal waves. This allows people to process signals in the frequency domain through Fourier transform, greatly enriching the methods of signal processing. In the context of discrete signal processing, the discrete Fourier transform formula is given by: (where n represents the index of the time-domain discrete signal, and m represents the index of the frequency-domain discrete signal)

$$X(m) = \sum_{n=0}^{N-1} x(n) \times e^{\frac{-2J\pi mn}{N}} \tag{3}$$

Converting a time-domain signal into a frequency-domain signal allows for signal transformation in the frequency domain. This forms the basis for implementing filters to selectively process signals based on their frequencies.

The key to the operation of a digital filter is to set a transformation function h(i). This transformation function takes different forms in different types of digital signal filters and is crucial in designing different digital signal filters. By set the input signal in the time domain with the transformation function h(i), we obtain the output signal.

$$y(i) = x(i) * h(i) \tag{4}$$

The transformation using the time-domain convolution theorem. Then we have:

$$y(w) = x(w) \times h(w) \tag{5}$$

The result y(w) obtained from equation (6) has reduced specific frequencies compared to x(w), thus achieving the effect of filtering. Since equation (6) involves multiplication, the functionality of digital filtering can be realized using a multiplier.

2.3. The relationship between the two

It can be observed that in classical digital signal filters, the implementation of their functions mainly relies on multiplication operations [7]. Therefore, the multiplier is an important component of classical digital signal filters and serves as the foundation for their implementation. In modern digital filters, although the algorithms differ significantly from classical digital signal filters, they still heavily rely on multiplication operations. Thus, the performance of the multiplier has a significant impact on the performance of digital signal filters. By optimizing the multiplier, the performance of the digital signal filter can be significantly improved. This article aims to reduce the power consumption of digital signal filter operation by designing a low-power multiplier [8].

3. Multiplier optimization

3.1. Adder selection

3.1.1 Full adder

The 1-bit full adder can realize the addition of two input signals, output the result and calculate whether there is a carry, its truth table is as follows in table 1.

Table 1. Three Scheme comparing

a	b	q	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Obtained from the truth table, and its logical expression is as follows:

$$q = a \oplus b \qquad c = a \times b \tag{6}$$

3.1.2 A multi-bit full adder

The full adder introduced earlier can only perform addition for two single-digit numbers. However, in practical applications, it is often necessary to calculate the sum of two multi-digit numbers. This requires a multi-digit full adder to achieve this functionality. The most direct design of a multi-digit full adder is to connect multiple full adders mentioned earlier in series. The figure 1 below shows the logic gate circuit diagram of a 4-bit full adder.

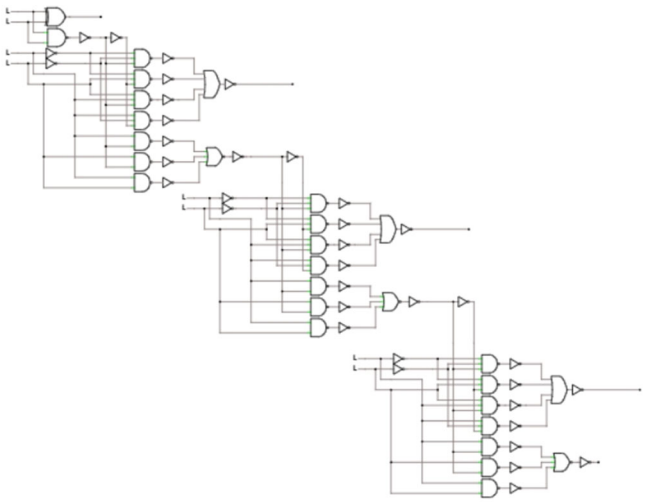


Fig. 1 The logic gate circuit diagram of a 4-bit full adder (Photo/Picture credit: Original)

The following is the calculation of important parameters for the critical path of a 4-bit carry lookahead adder (G represents the total logical effort of the critical path, g represents the logical effort of each gate circuit in the critical path. H represents the electric effort in the critical path, C represents the capacitance. In this case, the load capacitance of the 4-bit carry lookahead adder is 32F. B represents the total branch effort in the critical path, b represents the branch effort of each branch. F represents the path effort in the critical path. N represents the number of logic gates in the critical circuit. represents the stage effort in the critical path. represents the minimum delay of the critical path.):

$$G = \prod g = \frac{4}{3} \times 1 \times \frac{4}{3} \times 1 \times \frac{7}{3} \times 1 \times \frac{4}{3} \times 1 \times \frac{7}{3} \times 1 \times \frac{4}{3} \times 1 \times \frac{7}{3} \times 1 \approx 40.15 \quad (7)$$

$$H = \frac{C_{out}}{C_{in}} = \frac{32}{\frac{4}{3}} = 24 \quad (8)$$

$$B = \prod b = 4 \times 4.5 \times 4.5 \times 4.5 = 364.5 \quad (9)$$

$$F = G \times H \times B = 351232.2 \quad (10)$$

$$N = 14 \quad (11)$$

$$f^* = F^{\frac{1}{N}} = 2.490 \quad (12)$$

$$D_{min} = \sum p + N \times F = 58.86 \quad (13)$$

$$E = C \times V_{dd}^2 = 51.33 \quad (14)$$

3.1.3 carry-lookahead adder

The carry-lookahead adder is also derived from the full adder. When the inputs are more than one digit, the most direct way is to connect multiple full adders in series to achieve multi-digit addition. However, this leads to inefficiency in operation. The principle of the carry-lookahead adder is to compute and directly list the logical expressions between each carry and input, thereby avoiding the situation where the carry from the higher positions of the traditional full adder needs to wait for the completion of the lower positions' computation before it can proceed. This significantly improves the efficiency of the adder's operation and reduces its power consumption [9]. The logical expression of the carry-lookahead adder is as follows (where Ci represents the carry from the i-th position, and Si represents the sum of the two input numbers at the i-th position, counting from the least significant bit to the most significant bit)

$$G_i = a_i \times b_i \quad (15)$$

$$P_i = a_i \oplus b_i \quad (16)$$

$$C_1 = G_0 + P_0 \times C_0 \tag{17}$$

$$C_2 = G_1 + P_1 \times G_0 + P_1 \times P_0 \times C_0 \tag{18}$$

$$C_n = G_{n-1} + P_{n-1} \times G_{n-2} + P_{n-1} \times P_{n-2} \times G_{n-3} + \ldots + P_{n-1} \times P_{n-2} \times \ldots \times P_1 \times C_0 \tag{19}$$

The logic gate circuit diagram for a 4-bit carry-lookahead adder, based on the given logical expressions, is shown below. The critical path of this circuit is also marked in the figure 2.

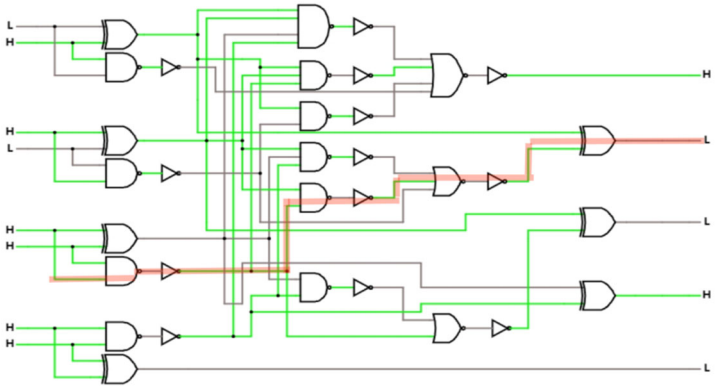


Fig. 2 The critical path of this circuit (Photo/Picture credit: Original)

The important parameters for calculating the critical path of a 4-bit carry-lookahead adder are as follows:

$$G = \prod g = \frac{4}{3} \times 1 \times \frac{4}{3} \times 1 \times \frac{5}{3} \times 1 \times 4 \approx 11.85 \tag{20}$$

$$H = \frac{C_{out}}{C_{in}} = \frac{32}{\frac{4}{3}} = 24 \tag{21}$$

$$B = \prod b = 3.25 \tag{22}$$

$$F = G \times H \times B = 924.3 \tag{23}$$

$$N = 7 \tag{24}$$

$$f^* = F^{\frac{1}{N}} = 2.653 \tag{25}$$

$$D_{min} = \sum p + N \times F = 32.571 \tag{26}$$

$$E = C \times V_{dd}^2 = 43.33 \tag{27}$$

Based on the parameters, compared to traditional multi-digit full adders, the carry-lookahead adder has a significantly reduced critical path delay and reduced energy consumption under the same conditions. Due to its obvious advantage in delay, with further optimization of the size and set voltage of the carry-lookahead adder, its energy efficiency advantage will become even more apparent. Therefore, the carry-lookahead adder is selected as the addition component of the low-power multiplier.

3.2. Algorithm optimization

The operational flowchart of a traditional multiplier is as follows in figure 3.

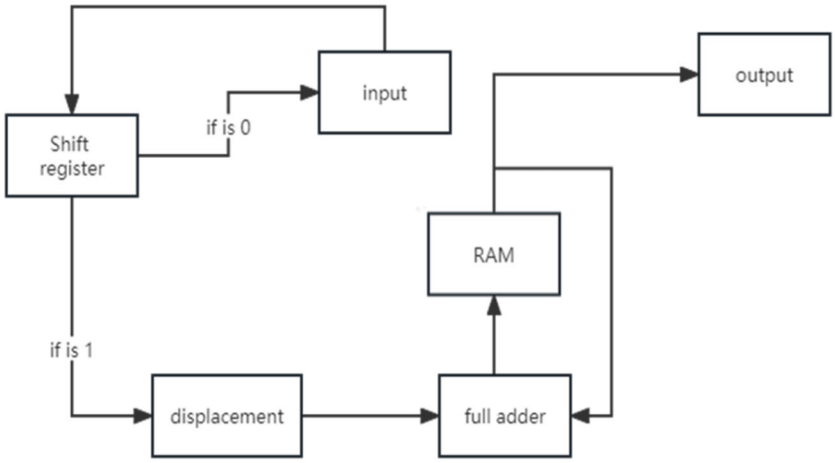


Fig. 3 The operational flowchart (Photo/Picture credit: Original)

It can be observed that the operational flowchart of a traditional multiplier involves decomposing one of the multiplicands and evaluating each bit individually. If the bit is 1, the other multiplicand is shifted i bits to the left (where i represents the position of the bit), added to the previous result, and the process is repeated for the next bit. If the bit is 0, the next bit is evaluated directly. This process is repeated for multiple cycles to achieve multiplication of the two numbers. This results in significant delay and high-power consumption [10]. In order to reduce the power consumption of the multiplier, this paper proposes an improvement in the operational flow by increasing the number of adders and shift registers. This allows for the evaluation of four bits in a single cycle, to some extent reducing the delay of the multiplier. Additionally, by reducing the number of repeated operations caused by looping in certain stages, there is a decrease in energy consumption. However, this improvement also leads to an increase in area requirements. The specific process is as follow in figure 4.

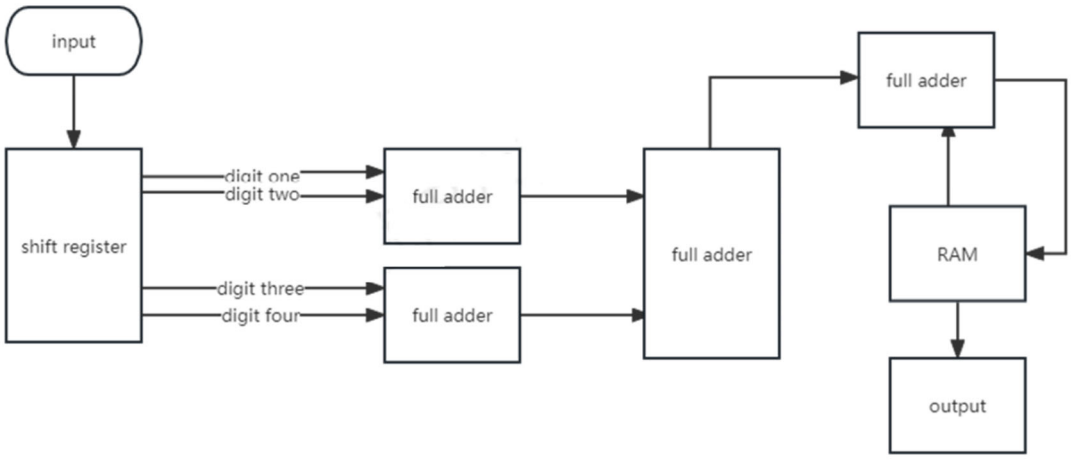


Fig. 4 The specific process (Photo/Picture credit: Original)

Compared to the traditional multiplier, the operational method used in this paper significantly reduces the number of operations of individual shift registers. Therefore, in comparison to the traditional multiplier, it achieves a reduction in energy consumption on the critical path. With a load resistance of 32F and =1, the calculation of the key parameters is as follows:

$$G = \prod g \approx 269964327303894.211 \tag{28}$$

$$H = \frac{C_{out}}{C_{in}} = \frac{32}{\frac{4}{3}} = 24 \tag{29}$$

$$B = \prod b = 4116773.808 \tag{30}$$

$$F = G \times H \times B = 2667316972173626266194 3.012 \tag{31}$$

$$N = 38 \tag{32}$$

$$f^* = F^{\frac{1}{N}} = 3.892 \tag{33}$$

$$D_{min} = \sum p + N \times F = 356.896 \tag{34}$$

$$E = C \times V_{dd}^2 = 167.33 \tag{35}$$

3.3. Global optimization

3.3.1 Change V_{dd} only

Due to the formula

$$E = C \times V_{dd}^2 \tag{36}$$

It can be observed that in order to reduce energy consumption, it can be achieved by adjusting the size, as well as by changing the capacitance of logic gates in the circuit. Changing the capacitance of logic gates can be achieved by altering their size. Therefore, this article mainly aims to achieve further energy consumption reduction by adjusting the size of the adder logic gates in the multiplier circuit.

And due to the formula

$$D = \frac{C \times V_{dd}}{k \times (V_{dd} - V_T)^2} \tag{37}$$

It can be observed that as energy consumption decreases, the delay of the circuit's critical path increases. Therefore, this article aims to ensure that.

$$D = 1.5 \times D_{min} \tag{38}$$

While the circuit delay meets the requirement of being within a reasonable range, also ensuring a significant reduction in circuit energy consumption. Based on the calculations, it can be determined that the above conditions are met. At that time, the critical path delay of the multiplier is 530.3035, and the energy consumption of the multiplier is 129.68.

3.3.2 Change size only

Changing the size of a logic gate implies changing the capacitance of the logic gate. So, we can get the formula.

$$E = size \times V_{dd}^2 \tag{39}$$

For the convenience of calculation, in this text, only the size of the last logic gate in the circuit is varied to satisfy the circuit. The sizes of the other logic gates are all 1. Through calculation, it can be determined that when the size of the last logic gate is [insert the specific value], the delay of the multiplier is 530.3035, and the delay of the critical path is 116.15.

4. Conclusion

This article utilizes a carry-lookahead adder as a component of the multiplier through a comparative approach. By increasing the number of adders, the multiplier's computation is optimized. The designed multiplier has a minimum delay of 356.896 on the critical path and a power consumption of 167.33 when the size is 1 and the load capacitance is 32F. When only the optimization of the adders is performed, the minimum delay on the critical path is 530.3035, and the power consumption remains at 129.68. Similarly, when only the optimization of the logic gate size is performed, the minimum delay on the critical path is 530.3035, and the power consumption remains at 116.15.

In addition to the methods used in this article, there are other ways to further reduce the power consumption of multipliers used in digital signal filters. For example, changing the ratio of the length and width of MOS transistors in logic gates, adopting more advanced algorithms to optimize

multiplier operations, and considering the use of new semiconductor compounds or memristors as new basic components. These approaches have shown significant effects in reducing the power consumption of multipliers.

References

[1] Zhao Zixi.Based on Matlab, the signal analysis and digital filter design. Computer Knowledge and Technology, 2021, 17(29): 114-115+119.

[2] Zeng Hui.Application research of digital signal processing technology in electronic information engineering. Electronic Quality, 2022(09): 81-84.

[3] Zhang Xixi, Lu Jicheng, Li Xiayu et al.Optimized structure of a special digital filter. Microelectronics, 2018, 48(04): 429-432.

[4] Zhang Defeng, Comprehensive Introduction to MATLAB Digital Image Processing. Publisher: Electronic Industry Press, Beijing, China, 2010.

[5] Zhang Yawei, Shi Qiangqiang, Yao Fenggang, Configurable Digital Filter Design based on FPGA. Foreign Electronic Measurement Technology, 2019, 38(10): 102-108.

[6] Li Feixiong, Jiang Lin, Design of a Novel Pipeline Booth Multiplier Structure. Electronic Science and Technology, 2013, 26(08): 46-48+67.

[7] Zhang Mingying, Design of a 32-bit Low-power High-speed Multiplier. Microprocessors, 2016, 37(01): 18-21.

[8] Li Yanping, Wang Wenjun, Design of 16-bit Improved Conditional Carry Adder Circuit. Ordnance Industry Automation, 2020, 39(11): 44-47.

[9] Zhou Ranran, Zhou Wencheng, Wang Yong, et al. Design of a Carry-Ahead Adder Based on 55nm Process. Chinese Journal of Integrated Circuits, 2022.

[10] Liu Haifeng, Wang Fang, wang wei, et al. Design of Logic Gates and Two-bit Toffoli Multiplier based on Memristors. Microelectronics and Computer Science, 2018, 35(10): 46-52.