Dean&Francis

ISSN 2959-6157

Advancements and Applications of Hardware Adders: Principles, Functions, and Future Challenges

Ruoqi Wu

School of Physics, Huazhong University of Science and Technology, Wuhan city, China

u202210169@hust.edu.cn

Abstract:

This paper reviews the principles and applications of various types of adders, including Half Adder, Full Adder, Ripple-Carry Adder (RCA), Lookahead Carry Adder (LCA), Carry-Bypass Adder (CBA), and Carry Select Adder (CSA). The basic logic of each adder type is analyzed, emphasizing their performance characteristics, such as area, power consumption, and delay. Applications of adders in fields like image processing, portable electronics, and medical data encryption are discussed, highlighting how different types of adders are tailored to specific performance requirements. Approximate Full Adders (AFA) and low-power, high-speed adders are introduced as solutions for energy-efficient image processing and portable electronic applications, respectively. Challenges in reducing power consumption and improving approximate computing accuracy are addressed, with a focus on novel design approaches such as powerless AND gates, groundless OR gates, and error detection and correction methods. Future expectations for adder optimization are explored, including trends in power reduction, computing efficiency, and improved performance for various application scenarios.

Keywords: Half adder; full adder; carry-bypass adder; carry select adder.

1. Introduction

Adders are fundamental components in digital electronics, widely used in computing systems to perform arithmetic operations, particularly addition, which is essential for tasks involving subtraction, multiplication, and division when converted into addition in binary systems. The core function of adders lies in computing sums, and they are implemented in a variety of logic circuits made of NMOS and PMOS transistors. As the complexity of digital circuits grows, optimizing the performance of adders in terms of power consumption, area, and delay becomes increasingly critical. This has led to the development of various types of adders, each designed to meet specific performance requirements [1][2]. In

recent years, advancements in electronic engineering have introduced several types of adders such as the Half Adder, Full Adder, Ripple-Carry Adder (RCA), Lookahead Carry Adder (LCA), Carry-Bypass Adder (CBA), and Carry Select Adder (CSA).

Each adder has unique benefits and trade-offs, depending on factors like speed, power efficiency, and transistor count. Approximate Full Adders (AFA) have also gained attention for their use in applications where computational precision can be compromised for power savings, such as image processing and portable electronics [3][4]. These advancements highlight the continuous effort to strike a balance between accuracy, efficiency, and performance across various domains.

This paper provides a comprehensive review of the principles and functions of different types of adders, along with their applications in fields such as image processing, portable electronics, and medical data encryption. Additionally, it explores the current challenges facing hardware adders, particularly in reducing power consumption and enhancing approximate computing. Future directions in adder design, including logic optimization and hardware improvements, are discussed, emphasizing the need for innovations that address the growing demands of modern integrated circuits [5][6].

2. Basic Principles of Adders

2.1 Half Adder, Full Adder, and Ripple-Carry Adder (RCA)

2.1.1 Half adder

As show in the Fig. 1 and table 1. Here is the logic expression of the half adder: $s = a?b, c = ab$.

Table 1. The truth table of a half adder.

input		output	
a	*b*	*s*	*c*
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

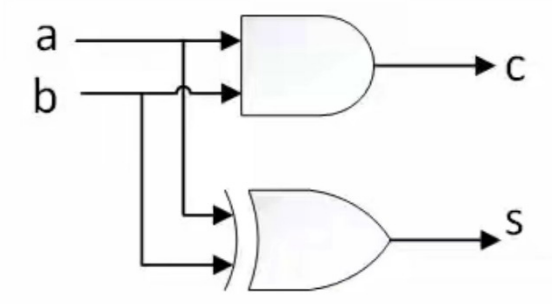

Fig. 1 The circuit diagram of a half adder (Photo credit: Original).

2.1.2 Full adder

Table 2. The truth table of a full adder.

input			output	
a	*b*	c_{in}	*s*	c_{out}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1

ISSN 2959-6157

0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

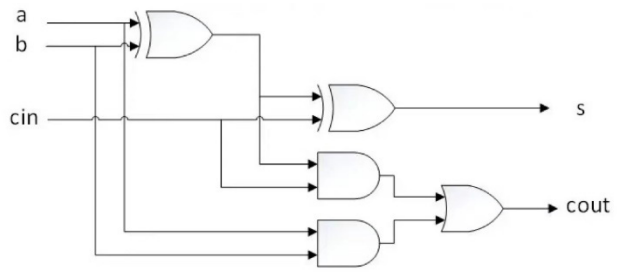

Fig. 2 The circuit diagram of a full adder (Photo credit: Original).

As show in the Fig. 2 and table 2.

2.1.3 Ripple-Carry Adder (RCA)

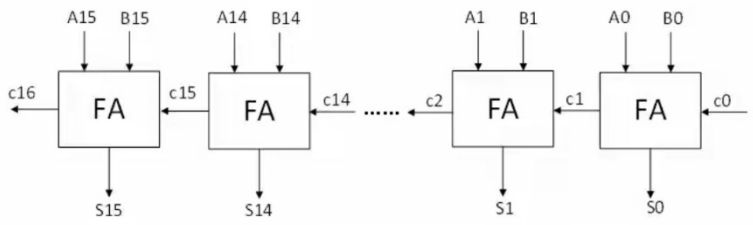

Fig. 3 The structure diagram of a 16-bit RCA (Photo credit: Original).

As show in the Fig. 3. As can be seen from Fig. 3, the calculation of each bit depends on the carry result of the previous bit, leading to the fact that this kind of adder has a huge delay. Even so, the logic in this adder is simple, so people sometimes still use it when there aren't too many digits.

2.2 Lookahead Carry Adder (LCA)

The main idea of a LCA is parallel computing. This will greatly reduce the delay generated in the RCA, because the computation of a bit doesn't need to wait for the result of the previous bit.

There are definitions of P_i and G_i:

$\begin{cases} P_i = a_i \oplus b_i \\ G_i = a_i b_i \end{cases} i = 0,1,\ldots,N-1$, where i is the digit order number. Then, the logic expression can be written like:

$\begin{cases} s_i = P_i \oplus c_{i-1} \\ c_i = G_i + c_{i-1} P_i \\ c_0 = c_m \\ c_{out} = c_N \end{cases}$. The idea is to compute the P_i and G_i of each bit at the same time, and then compute the s_i and c_i using all the available P and G.

Take a 4-bit LCA as an example. The logic expression is

$$\begin{cases} c_1 = G_0 + c_0 P_0 \\ c_2 = G_1 + G_0 P_1 + c_0 P_1 P_0 \\ c_3 = G_2 + G_1 P_2 + G_0 P_2 P_1 + c_0 P_2 P_1 P_0 \\ c_4 = G_3 + G_2 P_3 + G_1 P_3 P_2 + G_0 P_3 P_2 P_1 + c_0 P_3 P_2 P_1 P_0 \end{cases}$$

Its structure diagram is shown in Fig. 4.

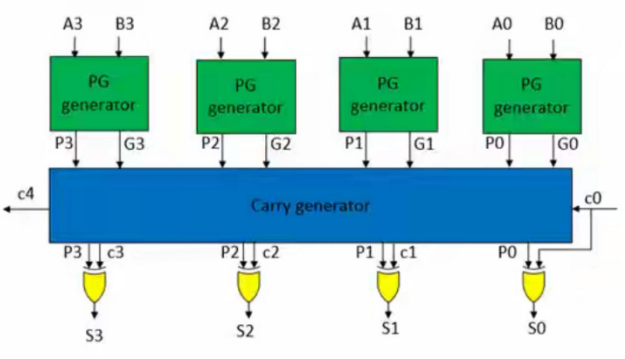

Fig. 4 The structure diagram of a 4-bit LCA (Photo credit: Original).

As show in the Fig. 4. LCA has a higher computing speed

than the RCA, owing to the parallel computing [1]. However, LCA apparently has a more complicated logic, leading to larger fan-in and fan-out numbers. Also, it will take up more area than RCA.

2.3 Carry-Bypass Adder (CBA)

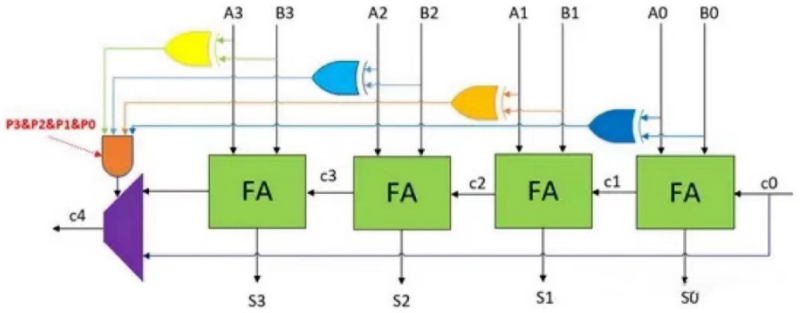

Fig. 5 The structure diagram of a 4-bit CBA (Photo credit: Original).

As show in the Fig. 5. The definition of P and G mentioned in 2.3 still work here. According to Fig. 5, when $P_1P_2P_3P_4=1$, the selector (labeled purple) will directly send c_0 to c_4. It is true in terms of logic: $P_1P_2P_3P_4=1$ means that $P_1=P_2=P_3=P_4=1$, so $c_4 = G_3+G_2P_3+G_1P_3P_2+G_0P_3P_2P_1+c_0P_3P_2P_1P_0 = G_3+G_2++G_1+G_0+c_0$. Noting that when $P=1$, $G=0$, then $c_4=c_0$.

CBA uses a bypass logic to skip the computation of the carry in each bit, thus it is also called Carry Skip Adder.

2.4 Carry Select Adder (CSA)

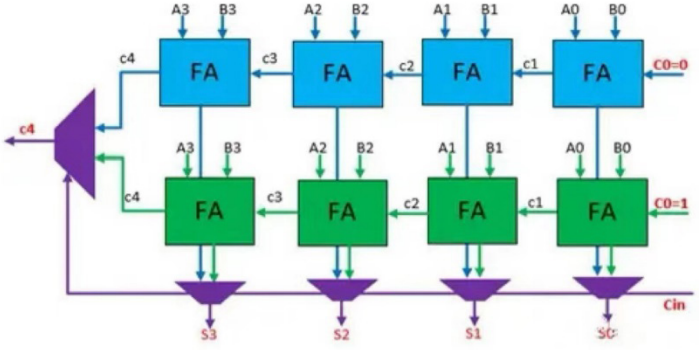

Fig. 6 The structure diagram of a 4-bit CSA (Photo credit: Original).

As show in the Fig. 6. When the c_{in} comes in, the selector (labeled purple) will choose the result from the blue RCA or the green RCA based on the real number of c_{in}.

This is a typical way to trade area for computing speed. Because this method reduces the carry propagation delay compared to RCA [2]. CSA also requires double the number of transistors used in RCA. Therefore, if the requirement of computing speed is taxing and the area and transistors consumption are not a problem, then the CSA will be a good choice.

3. Functions Implemented by Adders

3.1 Image Processing

The idea is to give every pixel an 8-bit binary unsigned integer, then compute the sum of the corresponding pixels from two images using adders. Since this circuit is not designed for accurate computation, accuracy could be given up for lower power consumption and area. If the error rate is not too high, then this method can work in image process. Thus, people made some improvements on the full adder. And this new adder is called Approximate Full Adder (AFA).

As show in the Fig. 7 and table 3. Take AFA1 (one type of AFA) as an example. AFA1 discards all the XOR gates in the full adder and replace them with OR gates. Its circuit diagram is shown in Fig. 7 and its truth table is shown in Table 3.

ISSN 2959-6157

Table 3. The truth table of a AFA1.

input			output	
a	b	c_{in}	s	c_{out}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	1	0
0	0	1	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	1	1

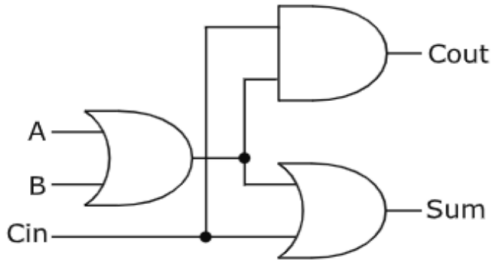

Fig. 7 The circuit diagram of a AFA1 (Photo credit: Original).

The error result is labeled in red. The error rate for s is 37.5% and for c_{out} is 12.5%. The performance test result comes out that the delay of AFA1 is $213ps$ and the area is $36.59\mu m^2$, both of which are about the half of a full adder.

Since there are lots of logic combination could be used to approximately compute addition, the type of AFA is various, and the performance between different AFA is also different. See details in M. C. Parameshwara's work [3].

3.2 Portable Electronic

Some electronic products need to be made small due to the limitations of application scenarios. This places demanding requirements on the design of the adder: it needs to meet both small area and other good performance. The number of transistors used in different types of LPHS-FA ranges from 8 to 28, and their performance varies. The author specifically introduces on type of LPHS-FA in this article.

As show in the Fig. 8. Its circuit diagram is shown in Fig. 8.

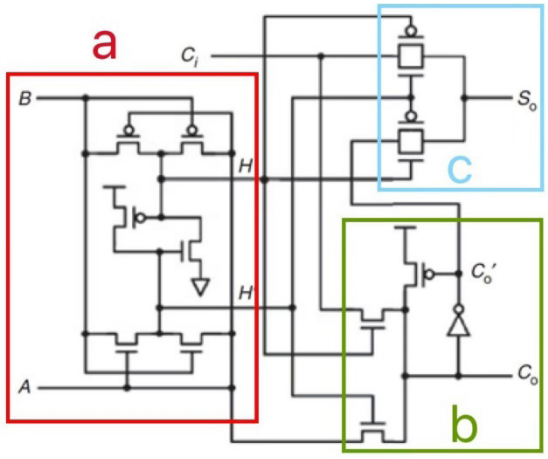

Fig. 8 The circuit diagram of a LPHS-FA (Photo credit: Original).

The circuit is consisted three parts (part a labeled red, part b labeled green, part c labeled blue). Part a is a XOR-XNOR module, which generates H and H'. In part b, H and H' work as enable signals. To fix the problem that the signal is weakened when passing the NMOS, a pull-up is placed after the NMOS. The pull-up is a combination of a PMOS and an inverser. When the output is 1, then the PMOS is activated and the V_{DD} becomes the output. Part c uses transmission gates to compute the sum.

The overall circuit only uses 15 transistors and the delay is $0.179ns$ [4]. This work has largely reduced the transistor consumption and delay in contrast to the previous adder design.

3.3 Medical Data Encryption

With the development of internet, online medical treatment is getting more and more popular. In order to protect patients' privacy, medical image data needs to be encrypted during transmission. Weighted shift approximate ad-

ders (WSAA) are widely used in this encryption.
The principle of WSAA is as follow:
Step1. Shift the input A, which is the input pixel cyclically to the right. The number of shift bits is equal to the Hamming weight of A.
Step2. Shift the key B one bit to the left.
Step3.Treat A and B as addends, and the shifted B as a carry. The sum is the encrypted data.
It turns out that this encryption method is safe against statistical attack, entropy attack and differential attack [5].

4. Challenges and Future Expectations

4.1 Reducing Power Consumption

With the increase of the scale of integrated circuits, the problem of power consumption is becoming a real challenge. Generally, attempts should be made in two ways: one is to create better performance transistors, which can be left for semiconductor physicists; the another is to redesign the logic gates and logic circuits with different combinations and orders of transistors. Here is a successful example.

4.1.1 Powerless AND gate and groundless OR gate

The circuit of powerless AND gate is shown in Fig. 9.

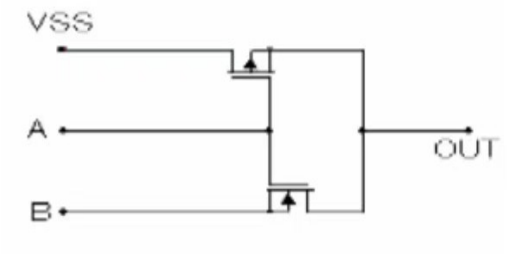

Fig. 9 The circuit diagram of a powerless AND gate (Photo credit: Original).

As show in the Fig. 9. The powerless AND gate uses a PMOS and a NMOS. When $A=0$, the PMOS is activated, leading to the output to be 0. When $A=1$, the NOMS is activated, leading to the output depending on B. This means that only when $A=B=1$, the output is 1, which is exactly the logic of AND. This circuit realize it without a power supply, making it economical in power consumption.
The circuit of groundless OR gate is shown in Fig. 10.

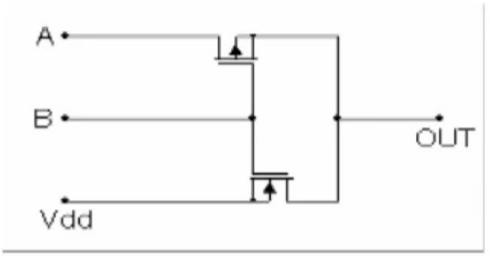

Fig. 10 The circuit diagram of a groundless OR gate (Photo credit: Original).

As show in the Fig. 10. The groundless OR gate uses a PMOS and a NMOS. When $B=1$, the NMOS is activated, leading to the output to be 1. When $B=0$, the PMOS is activated, leading to the output depending on A. This means that only when $A=B=0$, the output is 0, which is exactly the logic of OR. This circuit realize it without ground, also making it economical in power consumption.

4.1.2 New Carry Look-ahead Adder (NCLA)

In the NCLA, all the AND and OR gates adopt the design of powerless AND gate and groundless OR gate. The total number of transistors used in MCLA is 136, and the total number of transistors used in NCLA is 74 [6]. Compared with the previously proposed MCLA, NCLA has significant advantages in area, power consumption, and speed.

4.2 Approximate Computing

Trading computing accuracy for computing efficiency has already become a trend for hardware design. However, the error rate of the existing AFA is less than satisfactory. Thus, how to balance the performance and accuracy becomes a challenge in approximate computing. Efforts should be put in the following aspects to overcome the challenge:
Error Distance: By utilizing the concept of carrier life time, the probability of erroneous carry propagation to the most significant bit can be reduced by treating the high bit adder as the low bit adder [7].
Error rate: Introduce error detection and correction logic to reduce error rates. Find the specific logic that generates the error, mark the error carry position, and then invert it [8].
Empact assessment: Using software algorithm- significance-driven approach, recognize the elastic computing [9] [10].

ISSN 2959-6157

5. Conclusion

In this paper, a comprehensive review of various types of adders, including Half Adder, Full Adder, Ripple-Carry Adder (RCA), Lookahead Carry Adder (LCA), Carry-Bypass Adder (CBA), and Carry Select Adder (CSA), has been presented. The principles and logic behind each type of adder were analyzed, highlighting their strengths and trade-offs in terms of speed, area, and power consumption. Additionally, applications of adders in areas such as image processing, portable electronics, and medical data encryption were explored, demonstrating how different adder designs cater to specific performance needs. The challenges related to reducing power consumption and improving the efficiency of approximate computing were also discussed, along with innovative approaches like powerless AND gates and groundless OR gates. Looking ahead, future research should focus on developing more efficient adder designs that address the growing complexity of modern integrated circuits. This includes further reducing power consumption, minimizing delay, and improving the accuracy of approximate computing. Additionally, the integration of novel materials and the continued optimization of transistor technology will play a crucial role in enhancing the performance of adders in advanced applications. The evolution of adder designs will be vital for meeting the demands of future technologies, such as artificial intelligence, IoT devices, and energy-efficient computing systems.

6. References

[1] Raahemifar K., Ahmadi M. Fast carry-look-ahead adder. Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411), Edmonton, AB, Canada, 1999, 529-532 (vol. 1).

[2] Bedrij O. J. Carry-select adder. Electronic Computers IRE Transactions on, 1962, EC-11.

[3] Parameshwara M. C. Approximate full adders for energy-efficient image processing applications. Journal of Circuits, Systems and Computers, 2021, 30(13).

[4] Zhang Y., Xu H., Zhu X., et al. Detection and quantization technique of optical distributed acoustic coupling based on φ-OTDR. Journal of Shanghai Jiaotong University (Science), 2020, 25: 208–213.

[5] Tung C.-K., Shieh S.-H., Cheng C.-H. Low-power high-speed full adder for portable electronic applications. Electronics Letters, 2013, 49(17): 1063-1064.

[6] Manikandan, Nagarajan, Muthaiah, Teekaraman, Yuvaraja, Kuppusamy, Ramya, Radhakrishnan, Arun. A novel random error approximate adder-based lightweight medical image encryption scheme for secure remote monitoring of health data. Security and Communication Networks, 2021, 3570904, 14 pages.

[7] Thapliyal H., Arabnia H. R., Srinivas M. B. Reduced area low power high throughput BCD adders for IEEE 754r format. arXiv:cs/0609036, 2006.

[8] Dutt S., Nandi S., Trivedi G. Analysis and design of adders for approximate computing. ACM Transactions on Embedded Computing Systems, 2017, 17(2).

[9] Venkataramani S., Chakradhar S. T., Roy K., Raghunathan A. Approximate computing and the quest for computing efficiency. DAC ,15: Proceedings of the 52nd Annual Design Automation Conference, 2015, 120: 1-6.

[10] Xu H., Zhu X., Zhao Z., Wei X., Wang X., Zuo J. Research of pipeline leak detection technology and application prospect of petrochemical wharf. 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 2020, 263-271.