Explanation of the principles of the Booth algorithm, Verilog implementation and simulation

Wangye Jiang

Changshu Institute of Technology, Changshu Institute of Technology, Changshu, 215505, China

Corresponding author: Z05122208@cslg.edu.cn

Abstract. In an era marked by heightened constraints on computer processing speed and storage capacity, optimizing computational efficiency and space utilization assumes paramount importance. Consequently, the deployment of proficient algorithmic tools holds significant significance within the realm of computing. The principal categories of multipliers encompass conventional multipliers, shift-and-add multipliers, Look-Up Table (LUT) multipliers, and Booth algorithm multipliers. Among these, the Booth algorithm substantially influences multiplier performance enhancement. This paper investigation commences with a comprehensive elucidation of the mathematical underpinnings of the Booth algorithm. Subsequently, it involves the development of code corresponding to the mathematical expressions, which is executed within the VIVADO environment. Given the inherent simplicity of the experiment, virtual simulation is directly conducted in VIVADO. The analysis is grounded in the examination of data waveforms. Ultimately, the output waveform is correlated with the BOOTH mathematical formula, culminating in the successful implementation of the BOOTH algorithm. This paper aims to effectively curtail the presence of '1's in computer data, thereby reducing the computational steps and conserving operational space. This, in turn, leads to a notable acceleration in the computational speed, accomplishing the objective of enhancing computational efficiency.

Keywords: Computational Efficiency, Booth Algorithm, Multipliers, Algorithmic Tools.

1. Introduction

Verilog, recognized as the predominant hardware description language in contemporary times, stands out as an exceedingly versatile means for articulating intricate chip designs and electronic systems. Consequently, this study has elected to harness Verilog as the programming medium for the Booth algorithm. In the current landscape of digital systems for signal processing and communication, the urgency of timely feedback has reached unprecedented levels, leading to an escalating demand for the applicability of the Booth algorithm [1]. The significance of multiplier performance has correspondingly intensified over time [2]. Pragmatic utilization of the Booth algorithm yields considerable enhancements in operational efficiency, such as the expeditious resolution of IP addresses and optimization of computer fixed-point multiplication, among other functionalities [3]. This is especially pertinent in domains like digital signal processing, where operations like convolutions and matrix multiplications entail copious accumulation and multiplication processes, for which Verilog offers superior flexibility in contrast to embedded multipliers, thus affording the freedom to craft multiplicative structures using the Verilog language.

The core objective of this research is to employ Verilog in elucidating the principles underpinning the Booth multiplier. The envisioned outcome entails a substantial reduction in the incidence of "1"s within the multiplier, thereby curtailing the frequency of accumulation and minimizing computational overhead. This, in turn, bestows significant improvements in operational speed. Notably, the implications of this research transcend computer-centric applications, promising far-reaching impacts across diverse fields. The ultimate stage of this study involves the verification of the Booth algorithm through simulation in the VIVADO environment.

2. Theoretical basis and software analysis

2.1. Booth principle elaboration

To appreciate the advantages of the Booth algorithm, it is imperative to first grasp the operational framework of the conventional multiplier. A high-performance multiplier represents a pivotal component within modern microprocessors and serves as the linchpin of real-time image processing and digital signal processing [4]. Computer multiplication fundamentally comprises two sequential steps: the initial step involves multiplying each bit of the multiplicand by each bit of the multiplier, while the subsequent step encompasses multiplying the partial products by each bit of the final product. Conventionally, multiplication has been envisaged as a form of "repetition." For instance, the product of a and b entails adding a to itself b times. This repetitive addition continues until the multiplier b is decremented to zero, culminating in the final product. The design of the multiplier module is founded on this conceptual framework. Nevertheless, it is paramount to address the question of whether the product is positive or negative. Indeed, the sign of the product in a multiplier hinges on the outcome of differentiating the two product terms [5]. This association is elucidated in Table 1.

Table 1. Mapping

A	B	C (Result)
1	1	0
0	0	0
1	0	1
0	1	0

Undoubtedly, this conventional multiplier offers a unique advantage by conserving resources, particularly in scenarios involving low-bit width operations where it finds widespread utility. However, it is not without its corresponding drawback. In the face of high-bit width operations, the operational speed experiences a substantial reduction, inevitably leading to a heightened consumption of clock signals. To mitigate the proliferation of "1"s and achieve a significant boost in operational speed, a mathematical method has been devised – the Booth algorithm. Thus, to minimize the incidence of "1"s and, consequently, realize a notable enhancement in operational speed, the Booth algorithm is employed. By utilizing the Booth algorithm and its associated Booth coding module, it is feasible to halve the number of partial products, thus reducing the requirement for parallel additions and ultimately augmenting the multiplier's performance [6, 7].

Firstly, this paper assume that there are n-bit binary numbers f (complement), alternatively f(-1) = 0, which can be expressed as:

$$\begin{aligned} &F=-2^{(n-1)}*f(n-1)+2^{(n-2)}*f(n-2)+.....+2^{(0)}f(0)+f(-1); \\ &=-2^{(n-1)}*f(n-1)+2^{(n-2)}*f(n-2)+2^{(n-2)}*f(n-2)-2^{(n-2)}*f(n-2)+..... \\ &\quad +2^{(0)}*f(0)+2^{(0)}*f(0)-2^{(0)}*f(0)+f(-1)+f(-1)-f(-1) \\ &=2^{(n-1)}*(f(n-2)-f(n-1))+2^{(n-2)}(f(n-3)-f(n-2))+.....+2^{(0)}(f(-1)-f(0)) \end{aligned} \tag{1}$$

At this point it is possible to generalize and summaries the truth table 2.

Table 2. Truth table

fb	fn	fa
0	0	0
1	1	0
1	0	-1
0	1	1

At this point, the complement partial product will be reduced which will increase the rate of operation dramatically, but it is common to use Radix-4booth in our daily operations, so it is almost the same as the Radix-2booth derivation, and It can be found by analogy to get the expression and truth table of the Radix-4booth algorithm easily:

$$\begin{aligned} F&=-2^{(n-1)}*f(n-1)+2^{(n-2)}*f(n-2)+.....+2^{(0)}f(0)+f(-1); \\ &=(-2f(n-1)+f(n-2)+f(n-3))2^{(n-2)}+(-2f(n-2)+f(n-3)+f(n-4))2^{(n-4)} \\ &\quad +.....+(-2f(1)+f(0)+f(-1))2^{(0)} \end{aligned} \tag{2}$$

So the final expression can be written as: -2*f(i+1)+f(i)+f(i-1). The truth table can be listed as follows: Let Y be multiplied by F:

Table 3. Truth table

F(i+1)	F(i)	F(i-1)	-2*F(i+1)+F(i)+F(i-1)	Complement Partial product
0	0	0	+0	0
0	0	1	+1	1*Y
0	1	0	+1	1*Y
0	1	1	+2	2*Y
1	0	0	-2	-2*Y
1	0	1	-1	-1*Y
1	1	0	-1	-1*Y
1	1	1	0	0

This gives the truth table 3 for the base-4booth algorithm, and which can find our partial product by using the result of -2*F(i+1)+F(i)+F(i-1).Compared to Radix-2 Booth encoding, Radix-4 Booth encoding will halve the number of partial sums accumulated by multiplication. Next is the core idea of our Verilog writing

2.2. Software background and main features

Vivado Design Suite, introduced in 2012 by FPGA manufacturer Xilinx, Inc., represents an all-encompassing, integrated design environment. This suite comprises cutting-edge tools that span from system-level design down to the integrated circuit (IC) level. These tools are founded upon a shared, scalable data model and a unified debugging environment. Notably, Vivado operates within an open framework rooted in standards such as the AMBA AXI4 interconnect specification, IP-XACT IP packaging metadata, Tool Command Language (TCL), Synopsys System Discipline Constraints (SDC), and other industry-established norms [8]. This framework is tailored to accommodate the specific requirements of individual design workflows.

One of Vivado's prominent features is its integrated High-Level Synthesis (HLS) tool. This sophisticated tool empowers users to directly program FPGA devices using C, C++, and System C. Consequently, it eliminates the necessity for manual RTL (Register-Transfer Level) creation, instead facilitating the generation of data-level (DL-level) cores through high-level synthesis [9]. Vivado, underpinned by the comprehensive range of programmable technologies, possesses the capability to accommodate designs of considerable complexity, scaling to support up to 100 million equivalent ASIC gates [10]. Notably, this software serves as the platform for the Verilog programming conducted in this research endeavor.

2.3. Verilog implementation and virtual simulation

Figure 1 illustrates the fundamental diagram of the Verilog flow. At the heart of the programming lies the essential rhythm, commencing with the identification of the BOOTH factor. Subsequently, it progresses from the BOOTH factor towards the formulation of the truth value equation as delineated in Table 3.

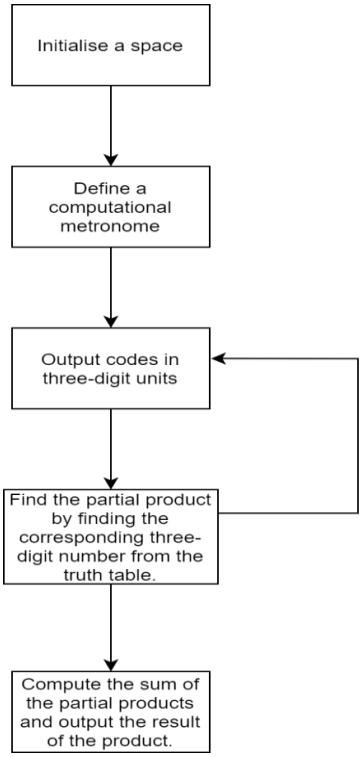


Fig 1. Core diagram of the Verilog flow (Photo/Picture credit: Original)

Figure 2 shows the virtual simulation results.

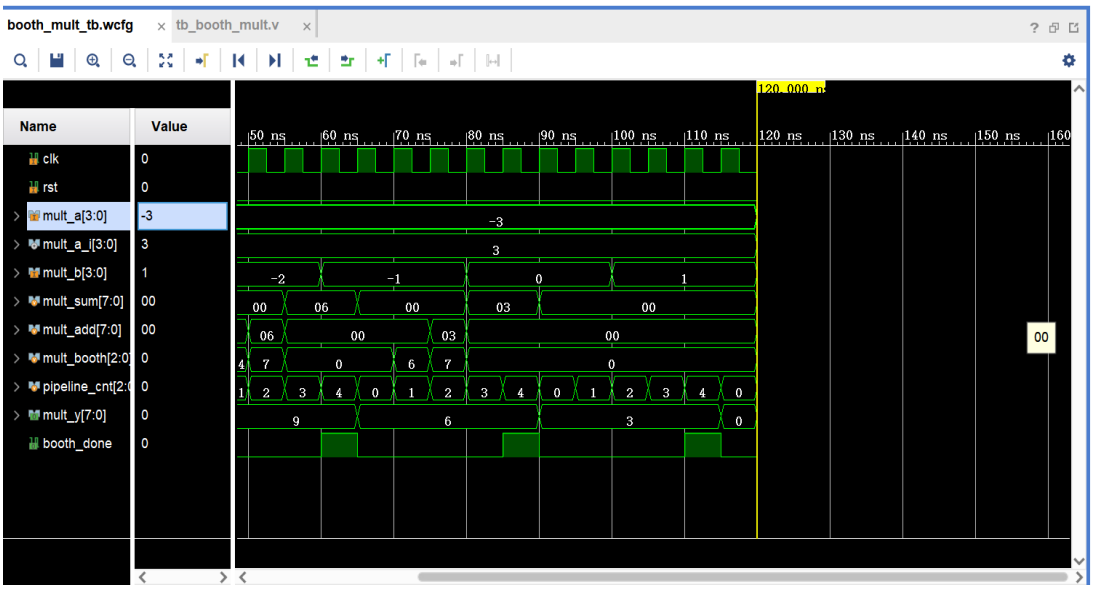


Fig 2. Virtual simulation results (Photo/Picture credit: Original)

The illustration above portrays the virtual simulation experiment's schematic. Upon analysis, it becomes evident that when the reset signal is triggered, the product assumes a value of 0. Subsequently, as each multiplied number is introduced, the product evolves as follows: when multiplied by -3, it equals 9; with -2, it reaches 6; and when multiplied by -1, it amounts to 3. This cascade of results demonstrates the successful execution of the Booth multiplication algorithm.

3. Application scenarios, challenges, and prospects for the future

3.1. Application Scenarios

The focus of this research, centered on the Verilog implementation of the Booth algorithm, is poised to make significant inroads across diverse application domains. Foremost among these applications is its pivotal role in the realm of high-performance multipliers, a linchpin in the architecture of contemporary microprocessors. These multipliers, often hailed as the workhorses of modern computing, play a critical role in enabling real-time image processing and the seamless execution of intricate digital signal processing tasks [11]. The implementation of the Booth algorithm using Verilog holds immense promise in elevating the computational efficiency of these processors.

Beyond its implications for microprocessors, this research extends its influence to a myriad of fields where computational efficiency is of paramount importance. Its ability to augment processing speeds is particularly advantageous in contexts such as IP address matching and computer fixed-point multiplication. In the domain of digital signal processing, a field replete with convolutional operations and complex matrix manipulations, the reduction of "1"s within data structures emerges as a game-changer. This not only enhances operational efficiency but also promises significant energy savings, aligning with the growing demand for sustainable computing solutions.

3.2. Challenges

While the Booth algorithm offers undeniable advantages, it is important to acknowledge that its application is not without its unique set of challenges. In contexts characterized by low-bit width operations, the algorithm shines by its adeptness in resource conservation. However, its prowess diminishes notably when confronted with high-bit width operations, resulting in a marked reduction in operational speed. This challenge underscores the need to delve deeper into the development of solutions that can harmonize the efficiency of the Booth algorithm across a spectrum of scenarios.

Moreover, one critical challenge to address pertains to the consumption of clock signals, particularly in operations involving high-bit width multipliers. The generation and transmission of clock signals are crucial to the proper functioning of digital systems, but when used extensively in high-bit width computations, they can lead to escalated power consumption [12]. Striking a balance between computational speed and power efficiency becomes paramount, requiring innovative approaches to ensure that the advantages of the Booth algorithm can be harnessed while mitigating the potential drawbacks.

3.3. Prospects

Anticipating the road ahead, this research delineates a promising trajectory toward elevating computational efficiency. The synergistic employment of Verilog and the Vivado Design Suite, in conjunction with the Booth algorithm, ushers in a realm of possibilities for future research and development in the realm of digital systems and FPGA-based applications. This confluence of tools and methodologies lays the foundation for potential innovations that could reshape the digital landscape.

In the upcoming stages of exploration, one can envisage a focus on refining the performance of the Booth algorithm, particularly in high-bit width scenarios. These endeavors may entail the integration of supplementary algorithms or novel methodologies that enhance the algorithm's adaptability across a spectrum of computational contexts. By addressing the challenges associated with high-bit width operations, researchers can extend the algorithm's reach and applicability, thereby enhancing its overall efficiency [13].

The vistas of opportunity also extend to the realm of energy-efficient computing. The reduction of "1"s in computations not only promises augmented computational speed but also carries profound implications for power conservation. In an era marked by a burgeoning demand for sustainable computing solutions, this research serves as a cornerstone for future initiatives aimed at marrying

computational speed with minimal power consumption, thus aligning with the global aspiration for energy-efficient computing solutions.

4. Conclusion

In conclusion, the research presented in this paper delves into the implementation of the Booth algorithm using Verilog, along with the utilization of the Vivado Design Suite. This research is situated within an array of application scenarios and holds the potential to address various computational challenges while offering promising prospects for the future. The study's primary application lies in the realm of high-performance multipliers, a pivotal element in contemporary microprocessors and critical for real-time image processing and digital signal processing. The Verilog and Vivado Design Suite combination's adaptability in FPGA-based applications enhances its relevance in various computational tasks.

One of the standout features of this research is its capacity to efficiently reduce the presence of "1"s in data, a capability with broad-reaching implications. This property finds practical utility in diverse computational tasks, such as IP address matching and computer fixed-point multiplication, and proves particularly valuable in digital signal processing applications, which involve convolutional and matrix operations.

However, the application of the Booth algorithm also presents challenges, particularly in scenarios involving high-bit width operations. Balancing the algorithm's efficiency across different scenarios while managing power consumption is a significant area of concern.

Looking ahead, this research offers a promising path for enhancing computational efficiency. It lays the groundwork for further exploration and development in the field of digital systems and FPGA-based applications. Future endeavors may involve optimizing the Booth algorithm's performance in high-bit width scenarios and exploring methods to reduce power consumption.

References

[1] Booth A D. A signed binary multiplication technique. The Quarterly Journal of Mechanics and Applied Mathematics, 1951, 4(2): 236-240.

[2] Cheng Q, Dai L, Huang M, et al. A Low-Power Sparse Convolutional Neural Network Accelerator with Pre-Encoding Radix-4 Booth Multiplier. IEEE Transactions on Circuits and Systems II: Express Briefs, 2022.

[3] Khurshid B, Khan J J. An Efficient Fixed-Point Multiplier Based on CORDIC Algorithm. Journal of Circuits, Systems and Computers, 2021, 30(05): 2150080.

[4] Zhang Y. Baseband and Lo Techniques with Integrated CMOS for Wideband Millimeter-Wave Sensing. University of California, Los Angeles, 2020.

[5] Sharma R, Kaur M, Singh G. Design and FPGA implementation of optimized 32-bit Vedic multiplier and square architectures. 2015 International Conference on Industrial Instrumentation and Control (ICIC). IEEE, 2015: 960-964.

[6] Katti R. A modified Booth algorithm for high radix fixed-point multiplication. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1994, 2(4): 522-524.

[7] Liudon. 16×16 Parallel Multiplier Design Using Booth Algorithm,"Modern electronic technology. Xi'an Jiaotong University, School of Computer and Communication Engineering, 2022.

[8] Rashid M, Hazzazi M M, Khan S Z, et al. A Novel Low-Area Point Multiplication Architecture for Elliptic-Curve Cryptography. Electronics, 2021.

[9] Sai A K, Hanumantha T R. An adaptive core mapping algorithm on NoC for future heterogeneous system-on-chip. Computers and Electrical Engineering, 2021, 95.

[10] Yeshiwas F, Endalew A. Design and FPGA realization of incremental fuzzy controller for stirred tank heater. Soft Computing, 2023, 27(22).

[11] C A S, Suyash G, Rahul R. Design of Efficient Multiply-Accumulate Unit for Convolutional Neural Networks. Journal of Physics: Conference Series, 2023, 2571(1).

[12] Aditya M, Santosh P. Power and delay efficient fir filter design using ESSA and VL-CSKA based booth multiplier. Microprocessors and Microsystems, 2021, 86.

[13] Joseph C. Computer Arithmetic and Verilog HDL Fundamentals. CRC Press, 2017.