Fast Binary Counters and Compressors Generated by Sorting Network

[1]Pallepati Deepika, [2]Marneni Swapna, *[3]Srinivas Bachu
[1,2,3]Department of ECE,
[1,2,3]Siddhartha Institute of Technology & Sciences, Narapally, Ghatkesar, Medchal-Malkajgiri, Telangana, India
bachusrinivas@gmail.com

Abstract: A crucial path in different DSP units involves the parallel summing of several operands. Compressors and counters with a high compression ratio are required to accelerate the summation. This article presents the sorting network based exact/approximate (4:2) compressors and quick saturated binary counters. When one-hot code sequences are the only way to encode reordered sequences, sorting networks are employed. There is an uneven distribution of inputs to the counter. By connecting the rearranged sequence to the one-hot code sequence using three distinct Boolean equations, the counter's output Boolean expressions can be considerably reduced. In order to achieve 27.0% improvement in delay, 26.2% improvement in area-delay product, and 52.0% improvement in power-delay product, we use the aforementioned technique to construct and optimise the (7,3) counter. The (15,4) counter is built in a similar fashion; it uses far less power and space while achieving a delay that is around 35.3% shorter. Instead, the performance of the built (31.5) counters increases by about 26.7% as the area increases. Incorporating counters into a 16×16 bit multiplier improves the multiplier's performance in area delay product by 31.8% and power delay product by 32.1% when compared to other designs of counters. The area-delay product is improved by 10.2% to 37.4% and the power-delay product by 22.3% to 48.0% when an 8×8 bit approximate multiplier is used to construct exact/approximate (4:2) compressors.

Index Terms **- Binary Counters, Sorting Network, Compressors.**

I. INTRODUCTION

The critical path, which is essentially the sum of several operands, is an integral part of many digital signal processing (DSP) units. To add up all the partial products, the basic multiplier circuit employs the Wallace Tree structure [1]. The performance of a basic multiplier is limited by the structure of the Wallace Tree. Elliptic Curve Cryptography (ECC) and public-key cryptosystems like RSA are able to generate modular multipliers through the use of a big number multiplier based on either the Toom-Cook technique [4] or the Karatsuba algorithm [3].

These two algorithms have been the subject of numerous articles, and hardware implementations of both have emerged. All of these articles, including [5], make use of the summation of many operands in a significant number of the circuit's components. As a post-quantum cryptosystem, fully homomorphic encryption (FHE) provides strong protection for data stored in the cloud. To speed up the multiplication of huge numbers and polynomials, however, it desperately needs the Number Theoretic Transform (NTT) [6]. The fundamental processing unit of certain high radix [6] NTT implementations is made up of the summation of a number of different operands. Two well-known methods for combining numerous operands are the Reduced Wallace Tree [2] and its better variant, the Wallace Tree structure [1]. These techniques use complete adders as (3,2) counters, which eats time at a logarithmic rate, to speed up the summing. It is also possible to refer to this kind of structure as a carry save structure.

A number of digital signal processing (DSP) units heavily use the critical path, which is largely composed of the sum of several operands. In a simple multiplier circuit, the partial products are added using the Wallace Tree structure. The basic multiplier is limited by how well the Wallace Tree structure works. A modular multiplier derived from the Toom-Cook method [4] or the Karatsuba algorithm [3] is utilised by elliptic curve cryptography (ECC) and public-key cryptosystems such as RSA.

II. LITERATURE REVIEW

These two algorithms have been the subject of numerous articles, and hardware implementations of both have emerged. Using the addition of several operands, the publications, such as [5], define many circuit components. When it comes to cloud computing, a postquantum cryptosystem that provides strong security is fully homomorphic encryption, or FHE. The speedup of big number and polynomial multiplication, however, is severely hindered without the number theoretic transform (NTT) [6]. The fundamental processing unit of certain high radix [6] NTT implementations is made up of the summation of a number of different operands. When it comes to summing up a large number of operands, two of the most famous methods are the reduced Wallace tree [2] and the original Wallace tree structure [1]. The summing speed is increased by a logarithmic amount of time using these methods, which use full adders as (3,2) counters. There is another name for this kind of organisation, which is the carry–save structure. Since that time, numerous articles have been written that explore how to create a structure that is more time efficient in order to speed up the summation process, such as [7-10]. To design a counter or compressor that can handle more bits at the same weight while maintaining performance is the primary objective when going for a higher compression ratio than the (3,2) counter.

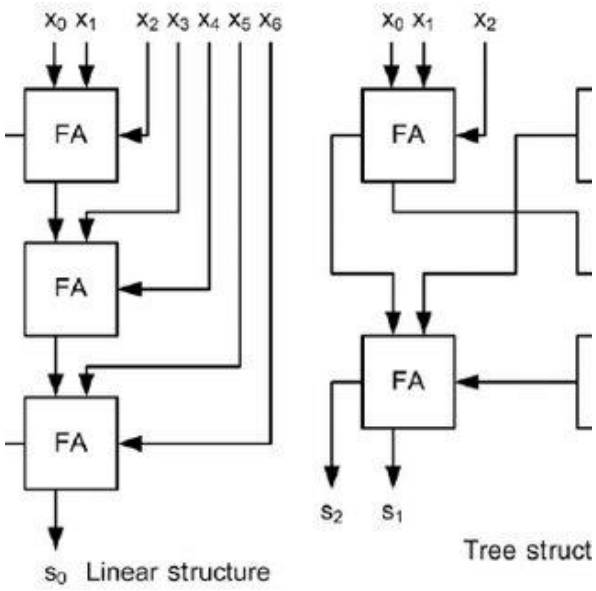

Fig. 1. Using complete adders, merge the (7,3) counter.

A bias voltage of 0.55 V is also applied to each PMOS load transistor. The circuit has a worst-case delay of 1.3 nanoseconds when it is not loaded, and it employs four buffer/inverter circuits for each output, which results in a delay of 1.6 nanoseconds. The configuration exhibits delay characteristics that are equivalent to those of counter circuits constructed employing source-coupled complete adder circuits in a tree design (7, 3), as depicted in Figure 1.

III. PROPOSED (7,3) COUNTER

In this section, a 7-and-3 counter is designed. A brief review of the design in [11] is in order before we get on to the main point of comparison. After considering the recommendation in [11], they built a (7,3) saturation counter using a symmetric stacking structure and an extremely fast (6,3) counter. The design is the quickest when compared to other (7,3) counters, however it has poor delay performance due to optimising it and just adding a MUX to the critical route. Our solution to the issue in [11] is to build a (7,3) counter directly. We begin with two sorting networks in an asymmetrical fashion, as shown in Figure 2, as opposed to the symmetric stacking structure.

Some Characteristics of Sorting Network

Two features of sorting networks are summarised here in accordance with the review given in the preceding section. Since "1" is larger than "0," the first "1" is at the beginning of the sequence when there are any "1"s and the last "0" is at the end of the sequence when there are any "0"s, as illustrated in Figure 2. For a reordered sequence to have both "1"s and "0"s, the two values must meet at some point. By inserting a fixed one-bit "1" at the start and a fixed one-bit "0" at the end of the rearranged sequence, we may manage sequences that consist solely of "1"s or "0"s. The 0,1-junction will always exit due to this.

Secondly, the original sequence (which served as inputs to two sorting networks) and the rearranged sequence both contain an equal amount of "1"s and "0"s. We disregard the fixed padded "1" when counting, even though it would alter the overall number of "1"s in the padded sequence.

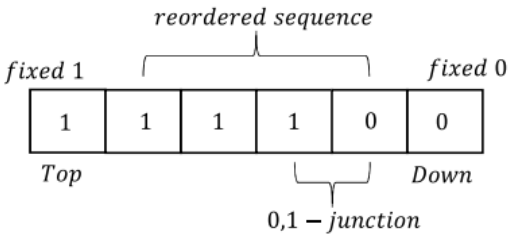

Fig. 2. Definition of a sequence.

One-Hot Code Generation

Asymmetric pre-reorder: Three layers of binary sorters are required for both 3-way and 4-way sorting networks (in the latter case, the two sorters on the same layer can be computed in concurrently), as shown in Figure 3. The basic two-input logical

gate layer, textcolorred, is consumed by each layer of binary sorters (Fig. 4). Therefore, the amount of time required for 3-way and 4-way sorting networks is nearly identical.

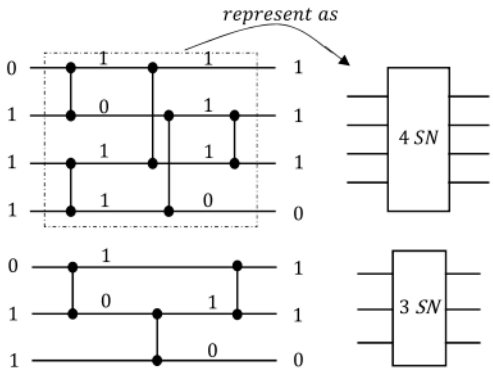

Fig. 3. 3-way and 4-way sorting networks.

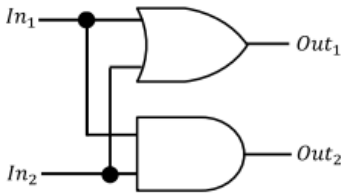

Fig. 4. Tow inputs binary sorter.

To divide a (7,3) counter's seven inputs in half, this is the foundation. The first part contains four bits, while the second part contains three.

IV. RESULTS

Figure 5 shows the proposed counters, which are more versatile than current designs because to their improved ADP performance and reduced latency where speed is required.

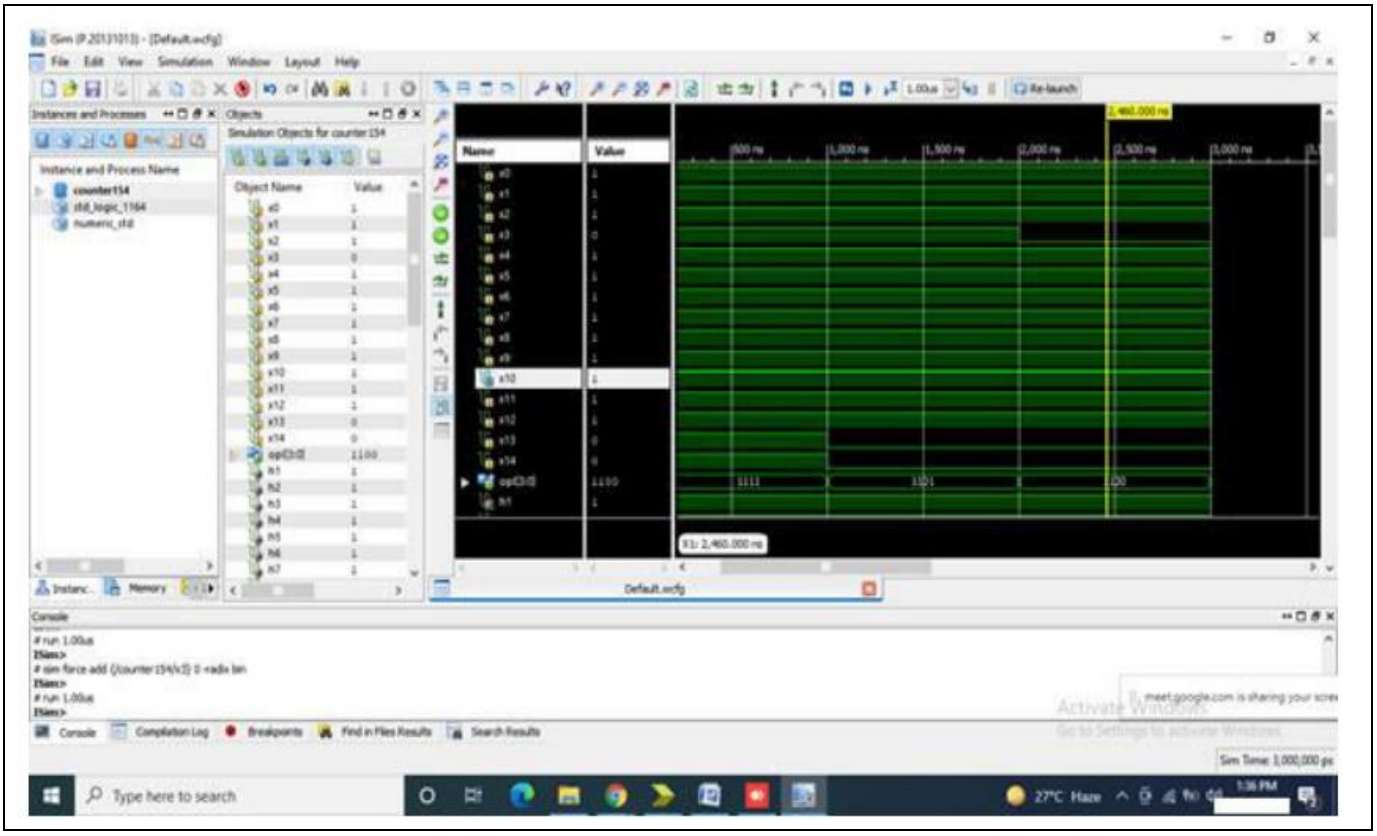

Fig 5: The ADP construct of (7,3), (15,4) counters.

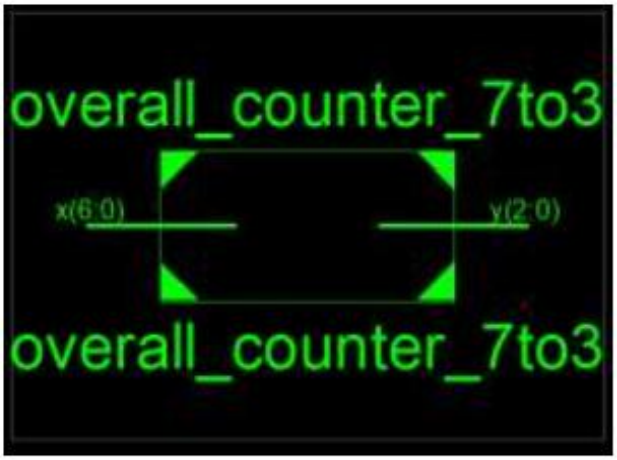

Fig 6: Top module for 7:3 counter

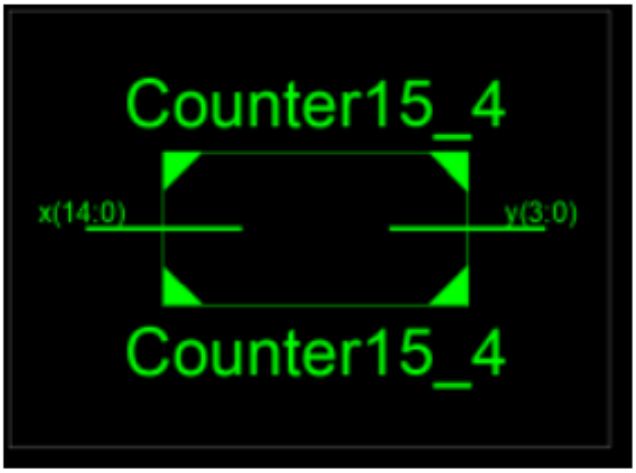

Fig. 8: Top module for 15:4 counter

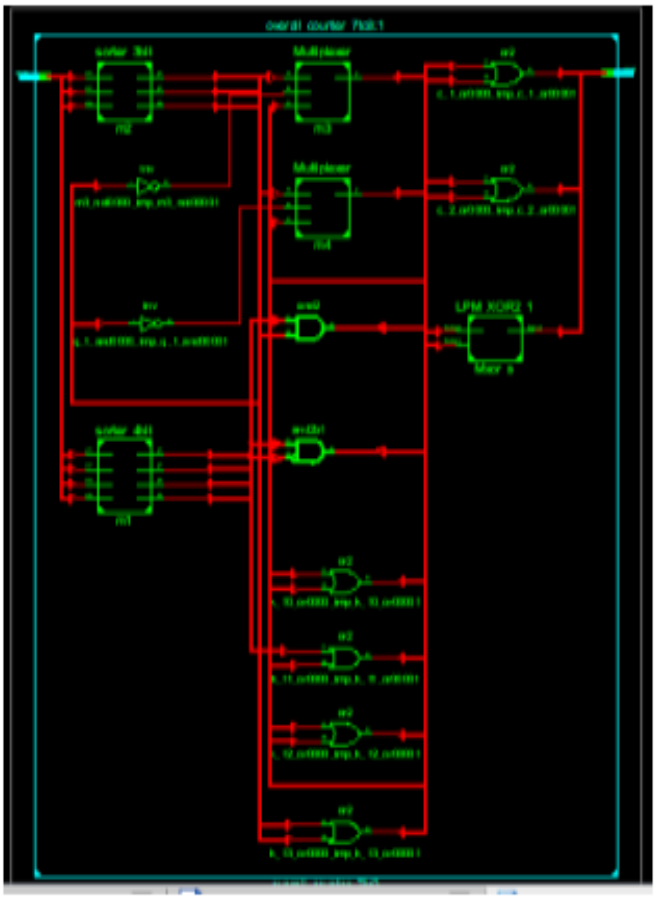

Fig 7: RTL schematic for 7:3 counters

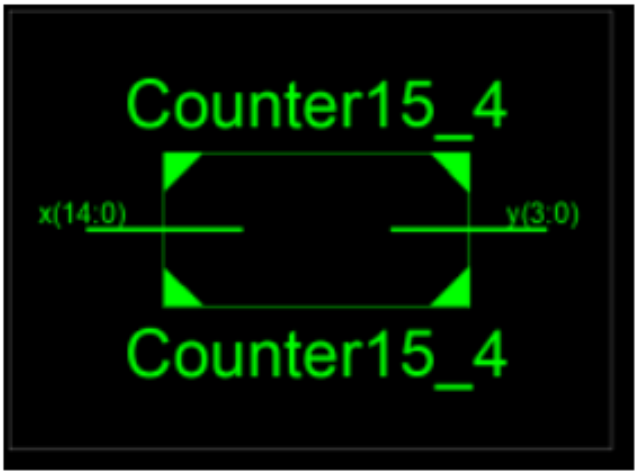

Fig. 8: Top module for 15:4 counter

V. CONCLUSION

A crucial path in different DSP units involves the parallel summing of several operands. Compressors and counters with a high compression ratio are required to accelerate the summation. We create (7,3) and propose a new counter design approach based on sorting networks in this paper. In comparison to other designs, the (7,3) counter uses less power and area while having an 8.1% to 27.0% reduction in delay. In this work, we present a current-mode multi-operand adder that operates in the same way as a (7, 3) counter circuit. All of the inputs and outputs will work with SCL circuits. The suggested circuit makes use of 28% less transistors than the standard SCL version. The proposed multi-operand adder has an area need that is 23% lower than its SCL equivalent; both are laid out. T Since the (15,4) counter reduces latency by 14.9% to 35.2% under critical speed conditions and outperforms existing designs in ADP and PDP by 14.7% to 49.0% and 41.2% to 72.7% under critical area or power conditions, respectively, it is more adaptable than existing designs.

REFERENCES

[1] C. S. Wallace, "A suggestion for a fast multiplier," IEEE Trans. Electron. Comput., vol. EC-13, no. 1, pp. 14–17, Feb. 1964, doi:10.1109/PGEC.1964.263830.
[2] R. S. Waters and E. E. Swartzlander, "A reduced complexity Wallace multiplier reduction," IEEE Trans. Comput., vol. 59, no. 8, pp. 1134– 1137, Aug. 2010, doi: 10.1109/TC.2010.103.
[3] P. L. Montgomery, "Five, six, and seven-term karatsuba-like formulae,"IEEE Trans. Comput., vol. 54, no. 3, pp. 362–369, Mar. 2005, doi:10.1109/TC.2005.49.
[4] J. Ding, S. Li, and Z. Gu, "High-speed ECC processor over NIST prime fields applied with Toom–Cook multiplication," in IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 66, no. 3, pp. 1003–1016, Mar. 2019, doi:10.1109/TCSI.2018.2878598.
[5] R. Liu and S. Li, "A Design and Implementation of Montgomery Modular Multiplier," 2019 IEEE International Symposium on Circuits and Systems (ISCAS), Sapporo, Japan, 2019, pp. 1-4, doi: 10.1109/ISCAS.2019.8702684.
[6] W. Wang, X. Huang, N. Emmart and C. Weems, "VLSI Design of a Large-Number Multiplier for Fully Homomorphic Encryption," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 22, no. 9, pp. 1879-1887, Sept. 2014, doi: 10.1109/TVLSI.2013.2281786.
[7] Asif and Y. Kong, "Analysis of different architectures of counter based Wallace multipliers," 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, 2015, pp. 139-144, doi: 10.1109/ICCES.2015.7393034.
[8] A. Najafi, B. Mazloom-nezhad and A. Najafi, "Low-power and high-speed 4-2 compressor," 2013 36th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2013, pp. 66-69.
[9] A. Najafi, S. Timarchi and A. Najafi, "High-speed energy-efficient 5:2 compressor," 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, 2014, pp. 80-84, doi: 10.1109/MIPRO.2014.6859537.
[10] S. Asif and Y. Kong, "Design of an algorithmic Wallace multiplier using high speed counters," 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, 2015, pp. 133-138, doi: 10.1109/ICCES.2015.7393033.