BINARY COUNTERS GENERATED BY SORTING NETWORK

L VASU[1],
[1]Assistant Professor, Dept :of ECE,PBRVITS, KAVALI, AP,INDI
B GangaBhavani[2], T Sireesha[3], R Prathima[4], R Joseph[5] ,D Satyam Gautam[6]
[2,3,4,5,6] UG Students, Dept; of ECE, VEC, Kavali, AP, India.

Abstract - The concurrent addition of multiple operands constitutes a crucial component of the critical path in diverse units of digital signal processing. In order to expedite the process of summation, it is imperative to utilise counters and compressors with a high compression ratio. The present article introduces a new approach for the rapid generation of saturated binary counters, utilising the sorting network. The counter's inputs are partitioned into two distinct groups and subsequently processed through sorting networks to produce rearranged sequences that can be exclusively denoted by one-hot code sequences. Three distinct Boolean equations are derived from the rearranged sequence and the one-hot code sequence, resulting in a notable reduction in complexity of the counter's output Boolean expressions. By employing the aforementioned approach, we establish and subsequently enhance the (7,3) counter, which exhibits a maximum performance improvement of at least 20% in terms of delay and power compared to alternative designs. Likewise, the counter with dimensions (15,4) is fabricated, exhibiting notable reductions in both power consumption and physical footprint.

Index Terms - Binary Counters, Exact/Approximate 4:2 Compressor, multiplier, One-hot Code, Sorting network

I. INTRODUCTION

The increased degree of integration in today's VLSI technology has made it possible to integrate numerous complicated devices onto a single chip. Furthermore, the use of the digital domain is required for maintaining power in analogue circuit approaches. Multipliers are critical components in many applications because they considerably impact the overall performance of a circuit in terms of power consumption, delay, and complexity. In general, there are two ways that may be used to improve the overall efficiency of a multiplier in terms of power dissipation, latency, and area. The former is dependent on the effective use of the multiplier function, while the latter is dependent on the proper selection of a logic circuit for its implementation. Various ways for creating a high-performance and low-speed multiplier have been developed over many decades [1, 2]. It should be noted, however, that the intermediate computation needed by multiplication functions causes a linear drop in performance proportionate to the amount of the input bit size in these multiplication algorithms. This problem grows more serious as the number of input bits increases. Nonetheless, this problem may be avoided by adding partial products at the same time. As a result, the current work investigates potential ways for improving the efficiency of existing compressors[3,4].A slew of queries have recently been focused towards the creation of VLSI systems based on low-power consumption, with the goal of generating a varied variety of computing systems. Since the introduction of VLSI technology, a wide range of low-power dependent portable gadgets, such as handheld communication devices, laptop computers, and personal digital assistants, have been conceived and manufactured for a variety of uses. The challenging goals of generating significant chip density and throughput often need satisfying the requirements for low energy consumption in the majority of circumstances. The presence of bottlenecks has presented several obstacles to design engineers working on VLSI circuits for low-power applications. The growth of this sector has resulted in the development of circuits with high speed, maximum throughput, small chip area, and low power consumption[5-8]. The Wallace tree structure [5] and its modified approach reduced Wallace tree [6] are the most well-known methods of multiple operands summation. These approaches speed the summing by using complete adders as (3,2) counters, resulting in logarithmic time consumption. This structure is also known as a carry-save structure. Many articles have since studied ways to build a more time-efficient structure to speed the summing, such as [11]-[17]. By considering additional bits at the same weight, the fundamental aim is to build a counter or a compressor with a larger compression ratio than the (3,2) counter. Figure 1 depicts a simple (7,3) counter arrangement paired with complete adders. Some articles have examined counters (4,3), (5,3), and (6,3) [10], as well as (7,3) [11] and (15,4) [12]. They count how many "1"s there are in the inputs. If a counter is a saturated counter, and its compressed results exactly replicate all of the "1"s in the inputs, its compression efficiency meets the limit.

III.EXISTING SYSTEM

. The process of adding a 7:3 compressor involves the utilization of four Full Adders. The initial Full Adder (FA1) performs the addition operation on X1, X2, and X3, resulting in the generation of Carry1 and Sum1. The Carry1 is subsequently utilized as Cout1. The Full Adder denoted as FA2 performs the arithmetic operation of adding the binary digits Sum1, X4, and X5, resulting in the generation of Cout2 and Sum2. The Carry and Sum outputs of a 7:3 compressor are obtained by adding the inputs Sum2 of FA2, Cin1, and Cin2. The increasing demand for low power architectures has led to a growing interest in inexact circuits , which priorities energy/power, delay, and area over exactness of output, while maintaining a reasonable level of accuracy. The rising popularity of imprecise circuits can be attributed to the significant enhancement of the three parameters. Previously, the original circuit was being approximated through the scaling of the supply voltage, vdd, which allowed for a certain degree of error tolerance .However, this approach had significant drawbacks, as it resulted in an increase in the hardware of the overall circuit in the form of level shifters for fine-tuning the supply voltage. In order to mitigate these limitations, alternative techniques at the architectural level that have minimal hardware requirements, such as probabilistic pruning and probabilistic logic reduction , have been devised. One method involves eliminating extraneous hardware during circuit design, while the other involves flipping bits in the minterms of Boolean functions. This results in improvements in the three dimensions of energy/power, delay, and area, with a slight trade-off in accuracy. The utilisation of approximate compressors was employed in the development of Inexact Multipliers, as documented in references 50 through 53. The research cited in reference 54 investigates the feasibility of employing decimal compressors for the purpose of addressing decimal multiplication. The primary objective of the design is to concentrate on compressors, which are fundamental constituents of multiplication circuitry that are commonly employed in high-speed circuit design. Similarly, a 15:4 counter has been designed as illustrated in Figure 3.8.

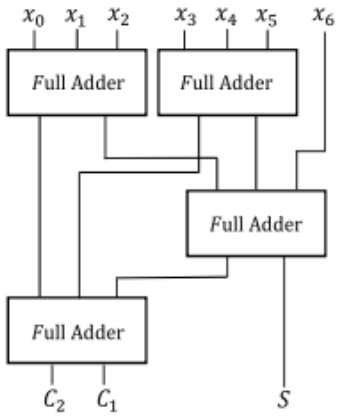

Figure 1: 7:3 Compressor Example

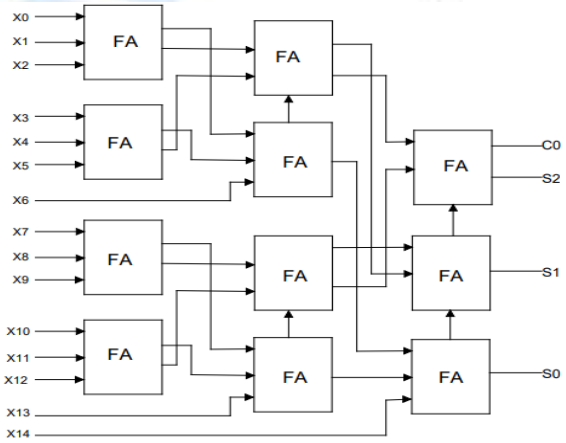

Figure 2: 15:4 Compressor Example

III.PROPOSED METHOD

7:3 Counter

In this section, we construct an efficient (7,3) counter. As the main comparison object, we first briefly review the design in proposed a very fast (6,3) counter with a symmetric stacking structure, and they constructed a (7,3) saturated counter on the basis of this (6,3) counter. Although it is the fastest compared to other (7,3) counter designs, its delay performance is worse because of simply introducing a MUX on the critical path without any optimization. To solve the problem in, we propose this method of directly construct a (7,3) counter. Unlike the symmetric stacking structure, we start with two sorting networks asymmetrically. The (7,3) counter generates three outputs, namely C2, C1, and S, where C2 holds the highest significant weight and S holds the least significant weight. The tabulated data in Table I displays the aggregate quantities of "1"s present in the input 7 bits that correspond to the outputs, specifically represented as Num = 2^2C2 + 2^1C1 + 2^0 S. The output sequence of a four-way sorting network is represented by the symbol "H", which consists of four elements denoted as H1 through H4 in a left-to-right order. The output sequence generated by a three-way sorting network is represented as sequence I, comprising of I1 to I3 in a left-to-right order. As per the data presented in Table I, it can be inferred that the input sequence of the (7,3) counter contains a minimum of four occurrences of the digit "1" when C2 is equal to 1. As previously mentioned, the value P4 = 1 indicates the presence of four consecutive "1"s in sequence H, which is also present in the input sequence of 4 SN due to the sorting network's inability to alter the total number of "1"s. Additionally, the

value Q0 = 1 signifies the absence of any "1"s in sequence I. The equation P4&Q0 = 1 indicates that the seven-bit input contains a total of four "1"s, resulting from the sum of P4 and Q0. The value of C2 is equivalent to 1 in cases where the total of the subscripts of P and Q is greater than or equal to 4, as a consequence of this form of representation.

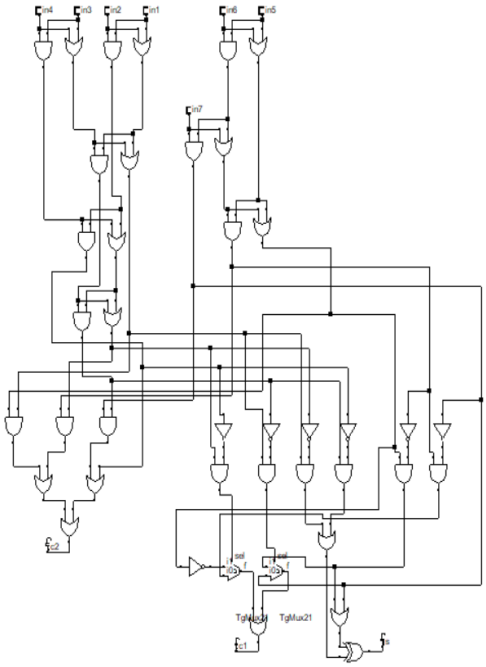

Figure 3: Proposed 7:3 Counter

Proposed 15:4 Counter

An efficient (7,3) saturated counter has been implemented. However, for certain applications, a (15,4) saturated counter would prove to be advantageous. In this section, the aforementioned techniques are employed to fabricate (15,4) fully-loaded counters, which are subsequently expounded upon in a concise yet lucid manner. The 8-way sorting network is illustrated in Figure 4. The output of this sorting network is achieved through the utilization of six layers of fundamental logic gates. By eliminating a single component from the 8-way sorting network, it is possible to derive a seven-way sorting network that utilizes six layers of fundamental logic gates.

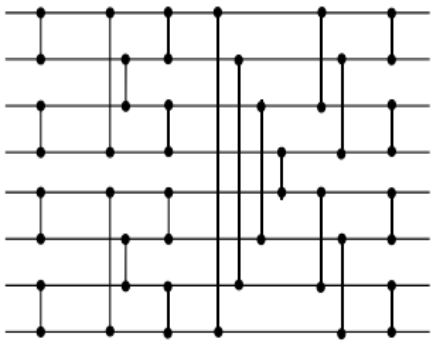

Figure 4: 8-bit Sorter

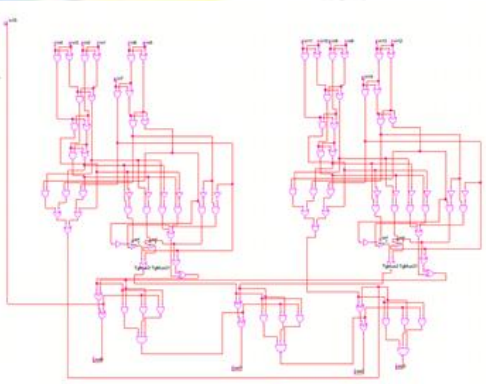

Figure 5: 15:4 Proposed Counter

IV.RESULTS DISCUSSION

To assess the effectiveness of the proposed fast counters in cryptography as well as DSP applications, the counters have been designed and executed in Microwind tools using CMOS 45nm technology. Performance indicators including power consumption, delay are the foundation of the analytical assessment. The existing and proposed counter circuits detailed in the results are performed at a temperature of 27°C, to guarantee uniformity in comparisons. From the simulation results of 3-bit and 4-bit sorters as shown in Figures 6 and 7 it is evident that for both 4 SN and 3 SN, the input sequences are reordered in the form of the larger number at the top and the smaller number at the bottom after three layers of sorter. Power and Delay given in table 1 and 2

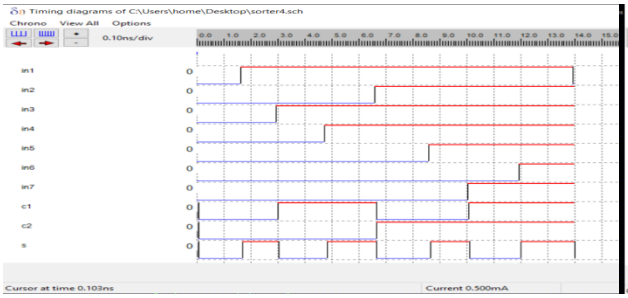

Figure 6: 7:3 Counter Result

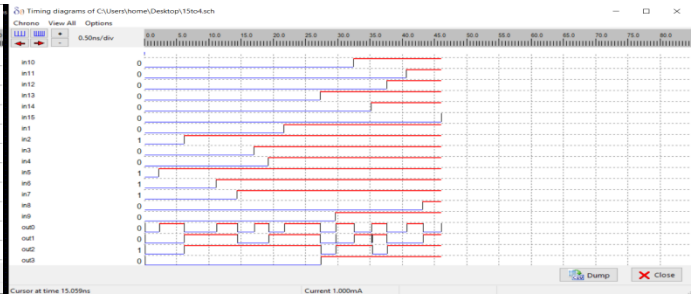

Figure 7 : 15:4 Counter Result

Table 3: Performance Comparison of 7:3 Counters

Technology	Power(uW)	Delay
Proposed	144.919	85.6ps
Existing	154.805	314ps

Table 4: Performance Comparison of 15:4 Counters

	Power Consumption	Delay
Proposed	404.854Uw	455.15ps
Existing	552.389uW	638.51ps

V CONCLUSION

The present article introduces a novel technique for counter design that relies on a sorting network. The proposed approach is utilised to construct (7,3) and (15,4) counters. The (7,3) counter exhibits a delay reduction of at least 60% compared to alternative designs, while also demonstrating lower power consumption. The (15,4) counter exhibits greater flexibility in comparison to pre-existing designs by attaining a 20% reduction in delay during instances of critical speed, while also exhibiting superior performance in terms of power consumption. Moreover, as the pace of technological progress accelerates, novel innovative remedies and design components could be devised to meet the requirements of technological scalability benchmarks.

REFERENCES:

1. Bahadori Milad, Kamal Mehdi, Afzali-Kusha Ali and Pedram Massoud , “A comparative study on performance and reliability of 32-bit binary adders”, Integration, the VLSI Journal, **vol. 53, no.1, Mar., pp. 54-67, 2016.**
2. Gupta J, Grover A, Wadhwa G K and Grover N Multipliers using low power adder cells using 1 0nm technology . *In Proc. of International Symposium on Computational and Business Intelligence*, **pp. 3-6, 2013 .**
3. Jamshidi V, Fazeli M and Patooghy A A low power hybrid MTJ/CMOS (4-2 compressor for fast arithmetic circuits. *In Proc. of International Symposium on Computer Architecture and Digital Systems*, **pp. 1-6,2015**.
4. Jin-Fa Lin Yin-Tsung Hwang Ming-Hwa Sheu Low Power 10-transistor full adder design based on degenerate pass transistor logic. *In Proc. of IEEE International Symposium on Circuits and Systems*, **pp. 496-499, 2012 .**
5. V. Elamaran and H.N. Upadhyay, "Area delay and power comparisons of fault-tolerant systems with TMR using different voter circuits", *International Journal of Signal and Imaging Systems Engineering*, **vol. 10, no. 1-2, pp. 63-71, June 2017**.
6. . 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.
7. 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.
8. 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.
9. 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.
10. R. Liu and S. Li, “A design and implementation of montgomery modular multiplier,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Sapporo, Japan, May 2019, pp. 1–4, doi: 10.1109/ISCAS.2019.8702684.