International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1004-1013

Design and Implementation of Compressors and Quick Binary Counters Produced by Sorting Network on FPGA

Dr.P.Prasanna Murali Krishna [(1)], Mrs.N.Swarupa Rani [(2)], Shaik Moulali [(3),] Muddeti Amarnath [(4),] Dudekula Kasim Vali [(5),] Gurrala Narasimha [(6),]
[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,6,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract In this project, many digital signal processing devices, the critical route includes the summing of several operands in parallel. High compression ratio counters and compressors are required to accelerate the summing. This project presents a revolutionary approach to exact/approximate (4:2) compressors and fast saturated binary counters based on the sorting network. Asymmetrically split into two groups, the counter's inputs are then fed into sorting networks to produce rearranged sequences that can only be represented by one-hot code sequences. Three unique Boolean equations are constructed between the reordered sequence and the one-hot code sequence, which can greatly simplify the counter's output Boolean expressions. By building and further optimizing the (7,3) counter using the technique, we can outperform previous designs in terms of delay, area–delay product, and power–delay product, with maximum performances of 27.0%, 26.2%, and 52.0%, respectively. In a similar vein, the construction of the (15,4) counter results in a delay reduction of about 35.3% at the expense of a large reduction in power and area consumption. Approximately 26.7% more performance is achieved with the designed (31,5) counter, but the area increases instead. The performance of the multiplier in the area delay product and power delay product is 31.8% and 32.1% higher when the counters are embedded in a 16 × 16 bit multiplier, respectively, than when the counters are embedded in other counter designs. Additionally, we build exact/approximate (4:2) compressors based on sorting networks, and when integrated in an 8 × 8-bit approximate multiplier, they perform 10.2%–37.4% better in the area–delay product and 22.3%–48.0% better in the power–delay product.

Keywords: Cpmoressors,5G Communications, FPGA, Verilog HDL, Binary Counters and Sorting Networks.

I.INTRODUCTION

THE summation of multiple operands is widely used in various digital signal processing (DSP) units and constitutes a part of the critical path. A basic multiplier circuit adds all the partial products up with the Wallace Tree structure [1], whose performance is the bottleneck of the basic multiplier. Public-key cryptosystems, such as RSA and elliptic curve cryptography (ECC), use a big number multiplier based on the Toom-Cook [4] or Karatsuba algorithm [3] to construct modular multipliers. Many papers have studied these two algorithms and implemented them with hardware. In the papers, such as [5], many parts of the circuit utilize the summation of multiple operands. Fully homomorphic encryption (FHE) is a post-quantum cryptosystem that provides strong security in cloud computing, and it urgently needs number theoretic transform (NTT) [6] to accelerate large number multiplication and polynomial

multiplication. In some high radix [6] NTT implementations, the core processing unit is composed of the summation of multiple operands.

The most famous method of multiple operands summation is the Wallace tree structure [1] and its improved method reduced Wallace tree [2]. These methods use full adders as (3,2) counters to accelerate the summation, resulting in logarithmic time consumption. This type of structure is also called carry–save structure.

Since then, many papers have discussed how to construct a more time-efficient structure to accelerate the summation, such as [7]–[12]. The main idea is to construct a counter or a compressor with a higher compression ratio than the (3,2) counter by considering more bits at the same weight. For example, a basic (7,3) counter's structure combined with full adders is shown in Fig. 1. The compressors compress n rows into 2 rows by considering the carry bits between adjacent columns. Some papers have discussed (4,2) [8], (5,2) [9], and (7,2) [16] compressors, which compress four, five, or seven rows into two rows, respectively. However, they are still in the framework of full adders, which utilizes "XOR" gates as basic units, and their logical expressions are very difficult to simplify. The counters compress n rows into log2n rows. Some papers have discussed (4,3), (5,3), and (6,3) [10] and even (7,3) [11] and (15,4) [12] counters. They count the number of "1"s in the inputs. If a counter is a saturated counter, whose compressed results just right represent all the number of "1"s in the inputs, its compression efficiency achieves the limitation. The designs in [10] are all unsaturated and use too many "XOR" gates. Saturated counters are the main topics in this article. For example, (7,3) counter is a saturated counter because a 3-bit number can just right represent 0–7. Fritz and Fam [11] proposed a (6,3) counter that is constructed with a symmetric stacking structure. It is very fast compared with other designs but unsaturated. Then, Fritz and Fam [11] simply use a MUX on the critical path to construct a saturated (7,3) counter that affects the speed. Satish and Pande [17] reduce the (6:3) counter proposed in [11] to a (5:3) counter and combine three (5:3) counter into a (15:4) counter. However, this method is inefficient.

Approximate multipliers are widely used in many errors tolerant fields, such as digital image processing [18] and finite impulse response (FIR) filter [24], to accelerate the multiplication. Manikantta Reddy et al. [21] and Zervakis et al. [23] use approximate booth encoding and partial product perforation to get an approximate multiplier. Satish and Pande [17], Strollo et al. [18], Akbari et al. [20], and Venkatachalam and Ko [22] have discussed about approximate (4:2) compressors that can be used in approximate multipliers. The high-speed approximate (4:2) compressors in [18] are based on the symmetric stacking structure [11].

we propose a new method of designing saturated counters, (7,3) and (15,4) mainly, with higher efficiency. In this group of designs, we start with sorting networks asymmetrically. Then, with the help of two extended bits, we optimize the new design by logical simplification. We also construct exact/approximate (4:2) compressors with the sorting network.

1.2 REVIEW OF SORTING NETWORK

https://ijgst.com.2024.v13.i2.pp1004-1013

The sorting network [14] is an efficient parallel hardware network utilized for data sorting. The famous 0,1 principle [14] shows that, if a sorting network can sort a group of data whose elements are all 1-bit numbers, it can sort all types of numbers. In this article, we only adopt it for 1-bit data sorting [13].

A.Sorting Network Working Principle : The typical three- and four-way sorting networks [14] are shown in Fi
example: sequence [0, 1, 1, 1] represents the input of four-way sorting network (4 SN), and sequence [0, 1, 1] represents the input of three-way sorting network (3 SN). 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.

B.Sorter for 1-Bit Data: As mentioned above, the sorter reorders two inputs according to numerical magnitudes.
three layers of two-input basic logic gates.

multiplication, such as the binary multiplier or more complex algorithms like Booth's algorithm.

One-Hot Code:
One-hot encoding is a binary code in which only one bit is high (1) at a time, and all others are low (0). In the context of digital circuit design, it is often used to represent states in a state machine, where each state is uniquely identified by a single bit.

Sorting Network:
A sorting network is a network of comparators and wires designed to sort a sequence of numbers. It's a hardware implementation of a sorting algorithm and is often used in applications where sorting needs to be done quickly and in parallel.

II.EXISTING SYSTEM AND PROPOSED SYSTEM

1.EXISTING SYSTEM

Fig1. (4:2) compressor combined by full adders.

Binary Counter:
A binary counter is a digital circuit that counts in binary from 0 to 2^n - 1, where n is the number of bits in the counter. It increments by one with each clock cycle.

4:2 Compressor:
A 4:2 compressor is a digital circuit that takes in four input bits and produces two output bits. It compresses the information from four bits into two bits using logic gates.

Multiplier:
A multiplier is a digital circuit that performs the multiplication of two binary numbers. There are various algorithms for

2.PROPOSED SYSTEM

Fig2. Overall (7,3) counter circuit.

Fig3. Overall (15,4) counter circuit.
In this proposed system we construct an efficient (7,3) counter. As the main comparison object, we first briefly review the design in [11]. Fritz and Fam [11] 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 [11], we propose this method of directly construct a (7,3) counter. Unlike the symmetric stacking structure, we start with two sorting networks

asymmetrically, as illustrated in Fig. 2. By generating one-hot code sequences, we establish three special Boolean equations [see (2), (13), and (15)], which significantly simplifies the Boolean expressions related to outputs.

A. Some Characteristics of Sorting Network: we have two characteristics of sorting networks. First, as shown in Fig. 4, due to the fact that "1" is bigger than "0," all the "1"s are at the top of the sequence if there exist "1"s, and all the "0"s are at the bottom of the sequence if there exist "0"s. If there exist both "1"s and "0"s, there must be a position in the reordered sequence where there is the junction of "1" and "0." If there are only "1"s or "0"s, we can manage the sequence by padding fixed one bit "1" at the top and one bit "0" at the bottom of the reordered sequence to make sure that 0,1-junction always exits.

Second, the reordered sequence has the same total number of "1"s and "0"s as the original sequence (the inputs of two sorting networks). Although the padded "1" would influence the total number of "1"s in the padded sequence, it is fixed, so we ignore it while counting.

B. One-Hot Code Generation

1) Asymmetric Prereorder: As illustrated in Fig. 2, both three- and four-way sorting networks require three layers of binary sorter (the two binary sorters on the same layer in fourway sorting network can be calculated in parallel). Each layer of binary sorters consumes one basic two-input logical gate layer, as shown in Fig. 3. This means that the time consumed by the three- and four-way sorting networks is almost the same. Based on this, we divide the seven inputs of a (7,3) counter into two parts. One part contains 4 bits, while the other contains 3 bits. 2) Find 0,1-Junction and One-Hot Code Sequence: As shown in Fig. 4, the 0,1-junction can solely represent the reordered sequence under the promise of the extended fixed "0" and "1." Notice that the position of the 0,1-junction must be 1,0 from left to right. Therefore, we still utilize the fourway sorting network as an example, and then, we have the structure in Fig. 5

4.Construct (15,4) Counter

1) Seven- and Eight-Way Sorting Networks: Fig. 8 shows an eight-way sorting network [14]. This sorting network consumes six layers of basic logic gates to output the result. Removing one bit from the eight-way sorting network can obtain a seven-way sorting network that also consumes six layers of basic logic gates.

The outputs of the eight-way and seven-way sorting networks are denoted as sequence H (includes H1–H8 and extended to H0 − H9) and sequence I (includes I1–I7 and extended to I0–I8), respectively. By utilizing A&B logic, we get the one-hot code sequences P (P0–P8) and Q (Q0 − Q7). These sequences are like the sequences in (7,3) counter. Thus, we directly give out the key equations, as shown in (17)–(19). Note that (17) is just an example; this regular satisfies all the random separation. The symbol "" in (19) represents continuous "OR." 2) Boolean Expressions of (15,4) Counter: The 4-bit output of the (15,4) counter is denoted as C3C2C1 S. Table III shows the output corresponding to the number of the

number of “1”s in input sequence. Like the method by which we construct the (7,3) counter, first, establish logic equations between C3C2C1 S and sequences P and Q through the sum of the subscripts. Second, utilize (17)–(19) to optimize it. Then we get (20)–(23). Because the original Boolean expressions are too long, here, we express them with the Verilog syntax (especially ternary operator: “a?b:c”).

III. RESULTS AND ANALYSIS DISCUSSION

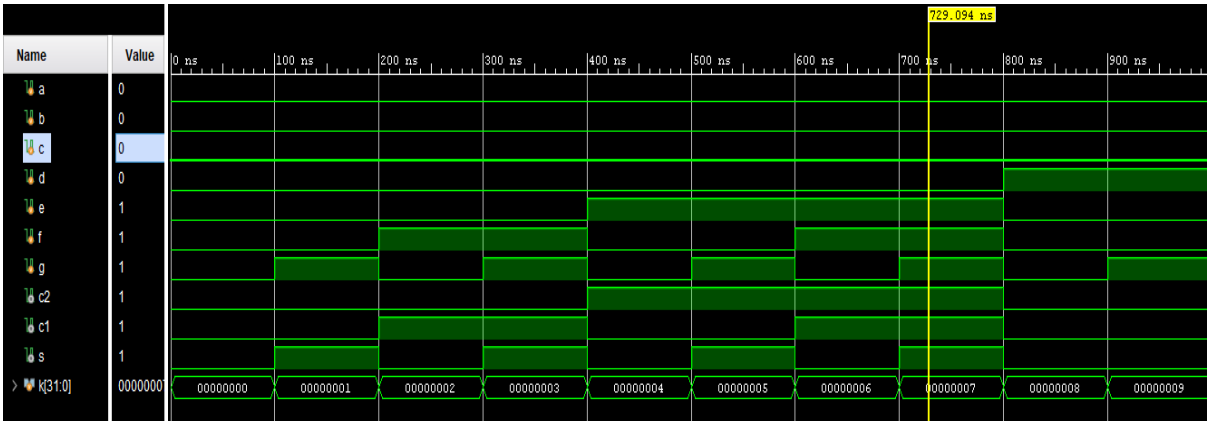
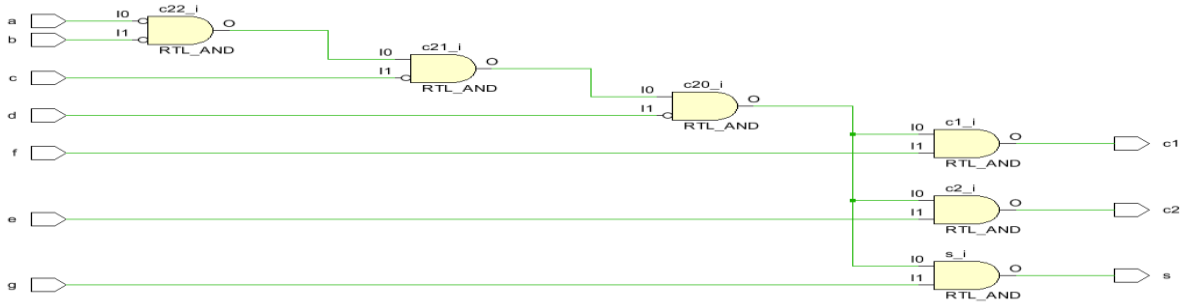


FIG1.SIMULATION OF 7X3 COUNTER

FIG2.SIMULATION OF 15X4 COUNTER



International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1004-1013

FIG3.RTL SCHEMATIC DIAGRAM 7X3 COUNTER

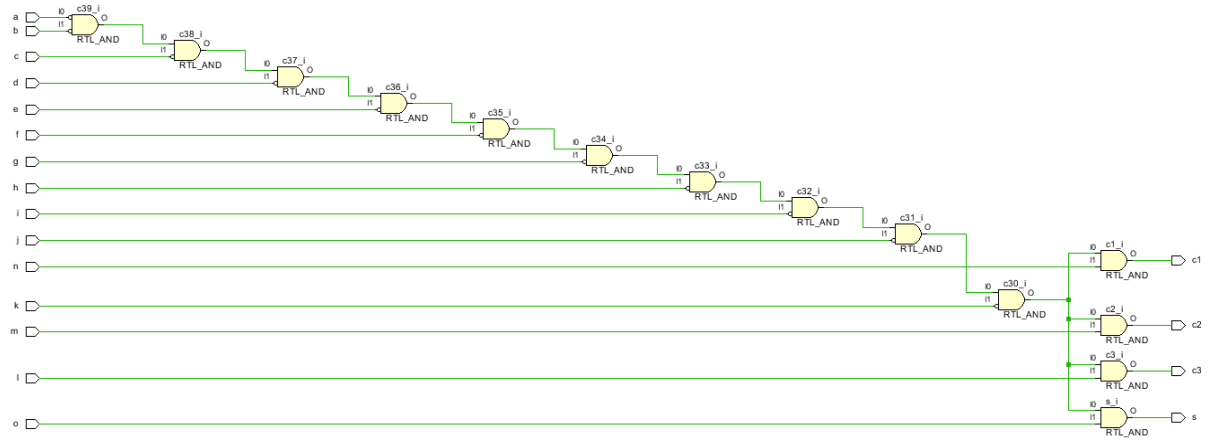


FIG4.RTL SCHEMATIC DIAGRAM 15X4 COUNTER

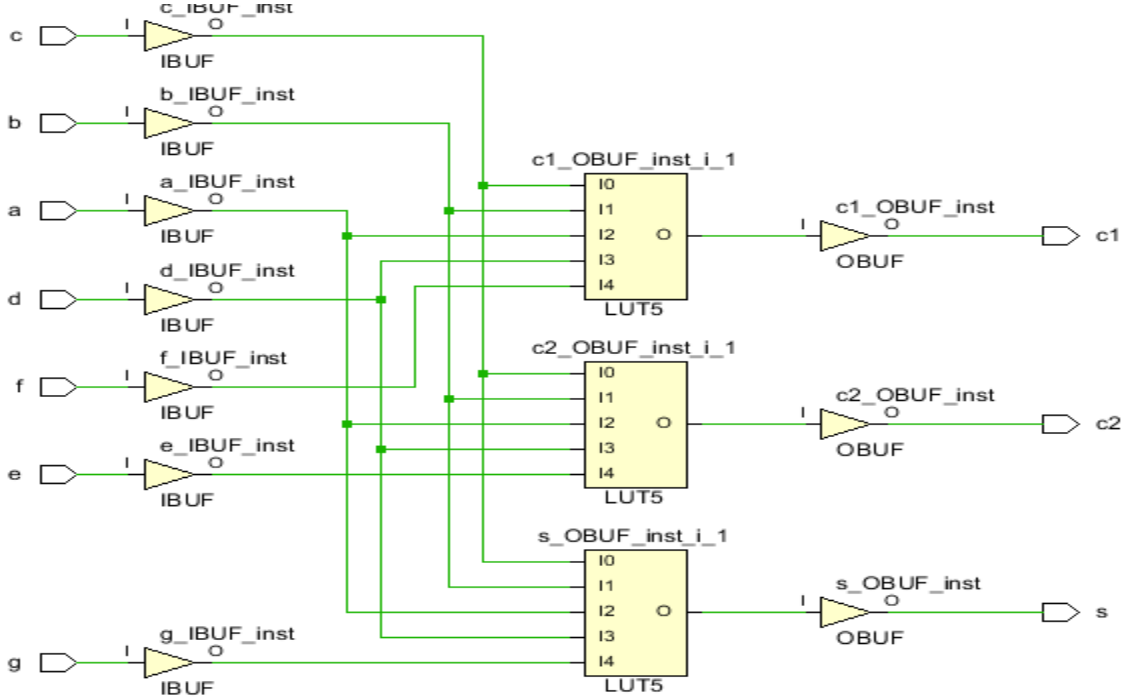


FIG5.RTL SYNTHESIS DIAGRAM 7X3 COUNTER

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1004-1013

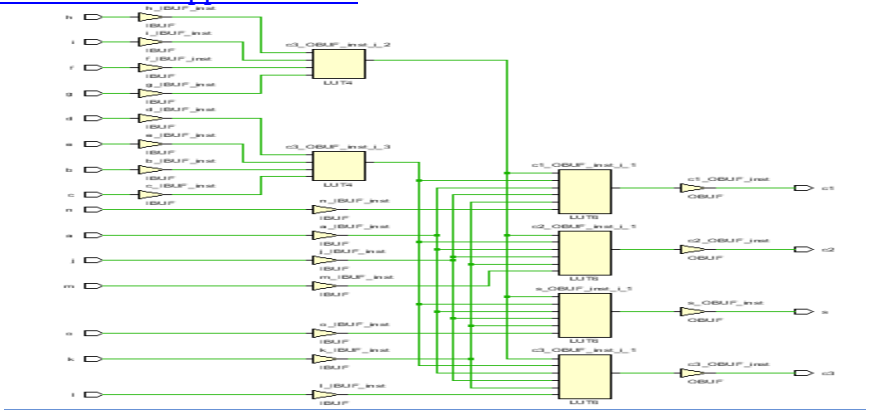


FIG5.RTL SYNTHESIS DIAGRAM 15X4 COUNTER

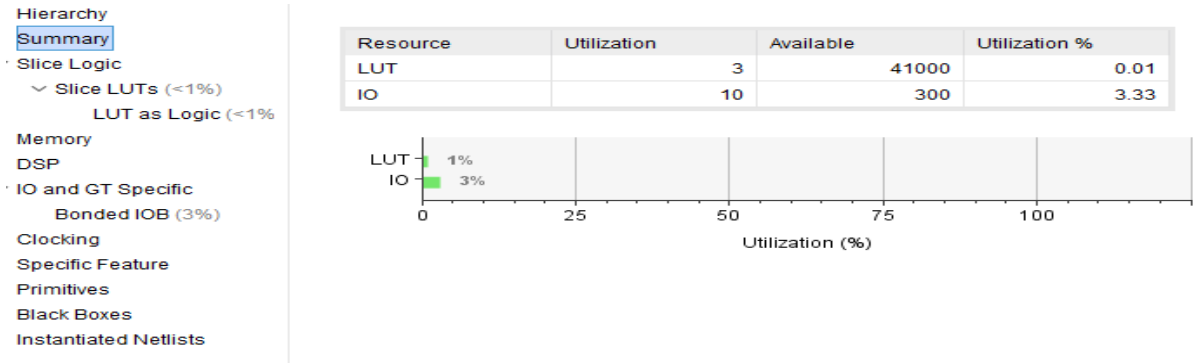


FIG7.UTLIZATION OF DIAGRAM 7X3 COUNTER

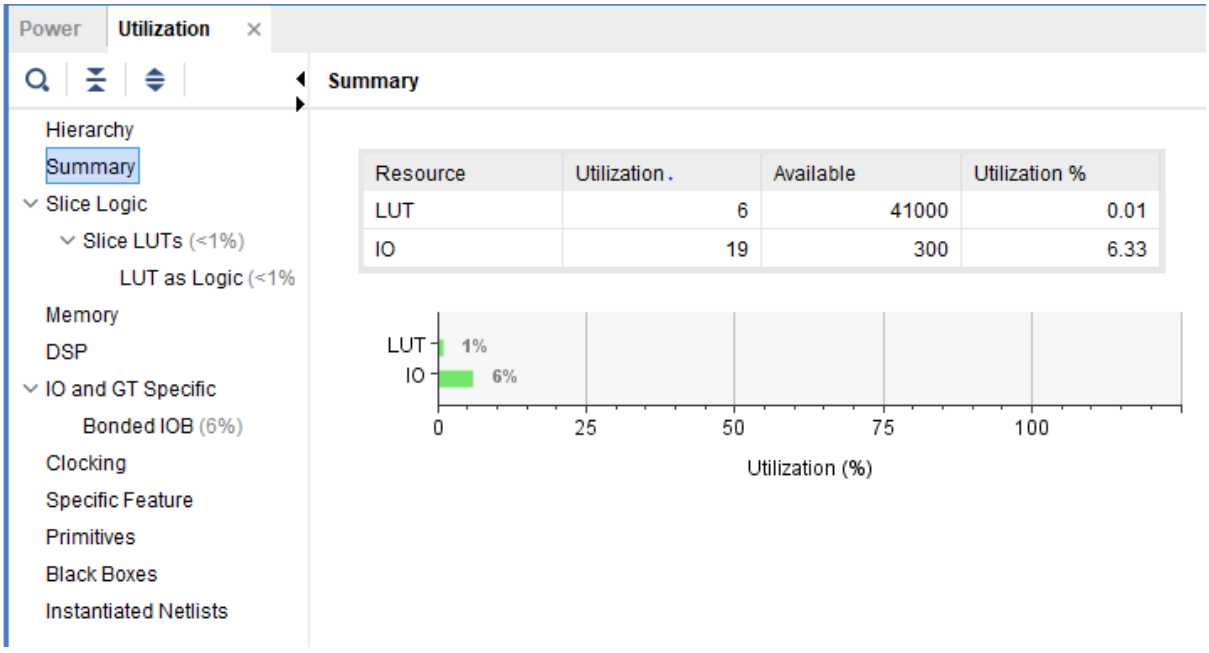


FIG8.UTLIZATION OF DIAGRAM 15X4 COUNTER

International Journal of Gender, Science and Technology
ISSN: 2040-0748
UGC Care Group I Journal.
Vol-13 Issue-02 Dec 2024
https://ijgst.com.2024.v13.i2.pp1004-1013

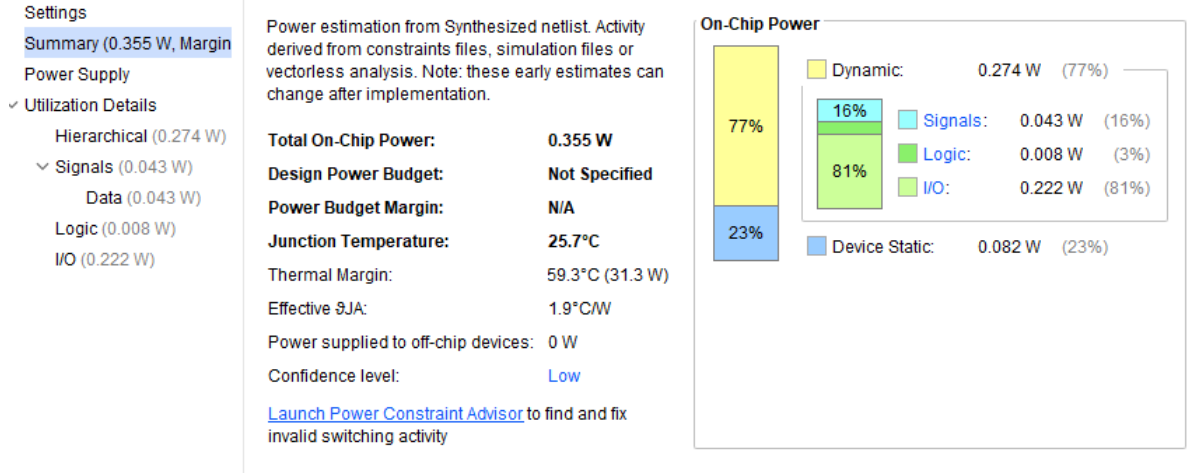


FIG9 POWER REPORT 7X3 COUNTER

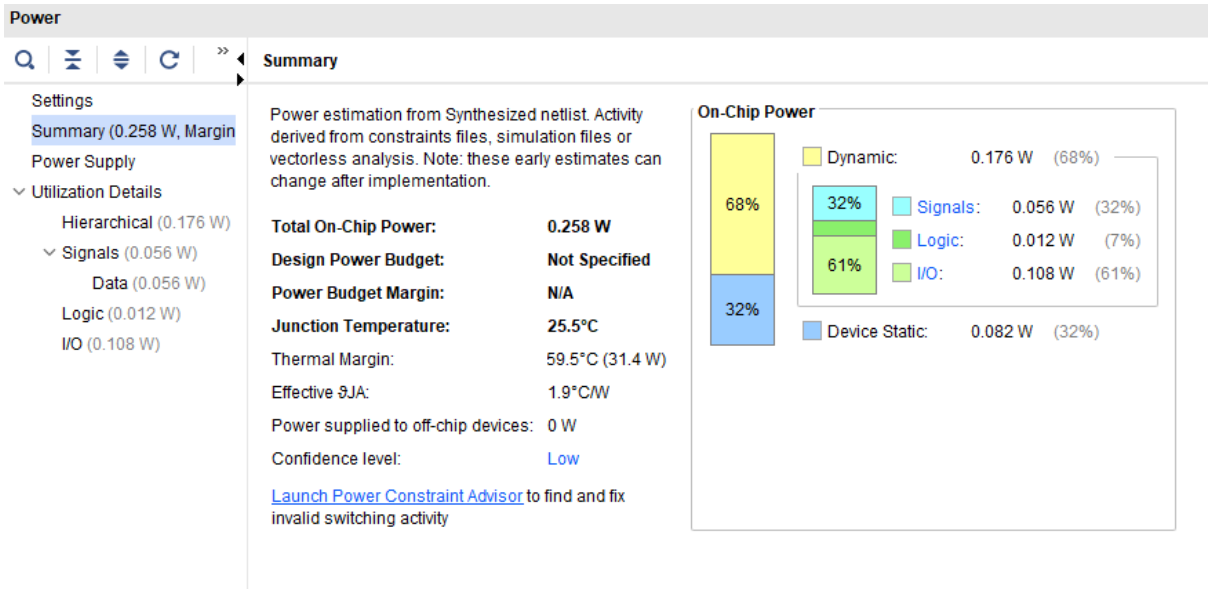


FIG10 POWER REPORT 15X4 COUNTER

Conclusion

In this project ,a new counter design method based on a sorting network is proposed, and we construct (7,3), (15,4), and (31,5) counters based on this method. The (7,3) counter has 8.1%–27.0% less delay than other designs and consumes less area and power. The (15,4) counter is more flexible than existing designs because it achieves 14.9%–35.2% less delay when the speed is critical and performs 14.7%–49.0% and 41.2%–72.7% better in ADP and PDP when the area or power is critical. The (31,5) counter has more than 22.0% shorter delay than other existing designs. When they are embedded in a 16-bit multiplier, the multiplier achieves 31.8% and 32.2% better in ADP and PDP in maximum than those embedded in other counter designs. Exact/approximate (4:2) compressors are also proposed based on a sorting network.

They perform approximately 10.2%–37.4% better in ADP and 22.3%–48.0% better in PDP when they are embedded in an 8-bit approximate multiplier.

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," in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), Sapporo, Japan, May 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 Trans. Very Large Scale Integr. (VLSI) Syst., vol. 22, no. 9, pp. 1879–1887, Sep. 2014, doi: 10.1109/TVLSI.2013.2281786.

[7] S. Asif and Y. Kong, "Analysis of different architectures of counter based wallace multipliers," in Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES), Cairo, Egypt, Dec. 2015, pp. 139–144, doi: 10.1109/ICCES.2015.7393034.

[8] A. Najafi, B. Mazloom-nezhad, and A. Najafi, "Low-power and highspeed 4-2 compressor," in Proc. 36th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), Opatija, Croatia, May 2013, pp. 66–69.

[9] A. Najafi, S. Timarchi, and A. Najafi, "High-speed energy-efficient 5:2 compressor," in Proc. 37th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO), Opatija, Croatia, May 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," in Proc. 10th Int. Conf. Comput. Eng. Syst. (ICCES), Cairo, Egypt, Dec. 2015, pp. 133–138, doi: 10.1109/ICCES.2015.7393033.

[11] C. Fritz and A. T. Fam, "Fast binary counters based on symmetric stacking," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 10, pp. 2971–2975, Oct. 2017, doi: 10.1109/TVLSI.2017.2723475.

[12] Q. Jiang and S. Li, "A design of manually optimized (15, 4) parallel counter," in Proc. Int. Conf. Electron Devices SolidState Circuits (EDSSC), Hsinchu, Taiwan, Oct. 2017, pp. 1–2, doi: 10.1109/EDSSC.2017.8126527.

[13] M. H. Najafi, D. J. Lilja, M. D. Riedel, and K. Bazargan, "Low-cost sorting network circuits using unary processing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 26, no. 8, pp. 1471–1480, Aug. 2018, doi: 10.1109/TVLSI.2018.2822300.

[14] D. E. Knuth, The Art of Computer Programming: Sorting and Searching, vol. 3. Reading, MA, USA: Addison-Wesley, 1973.

[15] M. Mehta, V. Parmar, and E. Swartzlander, "High-speed multiplier design

using multi-input counter and compressor circuits," in Proc. 10th IEEE Symp. Comput. Arithmetic, Grenoble, France, Jun. 1991, pp. 43–50, doi: 10.1109/ARITH.1991.145532.
[16] A. Fathi, B. Mashoufi, and S. Azizian, "Very fast, high-performance 5-2 and 7-2 compressors in CMOS process for rapid parallel accumulations," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 6, pp. 1403–1412, Jun. 2020, doi: 10.1109/TVLSI.2020. 2983458.
[17] T. Satish and K. S. Pande, "Multiplier using NAND based compressors," in Proc. 3rd Int. Conf. Electron., Mater. Eng. NanoTechnol. (IEMENTech), Kolkata, India, Aug. 2019, pp. 1–6, doi: 10.1109/IEMENTech48150.2019.8981067.
[18] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. D. Meo, "Comparison and extension of approximate 4-2 compressors for low-power approximate multipliers," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 67, no. 9, pp. 3021–3034, Sep. 2020, doi: 10.1109/TCSI.2020.2988353.
[19] Z. Yang, J. Han, and F. Lombardi, "Approximate compressors for errorresilient multiplier design," in Proc. IEEE Int. Symp. Defect Fault Tolerance VLSI Nanotechnol. Syst. (DFTS), Amherst, MA, USA, Oct. 2015, pp. 183–186, doi: 10.1109/DFT.2015.7315159.
[20] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-quality 4:2 compressors for utilizing in dynamic accuracy configurable multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 4, pp. 1352–1361, Apr. 2017, doi: 10.1109/TVLSI.2016.2643003.
[21] K. Manikantta Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi, "Design of approximate booth squarer for error-tolerant computing," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 28, no. 5, pp. 1230–1241, May 2020, doi: 10.1109/TVLSI.2020.2976131.
[22] S. Venkatachalam and S.-B. Ko, "Design of power and area efficient approximate multipliers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 25, no. 5, pp. 1782–1786, May 2017, doi: 10.1109/TVLSI.2016.2643639.
[23] G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris, and K. Pekmestzi, "Design-efficient approximate multiplication circuits through partial product perforation," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 24, no. 10, pp. 3105–3117, Oct. 2016, doi: 10.1109/TVLSI.2016.2535398.
[24] W. Liu et al., "Design and analysis of approximate redundant binary multipliers," IEEE Trans. Comput., vol. 68, no. 6, pp. 804–819, Jun. 2019, doi: 10.1109/TC.2018.2890222.