Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

Optimal Design of Multiplier Architectures and Sorting Networks for High-Performance Computing

A. V. S. S. Varma[1], Kasiprasad Mannepalli[2], Dr Akurathi Gangadhar[3]

[1]Research scholar, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
[2]Associate Professor, Department of Electronics and Communication Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, Andhra Pradesh, India
[3]Associate Professor, Department of Electronics and Communication Engineering, UCEN-JNTUK Narasaraopet, Andhra Pradesh, India

In the domain of high-performance computing, the efficiency of arithmetic operations like multiplication holds significant importance for overall system performance. Multipliers, serving as fundamental components, have a profound impact on computational throughput. Similarly, sorting networks are crucial for data organization and manipulation. This study proposes an integrated approach that combines multiplier design principles with sorting network techniques to enhance computational efficiency. By harnessing the strengths of both methodologies, the proposed technique aims to optimize area, power, and delay in arithmetic-intensive applications. Through a thorough investigation of multiplier architectures and sorting algorithms, this paper presents implementations of adders, and exact & approximate compressors with a compression ratio of 4:2 using sorting networks. These findings show substantial gains when integrated into an 8x8 bit exact multiplier, with gains ranging from 28.5% to 49.99% in the area to delay product (ADP) and 39.17% to 45.87% in the power to delay product (PDP). Integration of these enhancements into an approximate 8x8 bit multiplier yields similar results: 1.5% to 38.34% in the area to delay product (ADP) and -7.79% to 39.12% in the power to delay product (PDP).

Keywords: Adders, Exact & Approximate compressor with the ratio of 4:2, Multiplier, Sorting network.

1. Introduction

In the realm of digital signal processing (DSP) applications and microprocessors, multiplication stands as a fundamental operation frequently employed. Typically, the multiplication process of any multiplier involves three key steps: generating partial products, consolidating partial products, and summing up the final vector. The conventional array multiplier, while widely used, suffers from significant delay, which can be mitigated through techniques like the Dadda multiplier or the Wallace tree approach [1-3]. These methods have been extensively studied and applied in various hardware applications, including public-key cryptosystems where modular multipliers are constructed using large number multipliers based on algorithms such as Toom-Cook [5] or Karatsuba [4]. Both Toom-Cook and Karatsuba algorithms have been thoroughly investigated and utilized in hardware applications like the number theoretic transform (NTT) [6-7].

The Wallace tree multiplier [1], its simplified version, and the Dadda multiplier expedite Summation by utilizing the full-adder as a compressor with a ratio of 3:2, thus reducing time consumption and generating a carry- save structure. Several studies have investigated various architectures aimed at reducing latency and accelerating summation, including the development of compressors with compression ratios exceeding (3,2). These compressors limit the number of rows to two by taking into account the carry bits betweenneighboring columns.

Some research has explored compressors in (4,2) [9] and (5,2) [10] configurations and their effectiveness in reducing delays. However, the extensive use of XOR gates in these compressors complicates logical simplification, rendering the technique less efficient.

In contemporary applications, particularly in disciplines requiring inherent error tolerance like artificial intelligence, image processing, and recognition,approximations and the widespread adoption of approximate multipliers are becoming increasingly necessary. Approximate multipliers consist of approximate compressors or simpler circuits that replace portions of exact compressors, introducing some inaccuracies but enhancing power, speed, and area efficiency. Previous studies [16-22] have addressed the utilization of complete adders in constructing high-speed approximation (4:2) compressors.

In this paper, we propose an adders and an exact & approximate compressor with a compression ratio of 4:2 using sorting networks. These components are utilized in the design of an 8x8 multiplier, implemented in 90nm and 180nm CMOS technology.

The rest of this paper is organized as follows: Existing Methodology, Proposed Methodology, Sorting Network Review, Results and Comparisons,Conclusion and References.

2. EXISTING METHODOLOGY:

Half-Adder:

A half-adder is a basic digital circuit, illustrated in Figure 1, which adds two single-bit binary values and generates two single-bit outputs referred to as carry (C) and sum (S).

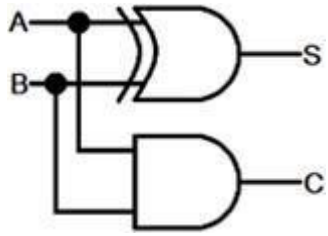

FIGURE 1: Half-Adder

Full-Adder:

Similar to a half-adder, a full-adder is a simple basic circuit thatcombines three single-bit inputs (A, B, & Ci), as shown in Figure 2. Itgenerates two single-bit outputs, namely carry (C) and sum (S).

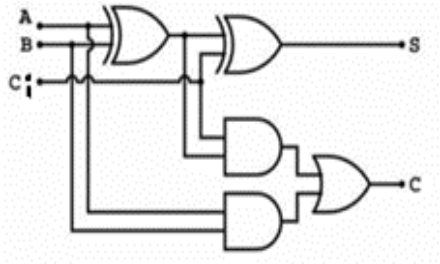

FIGURE 2: Full-Adder

Exact (4:2) Compressor:

The exact compressor with a compression ratio of 4:2 consists of fiveinput bits, three output bits, and two full-adders. This configuration maintainstotal accuracy without compromising any information. Figure 3 illustrates the block layout of the exact compressor with a compression ratio of 4:2.

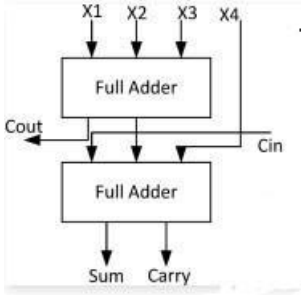

FIGURE 3: Exact Compressor of 4:2 compression ratio

3. REVIEW OF SORTING NETWORK:

A highly efficient parallel hardware algorithm used for sorting data is known as a sorting network. These networks ensure that every element is inthe correct order by comparing and

exchanging elements in a predeterminedsequence of comparisons. Unlike some software methodologies such as Quick and Merge sort, sorting networks do not depend on the size of the input data. Instead, they focus on the optimal arrangement of comparison operations, facilitating reliable and rapid sorting in hardware implementations. Sorting networks are particularly effective for sorting dataconsisting of 1-bit values, adhering to the well-known 0, 1 principle. In this study, we exclusively employ unary sorting techniques.

Data sorter for 1-bit:

The 1-bit data sorter primarily consists of a compare and exchange (CE) block. As depicted in Figure 4, this block comprises 2 single-bit inputs& outputs. The sorter ensures that the higher output is consistently positionedat the top, while the lower output is positioned at the bottom.

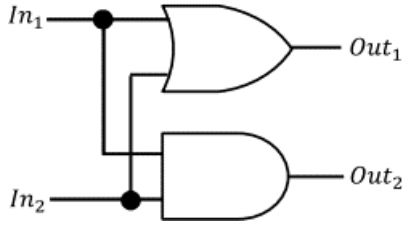

FIGURE 4: 1-bit Sorter

Three-way and Four-way Sorting networks:

The sorting network structure of 3&4 way depicted in Figure 5, is constructedusing a series of CE blocks arranged in a specific order. In the diagram, eachvertical line represents a CE block. As illustrated, the output is presented in descending order, with the higher values appearing first and the lower valuesfollowing, irrespective of the 1-bit input.

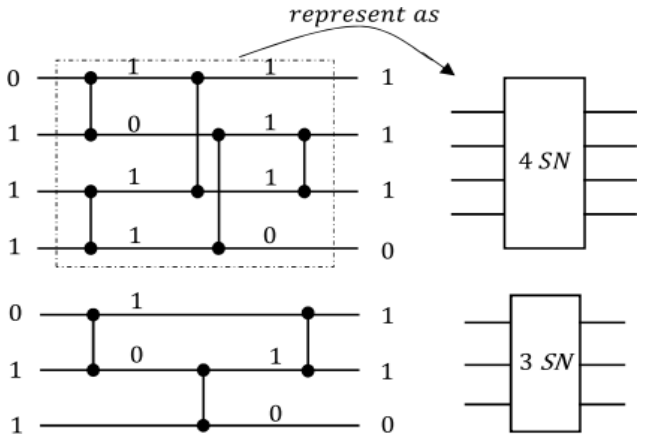

FIGURE 5: Three-way and Four-way sorting networks

4. PROPOSED METHODOLOGY:

Half-Adder:

Here, we are constructing a half-adder using a sorting network and a 1-bit sorter. The half-

adder has 2 inputs & 2 outputs. As shown in Figure 6, one output, carry (C), is provided directly after sorting, while the sum (S) outputis provided after connecting to an external circuit. This showcases the half- adder's operation without any loss of information, similar to the conventional design.

Below is the logical expression for Sum and Carry in a half-adder:

$Sum = (A|B)$

$Carry = B$

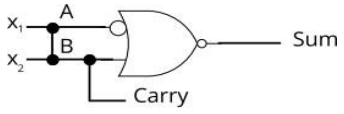

FIGURE 6: Half-Adder

Full-Adder:

This full-adder is constructed using a three-way sorting network (3SN). It features three inputs & two outputs, labeled as sum (S) & carry (C). The carry

(C) bit value is provided immediately after sorting the input data, while the sum (S) output is delivered after connecting an external circuit, as shown in Figure 7.

Their equations are as follows:

Sum = (A&B—)|C

Carry = B

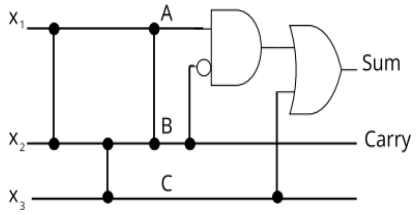

FIGURE 7: Full-Adder

Exact (4:2) Compressor:

The exact compressor with a ratio of 4:2 is constructed using a four- way sorting network (4SN). It comprises four inputs: one carry (Cin) & threeoutputs (Sum, Carry, Cout). In Figure 8, the operation is divided into three stages: sorting or half-sorting, an external circuit, and a full-adder. Through these stages, the output is obtained without any loss of data. The outputs of the half-sort stage are labeled as A, B, C, & D. Plugging these outputs into the external circuit yields the following equations:

–

$S_0 = (A\&B)|D$

$C_{out} = B$

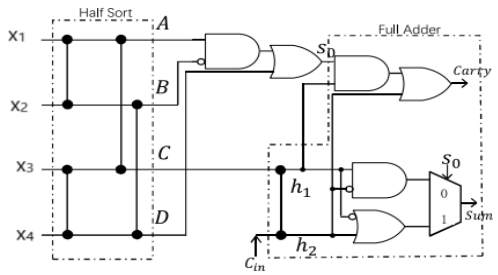

FIGURE 8: Exact Compressor with a compression ratio of 4:2

In the full-adder part, we utilize S_0 and C to compute the final outputs, Sum and Carry, as depicted in Figure 8. The equations for the finaloutputs are as follows:

$h_1 = C|C_{in} h_2 = C\&C_{in}$

$Carry = (S_0\&h_1)|h_2$

– –

$Sum = S_0? (h_1|h_2): (h_1\&h_2)$

Approximate [4:2] Compressor:

In Figure 9, we have designed an approximate 4:2 compressor using a4SN. This compressor consists of two phases: the sorting network and an external circuit. The outputs of the sorting network stage are labeled as A and

D. The D stage of the sorting network generates the smallest of the given inputs. By omitting this stage, we create an approximate compressor with a ratio of 4:2 Figure 9 leads to the following equations:

–

$S_0 = (A\&h_1)| h_2$

$Carry = h_1$

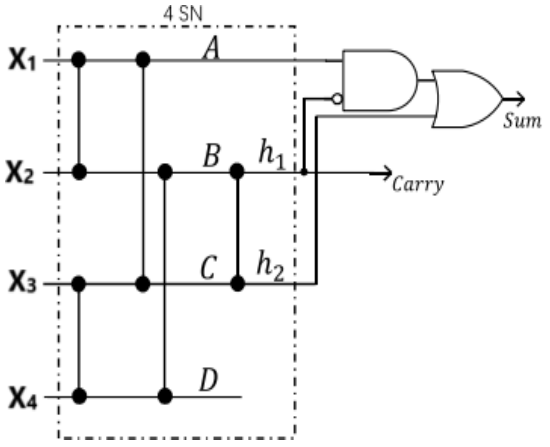

FIGURE 9: Approximate Compressor with a compression ratio of 4:2

X4..X1	YANG		STROLLO		LIN		PROPOSED	
	CS	E	CS	E	CS	E	CS	E
0000	00		00		00		00	
0001	01		01		01		01	
0010	01		01		01		01	
0011	10		10		10		10	
0100	01		01		01		01	
0101	10		10		10		10	
0110	10		10		10		10	
0111	11		11		11		11	
1000	01		01		01		01	
1001	10		10		10		10	
1010	10		10		10		10	
1011	11		11		11		11	
1100	10		10		10		10	
1101	11		11		11		11	
1110	11		11		11		11	
1111	11	-1	11	-1	10	-2	11	-1

TABLE 1: Truth Table of 4:2 Approximate Compressor

In addition to the proposed 4:2 approximate compressor, we are examining

three more 4:2 approximate compressors: Yang, Strollo, and Lin. Their truthtable is outlined in Table 1.

5. RESULTS AND COMPARISONS:

To assess the performance of the designed adders, compressors, and multipliers, we implemented them in Verilog HDL. Subsequently, they were synthesized using the Cadence (Genus) tool with 90nm and 180nm technology. The synthesis results are provided below

Results of Half-Adder:

According to the results presented in Table 2, the proposed half-adder designdemonstrates significant improvements compared to the previous design. Specifically, it outperforms the previous design by more than 17%, 22%, and 30% in terms of latency, ADP & PDP respectively. Additionally, it utilizes substantially less area and power.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	existing	12.867	0.378311	0.174	2.238858	0.065826	17.24%	22.11%	30.46%
	proposed	12.110	0.317869	0.144	1.74384	0.045773	-	-	-
180NM	existing	39.917	1.16932	0.3	11.9751	0.350796	28.33%	34.30%	40.94%
	proposed	36.590	0.963565	0.215	7.86685	0.207166	-	-	-

TABLE 2: Synthesis Results of Half-Adder

Results of Full-Adder:

Table 3 showcases the results of the proposed full-adder design, whichnot only consumes less area and power but also surpasses traditional designsin terms of latency, ADP & PDP by over 29%, 50%, and 72% respectively.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	existing	30.276	1.05483	0.434	13.13978	0.457796	29.95%	54.47%	72.03%
	proposed	19.679	0.421111	0.304	5.982416	0.128018	-	-	-
180NM	existing	93.139	3.04696	0.746	69.48169	2.273032	34.18%	50.63%	95.19%
	proposed	69.854	0.22258	0.491	34.29831	0.109287	-	-	-

TABLE 3: Synthesis Results of Full-Adder

Results of Exact (4:2) Compressor:

Table 4 presents the outcomes of the suggested exact compressor design, featuring a ratio of 4:2, demonstrating its superior efficacy concerning both area utilization & power efficiency. Furthermore, it surpasses previous designs by over 4% in delay, 22% in ADP, and 37% in PDP.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	existing	65.093	2.38763	0.7	45.5651	1.671341	4.28%	22.09%	37.96%
	proposed	52.983	1.54761	0.67	35.49861	1.036899	-	-	-
180NM	existing	199.584	7.09055	1.208	241.0975	8.565384	34.35%	44.20%	50.81%
	proposed	169.646	5.31304	0.793	134.5293	4.213241	-	-	-

TABLE 4: Synthesis Results of Exact (4:2) Compressor

Results of Approximate (4:2) Compressor:

The presented Table 5 illustrates the comparative performance metrics of the suggested approximate compressor with a ratio of 4:2 against three more existing approximate compressors of 4:2 ratio in both 90nm and 180nm technologies. It highlights significant reductions in delay, ADP,and PDP when compared to existing techniques.

Our thorough analysis indicates that the proposed design outperformsLin, Yang, and Strollo compressors in the context of area and & power. However, concerning delay, the suggested design exhibits 54.27%, 12.95%,

and 30.85% greater delay, ADP, and PDP respectively than Lin in 90nm technology. Nevertheless, the proposed architecture demonstrates improvedefficacy concerning delay, ADP,& PDP when the technology is upgraded to180nm.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	lin	42.386	1.10082	0.304	12.88534	0.334649	-54.27%	-12.95%	-30.85%
	yang	39.359	1.07822	0.411	16.17655	0.443148	-14.11%	10.02%	1.18%
	strollo	40.116	1.26757	0.645	25.87482	0.817583	27.28%	43.75%	46.43%
	proposed	31.033	0.933718	0.469	14.55448	0.437914	-	-	-
180NM	lin	133.056	4.2484	0.556	73.97914	2.36211	-23.74%	13.38%	6.65%
	yang	119.75	4.41569	0.671	80.35225	2.962928	-2.53%	20.25%	25.58%
	strollo	136.382	5.37506	1.114	151.9295	5.987817	38.24%	57.82%	63.17%
	proposed	93.139	3.20475	0.688	64.07963	2.204868	-	-	-

TABLE 5: Synthesis Results of Approximate (4:2) Compressor

Design of 8x8 Multiplier:

To demonstrate the effectiveness of the suggested designs, we implement an 8x8 bit multiplier as an application. The structural design of the 8x8 bit multiplier is demonstrated in Figure 10.

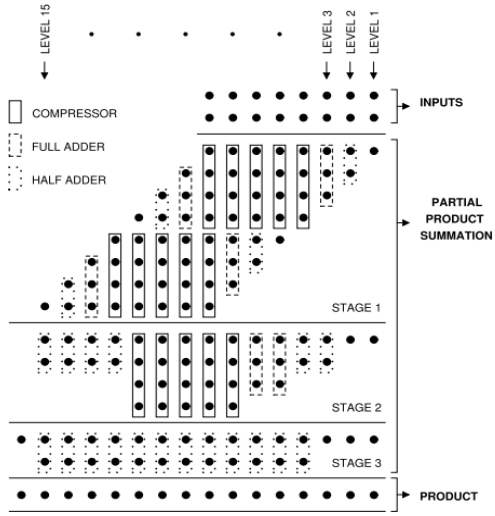

FIGURE 10: 8x8 Multiplier

Results of 8x8 Multiplier:

Table 6 presents the synthesis results of an 8x8 bit exact multiplier. The existing architecture of the 8x8 bit multiplier integrates standard adders& 4:2 exact compressor designs. In contrast, the suggested architecture incorporates a full-adder, a half-adder, & an exact 4:2 compressor based on sorting networks, all arranged within the 8x8 bit multiplier.

The proposed 8x8 bit multiplier utilizing sorting networks demonstratessuperior performance in terms of ADP and PDP.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	existing	1987.619	89.007	4.64	9222.552	412.9925	12%	28.55%	39.17%
	proposed	1613.711	61.5281	4.083	6588.782	251.2192	-	-	-
180NM	existing	6067.354	285.177	8.327	50522.86	2374.669	41.31%	49.99%	45.87%
	proposed	5169.226	262.984	4.887	25262.01	1285.203	-	-	-

TABLE 6: Synthesis Results of Exact 8x8 Multiplier

Table 7 presents the synthesis results for an 8x8 bit inaccurate multiplier. The proposed 8x8-bit approximate multiplier demonstrates superior performance in the context of area & power efficiency than the preceding three solutions. However, it exhibits a longer delay compared to Lin and Yang.

Despite this,the proposed approximate 8x8 bit multiplier outperforms existing approximate 8x8 bit multipliers in terms of ADP and PDP except Lin multiplier design in PDP.

NM TECHNOLOGY	METHOD	AREA (μm^2)	POWER (μ W)	DELAY (ns)	ADP (ns . μm^2)	PDP (ns . μW)	IMPROVE(MAX)		
							IN DELAY	IN ADP	IN PDP
90NM	lin	1354.094	44.0318	1.198	1622.205	52.7501	-19.94%	1.57%	-7.79%
	yang	1308.68	43.9108	1.312	1716.988	57.61097	-9.52%	7.00%	1.29%
	strollo	1320.034	45.5555	1.664	2196.537	75.80435	13.64%	27.30%	24.98%
	proposed	1111.129	39.5704	1.437	1596.692	56.86266	-	-	-
180NM	lin	4177.958	143.475	1.945	8126.128	279.0589	-10.53%	10.05%	-6.35%
	yang	3978.374	148.118	2.025	8056.207	299.939	-6.17%	9.27%	1.05%
	strollo	4227.854	173.871	2.804	11854.9	487.5343	23.32%	38.34%	39.12%
	proposed	3399.581	138.04	2.15	7309.099	296.786	-	-	-

TABLE 7: Synthesis Results of Approximate 8*8 Multiplier

6. CONCLUSION:

In the context of this study, we introduce adders, both exact & approximate compressors with a compression ratio of 4:2 utilizing sorting networks. When integrated into an 8x8 bit exact multiplier, our devised approach demonstrates superiority in ADP & PDP, showing enhancements of 28.5% to 49.99% for ADP and 39.17% to 45.87% for PDP, respectively.

Similarly, with an 8x8 bit approximate multiplier, improvements of 1.5% to38.34% for ADP and -7.79% to 39.12% for PDP are observed.

While there exists an anomaly in the behavior of the 8x8 bit approximate multiplier concerning PDP for the Lin multiplier design, the overarching trend suggests that PDP performance evolves positively with the advancement of nanometer (NM) technology.

Thus, we can conclude that sorting network strategies exhibit improved performance as NM technology advances.

References

[1]. C. S. Wallace, "A Suggestion for a Fast Multiplier," in IEEE Transactions on Electronic Computers, 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," in IEEE Transactions on Computers, vol. 59, no. 8, pp. 1134-1137, Aug. 2010, doi: 10.1109/TC.2010.103.

[3]. L. Dadda, "Some schemes for fast serial input multipliers," 1983 IEEE 6th Symposium on Computer Arithmetic (ARITH), Aarhus, Denmark, 1983, pp. 52-59, doi: 10.1109/ARITH.1983.6158074.

[4]. P. L. Montgomery, "Five, six, and seven-term Karatsuba-like formulae," in IEEE Transactions on Computers, vol. 54, no. 3, pp. 362-369, March 2005, doi: 10.1109/TC.2005.49.

[5]. J. Ding, S. Li and Z. Gu, "High-Speed ECC Processor Over NIST Prime Fields Applied With Toom–Cook Multiplication," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 3, pp. 1003-1016, March 2019, doi: 10.1109/TCSI.2018.2878598.

[6]. 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.

[7]. 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.

[8]. S. Asif and Y. Kong, "Analysis of different architectures of counter based Wallace multipliers," 2015 Tenth International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 2015, pp. 139-144, doi: 10.1109/ICCES.2015.7393034.

[9]. 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, Croatia, 2013, pp. 66-69.

[10]. M. H. Najafi, D. J. Lilja, M. D. Riedel and K. Bazargan, "Low-Cost Sorting Network Circuits Using Unary Processing," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 26, no. 8, pp. 1471-1480, Aug. 2018, doi: 10.1109/TVLSI.2018.2822300.

[11]. 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, Croatia, 2014, pp. 80-84, doi: 10.1109/MIPRO.2014.6859537.

[12]. W. Guo and S. Li, "Fast Binary Counters and Compressors Generated by Sorting Network," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 29, no. 6, pp. 1220-1230, June 2021, doi: 10.1109/TVLSI.2021.3067010.

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

[14]. M. Mehta, V. Parmar and E. Swartzlander, "High-speed multiplier design using multi-input

counter and compressor circuits," [1991] Proceedings 10th IEEE Symposium on Computer Arithmetic, Grenoble, France, 1991, pp. 43-50, doi: 10.1109/ARITH.1991.145532.

[15]. T. Satish and K. S. Pande, "Multiplier Using NAND Based Compressors," 2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology (IEMENTech), Kolkata, India, 2019, pp. 1-6, doi: 10.1109/IEMENTech48150.2019.8981067.

[16]. 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," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 67, no. 9, pp. 3021-3034, Sept. 2020, doi: 10.1109/TCSI.2020.2988353.

[17]. Z. Yang, J. Han and F. Lombardi, "Approximate compressors for error- resilient multiplier design," 2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), Amherst, MA, USA, 2015, pp. 183-186, doi: 10.1109/DFT.2015.7315159.

[18]. O. Akbari, M. Kamal, A. Afzali-Kusha and M. Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 4, pp. 1352-1361, April 2017, doi: 10.1109/TVLSI.2016.2643003.

[19]. K. Manikantta Reddy, M. H. Vasantha, Y. B. Nithin Kumar and D. Dwivedi, "Design of Approximate Booth Squarer for Error-Tolerant Computing," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 28, no. 5, pp. 1230-1241, May 2020, doi: 10.1109/TVLSI.2020.2976131.

[20]. S. Venkatachalam and S. -B. Ko, "Design of Power and Area Efficient Approximate Multipliers," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 25, no. 5, pp. 1782-1786, May 2017, doi: 10.1109/TVLSI.2016.2643639.

[21]. G. Zervakis, K. Tsoumanis, S. Xydis, D. Soudris and K. Pekmestzi, "Design-Efficient Approximate Multiplication Circuits Through Partial Product Perforation," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 24, no. 10, pp. 3105-3117, Oct. 2016, doi: 10.1109/TVLSI.2016.2535398.

[22]. P. J. Edavoor, S. Raveendran and A. D. Rahulkar, "Approximate Multiplier Design Using Novel Dual-Stage 4:2 Compressors," in IEEE Access, vol. 8, pp. 48337-48351, 2020, doi: 10.1109/ACCESS.2020.2978773.