A high speed and power efficient multiplier based on counter-based stacking

Chukkaluru Ravi Shankar Reddy[1], Padavala Venkata Gopi Kumar[2], Radhakrishnan Manikandan[3], Kuruva Bhavana[4]

[1]Department of Electronics and Communication Engineering, Sreenidhi University, Hyderabad, India
[2]Department of Electronics and Instrumentation Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
[3]Department of Electronics and Instrumentation Engineering, Annamalai University, Chidambaram, India
[4]Department of Electronics and Communication Engineering, Malla Reddy College of Engineering and Technology (MRCET), Hyderabad, India

Article Info

Article history:

Received Jan 3, 2023
Revised Jun 24, 2023
Accepted Jul 2, 2023

Keywords:

Compressor
Counter
Latency
Low power
Stacking

ABSTRACT

High speed and competent addition of various operands is an essential operation in the design any computational unit. The swiftness and power competence of multiplier circuits plays vital role in enlightening the overall performance of microprocessors. Multipliers play crucial role in the design of arithmetic logic unit (ALU) or any digital signal processor (DSP) that are effectively employed for filtering and convolution operations. The process of multiplication either binary numbers or fixed-point numbers yields in enormous partial products that are to be added to get final product. These partial products in number and the process of summing up partial products dictate the latency and power consumption of the multiplier design. Here, we present a novel binary counter design that hires stacking circuits, that groups all logic "1" bits as one, followed by a novel symmetric method to merge pairs of 3-bit stacks into 6-bit stacks and then changes them to binary counts. This results in drastic improvements in power and area utilization of the multiplier. Additionally, this paper also focuses on implementation of novel approximate compressor and exploits the same for the design of approximate multipliers that can be effectively employed in any electronic systems that are characterized by power and speed constraints.

Corresponding Author:

Padavala Venkata Gopi Kumar
Department of Electronics and Instrumentation Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, Telangana, India
Email: gopikumar_pv@vnrvjiet.in

1. INTRODUCTION

The general process of multiplication includes generation partial products and all these partial products have to be added to get end result. The figure of partial products obtained depends on the length of the numbers that are being multiplied. The increase in the length of numbers that are being multiplied proportionally increases the number of partial products and this increase in partial products proportionally dictates the complexity of adders which dictates the circuit performance parameters like power and speed. This creates necessity to design a novel adder and multiplier circuits that are efficient with respect to power and speed. Moreover, these multiplier circuits are of very much needed in the design of arithmetic logic unit (ALU) and digital signal processor that predominantly work for filtering and convolution operations. Hence, the design of power and speed efficient adder and multiplier circuits also aids in improvements in performance of various

DSP and embedded processors. This section discusses in detail about various mechanisms that can be effectively applied for achieving high speed multiplications and their technological advancements. Many researchers conclude that the speed improvements in multipliers can be obtained by various methodologies like using hybrid structures, reducing the partial products, by employing counters and compressors. Similarly, to target power savings the multiplier architecture employs exclusive OR (XOR) and exclusive-NOR (XNOR) gates and also tries to achieve power saving by reducing the number adders present in the architecture and by employing counter and compressors.

The design of hybrid high speed carry select adder using carry-lookahead adder (CLA) is presented in [1]. On similar lines analysis high speed radix 4 multiplier using Shannon adder that is suitable of digital signal processor (DSP) applications is presented in [2]. A similar sort of hybrid adder using analog and digital circuits is presented in [3]. The design of adders that are aimed to achieve power savings is given in [4], [5]. Design of highspeed multiplier using binary counters based on symmetric stacking is given in [6], [7] aims at achieving improvements in its speed by targeting the delays across critical paths. The new design of 7-2 and 5-2 ultra speed compressors [8] leave faith that further improvements speed can achieved by employing the with regular structures of high speed multipliers. Recently various versions of Wallace tree structures [9] were implemented by the use of various other adders like Kogge stone adder, Sklansky adder, Brent Kung adder, Ladner Fischer adder and Han Carlson adder to speed multiplication process by employing parallel prefix adders.

The well-known techniques like Wallace tree [10] and Dada tree [11] have successfully employed row compression techniques to achieve improvements in power and speed. A new design of column compression technique has been exploited [12]. High speed multiplier design by the use of counters and compressors is given in [13] offers significant step up over the one that is implemented by the use of (3.2) counter with respect to area overhead. Implementation of algorithmic Wallace tree multiplier using high speed counters is presented in [14] proves to be a superior strategy for the aiming power efficient high speed multiplication.

The new design Wallace tree multiplier [15] that could give significant improvements in overhead can be designed by employing majority logic. Another version of highspeed multiplier that employs CLA in the structures of Wallace tree and dada tree multiplier is given in [16]. The detailed study of various highspeed adders were given in [17] conclude that dada multiplier is quite faster than Wallace tree multiplier. The analysis of different counter-based architectures of Wallace tree multipliers is presented in [18] yields inference that counter- based multipliers achieves higher speed of operations while providing significant optimizations in area overhead.

According to Lin [19] proves that the significant improvements in speed can also be attained by employing stage reduced partial product reduction network that is built using parallel counters and shift compressors. The improvements in speed can also be achieved by the use of irreducible pentonomials [20] which is special case Galois field multipliers. Architectures that aim at high speed and low power consumption using multiplication were presented in [21], [22]. On similar lines, the design of 3-2 counter and 4-2 compressor designs that are well suitable for fast multiplication and the designs of low power 4-2 and 5-2 compressors are given in [23]–[25]respectively. The design of 4-2 compressor using XOR and XNOR is presented in [26]. The design of 7-2 compressor is presented in [27] gives considerable improvements in speed and power is by minimizing the delays associated with critical path. A 1.2-ns 16×16-bit binary multiplier using high speed compressors is presented in [28].

2. METHOD

The proposed multiplier is designed by employing the 6-3 compressor for reducing partial products. The 6-3 compressor is designed based on principle of stacking and finally the stacked count is converted into binary count. Here first we discuss complete details regarding counter design section 2.1, then detailed discussion on the process of stacking is carried out in section 2.2 and finally we end section 2.3 by the discussing the process involved in converting the stacked output to binary count.

2.1. Counter design

Figure 1 gives design details of 6-3 compressor. The basic block diagram of 6-3 compressor is given in Figure 1(a) that works on strategy of stacking. The operation of 6-3 is pretty straight forward; among the given six inputs first three bits (A0, A1, and A2) are given to one full adder1 (FA1) and the remaining three bits (A3, A4, and A5) are given to the other full adder (FA2). The sum output of FA1 and FA2 are further given to the half adder (HA1) to compute final sum output S where the carry outputs of FA1, FA2 and HA1 are given to the full adder 3 (FA3) to obtain final carry outputs C1 and C2

The schematic of 6:3 counter is depicted in Figure 1(b) is implemented by the use of CLA concept to make use of propagate and generate signals to speed up addition processes and this can aid in improving the latency of the multiplier. The P (propagate) and G (generate) equations of 6-3 counter are given in (1) and (2) respectively. The Boolean expressions for S (Sum), C1 (Carry Out1), and C2 (Carry out 2) are given by the (3)-(5) respectively.

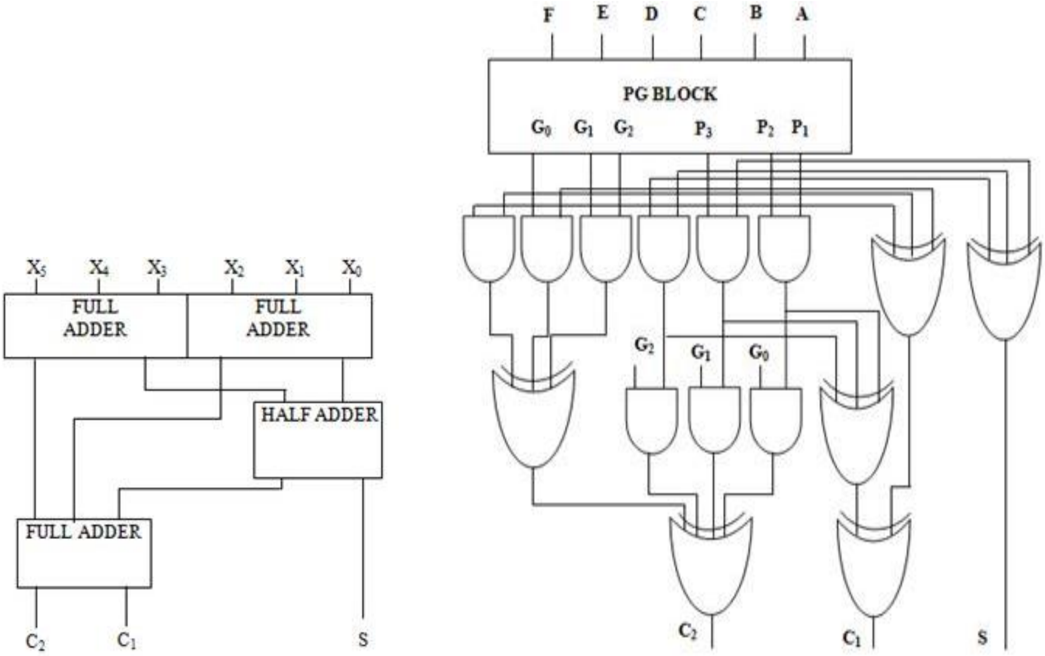
$$P0 \; = \; A \; B \; P1 \; = \; C \; D \; P2 \; = \; E \; F \tag{1}$$

$$G0 \; = \; A.B \; G1 \; = \; C.D \; G2 \; = \; E.F \tag{2}$$

$$S \; = \; P0 \; P1 \; P2 \tag{3}$$

$$C1 \; = \; (P0.P1 \; P0.P2 \; P1.P2) \; (G0G1 \; G2) \tag{4}$$

$$C2 \; = \; (G0.G1 \; + \; G0.G2 \; + \; G1.G2) + \big((P0.P1).G2\big) + \\ ((P0.P2).G1) \; + \; ((P1.P2).G0) \tag{5}$$



(a)

(b)

Figure 1. 6-3 compressor (a) basic principle and (b) circuit diagram

2.2. Bit stacking

The stacking is a process of grouping all input logic 1's together. After stacking, all these stacked bits are transformed to binary count to get 6-bit count. Initially stage 3-bit stacking circuits are employed to obtain three-bit stack then all employed three bits stacks are merged to obtain 6-bit stack. The basic stacking circuit and the process of stacking is given in Figure 2. Figure 2(a) gives 3-bit stacking circuit in which P1, P2 and P3 are given as inputs to the three-bit stacking circuit that yields Q1, Q2 and Q3 as outputs. As we are grouping logic 1's together, the total number of logic 1 bits in the output is same as total number of logic 1's at the input. The processes of grouping logic 1's together includes, grouping of all logic 1's to the left followed by the logic 0's. The outputs Q1, Q2 and Q3 of the 3-bit stacking circuit are characterised by (6), (7) and (8) respectively.

$$Q1 \; = \; P1 \; + \; P2 \; + \; P3 \tag{6}$$

$$Q2 \; = \; P0P1 \; + \; P0P2 \; + \; P1P2 \tag{7}$$

$$Q3 \; = \; P0P1P2 \tag{8}$$

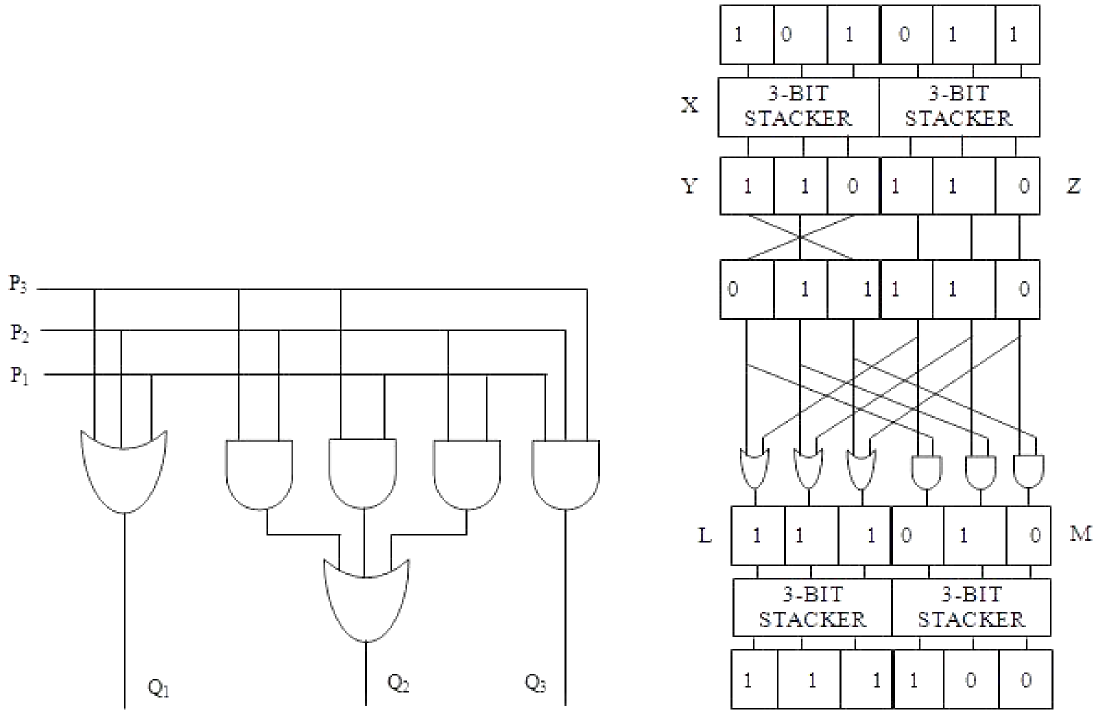
From the above functions it is quite clear that the output Q1 is logic 1 if any of its input is at logic1, output Q2 is logic l if at least two of its inputs are at logic 1 and output Q3 is logic1 if and only if all of its inputs are at logic 1. Now the outputs of two three-bit stackers are merged into binary count to obtain 6-bit count. To detailed explanation of these merging processes is illustrated in Figure 2(b). Now, let us assume there are six inputs X1, X2, X3….., X6. These six inputs are divided into two groups of 3bits of each and these three bits are stacked by employing 3-bit stacking circuits. Let X1, X2, and X3 are stacked to signals that is to say Y1, Y2 and Y3, and X4, X5, and X6 are stacked to signals named Z1, Z2, and Z3.

From an example illustrated in the Figure 2(b) it can be noticed that there is train of logic 1’s bounded by logic 0 bits. To get proper stack, we have to move all logic 1’s positioned to left followed by logic 0’s. To get this we employ two more three-bit stacking circuits which are fed by the output of merged six bit stacker i.e., Y3, Y2, Y1, Z1, Z2, and Z3. To have a better understanding these outputs obtained from the example are represented using two three-bit vectors namely L (L1, L2, and L3) and M (M1, M2, and M3) which are connected to two three-bit stacking circuits. These two 3-bit stacking circuits are combined to get another 6-bit stacking circuit. To have proper stack operation i.e to fill all logic 1’s positioned to left followed by logic 0’s, we apply strategy of filling vector L with logic 1’s before filling vector M. Hence, we define expressions based up on the requirement of proper stack operation.

$$L1 = Y3 + Z1 \tag{9}$$

$$L2 = Y2 + Z2 \tag{10}$$

$$L3 = Y1 + Z3 \tag{11}$$



(a) (b)

Figure 2. Stacking (a) circuit and (b) processes

In a ‘M’ bit vector if the total number of 1’s are less than are equal to three places then all M bits will be filled with logic 0. This drives few of the AND gates in the stacking with logic 1 as their inputs which aids in crafting power efficient architecture. For better understanding of the logic behind the crafting of stacking circuit notice that L1 L2 L3 and M1 M2 M3 will contain equal number of 1’s as input with only difference being that L bits are crammed with logic 1’s ahead of any of M bits.

2.3. Conversion of stacked bits to binary count

The successful implementation 6-3 counter requires conversion of stacked bits into binary counts. The intermediate values of Y, Z and M are employed to achieve this conversion. We know that the outputs C2, C1, and S are in binary representation of number of 1's present at the input of 6-bit stacker. Thus, output S can be determined by determining parity of the output present at initial layer of 3-bit stacker. If the number of 1's in X1, X2, and X3 are '0' or '2' then it results in even parity in Y. On the other hand, if the number of '1's in X4, X5, and X6 are '0' or '2' then it results in even parity in Z. Thus, to indicate even parity in Y and Z. Here Ye and Ze are used to represent even parity in Y and Z respectively. The Ye and Ze of Y and Z bits are given by (12) and (13) respectively.

$$Y_e = Y_1^c + Y_2Y_3^c \tag{12}$$

$$Z_e = Z_1^c + Z_2Z_3^c \tag{13}$$

The sum (S) indicates odd parity overall input bits (XOR operation) in other words addition of two numbers with distinct parities is odd. Although there is XOR gate to obtain sum is present in 6-3 compressor, this XOR gate is not associated with the critical path. Thus Sum (S) can be computed by the (14).

$$S = Y_eXORZ_e \tag{14}$$

On same lines to obtain C1, it has to be noted that C1 will be logic 1 for the counts of 2,3or 6. This gives raise to two cases. One is we have to verify at least two but not more than three inputs. For this we employ, Y, Z, M vectors. To verify for at least two inputs, we have to verify stack of length two. This may be done from crest level stacker or from two stacks whose length is one and this yield $Y + Z + Y_1Z_1$. On the other hand we have to verify that there are no more inputs than three and we should confirm that all bits of M are reset and M vector is only set in which the inputs are not more than three, this gives $(M_1+M_2+M_3)^c$. The other is that we have to verify all six inputs as logic '1'. This can be done by verifying all bits associated with Y and Z vectors. As Y and Z are bit stack it would be sufficient to verify right most bit in the stack. This yield $C1 = Y_2 + Z_2 + H_1I_1$. The computation of C2 is be easily done as its function is to set every time when there are minimum of 4-bits set, which gives $C2 = M_2 + M_2 + M_3$.

3. RESULTS AND DISCUSSION

The proposed architecture's namely, 6-3 compressor and the approximate multiplier that has designed with the aid of 6-3 compressor. The functional verification is carried out using modelsim and its implementation is carried in Xilinx to extract various features like power, area and speed. Here, power is expressed in mW, area is expressed in terms of LUT's and where as speed is expressed in nS.

3.1. Power consumption

The power consumed by the various architectures that are considered for experimental purpose are given in Table 1. The architectures column gives the details regarding the various multiplier architectures of interest. Similarly, the columns under static power and dynamic power give details regarding the static and dynamic powers consumed by the corresponding architectures. Finally, column total power gives estimates of total power consumed by an architecture which is almost equal to the summation of static and dynamic powers consumed by an architecture. Here, total power consumed by the proposed architecture is 155 mW. This includes the static power consumption of 33.6 mW and as well as dynamic power consumption of 121.83 mW.

Table 1. Comparison of power consumption of proposed multiplier

Architecture's	Static power	Dynamic power	Power (mW)
Wallace tree multiplier in [12]	33.6	151.74	185.34
Multiplier architecture by employing 4-2 and 5-2 compressor [18]	33.6	138.96	172.58
Binary multiplier based on stacking [6]	33.6	136.82	170.42
Proposed approximated binary multiplier	33.6	121.83	155.43

3.2. Area overhead

The overhead attained by various architectures that are considered for experimental purpose are given in Table 2. The column under architectures gives the details regarding the various multiplier architectures that on consider for the experimental work. The columns of total no. of 4 input lookup tables (LUT's) used and number of slices gives details regarding the number of 4-bit LUT’s and the slices that are employed in the architecture design. Finally, the column of area overhead gives the details regarding number equivalent gate counts of LUT’s and slices that are required to implement the given design. Here, the amount area overhead associated with the proposed multiplier in term of its gate equivalents is 1,146.

Table 2. Comparison of area overhead of proposed multiplier

Architecture’s	Total no. of LUT’s	Number of slices	Area overhead (GC)
Wallace tree multiplier in [12]	100	88	750
Multiplier architecture by employing 4-2 and 5-2 compressor [18]	163	101	1,042
Binary Multiplier based on stacking [6]	199	107	1,212
Proposed Approximated binary multiplier	188	100	1,146

3.3. Delay

On same lines delay attained by various architectures that are taken for the comparison purpose are presented in tabulated in Table 3 and the values under column delay gives, the delay attained by the corresponding architecture in nano seconds (nS). The values presented here clearly states that proposed architecture performs better in terms of its speed of operation. The design summary of proposed architecture and all the other architectures that are consider for experimental purpose is given in Table 4 in terms of their performance parameters of power, area overhead and delay. The columns of power, area overhead and delay give the values of power, area overhead and delay attained by the corresponding architecture. The total power consumed by the proposed architecture is of 155 mW, which includes static power consumption of 33.6 mW and dynamic power consumption of 121.83 mW. The amount area overhead associated with the proposed multiplier in term of its gate equivalents is 1,146 which includes 188 4-bit LUT’s and 100 slices. The total delay of incurred by the proposed is of 32.902 ns.

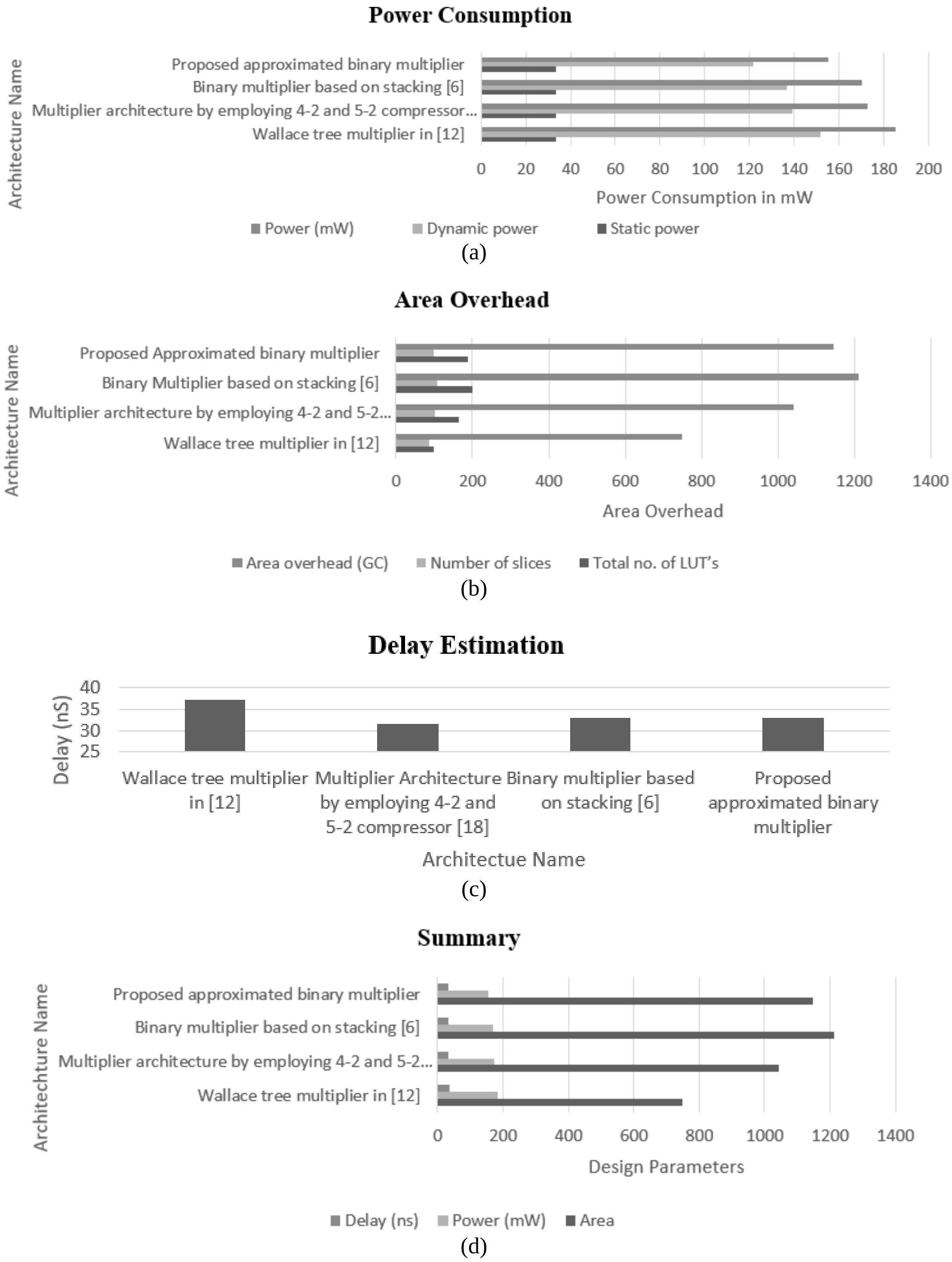
Table 3. Delay comparison of proposed multiplier

Architecture’s	Delay (ns)
Wallace tree multiplier in [12]	37.333
Multiplier Architecture by employing 4-2 and 5-2 compressor [18]	31.494
Binary multiplier based on stacking [6]	32.974
Proposed approximated binary multiplier	32.902

Table 4. Design parameters summary of various architectures

Architecture’s	Area	Power (mW)	Delay (ns)
Wallace tree multiplier in [12]	750	185.34	37.333
Multiplier architecture by employing 4-2 and 5-2 compressor [18]	1,042	172.58	31.494
Binary multiplier based on stacking [6]	1,212	170.42	32.974
Proposed approximated binary multiplier	1,146	155.43	32.902

The Figure 3 gives performance of various simulation parameters for all the architectures that are considered for experimental purpose. Figure 3(a) gives the pictorial representation of static, dynamic and total power consumed by the different architectures that are considered for the experimental purpose. The blue curve shows static power dissipation inoccured by various architectures and similarly, red color and green color cures gives details about dynamic and total power consumptions inoccured by different architectures. Figure 3(b) gives the pictorial representation of hardware requirements of various architectures in terms of LUT’s, slices and gate count. The red and blue color curve gives information about number for LUT’s and slices that utilized in the design of different architectures. Similarly, the green color curve gives information about total equivalent gate count attained by the LUT’s and slices. From these cures, it can be clearly observed that the proposed architecture has less hardware requirements than that of [6] and has slightly more hardware requirement in comparison with [12] and [18] which suffer from increased power consumption and latency. The red color graphs of pictorial representation given in Figure 3(c) indicates delay inoccured by different architectures that are consider for comparison. This clearly states that the proposed architecture attains very less delay compared to all the architectures that are taken for the comparison. Figure 3(d), gives finalized summary of the implemented architecture in terms of power consumption, area overhead and delay.



(a)

(b)

(c)

(d)

Figure 3. Simulation parameter (a) power consumption, (b) area over head, (c) delay, and (d) savings attained in power consumption, area over head and delay

4. CONCLUSION

Here we have presented a new compressor-based multiplier which is successful in eliminating the XOR gates associated with the critical path which in turn resulted in speed improvements. The experimental results convey fact that the proposed multiplier is successful in achieving 12% improvements in speed as compared to other existing architectures. The proposed architecture is also successful in achieving power improvements of 20% as compared to Wallace tree multiplier and 10% of power saving as compared to 5-2 and 4-2 multiplier architecture. Further, larger savings of power and increased speed of multiplier can obtained by employing very low power 7-2 compressors that work at ultra low power values and low power design concepts in the actual design of 7-2 compressors and adders that are employed for compression.

REFERENCES

[1] A. Simson and S. Deepak, "Design and implementation of high speed hybrid carry select adder," in *Proceedings of the 2021 1st International Conference on Advances in Electrical, Computing, Communications and Sustainable Technologies, ICAECT 2021*, Feb. 2021, pp. 1–6, doi: 10.1109/ICAECT49130.2021.9392452.

[2] V. Vijayakumar, K. T. Ilayarajaa, T. Ravi, and M. Sugadev, "Analysis of high speed hybrid full adder," in *Proceedings - International Conference on Artificial Intelligence and Smart Systems, ICAIS 2021*, Mar. 2021, pp. 1641–1645, doi: 10.1109/ICAIS50930.2021.9395998.

[3] N. Taherinejad and A. Abrishamifar, "A new high speed, low power adder; using hybrid analog-digital circuits," in *ECCTD 2009 - European Conference on Circuit Theory and Design Conference Program*, Aug. 2009, pp. 623–626, doi: 10.1109/ECCTD.2009.5275072.

[4] D. Radhakrishnan, "Low-voltage low-power CMOS full adder," *IEE Proceedings: Circuits, Devices and Systems*, vol. 148, no. 1, pp. 19–24, 2001, doi: 10.1049/ip-cds:20010170.

[5] A. Saberkari and S. B. Shokouhi, "A novel low-power low-voltage CMOS 1-bit full adder cell with the GDI technique," *2006 IJME-INTERTECH Conference*, 2006.

[6] D. KavyaShree, P. Samundiswary, and K. V. Gowreesrinivas, "High speed multipliers using counters based on symmetric stacking," in *2019 International Conference on Computer Communication and Informatics (ICCCI)*, Jan. 2019, pp. 1–6, doi: 10.1109/ICCCI.2019.8822185.

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

[8] 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 Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 6, pp. 1403–1412, Jun. 2020, doi: 10.1109/TVLSI.2020.2983458.

[9] Y. D. Ykuntam, K. Pavani, and K. Saladi, "Design and analysis of high speed wallace tree multiplier using parallel prefix adders for VLSI circuit designs," in *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*, Jul. 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225404.

[10] C. S. Wallace, "A suggestion for a fast multiplier," in *Computer Arithmetic: Volume I*, 2015, pp. 150–153.

[11] L. Dadda, "Some schemes for parallel multipliers," *Computer Arithmetic: Volume I*, pp. 137–144, 2015, doi: 10.1142/9789814651578.

[12] Z. Wang, G. A. Jullien, and W. C. Miller, "A new design technique for column compression multipliers," *IEEE Transactions on Computers*, vol. 44, no. 8, pp. 962–970, 1995, doi: 10.1109/12.403712.

[13] M. Mehta, V. Parmar, and E. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits," in *Proceedings - Symposium on Computer Arithmetic*, 1991, pp. 43–50, doi: 10.1109/arith.1991.145532.

[14] S. Asif and Y. Kong, "Design of an algorithmic Wallace multiplier using high speed counters," in *Proceedings - 2015 10th International Conference on Computer Engineering and Systems, ICCES 2015*, Dec. 2016, pp. 133–138, doi: 10.1109/ICCES.2015.7393033.

[15] R. S. Waters and E. E. Swartzlander, "A reduced complexity Wallace multiplier reduction," *IEEE Transactions on Computers*, vol. 59, no. 8, pp. 1134–1137, Aug. 2010, doi: 10.1109/TC.2010.103.

[16] W. Chu, A. I. Unwala, P. Wu, and E. E. Swartzlander, "Implementation of a high speed multiplier using carry lookahead adders," in *Conference Record - Asilomar Conference on Signals, Systems and Computers*, Nov. 2013, pp. 400–404, doi: 10.1109/ACSSC.2013.6810305.

[17] S. Abraham, S. Kaur, and S. Singh, "Study of various high speed multipliers," in *2015 International Conference on Computer Communication and Informatics, ICCCI 2015*, Jan. 2015, pp. 1–5, doi: 10.1109/ICCCI.2015.7218139.

[18] S. Asif and Y. Kong, "Analysis of different architectures of counter based Wallace multipliers," in *Proceedings - 2015 10th International Conference on Computer Engineering and Systems, ICCES 2015*, Dec. 2016, pp. 139–144, doi: 10.1109/ICCES.2015.7393034.

[19] R. Lin, "Fast multiplier schemes using large parallel counters and shift switches," in *Proceedings of the International Conference on High Performance Computing, HiPC*, 1997, pp. 302–308, doi: 10.1109/hipc.1997.634507.

[20] J. L. Imaña, "High-speed polynomial basis multipliers over GF(2m) for special pentanomials," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 63, no. 1, pp. 58–69, Jan. 2016, doi: 10.1109/TCSI.2015.2500419.

[21] S. Veeramachaneni, A. Lingamneni, M. K. Krishna, and M. B. Srinivas, "Novel architectures for efficient (m, n) parallel counters," in *Proceedings of the ACM Great Lakes Symposium on VLSI, GLSVLSI*, Mar. 2007, pp. 188–191, doi: 10.1145/1228784.1228833.

[22] Y. Horima, T. Onomi, M. Kobori, I. Shimizu, and K. Nakajima, "Improved design for parallel multiplier based on phase-mode logic," *IEEE Transactions on Applied Superconductivity*, vol. 13, no. 2 I, pp. 527–530, Jun. 2003, doi: 10.1109/TASC.2003.813924.

[23] S. F. Hsiao, M. R. Jiang, and J. S. Yeh, "Design of high-speed low-power 3-2 counter and 4-2 compressor for fast multipliers," *Electronics Letters*, vol. 34, no. 4, pp. 341–343, 1998, doi: 10.1049/el:19980306.

[24] K. Prasad and K. K. Parhi, "Low-power 4-2 and 5-2 compressors," in *Conference Record of the Asilomar Conference on Signals, Systems and Computers*, 2001, vol. 1, pp. 129–133, doi: 10.1109/ACSSC.2001.986892.

[25] J. Gu and C. H. Chang, "Low voltage, low power (5:2) compressor cell for fast arithmetic circuits," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2003, vol. 2, pp. 661–664, doi: 10.1109/icassp.2003.1202453.

[26] S. Kumar and M. Kumar, "4-2 compressor design with new XOR-XNOR module," in *International Conference on Advanced Computing and Communication Technologies, ACCT*, Feb. 2014, pp. 106–111, doi: 10.1109/ACCT.2014.36.

[27] M. Rouholamini, O. Kavehie, A. P. Mirbaha, S. J. Jasbi, and K. Navi, "A new design for 7:2 compressors," in *2007 IEEE/ACS International Conference on Computer Systems and Applications, AICCSA 2007*, May 2007, pp. 474–478, doi: 10.1109/AICCSA.2007.370924.

[28] A. Dandapat, S. Ghosal, P. Sarkar, and D. Mukhopadhyay, "A 1.2-ns16×16-bit binary multiplier using high speed compressors," *World Academy of Science, Engineering and Technology*, vol. 39, pp. 627–632, 2009.

BIOGRAPHIES OF AUTHORS

Dr. Chukkaluru Ravi Shankar Reddy has received his Ph.D., and M.Tech., degree from Jawaharlal Nehru Technological University Anantapur, Andhra Pradesh, India in 2016, and 2009 respectively. He received his B.Tech., degree from JNT University. His research interest includes digital system, computer aided design, testing and testability and image processing. He is currently working as professor in the Department Electronics and Communication Engineering in Malla Reddy College of Engineering and Technology (Autonomous) which is permanently affiliated to JNTU Hyderabad. He has more than 25 publications that are published in National and International reputed Conferences and Journals. He is reviewer for IEEE transactions very large-scale integration systems and international conference on soft computing and signal processing. He can be contacted at email: crsr@mrcet.ac.in.

Padavala Venkata Gopi Kumar Assistant Professor in the Department of Electronics and Instrumentation Engineering at VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad. He Has 12 years of teaching experience. He Holds a M.Tech. degree in Electronics and Communication Engineering with specialization in Digital Systems and Computer Electronics. His research areas are Image/signal processing, VLSI and embedded systems. He is pursuing Ph.D. in the area of VLSI. His research interests include image/signal processing, IoT, error correction and detection and test pattern generation and recognition. He can be contacted at email: gopikumar_pv@vnrvjiet.in.

Dr. Radhakrishnan Manikandan Assistant Professor in the Department of Electronics and Instrumentation Engineering at Annamalai University, Chidambaram. He have Received Ph.D. from Annamalai University. He also holds a M.Tech. degree in Process Control and Instrumentation. His specialization includes process control, industrial automation and robotics. He has 15 years of teaching experience including 5 years of Research. His research interest includes process control instrumentation, machine learning, automation, image processing, he is a Life Member of Indian Society for Technical Education. He can be contacted at email: rjmani.ei@gmail.com.

Kuruva Bhavana Assistant Professor in the Department of Electronics and Communication Engineering at Malla Reddy College of Engineering and Technology, Hyderabad. She has Eight years of experience in teaching. She received M.Tech., degree in Communication and Signal Processing at JNTUA, Anantapur, Andhra Pradesh India. Her research interests includes image processing, artificial intelligence, machine learning, wireless communications and internet of things. She can be contacted at email: bhavanakuruva@gmail.com.