Low Power and High Speed Dadda Multiplier using Carry Select Adder with Binary to Excess-1 Converter

Muteen Munawar
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan
munawar.muteen@gmail.com

Talha Khan
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan
ktalha504@gmail.com

Muhammad Rehman
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan
mrehman2015ee39@gmail.com

Zain Shabbir, Kamran Daniel
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan{zain.shabbir,
kamran.daniel}@uet.edu.pk

Ahmed Sheraz
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan
ahmadsheraz33.as@gmail.com

Muhammad Omer
Department of Electrical Engineering
University Of Engineering And
Technology, Lahore
Faisalabad, Pakistan
muhammadomer112@gmail.com

***Abstract*—As the digital electronic systems are getting better with the advancement in technology day by day; there is a need to build faster and more power-efficient multipliers, which are the major building block in most of the digital processing systems. Dadda tree multiplier is one of the effective multipliers consuming low power and quite faster than other multipliers i.e. Vedic, Wallace and booth redix4 multiplier. In this paper, we have designed a Dadda tree multiplier with carry select adder followed by binary to excess-1 converter which makes it much faster and power efficient. The results for this multiplier are comparatively better considering other existing multipliers.**

Keywords—Low Power; High Speed; Dadda Tree; carry select Adder; Binary to excess-1 converter

I. Introduction

In this era of modern technology, the need of the hour is to keep upgrading the digital electronics systems to encounter the speed and performance challenges of the new world problems. Depending on the excessive usability of Multipliers in digital processing systems, their performance must be progressively optimized with time. Multiplication process is one of the main factors, which allows the system to achieve high data rates depending upon its speed of functionality [1]. Also, the execution time of most of the algorithms in digital systems depends upon the multiplication process that's why multiplier with enhanced speed is much needed [2].

Moreover, high-performance processors are required for evaluating a large amount of data in different applications of Very Large Scale Integration (VSLI) and digital processing systems [3]. For this purpose multipliers play a vital role to achieve better performance results. Therefore the operating speed of multiplier is of great importance and must be considered while circuit designing in digital processing systems especially for general-purpose processors [4].

In recent years, many different attempts have been made to develop multipliers that provide high speed and consume less power. Basically, multipliers are of two types, one is serial and the other is parallel. Parallel multipliers are normally faster than serial multipliers [5]. The functionality of multipliers can be divided into three steps. Generation of partial product, addition of partial product and final addition [1]. To obtain the minimum delay, minimum power dissipation, and maximum output through a digital circuit, there are different algorithms and techniques already been proposed. Some of the existing techniques are merge delay transform [7-8], genetic and evolutionary algorithm [9-10-11], delay path Un-equalization [12] and carry-look-ahead logic [13]. All these techniques use CMOS technology for their implementation and execution process. To achieve a compact area and minimum time delay, the multiplier with pass transistor logic technique is used which also has very little capacitance. A power dissipated multiplier can be designed through a reversible logic technique. A multiplier with adiabatic static CMOS logic will dissipate less amount of heat during the propagation of charges. Dadda algorithm with optimized FA is used to deal with the propagation of carries and to decrease delay time [14]. A 4x4 multiplier, using CSLA, HA's, FA's and Dadda algorithm can be designed to obtain a better output in the least response time. In most of the digital processing systems, CSLA is considered as one of the fastest adders to solve arithmetic functions [15]. In digital adders, the main thing that limits the speed of addition is the propagation of carry to the next step. The carry select adder (CSLA) can be used in several processing systems to optimize the delay problem of carry propagation to produce a sum by producing some carries from which a single carry is selected [6]. But the main problem here is that the CSLA utilizes many pairs of ripple carry adders (RCA) to produce the partial sum and respective carries by seeing the carry as input. In the end, a multiplexer (MUX) selects the final sum and carry and this whole process depicts that the CSLA approach is not an area-efficient technique [6]. So, as a solution to this problem, a basic idea of using binary to excess-1 converter

(BEC) is proposed to advance the addition performing speed. In this work, a Dadda tree multiplier is presented where the RCA unit in normal CSLA is removed and RCA's operation is replaced by binary to excess-1 converter. Using CSLA with BEC rather than RCA and putting Cin = 1 gives faster and low power consuming multiplier. All the simulations of proposed work have been performed properly and are implemented using DSCH2 and Microwind2 software.

II. Dadda Tree Reduction Algorithm

In Dadda tree reduction, our objective is to decrease the height of partial products tree (Fig. 1) minimum to 2 in a stage succession. The restriction for Dadda tree is that in each of its stages, the reduction shouldn't be greater than 1.5 times. To reduce tree height to 2 at the last Dadda stage, we will anticipate the heights of the antecedent Dadda stages. As before last Dadda stage, height is 3, (2 x 1.5) and at the third last Dadda stage, height is 4, (3 x 1.5 = 4.5 rounded to 4). Thus for 4 x 4 multiplication Dadda Tree Reduction Algorithm has three stages depending on the height of tree. At the 1st stage, we have to reduce the height from 4 to 3. Here half adders (HA) or full adders (FA) can be used to reach the desired height. It can be seen in Fig 2, half adders are used to reduce the height. Here we use two HA's, which operate in parallel. So that when the carry from 4th column (from left) promotes to 3rd column, height will not become 4, at this point, because the second HA has already done its work and reduced 3rd column in 2 (height), the carry from the first HA increases its height to 3 only. Carry out from the second HA will forward to the 2nd column. Carry from the 3rd column will make of height 2nd column 3, thus we do not need any adders here in the first stage. Here both the HAs are working in parallel, which makes computation faster. Fig 3 describes the last two stages of reduction. Small rectangle shows the Half Adder while the other three (Black ones) show Full Adder (2nd stage). In the 3rd and last stage, RCA is used.

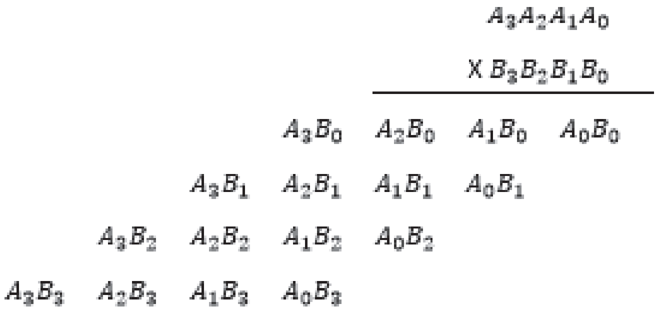

$A_3A_2A_1A_0$
X $B_3B_2B_1B_0$
A_3B_0 A_2B_0 A_1B_0 A_0B_0
A_3B_1 A_2B_1 A_1B_1 A_0B_1
A_3B_2 A_2B_2 A_1B_2 A_0B_2
A_3B_3 A_2B_3 A_1B_3 A_0B_3

Fig. 1. 4x4 Multiplication

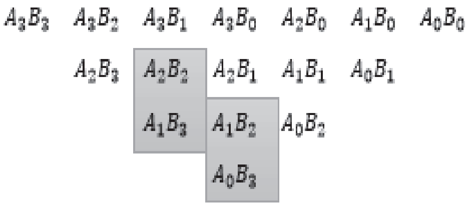

A_3B_3 A_3B_2 A_3B_1 A_3B_0 A_2B_0 A_1B_0 A_0B_0
A_2B_3 A_2B_2 A_2B_1 A_1B_1 A_0B_1
A_1B_3 A_1B_2 A_0B_2
A_0B_3

Fig. 2. Dadda Tree Height

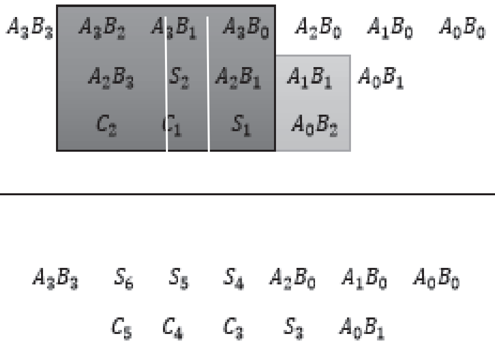

A_3B_3 A_3B_2 A_3B_1 A_3B_0 A_2B_0 A_1B_0 A_0B_0
A_2B_3 S_2 A_2B_1 A_1B_1 A_0B_1
C_2 C_1 S_1 A_0B_2
A_3B_3 S_6 S_5 S_4 A_2B_0 A_1B_0 A_0B_0
C_5 C_4 C_3 S_3 A_0B_1

Fig. 3. Final reduction stage

III. Proposed Multiplier

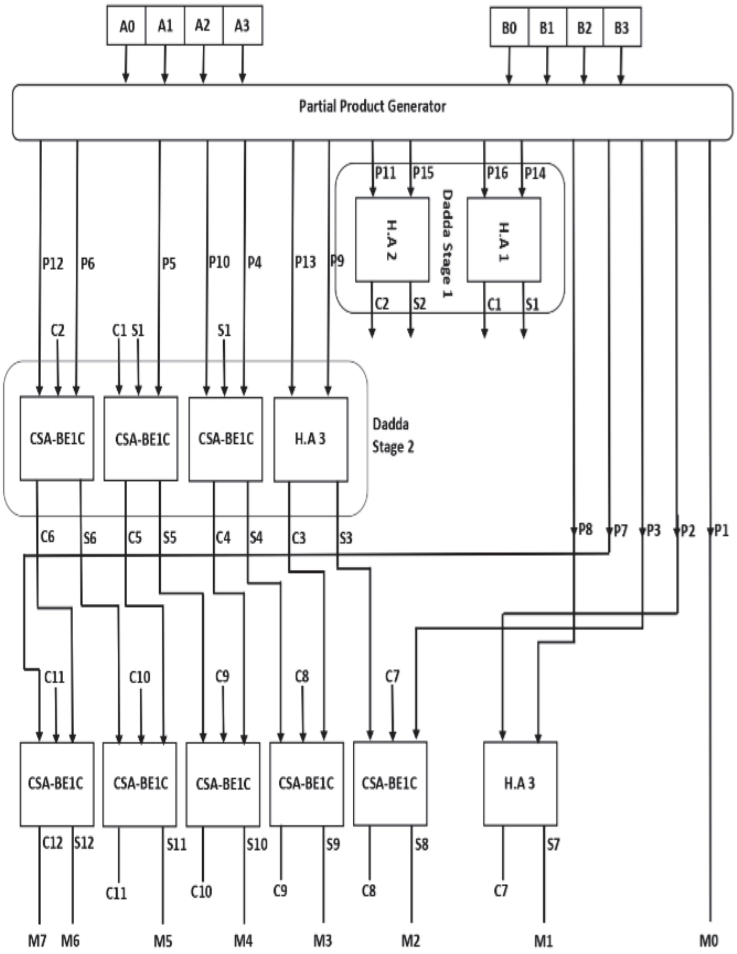


Fig. 4. Proposed Multiplier Block Diagram

In the proposed multiplier, the Dadda tree reduction Algorithm is modified to use the Carry Select adder with binary to excess 1 converter, instead of a normal Full Adder. Block diagram is shown in fig. 4. The working of the proposed multiplier is exactly the same as described in the previous section. However, the main block to discuss here is the CSA-BEX-1 block which is the core modification of the proposed Dadda algorithm multiplier.

The block diagram of CSA-BEX-1 is shown in Fig. 5 and quite easy to understand. As we know that a simple CSA (which is a better option than a simple FA) calculates two sums assuming carry Cin 0 and 1. In CSA-BEX1 case, when carry is 0 CSA-BEX1 works same as a simple CSA but when the carry is 1, a binary to excess-1 converter will be used, which is not an actual adder however it will add 1 carry in our sum which will be even faster and low power implementation of an adder. In the end, Mux will be used which selects the final results on the basis of actual carry in Cin.

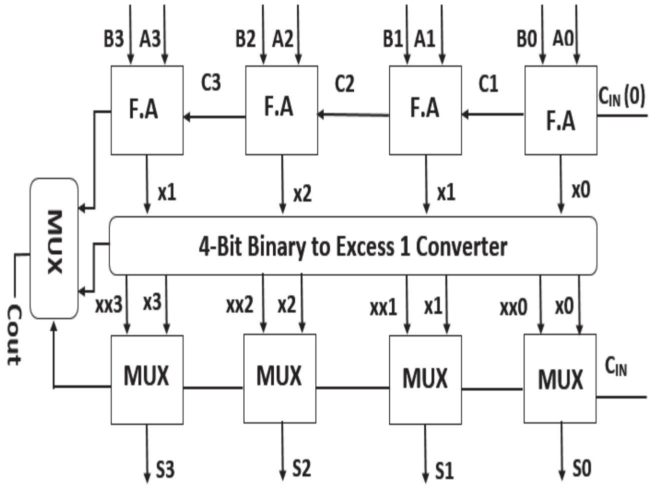
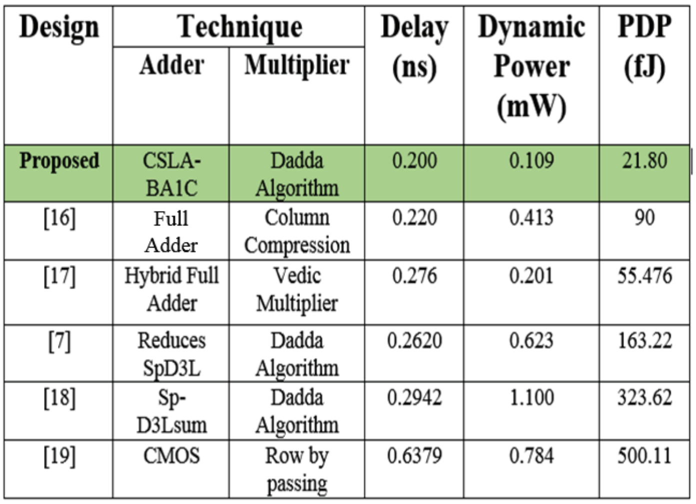


Fig. 5. 4-Bit Carry Select Adder with BEC-1 (CSA-BEX-1)

IV. RESULTS

The proposed and some other existing multipliers are comparatively evaluated on the basis of their dynamic power, delay, and PDP. The evaluation verifies that the proposed multiplier shows better performance results as compared to the results of other multipliers. The comparative results of power, delay, and PDP of the proposed multiplier with other multipliers are mentioned in Table. 1.

Fig.6, Fig.6, and Fig.8 show a graphical comparison of results of delay, dynamic power and PDP of different multipliers. It clearly shows that the performance analysis of the proposed multiplier is much improved than other multipliers. Fig.9 shows the layout design of proposed multiplier. All the transistor level circuits and layout designs are implemented and tested using DSCH2 and Microwind2 respectively.



Design	Technique		Delay (ns)	Dynamic Power (mW)	PDP (fJ)
	Adder	Multiplier			
Proposed	CSLA-BA1C	Dadda Algorithm	0.200	0.109	21.80
[16]	Full Adder	Column Compression	0.220	0.413	90
[17]	Hybrid Full Adder	Vedic Multiplier	0.276	0.201	55.476
[7]	Reduces SpD3L	Dadda Algorithm	0.2620	0.623	163.22
[18]	Sp-D3Lsum	Dadda Algorithm	0.2942	1.100	323.62
[19]	CMOS	Row by passing	0.6379	0.784	500.11

Table. 1. Comparison of Multipliers

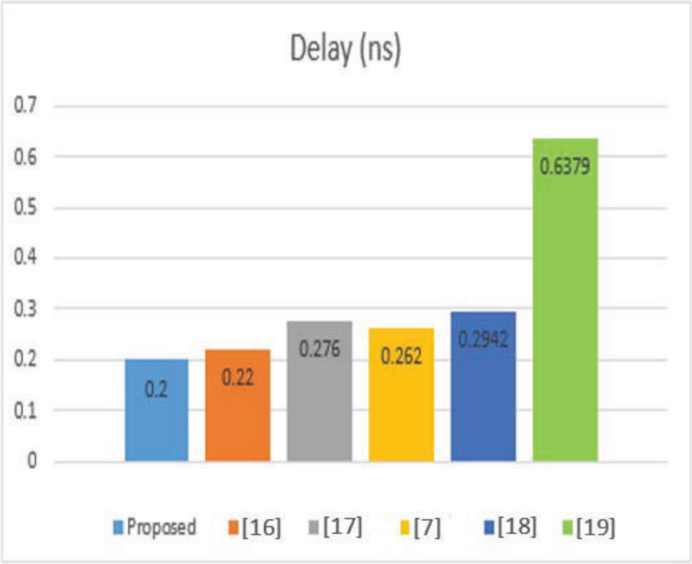


Fig. 6. Graphical comparison of Delay

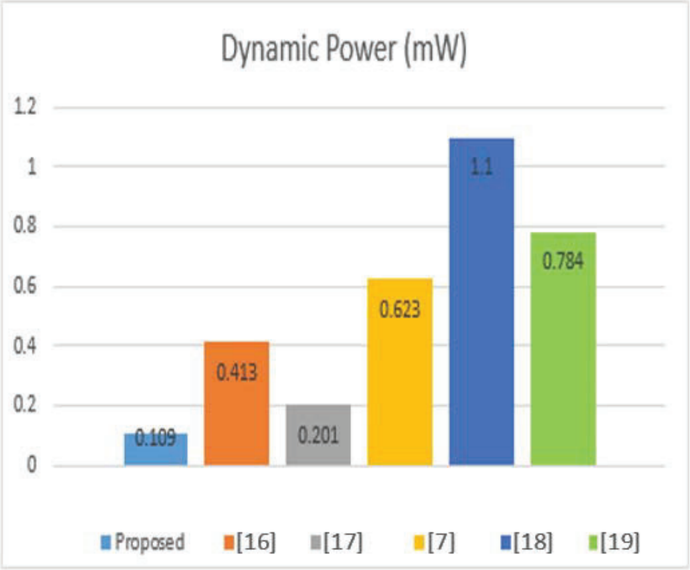


Fig. 7. Graphical comparison of Power

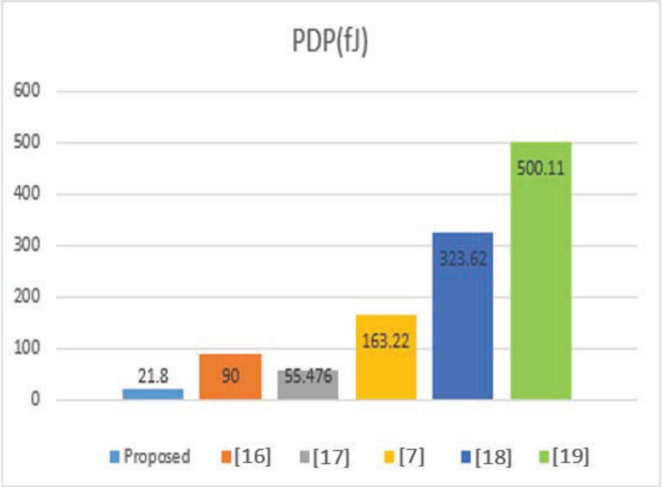


Fig. 8. Graphical comparison of PDP

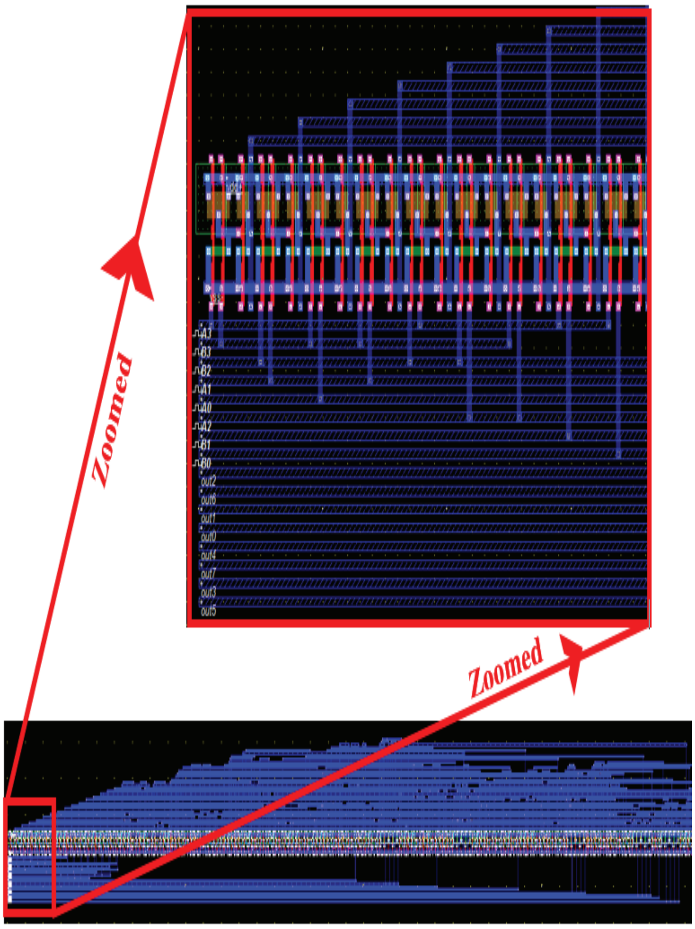


Fig. 9. Layout design of proposed multiplier

V. CONCLUSION

A modified Dadda tree Multiplier is proposed and designed on transistor-level using CMOS logic. All the results clearly determine that the proposed technique by using CSA-BEX-1 in the Dadda tree multiplier gives really good results as compared to different existing multipliers as mentioned. The main advantage of the proposed multiplier is that it consumes much lesser power than other cited multipliers. Moreover, it shows comparatively better results of PDP and quite faster than other multipliers as well.

VI. REFRENCES

[1] Ravi, Nirlakalla, et al. "A New Design for Array multiplier with Trade Off in power and Area." arXiv preprint arXiv:1111.7258 (2011).

[2] Vaidya, Sumit, and Deepak Dandekar. "Delay-power performance comparison of multipliers in vlsi circuit design." International Journal of Computer Networks & Communications (IJCNC) 2.4 (2010): 47-56.

[3] Kesava, R. Bala Sai, et al. "Low power and area efficient Wallace tree multiplier using carry select adder with binary to excess-1 converter." 2016 Conference on Advances in Signal Processing (CASP). IEEE, 2016.

[4] Bano, Nishat. "VLSI design of low power booth multiplier." International Journal of Scientific & Engineering Research 3.2 (2012): 2-4.

[5] Kaushik, Vikas, et al. "Comparative Analysis of Proposed Parallel Digital Multiplier with Dadda and Other Popular Multipliers." International Journal 4 (2017): 237-240.

[6] Sharma, Neeta, and Ravi Sindal. "Modified booth multiplier using wallace structure and efficient carry select adder." International Journal of Computer Applications 68.13 (2013).

[7] Shabbir, Z., Ghumman, A. R., & Chaudhry, S. M. (2016). A Reduced-sp-D3Lsum Adder-Based High Frequency 4× 4 Bit Multiplier Using Dadda Algorithm. Circuits, Systems, and Signal Processing, 35(9), 3113-3134.

[8] (13)Javali, R. A., Nayak, R. J., Mhetar, A. M., & Lakkannavar, M. C. (2014, November). Design of high speed carry save adder using carry lookahead adder. In International Conference on Circuits, Communication, Control and Computing (pp. 33-36). IEEE.

[9] Naqvi, S. Z., Hassan, S. Z., & Kamal, T. (2016, January). A power consumption and area improved design of IIR decimation filters via MDT. In 2016 International Conference on Intelligent Systems Engineering (ICISE) (pp. 146-151). IEEE.

[10] Kumar, P. Prem, K. Duraiswamy, and A. Jose Anand. "An optimized device sizing of analog circuits using genetic algorithm." *European Journal of Scientific Research* 69.3 (2012): 441-448.

[11] Vural, Revna Acar, et al. "CMOS differential amplifier area optimization with evolutionary algorithms." *Proceedings of the World Congress on Engineering and Computer Science*. Vol. 2. 2013.

[12] López, S., Garnica, Ó., Hidalgo, I., Lanchares, J., & Hermida, R. (2003, September). Power-consumption reduction in asynchronous circuits using delay path unequalization. In International Workshop on Power and Timing Modeling, Optimization and Simulation (pp. 151-160). Springer, Berlin, Heidelberg.

[13] Raminder Preet Pal Singh, Parveen Kumar, Balwinder Singh, "Performance Analysis of 32 bit Array Multiplier with Carrying Save Adder and with Carry Look ahead adder", in International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009.

[14] Riaz, M. H., Ahmed, S. A., Javaid, Q., & Kamal, T. (2018, January). Low power 4× 4 bit multiplier design using dadda algorithm and optimized full adder. In 2018 15th International Bhurban Conference on Applied Sciences and Technology (IBCAST) (pp. 392-396). IEEE.

[15] Amelifard, B., Fallah, F., & Pedram, M. (2005, March). Closing the gap between carry select adder and ripple carry adder: a new class of low-power high-performance adders. In Sixth international symposium on quality electronic design (isqed'05) (pp. 148-152). IEEE.

[16] Saha, Aloke, et al. "Novel CMOS multi-bit counter for speed-power optimization in multiplier design." AEU-International Journal of Electronics and Communications 95 (2018): 189-198.

[17] Lee, S. J., and S. H. Ruslan. "4x4 bit Vedic multiplier using 13T hybrid full adder in 90 nm CMOS technology." Journal of Fundamental and Applied Sciences 10.6S (2018): 438-450.

[18] Purohit, Sohan, and Martin Margala. "Investigating the impact of logic and circuit implementation on full adder performance." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 20.7 (2011): 1327-1331

[19] Kuo, Ko-Chi, and Chi-Wen Chou. "Low power and high speed multiplier design with row bypassing and parallel architecture." Microelectronics Journal 41.10 (2010): 639-650.