IETE Journal of Research

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tijr20

High-Speed Hybrid Multiplier Design Using a Hybrid Adder with FPGA Implementation

V. Thamizharasan & N. Kasthuri

To cite this article: V. Thamizharasan & N. Kasthuri (2023) High-Speed Hybrid Multiplier Design Using a Hybrid Adder with FPGA Implementation, IETE Journal of Research, 69:5, 2301-2309, DOI: 10.1080/03772063.2021.1912655

To link to this article: https://doi.org/10.1080/03772063.2021.1912655

Published online: 21 Apr 2021.

IETE JOURNAL OF RESEARCH
2023, VOL. 69, NO. 5, 2301–2309
https://doi.org/10.1080/03772063.2021.1912655

High-Speed Hybrid Multiplier Design Using a Hybrid Adder with FPGA Implementation

V. Thamizharasan [1] and N. Kasthuri[2]

[1]Department of Electronics and Communication Engineering, Erode Sengunthar Engineering College, Perundurai, Erode 638 057, India;
[2]Department of Electronics and Communication Engineering, Kongu Engineering College, Perundurai, Erode 638 052, India

ABSTRACT

The major role of electronic devices is providing low power dissipation and high speed with compact area. The speed of electronic devices depends on arithmetic operations. Multiplication is the important arithmetic operation in many VLSI signal processing applications. Hence, a high-speed multiplier is needed to design any signal processing module. Many multipliers are surveyed in the literature. They are Array, Wallace tree, booth, Vedic and Compressor-based multiplier. The speed of these multipliers depends on partial product accumulation. The hybrid parallel adder-based multiplier is proposed to improve the speed of multiplication compared to the existing technique. In this technique the partial products of, two consecutive bits (multiplicands), are added simultaneously with the help of a hybrid adder (Hancarlson, Weinberger and Ling adder). The proposed architecture is synthesized and simulated using Xilinx ISE 12.1 with various FPGA boards. Synthesized report shows that the speed of proposed multiplier (Spartan 6 FPGA implementation) is improved when compared to an Array multiplier (22.14%), Wallace tree multiplier (20.41%), Multiplier using compressor (13.89%), Vedic Multiplier using CLA (13.03%), Vedic Multiplier using RCA (3.54%), Modified Booth multiplier (4.42%) and Vedic Multiplier using HCA with BEC (3.28%).

KEYWORDS

Area; Delay; FPGA; Hybrid; Multiplier; Power; Vedic; VLSI; Xilinx ISE 12.1

1. INTRODUCTION

Speed, power and area are the primary targets in real-time application system. The most important module in DSP processor, CPU, digital filters, modulator, demodulator, cryptography systems and many signal processing blocks are multiplication and addition [1]. The high-speed and less area are the main parameters in portable devices such as Mobiles, PCs, Calculator, TV, and Watch. The speed of these portable devices depends on multiplier and adder. Hence, there is an increasing demand to improving the multiplier performance. Also a high-speed multiplier is a major pre-requisite of the fast arithmetic operation.

Multipliers are used for a variety of applications depending on the restraints of each application holds with it. Hence, it is necessary to investigate the area, power and delay parameter of the multiplier used in signal processing operations [2]. For multiplying two binary numbers, there are several types of multipliers considered: add-shift multiplier, array multiplier, carry-save addition, pipeline multiplier, ripple-carry parallel adder, serial-parallel multiplier, 2's complement multiplication, Vedic multiplier, booth multiplier, *etc*.

The rest of this paper is categorized as follows. A review of adders and multipliers is presented in sections 2 and 3. The hybrid adder-based multiplier architecture is disclosed in section 4. Section 5 presents simulated results and comparisons of our proposed work with the existing works. Conclusion of this paper is presented in section 6.

2. ADDERS

Adders are playing an important role in augmenting the execution of the digital system. In most of the signal processing architectures, adders are not only used in the ALUs operations, but also used in the other system modules. They are used to determine addresses, indices of table, and counting the number of instruction execution in the microprocessor. Several types of adders were considered for the addition of two numbers in binary: Ripple Carry Adder (RCA) [1], Carry Look ahead Adder (CLA) [3], Carry Save Adder (CSA), Carry Select Adder (CSELA) [3,4], Carry Increment Adder (CIA), Brent Kung Adder (BKA), Han Carlson Adder (HCA), Ling Adder, Weinberger Adder-based CSELA (WCSELA), *etc*.

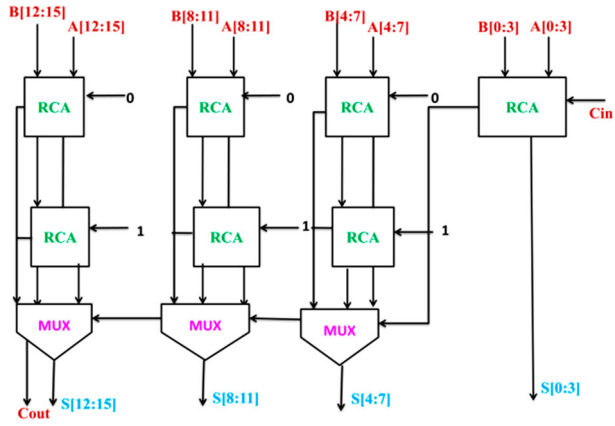

Figure 1: Architecture of CSEL Adder (CSELA)

2.1 Carry Select Adder (CSELA)

In CSELA, feasible values of the input carry (*i.e.* zero and one) of each stage are established in advance and the sum is evaluated in well advance. Architecture of CSELA is displayed in Figure 1. Generally, the CSELA is constructed using dual RCAs and multiplexer; it used to select the final carry and sum output. The main advantage of CSELA is to reduce the carry propagation time. Hence the speed of operation is improved [5]. But more full adders are needed when compared to RCA.

2.2 Hancarlson Adder (HCA)

Hancarlson adder is designed using the principle of parallel prefix addition. It is the combination of Kogge-Stone and BrentKung adder. KoggeStone adder provides less delay and BrentKung adder provides less area [6,7]. Hence, it has high speed, less power consumption and low hardware components than other adders.

Hancarlson adder consists of pre- and post-prefix stages. In the pre-processing stage, the following expressions are used to determine the generate (Gi) and propagate (Pi) signals.

$$\text{Generate} = \text{Ai\&Bi} \tag{1}$$

$$\text{Propagate} = \text{Ai} \oplus \text{Bi} \tag{2}$$

Furthermore, the k to i^{th} bits are extended to the blocks using the following generate and propagate expressions.

$$\text{Gi} = \text{G}_{i-1} + (\text{G}_{i-2}\&\text{P}_{i-1}) \tag{3}$$

$$\text{Pi} = \text{P}_{i-1}\&\text{P}_{i-2} \tag{4}$$

The approximate output sum bit is only needed in the post-processing stage. The above subset expressions are used to determine the final output carry bits. The final

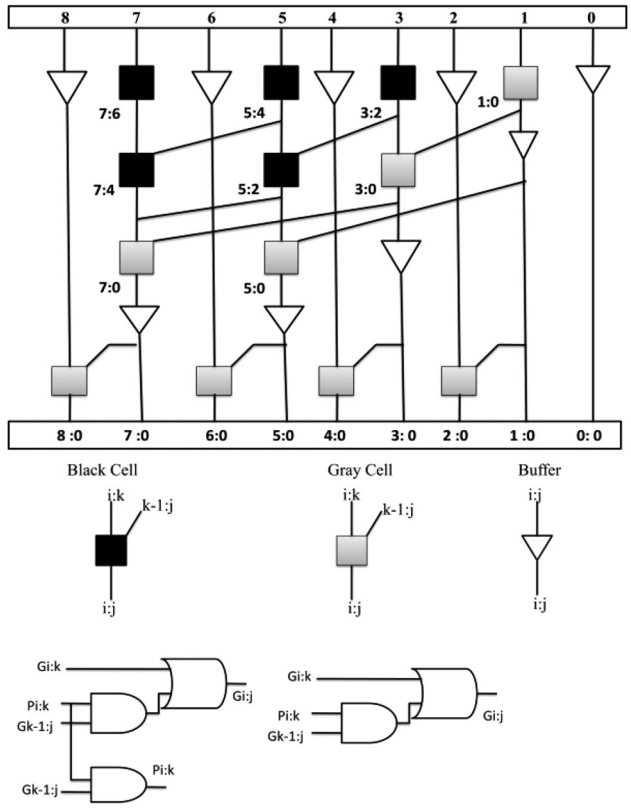

Figure 2: Han-Carlson adder

sum is obtained as follows.

$$\text{S}_i = \text{P}_i \hat{} \text{G}_{i-1:0} \tag{5}$$

The architecture of Hancarlson adder is displayed in Figure 2.

2.3 Ling Adder

In CSELA, RCA stage is changed to Ling adder, as Ling adder provides less delay and minimum chip size when compared to CLA. It is called Ling CSELA [6]. The following expressions are used to calculate the generate and propagate bitwise signals

$$\text{Generate(Gi)} = \text{Ai\&Bi} \tag{6}$$

$$\text{Propagate(Pi)} = \text{Ai} + \text{Bi} \tag{7}$$

The architecture of Modified LLCSELA is displayed in Figure 3.

2.4 Weinberger-Based CSELA (WCSELA)

The concept of Weinberger recurrence algorithm is used to compute the carry for improving the delay of adder.

The BK adder is replaced by Weinberger adder to create WCSELA [6]. The architecture of WCSELA is displayed

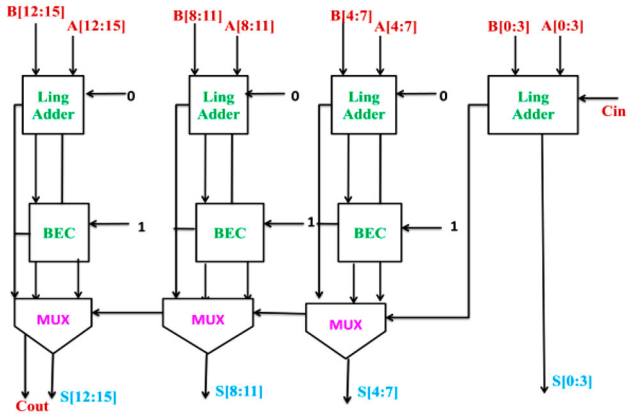

Figure 3: Structure for Modified Linear Ling CSELA

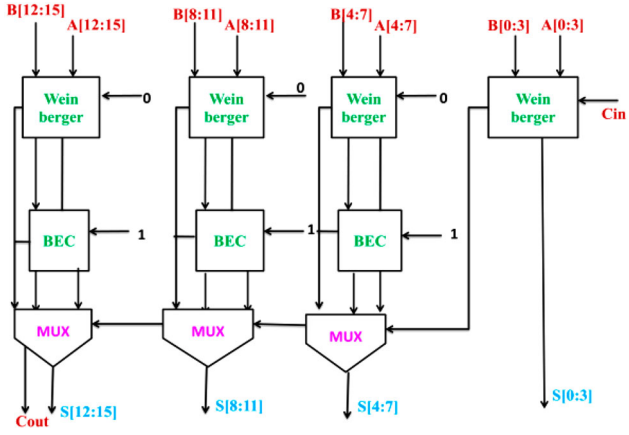

Figure 4: Architecture of Weinberger CSELA (WCSELA)

in Figure 4. This adder utilizes lower area (only minimum number of Gates required in BEC) when compared to RCA. Also the consumption of power and delay are low when compared to the above-mentioned designs.

3. MULTIPLICATION

Multiplication is an essential arithmetic operation in Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT), FIR filter [8], IIR filter, adaptive filter, digital modulation, demodulation, MAC operation, finite word length systems and multimedia applications [2,9]. The growing digital world is looking high-quality, high accuracy digital information. For example, the image and video processing systems require a high-quality image with high speed. To produce the high-quality and accuracy of output data in the signal processing module, it necessary to optimize the speed, area and power consumption of the arithmetic module [10]. Hence the module of multiplication is considered to be improving the speed of the system in terms of computation delay [2].

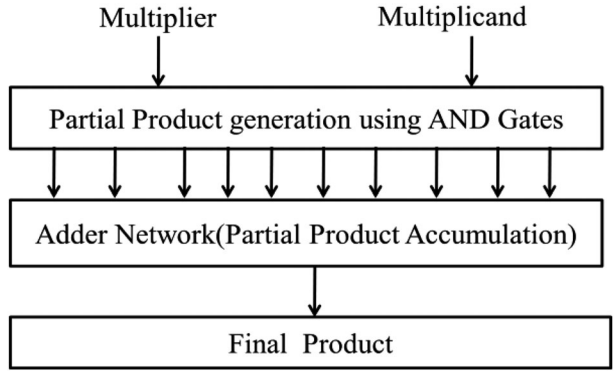

Figure 5: Architecture of general multiplier

The architecture of a general multiplier is displayed in Figure 5.

The multiplications are carried out by the following three steps:

(1) Recoding and partial product generation
(2) Partial product reduction and
(3) Accumulation of partial product.

The multipliers are classified, based on the partial product reduction, as linear array multipliers and tree multipliers.

In the array multiplier, the multiplications of two binary numbers are carried out based on the principle of add and shift method. This principle uses a regular structure. When a large number of bits are used, this multiplier produces large delay and high power consumption due to carry propagation. Hence it is necessary to concentrate on the carry propagation speed (critical path of output) of the multiplier [11,12].

In the tree multiplier, the partial products are arranged in row wise or column wise. Hence the number of hardware elements is reduced when compared to the array multiplier. But the tree multiplier has an irregular layout and complicated interconnects. These structures are added more physical design effort and increasing the delay due to wiring capacitance. Accumulation of a partial product is limiting the speed of multiplication operation. Hence it is necessary to concentrate on the partial product accumulation speed (critical path of output) in multiplier. Many efficient binary multiplier circuits are reported in the article [13].

The array multiplier was designed using a modified full adder-based multiplexer. This multiplier consumes only

low power. For increasing the speed, the Wallace tree multiplier is introduced. This increases the speed of partial product accumulation, because a carry-save adder is used instead of a ripple carry adder.

The multiplier using approximate 4-2 compressor shows a significant reduction in power consumption, delay and area than other multipliers [10]. Also this compressor-based multiplier shows a significant error rate [9]. The 8∗8 and 16∗16 Dadda multiplier was implemented using this compressor [14].

The further significant improvement of speed was achieved by using a modified binary multiplier based on Vedic mathematics when compared to the previous multiplier architectures [15,16]. Also the pipeline-based Vedic multiplier showed minimum power consumption with minimum delay when compared to the existing non-pipelined designs.

Modified booth and Wallace tree multiplier is the fastest multiplier. This modified booth multiplier reduced the number of partial products by half or one third of the multiplier bits [12]. Also the multiplier should be selected depending on the performance requirements and the nature of the applications.

Based on the above surveyed result, it's necessary to concentrate on partial product reduction and accumulation for improving the speed and area utilization of multiplication.

4. PROPOSED MULTIPLIER

4.1 Hybrid Adder

The adder is created using more than one logic circuit. This kind of adder is known as the hybrid-adder. The structure of the hybrid-adder is displayed in Figure 6. In this structure, A and B are the input signals in Module-I. The Module-II and Module-III are intermediate blocks of adder. Different types of adder techniques are used in the intermediate module for producing sum and carry outputs.

The two types of design structures are followed in the hybrid adder design.

(1) Homogeneous: Combining the similar type of more than one adder is called Homogeneous design.
(2) Heterogeneous: Combining the different type of more than one adder is called Heterogeneous design.

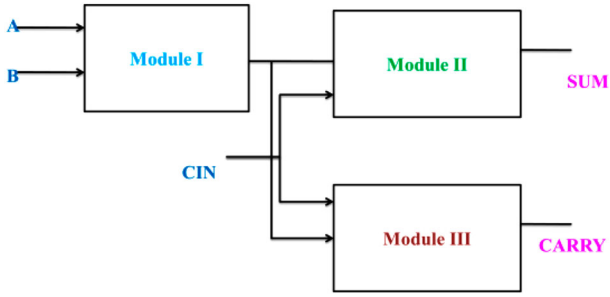

Figure 6: Structure of hybrid adder

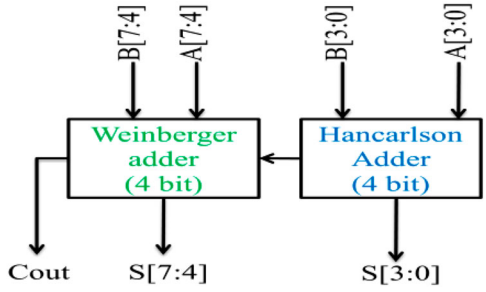

Figure 7: Structure of the proposed 8-bit hybrid adder

The proposed idea is to form a hybrid structure, using the above two techniques, to bring the high performance and low cost (chip size) products.

The major constraint of the above adders is the speed of operation, hence concentrating on the delay (critical path of output) of an adder. A new version of CSELA is proposed using a hybrid technology.

A variety of adders are discussed in Section 2. Among these, Hancarlson adder provides a less delay (speed 7.58 ns [17]) and utilization of area was improved (No. of Slices/LUT- 14 [17]). Similarly, the Ling adder provides less delay (0.11 ns [18]) and utilization of the area (chip size) was minimum when compared to CLA. Also Weinberger adder reduces logic stages (No. of Slices/LUT- 25 [19]) and final carry generation time is reduced when compared to other adders. Additionally, binary to excess one converter circuit can consume less power and area utilization (No. of Slices/LUT- 340 [20]). To improve the speed and area utilization of multiplier, the above-mentioned adders are incorporated to different modules/stages in the multiplier.

The 8-bit hybrid technology-based CSELA is displayed in Figure 7. This adder consists of two stages each with 4 bits. The Hancarlson adder and Weinberger adder are used in stage1 and stage 2, respectively.

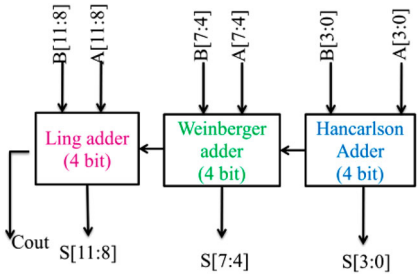

Figure 8: Structure of the proposed 12-bit hybrid adder

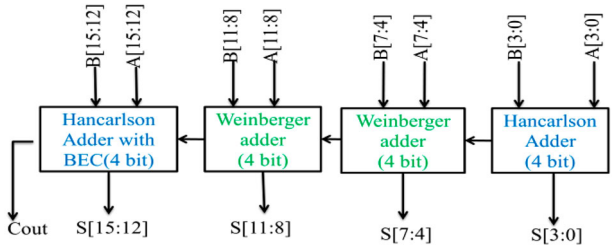

Figure 9: Structure of the proposed 16-bit hybrid adder

The 12-bit hybrid technology-based CSELA is displayed in Figure 8. This adder consists of three stages each with 4 bits. The first 4 bits are added using Hancarlson adder, the next 4 bits (bit 4–7) are added using Weinberger adder and the last 4 bits (bit 8–11) are added using ling adder.

Similarly, a 16-bit CSELA was designed using a hybrid technology as displayed in Figure 9. This adder consists of four stages each with 4 bits. The first and last stage (1 and 4) inputs (bits 0–3 and bits 12–15) are added using Hancarlson adder, the next two stage (stages 2 and 3) inputs are added using Weinberger adder.

4.2 Hybrid Multiplier

The architecture of the proposed multiplier is displayed in Figure 10. This architecture is a 8∗8 multiplier. In this structure c0 to c7 are the partial products of multiplicands. The partial products are generated using a series of AND gates.

Partial product generation is a first process of multiplication. Partial products are obtained by performing the logical AND operation with every bit of multiplier by every bit of multiplicand. For example, a 8∗8 Multiplier has multiplier A (A0 to A7) and multiplicand B (B0 to B7) each with 8 bits. In partial product generation, the first step involves performing logical AND operation of Multiplicand B0(LSB) with every bit of Multiplier A and the results are stored in C0(8 bit), mathematically represented as c0[0] = B0 AND A0,c0[1] = B0 AND A1 … , c0[7] = B0 AND A7. Similarly, Multiplicand B1 with

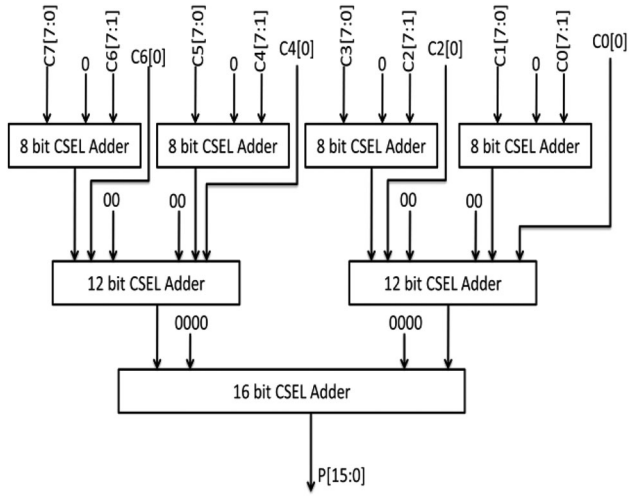

Figure 10: Architecture of the proposed multiplier

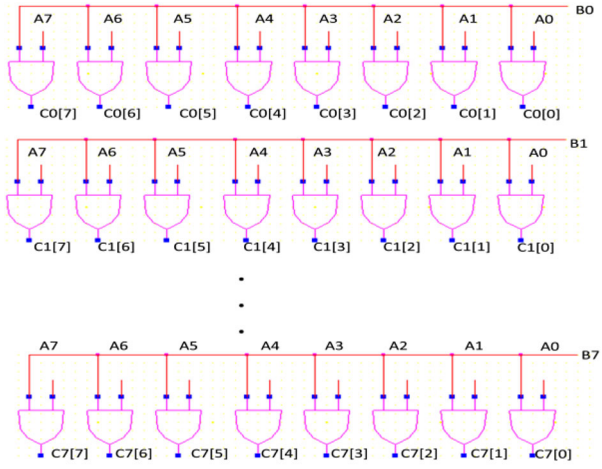

Figure 11: Architecture of partial product generation

every bit of Multiplier A and results are stored in C1(8 bit) and so on. The architecture of partial product generation is shown in Figure 11.

The proposed hybrid multiplier consists of 3 stages. In each stage, different-sized hybrid adders are used. Namely, an 8- bit CSELA (combination of Hancarlson and Weinberger adder each with 4 bit) is used in the first stage (4 numbers of 8-bit CSELA), 12-bit CSELA (combination of Hancarlson, Weinberger and ling adder each with 4 bits) is used in the second stage (2 numbers of 12-bit CSELA) and a 16-bit CSELA (combination of Hancarlson, Weinberger adder and Hancarlson with BEC each with 4 bit) is used in the third stage (1 number of 16-bit CSELA).

The first stage consists of four numbers of 8-bit CSELAs. In this stage, the partial products of each two consecutive

bits of multiplier are added simultaneously. The output of each adder is passed to the next stage adder input. The second stage consists of two numbers of 12-bit adders. In this stage, the first stage outputs are added simultaneously and the results are passed to the final stage 16-bit CSELA. This 16-bit adder output is a final product of the 8∗8 multiplier.

5. PERFORMANCE ANALYSIS

All the existing multipliers and adders are discussed in sections 2, 3 and the proposed designs are discussed in section 4. These adders and multipliers are implemented in XILINX ISE 12.1 with a family of various device and speed grade using Verilog HDL. Each adder and multiplier is simulated and verified separately. Figures 12–14 show that simulation result and device utilization summary of the proposed hybrid multiplier, respectively.

Tables 1 and 2 show utilization of delay, area, power of various adder and multiplier. Table 3 shows the proposed multiplier delay improvement (%) when compared to other multipliers.

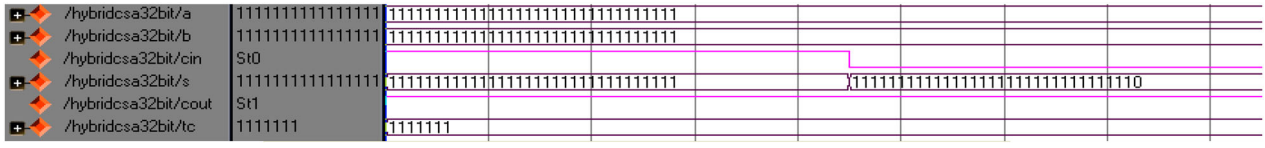

Figure 12: Simulation result of the proposed hybrid adder

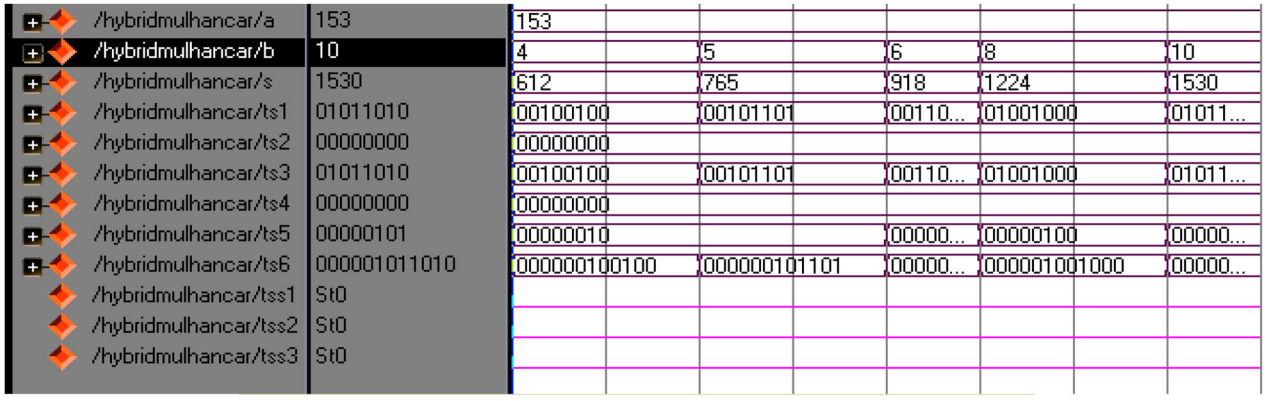

Figure 13: Simulation result of the proposed hybrid multiplier

Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	90	2400	3%
Number of fully used LUT-FF pairs	0	90	0%
Number of bonded IOBs	33	102	32%

Figure 14: Device utilization summary of the proposed hybrid multiplier

Table 1: Utilization of delay, area and power for various 16-bit adders in Spartan 6 FPGA

S.No.	Adder techniques	No. of LUTs	Delay (ns)	ADP
1.	RCA	26	5.09	132.34
2.	CBL Based CSELA	25	4.91	122.75
3.	CLA	24	4.60	110.4
4.	Weinberger BEC- based CSELA	30	3.32	99.6
5.	D Latched CSELA	42	4.21	176.82
6.	Brent Kung CSELA	36	3.44	123.84
7.	Conventional CSELA	24	4.94	118.56
8.	Conventional CSELA	24	4.94	118.56
9.	Ling CSELA	27	7.19	194.13
10.	Han Carlson BEC CSELA	33	3.06	100.98

ADP – area delay product.

Table 2: Utilization of delay, area and power for various multipliers in Spartan 6 FPGA

S.No	Multiplier techniques	Delay (ns)	Noof LUT	Power (uW)	ADP	PDP
1.	Array Multiplier	21.42	84	52.53	1799.28	1125.193
2.	Wallace tree multiplier	20.96	116	59.15	2431.36	1239.784
3.	Multiplier using compressor	19.37	120	53.52	2324.4	1036.682
4.	Vedic Multiplier using CLA	19.18	113	54.87	2167.34	1052.407
5.	Vedic Multiplier using RCA	17.29	108	53.86	1867.32	931.2394
6.	Modified Booth Multiplier	17.45	91	53.72	1587.95	937.414
7.	Vedic Multiplier using HCA with BEC	17.24	126	55.92	2172.24	964.0608
8.	Proposed Hybrid Multiplier	16.68	90	53.54	1501.2	893.0472

ADP – area delay product; PDP – power delay product.

Table 3: Delay improvement (%) when compared to other multipliers

S.No	Proposed Multiplier when compared to	Spartan 6
1.	Array Multiplier	22.14
2.	Wallace tree multiplier	20.41
3.	Multiplier using compressor	13.89
4.	Vedic Multiplier using CLA	13.03
5.	Vedic Multiplier using RCA	3.54
6.	Modified Booth Multiplier	4.42
7.	Vedic Multiplier using HCA with BEC	3.28

The above comparison shows that the delay of the proposed CSELA is improved by 22.14%, 20.41%, 13.89%, 13.03%, 3.54%, 4.42%, 3.28% when compared to Array Multiplier, Wallace tree multiplier, Multiplier using compressor, Vedic Multiplier using CLA, Vedic Multiplier using RCA, Modified Booth Multiplier, Vedic Multiplier using HCA with BEC, respectively.

Also it shows that the ADP of the proposed multiplier is improved by 16.56%, 38.25%, 35.41%, 30.73%, 19.60%, 5.46%, and 30.89% when compared to Array Multiplier, Wallace tree multiplier, Multiplier using compressor, Vedic Multiplier using CLA, Vedic Multiplier using RCA, Modified Booth Multiplier, Vedic Multiplier using HCA with BEC, respectively.

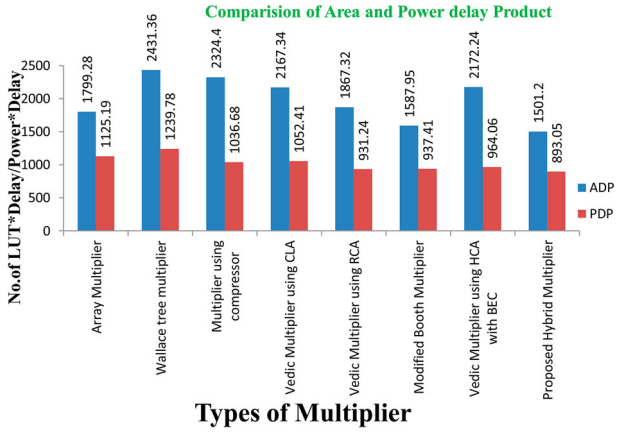

Figure 15: Area and power delay product comparisons of various multipliers

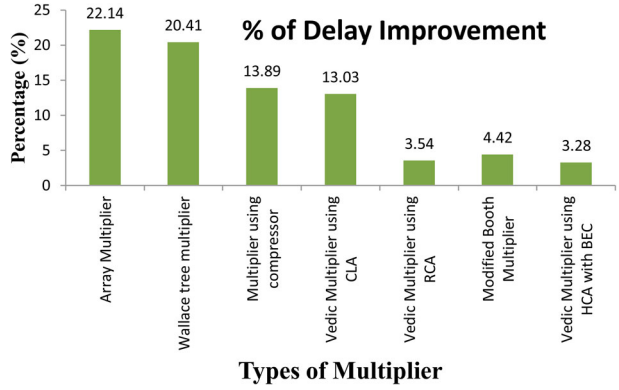

Figure 16: Proposed multiplier delay improvement (%) when compared to other multipliers

Similarly PDP of the proposed multiplier is improved by 20.67%, 27.96%, 13.85%, 15.14%, 4.10%, 4.73% and 7.36% when compared to Array Multiplier, Wallace tree multiplier, Multiplier using compressor, Vedic Multiplier using CLA, Vedic Multiplier using RCA and Modified Booth Multiplier, respectively.

The comparison chart for ADP and PDP of various Multipliers is shown in Figures 15 and 16, respectively. The proposed multiplier shows significant improvement in ADP and PDP.

Table 4 shows the comparisons of performance parameters of the FPGA for an existing and the proposed

Table 4: Comparison of synthesis (pre-layout) results of a 8∗8 multiplier

8∗8 Multiplier	Delay(ns)
Proposed design	16.68
Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor – Design I [9]	18.615
Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor Design – II [9]	18.39
Modified Binary Multiplier Circuit Based on Vedic Mathematics [21]	18.46
Design and Analysis of High Performance Multiplier Circuit [2]	20.7
Design and FPGA implementation of High Speed Vedic Multiplier [22]	23.644

multiplier and the results are analysed for the 8∗8 multiplier. From this table, it can be concluded that the critical path delay is decreased (speed is improved) by using the proposed multiplier (Hybrid) deign than the conventional and existing methods.

6. CONCLUSION

In this paper, a hybrid adder-based Multiplier (CSELA) is proposed and designed using Hancarlson adder, ling adder, Weinberger adder and BEC circuit. To reduce the delay and area of the multiplier, the final product of multiplier is calculated by each two consecutive multiplicand bits of partial products added simultaneously using different-sized hybrid adders. The simulation is carried out in Xilinx ISE 12.1 using Verilog HDL. The synthesized report shows that speed of the proposed multiplier in Spartan 6 FPGA implementation is improved by 22.14%, 20.41%, 13.89%, 13.03%, 3.54%, 4.42% and 3.28% when compared to Array multiplier, Wallace tree multiplier, Multiplier using compressor, Vedic Multiplier using CLA, Vedic Multiplier using RCA, Modified Booth Multiplier, and Vedic Multiplier using HCA with BEC, respectively. Furthermore, this work can be extended to design different input-sized (input bits) multiplier, filters, signal/Image processing module, multimedia application and cryptography applications.

ORCID

V. Thamizharasan http://orcid.org/0000-0001-7896-7164

REFERENCES

1. H. You, J. Yuan, W. Tang, and S. Qiao, "An energy and area efficient carry select adder with dual carry adder cell," *Electronics (Basel)*, Vol. 8, no. 10, pp. 1129–39, Oct. 2019.

2. I. Hussain, C. K. Pandey, and S. Chaudhury, "Design and analysis of high performance multiplier circuit," in *2019 Devices for Integrated Circuit (DevIC)*, Kalyani, India, 2019, pp. 245–7.

3. P. Balasubramanian, and N. Mastorakis, "Performance comparison of carry-lookahead and carry-select adders based on accurate and approximate additions," *Electronics (Basel)*, Vol. 7, no. 12, pp. 369–81, Dec. 2018.

4. B. S. Kandula, P. V. Kalluru, and S. P. Inty, "Design of area efficient VLSI architecture for carry select adder using logic optimization technique," *Comput. Intell.*, Vol. 36, 1–11, May 2020.

5. A. Liacha, A. K. Oudjida, F. Ferguene, M. Bakiri, and M. L. Berrandjia, "Design of high-speed, low-power, and area-efficient FIR filters," *IET Circuits Devices Syst.*, Vol. 12, no. 1, pp. 1–11, Jan. 2018. DOI:10.1049/iet-cds.2017.0058

6. N. Gaur, A. Mehra, P. Kumar, and S. Kallakuri, "16 Bit power efficient carry select adder," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, Mar. 2019, pp. 558–61.

7. N. U. Kumar, K. B. Sindhuri, K. D. Teja, and D. S. Satish, "Implementation and comparison of VLSI architectures of 16 bit carry select adder using Brent Kung adder," in *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, 2017, pp. 1–7.

8. V. Thamizharasan, and V. Parthipan, "An efficient VLSI architecture for FIR filter using computation sharing multiplier," *Int. J. Comput. Appl.*, Vol. 54, no. 14, pp. 1–6, Sep. 2012. DOI:10.5120/8631-1939

9. K. Sivanandam, and P. Kumar, "Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor," *Microprocess. Microsyst.*, Vol. 65, pp. 97–106, Jan. 2019. DOI:10.1016/j.micpro.2019.01.002

10. P. J. Edavoor, S. Raveendran, and A. D. Rahulkar, "Approximate multiplier design using novel dual-stage 4:2 compressors," *IEEE. Access.*, Vol. 8, pp. 48337–51, Mar. 2020. DOI:10.1109/ACCESS.2020.2978773

11. K. M. Reddy, M. H. Vasantha, Y. B. Nithin Kumar, and D. Dwivedi, "Design and analysis of multiplier using approximate 4-2 compressor," *Int. J. Electron.Commun. (AEÜ)*, Vol. 107, pp. 89–97, May 2019. DOI:10.1016/j.aeue.2019.05.021

12. N. V. V. K. Boppana, J. Kommareddy, and S. Ren, "Low-cost and high-performance 8×8 booth multiplier," *Circuits Syst. Signal Process.*, Vol. 38, no. 9, pp. 4357–68, Jan. 2019. DOI:10.1007/s00034-019-01044-x

13. B. Jeevan, and K. Sivani, "A new high-speed multiplier based on carry-look-ahead adder and compressor," in *VLSI Design: Circuits, Systems and Applications, Lecture Notes in Electrical Engineering*, Vol. 469, J. Li, A. Sankar, and P. Beulet, Eds. Singapore: Springer, 2018, pp. 69–78.

14. S. Perri, F. Spagnolo, F. Frustaci, and P. Corsonello, "Parallel architecture of power-of-two multipliers for FPGAs," *IET Circuits Devices Syst.*, Vol. 14, no. 3, pp. 381–9, Feb. 2020. DOI:10.1049/iet-cds.2019.0246

15. A. Garg, and G. Joshi, "Gate diffusion input based 4-bit Vedic multiplier design," *IET Circuits Devices Syst.*, Vol. 12, no. 6, pp. 764–70, Mar. 2018. DOI:10.1049/iet-cds.2017.0454

16. B. N. K. Reddy, "Design and implementation of high performance and area efficient square architecture using Vedic mathematics," *Analog Integr., Circ., Sig., Process*, Vol. 102, pp. 501–6, Mar. 2020. DOI:10.1007/s10470-019-01496-w

17. V. Neha, and A. Greeshma, "Design and execution of enhanced carry increment adder using Han-Carlson and

Kogge-Stone adder technique," in *Third International Conference on Electronics Communication and Aerospace Technology (ICECA 2019)*, Mar. 2019.

18. R. Suganya, and D. Meganathan, "High performance VLSI adders," in *3rd International Conference on Signal Processing, Communication and Networking (ICSCN)*, Oct. 2015.

19. G. Nidhi, K. Pradeep, and M. Anu, "16 Bit power efficient carry select adder," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Nov. 2019.

20. N. Udaya Kumar, K. Bala Sindhuri, D. Sai Satish, and K. Durga Teja, "Implementation and comparison of VLSI architectures of 16 bit carry select adder using Brent Kung adder," in *International Conference on Innovations in Power and Advanced Computing Technologies [i-PACT2017]*, Apr. 2017.

21. A. Shamim, and C. Saurabh, "Modified binary multiplier circuit based on Vedic mathematics," in *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, May 2019.

22. H. Inamul, K. P. Chandan, and C. Saurabh, "Design and FPGA implementation of high speed Vedic multiplier," *Int. J. Comput. Appl.*, Vol. 90, no. 16, pp. 6–9, Mar. 2014. DOI:10.5120/15802-4641

AUTHORS

V Thamizharasan is an assistant professor in the Department of Electronics and Communication Engineering at Erode Sengunthar Engineering College where he received his BE degree in electronics and communication engineering in 2008. He obtained ME degree in VLSI design from Kongu Engineering College, Erode (Anna University, Chennai) in 2010. He has 10 years of experience in the teaching field. He is a life member of ISTE. He has published about 6 papers in the reputed journals and in the national and international conferences. His research interests include low power VLSI, VLSI architecture and VLSI signal processing.

Corresponding author. Email: ecetamil@gmail.com

N Kasthuri is professor in the Department of electronics and communication engineering, Kongu Engineering College, India. She received her BE degree in electronics and communication engineering from AC Tech Karaikudi and ME degree in applied electronics from the Bharathiyar University, Tamilnadu, India. She obtained her doctoral degree in information and communication engineering from Anna University, Chennai, Tamilnadu, India. She has two decades of teaching experience. She has published about 52 papers in the reputed journals and in the national and international conferences. Her research interests cover signal processing, speech signal processing and embedded systems. She received research grants from various funding agencies. She is a life member of ISTE.

Email: kasthuri@kongu.ac.in