



OPEN

Design of efficient binary multiplier architecture using hybrid compressor with FPGA implementation

V. Thamizharasan^{1✉} & V. Parthipan²

In signal processing applications, the multipliers are essential component of arithmetic functional units in many applications, like digital signal processors, image/video processing, Machine Learning, Cryptography and Arithmetic & Logical units (ALU). In recent years, Profuse multipliers are there. In that, Vedic multiplier is one of the high-performance multiplications and it is used to signal/image processing applications. In order to ameliorate the performance of this multiplier further, by proposed a novel multiplier using hybrid compressor. The proposed hybrid compressor-based multiplier is designed and implemented in Field programmable Gate Array (FPGA—spartan 6). The synthesis result shows that the speed of proposed hybrid compressor-based multiplier gets improved as compared to Array multiplier (35.83%), Wallace tree multiplier (34.58%), Vedic Multiplier based on Carry look ahead adder (CLA) (28.49%), Vedic Multiplier based on Ripple carry adder (RCA) (20.65%), Booth Multiplication (21.65%) and Vedic Multiplication based on Han-Carlson Adder (HCA) (20.10%) and Hybrid multiplier using Carry Select Adder (CSELA) (17.81%) and Hybrid Vedic Multiplier (7.15%).

Keywords Hybrid, Multiplier, Compressor, Xilinx, Field programmable gate array (FPGA), Spartan6

In the smart life Digital era, Multimedia, Cryptography, Artificial Intelligence, Machine Learning, Deep Learning, Internet of Things (IoT) and 4G/5G Technology are going to play a crucial role¹. In these technologies, they involve huge basic arithmetic calculations, such as multiplication and addition with a huge volume of data². In that, the power utilization, area utilization and delay are the very important parameters to decide the entire system performance. The optimization of power consumption and area is great demand and very challenging assignment in High performance multipliers and adders³.

There are various multiplier architectures available based on the major classification of serial multiplier, parallel multiplier and serial/parallel multiplier. Also, various algorithms are available to implement the high-performance multiplication operation in the literature. Similarly various adders are available in the literature. They are classified based on the rippling of intermediate carry. Consequently, it is essential to scrutinize the utilization of area, power and speed of the multiplication and addition functional unit in the signal processing applications.

The pivot of the presented work is designing a high-speed multiplier using hybrid compressor. The showcased model modifies the multiplier architecture using 12:6 and 5:3 hybrid Compressor and 3:2 compressor and it implemented in FPGAs. The results claimed that optimal Speed, retaining architecture of parallelism as compared to its high-performance variant.

The paper is systematized as follows. The existing adders is presented “Review of adders” section. In “Proposed multiplier” section, the existing multipliers are described. In “Performance analysis” section, the proposed hybrid compressor-based multiplier is projected. In “Conclusion” section, the experimental results are compared and analyzed with existing technique. Finally, “Conclusion” section provides the conclusion of this article.

Review of adders

In digital Integrated Circuits (IC) design, addition operation is the brain of arithmetic module. It has special significance in portable devices, compactible devices, cryptography, Artificial Intelligence, Machine Learning, signal, audio, image and video processing⁴. The above applications require multi-bit adders and produce high

¹Department of ECE, Erode Sengunthar Engineering College, Erode, India. ²Department of ECE, Sri Eshwar College of Engineering, Coimbatore, India. ✉email: ecetamil@gmail.com

delay due to rippling of long carry propagations are the great issue in adders. Hence, there is a need in designing of high speed and low complexity architectures of adder⁵.

The ripple carry addition has less power utilization and occupies minimum area as compared to most of the adder architectures. But it will linearly increase the delay in bit size of the adder makes and it will be unsuitable for a high-speed application⁶.

In carry save adder, the 3-bits are added simultaneously and carry are stored in present stage and not propagated through the subsequent stages. The speed of this adder is improved due to carry generation. The pros of this adder are adding three input values at a time. However, cons of this adder are occupied large area due to more number of transistor and consume large power.

There are few high-speed adders available in literature such as carry skip adder, the carry look ahead adder (CLA), conditional adder, carry select adder (CSELA) and their combinations.

Carry Look Ahead Adder (CLA) is one of the fast adders. Because, based on generation and propagation principle the sum and carry generation can be done at same time. Here the output carry depends on only input carry irrespective of the bit size. The pros of this adder are less delay as compared to Ripple Carry adder (RCA). However, cons of this adder are occupied large area due to separate circuit for sum and carry generation. Also, the complexity of the circuit increases with increase bit size⁷.

Another one of the fastest adder is Carry select adder. This adder can perform the addition operation based on the pre assumption of input carry (Assume input carry $C_{in} = 1$ or 0). This adder can be improved the speed as compared to Ripple Carry adder (RCA) and Carry Look Ahead Adder (CLA), but it is consumed more area due to dual Ripple Carry adders (RCAs).

To reduce the size of Carry Select adder (CSELA) the binary to excess one conversion (BEC) is introduced. The Modified Carry Select adder (CSELA) designed by the second Ripple Carry adder (RCA) (for $C_{in} = 1$) is replaced by Binary to Excess one Converter (BEC). This Binary to Excess one Converter (BEC) occupied the less numbers of Gate (transistor) as compared to the Ripple Carry adder (RCA). Hence the area is reduced⁷.

It is further improved the delay of the Carry Select adder (CSELA) by introducing the concept of parallel prefix adder. It is named as Brunt Kung Adder (BK adder). The delay of circuit is improved by modifying the first stage by BK adder (Replacing RCA for $C_{in} = 0$ with BK adder) and second stage by Binary to Excess one Converter (BEC) (Replace RCA for $C_{in} = 1$ with BEC)⁸.

Furthermore, the utilization of area by Common Boolean logic (CBL) was developed. In this logic, the existing resource will be utilized to minimize the number of gates. In Carry Select adder (CSELA), the RCA for $C_{in} = 1$ is replaced by CBL⁸.

For optimizing the speed and area of CSELA addition Han-Carlson adder is further introduced. This adder is designed on the features of Kogge stone and Brent Kung adder (BK). Since Koggestone have a less delay and Brent Kung has a less area utilization. The combination of BK adder (For $C_{in} = 0$) and Koggestone adder (for $C_{in} = 1$) is named as Han Carlson adder⁹.

To ameliorate the speed of operation by introducing the parallel carry computation in addition is called as Weinberger recurrence algorithm. In Han-Carlson adder (HCA), the BK adder (for $c_{in} = 0$) is changed by the modified BK adder and Koggestone adder (for $c_{in} = 1$) is changed by the 5 bit Binary to Excess one Converter (BEC) module⁹.

Review of multipliers

In numerous applications, multiplication is one of the prominent components which conquest the overall performance of the Signal processing system. Multiplication is a key arithmetic operation, that has a most important primitive component in the performance of various digital processors, machine learning, internet of things, deep learning, Finite impulse response (FIR) filters, convolution, Fast Fourier Transform (FFT), distributed computing, Arithmetic and Logic Unit (ALU), signal/Image processing, and multimedia applications. It decides the area, delay, and overall performance of parallel implementations¹⁰.

In signal processing application devices, such as smart phones, Laptop, tablet, and Personal computers are required high performance multiplier with the importance of minimum area utilization is very important one. There are various methods available to implement the multiplication operation.

Multiplier are mainly categorized by two methods such as serial and parallel multiplication. In that, serial multiplication, every bit of multiplier is used for calculating the partial products. While parallel multiplication, the partial products of every bit of multiplier are calculated in parallel. The performance (speed) of the multiplication mainly depends on the generation of partial products. The speed is optimized in parallel implementation of multiplication with penalty of area utilization¹¹.

In multiplication operation, the bit by bit AND operation is followed by addition of partial products with the help of half and full adders. Speed of a multiplication mostly depends on the number of partial products generations and accumulation.

The multipliers are categorized based on reduction of partial products. There are Array multiplication, Wallace multiplication, Bypassing multiplier, Booth multiplication, Vedic multiplication and Booth recorded Wallace tree multiplication, Baugh Wooley multiplication, Braun multiplication and etc.,¹².

Approximation gives an alternate technique to optimizing the accuracy of multiplication without compensating another circuit.

In addition to the above multiplication, the truncated multiplication is a highly specialized type of multiplication that only determines the part of the product. Because, Approximate calculation is the best technique for an error tolerant and energy efficient applications, that exhibition of essential tolerate the erroneousness, such as signal processing and multimedia applications. Approximate computing was reduced the accuracy of

multiplication, nevertheless it still provides a faster result with less power consumption, this method was used in some part of arithmetic circuits in signal processing applications.

A various arithmetic architecture was designed using exact and deterministic principles. But, many applications, namely multimedia, signal/image processing can allow/permit the errors and generate results which are better for human perceptions. Since exact solutions are enough in these error-tolerant application to allow the computing systems to maintain quality and accuracy of the design. Hence it is necessary to concentrate, analyze and investigate the approximate additions and multiplications.

Review of compressor

Multipliers are omnipresent and essential arithmetic module in Very Large-Scale Integration VLSI architectures, particularly in signal processing applications and general-purpose processors. The speed and power utilization of multipliers are the main parameter to determine the performance of system. The optimum performance of multiplication will be achieved on the minimization of partial products generation during the multiplication process. The efficiency of multiplication can be optimized by minimizing the number of partial products within limit of power consumption. Hence, the compressors are specifically designed as an arithmetic module for optimizing the speed and area of an arithmetic module¹³.

There are various methods extensively applied to minimize the partial products such as recording technique, Booth Recorder, Wallace tree multiplication through carry save adder, Dadda multiplication using modified Wallace tree design and Carry Save Adder.

Compressor

The compressor is an important module and it is mostly used in Very Large-Scale Integration (VLSI) circuits and systems and their applications. It is commonly used as a processing element.

A (m, n) compressor consists of m-bit input with carry inputs 'Cin' and produces n-bit output with carry output Cout. The main advantages of compressors are producing the output without rippling of carry. Because of Cout is not depended on the input carry Cin. Also, horizontal and vertical signal paths of compressors are simple and regular structure than other existing technique¹⁴.

3:2 compressor

The most common and simple compressor is 3:2 compressor and it is called as full adder. It is defined as single bit adder and it consists of three inputs and two outputs. Also, it is designed with many Very Large-Scale Integration (VLSI) logic circuit design technique. It mainly consists of the three modules. The 1st module is to compute XOR / XNOR operation. The 2nd module is to determine the output 'Sum'. The 3rd module is a to evaluate the carry output signal. Full adder can be constructed with the help of these three modules.

The general equation for 3:2 Compressor is

$$A + B + C = \text{Sum} + \text{Carry}(2 - \text{bit}) \quad (1)$$

where

A, B and C are inputs.

Sum and Carry are outputs.

The block diagram and structure of 3:2 compressor is displayed in Fig. 1¹⁴. This structure consists of 2-Exclusive OR gates in critical path. The output 'sum' is computed from 2nd XOR gate and carry is produced by the multiplexer (MUX)¹³. The output equations for 3:2 compressor is

$$\text{Sum} = A \text{ XOR } B \text{ XOR } C \quad (2)$$

$$\text{Carry} = (A \text{ XOR } B) \cdot C + (A \text{ XOR } B)' \cdot A \quad (3)$$

The 3–2 compressor has minimum delay as compared to conventional full adder. In order to improve the speed of compressor, XOR gate is replaced by Multiplexer. In that selection input of multiplexer (MUX) is

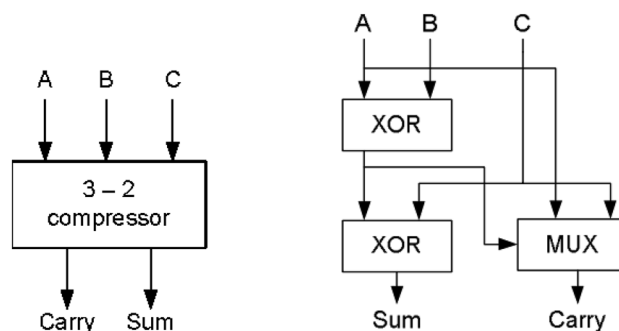


Figure 1. Architecture of 3:2 compressor¹⁴.

available before the input signal arrives. Hence, it minimizes the delay of the circuit due to reduction of switching time in critical path of transistor. This will reduce the significant amount of delay. The modified 3:2 compressor output equation is

$$\text{Sum} = (A \text{ XOR } B) \cdot C' + (A \text{ XOR } B)' \cdot C \quad (4)$$

$$\text{Carry} = (A \text{ XOR } B) \cdot C + (A \text{ XOR } B)' \cdot C \quad (5)$$

4:2 compressor

Generally, 4:2 compressor is a combination of pair of full adders. The general structure of 4:2 compressor is shown in Fig. 2. It accepts 4 inputs with one carry inputs and compress the two outputs namely 'sum' and 'carry'. It also generates the intermediate carry bit 'Cout'.

The relation between inputs and outputs of 4:2 compressor is

$$X1 + X2 + X3 + X4 + \text{Cin} = \text{Sum} + \text{Carry} + \text{Cout}. \quad (6)$$

where

$$\text{Sum} = X1 \oplus X2 \oplus X3 \oplus X4 \oplus \text{Cin} \quad (7)$$

$$\text{Cout} = (X1 \oplus X2) X3 + (X1 \oplus X2) X1 \quad (8)$$

$$\text{Carry} = (X1 \oplus X2 \oplus X3 \oplus X4) \text{Cin} + (X1 \oplus X2 \oplus X3 \oplus X4) X4 \quad (9)$$

A 4:2 compressor is generally designed by a combination of multiplexers and XOR gates. The 4:2 compressor was designed by simply cascading of 2 full adders and it is shown in Fig. 2¹³. It achieves the critical path delay of 3 XOR gate delay¹⁴.

In order to optimize the performance of the 4:2 compressor by design of multiplexer using full swing pass transistor logic to achieve optimized power utilization. The pros of this design are sum generation not dependent on Cout generation, as compared to 4:2 compressor structure shown in Fig. 2. Also, this design was achieved 18% delay improvement as compared to conventional full adder design.

Further it improves the speed of operation, proposed a modification of compression unit by rearranging the Boolean equation to improve the delay of carry computation. The modification of compression unit was designed by combined a NAND and NOR gate into an XOR gate.

For optimizing the performance of the arithmetic circuits by rearranging the Boolean/logic equation or derive the new logic equation from the truth table. The change of Multiplexer against with XOR logic in right places of the existing technique is implemented through Shannon's expansion technique¹⁵.

The performances are mainly determined by their speed of arithmetic calculation. Arithmetic computation Adders, Shifters and Multipliers are the crucial module of any Signal Processing applications. In addition, process due to huge carry computation delay and sequential behavior, existing digital system architecture is slow in nature.

Also, the Multiplication operation determines the speed of the most Digital Signal Processing (DSP) applications, hence it required high-speed multiplier for an efficient data path circuit design. In order to improve the speed of a multiplier, it minimizes the number of the partial product, since multiplication leads to series of addition of partial products.

Proposed multiplier

The major constraint of the multipliers are the speed (delay) of operation, Hence it is necessary to focus the critical path delay of an multiplier. Hence, the hybrid Compressor based multiplier is proposed to optimize the delay of multiplier as compared to the existing methods.

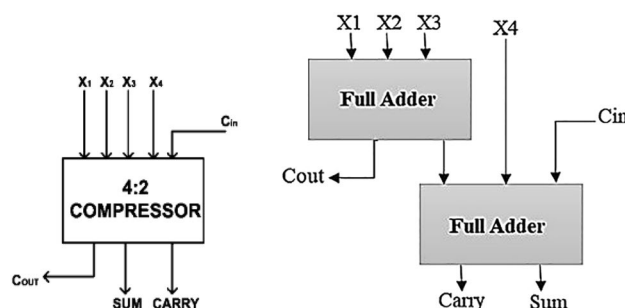


Figure 2. Structure of 4:2 compressor¹³.

Hybrid technique

Using more than one logic structure (styles) to design the Module of a system is known as hybrid technique. There are two types of logic styles are followed in the hybrid technique. They are (1) Homogeneous styles, it using same type of circuit style in all the stages is called Homogeneous structure. (2) Heterogeneous styles, it using the different type of circuit style in different stages is called Heterogeneous design. Consider an example of full adder. The full adder contains three modules such as two half adder unit and one OR gate module. By using two different logic styles in half adders and another one logic style is used in OR gate module. This type of structure is called as hybrid full adder. This hybrid technique will be optimizing the speed, size and power utilization of the circuit. The basic architecture of hybrid technique is shown in Fig. 3¹⁶.

Hybrid compressor

In order to optimize the delay and size of multiplication architecture, it incorporates the above-mentioned Compressors in different modules (stages) of the multiplier architecture.

The architecture of 12:6 hybrid compressor is shown in Fig. 4. This compressor consists of 3 stages each with 4:2 compressor. The 4:2 compressor (design-1), 4:2 compressor (design-2) and 4:2 compressor (design-3) are used in stage1, stage2 and stage 3, respectively. Also shown in Figs. 5, 6 and 7 respectively.

The Fig. 5 and 6, 4:2 compressor was achieved considerable reduction in Area delay product and power delay product as compared to existing design¹³. It also is improved the delay of minimum 7% and maximum 12.5%¹³. Similarly, this compressor will be suitable for high performance multiplier and their relevant applications.

The refined logical relations for Design-1,2 and 3¹³ are

$$\text{Condition} = X_4' \text{Cin} + \text{Cin}' X_4. \quad (10)$$

$$S_i = X_3'(X_1 \oplus X_2) + X_3(X_1 \odot X_2). \quad (11)$$

$$\text{Sum} = S_i'(X_4 \oplus \text{Cin}) + S_i(X_4 \odot \text{Cin}) \quad (12)$$

$$\text{Carry} = \text{Cin}(X_4 \oplus \text{Cin}) + S_i(X_4 \odot \text{Cin}) \quad (13)$$

$$\text{Cout} = (X_1 + X_2)X_3 + X_3(X_1 \cdot X_2) \quad (14)$$

The Architecture 5:3 hybrid compressor is shown in Fig. 8. This compressor consists of 2 stages each with 3:2 compressor. The 3:2 compressor (design-1) and 3:2 compressor (design-2) are used in stage1 and stage2 respectively. Also displayed in Figs. 9 and 10 respectively.

Proposed multiplication using hybrid compressor

The block diagram of the proposed hybrid compressor-based multiplier is displayed in Fig. 11. This structure is a 4 * 4 multiplier. The partial products are computed with help of series of AND gates.

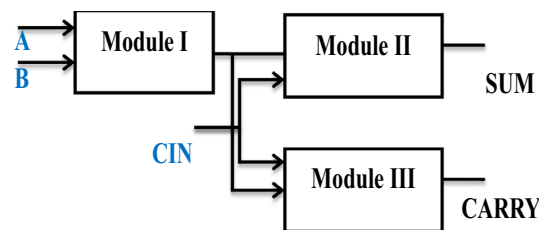


Figure 3. Block diagram for hybrid adder⁸.

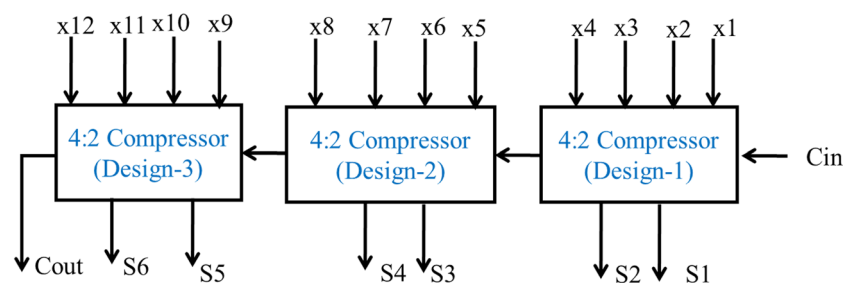
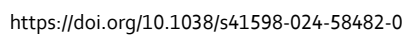


Figure 4. Architecture of 12:6 hybrid compressor.



nature portfolio

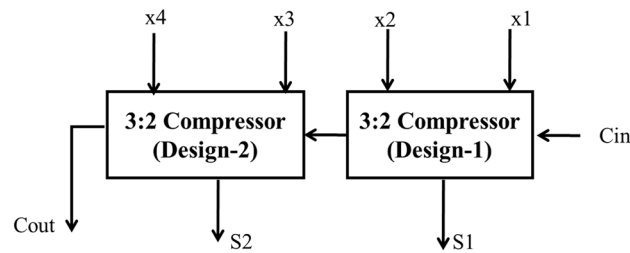


Figure 8. Architecture of 5:3 hybrid compressor.

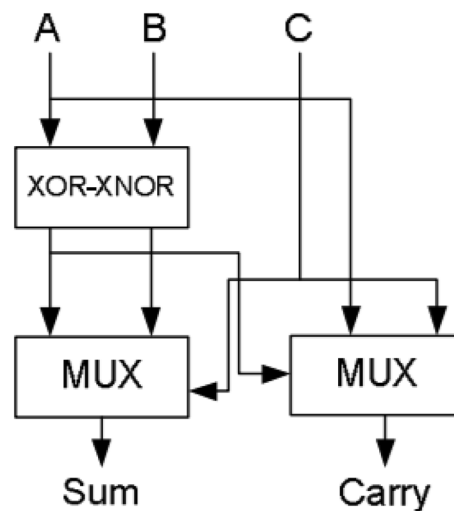


Figure 9. Architecture of 3:2 compressor (Design-1)¹⁸.

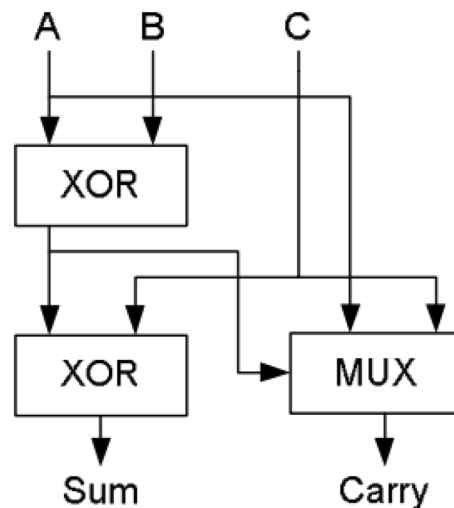


Figure 10. Structure of 3:2 compressor (Design-2)¹⁸.

Partial product computation is a 1st process of multiplier operation. Partial products are generated by multiplying each bit of multiplicand by multiplier. Consider an example of 4*4 Multiplication, Multiplier 'A' having 4 bits (A0 to A3) and Multiplicand 'B' having 4 bits (B0 to B3). In Computation of Partial product, the 1st step is multiplying LSB of B(B0) with every bit of multiplier A (A0 to A3) and outputs are stored in C (C0 to C3) each with 4 bit. It generally represented as $C_0 = B_0.A_0$ (logical AND operation), $C_1 = B_0.A_1$ and so on. Similarly, it multiplies B1, B2 and B3 bit with every bit of A in subsequence steps. The structure of partial product computation is displayed in Fig. 12.

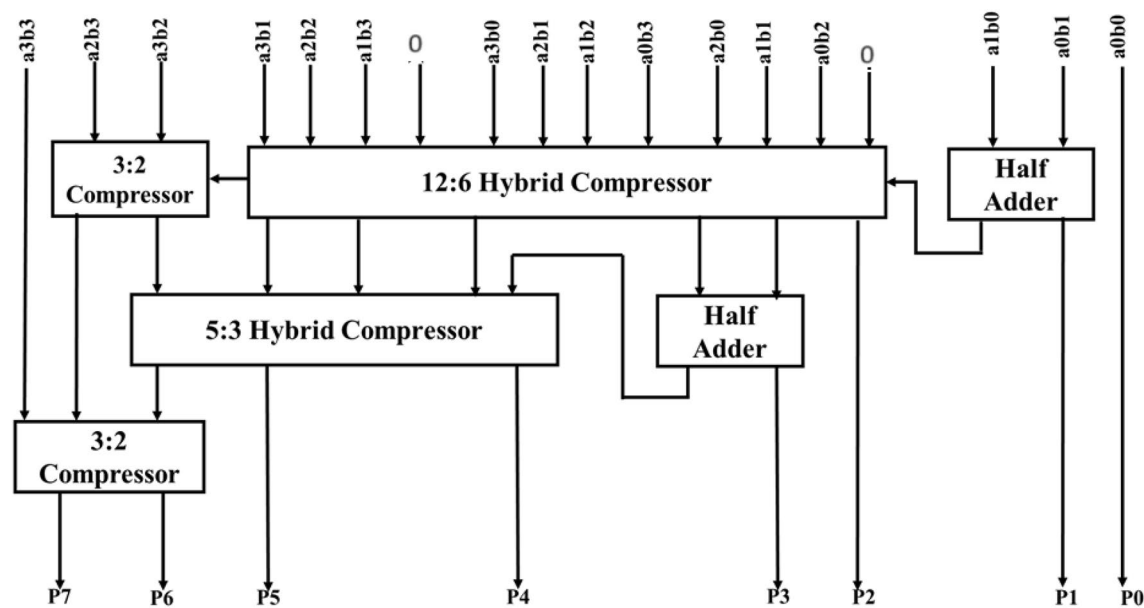


Figure 11. Structure of proposed 4*4 Multiplier using hybrid compressor.

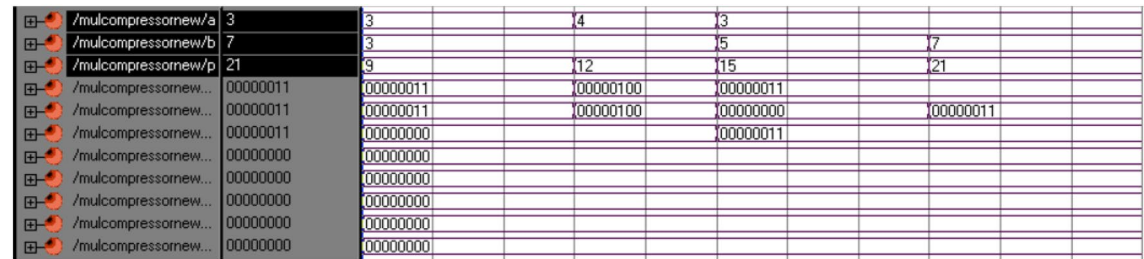


Figure 12. Simulation result of proposed hybrid compressor based multiplier.

The proposed hybrid compressor-based multiplier consists of three stages. In every stage, various-sized hybrid compressor and half adder are used. Namely, 12:6 compressor (combination of various design styles of 4:2 compressor) is used in the first stage (3 numbers of 4:2 compressor), 5:3 compressor (combination of various design styles of 3: 2 compressor) is used in the 2nd stage (two numbers of 3:2 compressor) and a one 3:2 compressor is used in the third stage (one number of 3:2 compressor).

Performance analysis

The different techniques of multiplier, adder and compressor are deliberated in the “Review of adders”, “Proposed multiplier” & “Performance analysis” sections and proposed multiplier is elucidated in “Conclusion” section. Simulation of all multiplier architecture done in XILINX ISE (Integrated Software Environment). The Figs. 12 and 13 are display the input/output wave form and percentage of device utilizations of hybrid compressor-based multiplier respectively. The same is implemented in spartan6 Field programmable Gate Array (FPGA) device. All the multiplication technique are verified their input and output individually.

The synthesized results indicated that the delay, Number of Look Up Tables (LUTs) (Size), power consumption of several multiplier technique and it shown in the Table 1. The percentage of speed improvements in terms of delay for hybrid compressor-based multiplier is shown in Fig. 15 as compared to existing multiplier techniques.

The comparison shows the speed in terms of delay of the hybrid Compressor based multiplier is improved 35.83%, 34.58%, 21.65%, 28.49%, 20.65%, 20.10%, 17.81%, 07.15% as compared to Array Multiplication, Wallace

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 13. Device utilization for proposed multiplier using hybrid compressor.

S. No.	Name of the technique	Delay (ns)	No. of LUTs*	Power (uW)	ADP*	PDP*
1	Proposed Multiplier using hybrid compressor	13.75	83	58.25	1141.25	800.93
2	Hybrid Vedic multiplication (VM) ¹²	14.81	86	55.63	1273.66	823.80
3	Hybrid multiplier based on carry select adder (CSELA) ⁸	16.73	90	53.43	1505.7	893.88
4	Vedic multiplier using Han Carlson Adder (HCA) ¹²	17.21	126	55.42	2168.46	953.77
5	Vedic multiplier using Ripple Carry Adder (RCA) ¹²	17.33	108	53.51	1871.64	927.32
6	Revised Booth Multiplier (RBM) ⁸	17.55	91	53.73	1597.05	942.96
7	Vedic Multiplier (VM) using Carry Look ahead Adder (CLA) ¹²	19.23	113	54.93	2172.99	1056.3
8	Wallace tree multiplier (WTM) ⁸	21.02	116	59.24	2438.32	1245.2
9	Array Multiplier (AM) ⁸	21.43	84	52.52	1800.12	1125.5

Table 1. Simulation result of different multiplier with Spartan6 FPGA implementation. *ADP, area delay product; PDP, power delay product; LUTs, look up tables.

tree multiplication, Booth Multiplier, Vedic Multiplier using Carry Look ahead Adder (CLA), Vedic Multiplication using Ripple Carry Adder (RCA), Vedic Multiplication using Han Carlson Adder (HCA), Hybrid Multiplier using Carry Select Adder (CSELA) and Hybrid Vedic Multiplier respectively.

Also, the Area delay product (ADP) of the hybrid compressor-based multiplier is enriched by 36.60%, 53.19%, 47.48%, 39.02%, 28.54%, 47.37%, 24.20% and 10.39% as compared to Array Multiplication, Wallace tree multiplication, Vedic Multiplication using Carry Look ahead Adder (CLA), Vedic Multiplier using Ripple Carry Adder (RCA), Booth Multiplication, Vedic Multiplier using Han Carlson Adder (HCA), Hybrid Multiplier using Carry Select Adder (CSELA) and Hybrid Vedic Multiplier respectively. As well as Power Delay Product (PDP) of the hybrid compressor-based multiplier is enhanced by 28.83%, 35.67%, 24.14%, 13.62%, 15.06%, 16.02%, 10.39% and 2.77% as compared to Array Multiplication, Wallace tree multiplication, Vedic Multiplication using Carry Look ahead Adder (CLA), Vedic Multiplier using Ripple Carry Adder (RCA), Booth Multiplication, Vedic Multiplier using Han Carlson Adder (HCA), Hybrid Multiplier using Carry Select Adder (CSELA) and Hybrid Vedic Multiplication respectively.

The Fig. 14 is displayed the Power Delay Product (PDP) and Area Delay Product (ADP) of different multiplication technique. The proposed compressor-based multiplier has a considerable improvement in PDP and ADP.

The comparison about delay improvement (%) of proposed multiplier using hybrid compressor is displayed in the chart as displayed in Fig. 15. Also delay of various multiplication techniques of research article is shown in Table 2. In that table it shows that the significant improvement delay is there in proposed multiplier using hybrid compressor.

Conclusion

In this investigation, Hybrid compressor-based multiplier architecture is experimented with various design styles of compressor. The results clearly indicated that the hybrid compressor-based multiplication has considerable improvement in speed of multiplication with reduced size as compared to existing multiplication technique. The proposed hybrid compressor-based multiplier architecture is successfully synthesized and simulated using Xilinx

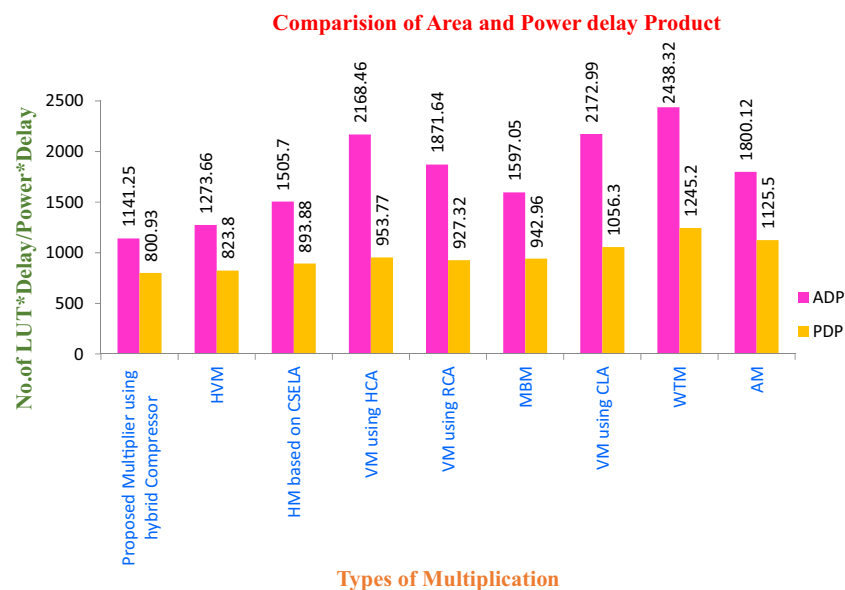


Figure 14. Analysis of ADP and PDP of various multiplier.

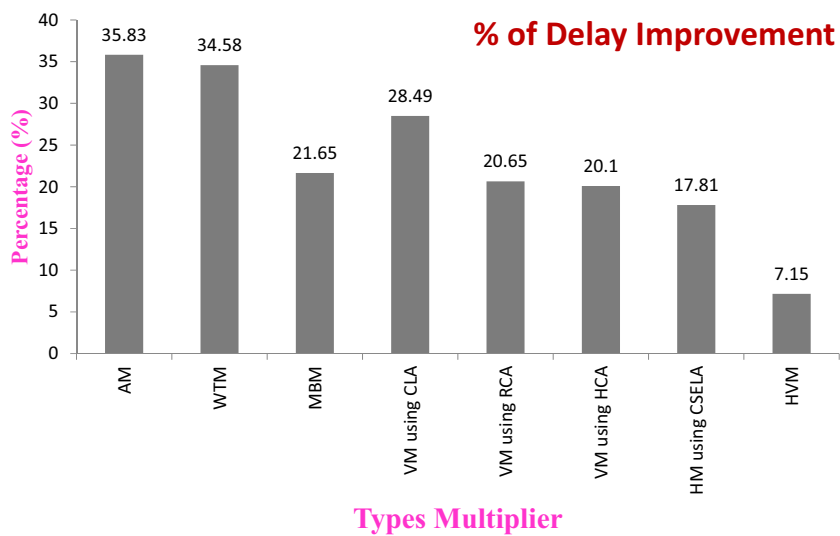


Figure 15. Percentage of delay improvements in Hybrid compressor-based multiplier as compared to other multiplier.

S. No.	Name of the multiplier in existing article	Delay (ns)	Area (No. of LUTs)	Power (uW)
1	High speed Vedic Multiplication ¹⁹	23.64	192	51.78
2	Hybrid Vedic Multiplier ¹²	14.80	86	55.70
3	Revised Vedic multiplier Design-2 ¹⁵	18.39	–	–
4	Revised Vedic Multiplier Design-1 ¹⁵	18.61	–	–
5	High Speed Hybrid Multiplier ⁸	16.68	90	53.54
6	High Performance Multiplier ¹⁷	20.70	92	50.17
7	Improved Binary Multiplication ¹⁸	18.46	125	51.27
8	Proposed Multiplier using hybrid compressor	13.75	83	58.25

Table 2. Delay analysis of 8*8 multiplier in existing article.

software and implemented on Field Programmable Gate Array (FPGA) boards. The synthesized result shows that the delay of proposed hybrid compressor-based multiplier is improved 35.83%, 34.58%, 21.65%, 28.49%, 20.65%, 20.10%, 17.81%, 07.15% as compared to Array Multiplication, Wallace tree multiplier, Booth Multiplier, Vedic Multiplier using Carry Look Ahead Adder (CLA), Vedic Multiplier Ripple Carry Adder (RCA), Vedic Multiplication using Han Carlson Adder (HCA), Hybrid Multiplier using Carry Select Adder (CSELA) and Hybrid Vedic Multiplier respectively. The comparative analysis of implementation results motivates authors to conclude that the proposed Hybrid compressors-based multiplier shall be a desirable choice for the implementation of high-performance signal²⁰ and image processing²¹ and other related applications²².

Data availability

The datasets used and/or analyzed during the current study available from the corresponding author on reasonable request.

Received: 21 December 2023; Accepted: 29 March 2024

Published online: 11 April 2024

References

1. Panda, A. K., Palisetty, R. & Ray, K. C. High-speed area-efficient VLSI architecture of three-operand binary adder. *IEEE Trans. Circuits Syst. I Regul. Pap.* **67**(11), 3944–3953. <https://doi.org/10.1109/TCSI.2020.3016275> (2020).

2. Perri, S., Spagnolo, F., Frustaci, F. & Corsonello, P. Efficient approximate adders for FPGA-based data-paths. *Electronics* **9**(9), 1529. <https://doi.org/10.3390/electronics9091529> (2020).

3. You, H., Yuan, J., Tang, W. & Qiao, S. An energy and area efficient carry select adder with dual carry adder cell. *Electronics* **8**(10), 1129–1139. <https://doi.org/10.3390/electronics8101129> (2019).

4. Radhakrishnan, P. & Themozhi, G. FPGA implementation of XOR-MUX full adder based DWT for signal processing applications. *J. Elsevier Microprocess. Microsyst.* **73**, 1–14. <https://doi.org/10.1016/j.micpro.2019.102961> (2020).

5. Garg, B. & Patel, S. K. Reconfigurable carry look-ahead adder trading accuracy for energy efficiency. *J. Signal Process. Syst.* **93**, 99–111. <https://doi.org/10.1007/s11265-020-01542-1> (2021).

6. Palanisamy, G., Natarajan, V. K. & Sundaram, K. Area-efficient parallel adder with faithful approximation for image and signal processing applications. *IET Image Process.* **13**(13), 2587–2594. <https://doi.org/10.1049/iet-ipr.2019.0580> (2018).
7. Balasubramanian, P. & Mastorakis, N. Performance comparison of carry-Lookahead and carry-select adders based on accurate and approximate additions. *Electronics* **7**(12), 369–381. <https://doi.org/10.3390/electronics7120369> (2018).
8. Thamizharasan, V. & Kasthuri, N. High speed hybrid multiplier design using a hybrid adder with FPGA implementation. *IETE J. Res.* **69**(05), 2587–2594. <https://doi.org/10.1080/03772063.2021.1912655> (2021).
9. Thamizharasan, V. & Kasthuri, N. Design of proficient two operand adder using hybrid carry select adder with FPGA implementation. *IETE J. Res.* <https://doi.org/10.1080/03772063.2022.2071771> (2022).
10. Thamizharasan, V. & Parthipan, V. An efficient VLSI architecture for FIR filter using computation sharing multiplier. *Int. J. Comput. Appl.* **54**(14), 1–6 (2012).
11. Thamizharasan, V. & Kasthuri, N. FPGA implementation of high performance digital FIR filter design using a hybrid adder and multiplier. *Int. J. Electron.* **110**(4), 587–607. <https://doi.org/10.1080/00207217.2022.2098387> (2023).
12. Thamizharasan, V. & Kasthuri, N. FPGA implementation of proficient Vedic multiplier architecture using hybrid carry select adder. *Int. J. Electron.* <https://doi.org/10.1080/00207217.2023.2245194> (2023).
13. Priyadharshni, M. *et al.* Logically optimal novel 4:2 compressor architectures for high-performance applications. *Arab. J. Sci. Eng.* **45**, 6199–6209. <https://doi.org/10.1007/s13369-020-04503-9> (2020).
14. Danasekar, S. An area efficient vedic multiplier for FFT processor implementation using 4–2 compressor adder. *Int. Journal of Electron.* <https://doi.org/10.1080/00207217.2023.2278434> (2023).
15. Sivanandam, K. & Kumar, P. Design and performance analysis of reconfigurable modified Vedic multiplier with 3-1-1-2 compressor. *Microprocess. Microsyst.* **65**, 97–106. <https://doi.org/10.1016/j.micpro.2019.01.002> (2019).
16. Kandpal, J., Tomar, A., Agarwal, M. & Sharma, K. K. High-speed hybrid-logic full adder using high-performance 10-T XOR–XNOR Cell. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **28**(6), 1413–1422. <https://doi.org/10.1109/TVLSI.2020.2983850> (2020).
17. Shamim, A. & Saurabh, C. Modified binary multiplier circuit based on Vedic mathematics. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, (May 2019).
18. Hussain, I., Pandey, C. K. & Chaudhury S. Design and analysis of high-performance multiplier circuit. In *2019 Devices for Integrated Circuit (DevIC)* 245–247 (Kalyani, India, Mar 2019).
19. Inamul, H., Chandan, K. P. & Saurabh, C. Design and FPGA implementation of high speed Vedic multiplier. *Int. J. Comput. Appl.* **90**(16), 6–9 (2014).
20. Cho, S. M., Meher, P. K., NhatTrung, L. T., Cho, H. J. & Park, S. Y. Design of very high-speed pipeline FIR filter through precise critical path analysis. *IEEE Access* **9**, 34722–34735. <https://doi.org/10.1109/ACCESS.2021.3061759> (2021).
21. Lyakhov, P., Valueva, M., Valuev, G. & Nagornov, N. High-performance digital filtering on truncated multiply-accumulate units in the residue number system. *IEEE Access* **8**, 209181–209190. <https://doi.org/10.1109/ACCESS.2020.3038496> (2020).
22. Prasath, K. S. R. & Subhendu, K. S. An approach for fixed coefficient RNS-based FIR filter. *Int. J. Electron.* **104**(8), 1358–1376. <https://doi.org/10.1080/00207217.2017.1296593> (2017).

Author contributions

Both authors are prepared the article. V.T.: Reviewing, Supervision, Validation, Investigation, Methodology, Software. V.P.: Editing, Methodology, Software.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to V.T.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2024