IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-6, June 2019 E-ISSN: 2347-2693

Design High Speed Radix-4 Complex Multiplier using CBL Adder

Jagrati Nayak[1*], Prashant Purohit[2]

[1,2]Dept. of Electronics and Communication, LNCT, Bhopal

DOI: https://doi.org/10.26438/ijcse/v7i6.10321035 | Available online at: www.ijcseonline.org

Accepted: 28/May/2019, Published: 30/Jun/2019

***Abstract*—** The main objective of this research paper is to design architecture for radix-4 complex Vedic multiplier by rectifying the problems in the existing method and to improve the speed by using the common Boolean logic (CBL). The multiplier algorithm is normally used for higher bit length applications and ordinary multiplier is good for lower order bits. These two methods are combined to produce the high speed multiplier for higher bit length applications. The problem of existing architecture is reduced by removing bits from the remainders. The proposed algorithm is implementation Xilinx software with Vertex-7 device family.

Keywords: - Vedic Multiplier, Complex Multiplier, Common Boolean Logic Adder, Xilinx Software

I. INTRODUCTION

In signal processing, a finite impulse response (FIR) channel is a channel whose reaction to any limited length input is of limited term, since it settles to zero in limited time. Rather than infinite impulse response (IIR) channels, which may have inward input and may keep on responding inconclusively (for the most part decaying). FIR channels are broadly utilized as a part of different DSP applications. In a few applications, the FIR channel circuit must have the capacity to work at high example rates, while in different applications, the FIR channel circuit must be a low-control circuit working at direct example rates. Parallel (or square) handling can be connected to computerized FIR channels to either expand the compelling throughput or lessen the power utilization of the first channel. While consecutive FIR channel usage has been given broad thought, next to no work has been done that arrangements straightforwardly with decreasing the equipment many-sided quality or power utilization of parallel FIR channels. Customarily, the use of parallel handling to a FIR channel includes the replication of the equipment units that exist in the first channel. The topology of the multiplier circuit additionally influences the resultant power utilization. Picking multipliers with more equipment expansiveness as opposed to profundity would decrease the postponement, as well as the aggregate power utilization. A considerable measure of outline strategies for low power computerized FIR channel have been proposed, for instance, a strategy executing FIR channel utilizing simply enrolled adders and hardwired shifts exist.

Parallel duplication is utilized to meet out the present prerequisite. Two kinds of parallel augmentations are exhibit duplication and tree increase. The fundamental multiplier is a basic cluster multiplier and it is planned in view of move and – include task. One of the cases for exhibit increase is the Braun multiplier and is intended for unsigned paired numbers. For tree structure Wallace multiplier is outlined and it is likewise for an unsigned double numbers. In the exhibit augmentation, for marked numbers Baugh – Wooley, Booth Multiplier and Modified Booth Algorithm (MBA) are utilized. Dadda is another kind of multiplier in light of tree structure and is utilized for the increase of the marked numbers. These traditional double multipliers for unsigned numbers are considered for examination. Vedic arithmetic is the arrangement of science followed in old India and mostly manages Vedic scientific formulae and their applications to different branches of math. The word 'Vedic' is gotten from the word 'Veda' which implies the storage facility of all information.

Vedic science was remade from the antiquated Indian sacred writings (Vedas) by Sri Bharati Krishna Tirthaji (1884-1960), after his eight years of research on Vedas. As indicated by his examination, Vedic arithmetic is principally in light of sixteen standards or word-formulae and thirteen sub-end products which are named as Sutras. This is an exceptionally intriguing field and exhibits some viable calculations which can be connected to different branches of Engineering, for example, Computing and Digital Signal Processing. Vedic science diminishes the many-sided quality in figurings that exist in customary arithmetic. By and large there are sixteen sutras accessible in Vedic arithmetic.

Among them just two sutras are pertinent for increased activity. They are Urdhava Triyakbhyam sutra (truly implies vertically and across) and Nikhilam Sutra (truly implies All from 9 and last from 10). Urdhava-Triyakbhyam is a non-specific technique for augmentation. The rationale behind Urdhava Triyakbhyam sutra is especially like the conventional cluster multiplier. Here the paired usage of this calculation is determined in light of a similar rationale utilized for decimal numbers. The double usage of Nikhilam Sutra isn't yet effective.

Vedic multiplier furthermore, basic Boolean rationale snake can contrast and regular strategy which is processed by Vedic multiplier, XOR entryway and half viper. Proposed procedure gives less way delay and less territory. Information grouping of Conventional strategy is significantly more than to proposed technique; however proposed technique has less spread postponement. Region and engendering postponement can be decreased by the guide of basic Boolean rationale viper. This viper will be composed like as swell convey snake.

II. COMPLEX MULTIPLIER

Suppose two numbers are complex then

$$A = A_r + jA_i \quad (1)$$

$$B = B_r + jB_i \quad (2)$$

The product of A and B then

$$P = A \times B \quad (3)$$

$$P = A_r \times B_r - A_i \times B_i + j(A_r \times B_i + A_i \times B_r) \quad (4)$$

$$P_r = A_r \times B_r - A_i \times B_i \quad (5)$$

$$P_i = A_r \times B_i + A_i \times B_r \quad (6)$$

Where P_r and P_i is speaks to the genuine and nonexistent piece of the yield of the mind boggling multiplier.

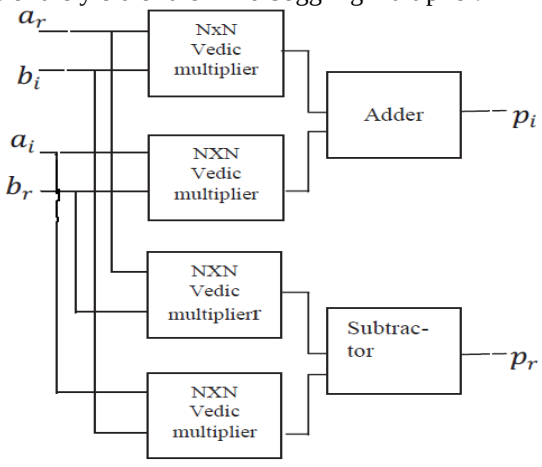

Figure 3: Block Diagram of Complex Multiplier for four Vedic Multiplier

Ar and Ai is speaks to the genuine and fanciful piece of the principal contribution of the unpredictable multiplier. Br and Bi is speaks to the genuine and nonexistent piece of the second contribution of the unpredictable multiplier.

Complex multiplier for four Vedic multipliers is shown in figure 4. In this block diagram reduce four Vedic multipliers to three Vedic multipliers is shown in below:

$$P_r = A_r \times B_r - A_i \times B_i = A_r(B_r + B_i) - B_i(A_r + A_i) \quad (7)$$

$$P_i = A_r \times B_i + A_i \times B_r = A_r(B_r + B_i) + B_r(A_i - A_r) \quad (8)$$

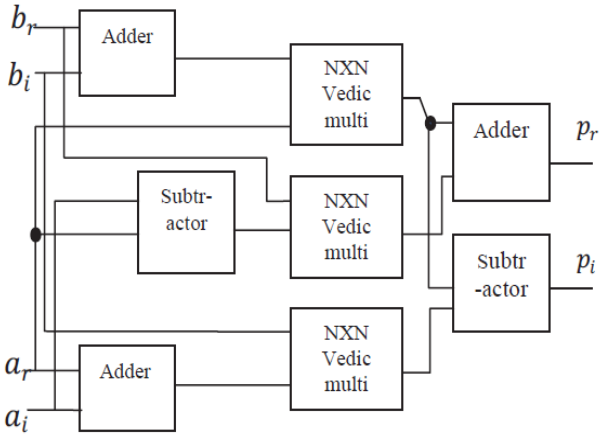

Figure 4: Block Diagram of Complex Multiplier for three Vedic Multiplier

III. VEDIC MULTIPLIER USING CBL

Rationale Diagram of Vedic Multiplier utilizing CBL adder is appeared in figure 5. In the long run, all the planning levels of computerized framework or IC's Packages rely upon number of entryways in a solitary chip that is likewise rung base approach.

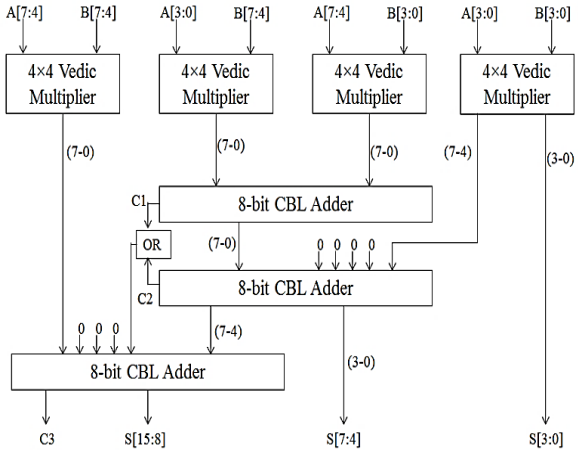

Figure 5: 8-bit Vedic Multiplier using CBL Adder

CBL adder can be decreased in regards to the territory or number of doors. On the off chance that we expel the main XOR door from adjusted KS adder nothing will be changed for result however zone and proliferation deferral will be decreased.

Common Boolean logic Adder

In a zone effective and low power half snake based Carry select viper (CSLA) utilizing normal Boolean rationale is outlined so as to upgrade the general framework execution as far as territory and power as contrast with other existing designs. Half viper is utilized to produce the incomplete entirety for cin=0 and basic Boolean rationale (CBL) is utilized for figuring halfway total for cin=1.

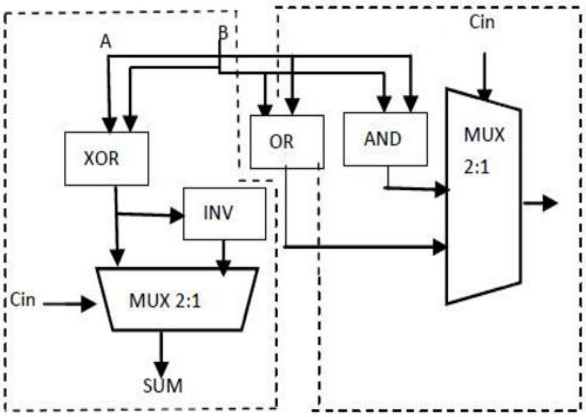

Figure 6: Block Diagram of CBL Adder

This engineering is utilized to expel the recreated viper cells in the traditional CSLA, spare number of entryway tallies and accomplish a low power. Through investigating reality table of a solitary piece full snake we recommend that for producing yield summation and convey motion for cin=0, require just a single XOR door and one AND entryway individually, the yield summation motion for cin=1 is simply the opposite as cin=0. Summed up figure of regular Boolean rationale Adder is appeared in figure 6.

IV. RADIX-4 ALGORITHM

To further decrease the number of partial products, algorithms with higher radix value are used. In radix-4 algorithm grouping of multiplier bits is done in such a way that each group consists of 3 bits as mentioned in table 1. Similarly the next pair is the overlapping of the first pair in which MSB of the first pair will be the LSB of the second pair and other two bits. Number of groups formed is dependent on number of multiplier bits. By applying this algorithm, the number of partial product rows to be accumulated is reduced from n in radix-2 algorithm to n/2 in radix-4 algorithm. The grouping of multiplier bits for 8-bit of multiplication is shown in figure 7.

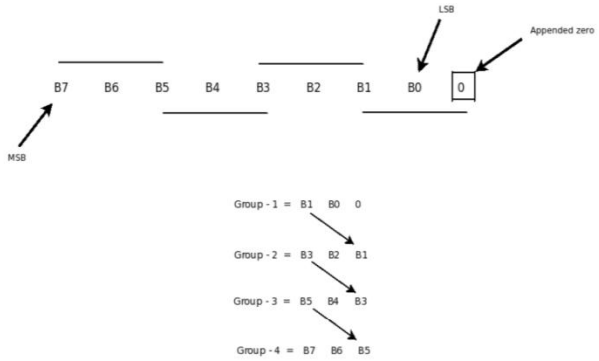

Figure 7: Grouping of multiplier bits in Radix-4 Booth algorithm

For 8-bit multiplier the number groups formed is four using radix-4 booth algorithm. Compared to radix-2 booth algorithm the number of partial products obtained in radix-4 booth algorithm is half because for 8-bit multiplier radix-2 algorithm produces eight partial products. The truth table and the respective operation are depicted in table 1. Similarly when radix-8 booth algorithm is applied to multiplier of 8-bits each group will consists of four bits and the number of groups formed is 3. For 8x8 multiplications, radix-4 uses four stages to compute the final product and radix-8 booth algorithm uses three stages to compute the product. In this thesis, radix-4 booth algorithm is used for 8x8 multiplications because number components used in radix-4 encoding style.

Table 1: Truth Table for Radix-4 Booth algorithm

B_{i+1}	B_i	B_{i-1}	Operation	Y_{i+1}	Y_i	Y_{i-1}
0	0	0	+0	0	0	0
0	0	1	+A	0	1	0
0	1	0	+A	0	1	0
0	1	1	+2A	0	0	1
1	0	0	-2A	1	0	1
1	0	1	-A	1	1	0
1	1	0	-A	1	1	0
1	1	1	-0	1	0	0

V. SIMULATION ANALYSIS

Simulation of these tests should be possible by utilizing Xilinx 14.2 I VHDL instrument. In this paper we are concentrating on engendering delay. Spread postpone must be less for better execution of advanced circuit.

As appeared in table I the quantity of cut, number of LUTs, delay are acquired for the complex Vedic multiplier utilizing basic Boolean rationale viper and past calculation. From the

investigation of the outcomes, it is discovered that the complex Vedic multiplier utilizing basic Boolean rationale snake gives a predominant execution as contrasted and past calculation for Xilinx programming.

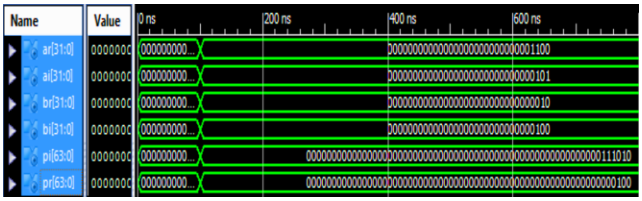

Figure 8: Output Binary Waveform of Radix-4 Complex Multiplier using CBL Adder

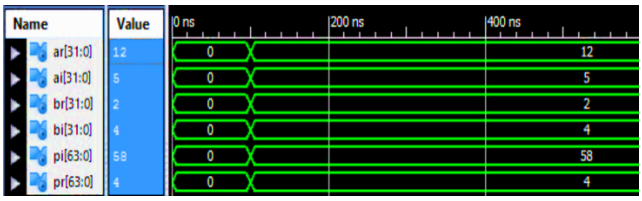

Figure 9: Output Decimal Waveform of Radix-4 Complex Multiplier using CBL Adder

From the analysis of the results, it is found that the complex Vedic multiplier using CBL adder gives a superior performance as compared with previous algorithm for Vertex-7 device family. The output waveform of the complex multiplier using CBL adder is shown in figure 9 and figure 7 respectively.

Table II: Comparison Result for 32-bit Radix-4 Complex Vedic Multiplier for four Vedic Multiplier

Design	Number of LUTs	Number of IOBs	Delay
Radix-4 Complex Vedic Multiplier [2]	10416	256	25.979 ns
Radix-4 Complex Vedic Multiplier using Ripple Carry Adder	10642	256	26.927 ns
Radix-4 Complex Vedic Multiplier using CBL Adder	10222	256	25.204 ns

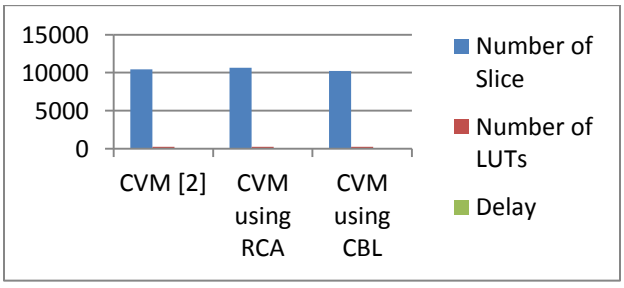

Figure 10: Bar graph of the 32-bit Radix-4 Complex Vedic multiplier

Figure 10 shows the graphical illustration of the performance of CVM using CBL adder discussed in this research work in term of number of slice, number of LUTs and delay. From the above graphical representation it can be inferred that the CVM using CBL adder gives the best performance as compared with previous algorithm.

VI. CONCLUSION

In this paper design of CBL adder, Vedic multiplier, complex Vedic multiplier and FIR filter is presented. From implementation results it is observed that the FIR filter based on complex Vedic multiplier consumes less delay compare to previous design. The architecture designs of 32 x32-bit; Modified Radix-4 Booth Encoder Multiplier is done.

REFERENCE

[1] D. Kalaiyarasi and M. Saraswathi, "Design of an Efficient High Speed Radix-4 Booth Multiplier for both Signed and Unsigned Numbers", 4th International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), IEEE 2018.

[2] Prof. S. B. Patil, Miss. Pritam H. Langade, "Design of Improved Systolic Array Multiplier and Its Implementation on FPGA", International Journal of Engineering Research and General Science Volume 3, Issue 6, November-December, 2015

[3] Elisardo Antelo, Paolo Montuschi and Alberto Nannarelli, "Improved 64-bit Radix-16 Booth Multiplier Based on Partial Product Array Height Reduction", IEEE Transactions On Circuits And Systems—I: Regular Papers, Vol. 64, No. 2, February 2017.

[4] Kavita and Jasbir Kaur, "Design and Implementation of an Efficient Modified Booth Multiplier using VHDL", Special Issue: Proceedings of 2nd International Conference on Emerging Trends in Engineering and Management, ICETEM 2013.

[5] Shiann-Rong Kuang, Jiun-Ping Wang and Cang-Yuan Guo, "Modified Booth Multipliers With a Regular Partial Product Array", IEEE Transactions On Circuits And Systems—Ii: Express Briefs, Vol. 56, No. 5, May 2009.

[6] S. Vassiliadis, E. Schwarz, and D. Hanrahan, "A general proof for overlapped multiple-bit scanning multiplications," IEEE Trans. Comput., vol. 38, no. 2, pp. 172–183, Feb. 1989.

[7] D. Dobberpuhl et al., "A 200-MHz 64-b dual-issue CMOS microprocessor," IEEE J. Solid-State Circuits, vol. 27, no. 11, pp. 1555–1567, Nov. 1992.

[8] E. M. Schwarz, R. M. A. III, and L. J. Sigal, "A radix-8 CMOS S/390 multiplier," in Proc. 13th IEEE Symp. Comput. Arithmetic (ARITH), Jul. 1997, pp. 2–9.

[9] J.Clouser etal.,"A600-MHz superscalar floating-point processor," IEEE J. Solid-State Circuits, vol. 34, no. 7, pp. 1026–1029, Jul. 1999.

[10] S. Oberman, "Floating point division and square root algorithms and implementation in the AMD-K7 microprocessor," in Proc. 14th IEEE Symp. Comput. Arithmetic (ARITH), Apr. 1999, pp. 106–115.

[11] R. Senthinathan et al., "A 650-MHz, IA-32 microprocessor with enhanced data streaming for graphics and video," IEEE J. Solid-State Circuits, vol. 34, no. 11, pp. 1454–1465, Nov. 1999.

