Optimized Design of Efficient Finite Impulse Response Filter using Radix-4 Booth Multiplier

Sujit Kumar[1], Prof. Satyarth Tiwari[2]
M. Tech. Scholar, Department of Electronics and Communication, RKDF College of Engineering, Bhabha University, Bhopal[1]
Guide, Department of Electronics and Communication, RKDF College of Engineering, Bhabha University, Bhopal[2]

Abstract- The main objective of this research paper is to design architecture for finite impulse response (FIR) filter using radix-4 booth multiplier and common Boolean logic (CBL) adder. Finite Impulse response (FIR) filters are extensively utilized in digital signal processing in which different filter parts operate at different rates. It has applications in communication transmitters and receivers. FIR filters when implemented use multipliers and accumulators. There are various types of multiplier structure algorithms and their variations such as Combinational multiplier, Wallace Tree multiplier, Array multiplier and Sequential multiplier and Booth multiplier. Booth multipliers reduce the resulting number of partial products generated as a result of multiplication of two binary numbers. FIR filter has been implemented using with radix-4 booth recoding algorithm. The implement system is simulated Xilinx software and calculated parameters i.e. number of slice, look up table and delay.

Keywords –Common Boolean Logic Adder, Xilinx Software, Finite Impulse Response, Radix-4, Booth Multiplier

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 [1, 2].

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 [3, 4].

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 [5, 6].

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 [7].

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 [8, 9].

In this research paper, a novel architecture of FIR filters is based on radix-4 booth multiplier and common Boolean logic adder.

II. PROPSOED METHODOLOGY

An FIR filter is also called as recursive filter in which in addition to input values it also uses previous output values. These, similar to the past information esteems, are put away in the processor's memory. The word recursive actually signifies "running back", and alludes to the way that beforehand figured yield esteems backpedal into the count of the most recent yield. The articulation for a recursive channel in this way contains not just terms including the info esteems (x_n, x_{n-1}, x_{n-2} ...) yet in addition terms in y_n, y_{n-1}, y_{n-2}...

From this clarification, FIR channels require more counts to be performed, since there are past yield terms in the channel articulation and in addition input terms. To accomplish a given recurrence reaction trademark utilizing a recursive channel by and large requires a much lower arrange channel, and thusly less terms to be assessed by the processor, than the proportional non-recursive channel. The term computerized channel emerges on the grounds that these channels work on discrete-time signals [4].

The term finite impulse response arises because the filter output is computed as a weighted, finite term sum, of past, present, and perhaps future values of the filter input, i.e.,

$$y[n] = \sum_{k=-M_{-1}}^{M_2} b_k \, x[n-k] \qquad (1)$$

Where both M1 and M2 are finite

One of the simplest FIR filters that may be considered is a 3–term moving average filter of the form

$$y[n] = \frac{1}{3}(x[n+1] + x[n] + x[n-1]) \qquad (2)$$

An FIR filter is based on a feed-forward difference equation– Feed-forward means that there is no feedback of past or future outputs to form the present output, just input related terms [5].

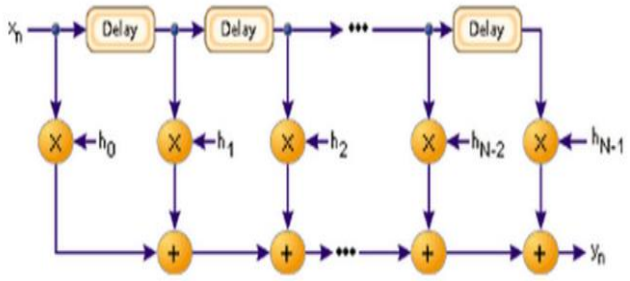

Figure 1: Logical Structure of FIR Filter

III. 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 2.

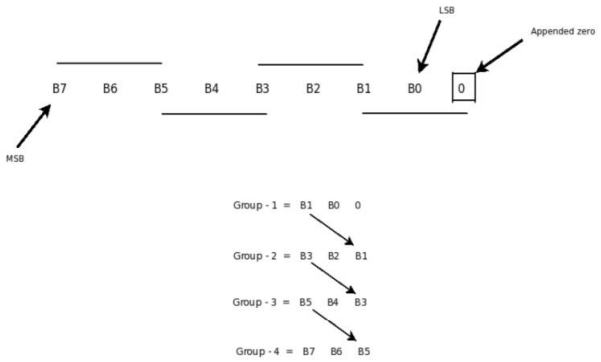

Figure 2: Grouping of multiplier bits in Radix-4 Booth algorithm

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 is 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

IV. 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.
Take a look at the VTS and RTL of FIR_4tap in fig. 3 & fig. 4. In fig. 5, there is a simulation result for and in fig. 6 there is a waveform.

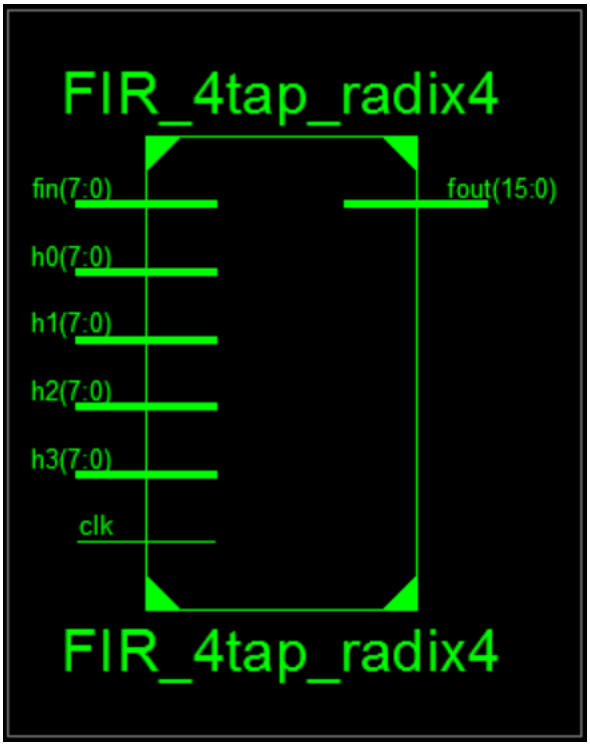

Figure 3: Examine VTS of FIR_4tap

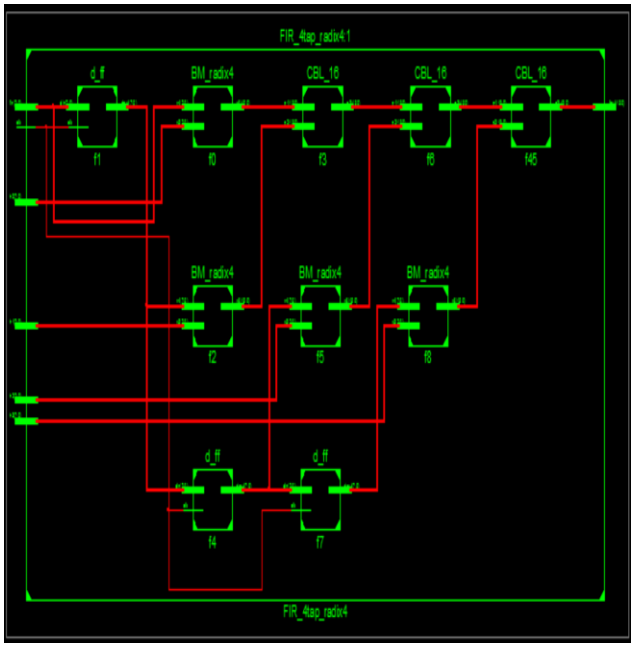

Figure 4: Examine RTL of FIR_4tap

```
Device utilization summary:
---------------------------

Selected Device : 6slx4tqg144-3


Slice Logic Utilization:
 Number of Slice Registers:              28  out of   4800     0%
 Number of Slice LUTs:                  607  out of   2400    25%
    Number used as Logic:               603  out of   2400    25%
    Number used as Memory:                4  out of   1200     0%
       Number used as SRL:                4

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:    622
   Number with an unused Flip Flop:     594  out of    622    95%
   Number with an unused LUT:            15  out of    622     2%
   Number of fully used LUT-FF pairs:    13  out of    622     2%
   Number of unique control sets:         2

IO Utilization:
 Number of IOs:                         57
 Number of bonded IOBs:                 57  out of    102    55%

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.662ns (Maximum Frequency: 601.811MHz)
   Minimum input arrival time before clock: 2.644ns
   Maximum output required time after clock: 22.522ns
   Maximum combinational path delay: 23.081ns
```

Figure 5: Examine Simulation of FIR_4tap

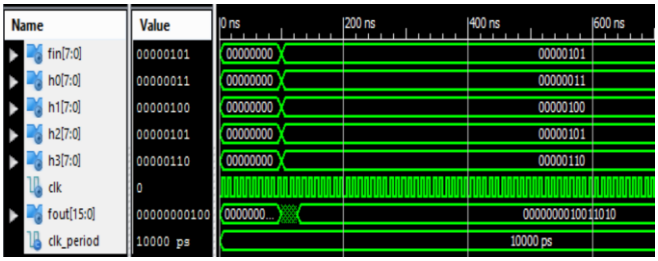

Figure 6: Examine Waveform of FIR_4tap

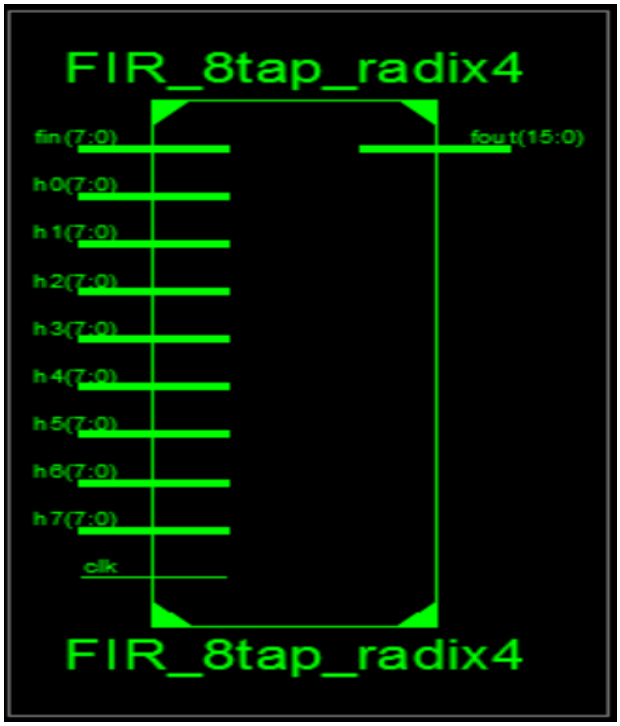

Figure 7: Examine VTS of FIR_8tap

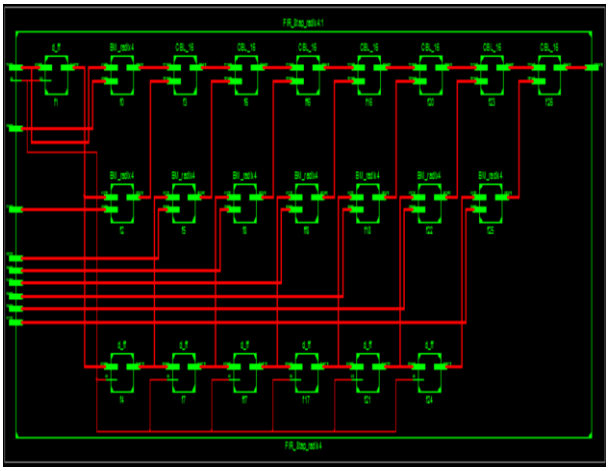

Figure 8: Examine RTL of FIR_8tap

```
Device utilization summary:
---------------------------

Selected Device : 6slx4tqg144-3


Slice Logic Utilization:
 Number of Slice Registers:              78  out of   4800     1%
 Number of Slice LUTs:                 1232  out of   2400    51%
    Number used as Logic:              1226  out of   2400    51%
    Number used as Memory:                6  out of   1200     0%
       Number used as SRL:                6

Slice Logic Distribution:
 Number of LUT Flip Flop pairs used:   1279
   Number with an unused Flip Flop:    1201  out of   1279    93%
   Number with an unused LUT:            47  out of   1279     3%
   Number of fully used LUT-FF pairs:    31  out of   1279     2%
   Number of unique control sets:         2

IO Utilization:
 Number of IOs:                          89
 Number of bonded IOBs:                  89  out of    102    87%

Timing Summary:
---------------
Speed Grade: -3

   Minimum period: 1.682ns (Maximum Frequency: 594.513MHz)
   Minimum input arrival time before clock: 2.644ns
   Maximum output required time after clock: 28.615ns
   Maximum combinational path delay: 29.515ns
```

Figure 9: Examine Simulation of FIR_8tap

V. CONCLUSION

In this paper design of CBL adder, radix-4 booth multiplier and FIR filter is presented. From implementation results it is observed that the FIR filter based on booth multiplier and common Boolean logic adder consumes less delay compare to previous design. The architecture design of 4-bit and 8-bit FIR filter is done.

REFRENCES

[1] S. China Venkateshwarlu, Mohammad, Chandra Shaker Pittala and Rajeev Ratna Vallabhuni , “Optimized Design of Power Efficient FIR Filter Using Modified Booth Multiplier”, 4th International Conference on Recent Trends in Computer Science and Technology, IEEE 2021.

[2] Ghasemi, Mir Majid, Amir Fathi, Morteza Mousazadeh, and Abdollah Khoei, “A new high speed and low power decoder/encoder for Radix-4 Booth multiplier,” International Journal of Circuit Theory and Applications, 2021.

[3] Chang, Yen-Jen, Yu-Cheng Cheng, Shao-Chi Liao, and Chun-Huo Hsiao, “A Low Power Radix-4 Booth Multiplier with Pre-Encoded Mechanism,” IEEE Access 8, 2020.

[4] Akshitha V. Ramesh et al. "Implementation and Design of FIR Filters using Verilog HDL and FPGA" Perspectives in Communication Embedded-systems and Signal-processing-PiCES vol. 4.5 pp. 85-88, 2020.

[5] Ranjeeta Yadav Rohit Tripathi and Sachin Yadav "FPGA Implementation of Efficient FIR Filter" International Journal of Engineering and Advanced Technology (IJEAT) vol. 9, no. 3 February 2020.

[6] Zhang, Tingting, Weiqiang Liu, Jie Han, and Fabrizio Lombardi, “Design and Analysis of Majority Logic Based Approximate Radix-4 Booth Encoders,” In IEEE/ACM International Symposium on Nanoscale Architectures (NANOARCH), pp. 1-6. IEEE, 2019.

[7] Gharabaghlo, Nader Sharifi, and Tohid Moradi Khaneshan, “Performance Analysis of High Speed Radix-4 Booth Encoders in CMOS Technology,” Majlesi Journal of Electrical Engineering 13, no. 3, 49-57, 2019.

[8] Madugula Sumalatha Panchala Venkata Naganjaneyulu and K. Satya Prasad "Low power and low area VLSI implementation of vedic design FIR filter for ECG signal de-noising" Microprocessors and Microsystems vol. 71 pp. 102883, 2019.

[9] Oguzhan COSKUN and A. V. C. I. Kemal "FPGA Schematic Implementations and Comparison of FIR Digital Filter Structures" Balkan Journal of Electrical and Computer Engineering vol. 6.1 pp. 20-28, 2018.

[10] 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.

[11] S. Rengaprakash et al. "FPGA implementation of fast running FIR filters" International Conference on Wireless Communications Signal Processing and Networking (WiSPNET). IEEE 2017.

[12] 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.

[13] 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.

[14] 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.

[15] R. Riedlinger etal.,“A32nm,3.1billion transistor, 12wide issue itanium processor for mission-critical servers,” IEEE J. Solid-State Circuits, vol. 47, no. 1, pp. 177–193, Jan. 2012.

[16] 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.