FPGA Implimentation of High Speed Digital Linear Phase Parallel FIR Filter

Manasa B
Dept.of Iinstrumentation and Technology
Dayananda Sagar College Of Engineering
Bangalore, India

Krishnamohan
Dept. of Iinstrumentation and Technology
Dayananda Sagar College Of Engineering
Bangalore, India

***Abstract*— Signal processing ranks among the most demanding applications of digital design concepts. As the growth of multimedia application is increasing day by day worldwide, the demand for high-performance and low power digital signal processing is getting higher and higher. The FIR digital filter is one among the fundamental devices which are most widely used in DSP field. In recent days filters with large lengths are in great use. So parallel processing is essential for efficient results. This project proposes new parallel FIR filter structure.**

This paper proposes new parallel FIR filter which makes use of fast FIR algorithm (FFAs). By using this hardware cost is reduced. Along with this usage of vedic multiplier is done which increases the speed of operation. Fundamental and the core of all the Digital Signal Processors (DSPs) are its multipliers and speed of the DSPs is mainly determined by the speed of its multipliers. Multiplication is the fundamental operation with intensive arithmetic computations. The important parameters associated with multiplication algorithms performed in DSP applications are latency and throughput. Latency is the real delay of computing a function. Throughput is a measure of the number of computations can be performed in a given period of time. The execution time of most DSP algorithms is dependent on its multipliers, so there is a need for high speed multiplier arises. So proposed concept of FIR filter based on FFA and vedic multiplier usage can lead to fast operation.

Key words—Digital signal processing (DSP), fast FIR algorithms (FFAs), parallel FIR, vedic multiplier, very large scale integration (VLSI).

I. INTRODUCTION

With the rapid growth of multimedia application, the demand for high-performance, low-power digital signal processing (DSP) is also growing. The FIR digital filter is one of the most widely used fundamental devices in DSP systems, ranging from wireless communications to video and image processing. In some applications FIR filters are operated at high frequencies like, video processing. For few other applications it need to operate at low frequencies such as multiple-input–multiple-output systems used in cellular wireless communication. For few more applications where narrow transition band characteristics are required, higher order in the FIR filter is need to be used.

FIR filters are digital filters. These hav finite impulse response. As they do not have the feedback, they are also known as non-recursive digital filters. But even we can use recursive algorithms for FIR filter realization. Various methods being used to design FIR filters and most of them are based on ideal filter approximation. Achieving ideal characteristics of filter is impossible, so the objective is to achieve good characteristics of FIR filters the order of the filter increases, the transfer function of FIR filter approaches the ideal. And hence it increases the complexity and the time needed for processing input samples of a signal which need to be filtered. Digital signal processing is the new and rapidly developing technology in the past thirty years, the FIR filter is one of the most important parts of DSP, it mainly consists of a finite number of sampling points, it always has stable system structure, and easy to implement linear phase in the symmetry conditions, because of these reasons it is used widely. With the rapid development of large-scale integrated circuits and the computer technology, the real-time, reliability and rapidity of modern digital filtering technique is required, so it is becoming an increasingly popular method to realize FIR filter in FPGA. There is a need to improve traditional FIR filter in area, speed and power parameters. parallel architecture results in improvement of speed and area utilization.

There have been some papers proposing ways to reduce the complexity of the parallel FIR filter in the past. In [1]–[4], polyphase decomposition is mainly manipulated, where the small-sized parallel FIR filter structures are derived first and then the larger block-sized ones can be constructed by cascading or by iterating small-sized parallel FIR filtering blocks. Fast FIR algorithms (FFAs) introduced in [1]–[3] show that they can implement an L-parallel filter using approximately $(2L - 1)$ subfilter blocks, each of which is of length N/L. It reduces the required number of multipliers to $(2N - N/L)$ from $L \times N$. Along with the parallel architecture, multipliers required in the operation are replaced by vedic multiplier.

Vedic mathematics is an extract from four Vedas (books of wisdom). Owing to its simplicity and regularity, it finds its usage and applications in various fields such as, geometry, trigonometry, quadratic equations, factorization and calculus. The power of Vedic mathematics is not only confined to its regularity, simplicity, but also it is logical. These are the key features which made Vedic mathematics, become so popular and thus it has become one of the leading topics of research not only in India but abroad as well. A high speed multiplier design (ASIC) using Vedic mathematics was presented in [7]. A time, area, power efficient multiplier architecture using Vedic mathematics was shown in [8]. A fast, low power multiplier architecture based on Vedic mathematics was shown in [9]. The awe striking and the power full feature of Vedic mathematics lies in the fact that it simplifies the complicated looking calculations in conventional mathematics to a simple one in a much faster and efficient manner. This is attributed to the fact that the Vedic formulae are claimed to be based on the "natural principles on which the human mind works". As multipliers are the key components in the FIR filter design, the conventional multipliers are replaced by the vedic multipliers which in turn helps to speed up the parallel FIR filter operation.

A brief introduction of FFAs is reviewed in Section II. In Section III, the proposed parallel FIR filter architectures are presented, and vedic multipliers are discussed. Finally, the conclusion is given in Section IV.

II. FFA

N tap filter can be expressed in general form as:

$$y(n) = \sum_{i=0}^{N} h_i\, x(n-i)$$
$$= h_0 x(n) + h_1 x(n-1) + \cdots + h_N x$$

where:

x(n) is the input signal. y(n) is the output signal. are the filter coefficients, known as tap weights. These make up the impulse response. N is the filter order.

The traditional existing FIRs are based on serial multiplication. Summing of products for the given finite terms leads to arithmetic complexity because as the filter order increases, complexity increases. Many algorithms are known to reduce the arithmetic complexity of FIR filtering. As there is more numbers multipliers in the existing traditional FIR filters, leads to more delay. There is a need to reduce the number of multipliers. As multipliers are slower than adders and weigh more in case of silicon area, by reducing the number multipliers in terms of adders, we can design better filter. Using symmetrical properties we design the parallel structure. Also multipliers are converted to adders and subtractors by using FFA. Further by using vedic multipliers instead of conventional ones the delay can be reduced in better way.

The traditional *L*-parallel FIR filter can be derived using polyphase decomposition as [3]

$$\sum_{p=0}^{L-1} Y_p(Z^L)Z^{-p} = \sum_{q=0}^{L-1} X_q(Z^L)Z^{-q} \sum_{r=0}^{L-1} H_r(Z^L) \ldots (2)$$

Where p, q, & r =0,1,2,3…,L-1

A. *For 2 × 2 FFA (L = 2)*

According to (2), a two-parallel FIR filter can be expressed as [1], [3]

$$Y_0 = H_0X_0 + Z^{-2}H_1X_1$$

$$Y_1 = H_0X_1 + Z^{-2}H_1X_0 \ldots\ldots\ldots(3)$$

However, (3) can be written as

$$Y_0 = H_0X_0 + Z^{-2}H_1X_1$$

$$Y_1 = (H_0 + H_1)(X_0 + X_1) - H_0H_0 - H_1H_1 \ldots\ldots\ldots(4)$$

The two-parallel (*L* = 2) FIR filter implementation using the FFA obtained from (4) is shown in Figure 1.

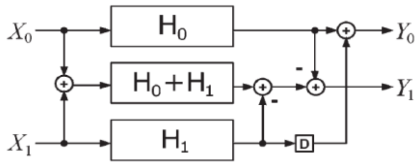

fig.1 Two-parallel FIR filter implementation using the FFA

B. *For 3 × 3 FFA (L = 3)*

By the similar approach, a three-parallel FIR filter using the FFA can be expressed as [1], [3]

$$Y_0 = H_0X_0 - Z^{-3}H_2X_2 + Z^{-3}[(H_1 + H_2)(X_1 + X_2) - H_1X_1$$

$$Y_1 = [(H_0 + H_1)(X_1 + X_2) - H_1X_1] - (H_0X_0 - Z^{-3}H_2X_2)$$

$$Y_2 = [(H_0 + H_1 + H_2)(X_0 + X_1 + X_2)]$$
$$-[(H_0 + H_1)(X_0 + X_1) - H_1X_1] - [(H_1 + H_2)(X_1 + X_2) - H_1X_1]$$

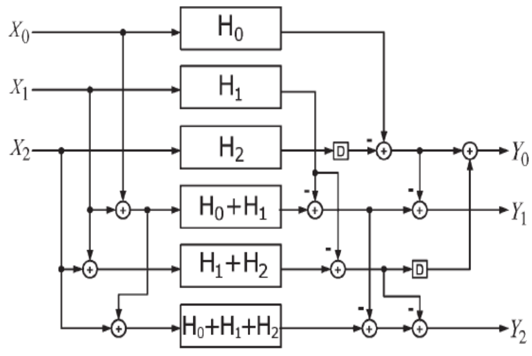

fig.2 Three-parallel FIR filter implementation using the FFA.

III. PROPOSED 3*3 PARALLEL FIR

The existing three-parallel FFA structure naturally has benefits in terms of speed and area. However, the existing three-parallel FFA structure is advantageous in terms of speed. In this section, new three-parallel FIR filter structures are proposed, which makes use of vedic multiplier.

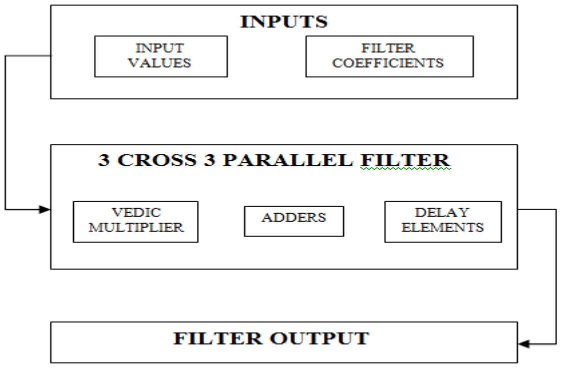

fig.3 project block diagram

3*3 parallel FIR filter as shown in figure2. For better operation and to increase the speed of the operation, vedic multiplier is used. Conventional multipliers used to design the 3*3 parallel filter are replaced by the vedic multipliers. Multiplication methods are extensively discussed in Vedic mathematics. There are several tricks and short cuts that are suggested by VM to optimize the process. A time, area, power efficient multiplier architecture using Vedic mathematics was shown in [8]. In this a comparative study of the array multiplier, Booth multiplier Wallace tree multiplier, Carry save multiplier and Vedic multiplier was done in detail. The study clearly showed that though array and booth multipliers are faster among the conventional multipliers, they are so because of some trade-off with complexity and high power consumption respectively.

Vedic multiplication is applicable to all cases of multiplication and also in the division of a large number by another large number. Vedic multiplication can be done with the technique “vertically and crosswise”. Below we discuss multiplication of two, 4 digit numbers with this method [10-11]. Ex.1. the product of 1111 and 1111 using vertically and crosswise method is given below. Methodology of Parallel Calculation

1 1 1 1
1 1 1 1
1×1=1

1 1 1 1
1 1 1 1
1×1+1×1=2

1 1 1 1
1 1 1 1
1×1+1×1+1×1=3

1 1 1 1
1 1 1 1
1×1+1×1+1×1+1×1=4

1 1 1 1
1 1 1 1
1×1+1×1+1×1=3

1 1 1 1
1 1 1 1
1×1+1×1=2

1 1 1 1
1 1 1 1
1×1=1

Final answer=1234321

In the below section ,The hardware architecture of 2X2, 4x4 and 8x8 bit Vedic multiplier module are displayed. For the multiplication of two binary numbers vedic mathematics is used here and proposed architectures are discussed. The advantage of Vedic multiplier is that here partial product generation and additions are done concurrently. So we can say it is well adapted to parallel processing. This feature makes it more attractive and useful for binary multiplications. This in turn reduces delay, which is the primary motivation behind this work.

A. *Vedic Multiplier for 2x2 bit Module*

The method is explained below for two, 2 bit numbers *A* and *B* where $A = a1a0$ and $B = b1b0$ as shown in Fig. 4. Firstly, the least significant bits are multiplied which gives the least significant bit of the final product (vertical). Then, the LSB of the multiplicand is multiplied with the next higher bit of the multiplier and added with, the product of LSB of multiplier and next higher bit of the multiplicand (crosswise). The sum gives second bit of the final product and the carry is added with the partial product obtained by multiplying the most significant bits to give the sum and carry. The sum is the third corresponding bit and carry becomes the fourth bit Of the finel product. The 2X2 Vedic multiplier module is implemented using four input AND gates & two half-adders which is displayed in its block diagram in Fig. 4. It is found that the hardware architecture of 2x2 bit Vedic multiplier is same as the hardware architecture of 2x2 bit conventional Array Multiplier [12]. Hence it is concluded that multiplication of 2 bit binary numbers by Vedic method does not made significant effect in improvement of the multiplier‟s efficiency. Very precisely we can state that the total delay is only 2-half adder delays, after final bit products are generated, which is very similar to Array multiplier. So we switch over to the implementation of 4x4 bit Vedic multiplier which uses the 2x2 bit multiplier as a basic building block. The same method can be extended for input bits 4 & 8. But for higher no. of bits in input, little modification is required.

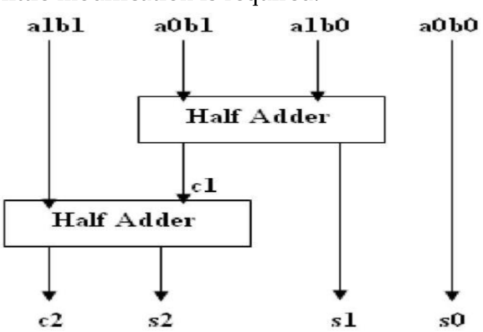

fig.4 2 bit vedic multiplier

B. *Vedic Multiplier for 4x4 bit Module*

The 4x4 bit Vedic multiplier module is implemented using four 2x2 bit Vedic multiplier modules. 2×2 multiplier is as discussed in Fig. 4. Consider a 4x4 multiplications, let A= A3 A2 A1 A0 and B= B3 B2 B1 B0. The output line for the multiplication result is – S7 S6 S5 S4 S3 S2 S1 S0. Now divide A and B into two parts, say A3A2 & A1 A0 for A and B3 B2 & B1B0 for B. By making use of fundamental of Vedic multiplication, taking two bit at a time and using 2 bit multiplier block, we can have the following structure for multiplication as shown in figure5. To get final product (S7 S6 S5 S4 S3 S2 S1 S0), four 2x2 bit Vedic multiplier (Fig. 4) and three 4-bit Ripple-Carry (RC) Adders are required. This proposed Vedic multiplier can be used to reduce delay. This proposed new architecture is efficient in terms of speed. The arrangements of RC Adders shown in Fig. 5, helps us to reduce delay. In the same way 8x8 Vedic multiplier modules are implemented by using four 4x4 multiplier modules.

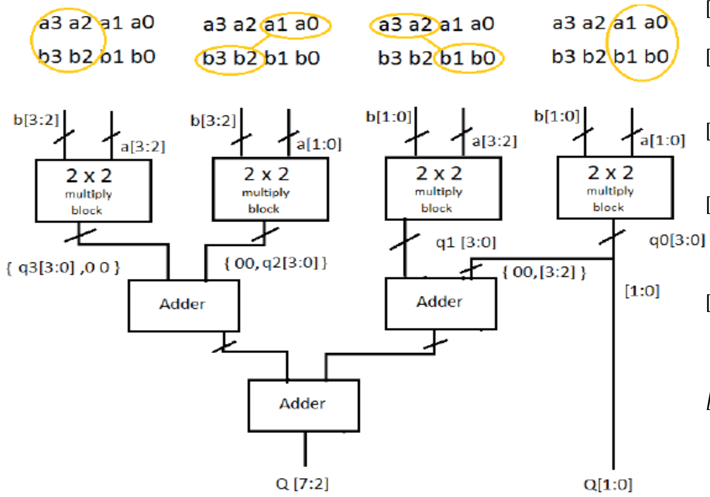

fig.5 4bit vedic multiplier

C. *Vedic Multiplier for 8x8 bit Module*

The 8x8 bit Vedic multiplier can be easily implemented by using four 4x4 bit Vedic multipliers. Consider an 8x8 multiplications, say A= A7 A6 A5 A4 A3 A2 A1 A0 and B= B7 B6 B5B4 B3 B2 B1B0. The output line for the multiplication result will be of 16 bits as – S15 S14 S13 S12 S11 S10 S9 S8 S7 S6bS5S4 S3 S2 S1 S0. Now divide A and B into two parts, say the 8 bit multiplicand A can be decomposed into pair of 4 bits AH-AL. Similarly multiplicand B can be decomposed into BH-BL. The 16 bit product can be obtained by using the fundamental of Vedic multiplication, taking four bits at a time and using 4 bit multiplier block as discussed. Thus we can perform the multiplication. The outputs of 4x4 bit multipliers are added accordingly to obtain the final product. Here total three Ripple-Carry Adders are required. This 8×8 multiplier is used in the parallel FIR filter. So that speed of operation is increased.

IV. CONCLUSION

In this brief, we have presented new parallel FIR filter structures. In parallel FIR filter implementation multipliers are the major portions in hardware consumption. The proposed new structures exploit the vedic multipliers. It is a method for hierarchical multiplier design and it clearly indicates the computational advantages offered by Vedic methods. So by making use of parallel structure and vedic multipliers, compared to the traditional FIR filter the proposed parallel FIR filter works better in terms of execution time.

REFERENCES

[1] A. Parker and K. K. Parhi, "Low-area/power parallel FIR digital filter implementations," *J. VLSI Signal Process. Syst.*, vol. 17, no. 1, pp. 75–92, Sep. 1997.

[2] J. G. Chung and K. K. Parhi, "Frequency-spectrum-based low-area lowpower parallel FIR filter design," *EURASIP J. Appl. Signal Process.*, vol. 2002, no. 9, pp. 444–453, Jan. 2002.

[3] K. K. Parhi, *VLSI Digital Signal Processing systems: Design and Implementation*. New York: Wiley, 1999.

[4] Z.-J. Mou and P. Duhamel, "Short-length FIR filters and their use in fast nonrecursive filtering," *IEEE Trans. Signal Process.*, vol. 39, no. 6, pp. 1322–1332, Jun. 1991.

[5] Cheng and K. K. Parhi, "Low-cost parallel FIR structures with 2-stage parallelism," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 280–290, Feb. 2007.

[6] Y.-C. Tsao and K. Choi, "Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 20, no. 2, pp. 366–371, Feb. 2010.

[7] Prabir Saha, Arindham Banerjee, Partha Bhattacharyya, Anup Dhandapat, "High speed design of complex multiplier using Vedic mathematics", *Proceedings of the 2011 IEEE students technology symposium, IIT Kharagpur, pp. 237-241, Jan.* 2011.

[8] Himanshu Thapliyal and Hamid R. Arabnia, "A Time-Area- Power Efficient Multiplier and Square Architecture Based On Ancient Indian Vedic Mathematics", *Department of Computer Science, The University of Georgia, 415 Graduate Studies Research Center Athens, Georgia 30602-7404, U.S.A.*

[9] Abu-Shama, M. B. Maaz, M. A. Bayoumi, "A Fast and Low Power Multiplier Architecture", *The Center for Advanced Computer Studies, The University of Southwestern Louisiana Lafayette, LA 70504.*

[10] Parth Mehta and Dhanashri Gawali, "Conventional versus Vedic mathematics method for Hardware implementation of a multiplier", International conference on Advances in Computing, Control, and Telecommunication Technologies, pp. 640-642, 2009.

[11] Ramalatha, M.Dayalan, K D Dharani, P Priya, and S Deoborah, "High Speed Energy Efficient ALU Design using Vedic Multiplication Techniques", International Conference on Advances In Computationa Tools for Engineering Applications (ACTEA) IEEE, pp. 600-603, July15-17, 2009.

[12] M. Morris Mano, "Computer System Architecture", 3rd edition, Prientice-Hall, New Jersey, USA, 1993, pp. 346-348.