World Engineering & Applied Sciences Journal 8 (4): 187-191, 2017
ISSN 2079-2204

DOI: 10.5829/idosi.weasj.2017.187.191

Efficient Multiplier Based Architecture for Transpose Form Block FIR Filter

[1]S. Banupriya, [2]V. Ravichandran and [3]E. Vidhya

[1]Roever Engineering College, ECE, Tamilnadu, India
[2]Professor, Departmant of Electronics and Communication Engineering,
Roever Engineering College, Anna University, Tamilnadu, India
[3]Assistant Professor, Department of Electronics and Communication Engineering,
Roever Engineering College, Anna University, Tamilnadu, India

Abstract: This paper present the possibility of realization of block FIR(Finite impulse response filter) in transpose form configuration for area-delay efficient realization of large order FIR filters for both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form FIR filter. A low-complexity design using the multiple constant multiplications (MCM) scheme is also presented for the block implementation of fixed FIR filters. The proposed Vedic based structure involves significantly less area delay product (ADP) than the existing Array based block implementation of direct-form structure and it is also suitable for medium or large filter lengths, The parameters are area and delay are reduced using vedic multiplier based architecture. The implemented transpose form block FIR filter structure achieved less area and delay than existing technique (Direct form structure with array multiplier).The MODEL SIM and Xilinx software tool is used for simulation.

Key words: Finite impulse response (FIR) filter · Block processing · Reconfigurable FIR filter(RFIR) · Area delay product(ADP)

INTRODUCTION

In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying). The impulse response of an Nth-order discrete-time FIR filter (i.e., with a Kronecker delta impulse input) lasts for N + 1 samples and then settles to zero. FIR filters can be discrete-time or continuous-time and digital or analog.

Digital filters that have an impulse response which reaches zero in a finite number of steps are (appropriately enough) called Finite Impulse Response (FIR) filters. An FIR filter can be implemented non-recursively by convolving its impulse response (which is often used to define an FIR filter) with the time data sequence it is filtering. FIR filters are somewhat simpler than Infinite Impulse Response (IIR) filters, which contain one or more feedback terms and must be implemented with difference equations or some other recursive technique.

They can easily be designed to be "linear phase" (and usually are). Put simply, linear-phase filters delay the input signal, but don't distort its phase. They are simple to implement. On most DSP microprocessors, the FIR calculation can be done by looping a single instruction. They are suited to multi-rate applications. By multi-rate, there are two methods either "decimation" (reducing the sampling rate),and "interpolation" (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will discard (so the feedback will be incorporated into the filter)

They have desirable numeric properties. In practice, all DSP filters must be implemented using "finite-precision" arithmetic, that is, a limited number of bits. The

Corresponding Author: S. Banupriya, Roever Engineering College, ECE, Tamilnadu, India.

use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters have no feedback, so they can usually be implemented using fewer bits and the designer has fewer practical problems to solve related to non-ideal arithmetic. They can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.) This is an important consideration when using fixed-point DSP's, because it makes the implementation much simpler.

Literature Survey: The existing multiplier-based structures use either direct form configuration or transpose form configuration. But, the multiplier-less structures of use transpose form configuration, whereas the Distributed Arithmetic (DA)-based structure of uses direct-form configuration and do not find any specific block-based design for Reconfigurable FIR (RFIR) filter in the literature. But, the block structure obtained is not efficient for large filter lengths and variable filter coefficients, such as SDR channelizer. Two methods are mainly used in the literature. One of the method is Block-processing method. This is popularly used to derive high-throughput hardware structures. It not only provides throughput-scalable design but also improves the area-delay efficiency. The derivation of block-based FIR structure is straightforward when direct-form configuration is used [1], whereas the transpose form configuration does not directly support block processing. But, to take the computational advantage of the MCM, FIR filter is required to be realized by transpose form configuration. Apart from that, transpose form structures are inherently pipelined and supposed to offer higher operating frequency to support higher sampling rate.

MCM method is another one method. This method is used to reduces the number of additions required for the realization of multiplications by common sub expression sharing, when a given input is multiplied with a set of constants. The MCM scheme is more effective, when a common operand is multiplied with more number of constants. Therefore, the MCM scheme is suitable for the implementation of large order FIR filters with fixed coefficients. But, MCM blocks can be formed only in the transpose form configuration of FIR filters.

Various researchers have been suggested several designs for efficient realization of Finite impulse response filter and using Distributed Arithmetic (DA) [2]. Computation sharing multiplication Technique is used to reduce the power consumption and improve the performance of FIR filter[3].Distributed arithmetic method is used for Computation of circular convolution [4]. One dimensional and two dimensional systolic structure used for area delay efficient realization of FIR Filter [5]. Distributed arithmetic based computation is used for efficient memory based implementation of FIR filter where the filter outputs are computed as inner product of input sample vectors and filter coefficient vector [6].

Finite impulse response filter used in several DSP applications such as loud speaker equalization, echo cancellation, adaptive noise cancellation, speech processing and various communication applications including software defined radio(SDR) [7]. Many of these applications require FIR filters of large order to meet the high frequency specifications [8-10]. Very often these filters need to support high sampling rate for high speed digital communication[11].There are some application such as SDR channelizer, where FIR filters need to be implemented in a Reconfigurable hardware to support multi standard wireless communication [12]. Several designs have been suggested for efficient realization of RFIR filter using general multiplier and constant multiplication [13-15]. Digit based reconfigurable FIR filter architecture has been proposed for reduce the power and Canonic sign digit(CSD) based RFIR filter has been proposed where the nonzero CSD values are modified to to reduce the precision of filter coefficients without significant impact on filter behavior. But the reconfiguration overhead is significantly large and does not provide an area-delay efficient structure [13]. Two new reconfigurable architecture has been proposed for low complexity FIR filters [14]. Distributed arithmetic based computation approaches for high throughput Reconfigurable implementation of FIR filter whose filter coefficient change during run time [15].

The existing multiplier-based structure uses either direct form or transpose form configuration. But the transpose form configuration use less multiplier, whereas the DA-based structure of [15] uses direct form configuration. But, we do not find any specific block based design for RFIR filter. Least mean square (LMS) algorithm proposed for reducing the Area Delay Product (ADP) [17], but block structure obtained is not efficient for large order filter length and variable coefficient such us SDR channelizer. Therefore, the design methods in [16] and [17] are more suitable for 2-D FIR filters and least mean square adaptive filter.

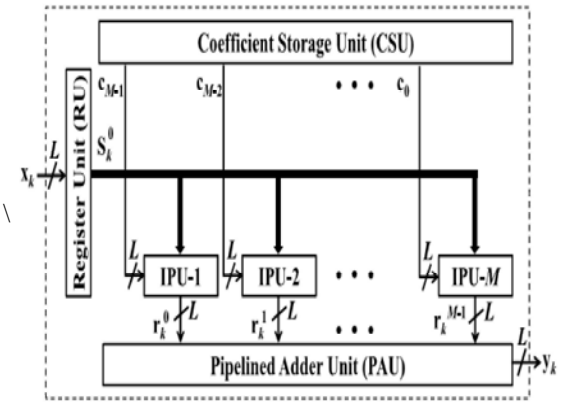


Fig. 1: Proposed structure of FIR filter

Proposed Architecture: There are several applications where the coefficients of FIR filters remain fixed, while in some other applications, like SDR channelizer that requires separate FIR filters of different specifications to extract one of the desired narrow- band channels from the wideband RF front end. These FIR filters need to be implemented in a RFIR structure to support multi standard wireless communication. In this section, present a structure of block FIR filter for such reconfigurable applications.

Proposed Structure for Transpose Form Block FIR Filter for Reconfigurable Applications: The proposed structure for block FIR filter is [based on the recurrence relation of (12)] shown in Fig. 1 for the block size $L = 8$. It consists of one coefficient selection unit (CSU), one register unit (RU), M number of inner- product units (IPUs) and one pipeline adder unit (PAU).

Register Unit (RU): The RU receives $\mathbf{x}_k$ during the k^{th} cycle and produces L rows of S^0_K in parallel. It is designed by number of Flip Flop based on Input bit length.

Co-efficient Storage Unit (CSU): The *CSU* stores coefficients of all the filters to be used for the reconfigurable application. It is implemented using N ROM LUTs, such that filter coefficients of any particular channel filter are obtained in one clock cycle, where N is the filter length.

The output of an FIR filter length N can be computed using the relation

$$Y(n)=\sum_{i=0}^{N-1} h(i).\ x(n-i).$$

where:

y(n) is the output of an FIR filter,
h(i) is the impulse response of the filter,
N is the filter order,
x(n-i) in these terms are commonly referred to as taps.

Inner Product Unit (IPU): L rows of S^0_k are transmitted to Multiple inner product units (IPU) of the proposed structure. The Multiple Inner product units also receive Multiple short-weight vectors from the Coefficient storage unit (CSU) such that during the k^{th} cycle, the $(m + 1)^{th}$. Inner product unit receives the weight vector $\mathbf{C}_{M-m-1}$ from the Coefficient storage unit and L rows of $\mathbf{S}^0_k$ form the Register unit(RU). Each IPU performs matrix-vector product of S^0_k with the short-weight vector **c**m and computes a block of L partial filter outputs ($\mathbf{r}_k^m$).

Inner Cell Unit (ICU): The $(l+1)^{th}$ IPC receives the $(l+1)^{th}$ row of $\mathbf{S}_0^k$ and the coefficient vector **c**m and computes a partial result of inner product r (kL -l), for $0 \le l \le L - 1$. Internal structure of (l + 1)th IPC for L = 8.

Pipelined Adder Unit (PAU): The Pipelined adder unit (PAU) involves L(M-1) adders and the same number of registers, where each register has a width of (B+ B'), B and B' respectively, being the bit width of input sample and filter coefficients. The PAU is Designed by number of Adder and Number of Register.

Flip Flop and Registers: Flip-flop is a 1 bit memory cell which can be used for storing the digital data. To increase the storage capacity in terms of number of bits, we have to use a group of flip-flop. Such a group of flip-flop is known as a Register. The n-bit register will consist of n number of flip-flop and it is capable of storing an n-bit word.

Vedic Multiplier: It gives explanation of several mathematical terms including arithmetic, geometry (plane, co-ordinate), trigonometry, quadratic equations, factorization and even calculus for various applications. In this paper 8-bit multiplier is designed using four 4x4 Vedic multipliers which employ Urdhva Tiryagbhyam sutra and Ripple carry adder(RCA) technique for partial product addition. The output of these Vedic multipliers is added by modifying the logic levels of ripple carry adder.

Block diagram of the proposed 8x8 multiplier is illustrated in Fig 2. The 8-bit input sequence is divided into two 4-bit numbers and given as inputs to the 4-bit multiplier blocks (a[7:4] & b[7:4], a[3:0] & b[7:4], a[7:4] &

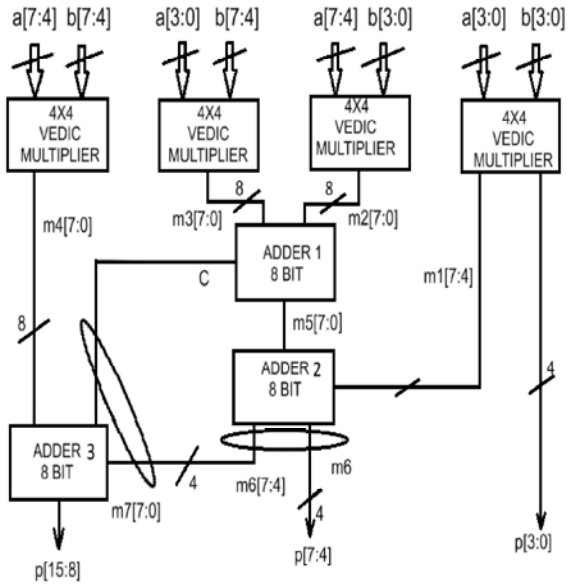
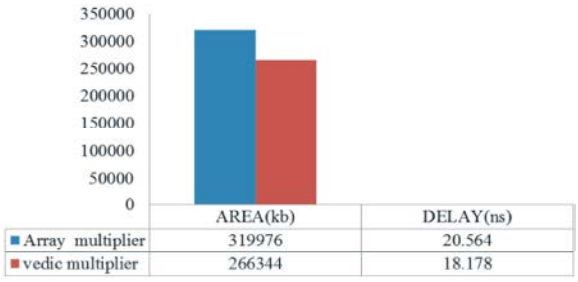


Fig. 2: Internal structure of vedic multiplier



	AREA(kb)	DELAY(ns)
Array multiplier	319976	20.564
vedic multiplier	266344	18.178

Fig. 3: Graphical representation of Array multiplier and Vedic multiplier

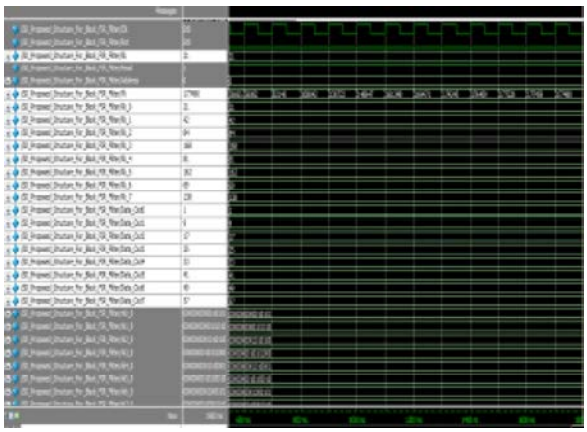


Fig. 4: FIR Filter Output Waveform

Table 1: Comparison between array based and vedic multiplier

S.No	Parameters	Array Multiplier (Existing method)	Vedic Multiplier (Proposed method)
1	Area	319976(kb)	266344(kb)
2	Delay	20.564(ns)	18.178(ns)
3	Gate count	43196	41538

b[3:0], a[3:0] & b[3:0]). The four multipliers used are similar and give 8-bit intermediate products which are added using overlapping logic with the help of three modified parallel adders (ADDER-1, ADDER-2 and ADDER-3).The partial products obtained from the four multipliers are demarcated into four regions as in Fig 2.

The four LSB product bits P[3:0] are directly obtained from one of the multipliers. The output of the second and third multiplier block is added directly using ADDER-1 as the second and third region is overlapping. Then the higher order bit of first multiplier block is added to the overlapping sum using ADDER-2 which gives the product P[7:4]. Finally, MSB bits P[15:8] are ?obtained by adding the fourth multiplier output to the carry from ADDER-1 (added at the fifth bit position) and higher order bits (acts as lower nibble of addend) of ADDER-3.

RESULT AND DISCUSSION

A generalized block formulation is presented for transpose form FIR filter. MCM scheme is also used for the block implementation of fixed FIR filters. The proposed Vedic based multiplier technique using in this architecture. From the synthesis report the gate count is 41538, area is 266344 Kb and the delay is 18.178 ns. The Table 1 shows the comparison between array based vedic based multiplier. The proposed vedic based multiplier transpose form FIR filter structure achieved less area and delay than the array based multiplier FIR filter structure. The Fig 3 shows the graphical representation of array based and vedic multiplier. Fig 4 shows the synthesis result of vedic multiplier based architecture.

CONCLUSION

This Paper has focused on dealing with array based multiplier and vedic multiplier using transpose form block FIR filter. Vedic based multiplier architecture mostly used for large order filter length of both fixed and reconfigurable coefficients. MCM scheme is also involved for the block implementation of fixed FIR filters. The proposed Vedic based structure reduced less area delay product (ADP) than the existing Array based block implementation of direct-form structure.. In future, the delay and area will be further reduced in transpose form FIR filter by applying another technique. This Proposed System Implemented using Verilog HDL and Simulated by Modelsim 6.4 c and Synthesized by Xilinx tool. The proposed system implemented in FPGA Spartan 3 XC3S 200 TQ-144.

REFERENCES

1. Mohanty, B.K., P.K. Meher, A. Al-Maadeed and A. Amira, 2014. Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters, IEEE Trans. Circuits Syst. I, Reg. Papers, 61(1): 120-33.
2. White, S.A., 1989. Applications of distributed arithmetic todigital signal processing: A tutorial review, IEEE ASSP Mag., 6(3): 4-19.
3. Park. J., *et al*, 2004. Computation sharing programmable FIR filter for low-power and high-performance applications, IEEE J. Solid State Circuits, 39(2): 348-357.
4. Meher, P.K., 2006. Hardware-efficient systolization of DA-based calculation of finite digital convolution, IEEE Trans.Circuits Syst. II, Exp. Briefs, 53(8): 707-711.
5. Meher, P.K., S. Chandrasekaran and A. Amira, 2008. FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic, IEEE Trans. Signal Process., 56(7): 3009-3017.
6. Meher, P.K., 2010. New approach to look-up-table design and memory based realization of FIR digital filter, IEEE Trans.Circuits Syst. I, Reg. Papers, 57(3): 592-603.
7. Proakis, J. and D.G. Manolakis, 1996. Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA: Prentice-Hall.
8. Hentschel, T. and G. Fettweis, 1999. Software radio receivers," in CDMA Techniques for Third Generation Mobile Systems. Dordrecht, The Netherlands: Kluwer, pp: 257-283.
9. Mirchandani, E., R.L. Zinser and J.B. Evans, 1995. A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals], IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., 39(10): 681-694.
10. Xing, D. and J. Chiu, 1993. Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system, in Proc. IEEE South eastcon, Apr. pp: 1-6.
11. Mitola, J., 2000. Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York, NY, USA: Wiley.
12. Vinod, A.P. and E.M. Lai, 2006. Low power and high-speed implementation of FIR filters for software defined radio receivers, IEEE Trans. Wireless Commun., 7(5): 1669-1675.
13. Chen, K.H. and T.D. Chiueh, 2006. A low-power digit-based reconfigurable FIR filter, IEEE Trans. Circuits Syst. II, Exp. Briefs, 53(8): 617-621.
14. Mahesh, R. and A.P. Vinod, 2010. New reconfigurable architectures for implementing FIR filters with low complexity, IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., 29(2): 275-288.
15. Park, S.Y. and P.K. Meher, 2014. Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digita filter, IEEE Trans. Circuits Syst. II, Exp. Briefs, 61(7): 511-515l.
16. Mohanty, B.K., P.K. Meher, A. Al-Maadeed and A. Amira, 2014. Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters, IEEE Trans. Circuits Syst. I, Reg. Papers, 61(1): 120-33.
17. Mohanty, B.K. and P.K. Meher, 2013. A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm, IEEE Trans. Signal Process., 61(4): 921-932.
18. Parhi, K.K., 1999. VLSI Digital Signal Processing Systems:Design and Implementation. New York, NY, USA: Wiley.