

An Efficient Realization of FIR Filter Architecture for Higher Lengths

Sheena Pathan¹, Rahamtula Shaik²

¹PG Scholar, Dept. of ECE, QIS Institute of Technology, Ongole, AP.

²Assistant Professor, Dept. of ECE, QIS Institute of Technology, Ongole, AP.

Abstract: A Finite Impulse Response (FIR) Filter whose impulse response is for a finite duration can be implemented in two forms: Direct form and Transpose Form. The direct form realization supports block processing whereas the transpose form doesn't. The transpose form structure is more efficient when we it comes to the realization of medium or higher order filters whereas direct form suits best for realizing only short length filters. The transpose form realization of FIR filters supports MCM (Multiple Constant Multiplications) scheme and this will lead to ease in computational analysis. In this paper, we analyze the block formulation of FIR filter in transpose form by using MCM based approach. It also includes detailed research on the impact of the MCM block in the structure. The proposed realization of FIR filter architecture results in reduction on delay and also facilitates area savings. The modified structure also speeds up the output generation process being less complex and enhances in saving the number of computations.

Keywords: Block processing, finite-impulse response (FIR) filter, MCM scheme, VLSI

1. Introduction

Digital signal processing is an important part of electronic devices whereas a signal is an integral part of DSP. A signal contains the information which we want to transfer from source to destination through a medium which is called channel. When any signal containing information travels from source to destination there will be some loss of information due to the presence of noise in the channel. This information should be retained in its original form by the signal until it reaches to the destination so that there is a complete information transfer without any loss in signal. A signal can have high frequency range, low frequency range and mid frequency range. Sometime useful information is contained by using only high frequencies or low frequencies or mid frequencies range. This useful information is extracted by filters. These filters are different for analog and digital signals. For DSP applications we use digital filters.

FIR filter is a type of digital filter which is used for linear characteristic applications. A Finite Impulse Response 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. FIR filter finds its applications in various fields including DSP, communication applications and so on [1]. Main requirement of these applications is that the FIR filter should be of large order so that it could suit all the frequency specifications [2]-[4]. For high speed digital communication purposes these filters also support high sampling rates [5].

Various types of techniques have been proposed for the designing of FIR filter. Designing of FIR filter by using MAC unit is easy as compared to window techniques. MAC unit is multiplier accumulation unit. It comprises of multiplier and adder. To make FIR filter faster a multiplier and an adder, that are to be selected for MAC unit should be faster. A reconfigurable booth multiplier can be used for this purpose since it is a high speed multiplier. Carry Look Ahead adder can be used for the purpose of final addition. This combination can make the device faster.

Finite Impulse Response(FIR) filter is one of the most important building blocks in many digital signal processing (DSP) circuits and systems. In high performance applications, FIR filters can be implemented using dedicated hardware such as application specific integrated circuit (ASIC). In these applications, the transposed direct form filter structure is preferred over the direct form filter structure due to its inherent pipelined accumulation section and these transpose form structures offer higher operating frequencies to support higher sampling rates.

Generally, an FIR filter can be implemented in direct form or transposed direct form (TDF). For very large scale integration (VLSI) implementation, the TDF is preferred over the direct form due to its inherent pipelined accumulation section. A TDF FIR filter consists of two parts which includes a multiple constant multiplication (MCM) block and a product accumulation block.

The products generated by the MCM block are delayed and accumulated in the product accumulation block to produce the filter output. The adders in the product accumulation block are referred to as Shift Adders (SAs) in the rest of the paper. To reduce the complexity of FIR filters, a lot of efforts have been put on the efficient implementation of MCM blocks and a lot of design techniques have been proposed. The SAs, on the other hand, are often ignored by most of the research, but the precision of the filter output may be sacrificed.

The MCM method [7],[11]-[13] reduces the complexity of FIR filter design by decreasing the number of additions required to realize the multiplications by using a common subexpression. This MCM scheme is efficient for implementing large order FIR filters with fixed coefficients since it is more appropriate when a common input is multiplied by more number of constants.

2. Mathematical Approach of the Transpose Form Block Fir Filter

For a FIR filter of length N , the output can be computed by using the relation:

$$y(n) = \sum_{i=0}^{N-1} h(i) \cdot x(n-i) \quad (1)$$

The above computation can also be expressed by the recurrence relation as follows:

$$Y(z) = \left[z^{-1} \left(\dots \left(z^{-1} \left(z^{-1} h(N-1) + h(N-2) \right) + h(N-3) \right) \dots + h(1) \right) + h(0) \right] X(z) \quad (2)$$

Let us consider that, the block FIR filter can process a block of L new input samples to generate a block of L output samples. Now, we can compute the filter output of the k th block as y_k . The expression for y_k can be obtained as:

$$y_k = X_k \cdot h \quad (3)$$

Where h is the weight vector and X_k is the input matrix. These can be defined as follows:

$$h = [h(0), h(1), \dots, h(N-1)]^T$$

$$X_k = [x_k^0 x_k^1 \dots x_k^L \dots x_k^{N-1}] \quad (4)$$

In the above expression, x_k^i is the $(i+1)$ th column of X_k and is defined as,

$$x_k^i = [x(kL-i)x(kL-i-1) \dots x(kL-i-L+1)]^T \quad (5)$$

The matrix-vector product y_k can be expressed in the form of scalar-vector product, by substituting (4) in (3)

$$y_k = \sum_{i=0}^{N-1} x_k^i \cdot h(i) \quad (6)$$

Considering N as a composite number, we can rewrite it as $N=ML$. Now we can express the index i as $i=l+mL$, for $0 \leq l \leq L-1$ and $0 \leq m \leq M-1$.

Putting $i=l+mL$ in (5) we get,

$$x_k^{l+mL} = x_{k-m}^l \quad (7)$$

Substituting (7) in (4), we get,

$$X_k = [x_k^0 x_k^1 \dots x_k^{L-1} x_{k-1}^0 x_{k-1}^1 \dots x_{k-1}^{L-1} \dots x_{k-M+1}^0 x_{k-M+1}^1 \dots x_{k-M+1}^{L-1}] \quad (8)$$

Now we substitute (8) in (3), then we get

$$y_k = \sum_{l=0}^{L-1} \sum_{m=0}^{M-1} x_{k-m}^l \cdot h(l+mL) \quad (9)$$

If you look through the input matrix X_k of (8), the current block is the data block x_k^0 while the rest are the blocks delayed by $1, 2, \dots, (M-1)$ cycles given by $\{x_{k-1}^0, x_{k-2}^0, \dots, x_{k-M+1}^0\}$. x_k^l is the overlapped block and $\{x_{k-1}^l, x_{k-2}^l, \dots, x_{k-M+1}^l\}$ are $1, 2, \dots, (M-1)$ clock cycles delayed blocks of x_k^l . In order to utilize this property, we can distribute the input matrix X_k into M number of small matrices given as S_k^l . These contain inner matrices such as S_k^0 containing L input blocks $\{x_k^0, x_{k-1}^0, \dots, x_{k-M+1}^0\}$ and S_k^l containing $\{x_{k-1}^l, x_{k-2}^l, \dots, x_{k-M+1}^l\}$. The matrix S_k^{M-1} would comprise of $\{x_{k-M+1}^0, x_{k-M+1}^1, \dots, x_{k-M+1}^{L-1}\}$. We can find S_k^m to be in symmetry and hence it will satisfy the symmetric relation. Therefore, we can rewrite it as follows:

$$S_k^m = S_{k-m}^0 \quad (10)$$

We can also decompose the coefficient vector h into small weight vectors given by $c_m = \{h(mL), h(mL+1), \dots, h(mL+L-1)\}$. S_k^m (for $1 \leq m \leq M-1$) are delayed by m clock cycles with

respect to S_k^0 . Taking into consideration S_{k-m}^0 and c_m , we can express (9) as a matrix-vector product as:

$$y_k = \sum_{m=0}^{M-1} r_k^m \quad (11a)$$

$$r_k^m = S_{k-m}^0 \cdot c_m \quad (11b)$$

We can also express the above equations in a recurrence form as:

$$Y(z) = S^0(z) [(z^{-1} (\dots (z^{-1} (z^{-1} c_{M-1} + c_{M-2}) + c_{M-3}) + \dots) + c_1) + c_0] \quad (12)$$

Where $S^0(z)$ and $Y(z)$ are the z -domain representation of S_k^0 and y_k respectively.

3. Existing Method

The existing method consists of transpose form block FIR filter architecture for fixed coefficients with MCM based approach. The existing method occupied larger area since it consisted of the MCM block. This MCM block lead to more complex representation thus making the computations to be more problematic. This has also lead to slow down the performance of the system and caused a delay in the output computation.

4. Proposed Method

For the proposed MCM bases implementation of block FIR filter, we will be taking into account the symmetry of input matrix S_k^0 . This would help us to discard the common subexpression and would decrease the number of shift add operations in the MCM blocks.

We can also write the recurrence relation of (12) as follows:

$$Y(z) = z^{-1} \dots z^{-1} (z^{-1} r_{M-1} + r_{M-2} + r_{M-3}) + \dots + r_1 + r_0 \quad (13)$$

Where r_m are the M intermediate data vectors for $0 \leq m \leq M-1$. Hence, we can write,

$$R = S_k^0 \cdot C \quad (14)$$

R and C can be defined as,

$$R = [r_0^T r_1^T \dots r_{M-1}^T] \quad (15a)$$

$$C = [c_0^T c_1^T \dots c_{M-1}^T] \quad (15b)$$

For filter length $N=16$ and block size $L=4$, we can express (14) in matrix product format as shown in (16).

$$R = \begin{bmatrix} x(4k) & x(4k-1) & x(4k-2) & x(4k-3) \\ x(4k-1) & x(4k-2) & x(4k-3) & x(4k-4) \\ x(4k-2) & x(4k-3) & x(4k-4) & x(4k-5) \\ x(4k-3) & x(4k-4) & x(4k-5) & x(4k-6) \end{bmatrix}$$

$$X \begin{bmatrix} h(0) & h(4) & h(8) & h(12) \\ h(1) & h(5) & h(9) & h(13) \\ h(2) & h(6) & h(10) & h(14) \\ h(3) & h(7) & h(11) & h(15) \end{bmatrix} \quad (16)$$

We can apply MCM in both horizontal and vertical direction of the coefficient matrix. This could be proved by the illustrations stated in Table I.

Table I: MCM in Transpose Form Block FIR filter for block size $L=4$ and filter length $N=16$

Input Sample	Group of Coefficients
$x(4k)$	{h(0),h(4),h(8),h(12)}
$x(4k-1)$	{h(0),h(4),h(8),h(12)} {h(1),h(5),h(9),h(13)}
$x(4k-2)$	{h(0),h(4),h(8),h(12)} {h(1),h(5),h(9),h(13)} {h(2),h(6),h(10),h(14)}
$x(4k-3)$	{h(0),h(4),h(8),h(12)} {h(1),h(5),h(9),h(13)} {h(2),h(6),h(10),h(14)} {h(3),h(7),h(11),h(15)}
$x(4k-4)$	{h(1),h(5),h(9),h(13)} {h(2),h(6),h(10),h(14)} {h(3),h(7),h(11),h(15)}
$x(4k-5)$	{h(2),h(6),h(10),h(14)} {h(3),h(7),h(11),h(15)}
$x(4k-6)$	{h(3),h(7),h(11),h(15)}

From the matrix product (16), we are able to see that the input sample $x(4k)$ is present only once whereas the sample $x(4k-3)$ makes its appearance in all four rows and columns. Therefore because of the presence of the input sample $x(4k-3)$ in all rows and columns it can get the involvement of all the four rows of the coefficient matrix in the MCM. But since $x(4k)$ appears only for one time, only the first row of the coefficient matrix elements will get involved with it in the MCM. The proposed structure for MCM-Based Implementation of FIR filter for block size $L=4$ is shown in Fig. 1. It consists of a register unit (RU). The outputs from the register unit are required to be mapped to Structured MCM Units so that the complexity could be reduced. The six Structured MCM entities will receive the six outputs from the RU which will act as six input samples for each unit. These six units are collectively termed as High Performance MCM block. The outputs from this block are fed to the adder network. This adder network performs shift adder operations to generate inner-product values ($r_{l,m}$), for $0 \leq l \leq L-1$ and $0 \leq m \leq (N/L)-1$. The outputs obtained from the adder network are fed to the PAU (Pipelined Adder Unit) and are added to produce a final block of output. The proposed structure involves high performance MCM block which helps us in speeding up the process because of its internal functionality. It also improves the delay in the output and facilitates the result generation fast. It also reduces the complexity of the circuit and makes computations easier.

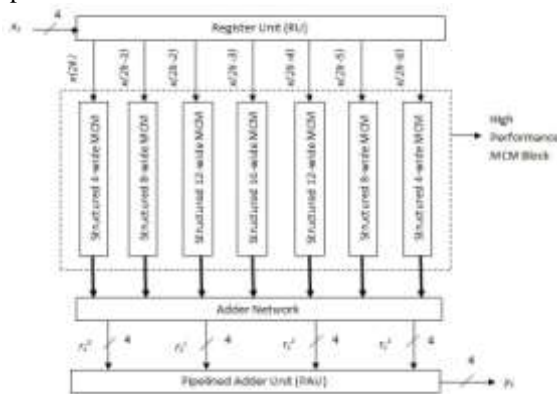


Figure 1: Proposed MCM-based Structure for fixed coefficient FIR filter of block size $L=4$ and filter length $N=16$

5. Results

We have implemented the proposed structure in VHDL for filter length $N=16$ and block size of $L=4$. The implementation results are tabulated and presented in tables II and III.

Table II: Timing Summary

Timing Parameter	Existing Method	Proposed Method	Improvisation in Delay
Minimum Period	24.562ns	5.415ns	77.95%
Maximum Frequency	40.713MHZ	184.685MHZ	-353.63%
Minimum input arrival time before clock	25.281ns	6.663ns	73.64%
Maximum output required time after clock	32.103ns	5.749ns	82.09%

Table III: Area Analysis

Parameter	Existing Method	Proposed Method	Area Accumulated
Number of slices used	515	319	38.05%
Number of slice Flip-Flops	41	430	-948.78%
Number of 4 input LUTs	887	512	42.28%
Number of bonded IOBs	57	57	0
Number of GCLKs	1	1	0

Fig.2 shows the test bench waveforms of MCM-based implementation of Fixed-Coefficient FIR filter.



Figure 2: Simulation Test Bench Wave forms of MCM-Based Implementation of Fixed-Coefficient FIR filter

6. Conclusion and Future Scope

The Fixed Coefficient FIR filter architecture has been realized in MCM based implementation in VHDL in Xilinx 14.7. This analysis has proved to be efficient since it has advanced the delay by 77.95%. Also it has speeded up the input arrival time and the output time required before clock. In terms of area, we were able to observe 38% of area savings which reduced the complexity of the block and facilitated in easy computations. In future, many more advancements can come up with the aim of further reducing delay and area consumption.

7. Acknowledgements

The authors hereby acknowledge their gratitude to the management of QIS Institute of Technology, Ongole to have provided laboratory facility for implementing this project.

References

- [1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
- [2] T. Hentschel and G. Fettweis, "Software radio receivers," in CDMA Techniques for Third Generation Mobile Systems. Dordrecht, The Netherlands: Kluwer, 1999, pp. 257–283.
- [3] E. Mirchandani, R. L. Zinser, Jr., and J. B. Evans, "A new adaptive noise cancellation scheme in the presence of crosstalk [speech signals]," IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 39, no. 10, pp. 681–694, Oct. 1995.
- [4] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in Proc. IEEE Southeastcon, Apr. 1993, p. 1–6.
- [5] J. Mitola, Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering. New York, NY, USA: Wiley, 2000.
- [6] A. P. Vinod and E. M. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," IEEE Trans. Wireless Commun., vol. 7, no. 5, pp. 1669–1675, Jul. 2006.
- [7] J. Park, W. Jeong, H. Mahmoodi-Meimand, Y. Wang, H. Choo, and K. Roy, "Computation sharing programmable FIR filter for low-power and high-performance applications," IEEE J. Solid State Circuits, vol. 39, no. 2, pp. 348–357, Feb. 2004.
- [8] K.-H. Chen and T.-D. Chiueh, "A low-power digit-based reconfigurable FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 617–621, Aug. 2006.
- [9] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 2, pp. 275–288, Feb. 2010.
- [10] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 61, no. 7, pp. 511–515, Jul. 2014.
- [11] P. K. Meher, "Hardware-efficient systolization of DA-based calculation of finite digital convolution," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.
- [12] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans. Signal Process., vol. 56, no. 7, pp. 3009–3017, Jul. 2008.
- [13] P. K. Meher, "New approach to look-up-table design and memorybased realization of FIR digital filter," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 57, no. 3, pp. 592–603, Mar. 2010.

Author Profile



Sheena Pathan received her B. Tech Degree in Electronics And Communication Engineering, from Acharya Nagarjuna University, Guntur. She is presently Pursuing M. Tech (VLSI & ES) in QIS Institute of Technology, Ongole (affiliated to JNTU, Kakinada), AP, India. Her current research interests are VLSI, Embedded Systems, digital signal processing and Communication.



Mr. Shaik. Rahamtula is currently working as assistant professor in ECE Department, QIS Institute of technology, Ongole, A.P, India. He received his B. Tech degree in the department of Electronics and Communication Engineering, from KMCET (Affiliated to JNTU Hyderabad). He received his M. Tech from QIS college of Engineering and Technology (Affiliated to JNTU Kakinada), Presently Pursuing Ph.D in VELS University, Chennai. His research interests in the area of Face recognition, Content based image retrieval, Image compression, VLSI, Embedded Systems, digital signal processing and Communication.