International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 03 | Mar -2017 www.irjet.net p-ISSN: 2395-0072

TRANSPOSE FORM FIR FILTER DESIGN FOR FIXED AND RECONFIGURABLE COEFFICIENTS

Ariprasath.S*[1],Dr.C.Santhi[2]

[1]*Pg scholar, Dept of ECE, Government College of Technology, Coimbatore, Tamilnadu, India*
[2]*Associatet professor, Dept of ECE, Government College of Technology, Coimbatore, Tamilnadu, India*

Abstract -*Transpose form finite impulse response (FIR) filter is naturally a pipelined structure which supports the multiple constant multiplications (MCM) technique but direct form FIR filter structure does not support MCM technique. The MCM is more effective in Transpose form when the common operand is multiple with the set of constant coefficients that reduce the computational delay. The implementation of MCM technique is easier in fixed coefficient Transpose form FIR filter but complex in reconfigurable coefficients. In fixed coefficients transpose FIR filter, area and delay are reduced by using MCM technique. The low-complexity design using the MCM technique is implemented for fixed coefficients transpose form FIR filters and multiplier-based design is used for reconfigurable transpose form FIR filter. The implemented transpose form FIR filter structure achieved less area and delay than the direct-form FIR filter structure. The XILINX software tool is used for simulation.*

***Key Words*:Transpose form FIR filter, multiple constant multiplications (MCM) technique, Block processing**

1. INTRODUCTION

Finite Impulse response (FIR) digital filter is used in several DSP applications, such as, echo cancellation, speech processing, equalization, adaptive noise cancellation, and various communication applications, including software-defined radio (SDR), etc. Many of these applications require FIR filters of large order to meet the stringent frequency specifications. And this filters need to support high sampling rate for high-speed digital communication. The number of multiplications and additions required for their filter output, increases linearly with the filter order.

There is no redundant computation available in the FIR filter, real-time implementation of a large order FIR filter in a resource constrained environment is a challenging task. Filter coefficients very often remain constant and known *a priori* in signal processing applications. This feature has been utilized to reduce the complexity of realization of multiplications. FIR filter has two configurations, namely direct form FIR filter and transposes form FIR filter.

The Transpose form FIR filter can be constructed from the direct form FIR filter by Exchanging the input and output and inverting the direction of signal flow.Generally,Transpose form FIR filters are support multiple constant multiplications (MCM) technique that results in saving of computation time**.** Transpose form finite-impulse response (FIR) filters are inherently pipelined and support multiple constant multiplications (MCM) technique that results in significant saving of computation. However, transpose form configuration does not directly support the block processing unlike direct form configuration. Generally,Transpose form FIR filters are support multiple constant multiplications (MCM) technique that results in saving of computation time**.** TheMCM is an arithmetic operation that multiplies a set of fixed-point constants (Example: H0, H1, H2,...., Hn) with the same fixed-point variable (X).

Several designs have been designed by various researchers for efficient realization of FIR filters (having Fixed coefficients) using multiple constant multiplication (MCM) and distributed arithmetic(DA) [6] and methods [7]. DA-based designs use lookup tables (LUTs) to store pre-computed result stored the computational complexity.

The Transpose form FIR filter only needs *N* delay units, where *N* is the order of the filter – potentially half as much as direct form. This structure is obtained by reversing the order of the numerator and denominator sections of Direct Form, since they are in fact two linear systems. Then, one will notice that there are two columns of delays that tap off the center net, and these can be combined since they are redundant. The disadvantage is that Transpose form increases the possibility of arithmetic overflow for filters of high *Q* or resonance. This is because, conceptually, the signal is first passed through an all-pole filter (which normally boosts gain at the resonant frequencies) before the result of that is

saturated, then passed through an all-zero filter (which often attenuates much of what the all-pole half amplifies).

In this paper, we realize the possibility of block FIR filter in transpose form configuration to take advantage of the MCM technique and the naturally pipelining for area delay efficient realization of large order. The main contributions of this paper are as follows.

1) The Computational analysis of transpose form FIR filter and derivation of flow graph with reduced register complexity.

2) Block formulation for transpose form FIR filter.

3) Design of transpose form FIR filter block for reconfigurable applications.

4) A low-complexity design method using multiple constant multiplications technique for the block implementation of fixed FIR filters.

2. REALIZATION OF TRANSPOSE FORM FIR FILTER

The realization of FIR filter in transpose form configuration is discussed in this chapter. The data-flow graphs (DFG) of transpose form FIR filter for filter length *N*= 6 as shows below.

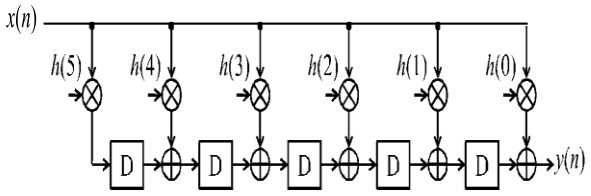


Fig.1. *DFG 1 of transpose form structure for N =6.For output y(n).*

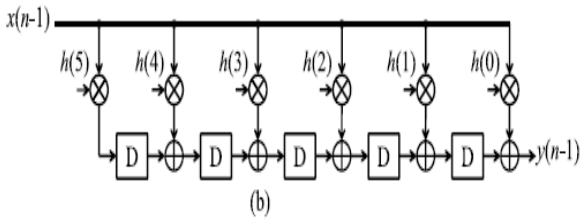


Fig.2. *DFG 2 of transpose form structure for N =6.For output y (n-1).*

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

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

A block of two successive outputs $\{y(n), y(n-1)\}$ that are derived from the above DFG's. The product values and their accumulation paths in DFG-1 and DFG-2 are shown in data- flow tables (DFT-1 and DFT-2) of Fig. 3 and 4. The arrows in DFT-1and DFT-2 of Fig. 3 and 4, represent the accumulation path of the products. We see that five values of each column of DFT-3 are same as those of DFT-4 (shown in Gray colour in Fig.3 and 4). These redundant computations of DFG-1 and DFG-2 can be avoided using nonoverlapped sequence of input blocks, as shown in figure 5.

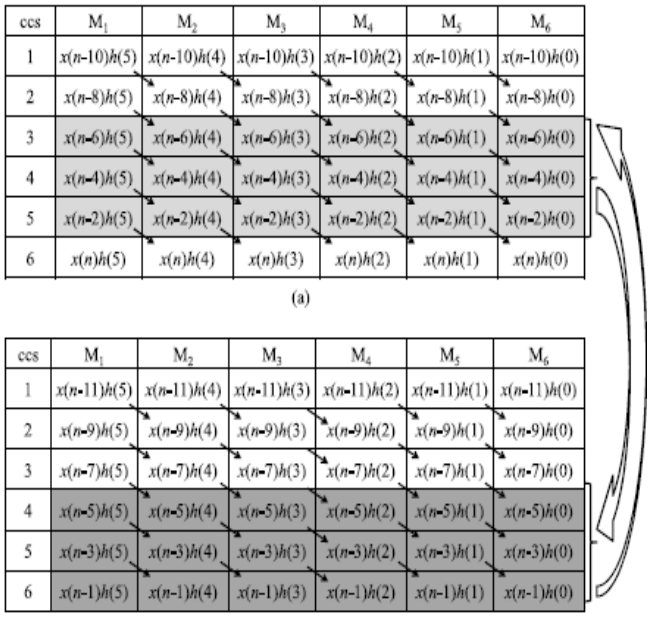
ccs	M_1	M_2	M_3	M_4	M_5	M_6
1	$x(n-5)h(5)$	$x(n-5)h(4)$	$x(n-5)h(3)$	$x(n-5)h(2)$	$x(n-5)h(1)$	$x(n-5)h(0)$
2	$x(n-4)h(5)$	$x(n-4)h(4)$	$x(n-4)h(3)$	$x(n-4)h(2)$	$x(n-4)h(1)$	$x(n-4)h(0)$
3	$x(n-3)h(5)$	$x(n-3)h(4)$	$x(n-3)h(3)$	$x(n-3)h(2)$	$x(n-3)h(1)$	$x(n-3)h(0)$
4	$x(n-2)h(5)$	$x(n-2)h(4)$	$x(n-2)h(3)$	$x(n-2)h(2)$	$x(n-2)h(1)$	$x(n-2)h(0)$
5	$x(n-1)h(5)$	$x(n-1)h(4)$	$x(n-1)h(3)$	$x(n-1)h(2)$	$x(n-1)h(1)$	$x(n-1)h(0)$
6	$x(n)h(5)$	$x(n)h(4)$	$x(n)h(3)$	$x(n)h(2)$	$x(n)h(1)$	$x(n)h(0)$

Fig.3.Corresponding to output y (n).

ccs	M_1	M_2	M_3	M_4	M_5	M_6
1	$x(n-6)h(5)$	$x(n-6)h(4)$	$x(n-6)h(3)$	$x(n-6)h(2)$	$x(n-6)h(1)$	$x(n-6)h(0)$
2	$x(n-5)h(5)$	$x(n-5)h(4)$	$x(n-5)h(3)$	$x(n-5)h(2)$	$x(n-5)h(1)$	$x(n-5)h(0)$
3	$x(n-4)h(5)$	$x(n-4)h(4)$	$x(n-4)h(3)$	$x(n-4)h(2)$	$x(n-4)h(1)$	$x(n-4)h(0)$
4	$x(n-3)h(5)$	$x(n-3)h(4)$	$x(n-3)h(3)$	$x(n-3)h(2)$	$x(n-3)h(1)$	$x(n-3)h(0)$
5	$x(n-2)h(5)$	$x(n-2)h(4)$	$x(n-2)h(3)$	$x(n-2)h(2)$	$x(n-2)h(1)$	$x(n-2)h(0)$
6	$x(n-1)h(5)$	$x(n-1)h(4)$	$x(n-1)h(3)$	$x(n-1)h(2)$	$x(n-1)h(1)$	$x(n-1)h(0)$

Fig.4.Corresponding to output $y(n-1)$.

Arrow: accumulation path of the products. Block-processing method is highly used to derive high-throughput in hardware structures but also improves the area-delay efficiency. The formation of block-based FIR structure is straightforward in the direct-form FIR, 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. The nonoverlapping input blocks are derived form the block processing. The block doesnot involve redundant computation. It is easy to find that the entries in gray cells in below (Fig.5) table correspond to the output $y(n)$ and correspond to $y(n-1)$.



2.1RECONFIGURABLE TRANSPOSE FORM FIR FILTER

The coefficients are varying for some applications in DSP, which is referred as reconfigurable coefficients. For example, Software-defined radio (SDR) channelizer, Where FIR filters need to be implemented in reconfigurable coefficients. The Reconfigurable structure for block FIR filter is shown in below for the block size *L* = 4. 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). 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 RU receives **x***k*during the *k*th cycle and produces *L* rows of **S**0*k* in parallel. *L* rows of **S**0*k*are transmitted to *M* IPUs of the reconfigurable structure.

The *M* IPUs also receive *M* short-weight vectors from the CSU, such that during the *k*th cycle, the *(m* + 1*)*th IPU receives the weight vector **c***M*−*m*−1 from the CSU and *L* rows of **S**0*k* form the 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 *(***r***mk)*.

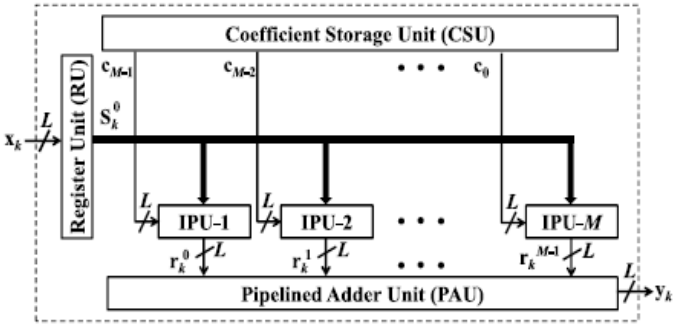


Fig.5. *Structure of Reconfigurable coefficient FIR filter*

In the coefficient storage unit, the coefficients used for the Reconfigurable applications are saved. 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 selection coefficient, usually denoted by the letter *s*, is a measure of differences in fitness. Selection coefficients are central to the quantitative description of evolution, since fitness differences determine the change in genotype frequencies attributable to selection.Pipelining is an important technique used in several applications such as digital signal processing (DSP) systems. Consider a 3-tap FIR filter, Assume the computational time for multiplication units is T_m and T_a for add units.

The critical path, representing the minimum time required for processing a new sample, is limited by 1 multiplication and 2 add function units. The sampling period is given by Tm+2Ta.

2.2. FIXED COEFFICIENT TRANSPOSE FORM FIR FILTER

The coefficients of FIR filter are fixed in some application which is known as fixed coefficients. The transpose form FIR filter is naturally a pipelined structure which supports the multiple constant multiplications (MCM) technique but direct form FIR filter structure does not support MCM technique. The MCM is more effective in Transpose form when the common operand is multiple with the set of constant coefficients that reduce the computational delay. The implementation of MCM technique is easier in fixed coefficient Transpose form FIR filter but complex in reconfigurable coefficients. In fixed coefficients transpose FIR filter, area and delay are reduced by using MCM technique.

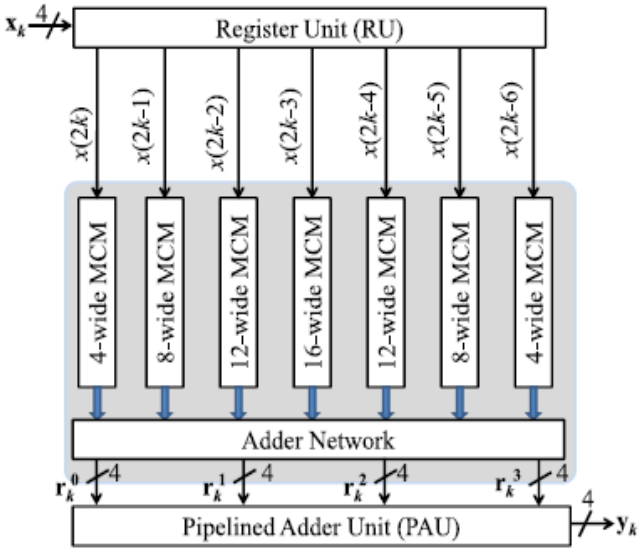


Fig.6. *MCM-based structure for fixed FIR filter of block size L = 4*

For fixed-coefficient implementation, the CSU is no longer required, since the structure is to be tailored for only one given filter. Similarly, IPUs are not required. The multiplications are required to be mapped to the MCM units for a low-complexity realization. In the following, we show that the proposed formulation for MCM-based implementation of block FIR filter makes use of the symmetry in input matrix **S**0*k* to perform horizontal and vertical common subexpression elimination and to minimize the number of shift-add operations in the MCM blocks.

To illustrate the computation of 4 tap Fixed FIR and 16 coefficients, we write it as a matrix product by using the above relation. We can observe that the input matrix contains six-input samples {*x(4k), x(4k* − 1*), x(4k* − 2*), x(4k* − 3*), x(4k* − 4*), x(4k* − 5*), x(4k* − 6*)*}, and multiplied with several constant coefficients. MCM can be applied in both horizontal and vertical direction of the coefficient matrix. The sample *x(4k*−3*)* appears in four rows or four columns of the above input matrix. Whereas *x(4k)* appears in only one row or one column. Therefore, all the four rows of coefficient matrix are involved in the MCM for the *x(4k* − 3*)*, whereas only the first row of coefficients are involved in the MCM for *x(4k)*. For larger values of *N* or the smaller block sizes, the row size of the coefficient matrix is larger that results in larger MCM size across all the samples, which results into larger saving in computational complexity.

3. RESULTS AND DISCUSSION

To implement the fixed and reconfigurable coefficients Transpose form FIR filter is simulated using Xilinx Tool. The area (no. of LUT's and flip-flop's clock buffers and I/O buffers) and delay were analysed for Fixed and reconfigurable coefficients. From the analysis, it is observed that the area and delay reduced when compare to the direct form FIR filter.

In the below simulation result is for Fixed coefficient Transpose form FIR filter with 15 coefficients. The input given to the filter is 0001(1) and the output achieved for the filter is 1010. The delay for the fixed transpose form FIR filter is 1.19 ns. The delay of the fixed transpose form filter is reduced when compare to the direct form filter delay. The delay of the direct form fir filter is 1.36ns.

Fig.**1.***Experimental result for the fixed FIR*

In the below simulation result is for reconfigurable coefficients Transpose form FIR filter. The input given to the filter is 0001(1) and the coefficients given for the filter is, 0000, 0001, 0010, 0011, 0100, and 0101. The output achieved for the filter is 1111 (15). The total number LUT used for the reconfigurable FIR is 152 and the Flip-flops are 63 and the shift register, clock buffers and I/O buffers are used 27 cells. The LUTs, Flip-flops are reduced when compare to the direct form FIR filter. The delay for the reconfigurable transpose form FIR filter is 1.27 ns. The delay of the fixed transpose form filter is reduced when compare to the direct form filter delay. The delay of the direct form fir filter is 1.35ns.

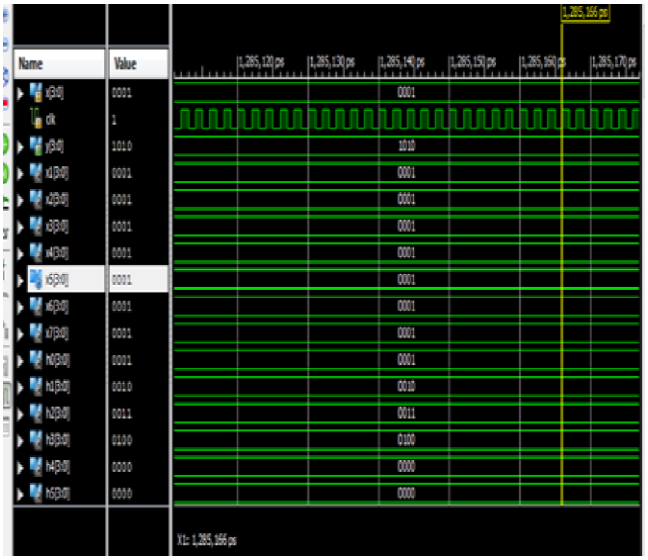


Fig.**2.** *Experimental result for the fixed FIR*

3.1 PERFORMANCE MEASUREMENT

TABLE I: COMPARISON TABLE RECONFIGURABLE FIR FOR THE AREA AND DELAY

	Delay(ns	**LUT's and flip-flop's**
Direct form reconfigurable FIR	1.35	368
Transpose form reconfigurable FIR	1.26	241

TABLE 2:COMPARISON TABLE FIXED FIR FOR THE AREA AND DELAY

	Delay(ns)	**LUT's and flip-flop's**
Direct form reconfigurable FIR	1.36	120
Transpose form reconfigurable FIR	1.198	60

4. CONCLUSION

In this paper, we implemented the design of Transpose form FIR filter for fixed coefficient and reconfigurable coefficients. The transpose form FIR filter is naturally a pipelined structure which supports the multiple constant multiplications (MCM) technique. The implementation of MCM technique is easier in fixed coefficient Transpose form FIR filter but complex in reconfigurable coefficients. In fixed coefficients transpose FIR filter, area and delay are reduced by using MCM technique. The low-complexity design using the MCM technique is implemented for fixed coefficients transpose form FIR filters and multiplier-based design is used for reconfigurable transpose form FIR filter. The implemented transpose form FIR filter structure achieved less area and delay than the direct-form FIR filter structure. In future, the delay and area will be further reduced in transpose form FIR filter by applying another technique

5. REFERENCES

[1] J. G. ProakisandD. G. Manolakis, "*DIGITAL SIGNAL PROCESSING: PRINCIPLES, ALGORITHMS ANDAPPLICATIONS*"Upper SaddleRiver,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],", vol. 39, no. 10, pp. 681–694, Oct. 1995

[4] 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*, Feb. 2004. p. 1–6.

[5] K.-H. Chen and T.-D.Chiueh, "A LOW-POWER DIGIT-BASED RECONFIGURABLE FIR FILTER," *IEEE Trans. Circuits Syst. II,* , vol. 53, no. 8, pp. 617–621, Aug. 2006.

[6] R. Mahesh and A. P. Vinod, "NEW RECONFIGURABLE ARCHITECTURES FOR IMPLEMENTING FIR FILTERS WITH LOW COMPLEXITY," vol. 29, no. 2, pp. 275–288, Feb. 2010

[7] 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.