International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 8, August - 2013

Transposed Form Fir Filter Implementation Using Reconfigurable Architectures

P.Sowjanya [1] , P.Pushpalatha [2]
Department of ECE, University college of engineering, JNTU Kakinada 533003, Kakinada,Andhra Pradesh

Abstract

*FIR filters require two factors Low complexity and reconfiguability it is main concern especially in case of higher order filters. Complexity is depend on coefficient multipliers in filters. Multiplication complexity is based on number of adders used for multiplication. The well known Architectures CSM based on filter coefficient partitioning and PSM implemented by usig BCSE algorithms are used in this paper to achieve the requirements. The BCSE algorithm eliminates the same bit patterns present in filter coefficients so that number of computations are reduced this will efficiently reduce the number of adders so the complexity of hardware circuitary is reduced . The proposed architectures are implemented and tested on FPGA and synthesized using Xilinx ISE. The obtained results show that this methods offers good area and power reduction compared to the existing methods.***

***Keywords*—Constant shift method, programmable shift method, Coefficient partitioning, Binary common subexpression elimination.**

1.Introduction

Finite impulse response filters are digital filters, which have a finite impulse response. FIR filters do not employ any feedbacks and are also known as non-recursive filters, convolution filters, or moving-average filters because you can express the output of a FIR filter as a finite convolution as shown in equation (1)

$$y = h * x \qquad - (1)$$

Where y is the output of FIR filter, h is filter coefficient and x is the input value.
The different architectures of FIR filters are

1.Direct Form FIR Filter
2.Transposed Form FIR Filter
3.Symmetric Form FIR Filter
4.Distributed Arithmetic FIR Filter

A variation of the direct FIR model is called the transposed FIR filter. It can be constructed from the direct form FIR filter by following the steps
1.Exchanging the input and output
2. Inverting the direction of signal flow
3. Substituting an adder by a fork

Transposed FIR filter is the preferred implementation of an FIR filter. The benefit of this filter is that there is no need for an extra pipeline stage for the adder of the products to achieve high throughput.

The complexity of FIR filters is mainly depend on coefficient multiplication. The two key factors that determine the complexity of coefficient multiplications in FIR filters are the number of LOs and LD. LOs represent the adders required in computing the sum of partial products in the multiplier, it will determine the area and power requirements of the filter circuit. LD is the number of adder steps in a maximal path of decomposed multiplications, which determines the speed of filtering operations. Therefore the focus of low-complexity FIR filter implementation algorithms is on reducing the number of LOs and LD in coefficient multipliers.

Many algorithms proposed in literature [4] for the implementation of low-power and high-speed FIR filters with a minimum number of LOs and LD. Among these approaches ,the CSE techniques based on binary formate produced the best hardware reduction compared to the CSD based CSE and graphical methods in [3] . The proposed CSM architecture depends on partitioning the filter coefficients into fixed groups and the PSM architecture is implemented by eliminating redundancy with the help of BCSE. CSM offer good power reduction but it will slightly increase the area requirement. PSM offer good tradeoff between power and area and by usig PSM architecture the length of the filter coefficients can be changed without any modification in hardware circuitry.

1. Existing method architecture

The existing method uses adder for the multiplication . Fig.I. shows the existing method architecture.

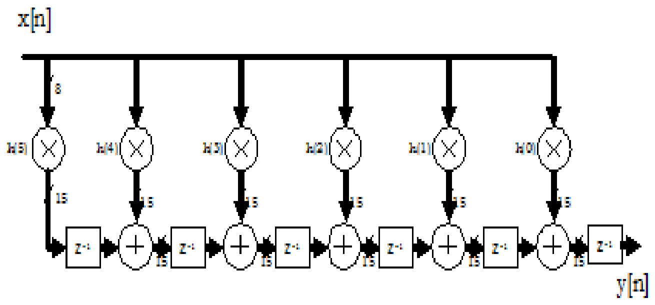

Fig. I. Existing method architecture

In Fig.I. x[n] represents the input, h(0),h(1)---h(5) represents the filter coefficients and z^{-1} represents the delay element. In this architecture the number of multipliers and and adders will increase with increase in number of filter order so the complexity of FIR filter is increased. This method uses CSD representation of filter coefficients the number of unpaired bits required to represent coefficients are more compared the proposed method in [6]. The look ahead method is used in CSD fror selecting patterns and to eliminate redundant horizontal and vertical common subexpressions in filter coefficients. The general representation of CSD for the ith order filter coefficient is expressed in equation (2).

$$hi = \sum_{j=0}^{B-1} 2^{a} ij \qquad -(2)$$

Where i,j are integers from 0 to n here n is length of filter coefficient. The number of bits required to represent filter coefficients and the complexity of FIR filter is reduced by usig proposed method.

2. Proposed Method Architecture

The proposed architectures based on transposed direct form FIR filter it is mainly consists of shift and add unit, multiplexer block and adder unit as shown in Fig. II.

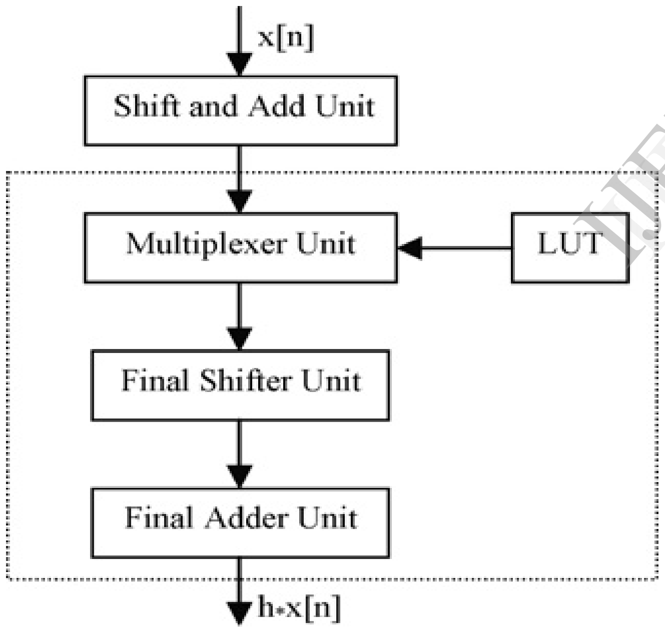

Fig. II. Architecture of the proposed method.

The dotted portion in Fig .II represents the processing element in the proposed architectures and these processing elements are different for CSM and PSM.

2.1.Shift and add unit

Shift and add unit is similar to the multiplier this unit adds the multiplicand x to itself y number of times where y is multiplier term. The shift and add unit architecture is shown in Fig.3.

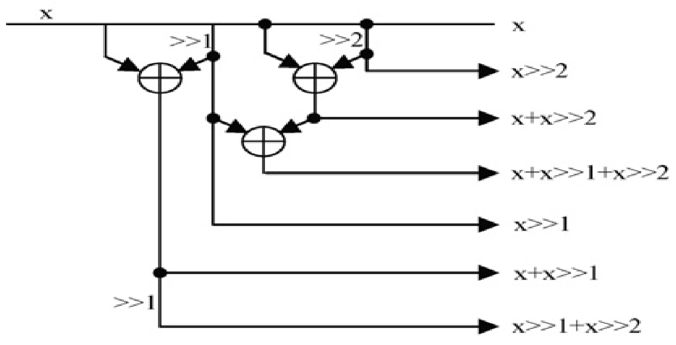

Fig. III. Architecture of shift and add unit.

In the figure x>>k represents the right shift of input by k times this shift and add unit require 3 adder stages but the conventional methods of multiplication require 5 adder stages so the proposed methods CSM and PSM in [6] use shift and add units for the multiplication purpose.

2.1. Multiplexer unit

This unit is to multiply the output of shift and add unit with the filter coefficients stored in look up table the input is variable and filter coefficients are constants and the multiplication here is MCM in [8]. The number of multiplexers required in CSM are based on number of groups that the filter coefficient is partitioned and in PSM the number of multiplexers required depends upon the number of non zero terms in the filter coefficients.

2.2. Final shifter unit

This unit shifts the result after all intermediate additions are performed the output of the final shifter unit is shown in equation (3)

$$y = 2^{-4}x + 2^{-6}x + 2^{-15}x + 2^{-16}x \qquad -(3)$$

After coefficient partitioning the output is

$$y = 2^{-4}(x + 2^{-2}x) + 2^{-15}(x + 2^{-1}x) \qquad -(4)$$

The final shifter unit is different for the architectures CSM and PSM for CSM the shifter is constant but for PSM the shifter is programmable.

A. Architecture of Constant Shift Method

CSM architecture is implemented by partitioning the filter coefficients and those coefficients are directly stored in LUT those grouped coefficients are used as select signals for the multiplexer. The number of multiplexers required for the CSM are n/3 here n is the filter coefficient length. If filter coefficient length is 6 then number of multiplexers required are 2. For better understanding consider the below example.

h='0.111111'

Number of non zero bits are 6so n=6and number of multiplexers are 6/3=2 and h can be expressed as

$$y = 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x + 2^{-6}x$$

After partitioning y can be written as

$$y = 2^{-1}(x + 2^{-1}x + 2^{-2}x + 2^{-3}x + 2^{-4}x + 2^{-5}x)$$

The Fig. IV shows the CSM architecture.

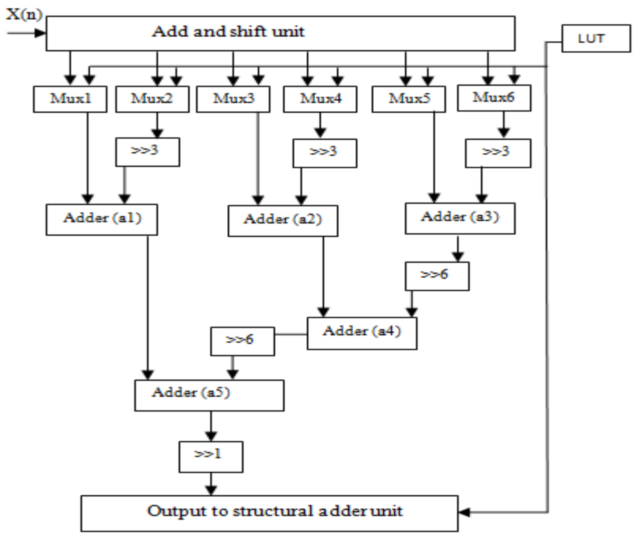

Fig. IV. CSM Architecture

The steps involved in CSM are as follows
Step 1: Take the input x.
Step 2: Read the coefficients from the LUT and use as the select signal for the multiplexers.
Step 3: Perform the final shifting function on the output of the multiplexer.
Step 4: Perform the addition of intermediate sums using the final adder unit.
Step 5: Store the final result, h*x, in the delay unit „D".
Step 6: Go to step 2 if the coefficients in the LUT are not finished, else go to 1

B. Architecture of Programmable Shift Method

The PSM method is implemented based on CSE algorithm. In this architecture instead of constant shifting used in CSM programmable shifters are used.Filter coefficients are coded and stored in LUT.The coding procedure is as follows

Consider h=[1010011001010011]

Assign 2=[101] and 3=[101]

By substituting the above values h will become as

h=[3000020003000020]

The h will be stored in LUT as 000001101011011110 and 100111111010000000 each coefficient in coded format. Consists of sign bit, shift values . As the coded formate of filter coefficients are in coded formate the number of binary digits required to represent filter coefficients are reduced compared to the other existing methods used in [2],[3]. The architecture of PSM is shown in Fig. IV.

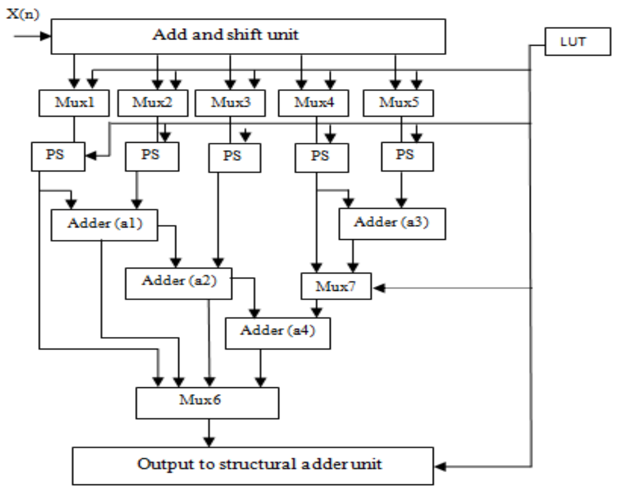

Fig.V. PSM architecture

The steps involved in PSM implementation are as
Step 1: Obtain the BCSs from filter coefficients using CSE algorithm.
Step 2: Store the resultant coefficients in the LUT.
Step 3: Take the input x.
Step 4: Read the coefficients from the LUT and use as the select signal for the multiplexers and the programmable shifters.
Step 5: Perform the final shifting function on the output of the multiplexer using PS
Step 6: Perform the addition of intermediate sums using the final adder unit.
Step 7: Store the final result in the delay unit "D".
Step 8: Go to step 4 if the coefficients in the LUT are not finished, else go to 3.

C. Comparison between CSM and PSM

1. In CSM filter coefficients are grouped and are used as select signals for multiplexer.
2. In PSM coding of filter coefficients are done by using BCSE.
3. By using PSM Number off adders required are reduced to great extent compared to CSM.
4. PSM is independent on filter coefficient length but CSM is depend on length of filter coefficient

4. Experimental results

4.1. Synthesis Results

The VHDL code is developed for the proposed architectures and are synthesized on Xilinx 9.2i.The below table shows the comparison between CSM and PSM in terms of Gate count and area occupation.

Table. 1
Synthesis Results for Existing method and proposed method

	Existing method		Proposed method	
parameter	CSD-CSM	CSD-PSM	BCSM	BPSM
Gate count	1534	1523	1420	1404
Area	5.6	5.2	4.3	4.1

From the Table I, it is clear that the number of gates required for Proposed method are less than that of the Existing method so the hardware complexity is reduced by using Proposed architecture and it is also clear by looking into the table the area requirement is also less for the Proposed method so the proposed methods CSM and PSM are best suitable methods for the design of low complex FIR filters.

Table. 2
Comparison between CSM and PSM in terms of delay

Parameter	CSM	PSM
Power	84	48.3
Delay	9.08	9.87

From Table. 2 it is clear that the power consumption is also reduced in PSM compared to that of CSM.

5.Conclusion

Hence the main objective of this project is to design FIR filter with the features of reconfigurability and Low Complexity. With the proposed architectures CSM and PSM the requirements are achieved. So it is clear that these architectures are well suited to design higher order filters compared to that of the other existing methods.

6.References

[1]. Mahesh, R. and Vinod A. P. (2010) "New Reconfigurable Architectures for Implementing FIR Filters with Low Complexity‟, computer-aided design of integrated circuits and systems, Vol. 29, No.
[2]. Vinod, A.P. and Lai, E.(2006) "Low Power and High-Speed Implementation of FIR Filters for Software Defined Radio Receivers‟, IEEE Trans. WirelessCommun., Vol. 5, No. 7, pp. 1669–1675.
[3]. Mitola,J.(2000) "Object-oriented approach wireless systems engineering," in Software Radio Architecture.
[4]. Wang,Y.and Mahmoodi,H. (2004) "Hardware architecture and VLSI implementation of a low-power high-performance polyphase channelizer with applicationsto subband adaptive filtering," in Proc. IEEE Int. Conf. Acoust. Speech Signal Process, vol.5., pp. 97–100.
[5]. Hartley, R.I. (1996) „Subexpression Sharing in Filters Using Canonic Signed Digit Multipliers‟, IEEE Trans. Circuit.
[6]. .A.P.Vinod and E. M.-K. Lai, "An efficient coefficient-partitioning algorithm for realizing low complexity digital filters," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 12, pp. 1936–1946, Dec. 2005.
[7]. A. P. Vinod and E. Lai, "Low power and high-speed implementation of FIR filters for software defined radio receivers," *IEEE Trans. WirelessCommun.*, vol. 5, no. 7, pp. 1669–1675, Jul. 2006.
[8]. H. Samueli, "An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients," *IEEE Trans. Circuits Syst.*, vol. 36, no. 7, pp. 1044–1057, Jul. 1989.
[9]. Y. C. Lim and S. R. Parker, "Discrete coefficient FIR digital filter design based upon an LMS criteria," *IEEE Trans. Circuits Syst.*, vol. CAS-30, no. 10, pp. 723–739, Oct. 1983.
[10]. K. C. Zangi and R. D. Koilpillai, "Software radio issues in cellular base stations," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 561–573, Apr. 1999.