# A Hardware Efficient FIR Filter for Wireless Sensor Networks

**Ch. A. Swamy, I. AdumBabu, J.Narendar,**

*Abstract*—**Finite-impulse response (FIR) Filter is widely used in wireless sensor networks as a signal pre-processing step. Because sensor nodes require a long working periods and ultra-low cost, traditional FIR structures are in applicable as multipliers occupy too much die size for such node's chips. This paper proposes novel FIR filter structures used in the design of application specific integrated circuits (ASICs) for sensor nodes, which can reduce the hardware cost to a minimum. The experiments show that the proposed FIR structure can lead to significant hardware savings from the traditional FIR filter. It's a better choice for sensor node ASICs**

*Keywords*—**Digital signal processing (DSP), application specific integrated circuits (ASICs), finite-impulse response (FIR) algorithm, very large scale integration (VLSI).**

## I. INTRODUCTION

Finite-impulse response (FIR) digital filters are one of the most widely used fundamental devices performed in digital signal processing systems, ranging from video and image processing to wireless communication. Many efforts have been directed toward deriving fast parallel filter structures in the past decades ,an approach to increase the throughput of FIR filters with reduced complexity hardware is presented. This approach starts with the short convolution algorithms, which are transposed to obtain computationally efficient parallel filter structures. Polyphase decomposition and fast FIR algorithms (FFA) are used to implement parallel FIR filter. The FFAs are iterated to get fast parallel FIR algorithms for larger block sizes. Parallel FIR sub-filters are implemented by a set of fast block filtering algorithms which are based on fast short length liner convolution algorithms. However, the number of additions will increase along with the convolution length, which will lead to complex pre-addition and post-addition matrices that are not practical for hardware implementation for wireless sensors. Cook-Toom algorithm and Winograd algorithm are put forward for fast convolution of any filter length, whereas their pre-addition and post-addition matrices may contain elements that are not in the set {-1,0,1},which makes it not suitable for hardware implementation either. A new hardware efficient fast parallel FIR filter structure is obtained by a linear convolution structure based on iterated short convolution algorithm, which can save a large amount of hardware cost. Designsare based on polyphase decomposition, and additional delay elements are integrated into the post-addition matrix, when the block sizes become large, these designs are irregular and require large amount of delay elements. The fast liner convolution is used to develop the small-sized filtering structures and larger block-sized filtering structures are constructed through iterations of the small-sized filtering structures. A new parallel FIR filter structure is presented, which exploits the inherent nature of symmetric coefficients and reduces half multipliers in sub-filter section at the expense of additional adders in preprocessing and postprocessing blocks. However, only under the condition that the number of taps is a multiple of 2 or 3 can the filter structure be beneficial. A computation reduction technique called computation sharing differential coefficient (CSDC) method is presented, which can be used to obtain low-complexity multiplierless implementation of FIR filters. However, we care about the hardware cost and energy consumption more than throughput in wireless sensor networks. Under the circumstances of an acceptable throughput, we need to reduce the hardware cost and energy consumption to a minimum. In other words, the goal is to find out the cheapest and the most energy-efficient way to implement FIR filters, rather than making the filter's processing speed fastest. The proposed FIR algorithm which can be used to implement FIR filters structures in wireless sensor sensor

Manuscript received May 06, 2014
**Ch. A. Swamy,** Assistant Professor, MLRITM, Dundigal (_chaswamy@gmail.com)
**I. AdumBabu,** Assistant Professor, MLRITM, Dundigal ( adumbabu@gmail.com)
**J.Narendar,** Assistant Professor, MLRITM, Dundigal ( jonna.narendar@gmail.com)

networks is described in the following. This paper is organized as follows. A brief introduction of FIR algorithm is given in Section II. In Section III, the proposed FIR structure is presented. Section IV investigates the complexity and comparisons. Section V shows the description of hardware implementation and the experimental results. The conclusion is given in Section VI.

## II. FIR ALGORITHM

An n-tap FIR filter can be expressed in the general form as (1),

$$y(n) = \sum_{i=0}^{N-1} h(i)x(n-i), \quad n=0,1,2,\ldots\infty,$$

where x(n) is an infinite-length input sequence and h(i) are the coefficients of the length-N FIR filter. N multipliers and N-1 adders are needed to implement an N-tap FIR filter. And the traditional L-parallel FIR filter can be derived using polyphase ecomposition, and we can get (2) as follows.

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

where $Y_p = \sum_{k=0}^{\infty} z^{-k}x(Lk+p)$, $X_q = \sum_{k=0}^{\infty} z^{-k}x(Lk+q)$,

$H_r = \sum_{k=0}^{(N/L)-1} z^{-k}x(Lk+r)$, p, q, r=0,1,2,....,L—1. From (2), we

From (2) we can figure out that the traditional L-parallel FIR filter will need $L^2$ FIR sub-filters of length N/L, which means LN multipliers and L (N-1) adders are required for hardware implementation Notice that our goal is to reduce the hardware cost and energy consumption to a minimum, whereas an L-parallel FIR filter will increase the hardware cost, which makes it an improper option of implementing FIR filters in wireless sensor network. Taking no account of processing speed, a simple way to implement FIR filters requires N multipliers and (N-1) adders. Processing speed does not play a pivotal role in wireless sensor networks. What we care about most is how cheap the node is and how long the node can work, not how fast we can get the results. As we all know, the phenomena of optimum processing speed and optimum hardware cost can't appear simultaneously. The reduction in hardware cost will definitely affect the processing speed of FIR filters. Obviously, it's a multi-objective optimization problem. But in this specific situation—wireless sensor network, it's reasonable to sacrifice a certain degree of processing speed for achieving the optimum hardware cost. Without doubt, the processing speed is tolerated and we can ensure that the system will not fail because of cannot get the filter results in time.

Therefore, we propose a novel FIR filter structure which can be used in the design of application specific integrated circuits (ASICs) for sensor nodes for the reason that the filter structure can reduce the hardware cost to a minimum.

## III. PROPOSED FIR ALGORITHM

To reduce the hardware cost to a minimum, the main idea behind the proposed FIR algorithm is actually pretty intuitive: to exchange multipliers and adders with shifters as shifters weigh less than multipliers and adders in terms of silicon area. All we need are an adder and a shifter to implement a FIR filter. Therefore, for an N-tap FIR filter the total amount of saved multipliers would be N, that is we do not need multipliers at all, the total amount of saved adders would be N-2, at the expense of one additional shifter, which occupies much less hardware resource and cut off die size than multipliers and adders. The proposed FIR algorithm will be presented in the flowing. First, encode all the coefficients of the length-N FIR filter in a binary representation with fixed point format, for example, 0.1172 should be presented as the sum of $2^{-3}$, - $2^{-7}$ and - $2^{-16}$. To make it easier to understand, an example is given in the following. Assuming that the coefficients of an 8-tap FIR filter are 0.3759, 0.3086, -0.1255, 0.06439, 0.2187, 0.0168, -0.00415 and 0.01758. And then their binary representations can be expressed as

$h(0) = 0.3759 = 2^{-1}+(-2^{-3})+2^{-10}$

$h(1) = 0.3086 = 2^{-2}+2^{-4}+(-2^{-8})$

$h(2) = -0.1255 = -2^{-3}+(-2^{-11})$

$h(3) = 0.6439 * 10^{-1}= 2^{-4}+2^{-9}$

$h(4) = 0.2187 = 2^{-2} +(-2^{-5})$

$h(5) = 0.0168 = 2^{-6}+2^{-10}+2^{-12}$

$h(6) = -0.415 * 10-2 = -2^{-8}+(-2^{-12})$

$h(7) = 0.1758 * 10\text{-}1 = 2^{-6}+2^{-9}$

According to (1), an 8-tap FIR filter's expansion can be expressed as

$$y(n)=h(0)*x(n)+ h(1)*x(n-1)+ h(2)*x(n-2)+ h(3)*x(n-3)+ h(4)*x(n-4)+ h(5)*x(n-5)+ h(6)*x(n-6)+ h(7)*x(n-7)$$

The coefficients mentioned before are substituted into (3). And we get can get (4) as shown in the following.

$$y(n)=(2^{-1}-2^{-3}+2^{-10})*x(n)+ (2^{-2}+2^{-4}-2^{-8})*x(n-1)+ (-2^{-3}-2^{-11})*x(n-2)+ (2^{-4}+2^{-9})*x(n-3)+ (2^{-2}-2^{-5})*x(n-4)+ (2^{-6}+2^{-10}+2^{-12})*x(n-5)+ (-2^{-8}-2^{-12})*x(n-6)+ (2^{-6}+2^{-9})*x(n-7)$$

Obviously, all the inputs should be multiplied by a sequence of numbers that are specific powers of 2, what's more, some inputs should be multiplied by the same number, for instance, both x(n) and x(n-2) have to multiply by -2-3, so we can get all the inputs that have to multiply by the same number together. In this way, we can get (5), which is shown as follows.

$$y(n)=2^{-1}*x(n)+ 2^{-2}[x(n-1)+x(n-4)]+ 2^{-3}*[-x(n)-x(n-2)]+ 2^{-4}*[x(n-1)+x(n-3)] - 2^{-5}*x(n-4)+ 2^{-6}*[x(n-5)+x(n-7)]-2^{-8}*x(n-1) +2^{-9}*[x(n-3)+x(n-7)]+2^{-10}*[x(n)+x(n-5)] - 2^{-11}*x(n-2)+2^{-12}*[x(n-5)-x(n-6)]$$

Whereas, (5) can be also expressed as

$$y(n)=2^{-1}*(\ldots(2^{-1}*(2^{-1}*U_1+U_2)+U_3)+\ldots.+U_{12})$$

Where $U_1=x(n-5)-x(n-6)$; $U_2=-x(n-2)$;
$U_3=x(n)+x(n-5)$; $U_4=x(n-3)-x(n-7)$;
$U_5=-x(n-1)$; $U_6=0$;
$U_7=x(n-5)+x(n-7)$; $U_8=-x(n-4)$;
$U_9=x(n-1)+x(n-3)$; $U_{10}=-x(n)+(-x(n-2))$;
$U_{11}=x(n-1)+x(n-4)$; $U_{12}=x(n)$;

Then, we calculate U1 by adding x(n-5) to -x(n-6), and then multiply U1 by 2-1,which means shift to the right by one bit, that is easy for hardware implementation. And then add the results to U2, shift to the right again, add to U3 which is calculated by adding x(n) to x(n-5), and so on, when we complete all the operations in (6), we can get y(n), the result of FIR filter. In this way, we can get the filter results by addition and shifting, rather than multiplication. Therefore, there is no need to use multipliers to implement FIR filters at all. All we need are an adder and a shifter. The hardware cost can be reduced to a minimum in this way.

For a conclusion, firstly, present all the coefficients in an N-tap FIR filter's expansion in a binary representation as shown in (4), then gather all the inputs that are supposed to multiply by the same number together, and (5) can be get, according to which we can calculate the filter results by addition and shifting. In this way, all we need are an adder and a shifter in the proposed FIR structures. Substituting multipliers and adder with shifter has advantages of silicon area and power consumption. In addition, the increased shifter stay fixed when the length of the FIR filters increase. This is the most hardware efficient way to implement FIR filters, which makes it very suitable to be used in sensor node ASICs.

## IV. COMPLEXITY ANALYSIS AND COMPARISON

As mentioned before, to implement an N-tap FIR filter, the number of required multiplier for the proposed FIR algorithm is zero, the number of required adders is one, at the expense of one additional shifter. Multipliers are no longer required in the proposed FIR structures. A comparison between the proposed and the traditional FIR algorithm with different length is summarized in Table I. As can be seen from the Table I, large amount of hardware resources can be saved if using the proposed FIR structure. For example, for an 8-tap FIR filter, 8 ultipliers and 6 adders can be saved at the expense of one additional shifter, for a 24-tap FIR filter, 24 multipliers and 22 adders can be saved at the expense of one shifter still, and for a 72-tap FIR filter, 72 multipliers and 70 adders can be saved at the same price. Since shifters weigh less than multipliers and adders in terms of silicon area, exchanging multipliers and adders with shifters is advantageous. In addition, the overhead from the additional shifters stay fixed and do not increase along with the length of the FIR filter, whereas the number of reduced multipliers and adders increase when the length of FIR filter becomes large. Consequently, the larger the length of the FIR filter is, the more the proposed FIR structures can save from the traditional FIRstructures. We can draw a conclusion that the proposed FIR structures can lead to significant hardware savings from the traditional FIR structures, especially when the length of the filter is large.The proposed FIR structure is very suitable to be used in the design of ASICs for sensor nodes.

## V.IMPLEMENTATION AND EXPERIMENT RESULT

The proposed and the traditional FIR algorithm are implemented in Verilog HDL with filter length of 8 and 24, word length 16-bit and 32-bit, respectively.

Two sets of the ideal low-pass FIR filter coefficients of structures are shown in fig.1 and fig.2 respectively. Fig.1 shows the flow chat of the proposed FIR structures. At first, we wait for new input data, as long as the time exceeds 5 seconds, we consider that there is no more data that need to be filtered and draw to an end. Otherwise, store the data into the data random access memory (RAM), which is used for storing the N latest input data, and then load the coefficients from the coefficients read only memory (ROM), which stores all the operations that need to be done to get the FIR filter results. As can be seen in (6), all we need to do are addition and shifting. As to subtraction, we can represent the data in complement form and add up them. Therefore, the coefficients ROM should store information about what operations to do and which data to deal with, according to that, we can read the data RAM to get the corresponding input data and calculate the FIR filter results. Once the calculation is done, we will wait for the

next input. In short, the data RAM stores the data and the coefficients ROM tells us what to do. Once new data arrive, store it in the data RAM first, and then read coefficients from the coefficients ROM, afterwards read corresponding data from the data RAM according to the coefficients and use the data to calculate the FIR filter results. Once the calculation is done, we complete the filtering and get the filter result. The structure frame is presented in fig. 2. There are five modules in the proposed FIR structures: clock generator, input data reader, filter, coefficient ROM and Data RAM. The clock generator module generates the clock and reset signal. The input data reader module provides the input data. As mentioned before, the coefficient ROM module stores the coefficients and the data RAM module stores the data. The filter module is for calculating and getting the filter result. To get the filter result, we need some information from the coefficient ROM module and the data RAM module: coef_men_address, coef_men_shift,

coef_men_sign and data_men_in. The first information is easy to understand, coef_men_address means the memory address of the coefficient being processed now. The coef_men_shift information means there is a need to shift to the right by one bit now, that is to say, we should multiply the current result by 2-1. The coef_men_sign information means we are dealing with a subtraction problem. As mentioned before, some special operations are

length 8 and 24 are generated by MATLAB using Remez Exchanging algorithm. The maximum absolute difference (MAD) algorithm introduced in [1] is used for coefficients quantization. To describe the hardware implementation of the proposed FIR structures in detail, the flow chat and the structure frame of the FIR

required when dealing with subtraction problems. Once the input data reader module provided the input data,the continuous operations would be carried out as follows. Firstly calculate the memory address of this new data, enable the data RAM module and store the data into the data RAM according to the address. Secondly enable the coefficient ROM module, read coefficients from the coefficient ROM one by one and the information necessary for getting the filter results are ready. Thirdly enable the data RAM module and read the corresponding data from the data RAM according to the information. Eventually calculate the data according to the information and get the filter results. A detailed description of hardware implementation of the proposed FIR structures is presented above. The experiment results are given in the following.

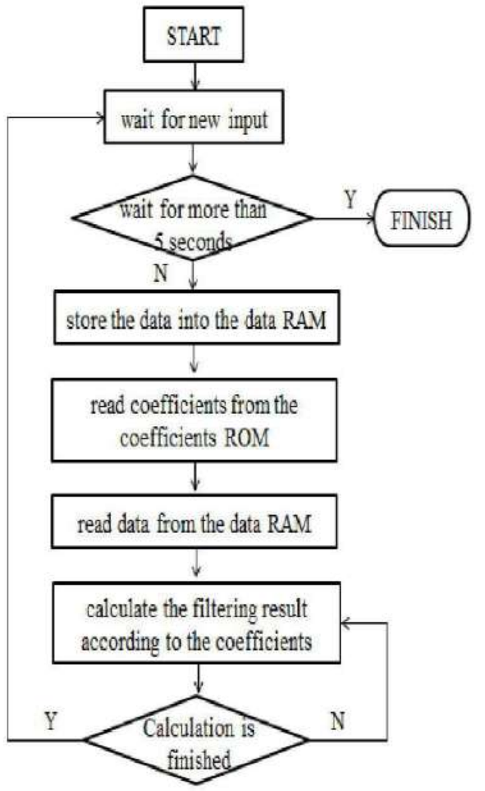
START
wait for new input
wait for more than 5 seconds
Y
FINISH
N
store the data into the data RAM
read coefficients from the coefficients ROM
read data from the data RAM
calculate the filtering result according to the coefficients
Y
Calculation is finished
N

Figure 1.

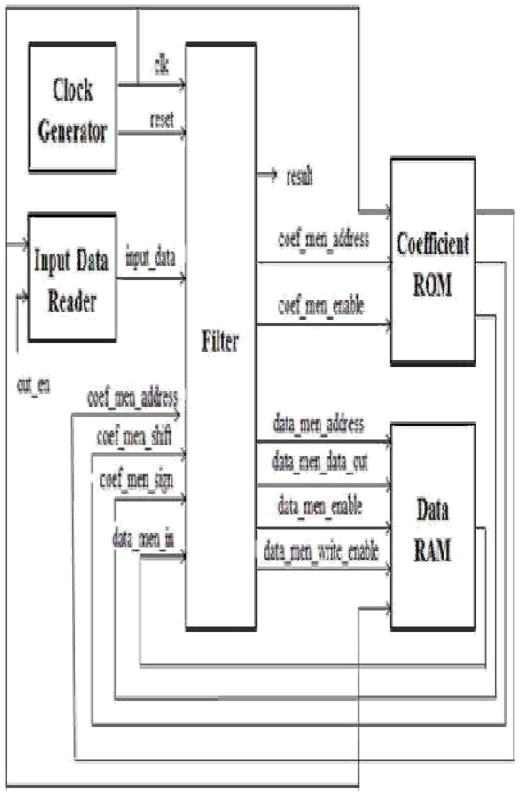

Table II and Table III give the comparison results of area and power between the proposed and the traditional FIR filter structures. The area results are synthesized by Synplify, and the target device is Altera CYCLONE II EP2C5TI144C8. The power comparison results are got by PowerPlay. From Table II and Table III, we can see that the proposed FIR structures can lead to significant savings from the traditional structures both in hardware cost and energy consumption. With the increase of the filter's length, the advantages of the proposed structures become more obvious. For example, for an 8-tap filter, 161 logical elements can be saved from the traditional structures. However, 427 logical elements can be saved for a 24-tap filter. As for energy consumption, for an 8-tap filter, 2.67micro-volt can be reduced, and 16.88micro-volt can be reduced for a 24-tap filter. The experiment results have shown that the proposed FIR algorithm is very efficient in reducing hardware cost and energy consumption, especially when the length of the FIR filter is large. The proposed FIR structures can do a better job than the traditional structure in reducing hardware cost. Therefore, the proposed FIR structure is a better choice for sensor node ASICs.

**TABLE I. COMPARSION OF PROPOSED AND TH E TRADITIANAL FIR ALGORITHM. NUMBER OF REQUIRED MULTIPLIERS (M.), REDUCED MULTIPILERS (R.M.), NUMBER OF REQUIRED ADDER (A.), REDUCED ADDERS (R.A.), AND NUMBER OF INCREASED SHIFTERS (I.S.)**

| Length | Algorithm | M | R.M | A | R.A | I.S |
|---|---|---|---|---|---|---|
| 8-tap | Traditional | 8 | 8 | 7 | 6 | 1 |
| | Proposed | 0 | | 1 | | |
| 24-tap | Traditional | 24 | 24 | 23 | 22 | |
| | Proposed | 0 | | 1 | | |
| 72-tap | Traditional | 72 | 72 | 71 | 70 | |
| | Proposed | 0 | | 1 | | |

**TABLE II. COMPARISON OF AREA(LOGICAL ELEMENTS)**

| Length | Algorithm | Area |
|---|---|---|
| 8-tap | Traditional | 325 |
| | Proposed | 164 |
| 24-tap | Traditional | 592 |
| | Proposed | 165 |

**TABLE III. COMPARISON OF POWER(MV)**

| Length | Algorithm | Area |
|---|---|---|
| 8-tap | Traditional | 10.01 |
| | Proposed | 7.34 |
| 24-tap | Traditional | 25.14 |
| | Proposed | 8.26 |

## VI. CONCLUSION

In this paper, we have presented new FIR filter structures for wireless sensor networks, which can reduce the hardware cost and power consumption to a minimum. Multipliers are the major portions in hardware consumption for FIR filters implementation. However, multipliers are no longer required in the proposed FIR structures, all we need are an adder and a shifter, which means a significant amount of multipliers and adders can be saved. Since multipliers and adders outweigh shifters in hardware cost, it's profitable to substitute multipliers and adders with shifters. Moreover, the number of reduced multipliers and adders increases along with the length of the FIR filter, whereas the increased shifter stays still (which is 1) when the length becomes large. In consequence, the larger the length of FIR filters is, the more the proposed structures can save from the traditional FIR structures, with respect to the hardware cost and energy consumption. Overall, in this paper, we have provided new FIR structures better than the traditional FIR structures in terms of hardware cost and energy consumption, which makes it a better choice for the design of ASICs for sensor nodes.

## REFERENCES

[1]C. Cheng and K. K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution," IEEE Trans. Circuits Syst. I, Reg. Paper, vol.51, no.8, pp. 1492-1500, Aug. 2004.

[2] J. G. Chung and K. K. Parhi, “Frequency-spectrum-based low-area lowpower parallel FIR filter design,”EUROSIP J. A ppl. Signal Process., vol. 2012, no. 9, pp, 444-453,2002

[3] 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..

[4] C. Cheng and K. K. Parhi, “Further complexity reduction of parallel FIR filters,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS 2005), Kobe, Japan, May 2005.

[5] C. 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. Tsao and K. Choi, “Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm,” IEEE Trans. VLSI Syst., vol., 20, no.2, pp. 366-371, Feb. 2012

[7] Y. Wang and K. Roy, “CSDC: A new complexity reduction technique for multiplierless implementation of digital FIR filters,” IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 9, pp. 1845-1853, Sep. 2005.

[8]H. Lee, J. Chung, and G. Sobelman, “FPGA-based digit-serial CSD FIR filter for image signal format conversion,” in Proc. Int. Conf. Signal Processing Application Technology (ICSPAT’98), Toronto, ON. Canada, Sept 1998.

[9] M. Martinez-Peiro, E. Boemo, and L. Wanhammer, “Design of highspeed multiplierless filters using a nonrecursive signed common subexpression algorithm,” IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process., vol. 49, no. 3, pp. 196-203, Mar. 2002.

[10] K. Muhammad and K. Roy, “A graph theoretic approach for synthesizing very low-complexity high-speed digital filters,” IEEE Trans. Comp.-Aided Des. Integr. Circuits, vol. 21, no. 2, pp. 204-216, Feb. 2002. 201