International Journal of Modern Trends in Engineering and Research
www.ijmter.com
e-ISSN No.:2349-9745, Date: 2-4 July, 2015

A Review on VLSI Implementation of Multiplierless FIR Filter Based On Distributed Arithmetic

Komal Lineswala[1],Nishit Vankawala[2], Prof. Rakesh Gajre[3]

[1]E.C (ICT), C.G.P.I.T UkaTarsadia University, komallineswala@gmail.com
[2]E.C (ICT), C.G.P.I.T UkaTarsadia University,nishitvankawala@gmail.com
[3]E.C (ICT), C.G.P.I.T UkaTarsadia University, rakesh.gajre@utu.ac.in

Abstract—The main target for any design is to implement a digital system that has high speed, low power consumption and has low hardware usage and memory requirement. So, a detailed review on one of the high speed and area efficient multiplierless technique for realizing FIR filter based on Distributed Arithmetic (DA) is presented. Various architectures for implementing high order FIR filter based on DA which makes minimum usage of hardware. And also architecture for high speed FIR filter is reviewed which attains maximum speed at a cost of some hardware. However, using DA results up to 50% reduction in total occupied area compared to direct FIR filter implementation.

Keywords- FIR Filter; Distributed Arithmetic (DA); DA-Offset Binary Code (DA-OBC); LUT less DA; FPGA.

I. INTRODUCTION

Over past several decades, the field of digital signal processing (DSP) has been developed in an area of science and technology. This development is due to significant advances in digital computer technology and integrated circuit fabrication. The rapid development in IC technology, starting with medium scale integration (MSI), and progressing to large scale integration (LSI), and then to very large scale integration (VLSI) of electronic circuits has spurred the development of powerful, smaller, faster, and cheaper digital computers and special purpose digital hardware. Thus, Digital signal processing (DSP) has created a major impact in the areas of digital communication, speech and image processing, adaptive filtering applications, satellite communication, wired and wireless communication, multimedia systems, biomedical instrumentation [1].

FPGA are on the verge of revolutionizing DSP in the manner that digital signal processors did several years ago. Many DSP algorithms that were built with ASICs or DSP processors are now replaced by FPGA. However, FPGA have many features in common with ASICs, such as reduction in size, weight, and power dissipation, higher throughput, better security against unauthorized copies, reduced device and inventory cost, and its advantages over ASICs are, such as reduction in development time, in-circuit reprogrammablity and lower NRE costs. Compared to PDSPs, FPGA design typically exploits parallelism [2].

1.1. FIR Filters

One of the most significant components in many DSP systems are digital filters. Digital filter is a LTI system used to perform spectral shaping or frequency selective filtering such as removing undesirable noise from desired signals, etc. Finite Impulse response (FIR) filter and Infinite impulse response (IIR) filter are the two types of digital filters. FIR filter is mostly used in DSP system due to its linear phase response, simple implementation and stability. FIR filter is one, whose impulse response is of finite duration and eventually reaches to zero, i.e., it has finite numbers of non-zero terms. As there is no feedback of past output to form present output, it's also called feed-forward network. The general difference equation for a FIR digital filter is:

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

where, y(n) is the filter output at discrete time instance n, h(k) is the k^{th} feed forward tap, or filter coefficient, and x(n-k) is the filter input delayed by k samples. The Σ denotes summation from $k = 0$ to N-1 where n is the number of feed forward taps in the FIR filter.

The direct implementation of N tap FIR filter requires N Multiply and Accumulate (MAC) blocks. Here to compute the output of filter, convolution of impulse response and input sequence is taken which means extensive use of multiplication operations and use of multiplier may become expensive in terms of area and speed.

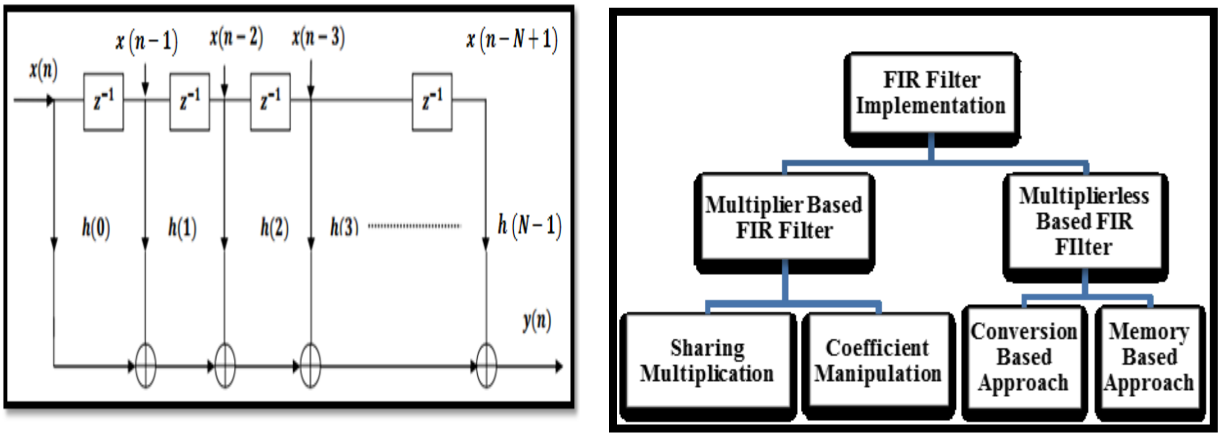


Figure 1. N-Tap FIR Filter and its Classification

1.2 Related Works

As for K-tap FIR filter K MAC blocks are required which not only increases design cost but also increases the complexity. In [3], the concept of Distributed Arithmetic (DA) was introduced to resolve the issues of conventional FIR filter, which is a multiplierless technique and based on 2's complement representation of data and novel bit position reordering. In [4] very first detailed description of DA was done in which a new approach for implementing FIR filter was proposed by storing possible outcomes of intermediate arithmetic operations, and using them to compute the output sample through series of repeated addition and shifting. This eliminated the use of multiplier and resulted in significant savings in terms of hardware cost and power consumption. DA is so named as Distributed Arithmetic because it is not just re-arrangement of adders, multipliers or registers but a new description where fundamental operation of convolution and multiplication are mixed such that arithmetic becomes distributed through the structure [5]. In [6], a review on application of Distributed Arithmetic to Digital Signal Processing was given which stated that DA is bit serial operation that forms an inner product of vector pairs in a single step. Also other modification to DA architecture was also made. The main drawback of DA was also discussed, that is, the size of look up tables (LUT) increases exponentially with the order of filter. In [7], DA Offset Binary Coding (DA-OBC) was proposed which reduced the size of LUT. And in [8] a modified DA architecture was introduced which replaced the use of LUT with multiplexer/adder pair.

This paper is organized as follows: Section II describes theoretical background of Distributed Arithmetic and its basic bit serial architecture. Different architectures for higher order FIR filter are discussed in Section III which describes different techniques to reduce memory requirement. And in Section IV architecture for high speed FIR filter is discussed and performance analysis of all these architecture is presented in Section V.

II. DISTRIBUTED ARITHMETIC

DA is one of the most significant FPGA technology used for implementing FIR filters. An FIR filter of N-length is described as:

$$y[n] = \sum_{k=0}^{N-1} h(k)\, x(n-k)$$

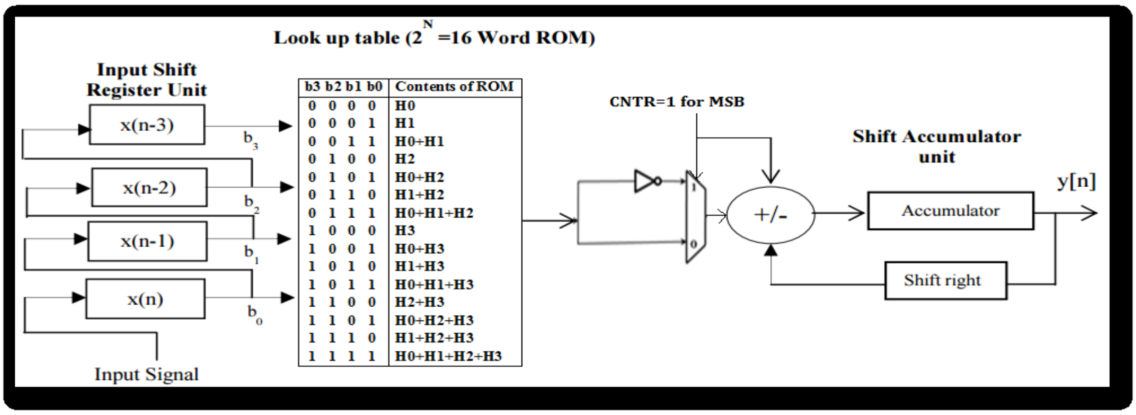
For convenience, let $x'(k) = x(n-k)$

Now, let $x'(k)$ be B-bit 2's complement binary number scaled such that $|x'(k)| < 1$ given by:

$$x'(k) = -2^{B} x'_{B}(k) + \sum_{b=0}^{B-1} x'_{b}(k) 2^{-b}$$

where, $x'_{b}(k)$ denotes bth bit of $x(k)$, $x'_{b}(k)$ € {0, 1}, $x'_{B}(k)$ is the sign bit and $x'_{B-1}(k)$ is the Least Significant Bit (LSB).

The term in the bracket has 2^{b} possible combinations that can be computed online (using RAM) or precomputed so that they can be stored in ROM. The N bit sequence of input can be used as address to the memory and operations such as shifting and addition are performed on the output of LUT or ROM and storing the result in accumulator.



b3	b2	b1	b0	Contents of ROM
0	0	0	0	H0
0	0	0	1	H1
0	0	1	1	H0+H1
0	1	0	0	H2
0	1	0	1	H0+H2
0	1	1	0	H1+H2
0	1	1	1	H0+H1+H2
1	0	0	0	H3
1	0	0	1	H0+H3
1	0	1	0	H1+H3
1	0	1	1	H0+H1+H3
1	1	0	0	H2+H3
1	1	0	1	H0+H2+H3
1	1	1	0	H1+H2+H3
1	1	1	1	H0+H1+H2+H3

Figure 2. 4-Tap LUT Based Distributed Arithmetic [2]

$$y[n] = \sum_{k=0}^{N-1} h(k) \left[-2^{B} x'_{B}(k) + \sum_{b=0}^{B-1} x'_{b}(k) 2^{-b} \right]$$

$$y[n] = -2^{B} \sum_{k=0}^{N-1} h(k) x'_{B}(k) + \sum_{b=0}^{B-1} 2^{-b} \left[\sum_{k=0}^{N-1} x'_{b}(k) h(k) \right]$$

So as to produce one filter output y[n], B clock cycles are required. The basic architecture of DA is shown above. This architecture consists of three main units that are input shift register unit, ROM based LUT unit, and accumulator/shift unit. The LUT contains all possible combination sum of filter coefficients, h(k) such that k=0,1,2,…,N-1.

The input shifter register unit stores the input sample. Firstly the input sample is given to parallel to serial converter which parallely loads the bits of sample into it and outputs the bit serially to the shift register unit. Here the shift register are arranged in a daisy chain form. Then after at every clock cycle, the bits from all shift register are shifted out starting from LSB. These bits from all shift registers are concatenated to form address line to select corresponding combination of coefficient sum. For signed DA scheme, the MSB is used to distinguish between positive and negative numbers. CNTR is the sign bit timing signal. In a clock period known as "sign bit time" all the MSB's, that is, sign bits from shift registers arrive simultaneously and during this sign bit time, CNTR=1 otherwise CNTR=0.

III. DA ARCHITECTURE FOR HIGHER ORDER FIR FILTERS

3.1. Memory Reduced DA.

But the main drawback with the above architecture is that as the order of filter N increases the size of LUT (2^N-words) grows exponentially. So the reduction of LUT or ROM size is of prime importance. This increased ROM size problem can be reduced by using two methods: ROM Decomposition and special coding of ROM content that is, using Offset Binary Coding (OBC).

3.1.1. ROM Decomposition.

In this method [9], memory size can be reduced by sub-dividing LUT's into number of LUT's called LUT partition and then by using adder, all ROM outputs are added. Thus, by using this method, memory size can be reduced from 2^N words to P x 2^M words.

3.1.2. DA-Offset Binary Code (DA-OBC).

Jung Pil et al. [7] proposed a method for ROM size reduction in which input is not represented in binary (0, 1) but instead represented in offset binary code, that is, in (-1, 1).

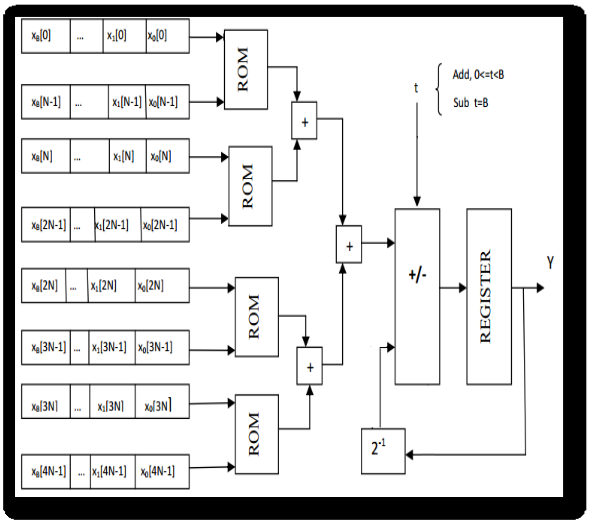
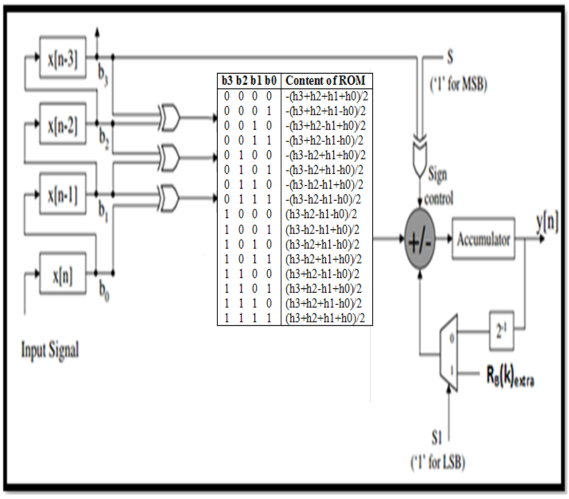

$x_B[0]$ ... $x_1[0]$ $x_0[0]$
$x_B[N-1]$ ... $x_1[N-1]$ $x_0[N-1]$
$x_B[N]$ ... $x_1[N]$ $x_0[N]$
$x_B[2N-1]$ ... $x_1[2N-1]$ $x_0[2N-1]$
$x_B[2N]$ ... $x_1[2N]$ $x_0[2N]$
$x_B[3N-1]$ ... $x_1[3N-1]$ $x_0[3N-1]$
$x_B[3N]$ ... $x_1[3N]$ $x_0[3N]$
$x_B[4N-1]$ ... $x_1[4N-1]$ $x_0[4N-1]$
ROM
ROM
ROM
ROM
+
+
+
t
Add, 0<=t<B
Sub t=B
+/-
REGISTER
Y
2^{-1}

Figure 3. N-Tap LUT or ROM Decomposition [9]



b3	b2	b1	b0	Content of ROM
0	0	0	0	-(h3+h2+h1+h0)/2
0	0	0	1	-(h3+h2+h1-h0)/2
0	0	1	0	-(h3+h2-h1+h0)/2
0	0	1	1	-(h3+h2-h1-h0)/2
0	1	0	0	-(h3-h2+h1+h0)/2
0	1	0	1	-(h3-h2+h1-h0)/2
0	1	1	0	-(h3-h2-h1+h0)/2
0	1	1	1	-(h3-h2-h1-h0)/2
1	0	0	0	(h3-h2-h1-h0)/2
1	0	0	1	(h3-h2-h1+h0)/2
1	0	1	0	(h3-h2+h1-h0)/2
1	0	1	1	(h3-h2+h1+h0)/2
1	1	0	0	(h3+h2-h1-h0)/2
1	1	0	1	(h3+h2-h1+h0)/2
1	1	1	0	(h3+h2+h1-h0)/2
1	1	1	1	(h3+h2+h1+h0)/2

Figure 4. 4-Tap DA-OBC[7]

Rewriting $x'(k)$ as:

$$x'(k) = \frac{1}{2}\left[x'(k) - (-x'(k))\right]$$

In 2's complement representation, negative of $x'(k)$ is written as:

$$-x'(k) = -x'_B(k) + \sum_{b=0}^{B-1} \overline{x'_b(k)} 2^{-b} + 2^{-(B-1)}$$

$$x'(k) = \frac{1}{2}\left[-\left(x'_B(k) - \overline{x'_B(k)}\right)\right] + \frac{1}{2}\left[\sum_{b=0}^{B-1}\left(x'_b(k) - \overline{x'_b(k)}\right)2^{-b} - 2^{-(B-1)}\right]$$

$$x'(k) = \frac{1}{2}\left[\sum_{b=0}^{B} c_b(k) 2^{-b} - 2^{-(B-1)}\right]$$

where,

$$c_b(k) = x'_b(k) - \overline{x'_b(k)}, \quad b \neq 0$$

$$c_B(k) = -\left(x'_B(k) - \overline{x'_B(k)}\right)$$

$$y[n] = \frac{1}{2}\sum_{k=0}^{N-1} h(k)\left[\sum_{b=0}^{B} c_b(k) 2^{-b} - 2^{-(B-1)}\right]$$

Let, $R_b(k) = \sum_{k=0}^{N-1} \frac{h(k)}{2} c_b(k), \quad \textit{for } 0 \leq b \leq B$

$$R_B(k)_{extra} = -\sum_{k=0}^{N-1} \frac{h(k)}{2}$$

Thus, $y[n] = \sum_{b=0}^{B} R_b(k) 2^{-b} + 2^{-(B-1)} R_b(k)_{extra}$

By using DA-OBC method, ROM size reduces to 2^{N-1}. In this architecture, XOR gates are used as address decoder to select corresponding coefficient sum. However, two control signals, S and S1, are used where S=1 when b=B otherwise S=0 and S1=1 when b=0 otherwise S1=0.

3.2. LUT less DA Architecture.

H.Yoo et al. [8] noticed from LUT in figure 1 that values in lower half of LUT (whose MSB=1) are mirror image of the values of upper half of LUT (whose MSB=0) and h(3) term. Therefore, LUT or ROM size can be reduced by factor of 2, that is, from 2^N words to 2^{N-1} words with a 2:1 MUX and an adder. By performing same LUT reduction procedure iteratively, LUT size can be reduced dramatically by replacing LUT with numbers of 2:1 MUX and full adders. For N-tap FIR filter, (N-1) numbers of full adders are required where number of MUXrequired is equal to number of coefficients.

3.3. Performance Analysis.

H. Yoo et al. [8] made a performance analysis related to area occupied by single DA base unit. However, the input shift register unit and adder/shifter units are not considered in analysis since they are common for all structures.

B_c represent word lengths of original LUT and k represents base unit size. It was noticed that LUT less version required fewer logic elements (LE) and memory compared to original LUT based DA.

IV. DA ARCHITECTURE FOR HIGH SPEED FIR FILTERS

If FIR filter is implemented using bit serial DA architecture then it will consume B clock cycles to compute one output. When 1 bit is processed at a time, DA is called 1BAAT-DA (serial DA) and when 2 bits are processed at a time, then DA is called 2BAAT-DA (parallel DA) and so on. FIR filter can also be implemented using parallel version of DA architecture. With this architecture, only one clock cycle is needed to compute one output. If FIR filter is implemented using bit serial DA architecture then it will consume B clock cycles to compute one output. When 1 bit is processed at a time, DA is called 1BAAT-DA (serial DA) and when 2 bits are processed at a time, then DA is called 2BAAT-DA (parallel DA) and so on. FIR filter can also be implemented using parallel version of DA architecture. With this architecture, only one clock cycle is needed to compute one output. The main advantage of using this architecture is that it can high speed compared to serial DA.
However, this architecture has twice throughput compared to serial DA, but number of LUT required is also twice that used in serial DA. So, resource usage is B times more than that used in serial DA.

Table 1. Performance Analysis[8]

Logic Functions	LUT Based DA (Fig. 2)	DA-OBC (Fig. 4)	LUTless DA (Fig. 5)
ROM Decoder	C(1,k)	C(2,k)	0
ROM Data	D(1,k,Bc)	D(2,k,Bc)	0
XOR	0	8k	0
2:1 MUX	0	6Bc	6k x Bc
Register	0	16Bc	0
Adder	0	0	(k-1) x 30Bc
Adder/Sub, Cin=0	0	0	0
Adder/Sub, Cin=1	0	0	0
Adder/Sub	0	0	0

V. CONCLUSION

Much architecture for higher order and high speed FIR filter implementation were reviewed. Thus from the analysis, hardware reduction of LUT less version occurs around 6 to 8 taps and thereafter reduction rate grows significantly as filter size increases. And higher speed can be achieved at a little hardware cost.

REFERENCES

[1]J.G.Proakis, D.G.Manolakis,Digital Signal Processing:Principles, Algorithms and Applications, NJ:Prentice Hall, 1996.

[2]U.Meyer-Baese,Digital Signal Processing with Field Programmable Gate Arrays, Springer, 2007.

[3] A.Croisier, D.J.Esteban, M.E.Levilion, and V.Rizo, "Digital Filter for PCM Encoded Signals," U.S. Patent 3777130, December 4,1973.

[4] A.Peled, and B.Liu, "A New Hardware Realization of Digital Filters," IEEE Transactions on Acoustics, Speech Signal Processing, pp. 456-462, December 1974.

[5] C.Sidney Burrus, "Digital Filter Strucutres Described by Distributed Arithmetic," IEEE Transactions on Circuits and Systems, pp. 674-680, December 1977.

[6] S.A.White, "Applications of Distributed Arithmetic to Digital Signal Processing. A Tutorial Review," IEEE ASSP Magazine, pp.4-19, July 1989.

[7] J.P.Choi, S.Shin, and J.G.Chung, "Efficient ROM Size Reduction for Distrubuted Aruthmetic," IEEE Symposium on Circuits and Systems (ISCAS), pp. 61-64, May 2000.

[8] H.Yoo, and D.V.Anderson, "Hardware Efficient Distributed Arithmetic for High Order Digital Filters," Proc. IEEE International Conference on Acoustics, Speech Signal Processing (ICASSP), pp. 125-128, March 2005.