International Journal of Research in Computer Science
eISSN 2249-8265 Volume 4 Issue 2 (2014) pp. 19-24
www.ijorcs.org, A Unit of White Globe Publications
doi: 10.7815/ijorcs.42.2014.081

FPGA IMPLEMENTATION OF FIR FILTER USING VARIOUS ALGORITHMS: A RETROSPECTIVE

Jinalkumari K. Dhobi[1], Dr. Y. B. Shukla[2], Dr. K.R.Bhatt[3]
[1]*P.G. Student, EC Dept., SVIT, Vasad*
Email: jinal.dhobi@gmail.com
[2]*Assoc. Prof., EC Dept., SVIT, Vasad*
Email: ybshukla2003@gmail.com
Assoc. Prof., EC Dept., SVIT, Vasad
Email: krbhattec@gmail.com

Abstract: *This manuscript is a thorough study of FPGA implementation of Finite Impulse response (FIR) with low cost and high performance. The key observation of this paper is an elaborate analysis about hardware implementations of FIR filters using different algorithm i.e., Distributed Arithmetic (DA), DA-Offset Binary Coding (DA-OBC), Common Sub-expression Elimination (CSE) and sum-of-power-of-two (SOPOT) with less resources and without affecting the performance of the original FIR Filter.*

Keywords*: DA, DA-OBC, CSE, SOPOT, FPGA*

I. INTRODUCTION

A digital filter is a system that carry out mathematical operation on a sampled of discrete time signal to modify or alter the component of the signal in time or frequency domain. It consist of an analog-to-digital converter at front end, followed by a microprocessor and some peripheral component i.e., memory to store data & filter coefficients. At the backend side a digital-to-analog converter is used to complete the output stage. For a real time applications, an FPGA or ASIC or Specialized DSP with parallel architecture is used instead of a general purpose microprocessor. Digital filters can be implemented in two ways, by convolution (FIR) and by recursion (IIR) [1]. A FIR filter has a number of useful properties compared to an IIR filter i.e. inherently stable, no feedback require, designed to be linear phase. With recent trend towards portable computing and wireless communication systems, power consumption has been an important design consideration [2] so the field programmable gate array (FPGA) is an alternative solution for realization of digital signal processing task.

This paper is organized as follows: section II introduced design method of the FIR filter, section III describes the various algorithms to implement the FIR filter on FPGA, Section IV discusses the Comparison of algorithms and finally the conclusion is presented in section V.

II. DESIGN METHODS

FIR filters are used where exact linear phase response is required. It is non-recursive filter, consists of two parts one is Approximation problem and second one is Realization problem. The steps of approximation are, first take the Ideal frequency response after that choose the Class of filter & Quality of approximation and finally select the Method to find the filter transfer function. The realization part select the structure to implement the transfer function. There are mainly three well-known methods for designing FIR filter namely the window method, Frequency sampling technique and Optimal filter design method. Among these three methods, window method is simple and efficient way to design an FIR filter [3]. In this method, first start with the ideal desired frequency response $H_d(e^{jw})$ of the specified filter then Compute the inverse DTFT of $H_d(e^{jw})$ i.e., $h_d(n)$ which is infinite in duration after that Choose an appropriate window function $w(n)$ and calculate the impulse response $h(n)$ of specified filter as $h(n) = h_d(n) * w(n)$ to truncated at some point n=M-1. Once $h(n)$ is determined, it's DTFT $H(e^{jw})$ and Z-transform $H(z)$ can be calculated for any further analysis. For truncation in third step of $h_d(n)$ to M-terms, direct truncation method using rectangular window $w(n)$ is multiplied with $h_d(n)$ which is infinite in nature but rectangular window contain sharp discontinuity which leads to Gibbs Phenomenon effect [4]. In order to reduce the ripples, instead of sharp discontinuity function, choose a window function having taper and decays toward zero gradually [4]. Some of window [5] commonly used are as followed:

1. Bartlett Triangular window:

$$w(n) = \begin{cases} \frac{2(n+1)}{N+1}, n = 0 \; to \; (N-1)/2 \\ 2 - \frac{2(n+1)}{N+1}, n = (N-1)/2 \; to \; (N-1) \\ 0 \qquad Otherwise \end{cases} \quad \ldots.(1)$$

2. Generalized cosine windows:

(Rectangular, Hanning, Hamming and Blackman)

$$w(n) = a - b\cos\left(\frac{2p(n+1)}{N+1}\right) + c\cos\left(\frac{4p(n+1)}{N+1}\right)$$

Where n = 0 to N-1 .... (2)

3. Kaiser window with parameter β:

$$w(n) = \frac{I_0(\beta\sqrt{1-\left(1-\frac{2(n)}{N-1}\right)^2}}{I_0(\beta)}$$

Where n= 0 to N-1 .... (3)

$$M = \frac{\delta_2 - 7.95}{2.286\Delta\omega} \quad \text{.... (4)}$$

$$\beta = \begin{cases} 0.1102(\delta_2 - 8.7), & \delta_2 \geq 50dB \\ 0.5842(\delta_2 - 21)^{0.4} + 0.07886(\delta_2 - 21), & 21dB < \delta_2 < 50dB \\ 0, & \delta_2 \leq 21dB \end{cases} \quad \text{.... (5)}$$

In Rectangular window, due to the direct truncation of $h_d(n)$ leads to the Gibbs phenomenon effect which apparent itself as a fixed percentage overshoot and ripple before and after an approximated discontinuity in the frequency response due to the non-uniform convergence of the Fourier series at the discontinuity [5]. According to the simulation result [5] obtained from FDAtool, the Bartlett window reduced the overshoot in the designed filter but spreads the transition region. The generalized cosine window, Hanning, Hamming and Blackman is complicated but provide a smooth truncation of ideal impulse response and a frequency response. The best window method is Kaiser Window because it has the shape parameter β which depending on the filter taps M was used to adjust the main lobe width and side lobe attenuation, choosing M can produce a variety of transition band and optimal stopband attenuation [6]. When the transition band Δω(rad) and the stopband attenuation $\delta_2 = -20log_{10}\alpha 2(dB)$ were given, the Kaiser FIR filter taps M and the shape parameter β can be obtained from equation (4) and (5) [6].

III. FPGA IMPLEMENTATION USING VARIOUS ALGORITHMS

The application of digital filter are widespread and include but are not limited to the Communication System, Audio System, Instrumentation, Image-Processing and enhancement, Speech synthesis. It is nowadays convenient to consider computer programs and digital hardware that can perform digital filtering as two different implementations of digital filters, namely, Software digital filters, and Hardware digital filters. Software digital filters can be implemented in terms of a high-level language, such as C++ or MATLAB, on a personal computer or workstation or by using a low-level language on a general-purpose digital signal processing chip. Hardware digital filter can be designed using a number of highly specialized interconnected VLSI chips like DSP, ASIC and FPGA. Software digital filter have no counterpart in the analog world and therefore, for non-real-time application they are the only choice. ASIC based Implementation of FIR Filter provides little costlier and long development time but gives high flexibility. The realization of FIR filter based on FPGA received extensive attention because it gives high flexibility, high performance, low cost and shorter development time [8]. The core of the FIR filter implementation is multiplication and accumulation (MAC) operation. The design method of MAC can be define using various algorithm and techniques. One is general direct MAC structure, which are expensive in hardware because of logic complexity and area usage. The others are Distributed Arithmetic (DA) [7], DA-Offset Binary Coding (DA-OBC) [9], Common Subexpression Elimination (CSE) [10] and Sum-of-power-of-two (SOPOT) [11] which reduced the required number of multiplier and adders.

A. Distributed arithmetic and DA-OBC

Distributed Arithmetic is a famous method, which converts calculation of MAC to a serial of look up table accesses and summation [8]. It was initially proposed by Croisier in 1973 [7] and further developed by Peled and Lui [8].The principle of DA algorithm is as follows:

An FIR filter of N order is shown as Eq. (6)

$$y(n) = \sum_{k=0}^{N-1} h_k x_k(n) \quad \text{.... (6)}$$

Where h_k the set of constant coefficient of the filter is, $y(n)$ is the output data, $x_k(n)$ is the input data, which can be expressed as Eq. (7) using m-bit 2's complementary binary number.

$$x_k = -x_{k,m-1} + \sum_{j=1}^{m-1} x_{k,m-1-j}\, 2^{-j} \quad \text{.... (7)}$$

Where $x_{k,m-1}$ is the most significant bit of x_k, Eq. (8) can be derived when x_k of Eq. (7) is substitute into Eq. (6).

$$y(n) = \sum_{k=0}^{N-1} h_k[-x_{k,m-1} + \sum_{j=1}^{m-1} x_{k,m-1-j}\, 2^{-j}]$$

$$= \sum_{k=0}^{N-1}\sum_{j=1}^{m-1} h_k\, x_{k,m-1-j}\, 2^{-j} - \sum_{k=0}^{N-1} h_k x_{k,m-1}$$

$$= \sum_{j=1}^{m-1} \sum_{k=0}^{N-1} h_k \, x_{k,m-1-j} \, 2^{-j} - \sum_{k=0}^{N-1} h_k x_{k,m-1} \quad \ldots. (8)$$

In Eq. (8) the calculation result of $\sum_{k=0}^{N-1} h_k \, x_{k,m-1-j}$ have 2^N kinds of different results because of the value of $x_{k,m-1-j}$ is 0 or 1. All the possible results of $\sum_{k=0}^{N-1} h_k \, x_{k,m-1-j}$ are constructed in advance and stored into one LUT. The contents of LUT can be fetch by using $x_{k,m-1-j}$ obtain from shift register unit as a LUT address signal. So the value of $y(n)$ calculated by shifting and accumulating operation.

Table 1: LUT contents of DA for 4-tap FIR filter

Address signal ($b_3 b_2 b_1 b_0$)	LUT content
0000	0
0001	h_0
0010	h_1
0011	$h_1 + h_0$
0100	h_2
0101	$h_2 + h_0$
0110	$h_2 + h_1$
0111	$h_2 + h_1 + h_0$
1000	h_3
1001	$h_3 + h_0$
1010	$h_3 + h_1$
1011	$h_3 + h_1 + h_0$
1100	$h_3 + h_2$
1101	$h_3 + h_2 + h_0$
1110	$h_3 + h_2 + h_1$
1111	$h_3 + h_2 + h_1 + h_0$

Hence, DA algorithm reduced the logic resources by converting complex MAC operation to a simple look-up table access and summations. The main disadvantage of DA algorithm is the size of LUT is large and also capacity of LUT will exponentially increase with the order of the filter. Bo Hong, et al [9] proposed a new algorithm which can reduced the capacity of LUT by half through the use of offset binary coding. The principle of DA-OBC algorithm is as follows [12]:

x_k can also be expressed as Eq. (9)

$$\begin{aligned} x_k &= \frac{1}{2}[x_k - (-x_k)] \\ &= \frac{1}{2}\left[-\left(x_{k,m-1} - \overline{x_{k,m-1}}\right)\right] \\ &+ \sum_{j=1}^{m-1} [\left(x_{k,m-1-j} - \overline{x_{k,m-1-j}}\right)2^{-j} \\ &- 2^{-(m-1)}] \end{aligned} \quad \ldots. (9)$$

They define $d_{k,j}$ as follows, the value of $d_{k,j}$ is -1 or +1.

$$d_{k,j} = \begin{cases} \left(x_{k,j} - \overline{x_{k,j}}\right), & j \neq m-1 \\ -\left(x_{k,m-1} - \overline{x_{k,m-1}}\right), & j = m-1 \end{cases} \quad \ldots. (10)$$

Eq. (9) can be simplified as Eq. (11).

$$x_k = \frac{1}{2}[\sum_{j=0}^{m-1} d_{k,m-1-j} \, 2^{-j} - 2^{-(m-1)}] \quad \ldots. (11)$$

Put Eq. (11) in to Eq. (6), and get

$$= \sum_{j=0}^{m-1} \left(\sum_{k=0}^{N-1} \frac{1}{2} h_k \, d_{k,m-1-j} \right) 2^{-j} - (\frac{1}{2} \sum_{j=0}^{N-1} h_j) 2^{-m-1} \quad \ldots. (12)$$

For convenience,

$$D_j = \sum_{k=0}^{N-1} \frac{1}{2} h_k \, d_{k,m-1-j} \;, \qquad D_{ex} = -\frac{1}{2} \sum_{j=0}^{N-1} h_j \quad \ldots. (13)$$

The value of D_j have 2^N kinds of different result which can be pre-computed and stored in a LUT and the LUT can be reduced by half [9] because the contents of LUT which are stored in addresses, are 1's complement of each other is in same magnitude with the reverse sign.

Table 2: LUT contents of DA-OBC for 4-tap FIR filter

Address $x_{0j} = 0$ $x_{1j} x_{2j} x_{3j}$	LUT Contents	Address $x_{0j} = 1$ $x_{1j} x_{2j} x_{3j}$	LUT Contents
000	$-\frac{h_0 + h_1 + h_2 + h}{2}$	111	$\frac{h_0 + h_1 + h_2 + h}{2}$
001	$-\frac{h_0 + h_1 + h_2 - h}{2}$	110	$\frac{h_0 + h_1 + h_2 - h}{2}$
010	$-\frac{h_0 + h_1 - h_2 + h}{2}$	101	$\frac{h_0 + h_1 - h_2 + h}{2}$
011	$-\frac{h_0 + h_1 - h_2 - h}{2}$	100	$\frac{h_0 + h_1 - h_2 - h}{2}$
100	$-\frac{h_0 - h_1 + h_2 + h}{2}$	011	$\frac{h_0 - h_1 + h_2 + h}{2}$
101	$-\frac{h_0 - h_1 + h_2 - h}{2}$	010	$\frac{h_0 - h_1 + h_2 - h}{2}$
110	$-\frac{h_0 - h_1 - h_2 + h}{2}$	001	$\frac{h_0 - h_1 - h_2 + h}{2}$
111	$-\frac{h_0 - h_1 - h_2 - h}{2}$	000	$\frac{h_0 - h_1 - h_2 - h}{2}$

B. Common Subexpression Elimination (CSE)

M. Thenmozhi, et al [10] proposed an algorithm which reduce the complexity is Common subexpression elimination (CSE). In this algorithm, the coefficient is based on canonical signed digit (CSD), which minimize the number of adder/subtractor used in each coefficient multiplier. The aim of CSE algorithm is to identify multiple event of identical bit patterns present in coefficients to remove the redundant multiplications which results in considerable reduction of adders as well as the complexity of FIR filter compared to the conventional implementation. This algorithm using binary representation of coefficients for the implementation of higher order FIR filter with less number of adder than CSD based CSE algorithm. The Binary CSE (BCSE) algorithm technique focuses on reducing redundant computations in coefficient multipliers by reusing the most common binary bit patterns (BCSs) presents in coefficients [13]. In n-bit binary number, the number of BCSs is$2^n - (n+1)$.

For example: A 3 bit binary representation can form four BCSs.

$[0\ 1\ 1] = x_2 = 2^{-1}x + 2^{-2}x$
$[1\ 0\ 1] = x_3 = x + 2^{-2}x$
$[1\ 1\ 0] = x_4 = x + 2^{-1}x$
$[1\ 1\ 1] = x_5 = x + 2^{-1}x + 2^{-2}x$

Where x is the input signal. The other BCSs such as $[0\ 0\ 1]$, $[0\ 1\ 0]$ and $[1\ 1\ 0]$ do not required any adder for implementation because they have only one nonzero bit. From above four BCSs they conclude that x_2 can be obtained by right shift operation without using any extra adders and also x_5 can be obtained from x_4 using an adder.

C. Sum-of-power-of-two (SOPOT)

Catalin Damian, et al [11] proposed a high speed and low area architecture for the implementation of FIR filter without any multiplication block.

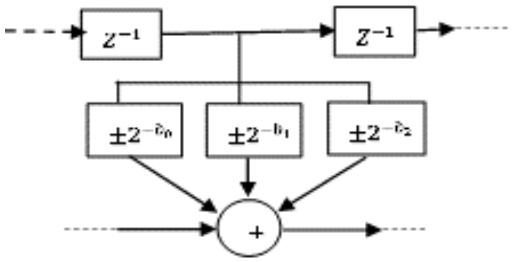

Figure 1: Direct FIR filter

Figure 2. describe the Architecture unit of FIR filter with SOPOT type coefficient. In this algorithm the coefficient values are integer's power-of-two or sum-of- power-of-two (SOPOT) with two or three terms and the multipliers can be replaced by shifters. In [15] and [16], FIR architectures presented based on theory of SOPOT of two terms. Because of error occurred in two terms the approximation of filter's coefficient is extended the algorithm to SOPOT with three terms.

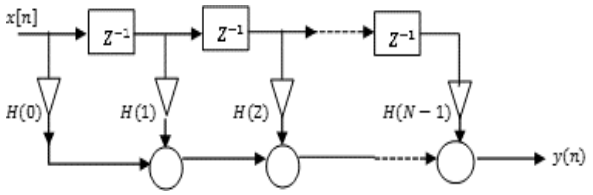

Figure 2. Architecture unit of FIR filter with SOPOT type coefficient

$$H[j] = \sum_{i=0}^{2} a_{i,j} \cdot 2^{b_{i,j}} \quad \text{.... (13)}$$

Where $a_{i,j} = \{-1,1\}$ and$b_{i,j} = \{-t, \dots, 0, \dots, u\}$; t and u determine the word length dynamic range of each filter coefficient. Larger the numbers t and u will gives the closer approximation to its original number.

Computational algorithm: The Computational algorithm is shown in figure 3. The algorithm consist of two repetitive structure; one by i is to calculate a[i,j] and b[i,j] for the f[j] co-efficient.

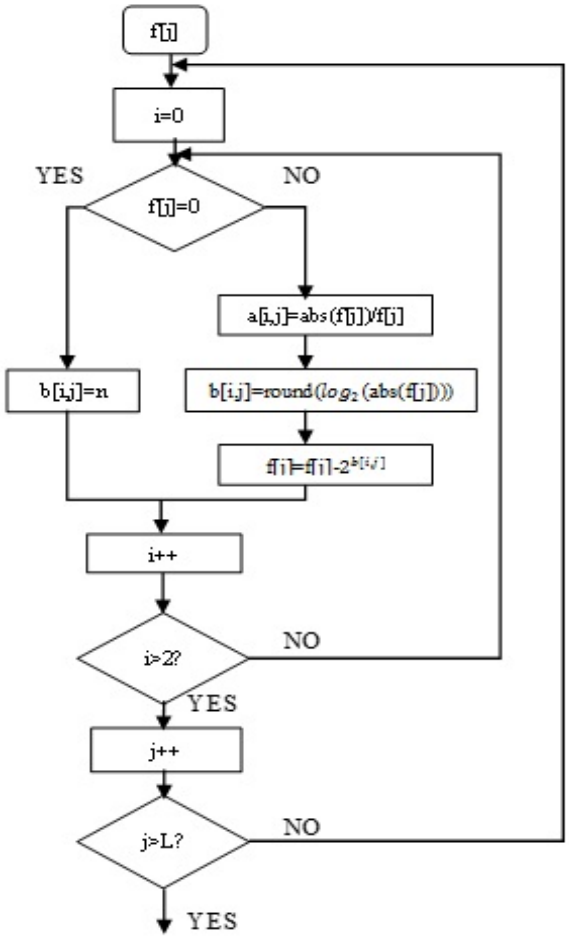

Figure 3: SOPOT $a_{i,k}$ and $b_{i,k}$ terms calculation algorithm

Table 3: Example of Two and Three SOPOT

Integer	Two SOPOT	Three SOPOT
13	$2^3 + 2^2$=12 (8%)	$2^3 + 2^2 + 2^0$=13 (0%)
25	$2^4 + 2^3$=24 (4%)	$2^4 + 2^3 + 2^0$=25 (0%)
67	$2^6 + 2^1$=66 (2%)	$2^6 + 2^1 + 2^0$=67 (0%)

IV. COMPARISION OF ALGORITHMS BASED ON SURVEY

DA algorithm reduces the complexity of FIR filter by converting MAC operation to serial of look up table. CSE algorithm reduces the adder by just finding the multiple event and SOPOT is converting coefficient into power of two format and eliminate the complete Multiplication block. Among the three algorithm DA is easy to implement at the cost of storage resources.

V. CONCLUSION

The realization structure of FIR filter consists of a MAC operation which is made up of multipliers and adders. DA algorithm, it completely eliminate the multiplication block by converting complicated MAC operation to look-up table access and summation at the cost of increasing storage resource where DA-OBC based architecture will decreases the LUT size by half and make operation speed faster. In CSE algorithm, It eliminate the multipliers and adders by identifying the multiple event that have an identical bit pattern and SOPOT algorithm diminished the complexity of working task by completely abolishing the multiplication block by only adders and shifter. So According to our survey, DA structure is easy to implement on FPGA because of pre-calculated results are already stored in LUTs and in FPGA it is easy to design but the main drawback of this algorithm is that, it uses an extra resources. Where the SOPOT is used for low power and low area application as it uses shift/add multiplexer based multiplier. The CSE algorithm is used to find and eliminate most common event among filter co-efficient which results in power and area saving by reducing multiplier with a small number of adders while implemented in FIR filters.

VI. REFERENCES

[1] Steven W. Smith, "The Scientist and Engineer's Guide to Digital Signal Processing". California Technical Publishing San Diego, California, Second Edition 1999.

[2] Wang, Wei, Swamy, M.N.S., Ahmad, M.O., "Low power FIR filter FPGA implementation based on distributed arithmetic and residue number system" *Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001 Midwest Symposium on,* vol.1, no., pp.102, 105 vol.1, 2001. doi: 10.1109/MWSCAS.2001.986125

[3] Saurabh Singh Rajput, Dr.S.S.Bhadauria, "Implementation of Fir Filter Using Efficient Window Function and Its Application in Filtering a Speech Signal" *International Journal of Electronics and Mechanical Controls,* Volume 1 Issue 1 November 2012.

[4] Sen M. Kuo, Bob. H. Lee and Wenshun Tian, "Real-Time Digital Signal Processing Implementations and Applications" John Wiley & Sons, Ltd, England, Second Edition 2006.

[5] Sonika Gupta, Aman Panghal, "Performance Analysis of FIR Filter Design by Using Rectangular, Hanning and Hamming Windows Method," *International Journal of Advanced Research in Computer Science and Software Engineering,* Volume 2, Issue 6, June 2012.

[6] Gao Jinding, Hou Yubao, Su Long, "Design and FPGA Implementation of Linear FIR Low-pass Filter Based on Kaiser Window Function", *International Conference on Intelligent Computation Technology and Automation (ICICTA)*, vol.2, no., pp.496-498, 28-29 March 2011. doi: 10.1109/ICICTA.2011.408

[7] Croisier, D.J.Esteban, M.E.Levilion, V.Rizo, "Digital Filter for PCM Encoded Signals" U.S.Patent, No.3, 777, 130, 1973.

[8] A. Peled and B. Liu, "A new hardware realization of digital filters" *IEEE Transactions on A.S.S.P., vol. ASSP-22,* pp. 456–462, December 1974.

[9] Bo Hong, Haibin Yin, Xiumin Wang, Ying Xiao, "Implementation of FIR filter on FPGA using DA-OBC algorithm", *2nd International Conference on Information Science and Engineering (ICISE)*, vol., no., pp.3761,3764, 4-6 Dec. 2010.

[10] M. Thenmozhi, N. Kirthika, "Analysis of Efficient Architectures for FIR filters using Common Subexpression Elimination Algorithm," *International Journal of Scientific & Technology Research,* Volume 1, Issue 4, May 2012.

[11] Damian, C. Lunca, E., "A low area FIR filter for FPGA implementation", *34th International Conference on Telecommunications and Signal Processing (TSP),* pp.521-524, 18-20 Aug. 2011. doi: 10.1109/TSP. 2011.6043675

[12] Heejong Yoo, David V.anderson, "Hardware-efficient distributed arithmetic architecture for high-order digital filters" in proc. *IEEE International conference on Acoustics, speech, and signal processing (ICASSP'05),*vol.5, pp.125-128 2005. doi: 10.1109/ICASSP.2005.1416256

[13] Mahesh, R. Vinod, A.P., "A New Common Subexpression Elimination Algorithm for Realizing Low-Complexity Higher Order Digital Filters", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* , vol.27, no.2, pp.217,229, Feb. 2008. doi: 10.1109/TCAD.2007.907064

[14] John G. Proakis., Dimitris G. Manokalis, "Digital Signal Processing Principle algorithms and

applications" PHI publication, New Jersey, Third Edition 2004.

[15] Catalin Damian, Cristian Zet, Cristian Fosalau, Mihai Cretu, "Real, Reactive and Apparent Power Computing Using FPGA and PWM Intermediary Conversion" XX IMEKO TC4, Florence, Italy, Sept. 2008.

[16] K. S. Yeung, S. C. Chan, "Multiplier-Less Digital Filters Using Programmable Sum-Of-Power-Of-Two (Sopot) Coefficients", *IEEE International Conference on Field Programmable Technology*, pp.78 – 84, Dec. 2002. doi: 10.1109/FPT.2002.1188667

How to cite

Jinalkumari K. Dhobi, Dr. Y. B. Shukla, Dr. K.R.Bhatt, "*FPGA Implementation of Fir Filter using Various Algorithms: A Retrospective*". *International Journal of Research in Computer Science*, 4 (2): pp. 19-24, March 2014. doi: 10.7815/ijorcs.42.2014.081