ENGINEERING IN ADVANCED RESEARCHSCIENCE AND TECHNOLOGY

ISSN 2352-8648
Vol.03, Issue.01
June-2022
Pages: -518-527

DENSITY AND LATENCY OPTIMIZED DIGITAL TRANSPOSE FINITE IMPULSE RESPONSE FILTER

[1]TOLETI SURYA SAI RAMA PRAVALLIKA, [2]M.RAVIKANTH
[1]M.Tech, Dept. of ECE, Kakinada Institute of Engineering And Technology, Korangi, Kakinada, A.P.
[2]Assistant Professor, Dept. of ECE, Kakinada Institute of Engineering And Technology, Korangi, A.P.

ABSTRACT Transpose form finite-impulse response (FIR) filters are inherently pipelined and support multiple constant multiplications (MCM) technique that results in significant saving of computation This concept presents a multiplier-free realization of the block finite impulse response (FIR) filter in transpose form configuration using binary constant shifts method (BCSM). As an extension of this concept, proposed structure involves significantly less area delay product (ADP) and less energy per sample (EPS) than the existing block implementation of Transpose -form structure for medium or large filter lengths using radix16 modified booth encoding multiplier, while for the short-length filters, the block implementation of FIR structure has less ADP than the existing structure.

Keywords: *finite impulse response, multiple constant multiplications, area delay product, Binary constant Shift, Booth encoding.*

INTRODUCTION FIR DIGITAL filters find extensive applications in mobile communication systems for applications such as channelization, channel equalization, matched filtering, and pulse shaping, due to their absolute stability and linear phase properties. The filters employed in mobile systems must be realized to consume less power and operate at high speed. Recently, with the advent of software defined radio (SDR) technology, finite impulse response (FIR) filter research has been focused on reconfigurable realizations. The fundamental idea of an SDR is to replace most of the analog signal processing in the transceivers with digital signal processing in order to provide the advantage of flexibility through reconfiguration. This will enable different air-interfaces to be implemented on a single generic hardware platform to support multistandard wireless communications [1]. Wideband receivers in SDR must be realized to meet the stringent specifications of low power consumption and high speed. Reconfigurability of the receiver to work with different wireless communication standards is another key requirement in an SDR. The most computationally intensive part of an SDR receiver is the channelizer since it operates at the highest sampling rate [2]. It extracts multiple narrowband channels from a wideband signal using a bank of FIR filters, called channel filters. Using polyphase filter structure, decimation can be done prior to channel filtering so that the channel filters need to operate only at relatively low sampling rates. This can relax the speed of operation of the filters to a good extent [3]. However due to the stringent adjacent channel attenuation specifications of wireless communication standards, higher order filters are required for channelization and consequently the complexity and power consumption of the receiver will be high. As the ultimate aim of the future multi-standard wireless communication receiver is to realize its functionalities in

Copyright @ 2022**ijearst. All rights reserved.**

mobile handsets, where its full utilization is possible, low power and low area implementation of FIR channel filters is inevitable.

LITERATURE SURVEY The research paper on the design of FIR filters are published in various journals and presented in many conferences. Here the paper selected describes the design of FIR filters using VHDL or Verilog language.Some of the paper represents the modular design approach of the FIR filters and which is implemented in spartan-3E FPGA/Xilinx Virtex-5 FPGA.The evaluation result shows good area/power efficiency and flexibility by using different architectures for application. .Most papers have used microprogrammed FIR filters design approach . Abdullah A.Aljuffri, Aiman S. Badawai , Mohammad S.Bensaleh, Abdulfattah M.Odeid and SayedManzoor Qasim [1] in paper entitled "FPGA implementation of scalable micro programmed FIR filter architectures using Wallace tree and Vedic multipliers". In this paper used Wallace Tree and Vedic multipliers for implementation of 8-tap and 16-tap sequential and parallel micro programmed FIR filters architectures The designs are realized using Xilinx virtex-5 FPGA. Synplify pro tool used for synthesis, translation, mapping and place and route process and Reports are generated by CAD tool. Performance analyze base on parameter such as minimum period, slice LUTs and maximum operating frequency. The sequential FIR filters architecture designed using Wallace Tree multiplier seems to be more efficient as compared to Vedic multipliers. For 8-tap FIR filter using Wallace Tree have minimum period 11.448 ns and maximum operating frequency 87.4 MHz And for 16-tap FIR filter using Wallace Tree have minimum period 10.491 ns and maximum operating frequency 85.3 MHz .A. Aljuffri. M. M. AlNahdi, A.A.Hemaid , O. A. Alshaalan, M. S. BenSaleh, A.M. Obeid and S. M. Qasim [2], in paper entitled, "ASIC realization and performance evaluation of scalable micro-programmed FIR filter architectures using Wallace tree and Vedic multiplier". In this paper, Wallace tree and Vedic multiplier are used for efficient realization of 8-tap and 16-tap sequential and parallel scalable micro-programmed FIR filter architectures. The designs of FIR filter are coded in VHDL. Lfoundary 150nm standard-cell based technology is used for the hardware realization of the proposed designs in ASIC. Synopsys Design Compiler is used for thegate-level synthesis. Analyze the performance based on area, Slice LUTs and critical path delays. Wallace tree multiplier using CSA (Carry Skip Adder) has minimum area and delay while Vedic using KSA (Kogge-Stone Adder) has maximum area and delay. For 8-tap FIR filter have period 6.62 ns for 8-tap filter have period 6.62 ns and area 29496 μm^2. For 16-tap FIR filter have period 6.63 ns and area 47463 μm^2 .Sushma .S and Shobha .S [3] in paper entitled, "Design and implementation of sequential micro programmed FIR filter using efficient multipliers on FPGA". In this paper 8-tap sequential FIR architecture is implemented. Implementation of 8-tap sequential digital FIR filter is presented Using Wallace Tree and Vedic multiplier which is Coded in VHDL. The designs are realized using Xilinx Virtex-5 FPGA. FPGA Resource utilization of Wallace Tree and Vedic multiplier has improved. Analyzed the performance based on the parameter minimum period, slice LUTs and maximum frequency. Implementation result have maximumoperating frequency 217.68 MHz, minimum period 4.595 ns and slice LUTs 99 [3]. Pramod Kumar Meher and Abbes Amira [4], in paper entitled, " FPGA realization of FIR filters by efficient and flexible systolization using Distributed Arithmetic". This paperpresent the realization of 8-tap and 16-tap Digital FIR filters by systolic decomposition of distributed arithmetic (DA). Implemented on Xilinx Virtex-E XCV2000E FPGA using hybrid combination of Handel-C and parameterizable VHDL cores. Analyze the performance on the basis of maximum

Copyright @ 2022**ijearst. All rights reserved.**

operating frequency. Implementation is found less area-delay complexity. Implementing 8-tap FIR filter give maximum operating frequency 74.025 MHz and for 16-tap FIR filter 67.222 MHz .S. C. Prasanna and S. P. Joy Vasantha Rani [5], in paper entitled, "Area and Speed efficient implementation of symmetric FIR Digital filter through reduced parallel LUT Decomposed DA approach

DIGITAL FIR FILTER: Multipliers are key components of many high performance systems such as FIR filters, microprocessors, digital signal processors, etc. A system's performance is generally determined by the performance of the multiplier because the multiplier is generally the slowest clement in the system. Furthermore, it is generally the most area consuming. Hence, optimizing the speed and area of the multiplier is a major design issue. However, area and speed are usually conflicting constraints so that improving speed results mostly in larger areas. As a result, whole spectrums of multipliers with different area-speed constraints are designed with fully parallel processing. In between are digit serial multipliers where single digits consisting of several bits are operated on. These multipliers have moderate performance in both speed and area. However, existing digit serial multipliers have been plagued by complicated switching systems and/or irregularities in design. Radix 2^n multipliers which operate on digits in a parallel fashion instead of bits bring the pipelining to the digit level and avoid most of the above problems. They were introduced by M. K. Ibrahim in 1993. These structures are iterative and modular. The pipelining done at the digit level brings the benefit of constant operation speed irrespective of the size of' the multiplier. The clock speed is only determined by the digit size which is already fixed before the design is implemented.

$$Y[n]=b_0x[n]+b_1x[n-1]+b_2x[n-2]+b_3x[n-3]$$

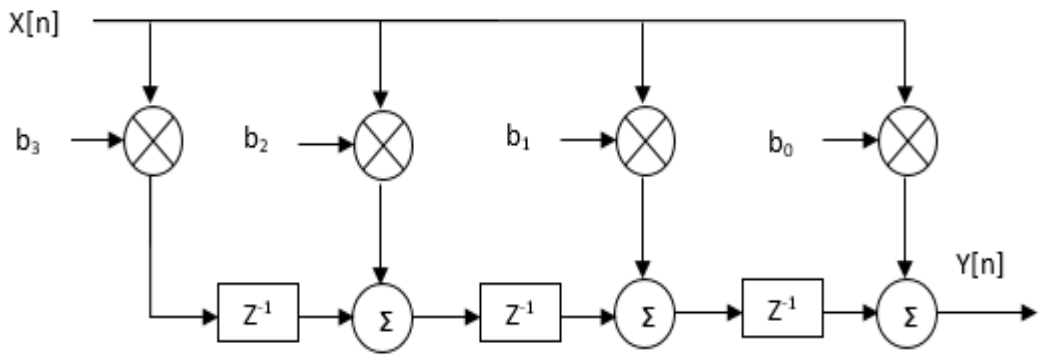


Fig1: DIGITAL FIR FILTER

In electronics, an adder or summer is a digital circuit that performs addition of numbers. In many computers and other kinds of processors, adders are used not only in the arithmetic logic unit(s), but also in other parts of the processor, where they are used to calculate addresses, table indices, and similar operations. Although adders can be constructed for many numerical representations, such as binary-coded decimal orexcess-3, the most common adders operate on binary numbers. In cases where two's complement or ones' complement is being used to represent negative numbers. We can view a full adder as a 3:2 loss compressor: it sums three one-bit inputs, and returns the result as a single two-bit number; that is, it maps 8 input values to 4 output values. Thus, for example, a binary input of 101results in an output of 1+0+1=10 (decimal number '2'). The carry-out represents bit one of the results, while the sum represents

Copyright @ 2022**ijearst. All rights reserved.**

bit zero. Likewise, a half adder can be used as a 2:2 loss compressor, compressing four possible inputs into three possible outputs. Such compressors can be used to speed up the summation of three or more addends. If the addends are exactly three, the layout is known as the carry-save adder. If the addends are four or more, more than one layer of compressors is necessary and there is various possible designs for the circuit.

MULTIPLIER-FREE BLOCK TRANSPOSE FORM FIR using BCSM

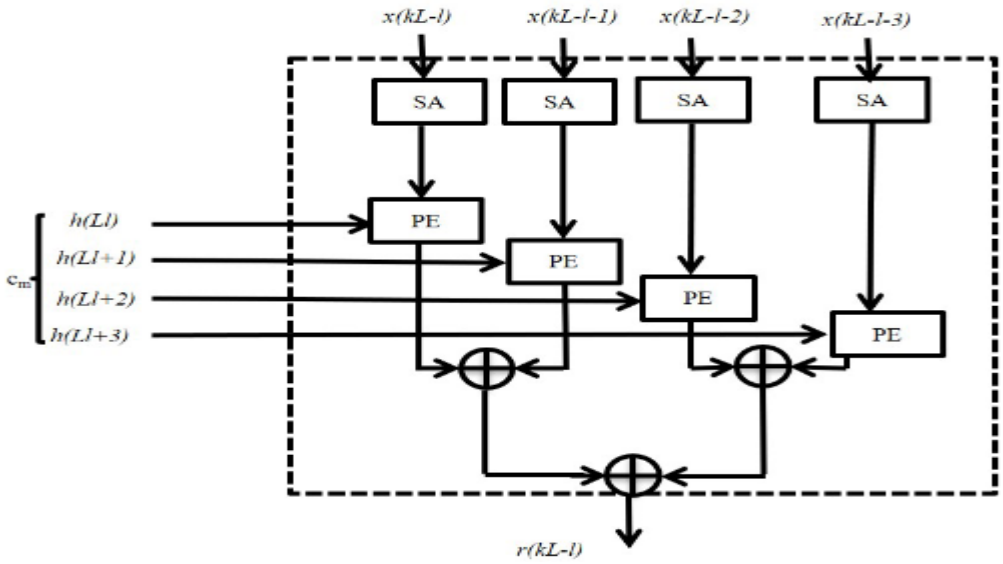


Fig2: BCSM DIGITAL FIR FILTER

Even-though the block FIR filter structure offers a through put improvement of L times when compared to that of the direct implementation, the number of multipliers and adders also increases with block size [3]. In this work, the multipliers in the IPC of block FIR filter are replaced with BCSM blocks [7] to make the structure multiplier-free and hardware efficient. Thus, each inner product cell of the IPUs of the block FIR filter is realized using L number of BCSM blocks as shown in above Figure resulting in a power efficient block FIR structure.

The BCSM block consists of a shift and add (SA) unit and processing elements (PEs) as discussed in [7]. The binary common sub-expressions (BCSs) based SA unit realizes all the binary common sub-expressions (BCSs) of the input [7]. For example, for a W0 = 12-bit representation of coefficient, h, having all non-zero bits, i.e., h = 0.1111_1111_1111, is the worst case scenario. The output of the FIR filter for the input x, is given by,

$$y = h * x = 2^{-1}x + 2^{-2}x + 2^{-3}x \quad \ldots \quad + 2^{-12}x$$

By splitting into a group of n = 3 bits starting from the most significant bit (MSB), it can be modified as,

$$y = 2^{-1}((x + 2^{-1}x + 2^{-2}x) + 2^{-3}(x + 2^{-1}x + 2^{-2}x) + \ \ldots. \\ + 2^{-9}(x + 2^{-1}x + 2^{-2}x))$$

Copyright @ 2022**ijearst. All rights reserved.**

The common term x+ 2−1x+ 2−2x corresponds to

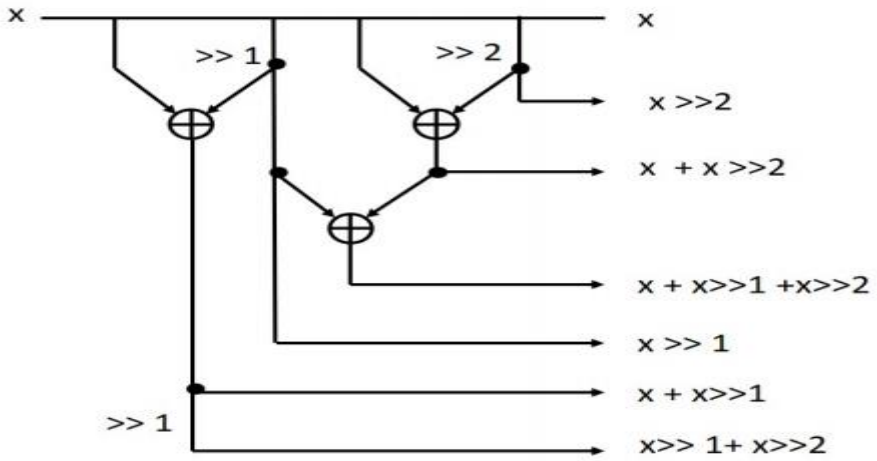


Fig3: Structure of shift and add unit

a 3-bit BCS [1 1 1], which can be obtained from the SA unit. The structure of SA unit, which realizes 3-bit BCSs. The shifting operation represented as "x >> sh" , indicates the input x is shifted right by sh units. It needs only 3 adders to realize all the 3-bit BCSs starting from [0 0 0] to [1 1 1]. For n-bit BCSs, the SA unit requires 2 n−1 −1 adders. It is proved in [7], that the 3-bit BCSs based BCSM is hardware efficient, when compared to larger sized BCSs representation and hence, the size of BCSs, n is taken as 3 in this work.

The adder unit performs the final summation after the PE operations to obtain the product term. The structure of the PE block for a word length of W0 = 12. The FIR filter coefficients are stored in the coefficient selection unit (CSU), and are fragmented into fixed groups of size n to form the select lines of the multiplexers. Based on the value of these select lines, the multiplexers select the corresponding output from the SA unit as shown in Fig. 6. For the word length W0 = 12, the PE includes, d W0 n e = 4, 8:1 multiplexers, Mux-1 to Mux-4. If the number of bits in the BCS is selected as n = 3, the select lines of these multiplexers are obtained by partitioning the coefficient bits into group of 3 as shown in Eq. (7). Let r1 to r4 denotes the outputs of the

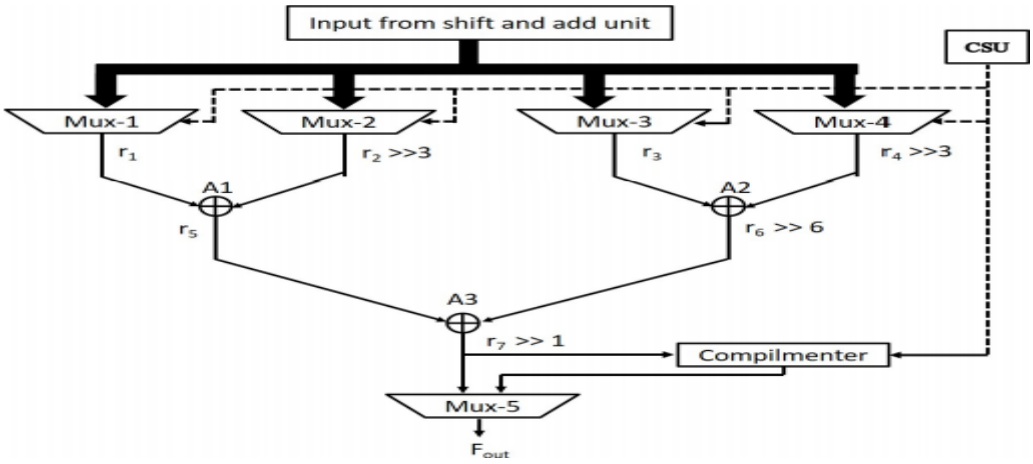


Fig4. Structure of PE of the BCSM architecture

Copyright @ 2022**ijearst. All rights reserved.**

MULTIPLICATION USINF MODIFIED BOOTH ENCODING:

RADIX-16 MODIFIED BOOTH ALGORITHM:

Booth recoding was originally introduced when multiplication was implemented using a series of shift-add operations. By recoding, the number of 1's in the multiplier could be reduced, and thereby the number of additions. The core idea is as follows:

Multiplier: $B=-b_{n-1}2^{n-1}+\sum_{i=0}^{n-2} b_i 2^i$

 Introduce new variables: $b'_i=-b_i+b_{i-1}$ for $i=0...n-1$ (assume $b_{-1}=0$).

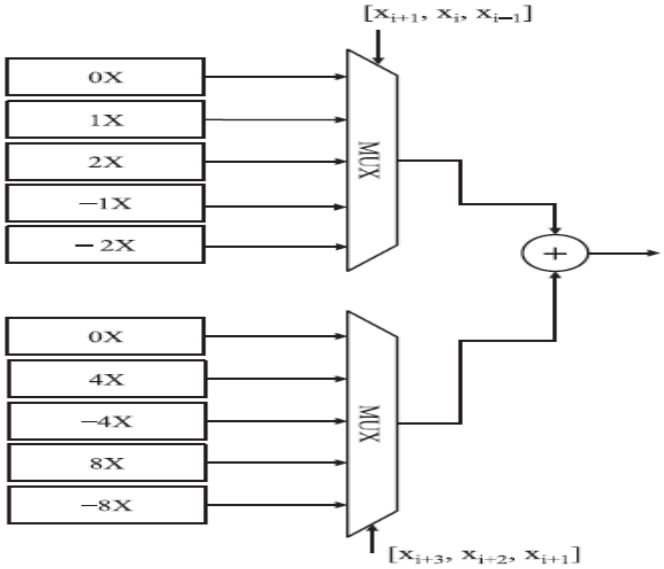
 Compute $P=\sum_{i=0}^{n-1}(b'_i \times 2^i \times A)$

 although this looks very similar to the normal product, note that any time $b_i=b_{i-1}$ there is no addition to be performed as the partial product will be zero. In the case of serial addition, these steps can be skipped, thus saving computation. To reduce the number of partia l products added while multiplying the multiplicand higher radix Booth Encoding algorithm is one of the most well-known techniques used .Radix 16 Booth algorithm which scan strings of five bits with the algorithm given below:

(1) Extend the sign bit 1 p osition if necessary to ensure that n is even.

(2) Append a 0 to the right of the LSB of the multiplier.

(3) According to the value of each vector,each Partial Product will be 0,y,+2y,+3y,+4y,+5y,+6y,+7y,+8y, -8Y, - 7y, - 6y, - 5y, - 4y, -3y, - 2Y, - Y. The multiplicati on of y is done by shifting y by one bit to the left. Thus, in any case, in designing n bit parallel multipliers, only n/4 partial products are generated



Copyright @ 2022**ijearst. All rights reserved.**

Fig5. Radix16 modified booth encoder

X_{i+3}	X_{i+2}	X_{i+1}	X_i	X_{i-1}	PP
0	0	0	0	0	0Y
0	0	0	0	1	1Y
0	0	0	1	0	1Y
0	0	0	1	1	2Y
0	0	1	0	0	2Y
0	0	1	0	1	3Y
0	0	1	1	0	3Y
0	0	1	1	1	4Y
0	1	0	0	0	4Y
0	1	0	0	1	5Y
0	1	0	1	0	5Y
0	1	0	1	1	6Y
0	1	1	0	0	6Y
0	1	1	0	1	7Y
0	1	1	1	0	7Y
0	1	1	1	1	8Y
1	0	0	0	0	-8Y
1	0	0	0	1	-7Y
1	0	0	1	0	-7Y
1	0	0	1	1	-6Y
1	0	1	0	0	-6Y
1	0	1	0	1	-5Y
1	0	1	1	0	-5Y
1	0	1	1	1	-4Y
1	1	0	0	0	-4Y
1	1	0	0	1	-3Y
1	1	0	1	0	-3Y
1	1	0	1	1	-2Y
1	1	1	0	0	-2Y
1	1	1	0	1	-1Y
1	1	1	1	0	-1Y
1	1	1	1	1	0Y

Table1 Radix16 modified booth encoding table

CONCLUSION AND FUTURE SCOPE In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for areadelay efficient realization of both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications. We have presented a scheme to identify the MCM blocks for horizontal and vertical subexpression elimination in the proposed block FIR filter for fixed coefficients to reduce the computational complexity. Along with all above implementations; Radix16 modified booth multiplier is designed for improved efficiency. Research can be extended in design of FIR filter using various optimization techniques ACO, PSO etc., In further work FIR filters can be design using evolutionary algorithms etc. Adaptive filtering advantage of achieving efficient filtering at continuously changing conditions and the effect of Fine word length effects and their remedies are discussed in the future scope of the work.

REFERENCES:

[1] P. P. Vaidyanathan, Multirate Systems and Filter Banks, Englewood Cliffs, N.J., USA: Prentice Hall, 1993.

[2] A. Sibille, C. Oestges and A. Zanella, MIMO: From Theory to Implementation, New York, NY, USA: Academic, 2010.

Copyright @ 2022**ijearst. All rights reserved.**

[3] N. Kanekawa, E. H. Ibe, T. Suga and Y. Uematsu, Dependability in Electronic Systems: Mitigation of Hardware Failures, Soft Errors, and Electro- Magnetic Disturbances, New York, NY, USA: Springer Verlag, 2010.

[4] M. Nicolaidis, "Design for soft error mitigation," IEEE Trans. Device Mater. Rel., vol. 5, no. 3, pp. 405–418, Sep. 2005.

[5] C. L. Chen and M. Y. Hsiao, "Error-correcting codes for semiconductor memory applications: A state-of-the-art review," IBM J. Res. Develop., vol. 28, no. 2, pp. 124–134, Mar. 1984.

[6] A. Reddy and P. Banarjee "Algorithm-based fault detection for signal processing applications," IEEE Trans. Comput., vol. 39, no. 10, pp. 1304–1308, Oct. 1990.

[7] S. Pontarelli, G. C. Cardarilli, M. Re, and A. Salsano, "Totally fault tolerant RNS based FIR filters," in Proc. IEEE IOLTS, 2008, pp. 192–194.

[8] Z. Gao, W. Yang, X. Chen, M. Zhao and J. Wang, "Fault missing rate analysis of the arithmetic residue codes based fault-tolerant FIR filter design," in Proc. IEEE IOLTS, 2012, pp. 130–133.

[9] B. Shim and N. Shanbhag, "Energy-efficient soft error-tolerant digital signal processing," IEEE Trans. Very Large Scale Integr. Syst., vol. 14,no. 4, pp. 336–348, Apr. 2006.

[10] Y.-H. Huang, "High-efficiency soft-error-tolerant digital signal processing using fine-grain subword-detection processing," IEEE Trans. Very Large Scale Integr. Syst., vol. 18, no 2, pp. 291–304, Feb. 2010.

[11] P. Reviriego, C. J. Bleakley, and J. A. Maestro, "Structural DMR: A technique for implementation of soft-error-tolerant FIR filters," IEEE Trans. Circuits Syst. II: Exp. Briefs, vol. 58, no. 8, pp. 512–516, Aug. 2011.

[12] P. Reviriego, S. Pontarelli, C. Bleakley and J. A. Maestro, "Area efficient concurrent error detection and correction for parallel filters," IET Electron. Lett., vol. 48, no 20, pp. 1258–1260, Sep. 2012.

[13] Z. Gao et al., "Fault tolerant parallel filters based on error correction codes," IEEE Trans. Very Large Scale Integr. Syst., vol. 23, no. 2, pp. 384–387, Feb. 2015.

[14] R. W. Hamming, "Error correcting and error detecting codes," Bell Sys. Tech. J., vol. 29, pp. 147–160, Apr. 1950.

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

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

[17] 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.

18] S. Trini et al., "FFT Spectrum Analyzer Project for Teaching Digital Signal Processing With FPGA Devices", IEEE Transactions on Education, Vol. 50, No. 3, 2007, pp. 229-235.

Copyright @ 2022**ijearst. All rights reserved.**

[19] S. Vaishali et al., "High-throughput and compact reconfigurable architectures for recursive filters", IET Communications, Vol. 12, No. 13, 2018, pp. 1616-1623.

[20] C. V. G. Edilberto, M. R. P. Diego, J. G. Edwar, "Implementation and simulation of IIR digital filters in FPGA using MatLab system generator", Proceeings of the IEEE 5th Colombian Workshop on Circuits and Systems, Bogota, Colombia, 16-17 October 2014.

[21] S. M. R. Islam, R. Sarker, S. Saha, A. F. M. N. Uddin, "Design of a programmable digital IIR filter based on FPGA", Proceedings of the International Conference on Informatics, Electronics & Vision, Dhaka, Bangladesh, 18-19 May 2012.

[22] Z. Zhao, G. Li, "Comparative study of the general- ized DFIIt structure and its equivalent state-space realization", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Pro- cessing, Hong Kong, China, 6-10 April 2003.

[23] A. Bhattacharyya, P. Sharma, N. Murali, S. S. Murty, "Development of FPGA based IIR Filter implementation of 2-degree of Freedom PID controller", Proceedings of the Annual IEEE India Conference, Hyderabad, India, 16-18 December 2011.

[24] V. K. Singh, R. N. Tripathi, T. Hanamoto, "HIL CoSimulation of Finite Set-Model Predictive Control Using FPGA for a Three-Phase VSI System", Energies, Vol. 11, No. 4, 2018, pp. 1-15.

[25] Y.-S. Kung et al., "Simulink/Modelsim Co-Simulation and FPGA Realization of Speed Control IC for PMSM Drive", Procedia Engineering, Vol. 23, 2011, pp. 718-727.

[26] Sudhi Sudharman, Bindiya TS, Member, IEEE "Multiplier-free Realization of High Throughout Transpose Form FIR Filter" 2020 IEEE.

Copyright @ 2022**ijearst. All rights reserved.**