http://www.jctjournals.com ISSN(ONLINE):2278 - 3814

Volume-12, Issue-10, October 2023
JOURNAL OF COMPUTING TECHNOLOGIES (JCT)
International Journal
Page Number: 01-06

Designing and Implementation High Performance FIR Filter Architecture using Distributive Arithmetic Technique:- A Review

[1]Rambhuwan Sahu, [2]Prof. Nishi Pandey, [3]Prof. Abhishek Agwekar
[1] M.Tech Scholar, [2]Assistant Professor ,[3]Head of department
[1, 2, 3] Truba Institute of Engineering and Information Technology, Bhopal, M.P., India
[1]Sahurambhuvan6@gmail.com ,[2]nishi.pandey@trubainstitute.ac.in, [3]abhiagwekar@gmail.com

Abstract— Multiplication is an important function in arithmetic operations. A CPU (central processing unit) devotes a considerable amount of processing time in performing arithmetic operations. Multiplication requires substantially more hard-ware resources and processing time than addition and sub-traction. Digital signal processors (DSPs) are the technology that is omnipresent in engineering Discipline. Fast multiplication is very important in DSPs for digital filter, convolution, Fourier transforms etc. Low complexity and configurability is the key features in emerging communication applications in order to support multi-standards and operation modes. To obtain these features, efficient implementations of finite impulse response (FIR) filter with multiplier-less distributive arithmetic technique is proposed in this paper. In this technique consist of Look Up Table (LUT), shift register and accumulator. Based on this technique multipliers in FIR filter are removed. Multiplication is performed through shift and addition operations. The LUT can be subdivided into a number of LUT to reduce the size of the LUT for higher order filter. Each LUT operates on a different set of filter taps. Analysis on the performance of various filter orders with different address length are done using Xilinx synthesis tool. The proposed architecture provides less latency and less area compared with existing structure of FIR filter. The proposed algorithm gives a flip flop 124 as compared with 312 flip flops for previous algorithm. Similarly, proposed design gives 114 adders and 0 multiplier as compared with 60 adders and 64 multipliers for previous 16-tap FIR filter algorithm.

Keywords— *Visual Light Communication (VLC), Density Function Theory (DFT), Ultraviolet (UV), Lithium Niobate (LN), Stoichiometric Lithium Tantalate (SLT), Atomistic Toolkit (ATK), Generalized Gradient Approximation GGA)*

I. INTRODUCTION

In signal processing the function of a filter is to remove unwanted parts of the signal, such as random noise, or to extract useful parts of the signal, such as the components lying within a certain frequency range.

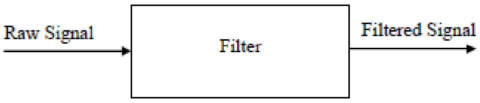

Figure 1: A Block Diagram of a Basic Filter

There are two types of filter analog and digital. FIR Filter is the kind of digital filter, which can be used to perform all kinds of filtering i.e. high pass, low pass, band pass and band reject etc.

A. Analog Filters

A analog filters utilizes simple electronic circuits made up from segments, for example, resistors and capacitors to create the required sifting impact. Such channel circuits are broadly utilized as a part of such applications as commotion diminishment, flag upgrade, and numerous different ranges [2].

B. Digital Filters

A digital filter can be defined as a filter which operates on digital signals, such as sound represented inside a computer. In order to convert an output digital signal into analog form, it is necessary to perform additional signal processing to obtain the perfect result and is demonstrated. The process of converting an analog signal into digital form is performed by sampling with a finite sampling frequency. If an input signal contains frequency components higher than half the sampling frequency, it will cause distortion to the original spectrum [1].

Basic Fourier transform theory states that the linear convolution of two sequences in the time domain is same as multiplication of two corresponding spectral sequences in the frequency domain. Sifting is fundamentally the increase of the flag range by the recurrence space drive reaction of the channel. For a perfect low-pass channel the pass band some portion of the flag range is duplicated by one and the stop-band some portion of the flag by zero. Simple Filters, Software-Based and Hard-wired Digital Filters Owing to the way that simple and computerized channels are physically executed, a simple channel is characteristically more size and power-proficient, albeit more segment delicate, than its advanced partner [2].

It can be executed in a clear way. All in all, as flag recurrence builds, the uniqueness in productivity increments. Attributes of uses where advanced channels with more size and proficient than simple channels are: straight stage, high stop band weakening, low pass band swell; the channel's reaction must be programmable or versatile; the channel must control stage and, low shape considers (a computerized channel's shape variable is the proportion of the channel's pass data transfer capacity in addition to the channel's move transmission capacity to the channel's pass transmission capacity). The process of converting an analog signal into digital form is performed by sampling with a finite sampling frequency

II. LITERATURE REVIEW

It can be executed in a clear way. All in all, as flag recurrence builds, the uniqueness in productivity increments. Attributes of uses where advanced channels with more size and proficient than simple channels are: straight stage, high stop band weakening, low pass band swell; the channel's reaction must be programmable or versatile; the channel must control stage and, low shape considers (a computerized channel's shape variable is the proportion of the channel's pass data transfer capacity in addition to the channel's move transmission capacity to the channel's pass transmission capacity). The process of converting an analog signal into digital form is performed by sampling with a finite sampling frequency.

Deepak Kumar Patel et al. (2016, [2]) - in this paper, the speed and range are presently the very beginnings of the principal configuration issues in advanced period. To build speed, while doing the duplication or expansion operations, has dependably been a fundamental prerequisite of planning of cutting edge framework and application. Convey Select Adder (CSA) is a quickest viper utilized as a part of numerous processors to fulfill quick number juggling capacity. A wide range of viper engineering outlines have been created to build the productivity of the snake. It is ordinarily realized that every second any processors performed a large number of work works in semiconductor industry. So when we do outlining of multipliers, one of the principle benchmarks is performing speed that ought to be taken in the brain. In this paper, we propose a method for outlining of FIR channel utilizing multiplier in light of compressor and convey select viper. Execution of all viper outlines is actualized for 16, 32 and 64 bit circuits.

K. Durga et al. (2016, [3]), in this paper, an effective engineering of FIR channel structure is exhibited. For accomplishing low power, reversible rationale method of operation is actualized in the plan. Territory overhead is the exchange off in the proposed outline. From the amalgamation comes about, the proposed low power FIR channel engineering offers 18.1 % of energy sparing when contrasted with the customary outline. The territory overhead is 2.6% for the proposed engineering.

Indranil Hatai et al. (2015, [4]), this brief proposes a two-step optimization technique for designing a reconfigurable VLSI architecture of an interpolation filter for multistandard digital up converter (DUC) to reduce the power and area consumption. The proposed technique initially reduces the number of multiplications per input sample and additions per input sample by 83% in comparison with individual implementation of each standard’s filter while designing a root-raised-cosine finite-impulse response filter for multistandard DUC for three different standards. In the next step, a 2-bit binary common sub expression (BCS)-based BCS elimination algorithm has been proposed to design an efficient constant multiplier, which is the basic element of any filter. This technique has succeeded in reducing the area and power usage by 41% and 38%, respectively, along with 36% improvement in operating frequency over a 3-bit BCS-based technique reported earlier, and can be considered more appropriate for designing the multi-standard DUC.

Shamim Akhter et all in (2018- Design and Analysis of Distributed Arithmetic based FIR Filter, In designing digital filters,Multiply-Accumulate (MAC) unit is used. MAC comprises of multiplier, adder and an accumulator. Faster adder and multiplier circuits are required for high speed MAC unit. But MAC based structures have disadvantages like high power dissipation, slow processing etc. The multiplication operation where input data is to be multiplied with the fixed coefficients considerably took large place to store their temporary data. So, memory based multiplication technique substitute multipliers to reduce area and latency of system. Distributed Arithmetic (DA) is one of the memory based technique. DA based technique substitute multipliers in FIR filters. In this paper, detailed analysis is presented for designing 16-Tap FIR filter using DA and Off-Set Binary Coding (OBC)-DA in VHDL. Synthesis is done using Xilinx ISE for Virtex-4 ML 402. Area, delay and power analysis is performed using Synopsys Design Compiler for 32/28 nm std_cell.

Narendiran S. in 2021, One of the essential components of a Digital Signal Processing (DSP) system is the Finite

Impulse Response (FIR) filter. FIR filter uses the Multiply and Accumulate (MAC) operation for its computation. Conventional MAC units are slow and consume high power, making them unsuitable for energy-constrained devices. The MAC operations in FIR filter uses constant filter coefficients as one of its inputs. This situation is well suited for a bit-serial technique such as Distributed Arithmetic (DA). However, the traditional DA has the drawback of using huge memory resources as the filter order increases. An efficient LUT-less Modified Distributed Arithmetic architecture is proposed in this paper to solve the memory problem. This architecture removes the need for precomputation of weighted sums needed for the LUT in a DA using multiplexers and adders. Also, the architecture is designed to extend the range of input values. Further, a 16-Tap FIR filter is designed, synthesized with Xilinx ISE, and implemented for an XC4VSX35-FF668-10 based FPGA to measure the performance of this architecture. Our implementation results show that the design uses fewer resources and achieves faster filtering than the filter's previous implementations.

Balaji M,in 2020 - This work presents a way to increase the throughput and energy efficiency of finite impulse response (FIR) filters through the efficient application of retiming and two-level pipelining. It is a challenge to increase the filter's throughput and energy efficiency while reducing latency and hardware complexity. The operations of addition and multiplication are divided using two-level pipelining. The break addition procedure is retimed. The architecture of m-tap filter (4-tap, 8-tap, 16-tap, 32-tap, and 64-tap) with n-bit input word length (4-bit, 8-bit, 16-bit, and 32-bit) Pipelined Retiming delay generation Filters (PRF), were designed. The proposed distributed arithmeticbased FIR Filter with pipelining has produced the least delay of 2.564ns for 4-tap with 8-bit input, and the maximum delay of 56.040ns for 64-tap with 32-bit word length. The proposed distributed arithmetic-based FIR Filter with retiming method has produced the least delay of 0.687ns for 4-tap with 8-bit input, and the maximum delay of 4.535ns for 64-tap with 32-bit word length. When compared with the pipelining method, the delay has been decreased by 73.20% for 4-tap with 8-bit input and 91.90% for 64-tap with 32-bit word length.

Ch. Pratyusha et all in 2019, In this paper the proposed efficient FIR filter architecture using a distributed arithmetic (DA) algorithm in which two issues are discussed in the conventional FIR filter. The FIR filter is well known to include delay elements, multipliers and adders. Due to the need for multipliers, this results in 2 demerits which are (i) increased in area and (ii) delayed increases that eventually lead to low efficiency (low speed). A notable feature of the proposed technique is to substitute a trivial amount of indexed LUT pages instead of conventional LUT based DA that it helps to maintain the access time lower. Also, significant idea connected with the proposed technique is required page can be thoroughly selected with the selection module without needing adders that result in reduced computation time. Furthermore, the proposed fast FIR filter is used for the powerful ECG noise elimination technique, which is prevalently used in biomedical and healthcare applications. The designs are simulated and synthesized by using Xilinx ISE. It can be seen from reports that our proposed DA consumes 30% less power for 11-tap FIR filters with a 40% shorter area, while the saving in power consumption for 8-tap FIR filters is 30% to 80% and 35% to 80% in the area. Especially in contrast with all the above-mentioned DA techniques, our enhanced quick FIR filters require less area and less power intake due to their lower memory requirements. All architectures are designed for FIR filters with 4 and 8 taps.
Manoj Srivastava et all in 2015, n this project use Distributed Arithmetic (DA) technique for FIR filter. In this technique consist of Look Up Table (LUT),shift register and accumulator. Based on this technique multipliers in FIR filter are removed. Multiplication is performedthrough shift and addition operations. The LUT can be subdivided into a number of LUT to reduce the size of the LUT forhigher order filter. Each LUT operates on a different set of filter taps. Analysis on the performance of various filter orderswith different address length are done using Xilinx synthesis tool. The proposed architecture provides less latency and lessarea compared with existing structure of FIR.

D. Kalaiyarasi et all in 2019, Distributed Arithmetic (DA) based architecture is an efficient technique to attain high throughput without hardware multiplier and also it is essential for bit serial operation. The DA based Finite Impulse Response (FIR) adaptive filter is well suited for hardware implementation in Field Programmable Gate Array (FPGA) device. In conventional DA the partial products of the filter coefficients have been pre-calculated and stored in Look up Table (LUT) which in turn will increase the logic elements and power. To overcome this problem DA based Least Mean Square (LMS) adaptive filter using offset binary coding (OBC) without LUT is proposed. The proposed method will reduce the logic elements by half when compared to the conventional DA based OBC filter. The Carry Save Accumulator (CSA) is used to carry out the operation of shift and accumulation. The proposed architecture is implemented in Quartus II 9.1 with the device as Stratix-EP2S15F484C3 which offers 13.72% high throughput, 56.92% reduction in logic elements, 42.84% reduction in power, 57.74% reduction in logical registers for $N = 16$ and for $N = 32$ the number of logical element is reduced to 80.87%, 66.66% reduction in power and 24.12% high throughput.

III. PROPOSED METHODOLOGY

A. Digital Fir Filter:-

Digital signal processing algorithms are increasingly employed in modern wireless communications and multimedia consumer electronics, such as cellular telephones and digital cameras. The new generation of telecommunication equipment often requires the use of high order high-speed low-power Finite Impulse Response (FIR) filters. The output of an N tap FIR filter, which is the convolution of the latest L input samples, is given in equation (3.1). L is the number of coefficients h (k) of the filter, and x (n) represents the input time series.

$$Y[n] = \Sigma\, h[k]\, x[n-k] \qquad k= 0,\ 1,\ \ldots,\ N-1$$

The conventional L tapped delay line realization of this inner product is shown in Figure 2. This implementation translates to L multiplications and L-1 additions per sample to compute the result. This can be implemented using a single Multiply Accumulate (MAC) engine, but it would require L –MAC cycles, before the next input sample can be processed.

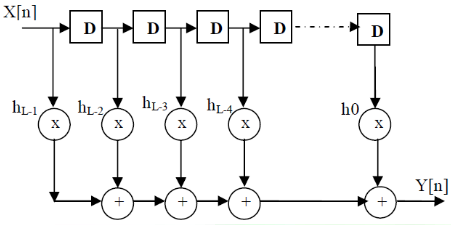

Figure 2: Traditional digital L-Tap FIR filter

Augmentation can be considered as a progression of rehashed increments. The number to be included is the multiplicand, the quantity of times that it is included is the multiplier and the outcome is the item. Hence adders, multipliers and defer components are critical segment to develop advanced FIR channel.

A definitive point of this examination is to outline a rapid and low power computerized FIR channels. In this exploration work, FIR channel is planned with cluster multiplier which thusly framed with eight unique sorts of snake units. So also, FIR channel is outlined with Braun, limb wooly and Wallace tree multiplier which thusly are framed with eight unique sorts of viper units. Among the outline of computerized FIR channel with four unique sorts of multiplier structures in which each structures is planned with an alternate kind of snake unit, Wallace tree multiplier structure creates better outcome in edge misfortune issue, speed and influence.

In view of the examination of various sorts of adders and multipliers, the NEW snake and Wallace tree multiplier executed utilizing NEW viper cell is turned out to be the best and it is utilized to develop FIR channels. This FIR channel turned out to be effective as far as power utilization and postpone which is the objective of the examination.

IV. DESIGN MODULES

The design of different bit FIR filter using VHDL presented in this chapter. This chapter covers the basic development stages of VHDL-based design followed by the basics of 16-bit FIR filter representation. Then the algorithms along with flow-charts for design of VHDL based 16-bit FIR filter to carry out any of operations of shift register, look up table and accumulator are presented.

A. Development Stage of Software

Regardless of the final product, VHDL designer has to follow the following four basic VHDL development stages as shown in Figure 3:

- ➢ Design
- ➢ Simulation
- ➢ Synthesis
- ➢ Design Implementation

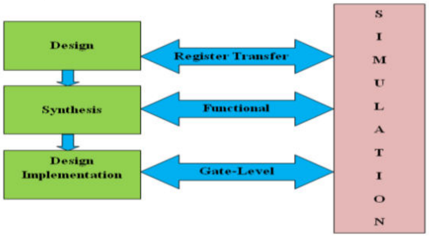

Figure 3: Development stages of VHDL

Design

The design process involves conversion of requirements into a format that represents the desired digital function(s). Common design formats are schematic capture, 25 hardware description language (HDL), or a combination of the two. Each method has its advantages and disadvantages but HDLs generally offer the greatest design flexibility. Schematic capture: Schematic capture is a graphical depiction of a digital design and shows the actual interconnection between each logic gate that produces the desired output function(s). Many of these logic gate symbols involve proprietary information which is available to the designer only through the specific vendor’s component library. It makes the design unrecognizable by competitors’ FPGA development tools and makes it vendor dependent. That means, the entire design process has to be repeated if a different vendor is used. View- Draw and EASE are examples of schematic capture tools by view logic and HDL respectively. The main advantage of schematic capture is that the graphical representation is easy to understand. But an increase in cost and time to reproduce a design for different vendors due to the design’s proprietary nature are its major drawbacks. HDL method: Hardware Description Languages (HDLs) use code to represent digital functions. “Firmware” often refers to the resulting HDL code. Use of HDL codes is a common and popular approach to FPGA design. One can create the source code with any text editor. HDLs can be generic (supported by multiple simulation and synthesis tool sets) like Verilog or VHDL (Very High Speed Integrated Circuit HDL), or vendor specific like Altera’s Hardware Description Language (AHDL), which is only recognizable by Altera’s design tool set. There are two writing styles for HDL designs: structural or behavioral. Structural HDL firmware is the software equivalent of a schematic capture design. Like schematic capture, a structural design uses vendor specific components to construct the desired digital functions. It is again vendor dependent and has the same disadvantages. Behavioral HDL firmware describes digital functions in generic or abstract terms that are generally vendor independent. This provides enough flexibility for code reuse in different vendor’s FPGAs with little or no code modification. Behavioral designs have advantages of its flexibility, time and cost-savings. Only those components are required to be changed for designs that require vendor specific resources, such as RAM. VHDL and Verilog are the most popular HDL languages.

V. CONCLUSION AND FUTURE SCOPE

Conclusion - FIR filter designed using high level system tools, shows that Verilog based filter design is better for low cost FPGA applications. System generator based FIR filter design works for high speed FPGA applications. Hence for low area designs, VHDL based design is better than System generator design. From the analysis of the results, it is found that the proposed FIR filter using DA algorithm gives a superior performance as compared with previous algorithm. It is observed from the result, shows that the B(N-1) + B'(N-1) flip flop, L(N-1) adders, NL multiplier and TM + TA + 3TFA cyclic period for previous design but (N- 1)L + NL flip flop, L(N-1) + (L-1)(N+2) adder, 0 multiplier and TA + (N-1) TFA cyclic period for proposed multiplier-less FIR filter implemented. From the analysis of the result, it is found that the proposed FIR filter using DA technique gives a superior performance as compared with previous algorithm. The maximum frequency and number of slice result are obtained for the proposed FIR filter using DA algorithm and previous algorithm. From the analysis of the results, it is found that the proposed FIR filter using DA algorithm gives a superior performance as compared with previous algorithm. It is observed from the table that the processing unit for FIR filter using DA technique uses 5736 number of slice, 154 number of slice flip flop, 10566 numbers of LUTs, 2.187 ns minimum period and 457.247 MHz maximum frequency for Spartan-3 device family. It is similarly that the processing unit for FIR filter using DA technique uses 5736 number of slice, 154 number of slice flip flop, 10566 number of LUTs, 1.298 ns minimum period and 770.416 MHz maximum frequency for vertex-2p device family.

Future Scope- FIR filters with proposed Vedic multiplier for the design and implementation of the different adder. As an extension to the present work, algorithms for composite radix could also be incorporated in the General radix algorithm to make it faster by reducing the operational complexity when the value of N is not a power of 2 or 4, but it can be expressed as a multiplication of various radices. In this thesis, two efficient designs of Adders are proposed which provides much optimized results in terms all important parameters considered. It would be interesting to implement higher bits modified Kogge-stone adder. This proposed architecture of Adders will be very efficient in various applications like Digital Electronics, Signal Processing and Robotics as Floating Point Arithmetic. Designed high speed convolution technique can use for designing the filter to filter the blurred signal. This can be used for image 49 processing and digital signal processing technique. By using high speed multiplier ALU, GPU and CPU can be designed. The algorithms presented may be extended to multidimensional data.

References

[1] Basant Kumar Mohanty, and Pramod Kumar Meher, "High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 78, No.04, pp. 01-09, April 2016.

[2] Indranil Hatai, Indrajit Chakrabarti, and Swapna Banerjee, "An Efficient VLSI Architecture of a Reconfigurable Pulse-Shaping FIR Interpolation Filter for Multi-standard DUC", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 23, No. 6, pp. 345-353, June 2015.

[3] Ravi H Bailmare, S. J. Honale and Pravin V Kinge, "Design And Implementation of Adaptive FIR Filter using Systolic Architecture", in International Journal of Current Engineering and Technology , Vol.4, No.6, pp. 143-152, June 2014.

[4] M. Usha, R. Ramadoss, "An Efficient Adaptive Fir Filter Based On Distributed Arithmetic", International Journal of Engineering Science Invention, Vol. 3, Issue. 4, pp. 15-20, April 2014.

[5] B. K. Mohanty, P. K. Meher, S. Al-Maadeed, and A. Amira, "Memory footprint reduction for power-efficient realization of 2-D finite impulse response filters," IEEE Transaction Circuits System. I, Reg. Papers, Vol. 61, No. 1, pp. 120–133, Jan. 2014.

[6] S. Y. Park and P. K. Meher, "Efficient FPGA and ASIC realizations of a DAbased reconfigurable FIR digital filter," IEEE Transaction Circuits System II, Exp. Briefs, Vol. 61, No. 7, pp. 511–515, July 2014.

[7] Sang Yoon Park and Pramod Kumar Meher, "Low power, High-throughput And Low- Area Adaptive FIR Filter Based on Distributed Arithmetic", in IEEE Transactions On Circuits And Systems-ii, Vol. 60, No. 6, pp. 346- 350, 2013.

[8] Basant K. Mohanty, And Pramod Kumar Meher, "A High-Performance Energy- efficient Architecture For FIR Adaptive Filter Based on New Distributed Arithmetic Formulation Of Block LMS Algorithm", In IEEE Transactions On Signal Processing, Vol. 61, No. 4, pp. 01-08, February, 2013.

[9] Pallavi Saxena, Urvashi Purohit, Priyanka Joshi, "Analysis of Low Power, Area Efficient and High Speed Fast Adder", International Journal Of Advanced Research In Computer And Communication Engineering, Vol. 2Issue 9, pp. 01-09, September 2013.

[10] R. Mahesh and A. P. Vinod, "New reconfigurable architectures for implementing FIR filters with low complexity," IEEE Transcation Computer-

Aided Design Integral Circuits System, Vol. 29, No. 2, pp. 275–288, Feb. 2010.

[11] P. K. Meher, “New approach to look-up-table design and memory based realization of FIR digital filter,” IEEE Transaction Circuits System I, Regular Papers, Vol. 57, No. 3, pp. 592–603, Mar. 2010.

[12] K. Meher, S. Chandrasekaran, and A. Amira, “FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic,” IEEE Transaction Signal Processing, Vol. 56, No. 7, pp. 3009–3017, July 2008.

[13] R. Mahesh and A. P. Vinod, “A new common subexpression elimination algorithm for realizing low-complexity higher order digital filters,” IEEE Transaction Computer-Aided Design Integral Circuits System, Vol. 27, No. 2, pp. 217–219, Feb. 2008.

[14] A. P. Vinod and E. M. Lai, “Low power and high-speed implementation of FIR filters for software defined radio receivers,” IEEE Trans. Wireless Communication, Vol. 7, No. 5, pp. 1669–1675, July 2006.

[15] K.-H. Chen and T.-D. Chiueh, “A low-power digit-based reconfigurable FIR filter,” IEEE Transaction Circuits System II, Exp. Briefs, Vol. 53, No. 8, pp. 617–621, Aug. 2006