



Implementation Multi-channel Symmetric Systolic FIR Filter Architectures for High-Performance DSP Applications

GAYATRI GADE, JYOTHULA SWATHI, NELAVELLI JAYAMMA

Dept of ECE,

Priyadarshini Institute of Science and Technology for Women Khammam.

ABSTRACT

High performance Finite Impulse Response (FIR) filters are extensively used in digital signal processing (DSP), communications, image processing and many more areas. This project approaches Field Programmable Gate Arrays (FPGAs) based on systolic FIR architectures using multi-channel technique with symmetric coefficients to ensure faster response and optimum area. The embedded Digital Signal Processing (DSP) blocks for its multiply-accumulator (MAC) perform accurate operations where the filter architecture is efficiently realized in the reconfigurable hardware platform. The characteristics of systolic FIR architecture are synchrony, modularity and regularity to make a perfect filter design. Also, its pipeline structure provides high throughput and symmetric technique reduces the memory size. Both of these techniques improve the overall performance of the FIR architecture in FPGA domain. The proposed FIR architecture has been successfully verified by Xilinx ISE 14.7 tool. The design method shows a great improvement of maximum operating frequency and save the area as compared to the earlier architectures.

208

DOI Number: [10.48047/nq.2020.18.12.NQ20256](https://doi.org/10.48047/nq.2020.18.12.NQ20256)

NeuroQuantology 2020; 18(12):208-220

INTRODUCTION

Objectives

An Integrating all the blocks to construct a model in a single package has become the need of the day. Field Programmable Gate Arrays (FPGA) has become most popular among all the integrating techniques available. Both logic design and DSP based applications are supported by FPGA's. Lot of research on FPGA's are carried out to accelerate DSP applications. Complicated system demands faster speed as well as quick and fast system performance. These could be done by optimizing a circuit to a single problem. To achieve high throughput, DSP systems dictate hardware intensive solutions. If 40 million instructions per second (MIPS) is the operating speed of a DSP chip, the bandwidth requirement is lower as compared to 500 KHz. Therefore, DSP processors are unfit

for implementation of many stages of communication system. FPGA's are reprogrammable for various equipment capacities with no expansion in the execution cost as connected with custom ICs and ASIC's. Reconfigurability and parallelism are the real points of interest of FPGA's. Size/adaptability of the rationale cell, speed and the directing engineering are central point for a FPGA based outline. The rationale cell executed on FPGA can be a basic transistor or an unpredictable structure like chip. FPGA's have two execution approaches. One is through SRAM based LUT and the second using multiplexer based rationale components. Multiplexer based execution is by and large quicker than SRAM based outline, yet possess marginally more prominent region. Commotion and other undesirable sign parts of a sign can be



evacuated by computerized channels. They work on a limited accuracy computerized representation of a sign. Simple signs are changed over to advance by testing, quantizing and coding. A small special purpose digital computer then processes the obtained digital signal and converts the input sequence into a desired output sequence. Advanced preparing of simple signs has a few points of interest. The PC can be time shared for a few uses and the computerized usage expense is impressively lower than that of its simple partner.

Advanced channel's exactness depends just on the word length, quantizing interim and the testing rate. Instead of RLC segments, the advanced channels utilize basic components like adders, multipliers, shifters and postponement components.

Subsequently they are not influenced by elements like temperature, warm float and others that influence the simple channel. Advanced channels can be altered by changing the PC calculation as opposed to simple framework, which must be physically remade. The execution of limited move band FIR channels requires impressively more number of juggling operations and equipment parts, for example, multipliers, adders and postponement components. To accomplish the same frequency detail, FIR channels require higher number of taps contrasted with FIR channels. To actualize FIR channels, extensive number of multiply and Accumulate (MAC) squares are required. The operational velocity of FIR channel is by and large a bottleneck for high channel throughput. This makes them unacceptable for FPGA usage. To stay away from MAC operations, various Multipliers less methods are being utilized. Among all the accessible strategies, Distributed Arithmetic (DA) is most appropriate for FPGA execution. DA utilizes Look-Up Tables, adders and movement registers to play out the separating operation. In this system, LUT's are pre developed for all the conceivable blends of inputs and the channel coefficients. They are conjured while performing channel operation.

In the present work an effective and upgraded Distributed Arithmetic based strategy for rapid reconfigurable outline and execution of FIR channels is created. In practical applications of digital signal processing, changing the sampling rate of a signal is a major challenge to be addressed. Multirate systems employing multiple sampling rates in the processing of digital signals are popular in DSP.

The objectives of the research work are listed as follows:

- To FIR, filters designed for DSP provided lesser throughput.
- It formed a Dedicated Design.
- Resulted in Reconfigurable Design.
- Efficient Memory utilization.
- To design a Reconfigurable architecture to meet different specifications.
- Throughput to be improved.
- Reduced Power consumption

Problem specification

The execution of FIR channels on FPGA taking into account conventional technique costs significant equipment assets, which conflicts with the diminishing of circuit scale and increment of framework pace. FIR channels utilizing Distributed Arithmetic is utilized to build the asset use while pipeline structure is additionally used to expand the framework speed. Moreover, the isolated LUT strategy is additionally used to diminish the required memory units. FIR filter implemented using basic Distributed Arithmetic architecture is based on bit serial operation resulting in increase in delay with decrease in speed of operation. This is because the entire co-efficient are stored in single LUT.

In Parallel DA architecture, instead of storing the co-efficient in single LUT as in traditional DA architecture, it is split into several ROM LUT's. All the LUT's are provided with different inputs at the same time, implying parallel mechanism. This increases the speed of operation. The disadvantages are, as the number of inputs increases, required ROM structures proportionately increases thereby resulting in an increase in hardware utilization on FPGA. Also data stored in ROM cannot be altered.

Parallel DA architecture was improvised with split LUT and slight modification in the structure. Instead of using ROM for each input data's, LUT is fed with a set of 4 data's. This again resulted in parallel operation. The structure provides flexibility to alter the contents of LUT as against ROM usage.

The limitations of the structure were, as the number of inputs increased, required LUT structures proportionately increased thereby resulting in an increase in hardware utilization on FPGA, ensuing in reduction of speed of operation. Disseminated Arithmetic based methodologies for high-throughput reconfigurable execution of limited drive reaction channels whose channel coefficients change amid runtime, traditionally, has a downside that for reconfigurable DA-based usage of FIR channel, the lookup tables (LUTs) are required to be actualized in RAM and the RAM-based LUT is observed to be expensive for FPGA usage.

- Existing FIR filters are not reconfigurable.
- Higher power consumption.
- Device utilization for the design is to be improved.
- Excessive Memory utilization.
- Need to design a Reconfigurable architecture to meet different specifications.
- Need to improve on speed of computation.
- Throughput need to be improved.
- Need to reduce Power consumption.

METHODOLOGIES

The methodology followed for the research work is as follows:

The work started with the understanding of the basic working principle of FIR filter, the key components required for the design of a filter, the area and design complexity involved in the usage of multipliers in the filter. After a broad literature survey, multiplier less technique such as Distributed Arithmetic Algorithm was found as an efficient method for design of FIR filter. As a First step of research work, implementation of FIR filter using system generator was done. Verification of behavioral functionality of the

developed model was done. Implementation of FIR filter using Xilinx tool and verification of behavioral functionality of the developed model was completed. Also analysis for memory utilization for the filter was made. Basic FIR filter based on Distributed Arithmetic was designed and implemented on FPGA. In a conventional DA, as the contents of the Look up Table is fixed, if there is a need for change in the filter coefficients, whole of the architecture on FPGA, had to be changed which was again a major setback. To overcome the mentioned setback, improvement or modification on the existing algorithm was done which resulted in Reconfigurable Distributed Arithmetic architecture where the filter coefficients could be changed at the runtime without any change in the architecture of the filter. Comparative analysis of the developed design is provided to show that the implemented design can achieve the best results in terms of area and power utilization. Application area of the implemented FIR filter is also discussed.

LITERATURE SURVEY

In authors existing Back EMF detection is important aspect of the sensor less control of BLDC motor. In this paper, the BEMF detection using FPGA based digital low pass filtering of the line to line voltage using Least Pth-norm FIR filter has been realized. Lower switches of the legs are fed with PWM and upper switches are fundamentally controlled. The HON_LPWM strategy is used for the speed control as well as for the BEMF detection. Various quantities, e.g., percentage current ripple, current magnitude, line-to-line back EMF, speed, and switching signals were analyzed for the case of varying voltage with fixed duty as well as for varying duty cycles with fixed DC voltage. It has been analyzed that the back EMF detection with this method has some limitations on duty cycle because intermittent up and down peaks in filtered BEMF waveform become high as we increase or decrease duty out of certain band. During the implementation of BEMF based sensor less techniques, these up and down peaks in BEMF waveforms have to be removed. For this, further improved filtering has to be

searched to assure proper wave shaping with least delays in line-to-line BEMF zero crossings for sensor less application. Whole system has been implemented using Xilinx based System Generator using Vivado 2014.4 design suite for FPGA based programming in connection with MATLAB/Simulink and WAVECT controller. Simulation results and hardware results have been verified.

In authors existing the application of FPGA-based FIR filtering to increase the usable bandwidth of a piezoelectric transducer used in optical phase locking. We experimentally perform system identification of the interferometer with the cross-correlation method integrated on the controller hardware. Our model is then used to implement an inverse filter designed to suppress the low frequency resonant modes of the piezoelectric transducer. This filter is realized as a 24th-order FIR filter on the FPGA, while the total input-output delay is kept at 350 ns.

The combination of the inverse filter and the piezoelectric transducer works as a nearly flat response position actuator, allowing us to use a proportional-integral (PI) control in order to achieve stability of the closed-loop system with significant improvements over a non-filtered PI control. Finally, because this controller is completely digital, it is straightforward to reproduce. Our control scheme is suitable for many experiments that require highly accurate control of flexible structures.

FIR FILTER OPERATION

Digital filtering is executed in two ways, utilizing either FIR Filters Simulink. An FIR Filter is a crucial sort of channel that is utilized as a part of DSP over FIR channel. The reaction of such a channel is limited in view of the non-attendance of criticism. Applications of FIR filter includes video and communication systems because of guaranteed stability, absence of overflow oscillations, ability to implement filters with linear phase response, ease of

The FIR filter represented by Equation (2.1) has a general structure as shown in Fig 2.1.

$$Y(n) = \sum_{k=0}^{N-1} h(k).X(n - k)$$

implementation, computational efficiency achieved to great level, possible to implement filters with coefficients less than one.

Also FIR filters are the key ingredient of DSP which are used in the applications such as signal pre-processing, antialiasing, band selection, and interpolation (increasing the sampling rate) and low pass filtering. FIR filter becomes the best choice when there should not be any noise in the filter as there is no necessity of truncation or rounding of the bits after multiplication.

MATLAB based Simulink Tool outlines the model based chart and reproduction, mechanized code era and check of relating configuration on top of the line FPGA's. Channel Design and Analysis (FDA) Tool is an effective device in MATLAB Signal Processing Tool box, with the assistance of which we can outline and break down Simulink, products various sorts of filters. The late advance in programming apparatus improvement is to bolster DSP applications broadly in FPGA's. Framework Generator for DSP™ is the Industry's driving abnormal state apparatus for planning elite DSP frameworks utilizing FPGA's.

System Generator instrument gives Simulink libraries to outline Arithmetic, Logical, Mathematical, Memory squares and DSP capacities. The DSP capacities incorporate FIR Filters and Transforms. The outlining of a Filter basically involves two fundamental strides that finish the configuration procedure. The initial step is the era of coefficients and the second step constitutes the re-enactment of channel utilizing the created coefficients. In spite of the fact that FIR Filter outline is confound, the focal points persevering permits them to be broadly utilized for sifting applications when contrasted with FIR channels. FIR channels do not give security at higher requests while the FIR partners are constantly steady and are especially valuable for applications which require definite direct stage reaction. The FIR filter represented by Eqn (2.1) has a general structure as shown in Fig 2.1.

Where the output of the filter is, is the input data, are the filter coefficients and N represents number of filter coefficients or taps. The filter structure requires N multipliers and $(N-1)$ adders for the implementation. An FIR Filter structure of Fig 2.1 consists of multipliers, adders and delay elements. Implementation of multipliers on FPGA consumes large area; also, the design of multipliers is a complex task, which requires large resources resulting in high cost.

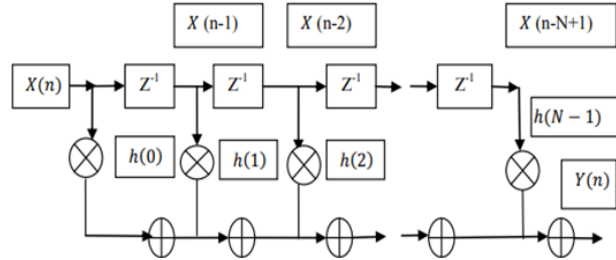


Fig 2.1 General FIR Filter structure

PROBLEM DEFINITION

FIR digital filter is widely used in several digital signal-processing applications, such as speech processing, loudspeaker equalization, echo cancellation, adaptive noise cancellation, and various communication applications, including software-defined radio (SDR) and so on. Many of these applications require FIR filters of large order to meet the stringent frequency specifications. Very often, these filters need to support high sampling rate for high-speed digital communication. The number of multiplications and additions required for each filter output, however, increases linearly with the filter order. Since there is no redundant computation available in the FIR filter algorithm, real-time implementation of a large order FIR filter in a resource-constrained environment is a challenging task. Filter coefficients very often remain constant and known a priori in signal processing applications. This feature has been utilized to reduce the complexity of realization of multiplications.

Several designs have been suggested by various researchers for efficient realization of FIR filters (having fixed coefficients) using DA and multiple constant multiplication (MCM) methods. DA-based designs use LUTs to store precomputed results to reduce the computational complexity. The MCM method on the other hand reduces the number of additions required for the realization of multiplications by common sub expression sharing, when a given input is multiplied with a set of constants. The MCM scheme is more effective, when a common operand is multiplied with more number of constants. Therefore, the MCM scheme is suitable for the implementation of large order FIR filters with fixed coefficients. But, MCM blocks can be formed only in the transpose form configuration of FIR filters.

Block-processing method is popularly used to derive high-throughput hardware structures. It not only provides throughput-scalable design but also improves the area- delay efficiency. The derivation of block-based FIR structure is straightforward when direct-form configuration is used, whereas the transpose form configuration does not directly support block processing. But, to take the computational advantage of the MCM, FIR filter is required to be realized by transpose form configuration.

EXISTING METHOD

In the world of smart technology, data security is a crucial entity for every individual. The techniques for securing data are in high demand and became challenging for the designers to come up with efficient ways based on the requirements. The expanding areas of networks and security demand more number of efficient algorithms for use in cryptography. Many researchers have done phenomenal work and existing various architectures with the advancements. Encryption is an essential criteria in cryptography for data handling and processing. When explicit information from one environment is to send to the other, then encryption of the data is necessary for security purposes. Cryptography provides pavement for this restraint.

It deals with encryption and decryption of the data such that the fundamental information transfer from the sender to the recipient remains confidential. The cryptographic algorithms which are used in advance systems has motive principles of authentication and data secrecy. Performing modular multiplication is one of the core operations in cryptography. Given two inputs P and Q of k bits each, where P is the multiplier and Q is the multiplicand and modulus N of k bits, modular multiplication is given by eq. (1), in which C represents the result of the multiplication. $P = p_{k-1} \dots p_1 p_0$ $Q = q_{k-1} \dots q_1 q_0$ $N = n_{k-1} \dots n_1 n_0$ $C = P * Q * \text{mod} (N)$ (1) In most of the cryptographic applications, modular multiplication algorithm tends to be a time-consuming process. Therefore, for the efficient functioning of the algorithm and to decrease time consumption, modular multiplication reduction is necessary.

CONVENTIONAL DIGIT-SERIAL MONTGOMERY MULTIPLIER

The conventional Montgomery multiplier [6] is constructed for m=6 and d=3, where m is the digit size of each input and d is the bit size. The first stage in computing the results of the multiplier is the processing stage. The partial product generation for the given multiplicand and multiplier takes place in this stage with the combination of And- network and Adder-network. The multiplicand and the multiplier are of 6-bit length each. The individual bit representation of multiplicand, multiplier and modulus N is given by eq. (3). The complete Montgomery multiplication algorithm takes place in 2 cycles. In the first cycle of the algorithm, 6 bits of multiplicand P and first 3 bits of multiplier Q are given as the inputs. In the second cycle, the next 3 bits of the multiplier Q will be acting as the inputs. To choose the bits of multiplier according to the cycle, 2:1 multiplexers are used. For each bit of the Q, a multiplexer is used and based on the selection line, the required bits are chosen. If the selection line is high for all the 3 multiplexers, first 3 bits of Q i.e., (q0, q1, q2) will be propagated as inputs. If the selection line is low, the next 3 bits of Q i.e., (q3, q4, q5) are propagated as inputs. In the first cycle of the algorithm, all the carry bits which are given as the inputs to the processing element are considered as zeros i.e, C=0. 6 bits of multiplicand and first 3 bits of the multiplier generates partial products as shown in eq. (4).

MODIFIED MONTGOMERY MULTIPLIER

The modified Montgomery multiplier consists of 3 stages as shown in Fig. (1). The first two stages perform the similar operation as in the conventional design [6]. The reduction cell in the conventional design is replaced with the existing 12:6 Astute Compressor, making the design into a modified one. This modification resulted in area and power efficient multiplier by reducing the complexity involved in the usage and design of multiplexers in the reduction cell of the conventional design and by declining the number of transistor count.

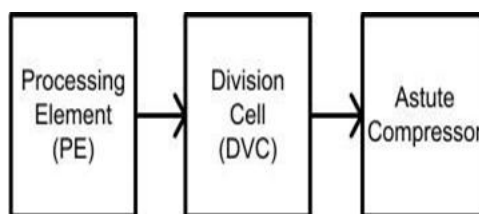


Fig. 1: Block Diagram of Modified Montgomery Multiplier

The modified Montgomery multiplier design works for any 6-bit length input P and Q, but the value of N is restricted to 3 and 7. The results produced by the multiplier are accurate. Using 45 nm technology in the Cadence Virtuoso tool, the design of modified Montgomery multiplier with a combination of CMOS and PTL logic [7] resulted in low power and area efficient design. The existing astute compressor

lessened the number transistors to a notable amount.

I. EXISTING 12:6 COMPRESSOR: ASUTE COMPRESSOR

The compressor is a circuit that is used to depreciate the area by down turning the transistors count and reduce the overall power and delay for more reliable performance of a design [8]. This paper presents a 12:6 Astute Compressor which compresses 12-bit input to



6-bit output as shown in Fig. (3). The compressor consists of a Half Adder (HA) and a series of Modified Full Adders (MFA).

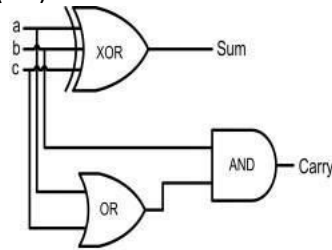


Fig. 2: Modified Full Adder (MFA)

The modified full adder works for all the input combinations except for the input combination of 101 which is binary equivalent of 5. All the adders are connected in series and the Carry of each adder will be propagated to the next successive adder as one of the inputs.

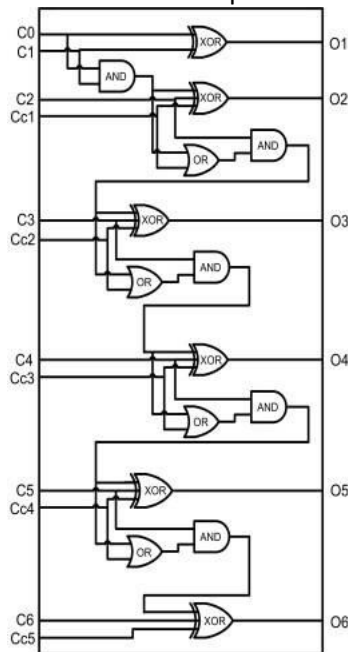


Fig. 3: Proposed 12:6 Compressor: Astute Compressor
Types of Compressors

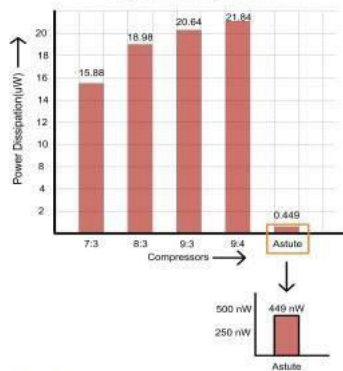


Fig. 4: Power Comparison of Various Compressors

The other inputs to the adders are the previous stage Carry bits C0 to C6 and Cc1 to Cc5. Propagation of the Carry bits to the next successive adder takes place till the end. The final adder in the compressor consists of only

Sum part and the Carry part is neglected for the exact output of 6 bit length. The output of the compressor is the final output of the multiplier after completion of 2-nd cycle. The Sum bits of each adder will constitute the output bits. The



first adder Sum bit is the Least Significant Bit (LSB) of the final result, and the last adder Sum bit is the Most Significant Bit (MSB). The design of the compressor has been done with a combination of PTL and CMOS logic style with 78 transistors, which is equal to 42% reduction in the transistor count by CMOS logic style. The compressor produces an average power of 0.449 uW and a delay of 4.455 ns. The comparison of the existing compressor with the other compressors on FPGA [9] in terms of power, delay, and number of logic gates used is shown in Table I. The graphical representation of power and delay comparison of the existing astute compressor with the others compressors are shown in Fig. (4), Fig. (5) respectively.

EXISTING MONTGOMERY MULTIPLIER DESIGN USING ASTUTE COMPRESSOR

The existing multiplier is verified with the other multipliers designed using different techniques to demonstrate its effectiveness.

1) Analysis of Existing Multiplier in terms of Power Dissipation and transistor count:

The existing multiplier has been compared with the various multipliers, designed using different techniques in terms of power and number of transistor count as shown in table

II. The usage of pass transistor logic in combination with conventional CMOS logic resulted in yielding low power and area efficient design.

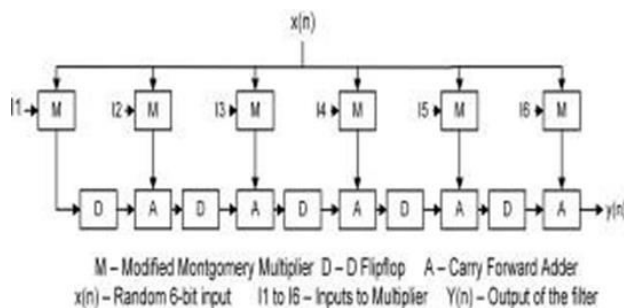


Fig. 7: 5th Order Low Pass FIR Filter

$y(n) = I1.x(n) + I2.x(n-1) + I6.x(n-5)$ (5) The general equation of the filter is given by eq. (5). The filter design consists of multipliers, D-flipflops, and a carry-forward adder [15] as the main elements as shown in Fig. (7). D-flipflop is used in order to reduce the critical path delay of the filter. The carry forward adder which is used in the filter consists of a half adder and full adders as its functional elements in which carry of each adder is propagated to the next adder in series. The inputs to the adder are the multiplier output and the D-flipflop output each of 6-bit length each. The adder produces a 6-bit output, and the final output of the circuit is the output of last adder present in the filter. The complete filter design has been done with 4254 transistor count and the filter produced an average power of 216 uW and a delay of 1.09 ns

PROPOSED METHOD

Finite impulse response (FIR) filters are performed vital role in signal processing by

virtue of its high stability and linear phase response. It also have high capacity, fast computation and in small size. Hardware efficient FIR filters are very much needful in various real time applications. In this context, the outstanding features of FPGAs such as high flexibility, high speed due to inherent parallelism and low development cost make the suitable selection in hardware industry. Digital filters are extensively used due to its better accuracy, high precision and more reliable as compare with its analog filters. Generally, two types of digital filters are available namely FIR and IIR. Digital FIR filters are stable, linear in phase and can be efficiently implemented in reconfigurable platform.

The limitation of IIR filters is only applicable in particular area. There are several methodologies present to improve FIR filter performance. Pipeline and parallel processing are the techniques to increase operating speed

of the FIR architecture at cost of hardware complexity and latency . Pipeline is basically inserting a register in each stage and parallel processing is the method to generate multiple inputs-outputs in one clock cycle. Another procedure to obtain high performance FIR filter is the process of Distributed Arithmetic (DA) technique which is basically multiplier-less architecture. DA based FIR filters in FPGA are cost effective. Systolic FIR filter is efficiently realized in hardware domain. The characteristics of systolic architecture are simplicity, modularity, regularity and controlled by a global clock with specific duty cycle . Systolic based FIR filters have maximum performance using pipeline and parallel processing technique.

The numerous advantages of systolic architecture incorporate in FIR architectures make a highly demand in signal processing applications. In literature, various FIR architectures have been realized, like direct FIR structure; transposed form, parallel FIR filter and multi-channel FIR filter. In article, the authors proposed Time Division Multiplexer (TDM) based multichannel FIR filter using concept of resource sharing principle. This technique reduced area with operating frequency up to 480MHz. Highorder systolic based FIR filters in FPGAs were presented in literature. The structure saved the slices and improved operating frequencies up to 526 MHz for 90 taps FIR filter using DA. The systolic based FIR architecture using DA technique is

being proposed in article [8]. The design implemented in FPGA and measured the less power-delay product. This paper presents hardware efficient single-channel symmetric systolic FIR (SSSFIR) filter and multi-channel symmetric systolic FIR (MSSFIR) filter. Multi-channel FIR filter operates in multiple data stream using same coefficients or different coefficients for all channels. Symmetric coefficients in FIR implementation save hardware resources. Key performance parameters of FPGAs are comparable between existing and proposed FIR filters. Implementation of the proposed FIR filters show that great reductions of slices (area) with high processing speed.

I. REALIZATION OF FIR FILTERS

The following sub-sections describe different types of FIR architectures.

A. Direct form FIR filter

An FIR filter basically consists of adders, multipliers and delay blocks which are called MAC units. A direct form FIRfilter is adder tree structure to increase hardware resource. The conventional FIR filter. The input sequence $x[n]$ are multiplied by N-bit coefficients of h_i and produce output sequence $y[n]$. The coefficients values control the characteristic of the FIR filter like low-pass filter, high-pass filter etc. The latency is measured in the order of \log_2 (taps number) and number of DSP blocks require for this system is (taps number/2). Fig. 1 shows FIR architecture using direct form.

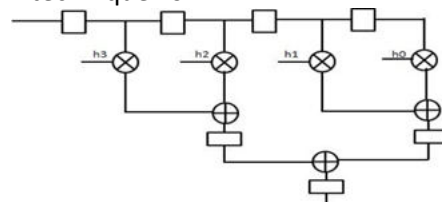


Fig. 1. Direct for FIR filter

B. Systolic FIR Filter

The systolic FIR filters normally are an optimum solution of direct form FIR filters. The systolic architecture has multiplier-adders chain to fully utilized pipeline FIR structure [9]. The input data are fed into series of one unit delay flipflops or registers which are representing as buffer units. The adder chain accumulates the inner products

and produces the final output. Fig. 2 shows systolic FIR filters with four coefficients in form of DSP blocks. The DSP blocks optimize design implementation with improve in latency. Due to absent of adder tree, this technique saves the hardware resources than direct form FIR architecture.



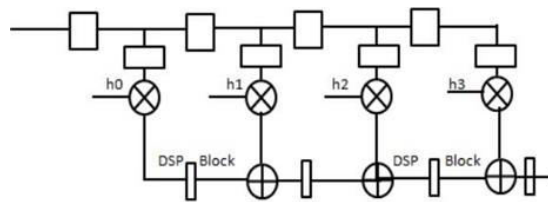


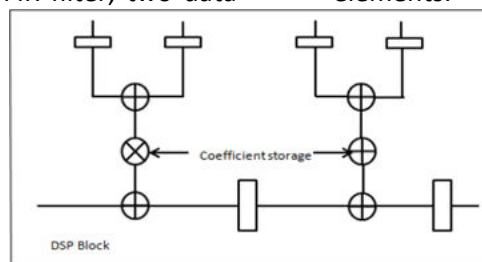
Fig. 2. Systolic FIR filter

The parameter such as latency is measured by tap number and the DSP blocks are half of the tap number. Systolic FIR filter consumes N slices for N coefficients.

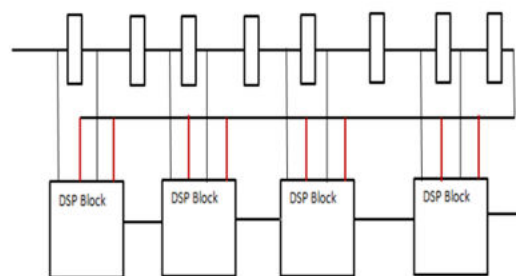
C. Single-channel Symmetric Systolic FIR Filter (SSSFIR)

In symmetric coefficient sequence of FIR filter can save hardware resource for its implementation stage [10]. The number of multipliers can halve by adding data before multiplication and this process allows twice sampling rate of the filter (assume same clock frequency). In a symmetric FIR filter, two data

samples are added and then multiply with a coefficient. Hence, two multipliers are replaced by one multiplier [11]. The resource utilization of the symmetric systolic FIR filter is calculated as Slices = $(n+1) * (N/2) + (n/2)$ (3) Where n and N represents the input bit width and coefficients number of the FIR filter respectively. The latency is the half of the number of taps and the numbers of DSP blocks are one fourth of the number of taps. Fig. 3 shows therealization of SSSFIR filter with 18-bit coefficients. The white boxes represent delay elements.



a) Basic DSP Block of SSSFIR filter



b) 16-tap SSSFIR filter with 4-DSP block

Fig. 3. SSSFIR filter

D. Multi-channel Symmetric Systolic FIR Filter (MSSFIR):

Multi-channel FIR filters have multiple data stream which are frequently required in signal processing. In this process, the same hardware resources are applicable for multiple applications. Considering a 4-channel FIR filter, the clock speed should be 4- times of the

incoming data in each channel. Similarly, the N -channel FIR filter clock speed would be N times for a single channel data rate. The clock speed initiate to compute the data output for every channel of the FIR filter. Fig. 4 shows a 16-tap MSSFIR filter. The numbers of channels are always less than the number of delay elements. Each rectangular box represents a delay

element and the red boxes represent the one element. unit more delay than black boxes delay

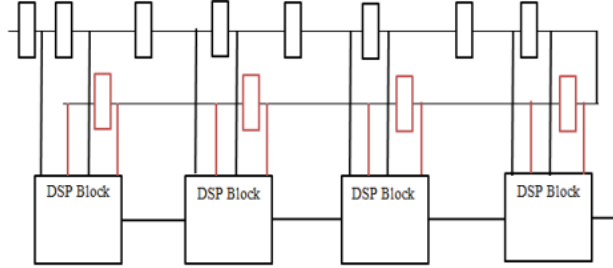


Fig. 4. 16-tap MSSFIR filter

In addition, symmetric coefficients reduce computation time and multiplier units. Hence, implementation of proposed MSSFIR architecture is less resources utilization inFPGA domain and performs high operating frequency as compare with other architectures likeDA FIR, Parallel FIR and simple systolic FIR filters.

SIMULATION RESULTS

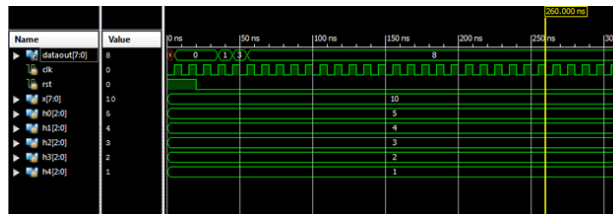


Figure 6.1 Simulation outcome

The figure 6.1 represents the simulation outcome of N-stage fir filter. Here, Ho, h1, h2, h3 and h4 are the impulse inputs, x is the data input and resultant outcome is stored into data out.

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	32	408000	0%
Number of Slice LUTs	116	204000	0%
Number of fully used LUT#FF pairs	22	126	17%
Number of bonded IOBs	33	600	5%
Number of BUFG/BUFGCTRLs	1	32	3%

Figure 6.2:Design summary

Figure 6.2 represents area utilization by the proposed method; out of available 408000 slice registers, only 32 are used. Out of available 20400-slice look up tables, only 116 are used, so the area consumption is reduced.

Cell:in->out	Fanout	Gate Delay	Net Delay	Logical Name (Net Name)
IBUF:I->O	11	0.000	0.475	h0_1_IBUF (h0_1_IBUF)
LUT3:I0->O	3	0.043	0.466	B1/C1/F2/Mxor_sum_xo<0>211 (B1/C1/F2/Mxor_sum_xo<0>2)
LUT5:I1->O	6	0.043	0.573	B1/C1/F2/Mxor_sum_xo<0>23 (B1/C1/F2/Mxor_sum_xo<0>2)
LUT6:I0->O	2	0.043	0.546	B1/C1/F1/b_cin_and_2_01 (B1/carry1)
LUT5:I0->O	3	0.043	0.299	B2/C2/F1/ou01 (B2/carry2)
LUT5:I4->O	3	0.043	0.299	B2/C3/F1/ou01 (B2/carry3)
LUT5:I4->O	3	0.043	0.299	B2/C4/F1/ou01 (B2/carry4)
LUT5:I4->O	3	0.043	0.299	B2/C5/F1/ou01 (B2/carry5)
LUT5:I4->O	3	0.043	0.466	B2/C6/F1/ou01 (B2/carry6)
LUT5:I1->O	1	0.043	0.428	d2<6>1 (d2<6>)
LUT4:I1->O	1	0.043	0.550	dataout<7>9 (dataout<7>9)
LUT6:I0->O	1	0.043	0.279	dataout<7>10 (dataout_7_OBUF)
OBUF:I->O		0.000		dataout_7_OBUF (dataout<7>)
Total		5.454ns (0.473ns logic, 4.981ns route)		(8.7% logic, 91.3% route)

Figure 6.3 Time summary

Figure 6.3 represents time utilization by the proposed method, and the existing method consumes total 5.45ns of path delays, which includes 0.473ns logical delays and 4.981ns of route delays, respectively.



A	B	C	D	E	F	G	H	I
Device			On-Chip	Power (W)	Used	Available	Utilization (%)	
Family	Virtex4		Logic	0.000	63	10944	1	
Part	xc4vfx12		Signals	0.000	132			
Package	ff668		DCMs	0.000	0	4	0	
Temp Grade	Commercial		I/Os	0.000	97	320	30	
Process	Typical		Leakage	0.165				
Speed Grade	-10		Total	0.165				
Environment			Thermal Properties		Effective TJA	Max Ambient	Junction Temp	
Ambient Temp (C)	50.0				(C/W)	(C)	(C)	
Use custom TJA?	No				9.3	83.5	51.5	
Custom TJA (C/W)	NA							
Airflow (LFM)	250							
Characterization								
PRODUCTION	v1.0.02-02-08							

Figure 6.4: Proposed power summary

Figure 6.4 represents power utilization by the proposed method, and the existing method consumes total 0.165 watts of power, respectively.

CONCLUSIONS & FUTURE ENHANCEMENTS

CONCLUSIONS

This project briefs prototype for implementing systolic-based symmetric FIR filter using multi-channel process. The proposed FIR filter is highly stable and provides excellent computational speed. The proposed architecture reduces total number of slices which indicate optimum area and also improves maximum frequency response. In comparison with earlier FIR filters suggested that the proposed symmetric multi-channel systolic FIR filter offers substantially reduction of hardware resources and improves the operating speed. In future, the modified algorithm uses limited numbers of slices on FPGA device and these can be suitable in modern communication system. In addition, the proposed solution can be used in signal processing applications.

FUTURE WORKS

Distributed Arithmetic (DA) based method is often preferred since it eliminates the need for hardware multipliers and is capable of implementing large filters with very high throughput. The most widely used design metrics to quantify the performance of an FIR filter are high throughput/speed (maximum frequency), area (occupied slices), Latency (clock cycles), power (milli watts). Bit serial DA algorithm computes the dot product of two vectors by a sequence of Look-Up Table (LUT) accesses followed by shift- VI accumulation operations of the LUT output. In this dissertation, an earnest effort has been made

to design four Field Programmable Gate Array (FPGA) based FIR filter architectures using DA with the goal of minimizing the area delay complexity. Against this background, a novel LUT-less architecture for the DA based FIR filter with the carry save adder and a modified left shift accumulator is developed that yields a higher speed and significant reduction in the area over the existing LUT-less designs. For high-speed applications, carry-save adders are a better choice than ripple carry and carry look ahead adders. The ensuing work of this dissertation is focused to further augment the speed of FIR filters. Two innovative architectures for DA based FIR filter using a new bit-serial shift accumulator was devised to meet the above perspective. The modified shiftaccumulator composed of pipelined bit serial adders reduces the critical path, thereby yielding a higher speed of operation with a small escalation in the number of occupied slices. The highlight of the dissertation has been the innovation of a systolic architecture for the FIR filter for applications that require medium speed with moderate area resources. This research work is further extended to develop a new architecture for high-speed FIR filter, with linear rows of processing elements and a new-pipelined shift accumulator tree that results in low latency and fewer hardware resources. The performance of the existing architectures has been substantially validated through their implementations on the Xilinx FPGA platform, and a detailed comparison of the design metrics with the existing vii methods is also presented. The results exhibit the important characteristics of each architecture in view of their performance, and the selection of a better



architecture for an application based on the requirements of modern digital technology era.

BIBLIOGRAPHY

- [1].Soni, Umesh Kumar, Maloth Naresh, and Ramesh Kumar Tripathi. "FPGA Based Speed Control and Back EMF Extraction from Line Voltages Using FIR Digital Filters for BLDCM." *Advances in Power and Control Engineering*. Springer, Singapore, 2020. 41-62.
- [2].Okada, Masanori, et al. "Extending the piezoelectric transducer bandwidth of an optical interferometer by suppressing resonance using a high dimensional FIR filter implemented on an FPGA." *Review of Scientific Instruments* 91.5 (2020): 055102.
- [3].Okada, Masanori, et al. "Extending the PZT bandwidth of an optical interferometer by suppressing resonance using a high dimensional FIR filter implemented on an FPGA." *arXiv preprint arXiv:1912.10707* (2019).
- [4].Mahabub, Atik. "Design and implementation of cost-effective FIR filter for EEG signal on FPGA." *Australian Journal of Electrical and Electronics Engineering* (2020): 1-9.
- [5].Khurshid, Burhan, et al. "Achieving Performance Speed-Up in FPGA Based FIR Filters Using DSP Macro Blocks." *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*. IEEE, 2019.

- [6].Sudharsan, R. Raja, and J. Deny. "Field Programmable Gate Array (FPGA)-Based Fast and Low-Pass Finite Impulse Response (FIR) Filter." *Intelligent Computing and Innovation on Data Science*. Springer, Singapore, 2020. 199-206.
- [7].Balachandran, G., and Praveen Kumar Gupta. "FPGA-Based Electrocardiography Signal Analysis System using (FIR) Filter." *International Journal* 8.1 (2020): 17- 19.
- [8].Shahbaz, Muhammad Mateen, Aasim Wakeel, and Bahram Khan. "FPGA Based Implementation of FIR Filter for FOFDM Waveform." *2019 2nd International Conference on Communication, Computing and Digital systems (C-CODE)*. IEEE, 2019.
- [9].Khan, Mansoor, and Shahrukh Agha. "Least squares linear phase FIR filter design and its VLSI implementation." *Analog Integrated Circuits and Signal Processing* (2020): 1-11.
- [10]. Zhang, Ning, et al. "FPGA-Based Implementation of Reconfigurable Floating-Point FIR Digital Filter." *International Conference in Communications, Signal Processing, and Systems*. Springer, Singapore, 2019.