Journal of Engineering Sciences
Vol 13, Issue 06, June/2022
ISSN NO:0377-9254

FPGA IMPLEMENTATION OF SYMMETRIC SYSTOLIC FIR FILTER USING MULTI-CHANNEL TECHNIQUE

1RAMIJABI SHAIK,2ANJANEYULU YERUGUDIDINNE,3PATTAN AYUB KHAN,4 Dr.M.M.RAGHAVENDRA, 5Dr.G.LAKSHMINARAYANA
123B.Tech Student, 45PROFESSOR
DEPT OF ECE
SVR ENGINEERING COLLEGE, NANDYAL

ABSTRACT

High performance Finite Impulse Response (FIR) filters are extensively used in digital signal processing (DSP), communications, image processing and many more areas. This paper 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 and then implemented on Virtex-5 FPGA board. The design method shows a great improvement of maximum operating frequency and save the area as compared to the earlier architectures.

I.INTRODUCTION

1.1Objectives

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 basically 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.

1.2 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.

1.3Methodologies

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 behavioural functionality of the developed model was done. Implementation of FIR filter using Xilinx tool and verification of behavioural 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.

II.LITERATURE SURVEY

In [1] authors existing Back EMF detection is important aspect of the sensorless 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 *P*th-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 sensorless techniques, these up and down peaks in BEMF waveforms have to be removed. For this, further improved filtering has to be searched so as to assure proper wave shaping with least delays in line to line BEMF zero crossings for sensorless 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 [2] 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.

III.PROBLEM DEFINATION

FIR digital filter is widely used in several digital signal processing applications, such as speech processing, loud speaker 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.

Apart from that, transpose form structures are inherently pipelined and supposed to offer higher operating frequency to support higher sampling rate. There are some applications, such as SDR channelizer, where FIR filters need to be implemented in a reconfigurable hardware to support multistandard wireless communication. Several designs have been suggested during the last decade for efficient realization of reconfigurable FIR using general multipliers and constant multiplication schemes. A RIIR filter architecture using computation sharing vector-scaling technique has been existing. A canonic sign digit (CSD)-based RIIR filter, where the nonzero CSD values are modified to reduce the precision of filter coefficients without significant impact on filter behavior. But, the reconfiguration overhead is significantly large and does not provide an area-delay efficient structure. The architectures are more appropriate for lower order filters and not suitable for channel filters due to their large area complexity.

Constant shift method (CSM) and programmable shift method have been existing for RIIR filters, specifically for SDR channelizer. The existing multiplier-based structures use either directform configuration or transpose form configuration. But, the multiplier-less structures of use transpose form configuration, whereas the DA-based structure of uses direct-form configuration. But, we do not find any specific block-based design for RIIR filter in the literature. A block-based RIIR structure can easily be derived using the scheme existing. But, we find that the block structure obtained from and is not efficient for large filter lengths and variable filter coefficients, such as SDR channelizer. Therefore, the design methods existing are more suitable for 2-D FIR filters and block least mean square adaptive filters.

IV.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 [1]. Many researchers have done phenomenal work and existing various architectures with the advancements [2]. 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 = pk−1.....p1p0 Q = qk−1.....q1q0 N = nk−1.....n1n0 C = P ∗ Q ∗ 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 [3].

As the Montgomery multiplication speedup modular multiplication algorithm, it is one of the adaptable choices for the designers as it evades trial division, which is the strenuous algorithm for integer factorization. Embedding the Carry-Save Adders (CSA) reduces the critical path delay [4]. Performing addition and division arithmetic operations by the powers of two substitutes the trail division in Montgomery multiplication. Computing multiplications take a transformation from time-consuming to time-saving process with the Montgomery multiplication as there are no division and subtraction operations involved.

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). The outputs of the processing element are the Sum and Carry bits of full adders that are present in the adder network, and they will be driving the next stage. P = (pm−1, pm−2.....p0) Q = ((qm−1, qm−2.....qm−d),(qd−1, qd−2.....q0)) N = (nm−1, nm−2.....n0) C = 0 (3) C = X d i=0 mX−1 j=0 pj qi + C (4) The second stage is the division stage. An array of full adders performs the division operation in the algorithm. Bits of N operates as one of the inputs to the full adders and the other inputs are Sum and Carry bits of the adders involved in the successive stages. The outputs of adders after performing the division operation are the Carry bits from C0 to C6 and Cc1 to Cc6. The Carry bits will be acting as inputs to the final stage of the design. 3 D-flipflops in which 2 are of 6 bit input size and 1 single D-flipflop are used to store Carry bits generated from the division cell and to give them back as inputs to the processing element in the second stage. The usage of D-flipflops in the circuit reduces the critical path delay. A D-flipflop works under the presence of a clock signal and operates with a delay in input by one clock cycle. After the generation of Carry bits, the second cycle starts and in the second cycle of the multiplier, the next 3 bits of multiplicand and the Carry bits from the D-flipflop will repeat the process and produces the final output of the multiplication from the compressor. The output produced is of 6 bit length which is equal to the bit size of the given inputs to the multiplier.

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.

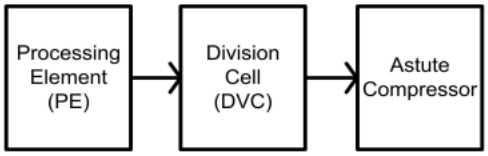


Fig. 1: Block Diagram of Modified Montgomery Multiplier

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.

EXISTING 12:6 COMPRESSOR: ASUTE COMPRESSOR

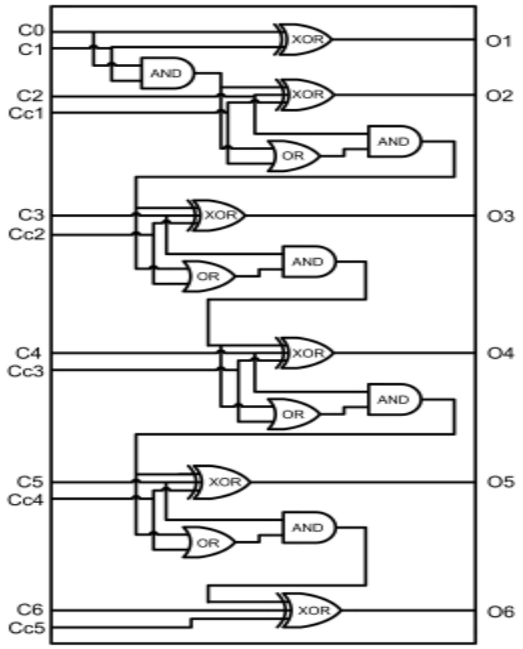


Fig. 3: Proposed 12:6 Compressor:Astute Compressor

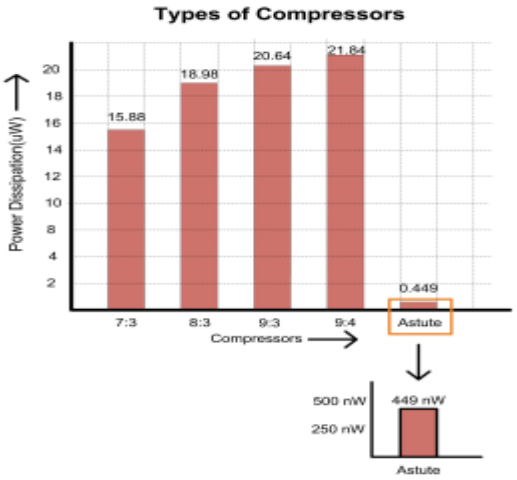


Fig. 4: Power Comparison of Various Compressors

V.PROPOSED METHOD

Finite impulse response (FIR) filters are performed vital role in signal processing by virtue of its high stability and linear phase response [1]. 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 [2]. 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 [3] [11] [12]. 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 [4]. 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 [5]. 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 [6], 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. High order systolic based FIR filters in FPGAs were presented in literature [7]. 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.

II. 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 FIR filter is adder tree structure to increase hardware resource. The conventional FIR filter. The input sequence x[n] are multiplied by N-bit coefficients of hi 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 log2 (taps number) and number of DSP blocks require for this system is (taps number/2).

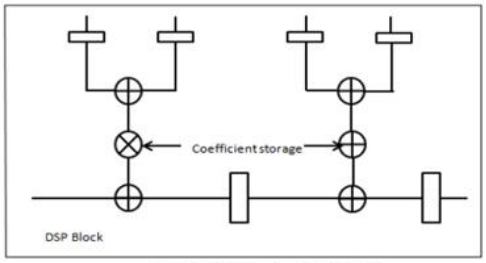
B. Systolic FIR Filter

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

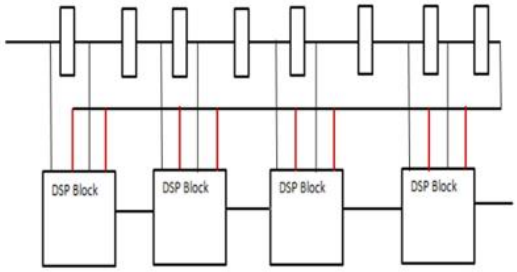
The parameter such as latency is measured by tap number and the DSP blocks are half of the tap number. Systolic FIR filter consumes N slices for N coefficients.

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 the realization 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. 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 Ntimes for a single channel data rate. The clock speed initiates 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 unit more delay than black boxes delay element.

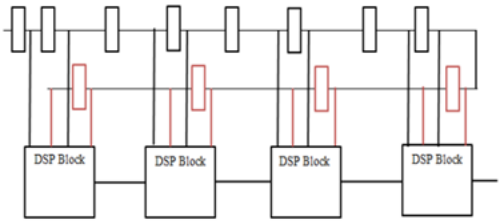


Fig. . 16-tap MSSFIR filter

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

VI.SIMULATION RESULTS

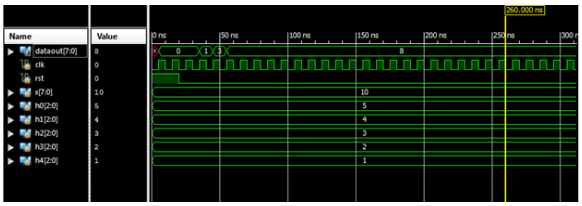


Figure : Simulation outcome

The figure 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 : Design summary

Figure 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.

```
Data Path: h0<1> to dataout<7>
                                  Gate     Net
  Cell:in->out      fanout   Delay   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_o1 (B1/carry1)
   LUT5:I0->O             3   0.043   0.299  B2/C2/F1/cout1 (B2/carry2)
   LUT5:I4->O             3   0.043   0.299  B2/C3/F1/cout1 (B2/carry3)
   LUT5:I4->O             3   0.043   0.299  B2/C4/F1/cout1 (B2/carry4)
   LUT5:I4->O             3   0.043   0.299  B2/C5/F1/cout1 (B2/carry5)
   LUT5:I4->O             3   0.043   0.466  B2/C6/F1/cout1 (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: Time summary

Figure represents time utilization by the propsoed 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.

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	IOs	0.000	97	320	30
Process	Typical	Leakage	0.165			
Speed Grade	-10	Total	0.165			

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	9.3	83.5	51.5

Characterization	
PRODUCTION	v1.0,02-02-08

Figure : Proposed power summary

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

VII.CONCLUSIONS

This paper 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 designs have been described in Verilog HDL and Virtex-5 Xilinx FPGA board is the target device for implementation. 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 areadelay 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 shift accumulator 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.

BIBILOGRAPHY

[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.