FPGA Realization of High Speed FIR Filter based on Distributed Arithmetic

K.G.Shanthi [#1], Dr.N.Nagarajan [*2], C.Kalieswari [#3]

[#] Department of Electronics & Communication, R.M.K Engineering College, Chennai, India
[1] kgs.ece@rmkec.ac.in
[3] ckalieswari@yahoo.com
[*] Department of ECE, Coimbatore Institute of Engineering and Technology, Coimbatore, India
[2] swekalnag@gmail.com

Abstract—Two high speed architectures for Distributed Arithmetic (DA) based Finite impulse response filter (FIR) using a new shift accumulator are presented in this paper. The proposed shift accumulator (SA) composed of pipelined bit serial adder results in very high speed compared with existing left shift and right shift accumulators. First design is a DA look up table (LUT) based FIR filter with and without partitioning using the proposed shift accumulator. Second is a systolic array architecture for DA based FIR filter with proposed SA. Both the architectures were implemented using Xilinx Virtex 6vlx240tff1156-1 device. Number of slices, minimum period and maximum frequency were the performance metrics obtained for different filter orders for both the architectures and the results reveal that both the designs have yielded significant improvement in speed.

Keyword-DA, FIR, Look up table, Shift accumulator, Bit serial adder, Multiply and Accumulate

I. INTRODUCTION

The most fundamental part used in many digital signal processing (DSP) applications is a finite impulse response filter because of its linear phase, stability and regular structure [1]. Designing a high-speed and hardware efficient FIR filter is a very challenging task as the complexity increases with the filter order. A higher order filter results in a sharper transition between a pass band and a stop band. Higher order filters are needed in many fields of signal processing such as image processing, speech processing, digital communication [1], [2] etc. The number of multiply and accumulate (MAC) operations required per filter output increases with filter order and hence higher order filters using multipliers occupy a large chip area and need high computation time. Multiplier-less memory based techniques have gained popularity over past two decades due to their high throughput processing capability and reduced dynamic power consumption. Memory based architectures are classified into direct read only memory (ROM) based architecture and distributed arithmetic. The direct ROM based implementation does the multiplication of inputs with the fixed coefficients by using LUTs that stores all possible precomputed product values corresponding to the input sample which results in faster output compared with the MAC based designs because memory access time is much lesser than multiplication time . DA is a bit serial operation which performs the inner product of two vectors by storing all possible intermediate computations in a LUT that is read by input vector followed by the shift accumulation operation. The advantage of FIR filters based on DA is that the time complexity depends only on the input word length and is independent of the order (N) of filter. These filters are implemented on field programmable gate arrays (FPGA) due to their high flexibility with the option to reconfigure, time-to-market, cost and performance [3].

DA algorithm for digital filter implementations was proposed by Croisier et al [4] in 1973 and a detailed discussion of DA was given by Abraham Peled and Bede Liu in 1974 at the Arden house workshop on digital signal processing [5]. A tutorial review on applications of distributed arithmetic to digital signal processing was given by S.A.White [6]. A review of the various memory based architectures for the implementation of FIR filters was given by Shanthi et al [7].The main drawback of DA method is that the memory size (2^N) grows exponentially as the filter order N increases. With the use of offset binary coding(OBC) the memory size can be reduced by half to 2^{N-1} words [2], [6],[8]. If a single term inside the LUT is relocated outside the LUT, then the lower half of the LUT is mirrored version of the upper half of the LUT with only the signs reversed which results in reducing the LUT size from 2^{N-2} to 2 in distributed arithmetic with modified offset binary coding (DA-MOBC) [9]. A LUT-less DA architecture achieved by recursive LUT reduction with multiplexers and ripple carry adders was given by H. Yoo and D. V. Anderson [10]. Area-efficient FIR filter design was proposed by Patrick Longa et al where the input sequence is reordered to implement a modified version of the shift accumulator stage [11].To reduce the memory-size of DA-based filters several memory-partitioning and multiple memory bank approaches along with flexible multi-bit data access mechanisms were presented [7], [12], [13]. P.K.Meher et al suggested an area-delay-power efficient implementation of FIR filter by systolic decomposition of DA based inner-product computation [14]. The main features of systolic design being their regularity, modularity of the structure and also produce high throughput by using pipelining or parallel

processing [15]. FPGA realization of FIR filters for high-speed and medium-speed by using modified DA architectures were suggested by Jiafeng Xie et al., using pipelined registers and pipelined shift adder tree [16].

The remaining part of the paper is organized as follows: Section II involves a brief overview of conventional DA. Section III explains the modified architecture for conventional DA based FIR filter using the proposed shift accumulator without and with decomposition. Section IV explains the modified systolic architecture for DA based FIR filter with the proposed shift accumulator. FPGA implementation and comparison of performance metrics of the proposed architecture with the existing methods is detailed in section V. Conclusion is presented in section VI.

II. CONVENTIONAL DISTRIBUTED ARITHMETIC (DA)

The output y[n] of an N- tap discrete-time linear finite impulse response filter is represented as

$$y[n]=\sum_{i=0}^{N-1} C_i x\left[n-i\right] \tag{1}$$

where C_i represents the fixed filter coefficients, x[n-i] is the input data which varies at every sampling instant. The input sample of the FIR filter is coded as B-bit 2's complement binary number given by

$$x[n-i]=-x_{i0}+\sum_{j=1}^{B-1} x_{ij}2^{-j} \tag{2}$$

where $x_{i,j} \in \{0, 1\}$, x_{i0} is the sign bit and $x_{i,B-1}$ is the Least significant bit (LSB).Substituting (2) in (1) and changing the order of summations, the output can be expressed as

$$y[n]=\sum_{i=0}^{N-1} C_i\left(-x_{i0}+\sum_{j=1}^{B-1} x_{ij}2^{-j}\right) \tag{3}$$

$$y[n]=-\left(\sum_{i=0}^{N-1} C_i x_{i0}\right)+\sum_{j=1}^{B-1}\left(\sum_{i=0}^{N-1} C_i x_{ij}\right)2^{-j} \tag{4}$$

For a given set of coefficients C_i (i = 0, 1, 2,…., N − 1), the terms in the brackets may take one of 2^N possible values that can be precomputed and stored in a LUT that can be read out from the ROM using the N bit sequence $\{x_{i,j}$ for $0 \le i \le N\}$ as address bits. These intermediate results are accumulated in B clock cycles to produce one filter output y[n]. Conventional LUT based design of a 4-tap (N =4) FIR filter consists of three units: Input shift register unit, Look up table unit and Shift accumulator unit as shown in Fig.1.

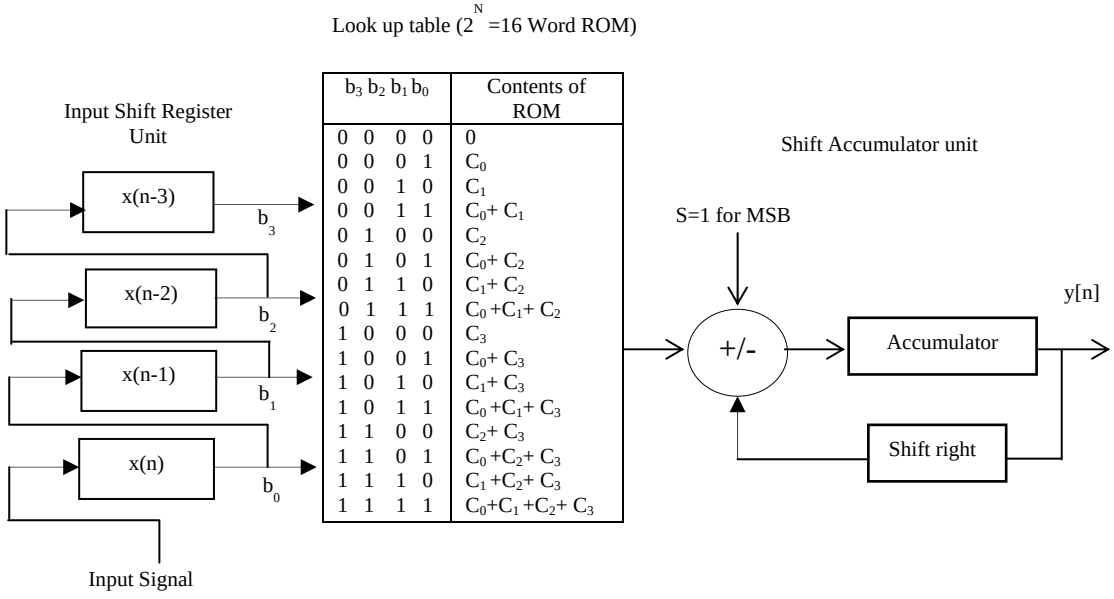


Fig.1. LUT-based design of a 4-tap (N =4) FIR filter using DA

The conventional shift accumulator shown in Fig.1 performs a shift right and add operation at every clock cycle and a subtraction in the last time slot called sign-bit time. The input signal given to the input shift register unit starts first with the least significant bits and the corresponding output read out from the LUT is fed as input

to the shift accumulator. The computation performed by conventional right shift accumulator (RSA) [4]-[6] is depicted in Fig.2. The computation performed by left shift accumulator (LSA) [11] is depicted in Fig.3 where the prerequisite being that the input signal given to the input shift register unit must start first with the most significant bits.

```
Step1: Initialize
         B=input length, count=0,
          Acc=0, Yin=0, Yout=0;
       End Intialization
Step2: Yin=LUT output
       If count=B-1 then
          Acc=(Acc>>1)-Yin,
          Yout [Count] = Acc [0],
          count = count+1;
       Else
          Acc=(Acc>>1)+Yin,
          Yout [Count] = Acc [0],
          count = count+1;
       Endif
Step3: If count=B then go to step 1
       Else go to step 2.
```

Fig.2. Algorithm for conventional right shift accumulator (RSA)

```
Step1: Initialize
         B=input length, count=0,
         Acc=0, Yin=0, Yout=0;
       End Intialization
Step2: Yin=LUT output
       If count=0 then
          Acc=(Acc<<1)-Yin,
          count = count+1;
       Else
          Acc=(Acc<<1)+Yin,
          count = count+1;
       Endif
Step3: If count=B then
       Yout= Acc , go to step1
       Else go to step 2.
```

Fig.3. Algorithm for conventional left shift accumulator (LSA)

III.DA BASED FIR FILTER WITH THE PROPOSED SHIFT ACCUMULATOR

A. Proposed Bit serial shift accumulator(BSA)

It is composed of pipelined bit serial adders. The building blocks of a bit serial adder [8] are a full adder and a D flip-flop as shown in Fig.4. The D flip-flop is reset at the beginning of the computation. Bit serial adder is also called as carry save adder as the carries are saved from one bit position to the next.

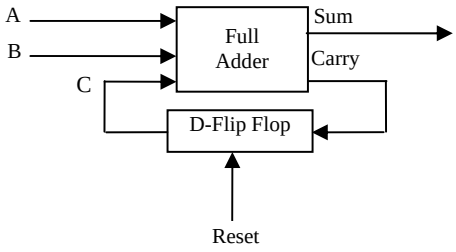


Fig.4. Bit serial adder

The expressions for the sum and carry are given by

$$Sum = A_i \oplus B_i \oplus C_i \quad (5)$$

$$Carry = A_i C_i + A_i B_i + B_i C_i \quad (6)$$

$$C_i = Carry_{i-1} \quad (7)$$

The proposed shift accumulator BSA consisting of pipelined bit serial adders/carry save adders as shown in Fig.5 results in a regular hardware structure with short delays between the clocking elements. Pipelining is the process of inserting pipelining latches along the data-path thereby reducing the critical path. Critical path in any design is the longest path between any two internal latches/flip-flops or between an input pad and an internal latch or between an internal latch and an output pad or an input pad and an output pad. Reduction in critical path results in increased clock speed. Hence the proposed SA using pipelined bit serial adders yields very high speed.

The proposed BSA performs a shift add operation in every clock cycle and a subtraction operation in the sign-bit time. In the first clock cycle, the input word to the SA is added to the initially cleared accumulator. In the next clock cycle, the next input word is added to the right shifted content of SA. This method is repeated until the sign-bit time where the corresponding input word has to be subtracted. The output of sign control unit is zero for all the clock cycles except for the sign bit time. "Sign-bit time" denotes the clock cycle in which the sign bit (MSB bit) of all the inputs arrive simultaneously and the output of sign control unit S=1. The subtraction in the sign bit time is achieved by inverting the input bits of the SA by the XOR gates whose other

input is the sign control bit S=1 and adding a one in the LSB position. One bit output is obtained in every clock cycle.

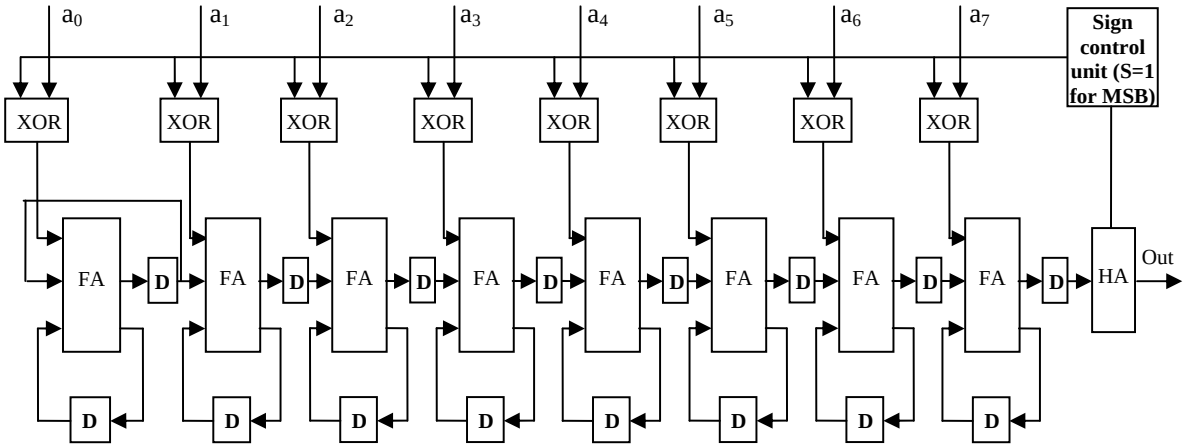


Fig.5. Proposed 8it shift accumulator with pipelined bit serial adders

B. DA based FIR filter with full ROM using the proposed shift accumulator

An 8-tap DA based FIR filter with full ROM using the proposed shift accumulator (BSA) is shown in Fig.6. It consists of a look up table of 2^8=256 locations containing precalculated sum of coefficients. The bank of input shift registers in Fig.6 stores eight consecutive input samples (x[n-i],i=0,1,2,3,4,5,6,7). The concatenation of rightmost bits of the shift registers becomes the address of the LUT. The input shift registers are shifted right at every clock cycle. The corresponding LUT entries are applied as inputs to the BSA which are also right shifted and accumulated in B consecutive times to generate the output y[n]. The input bits $\{x_{i0}\}$ that simultaneously arrive last are the sign bits and the corresponding clock period is called the "sign-bit time". The control signal S = 1 in the sign-bit time, otherwise S = 0. The use of proposed SA using pipelined bit serial adders/carry save adders yields very high speed when compared with the conventional right shift (RSA) and left shift accumulators LSA [11].

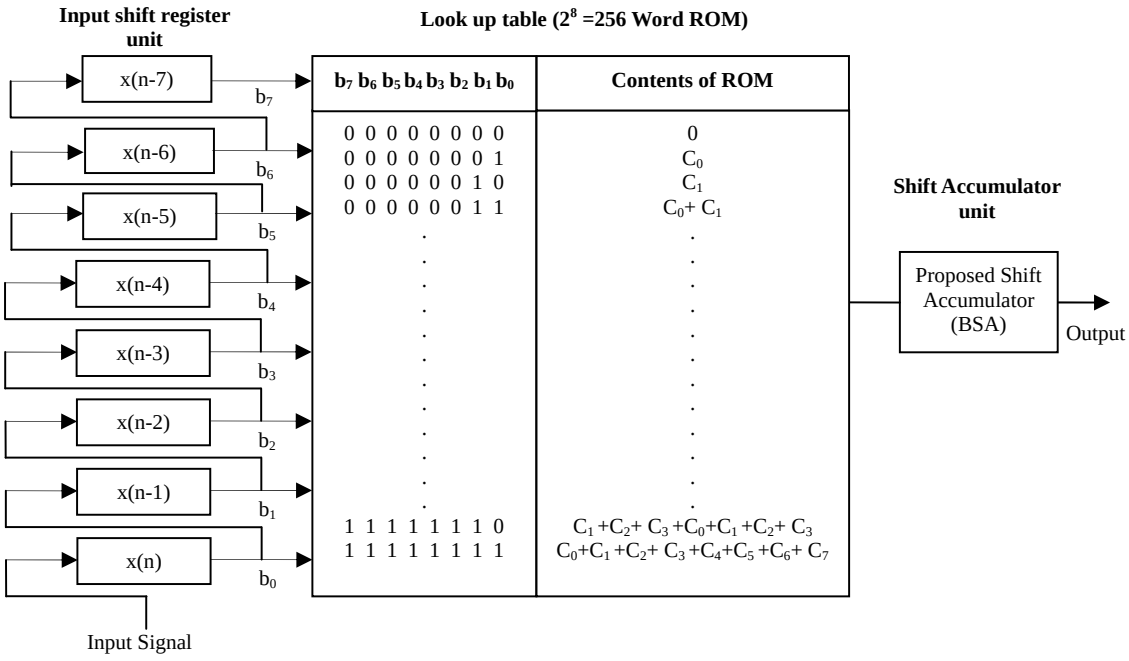


Fig.6. DA based 8-tap FIR filter with full LUT using the proposed shift accumulator

C. DA based FIR filter with LUT partitioning using the proposed shift accumulator

The size of memory (ROM) increases exponentially (2^N) as the order of filter N increases. The bottleneck for speed of the entire system is the memory access time when the ROM size is very large. This disadvantage of DA based FIR filter is overcome by dividing a larger LUT in to smaller LUTs and to combine their outputs with

adders [7], [12], [13]. The N-tap filter is divided into'd'smaller filters each having 'e' input lines such that N= d·e and it is assumed that N is not prime. The total number of clock cycles required for this implementation will be B+$\log_2$d where the additional second term is the number of clock cycles required to implement an adder tree to calculate the sum of the outputs from 'd' LUTs. The total memory requirements of such a decomposed filter are d·2^e memory locations. Hence equation (4) is rewritten as

$$y[n]=-\left(\sum_{z=0}^{d-1}\left[\sum_{i=ze}^{(z+1)e-1} c_i x_{i0}\right]\right)+\sum_{j=1}^{B-1}\left(\sum_{z=0}^{d-1}\left[\sum_{i=ze}^{(z+1)e-1} c_i x_{ij}\right]\right)2^{-j} \quad (8)$$

For example, a 64 tap DA FIR filter would require a large LUT with 2^{64} = 18446744073709551616 words. This problem can be overcome by breaking up the full LUT into 16 smaller LUT units with each having 4 input lines. Hence a single large LUT with 2^{64} memory elements is replaced by 16 LUTS each having only 2^4=16 memory elements which would require only 256 memory elements. The number clock cycles required for the partitioned LUT implementation is 20 whereas that of Full LUT implementation is 16 clock cycles for a input word length B=16.This shows that decrease in throughput is very less when compared with the large memory savings. Fig.7 shows the implementation of an 8-tap FIR filter based on equation (8) for d=2 and e=4.

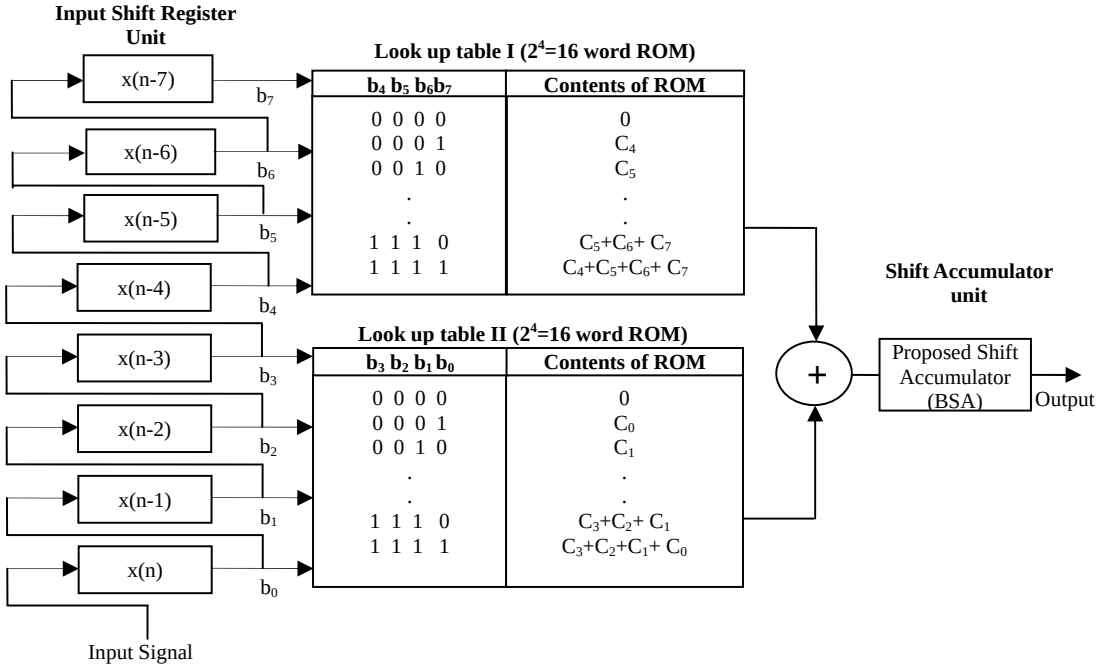


Fig.7. Decomposed 8 tap DA based FIR filter with two LUTs using the proposed shift accumulator

IV. Modified One Dimensional Systolic Architecture For DA Based FIR Filter

Systolic architectures denote a set of interconnected processing elements (PEs) that are capable of performing some simple computation [2], [15]. Information flows rhythmically between cells in a systolic array and communication with the outside world occurs only at the "boundary cells". All the cells in a systolic array are uniform and are fully pipelined. Systolic system is easy to implement because of its regularity, modularity and easy to reconfigure. Systolic architecture can result in cost-effective, high performance special-purpose systems for a wide range of problems.

One dimensional systolic array for decomposed DA based FIR filter based on equation (8) is shown in Fig.8. An N-tap filter is decomposed into 'd' processing elements each having 'e' input lines such that N= d·e. The input sequence x(n) is fed to the input shifter unit. The bank of shift registers in Fig.8 stores consecutive input samples(x[n-i],i=0,1,2,N-1). The concatenation of rightmost bits of the shift registers is given as input to the word parallel convertor that groups input into 'e' bits. The input shift register unit is shifted right at every clock cycle. The 'e' input bits are fed to the $(z+1)^{th}$ PE (for z= 0,1,2,….,d-1) in least significant bits to most significant bits order. To meet the causality requirement the input to each PE is delayed by one cycle period with respect to its preceding PE.

One dimensional systolic array in Fig.8 consists of processing elements (PE1) and output shift adder cell (SA). Function of PE1 is shown in Fig.9.Each PE1 contains a LUT and a adder. In every clock period, each PE1 reads the value stored in its LUT specified by 'e' bits of input vector, adds it to the input available to the cell

from its left and resultant sum is transferred as output to its right. Function of SA is shown in Fig.10. Output SA cell is a Bit serial shift accumulator (BSA) consisting of pipelined bit serial adders/carry save adders which results in high speed. The operation of BSA is shown in Fig.5.The first filter output is obtained after B+d clock cycles after the first input is given to the first PE1 and the successive outputs are obtained in every B cycles.

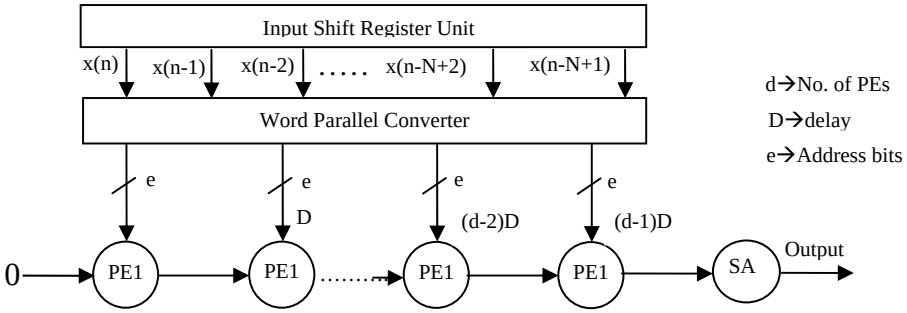
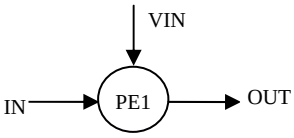
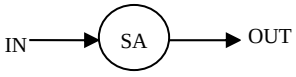


Fig.8. 1-D systolic Array for Decomposed DA based FIR filter



OUT=IN + LUT.Read (VIN)

Fig.9. Function of PE1



OUT=BSA (IN) where BSA is the Bit serial shift accumulation of input

Fig.10. Function of SA

V. FPGA Implementation And Comparison of Performance Metrics

The proposed shift accumulator (BSA) using pipelined bit serial adders/carry save adders, left shift accumulator (LSA), conventional right shift accumulator (RSA) were implemented for various input length using Xilinx Virtex 6vlx240tff1156-1FPGA device and a comparison of the performance metrics is presented in Table I. The results obtained clearly indicate that that the proposed BSA yields lesser delay as shown in Fig.11 and higher speed in terms of maximum frequency as shown in Table I. This is in line with the theory that states that use of pipelining latches increases speed.

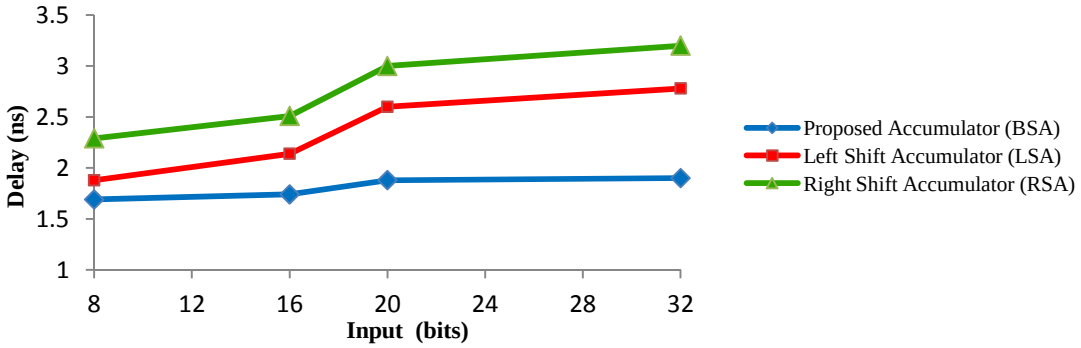


Fig.11. Comparison of delay of proposed shift accumulator with the existing shift accumulators

TABLE I
Comparison of Performance Metrics of Proposed Shift Accumulator with the Existing Shift Accumulators Using Virtex 6vlx240tff1156-1 FPGA Device

Input in Bits	Proposed accumulator using bit serial adders (BSA)		Left shift accumulator(LSA)		Right shift accumulator(RSA)	
	No. of Slices	Maximum Frequency (MHz)	No. of Slices	Maximum Frequency (MHz)	No. of Slices	Maximum Frequency (MHz)
8	17	591.72	5	533.19	11	435.82
16	37	573.62	7	466.96	24	397.85
20	38	531.33	13	384.39	33	333
32	56	525.40	19	359.71	38	312.89

To prove the performance enhancements, the modified DA based 8 tap FIR filter with full LUT using the proposed shift accumulator (BSA), DA based 8 tap FIR filter with RSA and LSA were implemented on Xilinx Virtex 6vlx240tff1156-1FPGA device for an input bit width of B=16 and 8 bit coefficients for filter orders varying from 8 to 64 and a comparison of the performance metrics is presented in Table II. The LUT with 256 locations was synthesized as a single Block RAM of size 256. Results shown in Table II clearly prove that for all values of N ranging from 8 to 64, the modified DA based 8 tap FIR filter with full LUT using the proposed shift accumulator (BSA) is superior to the existing methods of DA based full LUT FIR filter in terms of speed (maximum frequency) which has increased and lesser delay with a very small increase in the number of occupied slices.

TABLE II
Comparison of Performance Metrics of an 8 tap DA filter with full LUT using Virtex 6vlx240tff1156-1FPGA device

Conventional DA	No. of Slices	Minimum delay(ns)	Maximum frequency(MHz)
Using BSA	44	5.166	193.573
Using LSA	26	6.742	148.324
Using RSA	35	7.712	129.668

The greatest disadvantage of DA based FIR filter is that the LUT size (2^N) grows with the order of the filter. To overcome this problem, two factor decomposition of order of filter is presented in section III *C*. An 8-tap filter is decomposed into two LUTS each having 4 input address lines such that 8= 2 x 4. DA based FIR filter with LUT partitioning using the proposed bit serial shift accumulator (BSA), RSA and LSA were implemented on Xilinx Virtex 6vlx240tff1156-1FPGA device and a comparison of the performance metrics is presented in Table III. Partitioned LUTs are accessed using four bits of address. Tabulated results clearly demonstrate that the modified DA-FIR filter with BSA has yielded higher speed when compared with the DA-FIR Filter with LSA and RSA. Comparison of delay of the DA based FIR filter with two factor decomposition using different shift accumulators for various filter orders shown in Fig.12 also proves that proposed BSA has resulted in lesser delay.

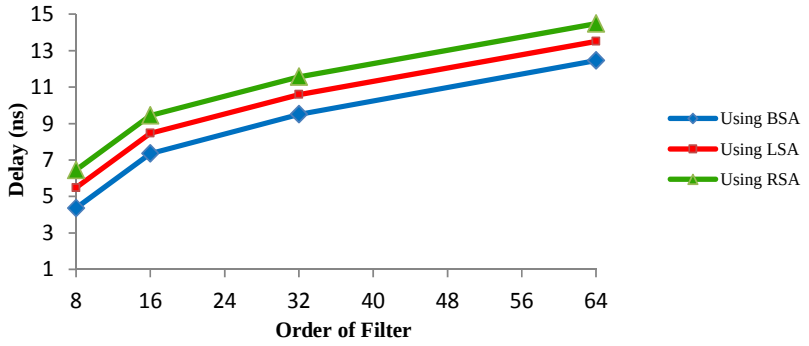


Fig.12. Comparison of delay of DA based FIR filters of various orders with two factor decomposition

TABLE III
Comparison of Performance Metrics of DA Based FIR Filters of Various Orders with Two Factor Decomposition

Tap	DA-FIR using BSA		DA-FIR using LSA		DA-FIR using RSA	
	Occupied slices	Maximum frequency (MHz)	Occupied slices	Maximum frequency (MHz)	Occupied slices	Maximum frequency (MHz)
8	36	229.043	26	182.382	34	154.967
16	51	135.906	35	118.078	47	105.943
32	87	105.175	80	94.411	81	86.490
64	148	80.302	129	74.008	138	69.051

Performance of one dimensional modified systolic architecture of DA based FIR filter explained in section IV for various filter orders is detailed in Table IV. Modified systolic architecture of DA based FIR filter and existing systolic array of DA based FIR filter [14] are both implemented on Xilinx Virtex-6 FPGA device for an input bit width of B=16 and 8 bit of filter coefficients. Tabulated values prove that the proposed DA-FIR using BSA results in a higher speed than the existing method [14].

TABLE IV
Comparison of Performance Metrics of DA Based FIR Filters using Systolic Architecture with Two Factor Decomposition

Tap	Proposed DA-FIR using BSA			DA-FIR using LSA [14]		
	Occupied slices	Minimum delay(ns)	Maximum frequency (MHz)	Occupied slices	Minimum delay(ns)	Maximum frequency (MHz)
8	32	2.936	340.599	21	3.346	298.864
16	44	2.944	339.712	31	3.355	298.102
32	62	2.956	338.341	50	3.358	297.765
64	108	2.962	337.573	94	3.368	296.928

VI. CONCLUSION

One of the most important objectives of DA based FIR filter is to operate at high speed. This is achieved by using the proposed shift accumulator composed of pipelined bit serial adders. Modified DA based FIR filter with full LUT as well as with partitioned LUTs using BSA showed significant improvement in speed than with the existing architectures using left shift accumulator and right shift accumulator. 1-D systolic array of DA based FIR filter with BSA has also resulted in higher speed than the existing architecture.2-D systolic architecture with B number of 1-D systolic arrays can be developed for high speed applications that would provide high throughput at the cost of more hardware. Future work is to develop more area-delay efficient architectures for DA based FIR filters and adaptive FIR filters to meet the growing requirements of DSP applications.

REFERENCES

[1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications., NJ: Prentice-Hall, 1996.
[2] K. K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. New York: Wiley, 1999.
[3] G. R. Goslin, A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance, XILINX, 1995.
[4] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals", U.S. Patent 3 777 130, Dec. 4, 1973.
[5] A. Peled and B. Liu, "A new hardware realization of digital filters", IEEE Transactions on Acoustic, Speech, Signal Processing, vol. 22, no. 6, Dec. 1974, pp.456–462.
[6] S. A. White, "Applications of the distributed arithmetic to digital signal processing: A tutorial review", IEEE ASSP Mag., vol. 6, no. 3, July, 1989, pp. 5–19.
[7] K.G.Shanthi and N.Nagarajan,"Memory based hardware efficient implementation of FIR Filters", International review on computer and software (IRECOS), July 2013,vol.8, no.7, pp.1718-1726.
[8] Wanhammer, L, DSP Integrated Circuits, Academic Press, 1999.
[9] P. Choi, S.-C. Shin and J.-G. Chung, "Efficient ROM size reduction for distributed arithmetic",IEEE International Symposium on Circuits and System (ISCAS), May 2000, vol. 2, pp. 61–64.
[10] H. Yoo and D. V. Anderson, "Hardware-efficient distributed arithmetic architecture for high-order digital filters", Proc. IEEE Int. Conf. on Acoustics, Speech, Signal Processing (ICASSP), March 2005, vol. 5, pp. v/125–v/128.
[11] Patrick Longa and Ali Miri, "Area-Efficient FIR Filter Design on FPGAs using Distributed Arithmetic", IEEE International Symposium on Signal Processing and Information Technology, 2006, pp.249-252.
[12] H.-R. Lee, C.-W. Jen and C.-M. Liu, "On the design automation of the memory-based VLSI architectures for FIR filters," IEEE Trans. Consumer. Electronics, vol. 39, no. 3, pp. 619–629, Aug. 1993.
[13] S.-S. Jeng, H.-C. Lin and S.-M. Chang, "FPGA implementation of FIR filter using M-bit parallel distributed arithmetic", Proc.2006, IEEE Int. Symp. Circuits Systems (ISCAS), May 2006, p. 4.
[14] P. K. Meher, S. Chandrasekaran, and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Transactions on Signal Processing., vol. 56, no. 7, July 2008, pp. 3009–3017.
[15] H. T. Kung, "Why systolic architectures?", IEEE Computer, vol. 15,no. 1, pp. 37–45, Jan. 1982.
[16] Jiafeng Xie n, Jianjun He, Guanzheng Tan," FPGA realization of FIR filters for high-speed and medium-speed by using modified distributed arithmetic architectures", Microelectronics Journal 41, April 2010 pp. 365–370.