Critical Path Reduction of Distributed Arithmetic Based FIR Filter

Sunita Badave
Department of Electrical and Electronics Engineering
M.I.T, Aurangabad
Maharashtra, India

Anjali Bhalchandra
Department of Electronics and Telecommunication Engineering,
G.E.C. Aurangabad,
Maharashtra, India

***Abstract*—Operating speed, which is reciprocal of critical path computation time, is one of the prominent design matrices of finite impulse response (FIR) filters. It is largely affected by both, system architecture as well as technique used to design arithmetic modules. A large computation time of multipliers in conventionally designed multipliers, limits the speed of system architecture. Distributed arithmetic is one of the techniques, used to provide multiplier-free multiplication in the implementation of FIR filter. However suffers from a sever limitation of exponential growth of look up table (LUT) with order of filter. An improved distributed arithmetic technique is addressed here to design for system architecture of FIR filter. In proposed technique, a single large LUT of conventional DA is replaced by number of smaller indexed LUT pages to restrict exponential growth and to reduce system access time. It also eliminates the use of adders. Selection module selects the desired value from desired page, which leads to reduce computational time of critical path. Trade off between access times of LUT pages and selection module helps to achieve minimum critical path so as to maximize the operating speed.**

Implementations are targeted to Xilinx ISE, Virtex IV devices. FIR filter with 8 bit data width of input sample results are presented here. It is observed that, proposed design perform significantly faster as compared to the conventional DA and existing DA based designs.

Keywords—Critical Path*; *Multiplier less FIR filter; Distributed Arithmetic; LUT Design; Indexed LUT

I. INTRODUCTION

Digital Signal Processing (DSP) systems are generally implemented using sequential circuits, where numbers of arithmetic modules in the longest path between any two storage elements are members of critical path. The Critical Path Computation Time (CPCT) determines the minimum feasible clock period and hence maximum allowable operating frequency of DSP system. Finite impulse response (FIR) digital filter is one of the widely used Linear Time Invariant (LTI) systems, has gained popularity in the field of digital signal processing due to its stability, linearity and ease of implementation. However, attention need to pay specifically while designing the high speed FIR filter, as CPCT is affected by both, system architecture as well as techniques used to design arithmetic modules. For such critical design of system architecture, fixed structure offered by Digital signal processor is not suitable. However, high nonrecurring engineering (NRE) costs and long development time for application specific integrated circuits (ASICs) are making field programmable gate arrays (FPGAs) more attractive for application specific DSP solutions. FPGA also offers design flexibility to arithmetic modules then ASICs.

For an N^{th} order FIR filter, each output sample is inner product of impulse response and input vector of latest N samples[1] given in (1).

$$Y(n) = \sum_{k=0}^{N-1} A_k X_{n-k} \tag{1}$$

For critical path minimization, direct implementation of (1) is not a cost effective solution because of two reasons. First, critical path increases with the order of filter and second, multiplier is an expensive arithmetic module with respect to area and computational time. More than two decade, many researchers [2-10] have worked on various multiplierless techniques for FIR filter design. In case of constant coefficient multiplication, look-up-table (LUT) multipliers [11-13] and distributed arithmetic (DA)[14-24] are two memory based approaches found in FIR filter design. An improved distributed Arithmetic technique is addressed here to design for system architecture for FIR filter, as its operating speed is almost independent with order of filter.

In recent years Distributed Arithmetic has gained substantial popularity due to its regular structure and high throughput capability, which results in cost-effective and efficient computing structure. This technique was first introduced by Croisier [14] and further development was carried out by Peled [15] for efficient implementation of digital filters in its serial form. Apart from its several advantages; DA based structure is facing a serious limitation of exponential growth of memory with order of filter. Many researchers [16-27] have addressed this problem, while dealing with this issues. Partial or full parallel structure with two and more than two bits [16,25] has been exploited to overcome the speed limitation, inherent to bit serial DA structure. Attempts were also been made to reduce memory requirement by recasting input data in Offset Binary Coding(OBC)[16], modified OBC and LUTless DA-OBC[19], instead of normal binary coding. Yoo and Anderson [22] extended this work and proposed a hardware efficient LUTless architecture, which gradually replaces LUT requirements with multiplexer/adder pairs. However gain in area reduction is achieved at the cost of increased critical path over the conventional design. LUT decomposition or slicing of LUT, proposed in [23], is one of the ways to restrict the exponential growth of memory. Though

this technique has elucidated a problem of exponential growth of memory, involves the fact that latency and access time are the dependent parameters of level of decomposition.

As the operating speed of a filter is governed by worst case critical path, improved technique is suggested in this paper to increase the speed of operation by reducing critical path. In proposed technique, a single large LUT of conventional DA is replaced by number of smaller indexed LUT pages to restrict exponential growth and to reduce system access time. Indexing the LUT pages eliminates the use of adders of existing techniques [16,17,19,22-24].

Selection module selects the desired value from desired page, and feed the value for further computation. Trade off between access times of LUTs and selection module helps to achieve minimum critical path so as to maximize the operating speed.

In organization of the paper, section II elaborates lookup table concept of conventional DA and proposed DA structures. Critical Path Computation Time (CPCT) analysis of previous and proposed techniques is given in section III. Section IV presents the realization of proposed architecture. Initially component level access time analysis of proposed design is presented in section V, followed by comparison of operating frequency of proposed and previous techniques. Paper is ended with conclusion, in section VI.

II. Conventional Distributed Arithmetic Algorithm For FIR Implementation

Distributed Arithmetic is one of the preferred methods of FIR filter implementation, as it eliminates the need of multiplier, particularly when multiplication is with constant coefficients. By this technique, sum-of-product terms in (1), can easily be transformed into addition. Let B be the word length of input samples, then, in an unsigned binary form, X(n) can be represented as:

$$X(n) = \sum_{i=0}^{B-1} x_{n,i} 2^i \tag{2}$$

where $x_{n,i}$ is the i^{th} bit of X(n). By Substituting the value of X(n) from (2) into (1), inner product can be expressed as:

$$Y(n) = \sum_{k=0}^{N-1} A_k \sum_{i=0}^{B-1} x_{ni} 2^i \tag{3}$$

TABLE I. Conventional Lut Design

LUT address bits				LUT contents
x_3	x_2	x_1	x_0	
0	0	0	0	0
0	0	0	1	A_0
0	0	1	0	A_1
0	0	1	1	$A_1 + A_0$
⋮	⋮	⋮	⋮	⋮
1	1	1	1	$A_3 + A_2 + A_1 + A_0$

Interchanging the sequence of summation in (3), results into:

$$Y(n) = \sum_{i=0}^{B-1} 2^i \sum_{k=0}^{N-1} A_k x_{ni} \tag{4}$$

Further, compressed form of (4), can be expressed as:

$$Y(n) = \sum_{i=0}^{B-1} 2^i \gamma \tag{5}$$

Where, $\gamma = A_0 x_{0,i} + A_1 x_{1,i} + \ldots + A_{N-2} x_{N-2,i} + A_{N-1} x_{N-1,i}$

$X_{ni} \in \{0,1\}$

Thus (5) creates 2^N possible values of γ. All these values can therefore be precomputed and stored in form of look up table shown in table. I. The filtering operation is performed by successively accumulating and shifting these precomputed values, based on the bit address formed by input samples, X(n). A method is proposed to choose desired size of LUT for minimum Critical Path Computation Time of LUT unit. Let N= (n+m); where n and m are arbitrary positive integers. A single large LUT size of 2^N, in conventional design is converted into 2^m LUT pages, each page with 2^n memory locations. Applying this concept to the (5), number of terms in γ can be divided into two groups: n LSB terms and m MSB terms. It is represented by:

$$\gamma = \left(A_0 x_{0,i} + A_1 x_{1,i} + \ldots + A_{n-2} x_{n-2,i} + A_{n-1} x_{n-1,i}\right) + \left(A_n x_{n,i} + \ldots + A_{n+m-1} x_{n+m-1,i}\right) \tag{6}$$

LSB n bits, defines the size of each LUT page, however, MSB m bits defines number of LUT pages. Instead of consisting coefficient sum in conventional look up table, proposed design LUT consists of indexed-sum-of-filter-coefficients.

TABLE II. Proposed Lut Design

n- LUT address bits				LUT contents of each page
x_3	x_2	x_1	x_0	
0	0	0	0	I + 0
0	0	0	1	$I + A_0$
0	0	1	0	$I + A_1$
0	0	1	1	$I + A_1 + A_0$
0	1	0	0	$I + A_2$
0	1	0	1	$I + A_2 + A_0$
0	1	1	0	$I + A_2 + A_1$
0	1	1	1	$I + A_2 + A_1 + A0$
1	0	0	0	$I + A_3$
1	0	0	1	$I + A_3 + A_0$
1	0	1	0	$I + A_3 + A_1$
1	0	1	1	$I + A_3 + A_1 + A_0$
1	1	0	0	$I + A_3 + A_2$
1	1	0	1	$I + A_3 + A_2 + A_0$
1	1	1	0	$I + A_3 + A_2 + A_1$
1	1	1	1	$I + A_3 + A_2 + A_1 + A_0$

TABLE III. Index Term for Each Lut Page

Page number	m - Address Bits		Index terms 'I' for LUT pages
	x_5	x_4	
0	0	0	0
1	0	1	A_4
2	1	0	A_5
3	1	1	$A_5 + A_4$

A page selector module selects desired output from one of the LUT pages, addressed by m bits. A desired combination of n and m facilitates to select the minimum execution time of LUT page and page selector module to attain maximum operating frequency. LUT page structure of 6th order filter, for n=4 and m=2 and indexed term of each page, is elaborated in table II and table III respectively. Each LUT page contains summation of filter coefficients and index term *I*.

III. Critical Path Computation Time Analysis of Proposed Architecture

In this section, CPCT analysis [13] of conventional DA [14-16], LUTless DA [19,22], sliced DA [16,17,23,24]and proposed DA based FIR filter techniques are elaborated. These designs are taken into consideration as they are found more comparable with proposed technique.

Conventional form of distributed arithmetic FIR filter given in fig.1 consists of bank of input registers, LUT unit, and accumulator/shifter unit. Apart from these hardware units, it needs control unit, which defines sequence of filter operation.

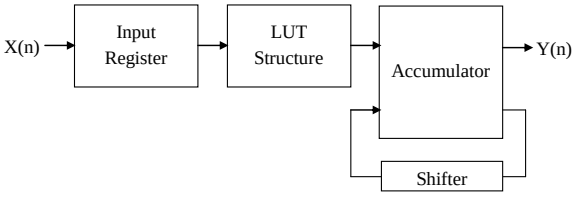

Fig. 1. Functional block diagram of conventional DA based FIR filter

Serially arriving input data values X(n) are stored in parallel form, in input register bank. Right shift of it in every clock cycle; create a word, which is used to address LUT. Successive shift and accumulation of LUT outputs in B cycle gives Y(n).

Data flow graph (DFG) of conventional DA based FIR filter, is as shown in fig.2. It consists of nodes L as LUT, A as accumulator and S as shifter. Access times of L and A are C_L and C_{as} respectively, contributes in critical path. Thus CPCT of conventional DA based FIR filter is expressed as:

$$CPCT_{(cnv)} = C_L + C_{as} \quad (7)$$

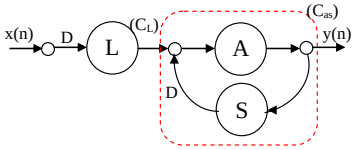

Fig. 2. Data flow graph of conventional DA based FIR filter

A. LUTless DA based FIR filter

Exponential growth of LUT is key issue while designing DA based FIR filter. Elimination of LUT is an attempt found in [13,24] to overcome exponential growth of LUT. In such LUTless structure, shown in fig.3, LUT is replaced by multiplexer-adder pair. On-line data generated by multiplexers are accumulated to create the filter output.

DFG of LUTless DA based FIR filter, shown in fig.4, consists of multiplexer node M, adder nodes T_a and

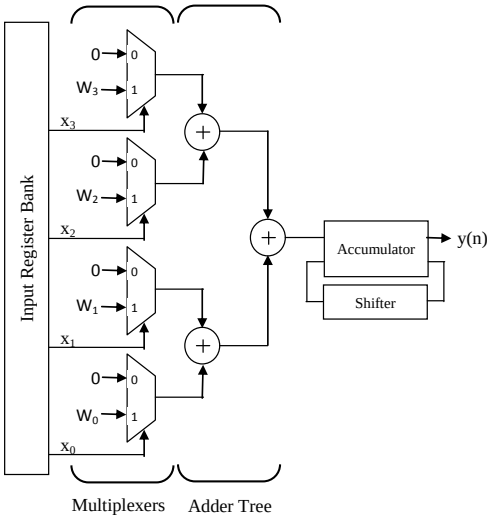

Fig. 3. LUTless DA based FIR filter

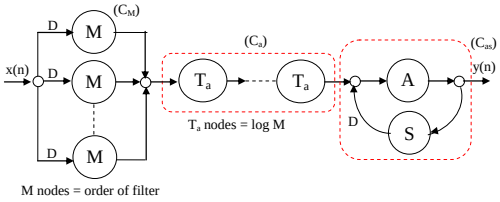

Fig. 4. Data flow graph of LUTless DA based FIR filter

accumulator node A. Though the number of multiplexers is governed by order of filter, access time of only one multiplexer contributes in CPCT, as they are operating concurrently.

Assuming the adders in adder tree are arranged in 4:2 form, access time of $\log_2(N)$ adders are taken into consideration while calculating CPCT of structure C_a. It will be expressed as:

$$C_a = \log_2 N \times T_a \quad (8)$$

Thus C_a is highly filter order dependent as indicated in (9). CPCT of structure becomes:

$$CPCT_{(LUTless)} = C_M + C_a + C_{as} \quad (9)$$

where C_M - access time of multiplexer.

C_a- access time adder tree

C_{as} – access time of accumulator/shifter unit.

B. Sliced LUT DA based FIR filter

Another well-known attempt found in [21,22,27] to restrict the exponential growth of LUT, is the use of multiple memory banks.

Latest, Longa and Miri [23], highlighted that, FIR filter structure will be an area efficient structure by replacing a single large LUT by number of 4-input, smaller LUTs. However, this arrangement leads to put a burden of an adder tree, as it is required to add partial terms generated by each smaller LUT. Generally such LUT arrangement is referred as partitioning or slicing of LUT. Architectural details of sliced DA based FIR filter is shown in fig.5.

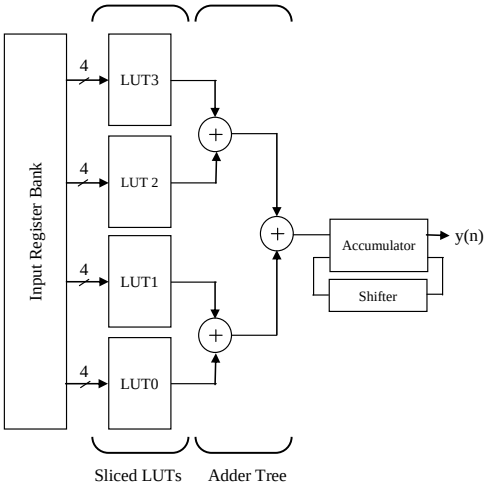

Fig. 5. Sliced LUT DA based FIR filter

Data flow graph of sliced LUT DA based FIR filter, shown in fig.6, consists of concurrently operating 4-input LUT nodes L_s, adder nodes Ta, accumulator A and shifter node S.

In this architecture, requirement of adders in adder tree is governed by number of slices. Assuming the order of filter is divisible by 4, for N^{th} order FIR filter, N/4 will be number of slices and (N/4)-1 will be the number of adders. Thus LUT node L_s, $[\log_2(N/4)]$ adders and accumulator are the members of critical path. So the CPCT of the structure will be:

$$CPCT_{(Slice)} = (C_{SL} + C_a + C_{as}) \quad (10)$$

Where

C_{SL} = access time of one slice of LUT.
C_a = access time of adder tree
= $[\log_2(N/4)]T_a$
T_a = access time of an adder.
C_{as} = access time of accumulator/shifter

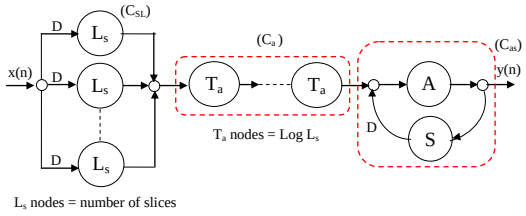

Fig. 6. Data flow graph of sliced LUT DA based FIR filter

Access time of LUT get reduced from C_L to C_{SL} due to slicing technique, however it has added the over heads of adder tree access time C_a in $CPCT_{(slice)}$.

C. Indexed LUT DA based FIR filter

LUTless and SlicedLUT has restricted the exponential growth [22,23], however it has increased the burden of access time of adder tree.

So an attempt is made, to eliminate the use of adder tree by designing an indexed LUT based FIR filter technique. In proposed design of Indexed LUT (ILUT) DA structure, node L of fig.2 is replaced by smaller, desirably indexed LUTs L_i and multiplexer M.

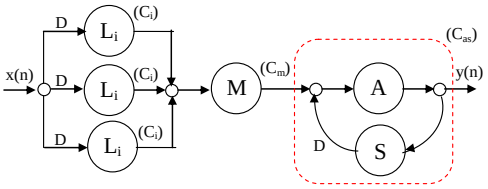

Fig. 7. Data flow graph of indexed LUT DA based FIR filter

DFG of the proposed design derived from (6), is shown in fig.7. CPCT of this structure, contributed by L_i-M-A nodes, will now be:

$$CPCT_{(Index)} = C_i + C_m + C_{as} \quad (11)$$

Where

C_i = access time of an indexed LUT.
C_m = access time of multiplexer
C_{as} = access time of accumulator/shifter

Access time C_i and C_m are interdependent. The trade off of an exponentially varying LUT with linearly varying multiplexer size helps to choose optimum CPCT of a structure. Hence, improves overall operating frequency of filter. It also eliminates the need of adder tree, which further helps to improve the operating frequency.

IV. Realization of Proposed Architecture

Proposed structure of indexed LUT DA based FIR filter is elaborated in following sections. It is built up with four major components bank of input registers, look-up-table unit, accumulator/shifter unit and control unit.

A. Input register bank

Register Bank, shown in fig.8, built up with N serial-in parallel-out shift registers, accepts X(n) input samples, n=0,1,..,N-1. In every clock pulse, register contents take a right shift and generates B terms of length N.

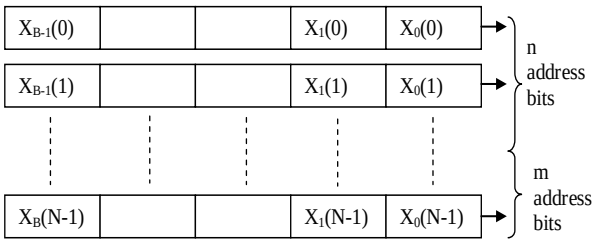

Fig. 8. Input register bank and address bifurcation

LUT address generated by register bank is split into two address groups *n* and *m*. LSB *n* bits define address of LUT, whereas number of LUT pages is defined by *m* bits.

B. Proposed LUT unit

Indexed LUT DA based FIR filter, comprises of indexed LUT pages, each of size 2^n and m bit multiplexer unit as a page selection module. It selects the desired value from desired page. Structural details of an example, considered in section 2*A*, of 6^{th} order FIR filter, with n=4 and m=2, is shown in fig.9. Four LUT pages, each with 16 locations are connected in parallel, by set of 4 address lines. A multiplexer unit of size 4:1 selects an appropriate output for further stage.

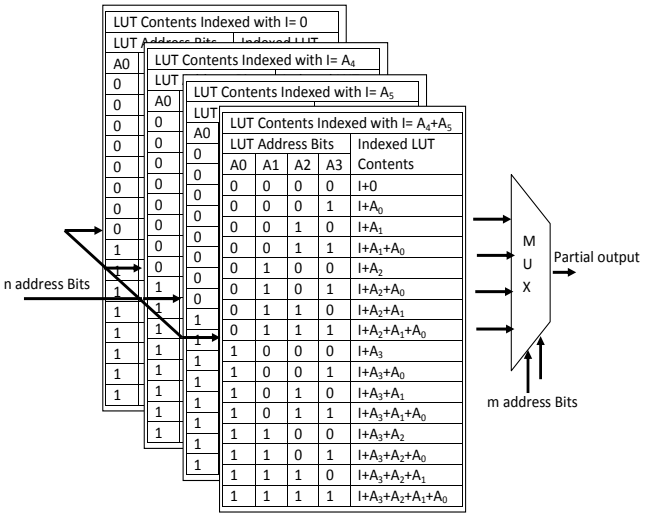

Fig. 9. Proposed structure of LUT unit

C. Accumulator and Shifter Unit

Accumulator and shifter are two separate combinational units, however jointly these are responsible for calculating the dot product term of filter output. Its hardware complexity is greatly influenced by the way of LUT addressed and accordingly a shift is given to accumulator/shifter unit to generate partial products.

D. Control Unit

It is a finite state machine, shown in fig.10, defines sequence of operation and has overall control on filtering operation.

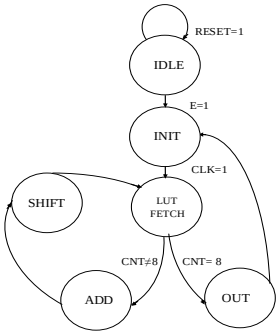

Fig. 10. Control unit of proposed structure

Filtering operation remains in idle state with application of reset. It starts with enable signal E and takes iteration equal to input precision for every clock cycle. At the end of count it gives filter output and operation begins with next fetch cycle.

V. Performance Analysis

Performance is evaluated based on operating frequency. Design is implemented on FPGA Vertex IV, for particular filter order N and for all possible combinations of n and m, as shown in table IV. Each node of proposed structure is critically analyzed for CPCT of proposed structure, for the range of filter from 4 to 8. Table IV gives the details of filter operating frequency with variation in access times of LUT page C_i and multiplexer unit C_m.

Graphical representation for 8th order FIR filer is shown in fig. 11. It indicates that, access time of LUT page C_i increases exponentially with n, at the same time access time of multiplexer C_m decreases linearly.

If f_{max} is assumed to be the maximum operating frequency, T_{sample} is the minimum time required to process each output sample, then

$$T_{sample} \geq CPCT$$
$$\geq C_i + C_m + C_{as} \quad (12)$$

As

$$f_{max} = 1/ T_{sample}$$
$$f_{max} \leq 1/ C_i + C_m + C_{as} \quad (13)$$

As CPCT minima of filter is obtained at the point of intersection of LUT access time C_i and MUX access time C_m, which leads to maximum operating frequency. Thus filter design corresponds to these values of m and n will be treated as optimized design.

TABLE IV. Access Time Analysis of LUT Unit Modules

Order of Filter	Address Line distribution		LUT Unit Access time analysis		Operating freq. in MHz
	n	m	C_i	C_m	
8	7	1	6.58	3.6	151.389
	6	2	5.45	4.06	160.937
	5	3	5.02	4.46	155.876
	4	4	4.65	4.8	184.834
	3	5	4.65	5.16	169.544
	2	6	4.6	5.5	168.714
	1	7	3.84	6.1	176.625
7	6	1	5.45	3.6	189.92
	5	2	5.02	4.06	180.874
	4	3	4.65	4.46	183.441
	3	4	4.65	4.8	191.18
	2	5	4.6	5.16	182.45
	1	6	3.84	5.5	190.13
6	5	1	5.02	3.6	190.3
	4	2	4.65	4.06	192.417
	3	3	4.65	4.46	205.495
	2	4	4.6	4.8	190.389
	1	5	3.84	5.16	192
5	4	1	4.65	3.6	206.793
	3	2	4.65	4.06	228.645
	2	3	4.6	4.46	239.664
	1	4	3.84	4.8	215.736
4	3	1	4.65	3.6	242.93
	2	2	4.6	4.06	242.93
	1	3	3.84	4.46	244.09

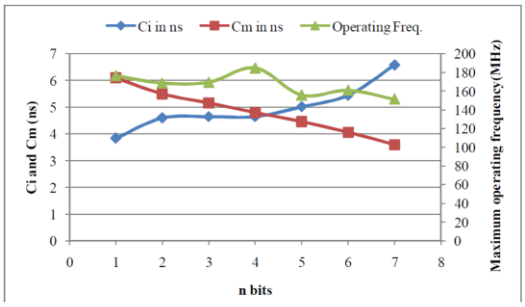

Fig. 11. Relation between access time analysis of LUT unit modules and operating frequency of 8th order FIR filter

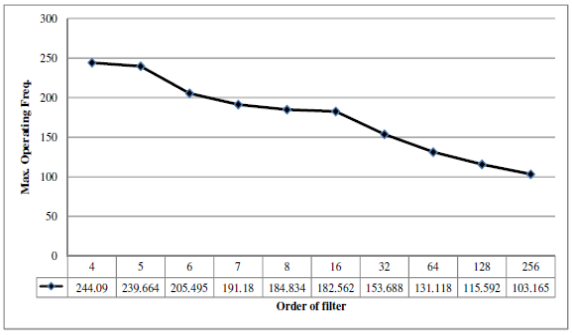

Fig. 12. Relation of maximum operating frequency with order of filter

This technique can further be extended to any desired order of filter. Filter performance upto 256 order is shown in fig. 12. Results obtained by the proposed technique are compared with Conventional DA, LUTless DA[22] and Sliced LUT DA[23]

TABLE V. OPERATING FREQUENCY COMPARISON OF VARIOUS ARCHITECTURES

Order of filter	Operating frequency of DA based filter in MHz			
	Conventional DA	LUTless DA	Sliced DA	Proposed DA design
4	242.4	242.93	240.13	244.09
5	239.01	239.06	220.037	239.664
6	200.95	174.074	200.122	205.495
7	184.65	175.503	185.685	191.18
8	176.22	174.28	167.726	184.834

techniques, which were implemented on Altera Stratix FPGA chip. To surmount the platform differences, these techniques are faithfully implemented on same platform as that of the proposed technique. Desired filter coefficients are obtained from FDATool, a special toolbox of MATLAB, which are truncated and scaled to 8-bit precision. Xilinx Integrated Software Environment (ISE) is used for performing synthesis and implementation of the designs.

To validate the correct functionality using random input, each implementation is simulated with the simulation tool provided by Xilinx.

A comparative study of maximum operating speed of conventional DA,LUTless DA, Sliced DA and proposed DA based filter techniques is presented in table V and its graphical representation is in fig.13.

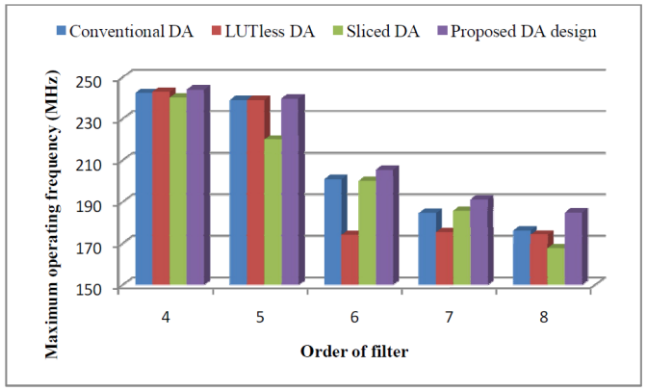

Fig. 13. Comparison of operating frequency

TABLE VI. STRUCTURAL COMPLEXITY OF PREVIOUS AND PROPOSED DESIGNS

Order of filter	Structural Complexities			
	Conventional DA	LUTless DA	Sliced DA	Proposed DA design
Input Register	NxB	NxB	NxB	NxB
Memory Bits	$M_C = 2^N x B$	-	$M_S = (a x 2^l) x B$	$M_I = (2^m x 2^n) xB$
Decoder	$N: 2^N$	-	$a(l:2^l)$	$2^m (n:2^n)$
Number of Adders	-	N-1	a-1	-
Depth of Adders	-	$B+\log_2 N$	$B+\log_2 a$	-
Multiplexers	-	-	-	$2^m:1$
CPCT	$C_L + C_{as}$	$C_M + C_a + C_{as}$	$C_{SL} + C_a + C_{as}$	$C_i + C_m + C_{as}$
Latency	B+1	B+1	B+1	B+1
Throughput	B+2	B+2	B+2	B+2

Operating frequency reduces with the order of filter is one of the obvious observations indicated in table V. It is also observed that operating frequency of proposed technique is higher than conventional DA and existing DA[22,23] techniques. No much gain in rise of frequency is obtained at 4^{th} order as techniques are get correlated with technology platform, however frequency growth is increasing along with the order of filter.

Structural complexities of N^{th} order filter are analyzed and performances are compared for random input samples x(n). Word length of input sample and filter coefficient is assumed to be of B bits, which makes size of input register bank to be same for all designs under consideration. Latency and throughput found same in all DA based structures; however operating speed of individual technique makes the value to differ.

For implementation of N^{th} order conventional DA based FIR filter requires memory array of 2^N x B bits and the size of decoder is $N:2^N$. CPCT of the structure is $(C_L + C_{as})$, increases exponentially due to exponential rise in C_L, however C_{as} is independent with order of filter. Thus it is almost constant in all structures. Structural complexities of conventional DA based FIR filters are considered as bench marks for performance comparison.

Slicing of single large memory reduces the memory requirement of design from 2^N X B of conventional DA to (a X 2^l) X B; where a and l are the factors of N. Thus decoder also get changed from single $N: 2^N$ to a, $l:2^l$. As multiple terms are generated by this technique, need at least a-1 adders to generate coefficient sum as partial term. A single large LUT is replaced by smaller LUTs, reduces LUT access time from C_L to C_{SL}, however it adds adder access time C_a, tending to increase CPCT of structure.

LUTless technique selects filter coefficient on-line by multiplexer, eliminates the need of memory and corresponding decoder at the cost of N-1 adders. As LUT is replaced by multiplexers and adders, C_M and C_a are the contributors of CPCT, which are highly filter order dependent.

In proposed technique, indexing of LUT pages reduces its access time C_i instead of C_L as well as eliminates C_a as a prime contributor of CPCT of LUTless and sliced LUT DA based techniques. It adds a small burden of LUT page selection

module Cm, to CPCT of structure. However it leads to reduce overall CPCT, leading to increase in operating frequency. This rise in frequency is significant with higher filter order as indicated in table V.

VI. CONCLUSION

For high speed FIR filter implementation in distributed arithmetic, the exponential rise of memory access time with the filter coefficients has always been considered to be a fundamental drawback. LUTless DA and sliced LUT DA based technique restricts exponential growth, however needs adders to generate partial term. Number of adders and depth of adders, is governed by order of filter in LUTless technique. However in sliced LUT based technique, number of slices defines number of adders. Even for particular filter order, number of adders increases with increase in number of slices, tending to increase CPCT of structure. An innovative technique to reduce CPCT of FIR filter is designed and implemented successfully, which leads to increase in operating frequency. Indexing of LUT restricts exponential growth and also completely eliminates need of adders which results in significant reduction in CPCT and maximizes operating frequency.

REFERENCES

[1] Mitra S. K., Digital filter structures: Digital Signal Processing-A Computer Based Approach, 3rd ed., India.Tata McGraw Hill, 2008, pp.427-437

[2] Henry Samueli, "An Improved Search Algorithm for the Design of Multiplierless FIR Filters with Powers-of-Two Coefficients", IEEE Transactions on Circuits and Systems, Vol 3.6 , No.7, pp.1044-1047, July1989.

[3] Joseph B. Evans, "An Efficient FIR Filter Architectures Suitable for FPGA Implementation", Proceedings of the IEEE International Symposium on Circuits and Systems(ISCAS), pp.226-228, 1993.

[4] Woo Jin Oh, Yong Hoon Lee, "Implementation of Programmable Multiplierless FIR Filters with Powers-Of-Two Coefficients", IEEE Transactions on Circuits and Systems-II Analog and Digital Signal Processing, Vol. 42, No. 8, pp.553-556, August 1995.

[5] Kei-Yong Khoo, Alan Kwentus, and Alan N. Willson, Jr.,"A Programmable FIR Digital Filter Using CSD Coefficients", IEEE Journal of Solid-State Circuits Vol II No 6, pp.869-874, June 1996.

[6] Dawoud D. S., "Realization of pipelined multiplier - free FIR digital filter", Proc. IEEE Africon Conference,pp.335–338, 1999.

[7] Marko Kosunen, Kari Halonen, "A Programmable Fir Filter Using Serial-In-Time Multiplication And Canonic Signed Digit Coefficients" Proceedings of the 7th IEEE International Conference on Electronics, Circuits and Systems(ICECS), pp.563-566, 2000.

[8] Kah-Howe Tan, Wen Fung Leong, Kadambari Kaluri, M A. Soderstrand and Louis G. Johnson, "FIR Filter Design Program that Matches Specifications Rather than Filter Coefficients Results in Large Savings in FPGA Resources" Proceedings of the IEEE International Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, Vol.2, pp.1349-1352, 2001.

[9] Zhangwen Tang, Jie Zhang and Hao Min, "A High-Speed, Programmable, CSD Coefficient FIR Filter", IEEE Transactions on Consumer Electronics, Vol. 48, No. 4, pp. 834-837, November 2002.

[10] K. S. Yeung and S. C. Chan, "Multiplier-Less Digital filters Using Programmable Sum-of-Power-of-Two(SO POT) Coefficients", Proceedings of the IEEE International Conference on Field-Programmable Technology(FPT), pp.78-84, 2002.

[11] Pramod Kumar Meher, "New Look-up-Table Optimizations for Memory-Based Multiplication", Proceedings of the 12th IEEE International Symposium on Integrated Circuits, ISIC 2009, pp.663-666, 2009.

[12] Pramod Kumar Meher, " New Approach to Look-Up-Table Design and Memory-Based Realization of FIR Digital Filter", IEEE Transactions on Circuits and Systems—I: Regular Papers, Vol. 57, No. 3, pp. 512-603, March 2010.

[13] Keshab K. Parhi, "VLSI Digital Signal Processing- Design and Implementation," in John Wiley & Sons, India,1999, pp.36-37,43-45.

[14] A. Croisier D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital filter for PCM encoded signals," *U.S. Patent3 777 130*, Dec. 4,1973

[15] A. Peled and B. Liu, "A new hardware realization of digital filters," in IEEE Trans. Acoust. Speech, Signal Process., vol. 22, no. 6, pp. 456–462, Dec.1974.

[16] S.A.White, "Applications of the Distributed Arithmetic to Digital Signal Processing: A Tutorial Review," IEEE ASSP Mag., vol. 6, no. 3, pp. 5–19, Jul. 1989.

[17] Wayne P. Burleson, Louis L. Scharf, " VLSI Design of Inner-Product Computers Using Distributed Arithmetic", Proceedings of the IEEE International Symposium on Circuits and Systems(ISCAS), pp.158-161, 1989.

[18] Rudi BabiE, Mitja Solar, Bruno Stiglic, "High Order FIR Digital Filter Realization in Distributed Arithmetic" Proceedings of 6th Mediterranean Electrotechnical Conference, pp.367-370,1991.

[19] Jung-Pal Choi Seung- Cheol Shin Jin- Gyun Chung, "Efficient Rom Size Reduction For .Distributed Arithmetic", IEEE International Symposium on Circuits and Systems(ISCAS), pp. II-61 to II-64, 2000.

[20] T. S.Chang and C.-W.Jen, "Hardware-efficient pipelined programmable FIR filter design", Proceeding of IEEE on Computers and Digital Techniques, vol.148, issue: 6, pp.227-232,2001.

[21] Chin-Chao Chen, Tay-Jyi Lin, Chih-Wei Liu, and Chein-Wei Jen, "Complexity-Aware Design of DA-Based FIR Filters", proceeding of IEEE Asia-Pacific Conference on Circuits and Systems, pp.445-448,2004.

[22] Heejong Yoo And David V. Anderson, "Hardware-Efficient Distributed Arithmetic Architecture For High-Order Digital Filters" Proceeding of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP),Vol.5, pp.V-125-V-128, 2005.

[23] Patrick Longa And Ali Miri, "Area-Efficient FIR Filter Design On FPGA Using Distributed Arithmetic", IEEE International Symposium on Signal Processing And Information Technology, pp. 248-252, 2006.

[24] M. Mehendale, S. D. Sherlekar, and G.Venkatesh, "Area-delay Tradeoff in distributed arithmetic based implementation of FIR filters," Proceedings of 10th International Conference of VLSI Design, pp. 124-129, 1997.

[25] Shiann-Shiun Jeng, Hsing-Chen Lin, And &Shu-Ming Chang, "FPGA Implementation of FIR Filter Using M-Bit Parallel Distributed Arithmetic" IEEE, International Symposium on Circuits and Systems(ISCAS),pp.875-878,2006.

[26] Pramod Kumar Meher, Shrutisagar Chandrasekaran, Abbes Amira, "FPGA Realization of FIR Filters by Efficient and Flexible Systolization Using Distributed Arithmetic", IEEE Transactions on Signal Processing, Vol. 56, No. 7, pp.3009-3017, 2008.

[27] B.K.Mohanty,P.K Meher, "A High-Performance FIR Filter Architecture for Fixed and Reconfigurable Applications", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, issue 99,1-9, 2015.

AUTHOR PROFILES

Sunita Mukund Badave received the B.E. degree in Electrical (Electronics Specialization) from Shivaji University in 1989 and M.E.Degree in Electrical from Dr.B.A.M.University., Aurangabad, India, in 1998. She is currently working toward the Ph.D. degree in Electronics at Dr.B.A.M.University. Her research interests include architectures and circuit design for digital signal processing. She has presented nearly 16 technical papers in at Nationally and Internationally. Mrs. S.M. Badave is *Member* of the Institute of Electronics and Telecommunication Engineers(IETE),India and life member of Indian Society for Technical Education(ISTE)India.She is also member of IAENG, International Association of Engineers.

Anjali S. Bhalchandra received the B.E. Electronics and Telecommunication degree and M.E. Electronics degree in 1985 and 1992 respectively. She has completed her Ph.D.

in Electronics from S.R.M.University, Nanded, India, in 2004. She has a scientific and technical background covering the areas of Electronics and Communication. Currently, she is Head of Electronics and Telecommunication Engineering Department and Associate Professor in Government College of Engineering, Aurangabad. Her research interest includes image processing, signal processing and communication. She has published more than 50 technical papers in various reputed journals and conference proceedings. Dr. Bhalchandra is a Fellow of the Institution of Engineers (IE), India and life member of Indian Society for Technical Education(ISTE)India.