RESEARCH ARTICLE OPEN ACCESS

Enhanced LUT For Modified Distributed Arithematic Architecture - FIR Filter

N Vivek*, Prof K Anusudha**
*(Department of Electronics Engineering, Pondicherry University, Puducherry)
** (Department of Electronics Engineering, Pondicherry University, Puducherry)

ABSTRACT
This paper presents Enhanced LUT for Modified Distributed Arithmetic Architecture for efficient implementation of finite impulse response (FIR) filter. In this technique consists of shift registers, Look Up Table (LUT) and accumulator. Based on this technique multipliers in Fir filter are replaced with LUT. Multiplications are performed using shift operation. Performance analysis of various filter orders with filter coefficients (with Kaiser Window technique) are synthesized using Verilog HDL.
Keywords-Finite Impulse Response (FIR), Modified Distributed Arithmetic (MDA), Look Up Table (LUT),Configurable Logic Blocks (CLBs)

I. INTRODUCTION

Digital Finite Impulse Response (FIR) filters are frequently used in most Digital Signal Processing (DSP) systems by virtue of stability and easy implementation, especially in the realm of communication and multimedia applications. In this sense, great effort has to be put in designing efficient architectures for digital signal processing functions such as FIR filters, which are widely used in signal processing, telecommunications and etc. The filters are major determinants of the performance and power consumption of the whole system. Since field programmable gate arrays (FPGAs) contain a very high number of Configurable Logic Blocks (CLBs), they become more feasible for implementing specific DSP functions such as FIR filters. The problem of designing FIR filters are suffering from a large number of multiplications, which leads to excessive area and power consumption. Many works have been focused on designing alternative algorithm such as Distributed Arithmetic (DA) Other works have development on optimizing multiplications by decomposing them into simple operations such as addition, subtraction and shift or sharing common sub-expressions.

In literature, several multipliers-less schemes had been proposed. These methods can be classified in two categories according to how they manipulate the filter coefficients for the multiply operation. The first type of multiplier-less technique is the conversion-based approach, in which the coefficients are transformed to other numeric representations whose hardware implementation or manipulation is more efficient than the traditional binary representation. Example of such techniques are the Canonic Sign Digit (CSD) method, in which coefficients are represented by a combination of powers of two in such a way that multiplication can be simply implemented with adder/subtractions and shifters, and the Dempster-Mcleod method, which similarly involves the representation of filter coefficients with powers of two but in this case partial results arranged in cascade to introduce further savings in the usage of adders. The second type of multiplier-less method involves use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations. These memory-based methods involve Constant Coefficient Multiplier method and the very-well known Distributed Arithmetic method

The paper is organized as follows: Section II and Section III give the introduction of Basic FIR filter and Distributed Arithmetic. Section IV explains the realization using Distributed Arithmetic and Section V deals with Modified Distributed FIR Filter and its Realization. Results are analyzed in Section VI. Section VII concludes the proposed work.

II. BASIC FIR FILTER (N-TAP)

In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response is of finite duration, because it settles to zero infinite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and many continue to respond indefinitely.

The basic N-tap Filter shown in Figure1,

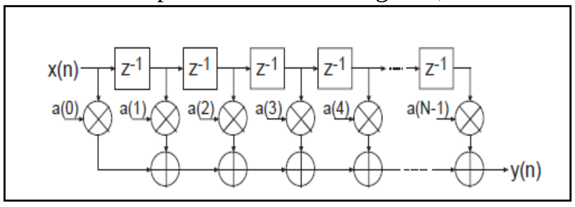

Figure 1 Basic FIR filter (N-tap)

$$y(n) = a_0x_0 + a_1x_1 + a_2x_2 \ldots\ldots. +a_{n-1}x_{n-1} \ldots(1)$$

Where,

x[n] is the input signal,

y[n] is the output signal,

$a_i$is the filter coefficients.

For FIR Filter of N order filter has

- Coefficients N+1
- Multipliers N+1
- Adders N

III. DISTRIBUTED ARITHMETIC

Distributed Arithmetic is one of the most well known methods of implementing FIR filters. An FIR filter of length K (bit size) is described as

$$y[n] = \sum_{k=0}^{N} h[k]x[n-k] \qquad \ldots (2)$$

Where h[k] is the filter coefficients and x[k]is the input data. For simplicity it can be writing as $x^0[k]$ =x [n-k] in equ (2) as

$$y[n] = \sum_{k=0}^{N} h[k]X^\circ[k] \qquad \ldots (3)$$

Where $X^0[k]$ is the B-bit two complement binary number to present data.

$$X^0(k) = -2^B . X_B[k] + \sum_{b=0}^{B-1} (x_b[k]. X^b) \qquad \ldots (4)$$

On substituting equation (4) in (3) we get (on solvingfurther)

$$y = -2^B . f(h[k], x_B[k]) + \sum_{b=0}^{B-1} 2^B . f(h[k], X_B[k]) \qquad \ldots (5)$$

We have$f(h[k], x_B[k]) = \sum_{k=0}^{K-1} h[k]. X_B[k]$. In equ.5 we observe the filter coefficients are pre-stored in LUT and addressed by $X_b = X_b[0]X_b[0], X_b[1], X_b[2], \ldots\ldots X_b[K-1]$. This way the MAC blocks of FIR filters are reduced to access (multiplier-less circuit) and summation with LUT.

IV. FIR FILTER REALIZATION USING DISTRIBUTED ARITHMETIC

The basic block Diagram of Distributed Arithmetic for 4 tap Fir Filter is shown below in Figure [2]

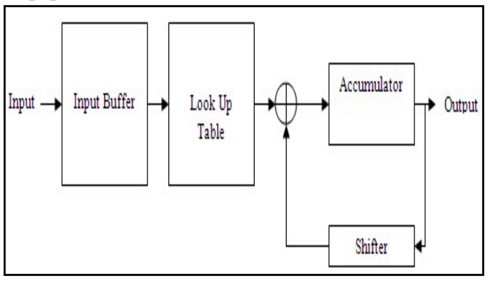

Figure2: Block diagram of DA Fir Filter

A Input buffers*:*

In the Block Diagram shows the Input buffers, as it stores the input values as the order that LSB values are taken in for first cycle and so on. During First iteration least significant bit taken and stored in register and it sent it to the further LUT stage and for next iteration the cycle repeats for N bits.

For example: X1=1101, X2=0010
X3=0010, X4=1111
X1[0] X2[0] X3[0] X4[0] =1001
X1[1] X2[1] X3[1] X4[1] =0111
X1[2] X2[2] X3[2] X4[2] =1001
X1[3] X2[3] X3[3] X4[4] =1001

Where X1, X2, X3, and X4 are the inputs of Fir Filter

B Look UP Table (LUT)

In this paper, for designing the LUT ROM is used to store the filter coefficients as shown in the table [1].

Table 1: Formula for getting Filter Coefficients used in LUT

b_h[N-1]	b_h[N-2]	…	b_h[0]	Coefficient Value
0	0	0	0	0
0	0	0	1	$h[0]b_0$
.	.	.	.	.
b_{N-1}	b_{N-2}		b_0	$h[0]b_0+\ldots.+ h[N-1]b_{N-1}$
1	1	1	1	h[0]+ …+h[N-1]

As the number of addresses increases, functions also increases. Hence complexity will also increase. Here addresses contain four bits and sixteen functions. If addresses have 8 bit then LUT will have 255 different combinations. To design a LUT of 255 different functions is complex and it will occupy more area.

C Scaling unit:

Scaling accumulator reduces the size, eliminates the multiplier and performs the operations serially. It requires adder, register and right shifter each of 8-bit size .Firstly input 'a' which is coming from LUT is added with input 'b' which is initially zero. It is stored in register and before shifting, the LSB bit is stored in register because in order to save the shifted bit. Then right shifted by one bit or divided by '2'. The next input 'a' coming from LUT at next clock cycle is added with previously right shifted output. It is again stored in register and right shifted by one bit and so on.

V. MODIFIED DISTRIBUTED FIR FILTER

A Partitioning Technique:

For Designing of 4 tap Fir DA filter we need LUT with 16 addresses in order to reduce the address, this paper reduces the address using partitioning technique. The Basic block diagram of modified Distributed Fir Filter is shown in Figure [4].

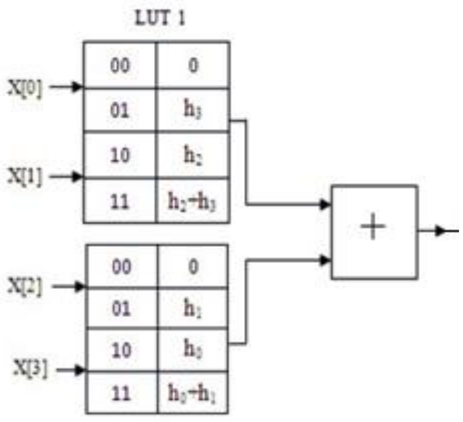

Figure 3: Modified Distributed FIR Filter (4 tap filter)

B Modified Look UP Table (LUT)

In this paper for the redesigning of LUT we used ROM to store the filter coefficients using the formula as shown in the Table[1].

$b_h[0]$	$b_h[1]$	$b_h[N-2]$	$b_h[N-1]$	$b_{hN[\,]}$	Coefficient Value
0	0	0	0	0	0
1	0	0	0	0	$h[0]b_0$
.	.	.	.	0	.
b_0	b_1	b_{N-2}	b_{N-1}	b_N	$h[0]b_0+….+h[N-1]b_{N-1}$
1	1	1	1	1	$h[0]+ …+h[N-1]$

Table 1: Formula for getting Filter Coefficients used in LUT

As the number of addresses increases, functions also increases. Hence complexity will also increase. Here addresses contain four bits and sixteen functions. If addresses have 8 bit then LUT will have 255 different combinations. To design a LUT of 255 different functions is complex and it will occupy more area.

C Scaling unit:

Scaling accumulator reduces the size, eliminates the multiplier and performs the operations serially. It requires adder, register and right shifter each of 8-bit size .Firstly input 'a' which is coming from LUT is added with input 'b' which is initially zero. It is stored in register and before shifting, the LSB bit is stored in register because in order to save the shifted bit. Then right shifted by one bit or divided by '2'. The next input 'a' coming from LUT at next clock cycle is added with previously right shifted output. It is again stored in register and right shifted by one bit and so on.

VI. V REALIZATION OF 4 TAP, 8TAP AND 16 TAP MDA FIR FILTER

This Paper mainly deals with the memory utilization and no of LUT's of FIR Filter of Enhanced LUT for Modified Distributed Arithmetic. The Table [2] shows with Various Filters with Orders of 4, 8 and 16 Tap Filter.

Table 2: Comparison of 4, 8 and 16 TAP MDA filter with LUT's and Memory Locations

Filter	No Partition		2-Partition		4-Partition		8-Partition	
	No. of LUTs	Memory Locations	No. of LUTs	Memory Locations	No. of LUTs	Memory Locations	No. of LUTs	Memory Locations
4-Tap	1	16	2	8	-	-	-	-
8-Tap	1	256	2	6	4	32	-	-
16-Tap	1	65536	2	256	4	128	8	32

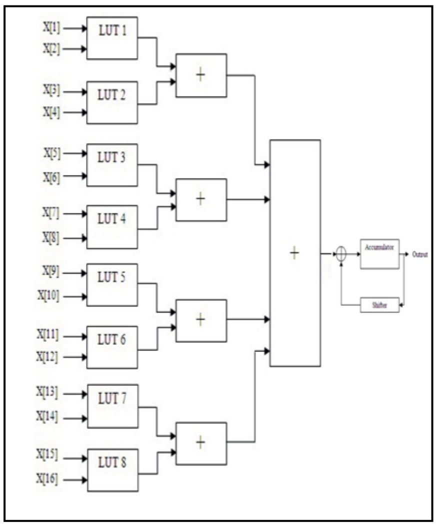

Figure5:16 Tap Modified DA FIRFilter

VII. RESULTS AND ANALYSIS

The performance analysis of Modified Distributed Fir filter of 4, 8 and 16 Tap filter is done in Family of Spartan 3E and the Device XC3S100E, package of CP132.

Table 3: Delay performances of 4, 8 and 16 TAP MDA filter

	without partition (ns)	2 Partition (ns)	4 Partition (ns)	8 Partition (ns)
4 Tap	12.450	14.490	--	--
8 Tap	25.811	21.828	22.807	--
16 Tap	--	39.551	38.173	60.760

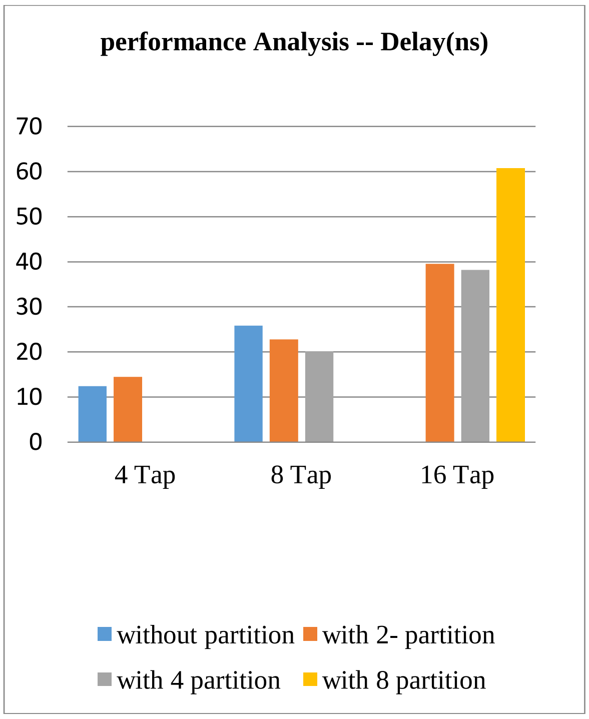

Figure 6: Modified Distributed Arithmetic FIR Filter

Table 5: Delay performances of 4, 8 and 16 tap MDA Filter of Proposed Model

	without partition (ns)	2 Partition (ns)	4 Partition (ns)	8 Partition (ns)
4 Tap	11.219	11.739	--	--
8 Tap	22.522	20.351	20.122	--
16 Tap	--	33.241	34.034	37.515

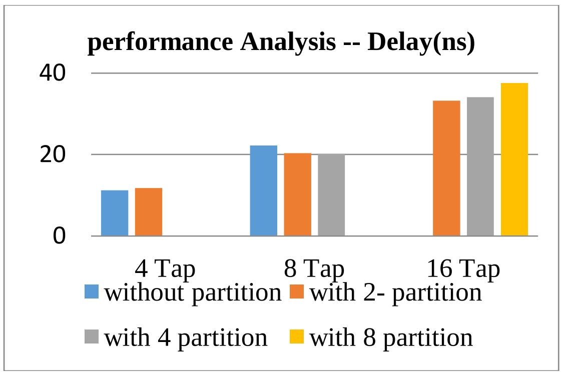

Figure 7: Proposedof Enhanced Modified Distributed Arithmetic FIR Filter

VIII. CONCLUSION

In this paper, the multiplier-less FIR filter is implemented using Enhanced LUT with Modified Distributed Arithmetic which consists of Look Up Table is involved using Verilog HDL.Conversion of Floating point Filter coefficients to Fixed point Filter coefficients in LUTs used by FDA Tool using Mat Lab, an Enhanced LUT with efficient Modified Distributed Arithmetic FIR Filter is proposed. Therefore by replacing the enhanced LUT and input buffers with the proposed model, reduces the area by memory address Locations. The reduced Number of logic levels of this work offers the great advantage in the reduction of delay.

REFERENCES

[1] M.Keerthi,,S.NagakishoreBhavanam,Jeevan Reddy K, “Distributed Arithmetic for FIR Filter Implementation on FPGA”, International Journal Of Engineering Research & Technology (IJERT)”, Vol.1, Issue.9, pp 1-8, November 2012

[2] Zhou, Yajun, and Pingzheng Shi, “Distributed Arithmetic for FIR Filter implementation on FPGA”, Proc. IEEE on International Conference in Multimedia Technology (ICMT), 2011 on, pp. 294-297, 2011.

[3] Yajun Zhou, pingzheng Shi, “Distributed Arithmetic for FIR Filter Implementation On FPGA” , IEEE International Conference on Communications, Circuits and Systems(ICCCAS), pp. 620-623, July 2007.

[4] Narendra Singh Pal, Harjit Pal Singh, R.K.Sarin, Sarabjeet Singh, “Implementation of High Speed FIR Filter using Serial and Parallel Distributed Arithmetic Algorithm” International Journal of Computer Applications, Volume 25– No.7, July 2011

[5] R , Nathiya R , “Realization Of Fir Filter Using Modified Distributed Arithmetic Architecture” , An International of Journal Signal & Image Processing(SIPIJ) Vol.3, No.1, February 2012

[6] JiafengXie, Jianjun He, Guanzheng Tan, "FPGA Realization of FIR filters for high-speed andmedium-speed by using modified distributed arithmetic architectures", Microelectronics journal, 41(2010) 365-370.

[7] Partrick Longa, Ali Miri, "Area-Efficient Fir Filter Design on FPGAs using Distributed Arithmetic", IEEE International Symposium on Signal Processing and Information Technology, pp: 248-252, 2006.

[8] Meher P K, Chandrasekaran S et al , " FPGA Realiazation of FIR Filters by efficient and flexible Systolization using Distributed Arithematic , IEEE Trans. on Signal Processing, Vol. 56,no. 7, pp.3009-3017,Jul 2008.

[9] S.A White, "Applications of the Disributed Arithematic to Digital Signal Processing :A tutorial review", IEEE ASSP Mag., vol. 44,no.3,pp.5-19,Jul.1989.

[10] Jung –Pil Choi,Seung-cheol Shin and Jin-Gyun Chung, "Eficient Rom Size Reduction For Distributed Arithematic,"in Proc.IEEE Int. Symp.Cricuits Syst.(ISCAS),May 2000,Vol. 2, pp. v/125-v/128.