[1.]Anjanappa KARTHIK, [2.]Madhusudhan MALAVIKAA, [3.]Sridhar Koundinya MANOJ

A LITERATURE SURVEY ON FIR DIGITAL FILTERS

[1-3.]Electronics and Communication Engineering, Bangalore Institute of Technology, Bangalore, INDIA

Abstract: In this paper, various efficient memory-based implementation of finite impulse response (FIR) filter have been analyzed. The latency, area, number of registers, delay and memory of various filter designs have been computed and analyzed. The comparison results have been carried out between the Distributed arithmetic (DA)-based computation which is known for efficient memory-based implementation of FIR filter, the conventional multiplier-based FIR filter design and the modified OMS multiplier-based FIR filter design. The performance of these designs helps the circuit designers to choose an area efficient alternative for real time implementation of higher order filters.

Keywords: Digital signal processing (DSP), Finite impulse response (FIR) filters, Look up table (LUT)-based computing, Distributed arithmetic (DA), Very large scale integration (VLSI)

1. INTRODUCTION

Finite Impulse Response filter is one of two main types of digital filters used in DSP applications. Several applications in digital communication, speech processing, seismic signal processing and several other areas of signal processing use large order FIR filters. Nowadays semiconductor memory has become cheaper, faster and more power-efficient. Memory-based computing is used for many digital signal processing (DSP) algorithms, which involve multiplication with a fixed set of coefficients. Memory-based structures or Memory-based systems are those systems where memory components like RAM or ROM is employed either as a part or whole of an arithmetic unit. Various optimization techniques to reduce the memory chips with lower cost include:

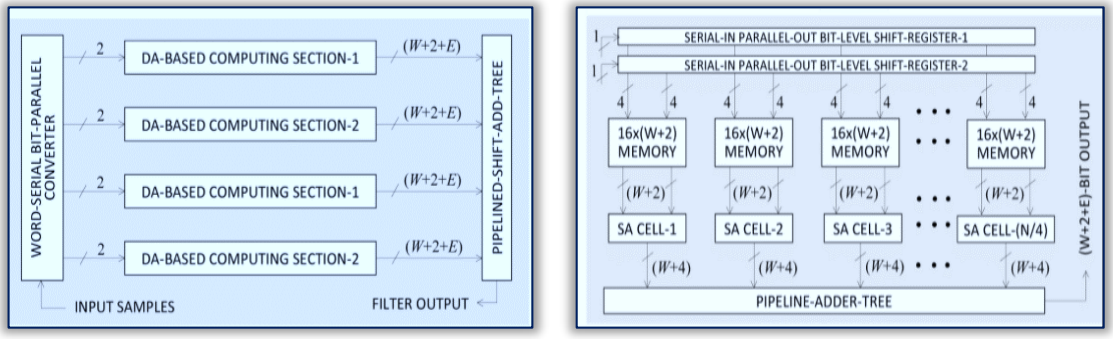
- Pipelining, a technique used in various DSP applications which increases speed for the critical path due to which clock speed is increased and power consumption is reduced.
- Retiming, a generalized form of pipelining which incorporates moving the structural positions of latches or registers to improve the performance of the system.
- Folding, a technique used to minimize the number of functional blocks in synthesizing DSP architectures.
- Unfolding, a technique used for duplicating the functional blocks to increase the throughput of the system.
- Systolic Architecture, a network of PE's that compute and pass data through the system.

The different design techniques for memory-based multipliers to be used in digital signal processing applications are discussed in the following section.

2. TYPES OF MEMORY BASED MULTIPLIERS

— Distributed arithmetic (DA)-based computation

The DA-based FIR filter structure results in minimum area and minimum area-delay product for address-length 4. The pipelined-adder-tree and pipelined-shift-add-tree are used to reduce the number of latches and latency. It consists of a pair of serial-in parallel-out bit-level shift-registers (SIPOSRs), (N/4) memory modules of size [16 x (W+2)], (N/4) shift-add (SA) cells and a pipelined shift-adder-tree. The structure takes N cycles to fill in the SIPOSRs, one cycle for memory access and one cycle to produce the output of the shift-add cell, ($\log_2 N$-2) cycles in the pipelined-adder-tree and two cycles at pipelined shift-add-tree. The latency for this structure is therefore (N+ $\log_2 N$+2) cycles, and has a throughput of one output per cycle.



(a) (b)

Figure 1: DA-based structure for FIR filter. (a)The DA-based FIR filter structure. (b) Structure of each section of the filter. E=$\log_2 N$

— Memory-Based FIR Filter Using Conventional LUT

An N-tap FIR filter for input word length L=8 consists of N memory-units, (N-1) AS cells and a delay register. During each cycle, the 8 bits of present input sample x(n) are fed to all the LUT-multipliers in parallel as a pair of 4-bit addresses X1 and X2. The LUT-multiplier consists of a dual-port memory unit of size [16 x (W+4)]. The Shift-Add (SA) cell, after left-shifting its input by 4 bits, adds the shifted value with its other input to produce a (W+8)-bit output. The output of the multipliers are fed to the add-subtract (AS) cells.

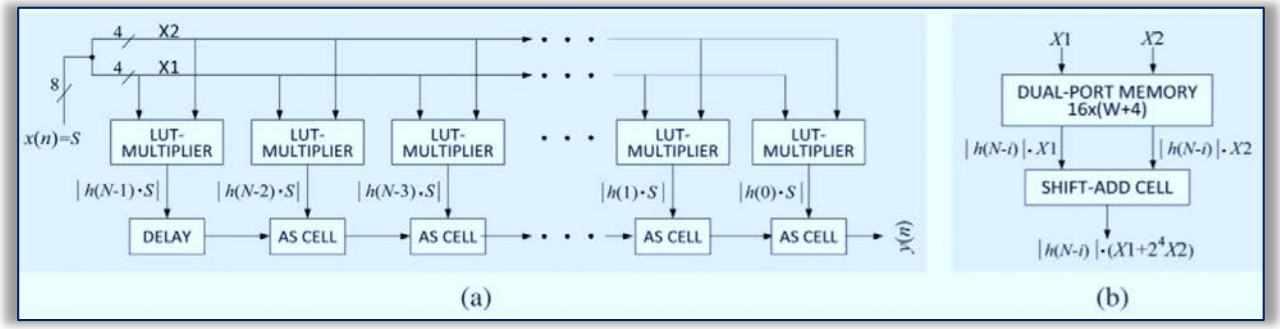


Figure 2: (a) Conventional LUT-multiplier-based structure of an -tap transposed form FIR filter for input-width L=8. (b) Structure of each LUT-multiplier.

— Optimized LUT Using Modified OMS

The APC–OMS combined design for L =5 with coefficient width W consists of an LUT of nine words of (W +4)-bit width, a four-to-nine-line address decoder, an address generator circuit, a barrel shifter and a control circuit for generating the RESET signal. The precomputed values of A × (2i +1) are stored as P_i, for i =0 ,1,2,...,7, at eight consecutive locations of the memory array. The decoder takes the 4-bit address from the address generator and generates nine word-select signals i.e.,{ w_i, for 0≤ i ≤8}, to fetch the referenced word from the LUT. The control circuit generates the control bits (s1 s0), a 2-bit binary equivalent of the required number of shifts according to the equations,

$$s0 = x0 + (x1 + x2)$$
$$s1 = (x0 + x1)$$

The RESET signal can be generated as (d3 AND x4). The address-generator circuit receives the 5-bit input operand X and maps it onto the 4-bit address word (d3d2d1d0).

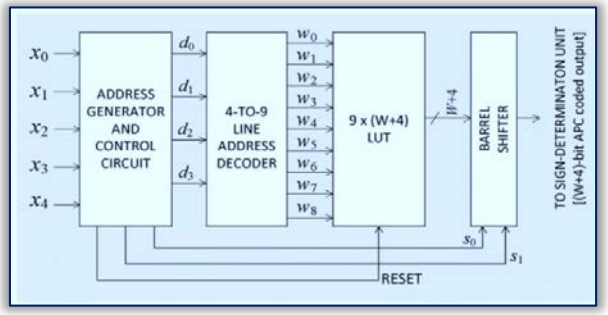


Figure 3: APC–OMS combined LUT design for the multiplication of W-bit fixed coefficient A with 5-bit input X.

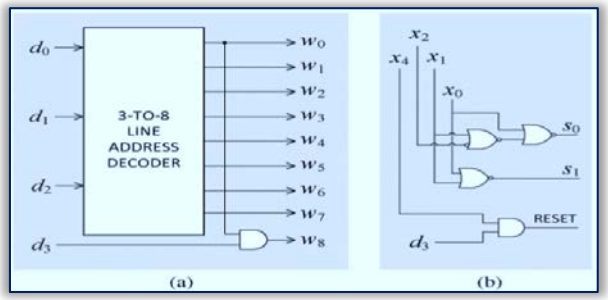


Figure 4: (a) Four-to-nine-line address-decoder. (b) Control circuit for generation of S0, S1and RESET

Table 1. OMS-Based design of the LUT of APC words for L =5

Input X'- x_3' x_2' x_1' x_0'	Product value	# of shifts	Shifted input, X"	Stored APC word	Address d_3 d_2 d_1 d_0
0 0 0 1	A	0	0 0 0 1	P0 = A	0 0 0 0
0 0 1 0	2 x A	1			
0 1 0 0	4 x A	2			
1 0 0 0	8 x A	3			
0 0 1 1	3A	0	0 0 1 1	P1 = 3A	0 0 0 1
0 1 1 0	2 x 3A	1			
1 1 0 0	4 x 3A	2			
0 1 0 1	5A	0	0 1 0 1	P2 = 5A	0 0 1 0
1 0 1 0	2 x 5A	1			
0 1 1 1	7A	0	0 1 1 1	P3 = 7A	0 0 1 1
1 1 1 0	2 x 7A	1			
1 0 0 1	9A	0	1 0 0 1	P4 = 9A	0 1 0 0
1 0 1 1	11A	0	1 0 1 1	P5 = 11A	0 1 0 1
1 1 0 1	13A	0	1 1 0 1	P6 = 13A	0 1 1 0
1 1 1 1	15A	0	1 1 1 1	P7 = 15A	0 1 1 1

— **Implementation of the LUT Multiplier Using APC for L =5**

For L=5 the structure of the LUT-based multiplier using APC technique consists of a four-input LUT of 16 words to store the APC values of product words as shown in the sixth column of Table 2, except the last row, where 2A is stored for input X = (00000) instead of storing a "0" for input X = (10000). It consists of an address-mapping circuit and an add/subtract circuit. The address-mapping circuit generates the desired address ($x'_3\ x'_2\ x'_1\ x'_0$). The address mapping can be implemented by multiplexing XL and X'_L using x_4 as the control bit. The address-mapping circuit can be optimized by realization of three XOR gates, three AND gates, two OR gates, and a NOT gate. The RESET is generated by a control circuit. The output of the LUT is added with or subtracted from 16A for x_4 =1 or 0, respectively. Thus, x4 is used as the control for the add/subtract cell.

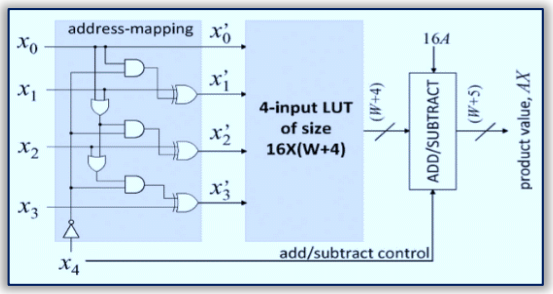


Figure 5: LUT-Based Multiplier for L =5 using the APC technique

Table 2. APC words for different input values for L =5

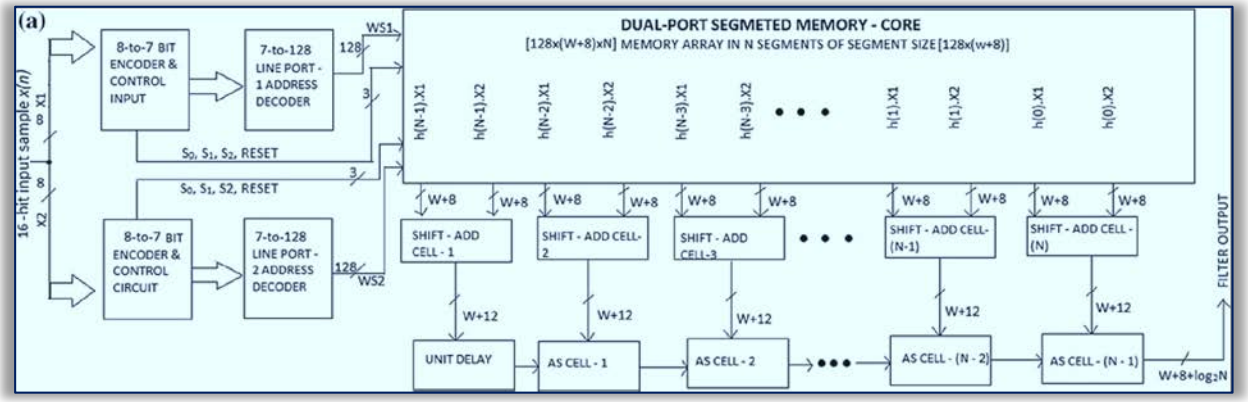
Input, X	Product values	Input, X	Product values	Address $x_3'\ x_2'\ x_1'\ x_0'$	APC words
00001	A	11111	31A	1111	15A
00010	2A	11110	30A	1110	14A
00011	3A	11101	29A	1101	13A
00100	4A	11100	28A	1100	12A
00101	5A	11011	27A	1011	11A
00110	6A	11010	26A	1010	10A
00111	7A	11001	25A	1001	9A
01000	8A	11000	24A	1000	8A
01001	9A	10111	23A	0111	7A
01010	10A	10110	22A	0110	6A
01011	11A	10101	21A	0101	5A
01100	12A	10100	20A	0100	4A
01101	13A	10011	19A	0011	3A
01110	14A	10010	18A	0010	2A
01111	15A	10001	17A	0001	A
10000	16A	10000	16A	0000	0

— **Modified OMS for LUT Optimization:**

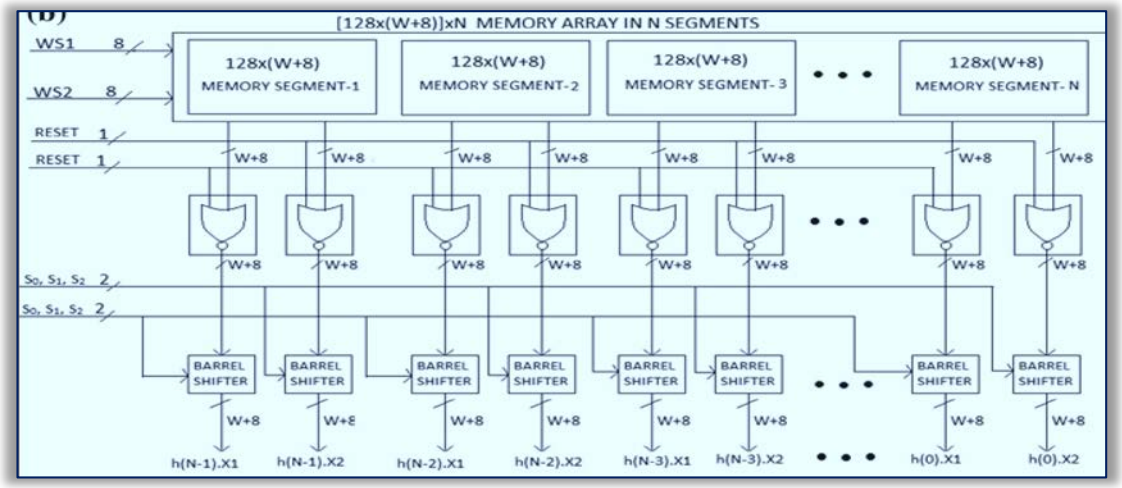
The memory-based structure of FIR filter (for 16-bit input) using LUT design is different from conventional design in two aspects:

- Instead of using N memory modules for N segments, only one memory module is used which reduces the hardware complexity for two (N-1) decoder circuits.
- Odd multiple storage is used in place of conventional design, so that multiplication using a dual-port memory segment could be implemented by $2^{L/2}/2$ words in LUT.

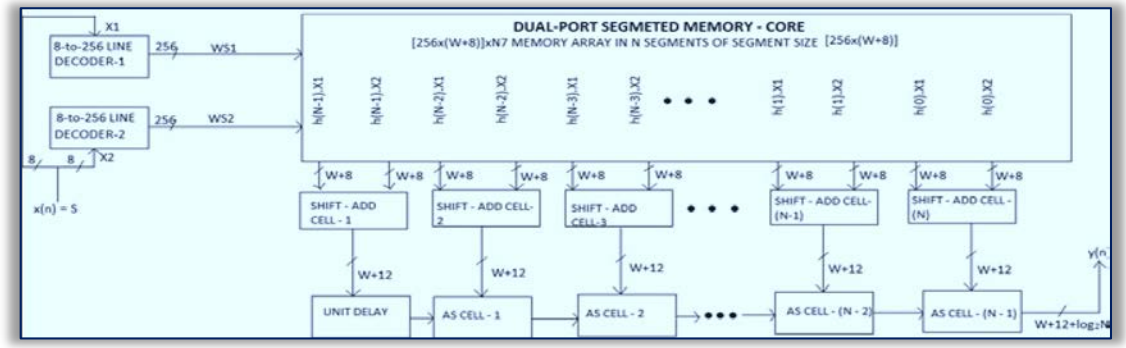
As shown in figure 6, the compiled representation of FIR filter has a dual-port segmented memory core, a delay register, and an array of N shift-add (SA) cells, (N-1) AS cells. It also consists of control circuits, a pair of 8-to-7-bit encoders and a pair of 7-to-128 line decoders to carry out the necessary control signals and word-select signals for the dual-port memory core (which consists [128 (W + 8)] N array of bit-level is devised in 128 rows of [(W + 8) N]-bit width).



(a)



(b)



(c)

Figure 6: a) Structure of Nth order FIR filter using OMS-LUT multiplier for 16 bit input. b) Dual-port segment memory core for Nth order filter. c) LUT multiplier-based structure of an N-tap FIR filter by transposed firm realization using segmented memory core using 16-bit input

3. COMPARISON OF DIFFERENT FIR DIGITAL FILTER DESIGNS

— Latency Comparison for different types of filter designs

From the graph above one can observe that the latency is higher for DA-Based filter and is least for Modified OMS for LUT. Conventional LUT Multiplier lies in between the two designs.

Table 3: Latencies of LUT-Multiplier-Based and DA Based designs for different filter order N and input word-size L

N	LUT -Based Design			DA-Based Design		
	L = 8	L = 16	L = 32	L = 8	L = 16	L = 32
8	13	14	15	16	17	18
16	21	22	23	25	26	27
32	37	38	39	42	43	44
64	69	70	71	75	76	77
128	133	134	135	140	141	142

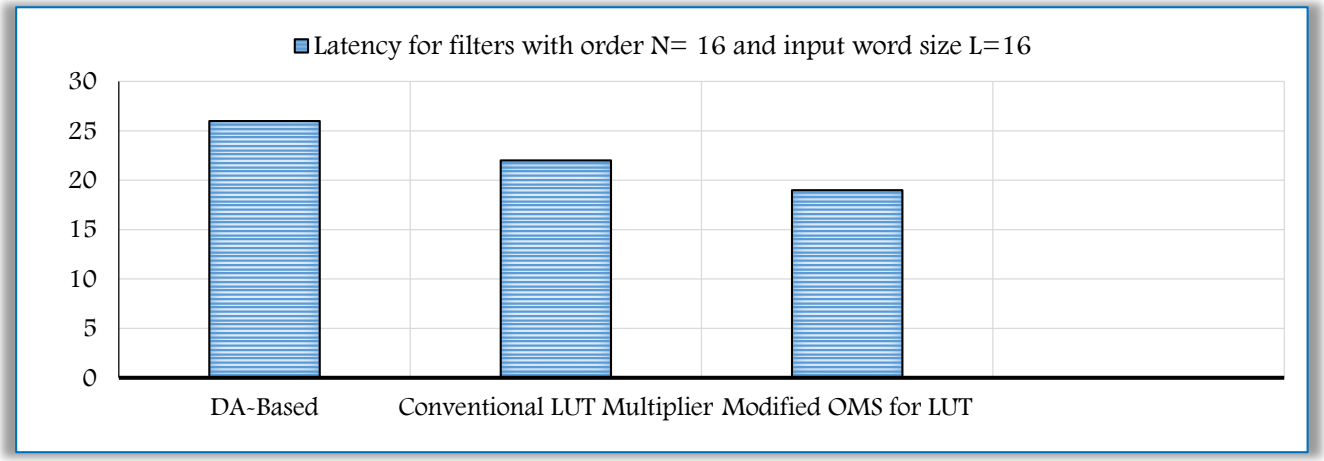


Figure 7: Different types of filter designs vs Latency

Table 4: Comparison of different filter designs for L=16 and N=16

Design	DA-Based	Memory-based using conventional LUT	Modified OMS for LUT Optimization
Latency	26	22	
Area (sq.um)	78895.96	82519.92	68558.92
Memory	$8k[(W+E-1)2^{N/2}$	[256×(W+8)×N]bit memory	[128×(W+8) ×N]bit memory
Registers	(W+8K+E)bit register	49,224(1-bit registers	39,208(1-bit registers)

— **Area Comparison for different types of filter designs**

The difference in area between Conventional LUT Multiplier and DA Based is less compared to difference between Conventional LUT Multiplier and Modified OMS for LUT. Hence area is much larger for Conventional LUT Multiplier followed by DA Based and then Modified OMS for LUT.

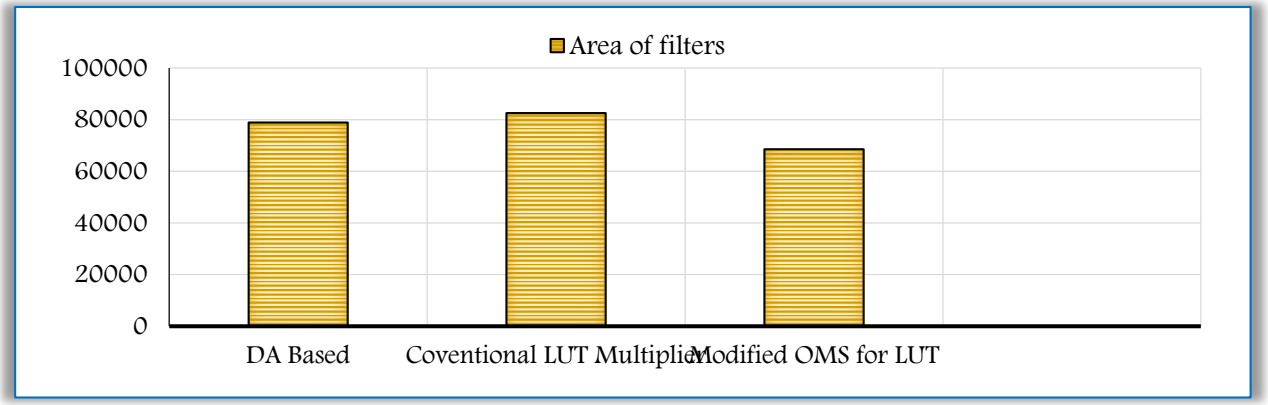


Figure 8: Different types of filter designs vs Area

— **Register Comparison for different types of filter designs**

From the graph conventional LUT Multiplier uses more number of register than Modified OMS for LUT.

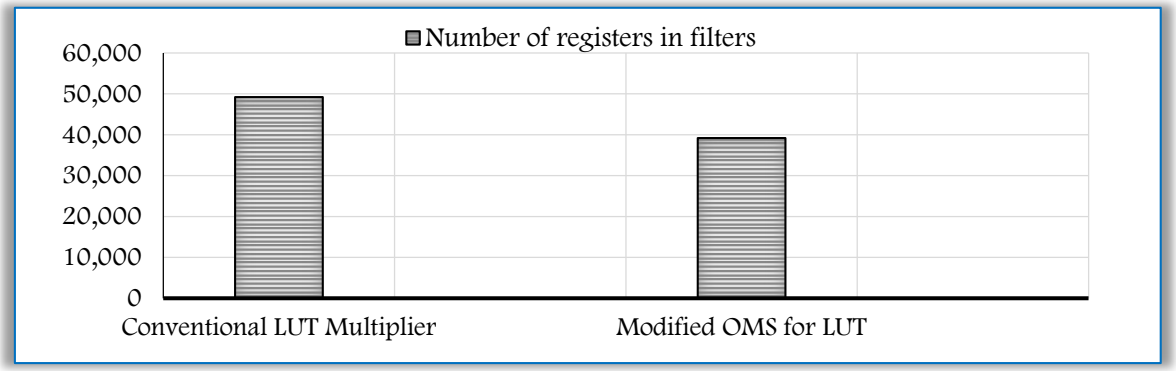


Figure 9: Different types of filter designs vs Number of Registers

— **Delay Comparison for different types of filter designs**

As observed from the delay graph, the delay is very much higher in DA Based compared to all other designs. LUT Optimization for modified OMS has the least delay. APC technique has higher delay than Modified OMS for LUT.

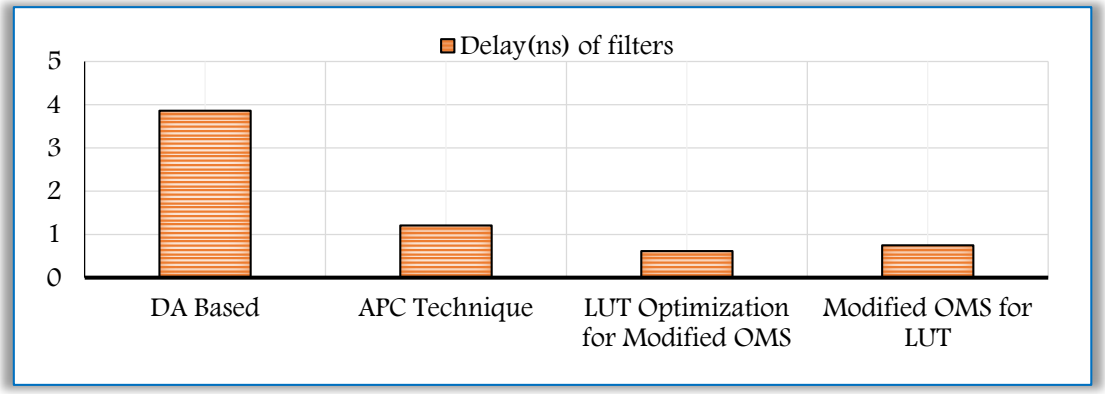


Figure 10: Different types of filter designs vs Delay

— Memory Comparison for different types of filter designs

For given order and input word size, memory is highest in conventional LUT Multiplier followed by DA Based and is lowest for Modified OMS for LUT.

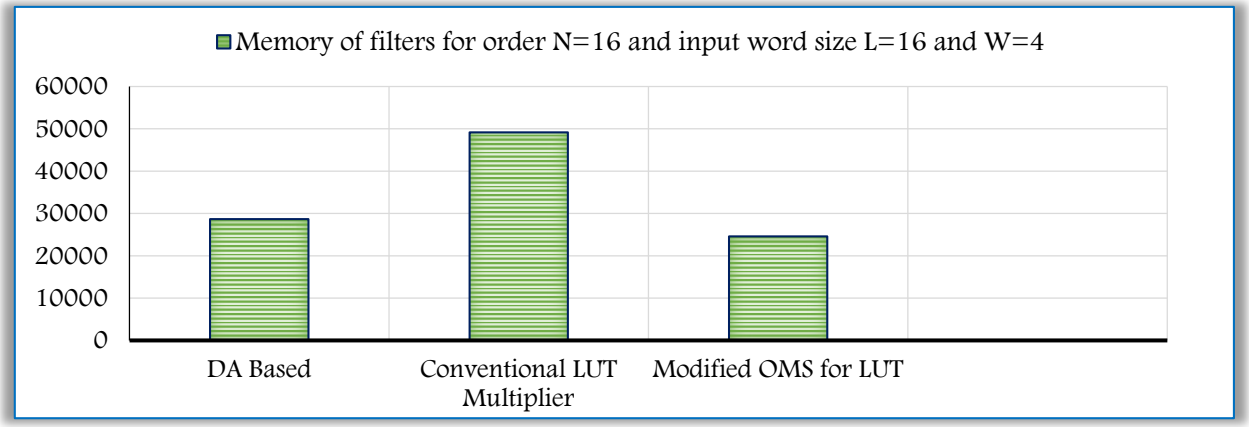


Figure 11: Different types of filter designs vs Memory

4. CONCLUSION

The different filter designs have been studied and comparison of different filter designs in terms of latency, area, register, delay, and memory is done. Based on these comparisons we observe that modified OMS multiplier-based FIR filter design requires less memory in comparison to conventional multiplier-based FIR filter and for the address of 16 bit it reduces the LUT-size by the use of a barrel shifter which is of three-stage and (W+8) NOR gates. Comparing this filter design with the conventional design, we see that the memory requirement is less in comparison to the conventional design. The complexity for the higher-order and higher input is reduced by the usage of modified OMS multiplier based FIR filter.

References

[1] Proakis JG, Manolakis DG (1996) Digital signal processing: principle algorithms and applications. Prentice-Hall, Upper Saddle River, NJ
[2] Parhi KK (1999) VLSI digital signal processing systems: design and implementation. Wiley, New York
[3] Meher PK (2010) New approach to Look-Up-Table design and memory-based realization of FIR digital filter. IEEE Trans Circuit Syst-I 57(3)
[4] Meher PK, Senior member IEEE (2010) LUT optimization for memory-based computation. IEEE Trans Circuit Syst-2 57(04)
[5] Sharma D, Vinitha CS, Kandari R, Neeraj, Modified OMS-LUT design for memory-based multiplication. In: IEEE proceedings, Computing for sustainable global development, Conference ID: 42835
[6] Sharma D, Sharma S, Payal S, Vinitha CS A new pipelined memory-based FIR Filter. In: IEEE proceedings, computing for sustainable global development, Conference ID: 42835

ISSN 1584 – 2665 (printed version); ISSN 2601 – 2332 (online); ISSN-L 1584 – 2665