



FPGA Based High Speed 8-Tap FIR Filter

Bogi Kartheek^(✉) and N. Purnachand

School of Electronics Engineering, VIT-AP University, Inavolu 522237, A.P., India

Abstract. FIR (Finite Impulse Response) filters play a predominant role in digital signal processing due to their phase linearity. It can be easily implemented in the hardware and is very stable compared to IIR due to its non-feedback nature. The main drawback of FIR filter is that it takes more time to compute since there are more coefficients. Hence it is very essential to speed up the multiplication process of the FIR filter system. One of the best techniques to improve the speed is to use memory-based computation using Anti-Symmetric product coding (APC) and Odd multiple storage (OMS) techniques. This paper uses the APC-OMS based multiplier in the design of 8-TAP FIR filter. The APC-OMS multiplier used in the 8-TAP FIR filter reduces the LUT size. The design is implemented and verified using Verilog HDL on a Xilinx Zynq-7000 series FPGA. The proposed design reduces the FPGA area utilization and improves the performance compared to the some of the state-of-the-art works. The proposed design can be operated at a maximum clock frequency of 464.04 MHz.

Keywords: FIR filters · Odd multiple storage · Anti-symmetric product coding · FPGA · Look-up-table

1 Introduction

The finite impulse response (FIR) filter is the significant filter to be used in digital signal processing Applications. “FIR digital filters find extensive applications in mobile communication systems such as channel equalization, matched filtering, and pulse shaping, due to their absolute stability and linear phase properties” [2]. FIR filters are used for certain applications where phase sensitivity is essential. Some of the applications are data communication, seismology, and mastering. FIR filters are designed on FPGA mainly because of its dedicated Hardware. The multiplier in FIR filter is designed using APC-OMS technique. FIR filter structure consists of Multipliers, Adders and Delay elements are shown in Fig. 1.

“Memory-based computing is a class of dedicated systems, where the computational functions are carried out by look-up tables (LUT)” [3, 5]. “Memory-based computing is well suited for many digital signal processing (DSP) algorithms, which involve multiplication with a fixed set of coefficients” [7]. “Optimization of LUT for Memory based computing can be performed using the APC-OMS technique and the odd multiples of the fixed coefficients are required to store in LUT which is termed as the odd multiple storage (OMS)” [4]. “While in the antisymmetric product coding (APC) approach, the product words are stored as antisymmetric pairs” [6].

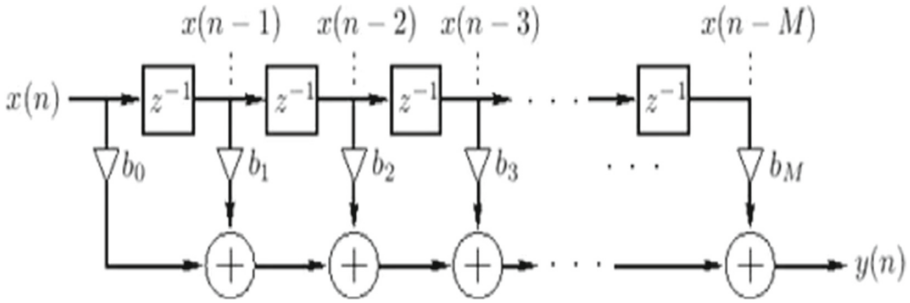


Fig. 1. Structure of FIR filter.

The equation of nth order digital filter (FIR) can be written as:

$$y(n) = \sum_{i=0}^N b_i * (x[n - i]) \tag{1}$$

- x[n] represents the I/p of the signal
- y[n] represents the o/p of the signal
- N represents order of an FIR filter (Delay points)
- N+1 represents no. of Taps.

2 Related Works

In [9], the authors proposed a high-speed FIR filter using adders and shifters and implemented the design on a Xilinx FPGA and achieved a maximum clock frequency of 235.026 MHz. The author has used add and shift method instead of multiplier to reduce chip size.

In [8], the authors proposed the design of FIR by means of vedic multiplier and implemented on Xilinx FPGA and achieved maximum clock frequency of 109 MHz.

In [7], the authors proposed LUT based multiplier design using APC-OMS based technique and implemented using TSMC 90nm technology. The author also synthesized CSD- based multiplier using the same technology library and compared it with LUT design and described the area utilization of LUT design is efficient over CSD- based multiplier.

In [11], the authors proposed an efficient FIR filter EMS multiplier and implemented using virtex-7, and achieved a maximum clock frequency of 433.46 MHz for the 16-TAP filter of input word size 4. Operating frequency can be higher by using APC-OMS based multiplier.

In [12], the authors proposed a FIR filter using birecorder multiplier and implemented. In a Xilinx FPGA and achieved maximum frequency of 157.227 MHz.

3 Techniques for Optimization of Memory

3.1 APC for the Optimization of Look up Table

The word length $L = 5$ of an input X and its values can be seen in the first and the third column of Table 1. Product values are defined by the multiplication of corresponding

X input with the fixed coefficient A. The sum of the product values, which are situated at the 2nd and 4th column equals 32A. The values situated at third column are two's compliment to the values which are situated at the first column of the Table 1. The final APC words are to be seen corresponding to different Addresses, which were written for different input values. These address inputs are located in the 5th column of Table 1. The terms u and v are defined as product values and are situated at second and fourth column. The subsequent equation determines the values of both u and v.

$$v = \frac{(u + v)}{2} - \frac{(v - u)}{2} \quad \text{and} \quad u = \frac{(u + v)}{2} + \frac{(v - u)}{2} \quad (2)$$

We know $(u + v) = 32A$, substituting this equation in Eq. (2)

$$v = 16A - \left[\frac{(v - u)}{2} \right] \quad \text{and} \quad u = 16A + \left[\frac{(v - u)}{2} \right] \quad (3)$$

In Eq. 3, we can see the negative symmetry on u and v. since considering the nature we can reduce the LUT size to half by storing the $[v-u]/2$ for the inputs situated at the same row of the table. The values in 2nd and 4th are known as product values. Those values are asymmetric to each other. The product can be found by using the next equation.

$$\text{Product word} = (\text{APC word}) * (\text{sign value}) + 16A \quad (4)$$

The product word can be found by the addition of 16A to the multiplication equation of sign value of MSB of input X with APC word. If the MSB (most significant bit) of input X is 0 then the sign value is -1 . Likewise, if the MSB of the input is 1, then the sign value is -1 .

3.2 OMS for Optimization of Look up Table

Address inputs from APC are taken as inputs in OMS (odd multiple storage) method. Those address inputs are situated in the 1st column of Table 2. And the corresponding product values of the address inputs are located in the 2nd column. The Required input values of shifted input are accessed by doing a left shift operation. Shifted APC can be found in the 5th column.

An active high signal (RESET signal) is given to reset the LUT output to derive the APC word 0. For 00000(X), the encoded word "16A" can be derived by left shifting the "2A" 3 times. It is stored at the address 1000.

Product value of input(X) 00000 is 0. For the input 00000(X), the APC word to be stored is 16A. APC words are said to be dissimilar to each other due to its procedure.

The inputs and product values are said to be unsigned values. It reduces half of the LUT size. Initially it requires 32 address locations, now it is reduced to half. It will be further reduced when these values are given to OMS technique.

Addresses from APC design are given as an input to OMS (odd multiple storage) in order to further reduce the size of LUT. The LUT size can be reduced by only storing the odd address and its corresponding product values.

Table 1. APC words of dissimilar various I/p values (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

Odd multiple storage design reduces the LUT size further lower than the APC (Anti-symmetric product coding). A maximum of three left shifts are produced by using Barrel shifter. It may possibly be further used to derive rest of all the even multiples of coefficient A.

In the Table 2, it has been seen that all the stored APC words are the odd multiples of coefficient A. Address d3, d2, d1, d0 derives the storage address of the APC words which defines the 5-bit input values.

Odd multiple storage technique can be the efficient way to reduce the size of the LUT. This can further reduce the design and resource utilization of any FPGA. Here it leads to reduce power dissipation as the resources are low.

4 Implementation of the Look up Table Multiplier

4.1 Design on Look up Table Multiplier Using APC-OMS Based Technique

The subsequent block diagram in Fig. 2 is the multiplier, which uses the APC-OMS technique. It is a Look up Table (LUT) multiplier. Address Generator and the control circuit block takes 5bit input X of L = 5 and generates output of 4bit address. Control circuit is used to control Barrel shifter by using S0 and S1. Address decoder takes d0, d1, d2 and d3 as inputs and decodes nine outputs. The nine outputs are given as I/p's for LUT block, there it generates 9 words with the width of 4bit.

Table 2. OMS Words of dissimilar various I/p values (L = 5)

Input (X ⁱ) x ₃ 'x ₂ 'x ₁ 'x ₀ '	Product values	No. of shifts	Shifted input (X ^m)	Stored APC word	Address d ₃ d ₂ d ₁ d ₀
0001	A	0	0001	A	0000
0010	2*A	1			
0100	4*A	2			
1000	8*A	3			
0011	3A	0	0011	3A	0001
0110	2*3A	1			
1100	4*3A	2			
0101	5A	0	0101	5A	0010
1010	2*5S	1			
0111	7A	0	0111	7A	0011
1110	2*7A	1			
1001	9A	0	1001	9A	0100
1011	11A	0	1011	11A	0101
1101	13A	0	1101	13A	0110
1111	15A	0	1111	15A	0111

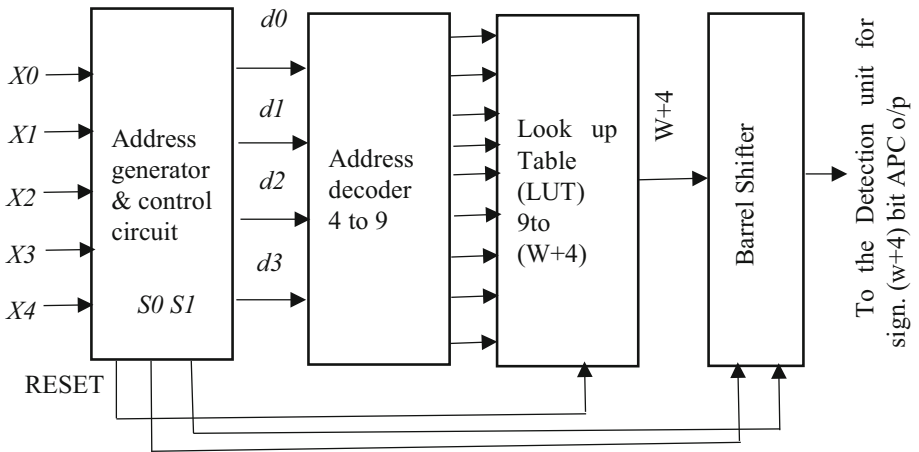


Fig. 2. Look up table multiplier using APC-OMS technique

The shifted product value from Barrel shifted goes to sign determination of input (X). The sign can be determined by the MSB (most significant bit) of the input X. If the input has the MSB as 0. Then the sign value should be taken as -1 and if the MSB of the input is 1, then the sign value is taken as -1.

5 Simulation Results

The 8-TAP FIR filter using APC-OMS based lookup table (LUT) multiplier is coded in Verilog HDL and simulated and synthesized in Xilinx vivado 2018.3. Figure 3, 4, 5, 6, 7, 8 and 9 and Table 3 shows the o/p waveforms of the FIR filter and its RTL schematic, synthesis, and summary report. Summary reports show the LUT utilization percentage amongst the Overall LUT's and also flip-flops utilization. This shows the consumption of Area and resources are very low. So, gradually the power consumption also reduces. Since, it is an 8-tap FIR filter, it requires 8 clock cycles to produce the output.

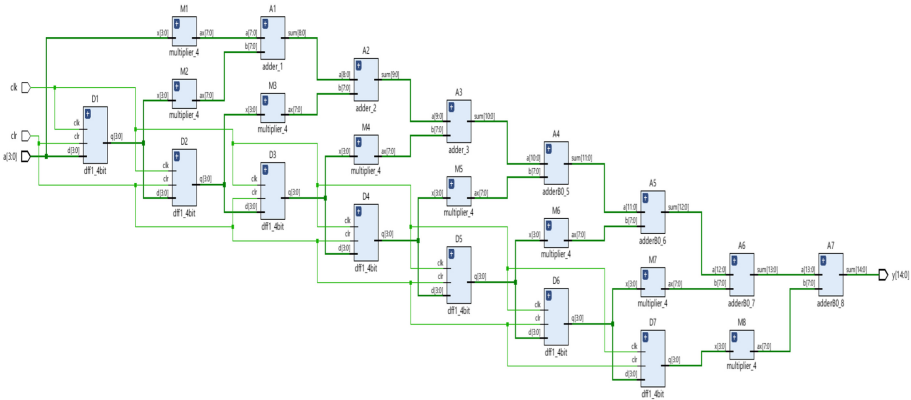


Fig. 3. RTL Schematic

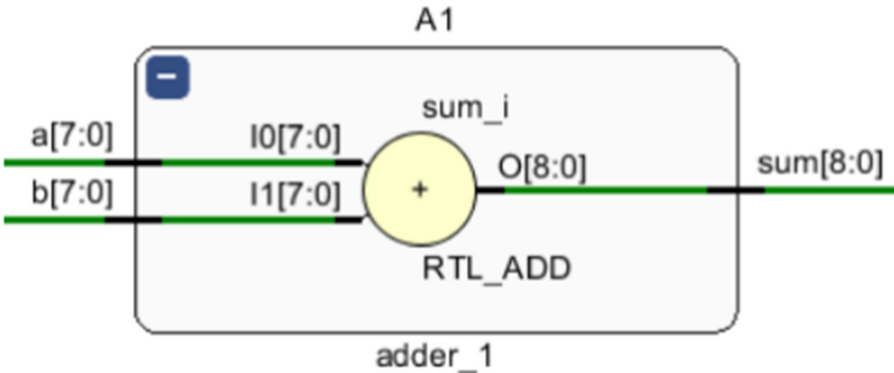


Fig. 4. Schematic diagram of adder module in FIR filter

Schematic diagram of Adder module of FIR filter is shown in the Fig. 4. Schematic block generates RTL_ADD (register transfer level) to describe an adder. Input A[7:0] and b[7:0] is given to the RTL input I0[7:0] and I1[7:0] and generates the output of O[8:0]. The output can be taken as sum [8:0]. Likewise, the synthesis diagram of D-Flip-Flop

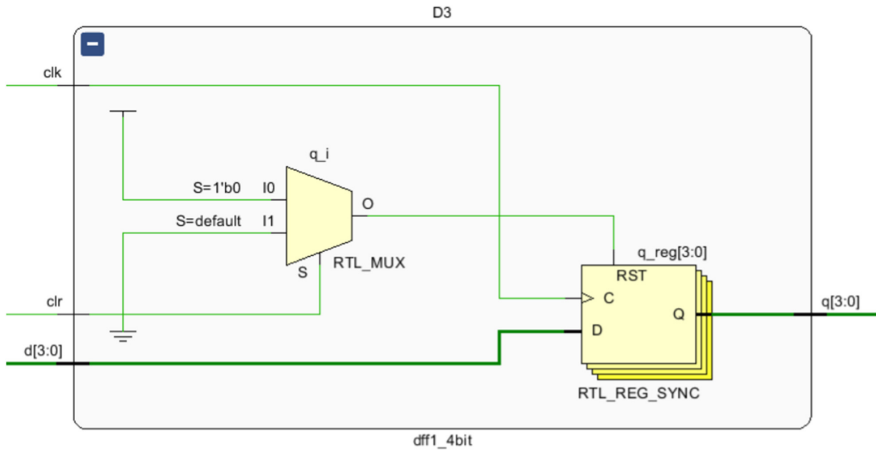


Fig. 5. Schematic diagram of D-Flip flop module in FIR filter

uses RTL_MUX and RTL_REG_SYNC. There are 7 such adders used to design the FIR filter. Since, 8-TAP filter requires 7 adders to design it.

From Fig. 5, the D-flip flop delays one clock period for the input signal to give the output. The delayed input can be convoluted with the multiplier and to be added with the next delayed input.

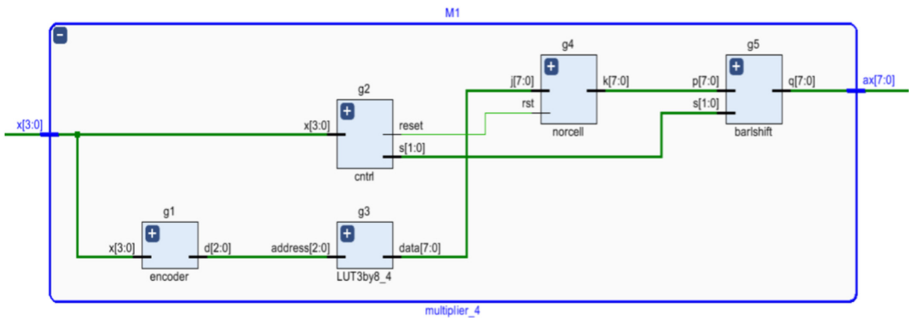


Fig. 6. Schematic diagram of Multiplier module in FIR filter

RTL Schematic is done using Xilinx vivado tool. The schematic diagram shows the multipliers, adders and the D-Flip flops. D Flip flops are used to delay the inputs. Here there are 8 multipliers, since it is an 8tap FIR filter and 7 D-Flip flops. The 7 Flip-flops represents the order of FIR filter. The multiplier has encoder, control, LUT3X8, nor cell and Barrel shifter. Control block is used to manage the complete signal. Nor cell is used for RESET operation. Figure 6 shows the schematic representation of the multiplier module. The barrel shifter shifts the products words based on the select lines S0 and S1.

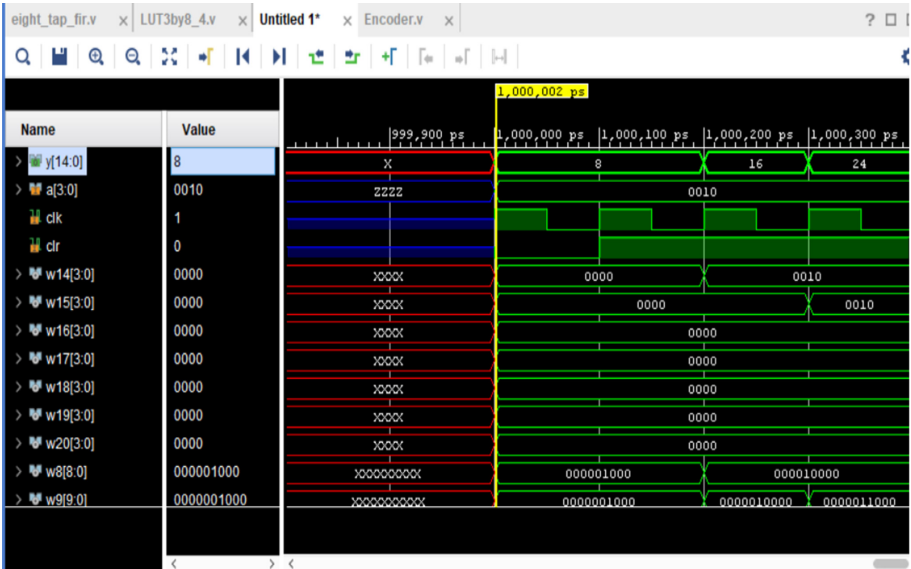


Fig. 7. Simulation result of 8-TAP FIR filter

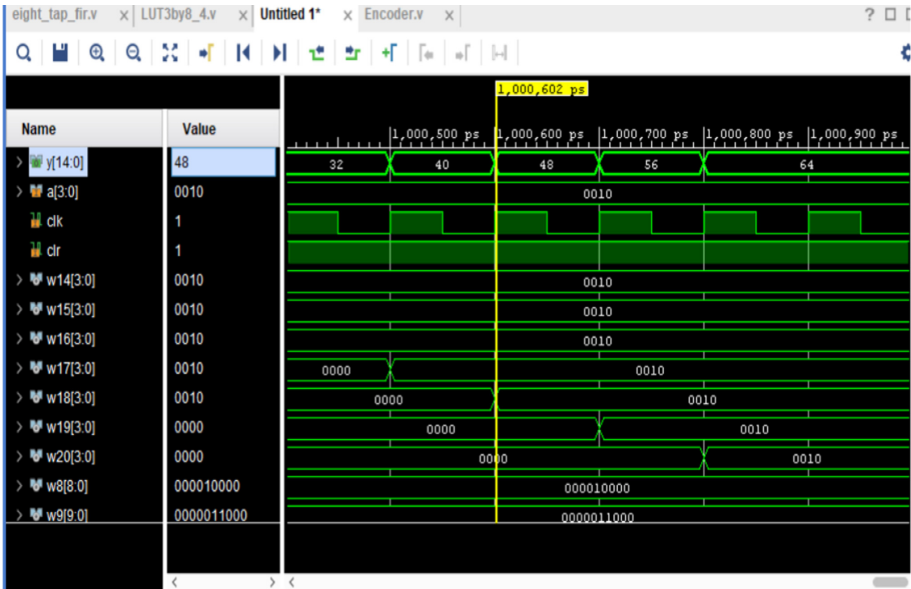


Fig. 8. Simulation result of 8-TAP FIR filter

From Fig. 7 and 8, the given input can be convoluted with the multiplier with symmetrical coefficients of 4. It gives linear phase response due to its stability. The simulation runs in ps(picoseconds) scale.

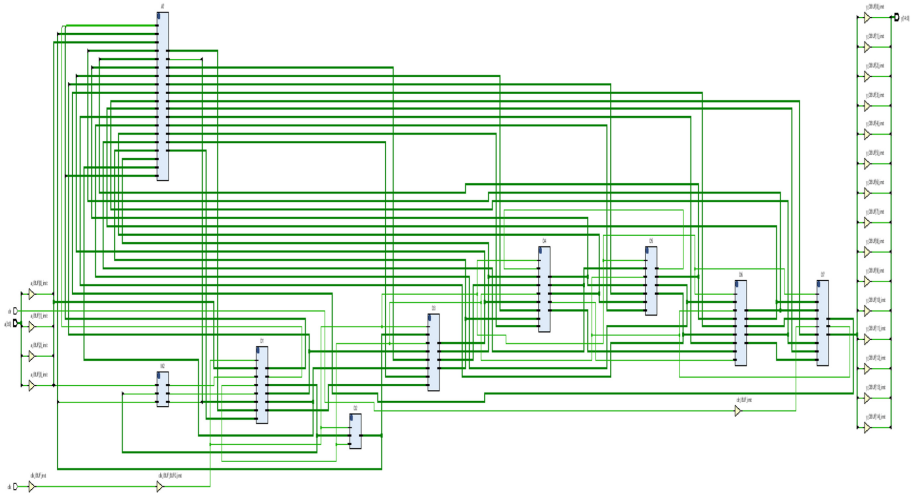


Fig. 9. Synthesis (schematic) on Zynq-7000 board library

Synthesis represents the design of FIR filter on zynq-7000 board and the usage of I/O ports, net lists, buffers and flip-flops of the board library.

In Fig. 7, we have given 0010 as an input (a [3:0]) and y [14:0] as an output. It took 8 cycles to produce output. The output y [14:0] is an amplified output. The usage of multipliers, symmetric coefficients upsurges the amplitude and response of the input signal. The maximum frequency response can be determined by creating the timing constraints in Xilinx. XDC file can be created at Top module of the design hierarchy. The input clock waveform of 10 ns is given to the implemented design to get the analysis of design regarding setup and hold time violations. The implemented design is free of setup and holds time violations and attained the max frequency of 464.04 MHz.

Table 3. Summary of synthesis report

Synthesis results			
	[8]	[9]	Proposed work
FPGA & technology	virtex-5 (65 nm)	Spartan-3 (45 nm)	zynq-XC7Z014S (28 nm)
# slices	-	144	26
FPGA slice LUTs	204	92	29
FPGA slice flip-flops	-	-	28
Max. clock frequency	109 MHz	235.02 MHz	464.04 MHz

The proposed work is compared with two other technologies. The usage of FPGA and Technology, no. of slice LUT's, no. of FPGA slice Flip-Flops and Maximum clock frequency is written in 1st column of Table 3. The utilisation of slice LUT's are lesser

than the compared reference FIR filters. We used zynq-XC7Z014S 28 nm technology to design and implementation of the FIR filter. Zynq-XC7Z014S is the library of zynq-7000 product family.

6 Conclusion

The Design of an 8-tap FIR filter using APC-OMS look up table multiplier remains completed by writing Verilog HDL code in Xilinx vivado 2018.3. The result shows the utilization of LUT's in FPGA (zynq-7000) library is 0.11% whereas the utilization of Flip-Flops is 0.03%. The maximum operating clock frequency of this design is 464.04 MHz and the Area utilization for this design is very low and thus it leads to less power consumption. The design of 16, 32 tap FIR filters can further implemented.

7 Future Scope

The design can be extended for higher order filters (like 16, 32 TAP filters) where high performance is required. It can be used in different FIR applications like mastering, study of seismology etc.

References

1. Proakis, J.G., Manolakis, D.G.: Digital Signal Processing: Principles, Algorithms and Applications. Prentice-Hall, Upper Saddle River, NJ (1996)
2. Vinod, A.P., Lai, E.: Low power and high speed implementation of FIR filters for software define radio receivers. *IEEE Trans. Wirel. Commun.* **5**(7), 1669–1675 (2006)
3. Schaller, R.R.: Technological innovation in the semiconductor industry: a case study of the international technology roadmap for semiconductors (itrs), Ph.D. dissertation, George Mason University (2004)
4. Meher, P.K.: New approach to LUT implementation and accumulation for memory-based multiplication. In: Proceedings of the IEEE ISCAS, pp. 453–456 (May 2009)
5. Meher, P.K.: Memory-based hardware for resource-constraint digital signal processing systems. In: 2007 6th International Conference on Information, Communications and Signal Processing. IEEE (2007)
6. Meher, P.K.: New look-up-table optimizations for memory-based multiplication. In: Proceedings of the 2009 12th International Symposium on Integrated Circuits. IEEE (2009)
7. Meher, P.K.: LUT optimization for memory-based computation. *IEEE Trans. Circ. Syst. II Express Briefs* **57**(4), 285–289 (2010)
8. AlJuffri, A.A., et al.: FPGA implementation of scalable microprogrammed FIR filter architectures using Wallace tree and Vedic multipliers. In: 2015 3rd International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAECE). IEEE (2015)
9. Thakur, R., Khare, K.: High speed FPGA implementation of FIR filter for DSP applications. *Int. J. Model. Optim.* **3**(1), 92–94 (2013)
10. Paul, L., Paul, R.: Modified APC-OMS technique for memory based computing. *Procedia Technol.* **25**, 606–612 (2016)
11. Vinitha, C.S., Sharma, R.K.: An efficient LUT design on FPGA for memory-based multiplication. *Iran. J. Electr. Electron. Eng.* **15**(4), 462–476 (2019)
12. Jayashree, M.: Design of high speed and area efficient FIR filter architecture using modified adder and multiplier. *Int. J. Eng. Tech.* **4**, 537–543 (2018)