Received: July 27, 2017 250

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Implementation of FIR Filter for Low Power and Area Minimization Using Shift –Add Method without Multipliers

Sumalatha Madugula[1]* Panchala Venkata Naganjaneyulu[2] Kodati Satya Prasad[1]

[1]*Jawaharlal Nehru Technological University, India*
[2]*MVR College of Engineering & Technology, India*
* Corresponding author's Email: sumalathaphd2017@gmail.com

Abstract: In VLSI design, performance parameters such as speed and hardware cost are important facts during the implementation of complex algorithms. In this paper, Low Cost Carry SeLect Adder Finite Impulse Response (LC-CSLA-FIR) is introduced to perform the FIR filter operations. With the help of shifter-I, shifter-II, and Shifter-III circuits, the FIR filter operation is performed without using any multipliers. In this new architecture, there is no need of storing the co-efficient in the ROM and number of multipliers. Due to the absence of multipliers, the quantity of hardware is reduced and the performance of the architecture is increase. With the help of Verilog code, the FIR filter architecture is verified in Modelsim software. After Field Programmable Gate Array (FPGA) analysis, Look Up Table (LUT), slices, flip flops, frequency and ASIC implementation area, power, delay, Area Power Product (APP), Area Delay Product (ADP) is improved in LC-CSLA-RFIR method compare to conventional methods.

Keywords: Shifter, Carry SeLect adder, Finite impulse response filter, Read only memory, Multipliers.

1. Introduction

Nowadays Finite impulse response filters have got huge importance in fields like digital signal processing, wireless communication and also in the fields where the complexity is less. People are expecting a system which have good accuracy, efficiency and error-free in their operation. For the past several years, some of the researcher are working on the improvement of the FIR filter implementation for many applications. The shift and add methods for designing the fast FIR filters in FPGA implementations enhance the speed of the operations of the systems [1]. A new approach is used to increase the computational speed in the designing of the filter, which has L paths and L blocks in their structures [2]. Block by block computation is the one which minimize the computation time of the system with the overlapped block filters [3]. The major work of the researcher is to reduce complexity, computation time and the area of the hardware to implement in FPGA. Many researchers have found the various forms of the digital filter design, to overcome above mentioned limitations. A linear phase filter design is used to minimize the number of complex adders and the Mixed Integer Linear Programming (MILP) was efficiently traversed the discrete filter coefficient [4]. A new iterated short convolution algorithm has been developed with the support of a mixed radix algorithm and fast convolution algorithm. In this long convolution can be split into various levels of short convolutions. This convolution algorithm iteratively finds the computations of all the short filters to enhance the speed of the algorithm. This algorithm is very much needed when the length of the filter and the level of parallelism are very long [5]. The new FIR filter is presented, which performs the symmetric coefficient and minimize the sub filter section in post processing blocks [6].

Another author has introduced the new method for area efficiency is spanning tree based on an unaltered booth multiplier in direct form realization. This method minimizes the area in the design of FIR filter, which was implemented by using Xilinx 14.2 ISE tool and Modelsim, programming in VHDL. To find out filter coefficients, MATLAB Simulink Tool

has been used in FIR filter [7]. To design an efficient Reconfigurable Interpolation Root-Raised Cosine (RRC) FIR filter, the author has followed the Binary Common Subexpression Elimination (BCSE) approach. In this method filter coefficients changes during the runtime for multi-standard wireless communication system called Software Defined Radio (SDR). This type of structures improves the performance of the system by considering parameters like speed and area compared to the previous Multiply Accumulate (MAC) based and Distributed Arithmetic (DA) based approaches [8]. In the new algorithm, another research work has introduced the adaptive FIR filter based on the linear prediction in order to suppress interference of the radio frequency signals. In this, they were compared in real time conditions of several variants of the low pass FIR filter with different lengths and different coefficient widths with the goal to minimize the power consumption of the radio station by keeping requiring accuracy for noise minimization [9].

The Radio Frequency Interference (RFI) and the Auger Engineering Radio Array (AERA) was studied in the electromagnetic section of air showers. In Radio stations, it can view that the radio signals produced by coherent emissions because of geomagnetic radiation and charge excess processes. AERA can be observed in the frequency range of 30 to 80 MHz. This band is highly polluted by human made RFI. The new method has been introduced with a Cyclone® VSE chip on embedded micro-controller, which have operating frequency of 925 MHz and significantly minimize the operating time of the FIR filter coefficients. But it requires more area to design this work, which causes the more critical path [10].

To resolve these limitations, LC-CSLA-FIR method is implemented in FPGA to reduce the hardware utilization and minimize the cost. The normal adders and other kinds of adders has long critical path, this consumes much power as well as delay in its operation. The LC-CSLA-FIR implementations minimizes the critical path length. This reduces the time required for the operation of the adder unit in FPGA implementation. In ASIC implementation, area, power, delay, APP, and ADP is improved and through FPGA implementation, LUT, slices, flip flop are improved in LC-CSLA-FIR method than conventional method.

This paper is composed as follows. In Section 2, explained the background with objective, motivation, and problem identification. In section 3, described some previous related work. In Section 4, shows LC-CSLA-FIR filter design architecture. In Section 5, mentioned experimental setup and results and discussion. The conclusion is made in Section 6.

2. Related work

Y.C. Tsao, and K. Choi [11] introduced the novel parallel FIR filter architecture for the signal operations, where the applications find less complexity. Two parallel structures consumed much area and power. To compensate this, new efficient parallel FFA was implemented with symmetric coefficients for pre-processing and post processing of the input data. Here the number of adders was not minimized up to the best constraints.

Deepshikha Bhat [12] introduced a new minimum power FIR filter design for wireless sensor networks on FPGA. This work used Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS) technology for implementation of FIR filters. LVCMOS technology has various standards for several wireless sensor networks and few were discussed in this work. This system consumes more hardware and cost of the system is high.

S.Y. Park, and P.K. Meher [13] proposed the new efficient ASIC implementation of DA based reconfigurable FIR digital filters. Here the DA unit effectively used the same registers to store the intermediate results of the computation. The comparison on several structures has been consider like complexity and amount of hardware. This paper achieved higher throughput, but consumes many areas to implement hardware.

Subhankar Bhattacharjee [14] introduced the power efficient FIR filter implementation for DSP applications based on FPGA with the support of Xilinx 6V1X130T1FF1156. Several forms of the structures were analyzed and observed that the pipeline FIR filter structure take a number of registers and indirectly it consumes more resources and power. So it is fit only for high speed DSP application.

Marian Pristach [15] developed an enhanced design for FIR filters with the support of block memories. The new architecture has RAM for storing the data and one MAC unit for the multiply and accumulation purpose. This design performs one by one calculation to reach the minimum requirement of hardware. It is very much helpful in the ASIC and FPGA implementation and also it minimizes area, power, but it needs a high operating frequency.

This all related works contains several problems like more area, power consumption is high, high critical path, more hardware utilization, and FPGA utilization is more. To overcome these problems, LC-CSLA-FIR method is introduced to improve the ASIC implementation results and FPGA implementation results.

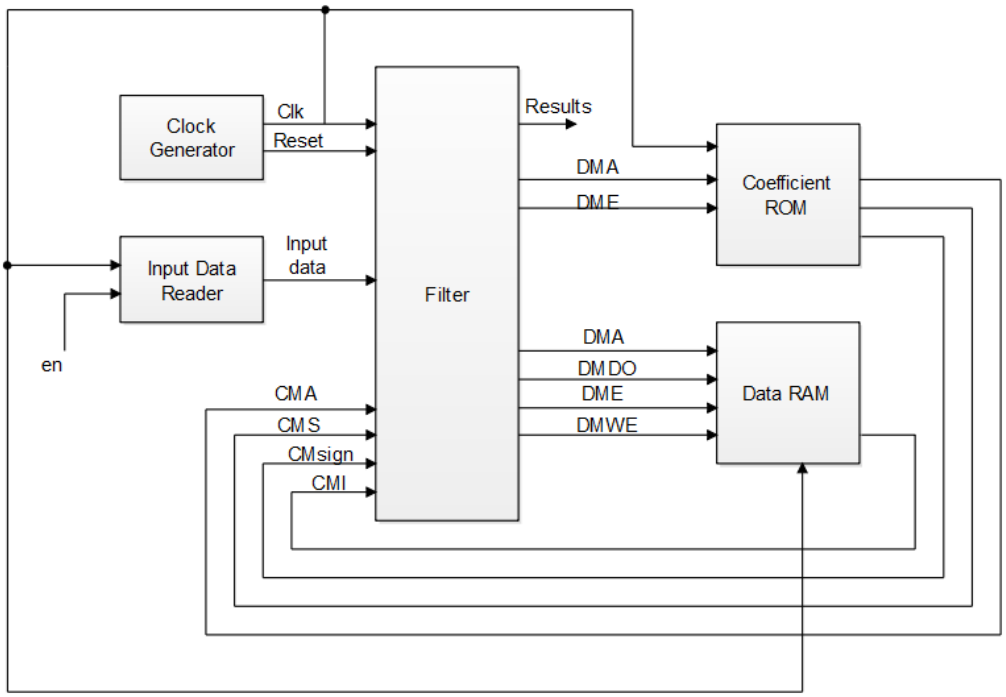

Figure.1 Traditional architecture of FIR filter

3. LC-CSLA-FIR method

The existing architecture of FIR filter is shown in Fig.1.That block consists of the coefficient ROM, data RAM, input data reader, clock generator, and a filter. The clock signal is generated from the clock generator. The coefficient values store in the coefficient ROM. The input data reader provides the input data value. The input data is moved to the data RAM for storing purposes. To perform the filtering operation, coefficient ROM, data RAM, CMA (coefficient memory address), CMS (coefficient memory shift), CMsign (coefficient memory sign) and CMI (coefficient memory input) are require. In CMA, the coefficient memory is stored to perform the filtering. With the help of CMS, right shifting (Division by 2^{-n}) and left shifting (Multiplying 2^{-1}) operation is executed. CMsign is used to decide addition or subtraction.

In the existing method, the co-efficient value stored in the ROM, which occupy more area. Also, the normal adder is used to perform the addition operation. To solve this problem, in the LC-CSLA-FIR method, the shifting information only stored in ROM and low area carries select adder is used to reduce the area, power, and delay.

The overall architecture of new FIR filter is presented in Fig.2. The block consists of the Read Only Memory (ROM), data Random Access Memory (RAM), Control Unit (CU), Address Generator (AG), Accumulator (AC), Clock Generator (CG), and a filter. The clock signal is generated by the clock generator. The shifting information can be stored in ROM and input data values is stored in RAM. From CU, the clock signal provides to the filter for computing filter output and reset signal is used to reset the registers in the filter block. The AG will generate the address which is used to read the data from ROM to perform the computation between filter coefficient and input data. The output of filter results is stored in the AC. To perform the filtering operation, N number of dividing portions are required that holds the shifting count. Once the data reader module provides the input data, the following operation is carried out. In the first stage, calculate the memory address and enable data RAM to store the input data. The second stage enables the coefficient ROM to perform the read coefficient one by one to get the filter results. Finally, enable data RAM according to the information. The shifter and CSLA adder helps to minimize the hardware cost. Implementation of FIR filters get different forms such as direct form-I, direct form-II, parallel implementation, polyphase, cascaded form, lattice and ladder structures, etc. To design efficient FIR filter, the designer has to consider some parameters such as area, power, cost and delay in the system.

The general form of the N-tap FIR filter is defined in Eq. (1).

$$Y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \qquad (1)$$
$$Here, n = 0,1,2,3,\ldots.\infty$$

Where, Y(n) is the output of FIR filter, h(k) is the co-efficient of FIR filter, X(n-k) denoted as the number of the input sequence. M multipliers and M-1 adders are required to implement M tap FIR filter. If we use poly phase decomposition for N parallel FIR filter, this will need N2 FIR sub filters of size M/N. It means that MN multipliers and N (M-1) adders are required in their implementation. As, this research work's aim is to minimize the power consumption and hardware cost, an N-parallel FIR filter will be increasing the hardware cost, which is not the best option for implementing FIR filters in wireless sensor network. In poly phase decomposition, it is not taking the processing speed of the system. Normally, the input value should be convoluted with a co-efficient value. For that, we need N number of the multipliers and an N-1 adders. That occupied more area to compute the results. LC-CSLA-FIR method is used to minimize the area, power, and delay in further. To minimize the cost of the hardware, the very important idea behind the LC-CSLA-FIR algorithm is to replace the multipliers and adders with the shifters and adders. Throughout the implementation of FIR filter, only adders and shifters are needed without using any multipliers.

In LC-CSLA-FIR method, shift-add-accumulate algorithm is used to minimize the area, power, and delay. To perform the division and multiplication logical right and left shifters are an efficient one. For consideration, signed or unsigned binary number is shifted left by n bits means, we can multiply the input value with 2^n. In the right shifting method, the input value is divided by 2n. So for all multiplication and division operation able to use the logic shifters.

For example, if the coefficient value is h, that value can be divided into some portion. In that divided portion, computation of the shifting information is stored in ROM. minimize the number of the adders present in FIR filter architecture, the registers feedback with an adder. Initially, Register holds the value "0". At a first clock cycle, the input address provides to the input of ROM and shifting operation is done with coefficients, whose ROM output is linked with the filter. Now, the filter output is given to the input of the accumulator. At the second clock cycle, the Register contains the first clock cycle filter output. That first clock cycle output is also given to the input of the adder, which helps to generate the filter output that also stored in the Register. At final clock cycle, the output which is stored in the Register is the filter output. If less number of adders and multipliers are used in the architecture, the overall cost and hardware utilization will be minimized. So, overall power consumption and area is minimized in the LC-CSLA-FIR method with the support of the logic shifter and adder.

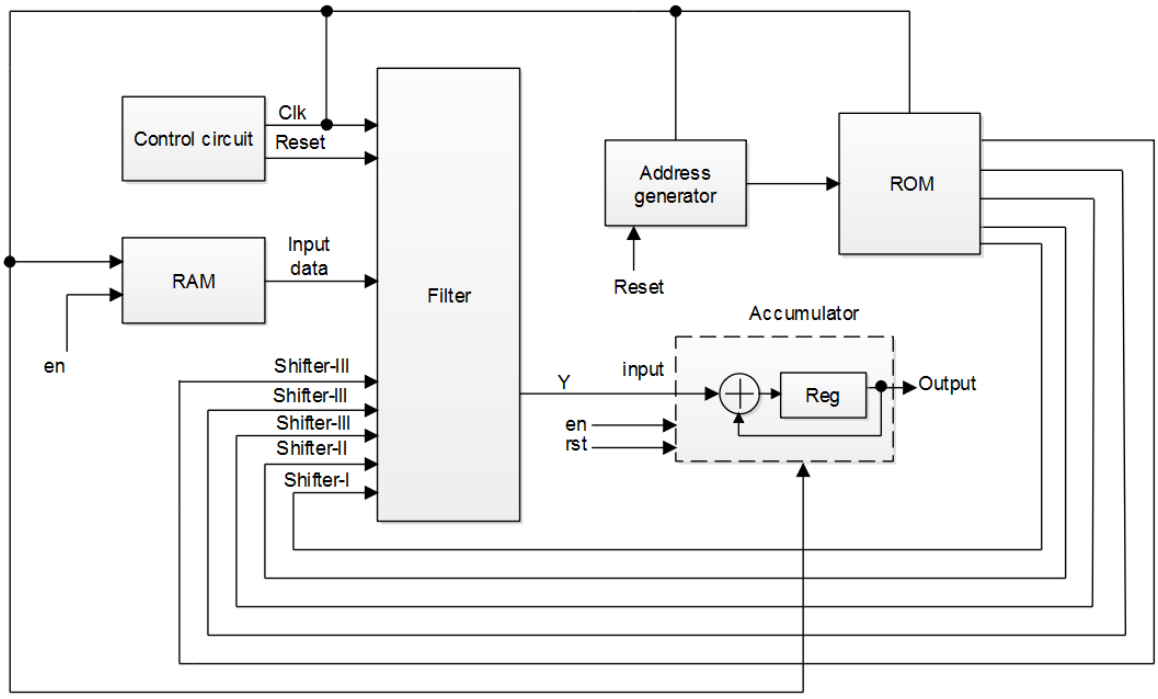

Figure.2 Overall architecture of new FIR filter

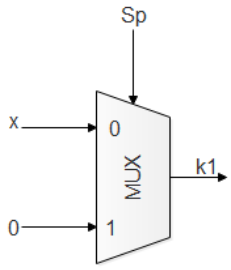

Figure.3 Circuit diagram of Shift-I

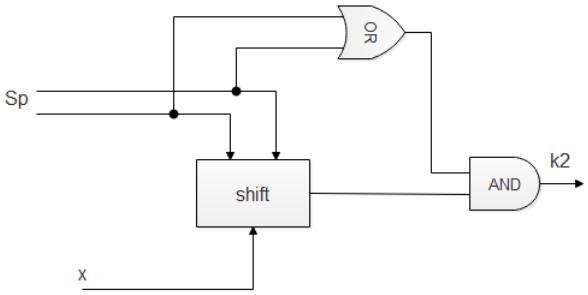

Figure.4 Circuit diagram of Shift-II

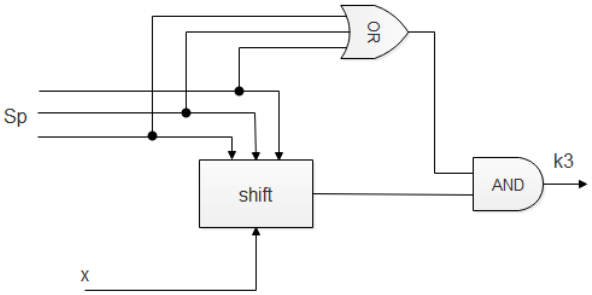

Figure.5 Circuit diagram of Shift-III

The shift- I circuit diagram is shown in Fig.3. The overall block mainly depends on the MUX performance. The shifting parameter (SP) is given to the selection line of the MUX. If the selected line is mentioned as 1 the output of the K becomes 0, otherwise based on the selection line the output of the K is delivered in the input signal, which is given to the MUX 0 level position.

Fig. 4 is a circuit diagram of the Shift-II, which consists of one OR gate, one AND gate, and shifter. The SP has the size of two bits, depending on the value of the SP. The RAM data is shifted and processed support the help of PE. The RAM data X is processed based on the parameter SP as it is presented in the Fig. 4. Fig.5 is the circuit diagram of the Shift-III. In this SP has the size of three bits which give the information regarding the number of shift operations to perform on the RAM data as it is presented in Fig. 5. To get the K value, SP is an important parameter.

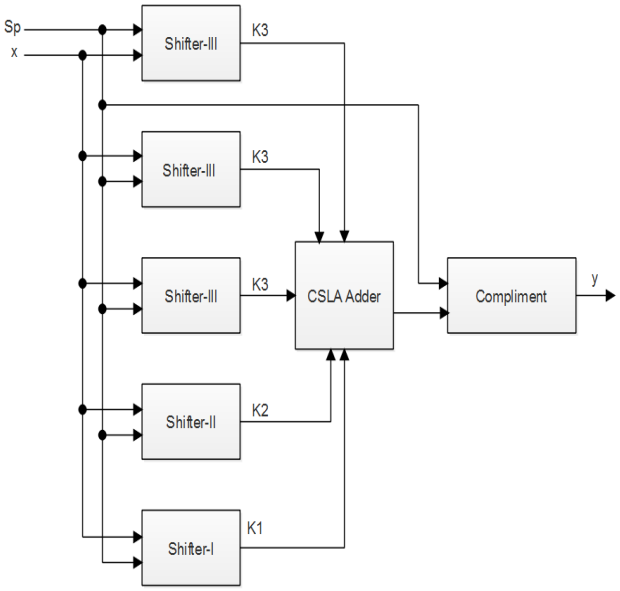

Figure.6 Block diagram of processing element

Processing Element (PE) is one of the major blocks in FIR filter design. Based on the PE, remaining block operates at good performance level. The complete circuit diagram of the PE is shown in Fig. 6. It consists of one shift-I, one shift-II, and three shift-III and adder unit. Here, the shifter is used instead of multipliers. Shifter-III represents for 3 bit, Shifter-II mentioned as 2 bit and Shifter-I denoted as bit 1. This shifting information is stored in the ROM. The input X is in the RAM, for each iteration taking one sample for processing. After processing of each stage samples are added with the support of area as well as power efficient CSLA adder instead of using normal adder. The results of the adder unit is signed as well as unsigned, to get the natural values of the signed values to take a compliment of the results as it is presented in Fig. 6. In the LC-CSLA-FIR method, co-efficient hasn't store into the ROM, only shifting information only stored in the ROM. So the number of ROM also reduced in the LC-CSLA-FIR method than the existing method.

To enhance adder performance instead of using normal adder, CSLA adder is used, which is given in Fig.7. This adder can achieve fast arithmetic operation in various data processing techniques. This adder is mainly used for reducing area and power consumption in the implementation. CSLA is manipulated in many computational structures to cut the carry propagation delay. The elementary knowledge of this work is to habit BEC (binary to excess-1 convertor) instead of RCA (ripple carry adder) with Cin=1. By using fewer numbers of logic gates, BEC logic is derived instead of using n-bit FA (Full Adder).

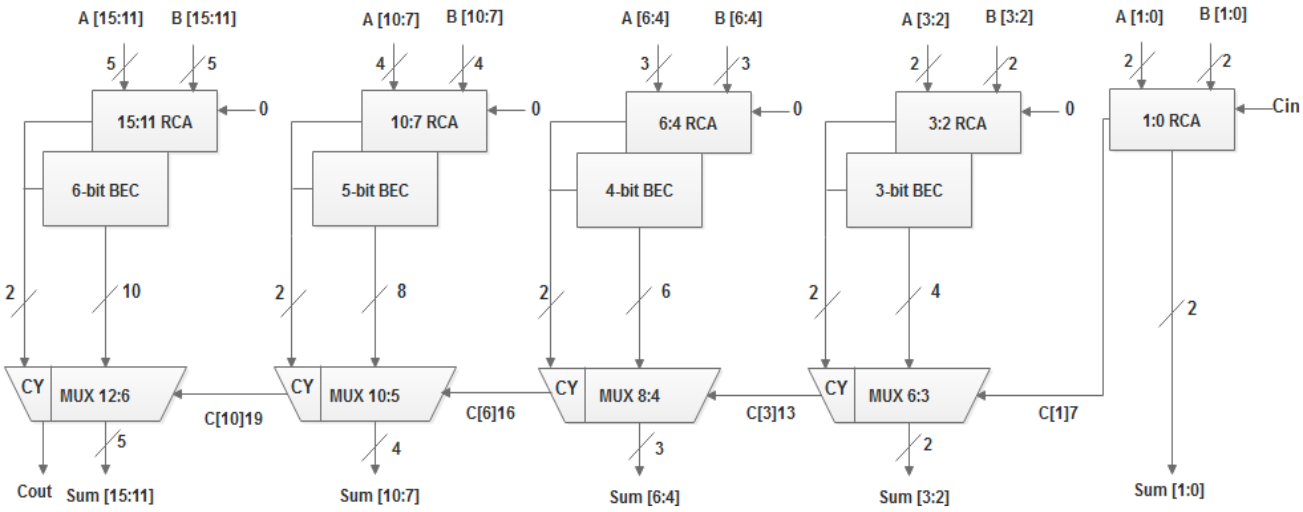

Figure.7 Low area carry select adder

The assembly of the LC-CSLA-FIR system using BEC for normal adder or RCA with Cin =1 to increase the power and area. 2-bit RCA which has one full adder and one-half adder for Cin =1 where 3-bit BEC is utilized that enhances one by the output from 2-bit RCA. According to this consideration, the time delay is reduced. It is the process on feedback values, which are the output of the MUX is depended on the input of the MUX. The input arrival time is lesser than the multiplexer selection input arrival time. By selecting BEC output or the straight inputs, there are two possibilities is obtainable such as parallel and multiplexer rendering to the regulate signal Cin. While designing CSLA, the area, power, and delay will be reduced. The multiplexer delay and MUX selection arrival time derived from the different kind of groups. With the help of shifter and CSLA adder huge amount of the area, power, and delay will be minimized.

4. Result and discussion

The LC-CSLA-FIR method RTL timing diagram is verified in Modelsim 10.1c using Verilog code. FPGA performance was analysed for different devices such as Virtex-4, Virtex-5, and Virtex-6 by using Xilinx ISE tool. In LC-CSLA-FIR work, ASIC implementation of FIR filter algorithm is verified by using cadence tools in 180nm as well as 45nm. Multipliers are not required to design the FIR filter, which optimize the area, power, and delay of the overall FIR filter architecture.

4.1 ASIC synthesis

This ASIC synthesis is implemented in Cadence tool for different technology such as 180nm and 45nm. From this tool, the performance is calculated such as area, power, and delay.

4.1.1. Area

With shrinking system size ASIC was able to accommodate maximum functionality in minimum area. The designer will specify area constraint and Cadence tool is used to optimize the area performance. The area is optimized by lesser number of cells and by replacing multiple cells with a single cell that includes both functionalities.

4.1.2. Power

Development of hand-held devices has led to reduction of battery size and hence low power consuming systems. Low power consumption has become a big requirement for a lot of designers.

4.1.3. Delay

The designer specifies the maximum delay between primary input and primary output. This is taken as maximum delay across any critical path. The comparison of area, power, delay, APP, and ADP for different technologies such as 180nm and 45nm is given in table 1. This table presents a comparison of Existing – I, Existing–II, Existing-III, and LC-CSLA-FIR. These four methods are implemented by Verilog and the outputs are tabulated. In Existing-I [13], FIR filter has been designed without using any optimized adder, which occupy more area. In Existing –II [6], Efficient FIR filter was developed for different taps. But, this method does not concentrate on FPGA performance. In Existing – III [7], FIR filter has been designed for radio applications. According to this paper, ASIC results are not taken from this work. In

Table 1. The performance of area, power and delay the LC-CSLA-FIR method for 180nm and 45nm technology

Technology	Method	Length	Area (um^2)	Power (nW)	Delay (ps)	APP (um^2 * nW)	ADP (um^2 * ps)
180nm	Existing- I [13]	8- tap	80368	1890454	8274.1	151932007072	664964832
		16- tap	72541	2952824.6	8326.3	214200805784	603976366
		32-tap	124587	5124798.3	13547	638483245802.1	1687780089
	Existing -II [6]	8- tap	51461	1808977	8073.4	93091765397	415444653.4
		16- tap	67221	2852929.6	8137.3	191776780641.6	546997443.3
		32-tap	105490	4674727.1	11444	493136951230	1207227560
	Existing-III [14]	8- tap	48754	1795796	7845.2	89989938184	382475130
		16- tap	63457	2742687	7984.2	174042688959	506640688
		32-tap	98745	4257915.2	9842.6	420447836424	971848290
	LC-CSLA -FIR	8- tap	43446	1706552	7343.6	74142858192	319050045.6
		16- tap	57396	2617388	7467.9	150227601648	428627588.4
		32-tap	84663	3736800	7691.1	316368698400	651151599.3
45nm	Existing -I [13]	8- tap	5248	186239.8	3147	977382272	16515456
		16- tap	6984	298453.53	3420.4	2084395752	23885280
		32-tap	12473	461847.21	5264.4	5760617631	65657872
	Existing-II [6]	8- tap	4761	166514.8	2998.0	792773154	14273478
		16- tap	6708	264621.53	3050.4	1775077668	20459400
		32-tap	10994	440997.21	5097.4	4826273510.016	55781568.6
	Existing-III [14]	8- tap	4287.6	159845.5	2887.2	685255515	12367908
		16- tap	6212.2	256987.2	2863.9	1596454641.4	17785528.6
		32-tap	9546.2	391475.9	3847.2	3737098645	36724231.4
	LC-CSLA -FIR	8- tap	3699.96	151215.2	2726.4	234590538.81	10087570.944
		16- tap	5358.47	247354.5	2692.1	1325441667.615	14423736.087
		32-tap	8330.08	373019.6	2836.6	3107248270	23623880.928

LC-CSLA-FIR method, both FPGA and ASIC results are concentrated to reduce the hardware utilization. This table includes the different kind of length like 8 tap, 16 tap, and 32 tap. In existing method, a normal digital adder is used to perform the accumulation operation, which occupy more area. In LC-CSLA-FIR method, carry select adder is used in the accumulator, which require less space to operate the shifting and accumulation. Due to this CSLA adder, the area, power, delay, APP, and ADP is minimized in LC-CSLA-FIR architecture than conventional filter methods. If the transistor size reduced, it produces a better result. So, 45nm technology is most preferable to reduce the area, power, and delay. From this table, it's clears that LC-CSLA-FIR filter occupy less area, power, and delay than conventional methods. The comparison graphs of area, power, and delay are presented in Figs. 8, 9, and 10. That results are drawn by using 180nm technology as well as 45nm technology for different length like 8 tap, 16 tap, and 32 tap. According to that graph, blue line is represented as existing- I, orange line is mentioned as existing- II, grey line represents as existing-III, and yellow line mentioned as LC-CSLA-FIR method.

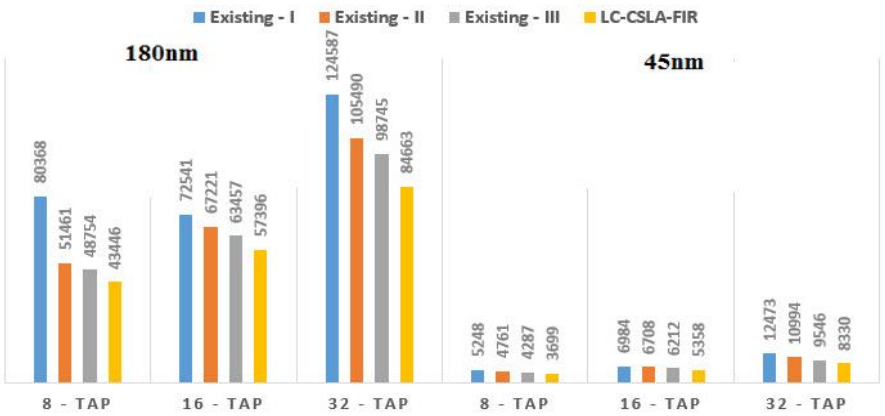

Figure.8 Area performance of different methods for 180nm and 45nm CMOS technology

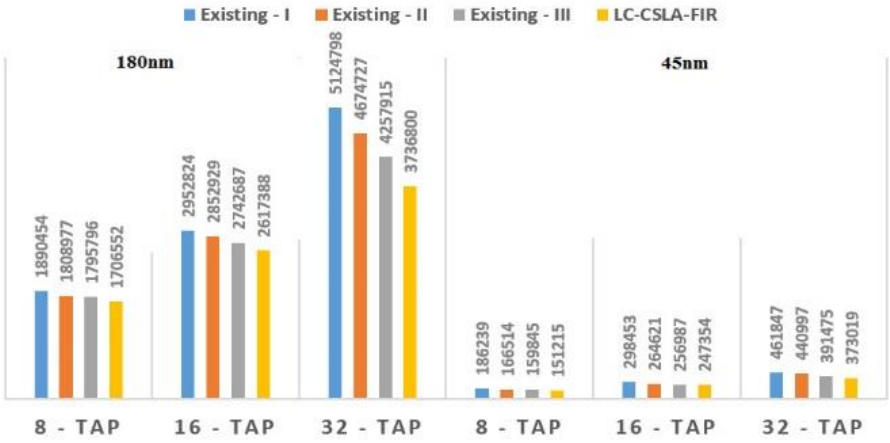

Figure. 9 Power performance of different methods for 45nm CMOS technology

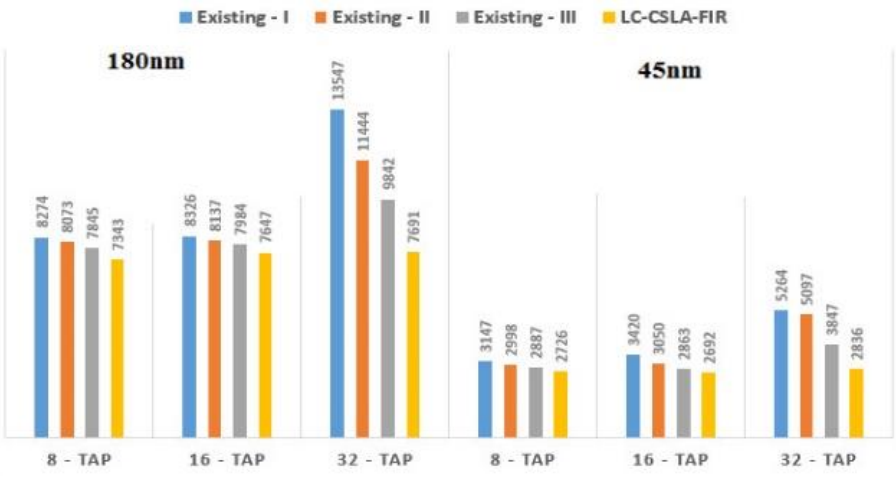

Figure.10 Delay performance of different methods for 180nm CMOS technology

Table 2. Reduced percentage of area, power, delay, APP, and ADP for LC-CSLA-FIR method

Technology	Window	Reduced % of Area	Reduced % of power	Reduced % of APP	Reduced % of ADP
180nm	8 tap	10.88	4.96	17.6	16.58
	16 tap	9.55	4.56	13.68	18.2
	32 tap	14.26	12.23	24.75	32.99
	Average	**11.56**	**7.25**	**18.67**	**22.59**
45nm	8 tap	13.71	5.39	65.76	18.43
	16 tap	13.74	3.74	16.97	18.9
	32 tap	12.73	4.71	16.85	35.67
	Average	**13.39**	**4.61**	**33.19**	**24.33**

From this graph, it is clear that LC-CSLA-FIR method consume less area, less power, less delay than conventional methods.

The reduction percentage of area, power, delay, APP, and ADP for different taps like 8 tap, 16 tap, and 32 tap is given in tab.2. This architecture result has been taken in both 180nm and 45nm technology. In 180nm technology, 11.56% of area, 7.25% of power, 18.67% of APP, and 22.59% of ADP is minimized in LC-CSLA-FIR as well as 45nm technology, 13.39% of area, 4.61% of power, 33.19% of APP, and 24.33% of ADP is reduced in LC-CSLA-FIR method than conventional method.

4.2 FPGA synthesis

This FPGA synthesis is implemented in Xilinx tool for different devices such as Virtex-4, Virtex-5, and Virtex-6. From this tool, the performance will be calculated such as LUT, flip flop, Slices, and Frequency.

4.2.1. LUT

A LUT, which stands for Look Up Table, in general terms is basically a table that determines what the output is for any given input(s). In the context of combinational logic, it is the truth table. This truth table effectively defines how your combinatorial logic behaves.

4.2.2. Flip flop

Flip-flops are binary shift registers used to synchronize logic and save logical states between clock cycles within an FPGA circuit. On every clock edge, a flip-flop latches the 1 or 0 (TRUE or FALSE) value on its input and holds that value constant until the next clock edge.

4.2.3. Slices

Logic resources are resources on the FPGA that perform logic functions. Logic resources are grouped in slices to create configurable logic blocks. A slice contains a set number of LUTs, flip-flops and multiplexers. A LUT is a collection of logic gates hard-wired on the FPGA.

4.2.4. Frequency

Frequency is defined as the rate at which something occurs over a particular period of time or in a given sample.

Table 3. FPGA performance of different devices for different design

Target FPGA	Circuit		LUT	Flip flop	Slice	RAM	ROM	Frequency
Virtex4 xc4vfx12	Existing-I [13]	8-tap	536/10944	48/10944	279/5472	1	2	51.25
		16-tap	526/10944	45/10944	278/5472	1	2	62.36
		32-tap	564/10944	46/10944	294/5472	1	2	42.21
	Existing- II [6]	8-tap	516/10944	42/10944	269/5472	1	2	83.858
		16-tap	518/10944	43/10944	268/5472	1	2	83.169
		32-tap	553/10944	42/10944	288/5472	1	2	78.539
	Existing-III [14]	8-tap	420/10944	40/10944	209/5472	1	2	57.21
		16-tap	494/10944	42/10944	248/5472	1	2	56.21
		32-tap	524/10944	41/10944	248/5472	1	2	63.54
	LC-CSLA-FIR	8-tap	365/10944	38/10944	196/5742	1	1	65.852
		16-tap	379/10944	39/10944	198/5742	1	1	61.267
		32-tap	445/10944	40/10944	230/5472	1	1	64.875
Virtex5 xc5vlx20T	Existing-I [13]	8-tap	286/12480	38/12480	125/3120	1	2	55.98
		16-tap	288/12480	42/12480	105/3120	1	2	54.39
		32-tap	381/12480	44/12480	189/3120	1	2	75.28
	Existing- II [6]	8-tap	269/12480	36/12480	111/3120	1	2	100.441
		16-tap	274/12480	39/12480	99/3120	1	2	88.790
		32-tap	373/12480	41/12480	162/3120	1	2	104.980
	Existing-III [14]	8-tap	215/12480	35/12480	88/3120	1	2	98.56
		16-tap	222/12480	38/12480	78/3120	1	2	92.45
		32-tap	305/12480	40/12480	104/3120	1	2	88.33
	LC-CSLA-FIR	8-tap	168/12480	34/12480	45/3120	1	1	91.654
		16-tap	197/12480	36/12480	54/3120	1	1	92.336
		32-tap	181/12480	38/12480	49/3120	1	1	88.975
Vitex6 xc6vcx75t	Existing-I [13]	8-tap	345/46560	54/93120	125/11640	1	2	107.56
		16-tap	398/46560	44/93120	135/11640	1	2	121.2
		32-tap	446/46560	52/93120	152/11640	1	2	115.65
	Existing- II [6]	8-tap	325/46560	50/93120	117/11640	1	2	127.677
		16-tap	369/46560	42/93120	127/11640	1	2	115.873
		32-tap	417/46560	48/93120	141/11640	1	2	117.233
	Existing-III [14]	8-tap	305/46560	44/93120	98/11640	1	2	115.6
		16-tap	248/46560	39/93120	78/11640	1	2	117.23
		32-tap	266/46560	42/93120	89/11640	1	2	110.2
	LC-CSLA-FIR	8-tap	184/46560	34/93120	53/11640	1	1	115.391
		16-tap	186/46560	36/93120	56/11640	1	1	112.626
		32-tap	186/46560	39/93120	56/11640	1	1	112.687

The table 3 presents the comparison of the parameters such as LUTs, the number of flip flops, slices, RAM, ROM and operating frequency for different FPGA devices such as vertex 4, vertex 5 and vertex 6. From this table, it is concluded that LUT, flip flop, slices, and ROM is reduced in LC-CSLA-FIR method compared to the existing method. Due to the reduction of those parameters, the area is optimized in FIR filter. The operating frequency is also analysed for different devices.

FPGA performance of LUT, flip flops, and slices are presented in Figs. 11, 12, and 13. This results have been taken for different devices such as Virtex- 4, virtex-5, and virtex-6. From this graph, it's clears that all the FPGA performance is improved in LC-CSLA-RFIR design than conventional design.

The FIR filter waveform result is shown in Fig.14, which is taken from Modelsim software. Data_out value is randomly generated from RAM, which is multiplied with co-efficient. In LC-CSLA-FIR method, shifter is used to perform FIR operation instead of using multiplier. In this diagram, s1 to s5 are denoted as shifting information, which are denoted as yellow color. For example, data_out value is mentioned as 220. Now, the shifting operation is performed with the help of those 5 shifting information. In this particular clock cycle, s2 value is 3, rest of the values become zero. So 2^3= 8. Then, add 1 value with 8, when data_out is equal to K1 (rose color). Now, performed the FIR operation 220 x 9 =1980, which stored in y. Acc contains 0 value that is added to y. That added result also stored in the next cycle Acc. Finally, FIR filter output is delivered from Acc.

The RTL schematic of FIR filter is shown in Fig. 15. This schematic obtained from a Synplify Pro by using Verilog code. There was a separate code for each block such as ROM, addr_gen, RAM, Decoder, PE and CSLA adder. Normally the input value performs shifting operation with co-efficient, which produces the output in Acc.

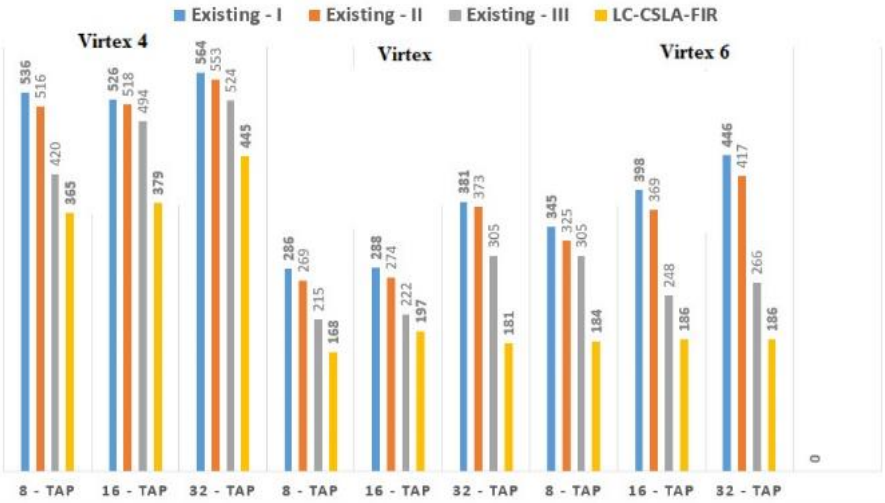

Figure.11 LUT performance of different methods for different Virtex devices

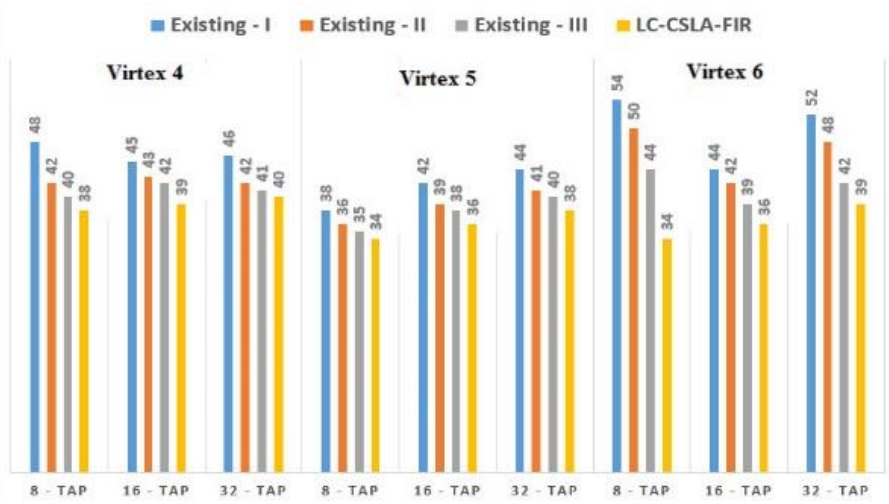

Figure.12 Flip flop performance of different methods for different Virtex devices

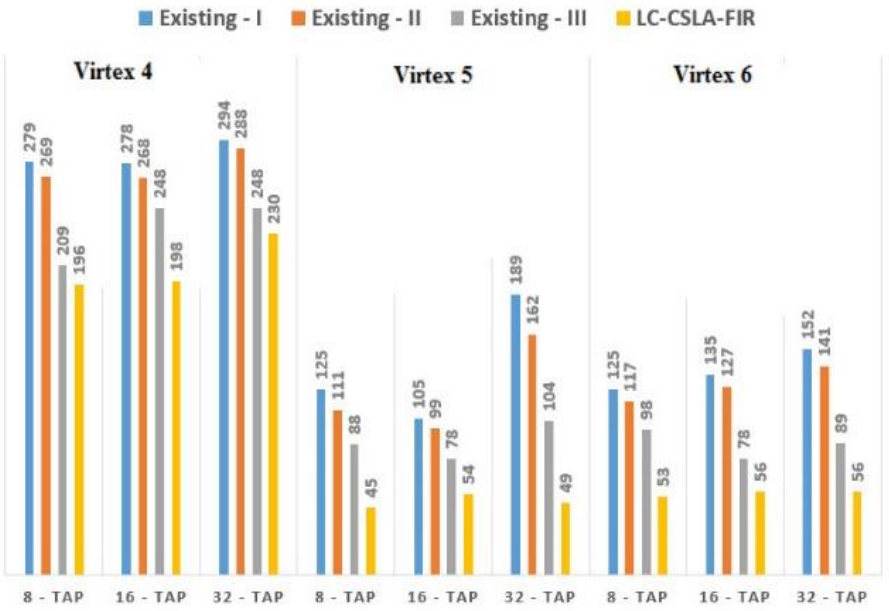

Figure.13 slices performance of different methods for different Virtex devices

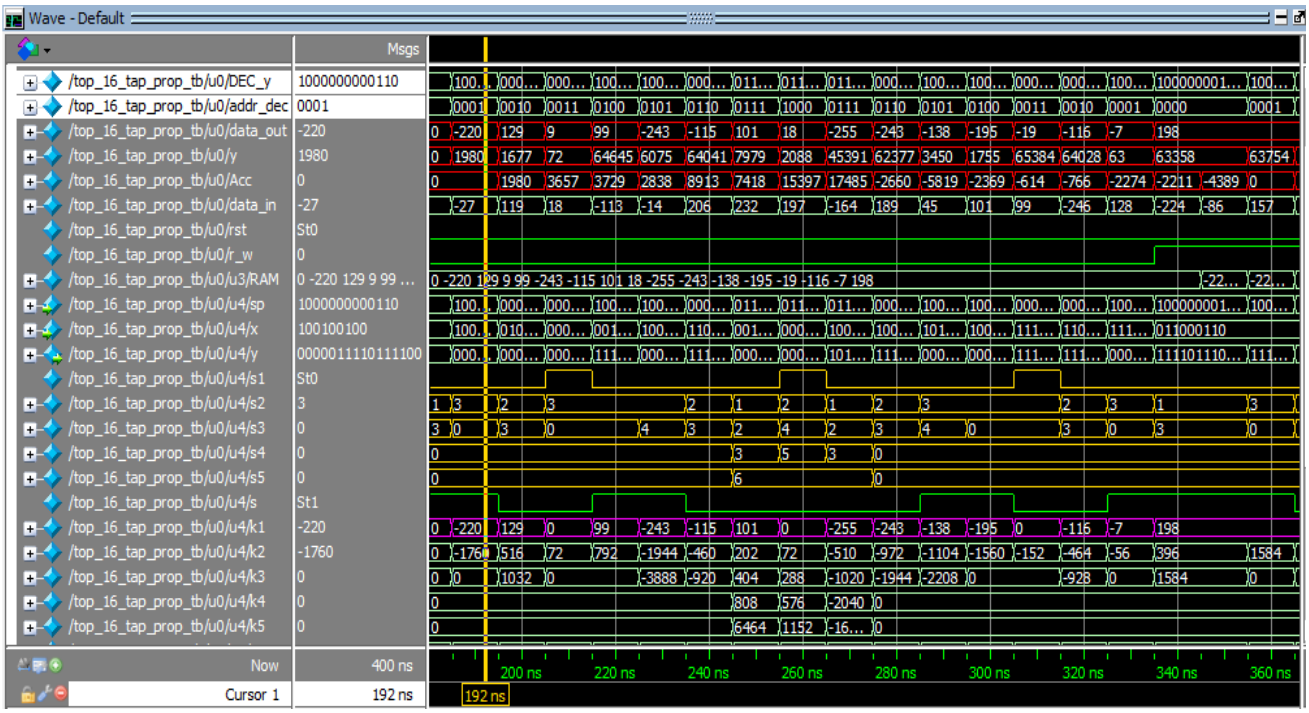

Figure.14 FIR filter result

The RTL schematic of LC-CSLA-FIR design for 8 tap is shown in Fig. 16, which is taken from Cadence tool. For ASIC implementation, same code has been used that is used for the FPGA implementation. The Cadence RTL compiler is used to convert RTL Verilog into Gate level Verilog. Verilog codes are read by using a tcl file and corresponding libraries also set into the tcl file. After synthesizing, Area, Power and Delay result is displayed in cadence tool. The overall cadence output of LC-CSLA-FIR method is shown in Fig.17. From cadence tool, the results have been taken, which is shown as a screenshot for verification purpose. From this screenshot, it's clears that total area, total delay, total power, APP and ADP are reduced in LC-CSLA-FIR method than conventional methods.

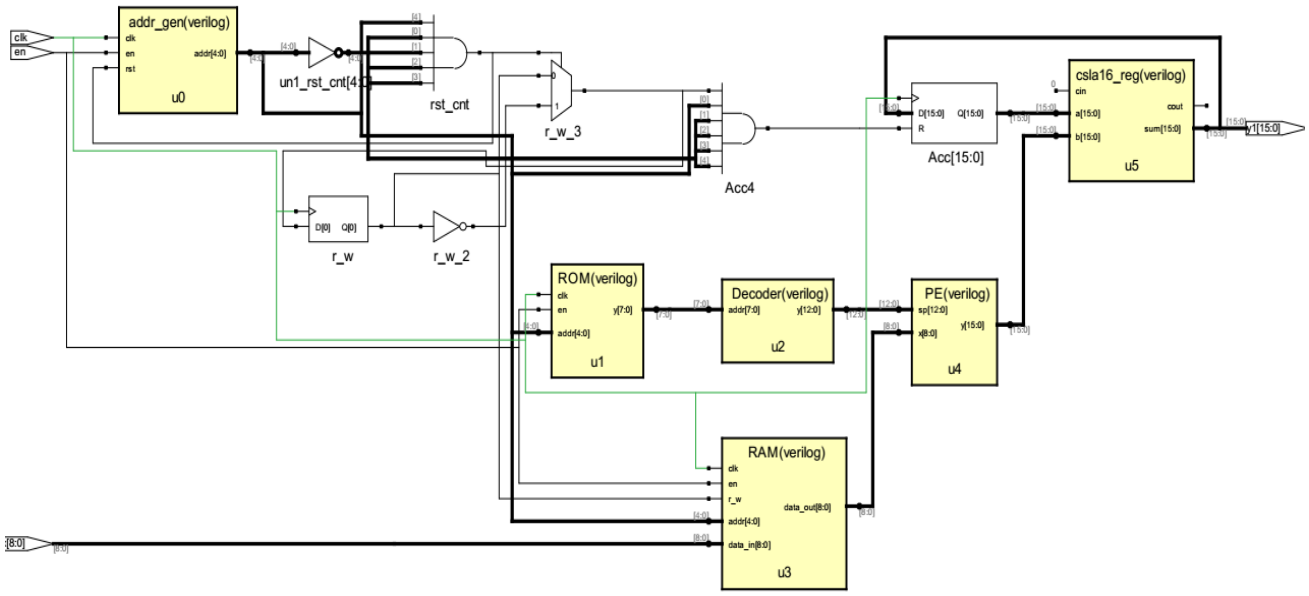

Figure.15 RTL schematic of FIR filter

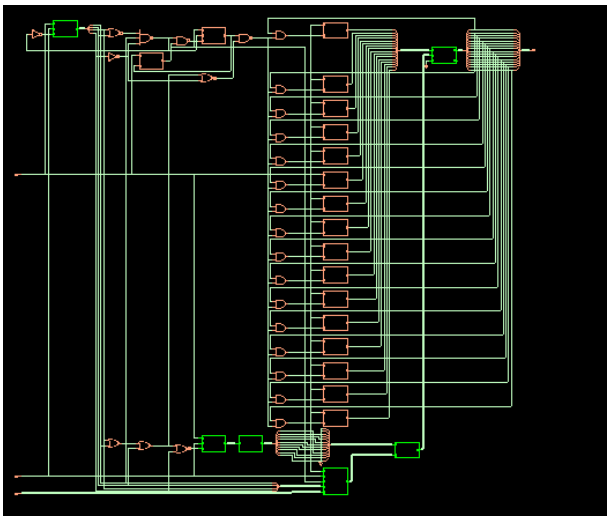

Figure.16 RTL schematic of FIR filter for 180nm technology

Total Area	Switching (nW)	Arrival (ps)
43446.00	1706552.63	7019.60
800.00	37936.22	7019.60
1066.00	29674.28	7078.00
441.00	12503.39	
16059.00	750200.50	
18966.00	562018.37	7078.00
1479.00	54801.76	7157.10
304.00	3387.87	7157.10
838.00	41156.00	7343.60
1641.00	66885.04	
1641.00	34812.40	

Figure.17 Area, power and delay analysis for LC-CSLA-FIR for 8 tap in 180nm

5. Conclusion

In this paper, LC-CSLA-FIR architecture has been implemented in Modelsim software by using Verilog code. In this method, FIR filter has been implemented by using control unit, RAM, co-efficient RAM, and accumulator circuit. Without using any multiplier, the FIR architecture has been designed, which consume less access time and less area. Area, power, and delay have been evaluated for different taps like 8-tap, 16-tap, and 32-tap. In the FPGA implementation, LUT, slices, and flip flops is improved in LC-CSLA-FIR. In ASIC 180nm technology, 11.56% of area, 7.25% of power, 18.67% of APP, and 22.59% of ADP is minimized in LC-CSLA-FIR as well as 45nm technology, 13.39% of area, 4.61% of power, 33.19% of APP, and 24.33% of ADP is reduced in LC-CSLA-FIR method compared to conventional method. In future, this FIR filter design will perform architecture level optimization to further reduce the hardware utilization such as LUT, slices, and flip flop.

References

[1] S.S. Mirzaei, A. Hosangadi, and R. Kastner, “FPGA implementation of high speed FIR filters using add and shift method”, In: *Proc. of International Conf. on Computer Design*, pp.308-313, 2007.

[2] J.I. Acha, “Computational structures for fast implementation of L-path and L-block digital filters”, *IEEE Transactions on Circuits and Systems*, Vol.36, No.6, pp.805-812, 1989.

[3] S. Lin and S.K. Mitra, “Overlapped block digital filtering”, *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, Vol.43, No.8, pp.586-596, 1996.

[4] D. Shi and Y.J. Yu, "Design of linear phase FIR filters with high probability of achieving minimum number of adders", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.58, No.1, pp.126-136, 2011.
[5] C. Cheng and K.K. Parhi, "Hardware efficient fast parallel FIR filter structures based on iterated short convolution", In: *Proc. of International Conf. on Circuits and Systems*, Vol.3, p. III-361-4, 2004.
[6] C. Xu, S. Yin, Y. Qin, and H. Zou, "A novel hardware efficient FIR filter for wireless sensor networks," In: *Proc. of International Conf. on Ubiquitous and Future Networks*, Vol.14, pp.197-201, 2016.
[7] G. Haridasa, and D.S. Georgeb, "Area Efficient Low Power Modified Booth Multiplier for FIR Filter", *Procedia Technology*, Vol.24, pp.1163-1169, 2016.
[8] S.C. Prasanna and S.P.J.V. Rani, "An efficient implementation of reconfigurable interpolation root-raised cosine FIR filter for software-defined radio application", *Sādhanā*, Vol.41, No.10, pp. 1189-1201, 2016.
[9] Z. Szadkowski, D. Głas, C. Timmermans, and T. Wijnen, "First Results from the FPGA/NIOS Adaptive FIR Filter Using Linear Prediction Implemented in the Auger Engineering Radio Array", *IEEE Transactions on Nuclear Science*, Vol.62, No.3, pp.977-984, 2015.
[10] Z. Szadkowski and D. Głas, "Adaptive Linear Predictor FIR Filter Based on the Cyclone V FPGA With HPS to Reduce Narrow Band RFI in Radio Detection of Cosmic Rays", *IEEE Transactions on Nuclear Science*, Vol.63, No.3, pp.1455-1462, 2016.
[11] Y.C. Tsao and K. Choi, "Area-efficient parallel FIR digital filter structures for symmetric convolutions based on fast FIR algorithm", *IEEE Transactions on Very Large Scale Integration Systems*, Vol.20, No.2, pp.366-371, 2012.
[12] D. Bhat, A. Kaur, and S. Singh, "Wireless sensor network specific low power FIR filter design and implementation on FPGA", In: *Proc. of the 2nd International Conf. on Computing for Sustainable Global Development*, 2015.
[13] S.Y. Park, and P.K. Meher, "Efficient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.61, No.7, pp.511-515, 2014.
[14] S. Bhattacharjee, S. Sil, and A. Chakrabarti, "Evaluation of Power Efficient FIR Filter for FPGA based DSP Applications", *Procedia Technology*, Vol.10, pp.856-865, 2013.
[15] M. Pristach, V. Dvorak, and L. Fujcik, "Enhanced Architecture of FIR Filters Using Block Memories", *IFAC-PapersOnLine*, Vol.48, No.4, pp.306-311, 2015.