



## A Low Power VLSI Implementation of Reconfigurable FIR Filter Using Carry Bypass Adder

**Kasarla Satish Reddy<sup>1\*</sup>    Hosahally Narayangowda Suresh<sup>1</sup>**

<sup>1</sup>*Bangalore Institute of Technology, India*

\* Corresponding author's Email: [ksathishphd2017@gmail.com](mailto:ksathishphd2017@gmail.com)

**Abstract:** Reconfigurable Finite Impulse Response (RFIR) filter plays an important role in Software Defined Radio (SDR) systems, whose filter coefficients change dynamically during runtime. In this paper, Low Cost Carry Bypass adder Reconfigurable Finite Impulse Response (LC-CBA-RFIR) is introduced to perform the RFIR filter operations. DRAM-based Reconfigurable Partial Product Generators (DRPPG) consists of MUX and dual port distributed RAM, which has co-efficient to perform a FIR filter operation. With the help of Verilog code, the RFIR filter architecture was verified in Modelsim software. The same Verilog code was used to analyse the ASIC performances such as area, power and delay Area Power Product (APP), Area Delay Product (ADP) as well as FPGA performances such as LUT, flip flop, slice and frequency. After implementing ASIC and FPGA, all the performance improved in LC-CBA-RFIR method compared to the conventional methods.

**Keywords:** Reconfigurable finite impulse response, Software defined radio, Carry bypass adder, DRAM-based reconfigurable partial product generator, ModelSim.

### 1. Introduction

Finite Impulse Response (FIR) filter plays an important role in several signal processing applications in communication schemes, which performs interference cancellation, channel equalization, spectral shaping, matched filtering and more. Nowadays, various implementation and architecture methods have been presented to improve the performance of filters in terms of system complexity and speed [1]. The FIR filter used in the majority of the Digital Signal Processing (DSP) applications is based on electronic systems [2]. The FIR filter coefficients change rapidly during execution time, in several application scenarios such as a digital up-converter, digital down converter, multi-channel filter and software-defined radio systems [3, 4]. Compared to the conventional non-RFIR filter designs with reconfiguration / without reconfiguration, RFIR filters were consuming less resource and power [5]. The discrete FIR filtering detects extensive utility in low-power and high-

performance Embedded Computing Systems (ECS) that range from wireless transmitters/ receivers to image and video processing units [6].

Present day research work mainly focused on the design of fully Integrated Circuits (IC) that used for wireless applications and employed in the most advanced fabrication methodologies. The main problems in portable telecommunication equipment are long battery life as well as weight, reduced cost and size, which needs low-power and small area integrated devices [7]. The adaptive filter significantly used in the DSP applications. The tap-delay line FIR filters whose weights updated by the "Widrow-Haff Least Mean Square" (WH-LSM) algorithm used as an adaptive filter not only because of its simplicity and but also due to its satisfactory convergence performance. So, the adaptive filter significantly employed in DSP applications [8]. The shared LUT design for DA based Reconfigurable FIR (RFIR) Digital Filter minimized the hardware cost by decomposing the RAM [9]. 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. [10].

In multi-media applications and mobile communications, RFIR filters are required because of their main advantage like low-cost, less area, low-power and high speed operation [11, 12]. The pipelined modified booth multiplier is used for RFIR filter architecture. This architecture has changed the order of the filter to reach significant savings in power consumption than existing architectures but this strategy is not possible for the low-power applications [13]. Low power 8-bit based RFIR filter with minimum power consumption system improved efficiency but it used only for 8-bit data [14]. Digital RFIR filter method consists of low power serial multiplier and serial adder, shift/adder, shift/multiplier combinational booth multiplier, folding transformation in linear phase architecture [15]. The normal adder has a long critical path and consumes more power. Also, hardware utilization and the execution time is more in previous works. In this work, the CBA is used instead of normal adder. Due to the CBA adder, the RFIR architecture achieves better performance in terms of less area, power, and delay. In ASIC the area, power and delay minimized by implementing in cadence encounter tool with 180nm and 45nm library technology. In FPGA implementation, the number of LUTs, slice and flip-flop decreased in CBA-RFIR for different kinds of Virtex devices such as Virtex 4, Virtex 5 and Virtex 6.

This paper is composed as follows. In section 2, described some previous related work. In Section 3, shows LC-CBA-RFIR design architecture. In Section 4, mentioned experimental setup and results and discussion. The conclusion is made in Section 5.

## 2. Related work

S. Y. Park, and P.K. Meher [16] illustrated a novel pipelined architecture for a lower power, high-throughput and low area adaptive filter based on DA. The throughput rate of the FIR design was maximized by the parallel LUT update and concurrent implementation of the filtering and weight operation. Reduction of the power consumption was improved by using a fast Bit-clock used for Carry-Save Accumulation (CSA) but it has a much slower clock speed for all the other operations.

S. Ramanathan, G. Anand, P. Reddy, and S.A. Sridevi [17] have presented a low-power adaptive

FIR filter based on DA with high-throughput, low-power and area. The Least Mean Square (LMS) algorithm is employed to update the weight and reduce the Mean Square Error (MSE) between the current filter outcome and the desired response. The pipelined DA table decreases switching activity and decreased power. The main limitation of this paper is that it is significantly focused on power consumption.

N. Sriram, and J. Selvakumar [18] used Pipelined Modified Booth Multiplier (PMBM) method used for implemented low power RFIR filter architecture. But limitation of this method is that delay value is high because of decreased system speed and throughput.

K.M. Basant, P.K.Meher, S.K. Singhal, M.N.S. Swamy [19] introduced the high-performance VLSI architecture for RFIR using DA. Here the author has analyzed the two kinds of structures and conclude that the direct form structure needs less number of registers compared to the transpose form structure. Reconfigurable block-based FIR filter with DA provides the scalability for higher block sizes and larger filter lengths. But the limitation of this method only discussed the structures for a block size of 4.

R. Jia, H.G. Yang, C.Y. Lin, R. Chen, X.G. Wang, and Z.H. Guo [20] introduced novel RFIR filter design based on statistics centric reconfigurable (SCR) FIR filter architecture. The experimental results were analyzed by considering performance parameters such as area, speed, and power for the high-order FIR filters and they have concluded that proposed RFIR filters have improvements in their performance over the conventional FIR filter but did not discuss about the dynamically reconfigurable mechanism.

For existing work, they have used normal full adder, pipeline adder, and parallel adder etc. That adder based techniques occupied more area, more power, high critical path, and more hardware utilization in FPGA implementation. To overcome these problems, LC-CBA-RFIR method is introduced for evaluating ASIC implementation results and FPGA implementation results.

## 3. LC-CBA- RFIR methodology

The CBA-RFIR technique consists of Serial-in-Parallel-out Shift Register (SIPOSR), Distributed Read Access Memory (DRAM) based Reconfigurable Partial Product Generator (RPPG), Pipeline Adder Tree (PAT) shifter and pipeline shift add three elements on the FPGA implementation. The CBA-RFIR technique based on RFIR filter structure has been implemented in FPGA with CBA.

### 3.1 The DA based LC-CBA-RFIR filter used for the FPG implementation

The FPGA methodology is developed from a dedicated hardware to a heterogeneous system, which is why it's a popular choice in the communication base stations instead of being a prototype platform. The reconfigurable DA-based on FIR filter is used for FPGA implementation by employing the CBA-RFIR technique. The LUT's are developed by using the DRAM with FPGA implementation. The multiple numbers of the partial inner-products  $S_{l,p}$  are retrieved from the DRAM simultaneously, so only one LUT value is read from DRAM, per cycle. Furthermore, if "L" is the bit width of the input, the duration of the sample period of the design is L times the operating clock period. That is not suitable for the application requiring high-throughput. DRAM is employed to develop LUT for each bit slice because of its high-resource consumption. Hence, decompose the partial inner-product generator into  $Q$  parallel sections and every section has  $R$  time-multiplexed

operations corresponding to  $R$  bit slices. When  $L$  is a Composite number provided by  $L = RQ$  (Here  $R$  and  $Q$  are two positive integers), index  $l$  in Eq. (1) is mapped with  $r + qp$  for  $r = 0,1,2, \dots, Q - 1$  to modify in Eq. (2) as

$$y = \sum_{l=0}^{L-1} 2^{-1} (\sum_{p=0}^{P-1} S_{l,p}) \tag{1}$$

$$S_{l,p} = \sum_{m=0}^{M-1} h(m+pM) [S(M+pM)] \tag{2}$$

Here,  $l = 0,1,2, \dots, L - 1$  and  $p = 0,1,2, \dots, P - 1$  since the sum of partial product is  $S_{l,p}$  of the  $M$  samples.

$$\sum_{q=1}^{Q-1} 2^{-RP} [\sum_{r=0}^{R-1} 2^{-1} (\sum_{p=0}^{P-1} r + q, R, P)] \tag{3}$$

In Eq. (3),  $q$  represents as index and  $r$  represents time index. The structure of the CBA-RFIR time-multiplexed DA based FIR filter by employing DRAM is shown in the Fig.1.

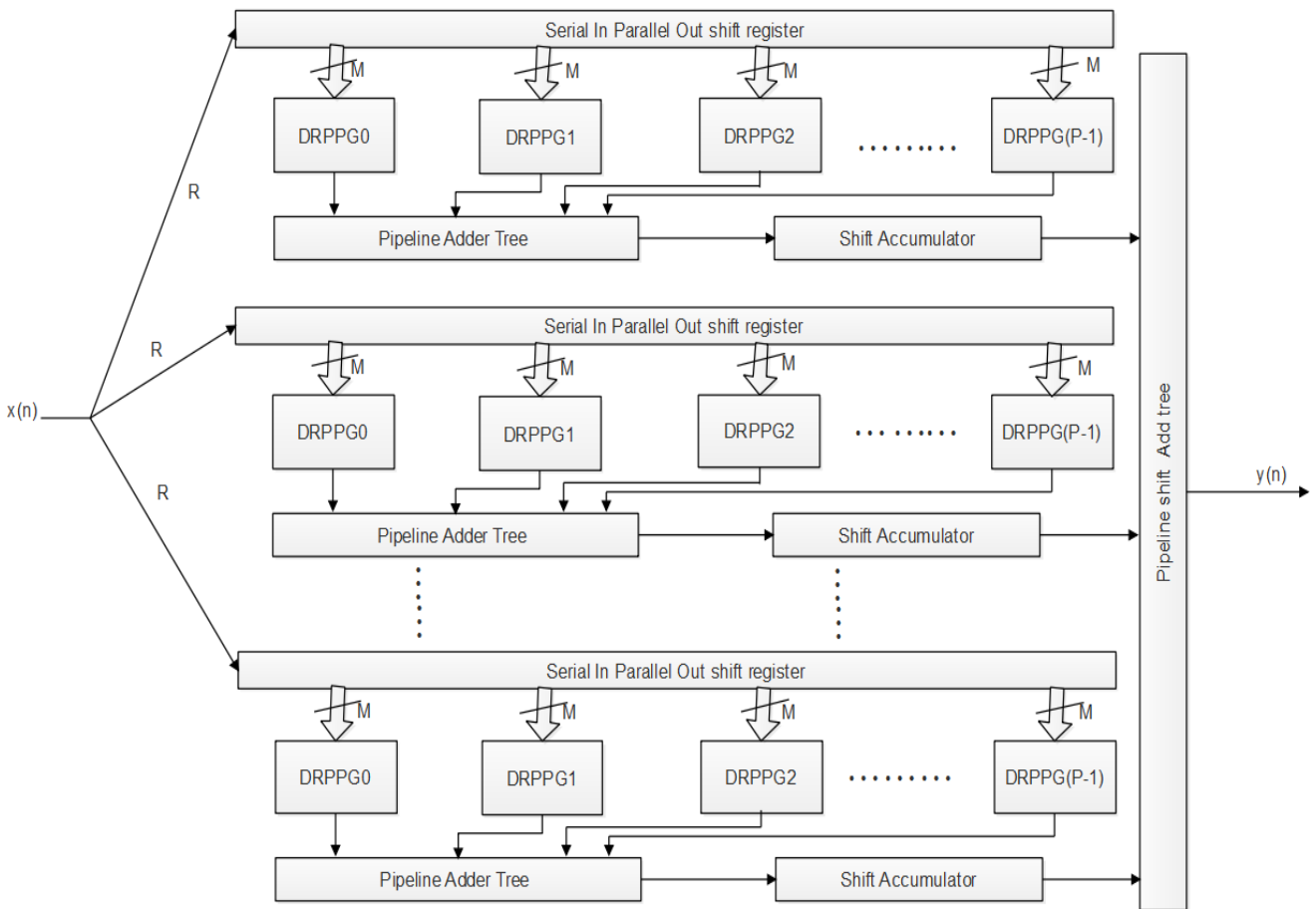


Figure.1 The CBA-RFIR time-multiplexed DA based FIR filter by employing DRAM.

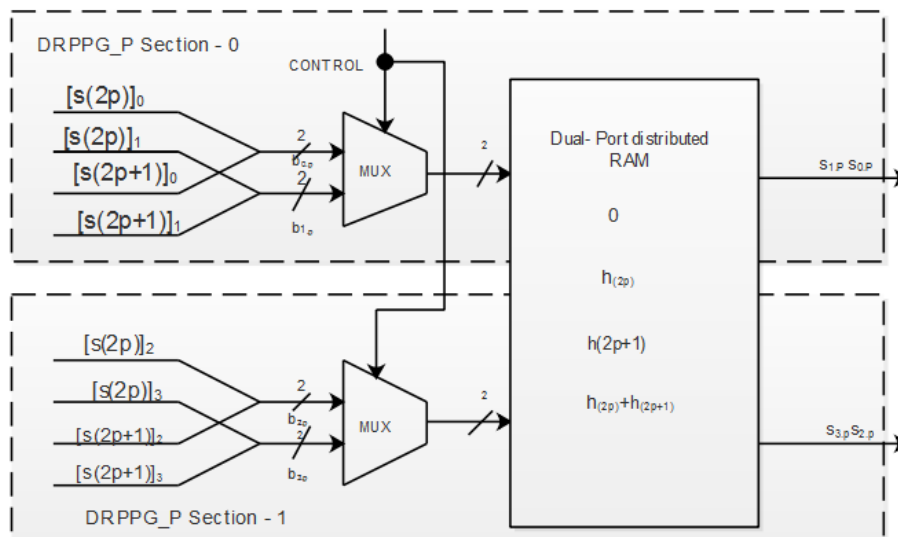


Figure.2 The structure of DRAM base DRPPG

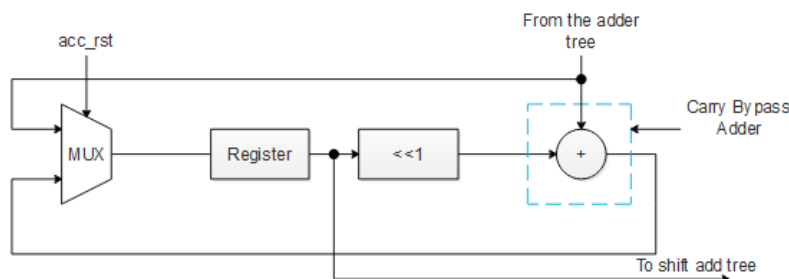


Figure.3 The structure of shift accumulator

To develop Eq. (3), the CBA-RFIR structure has  $Q$  section and every section consists of  $P$  DRAM based Reconfigurable Partial Product Generators (DRPPG) and the PAT to compute the rightmost result followed by Shift Accumulator (SA) which performs over  $R$  cycles according to the second summation. However, it employs dual-port DRAM to decrease the total size of the LUTs by half than that of a 2-DRPPGs in which two different sections shares a single DRAM. In the Fig.2 shows the structure of DRAM based on DRPPG. In  $r^{th}$  cycle is  $P$  DRPPG in  $q^{th}$  section operate  $P -$  partial inner product  $S_{r+q,R,p}$  used for  $p = 0,1,2, \dots, P - 1$  to add by using the Pipeline Adder Tree (PAT). The outcomes of the PATs are accumulated through AS on the  $R$  cycle shown in the Fig. 3. The accumulated rate is reset at each  $R$  cycle by control signal to keep the accumulator register ready to be utilized for calculation of the next filter output. The  $f_{clk}$  maximum operating clock period and the CBA-RFIR techniques up ports the input sample range by  $f_{clk}/R$ .

### 3.1.1. Carry bypass adder

In the CBA, Ripple Carry Adder (RCA) is employed to add four-bits at a time and the carry generated will be propagated to next stage with the help of multiplexer utilizing selected input as Bypass logic. Bypass logic is formed from the production values as it is computed in the CLA. Depending on the carrying value and bypass logic, the carry is propagated to the next stage. The CBA is an adder implementation, which improves the delay of an RCA. The 4-bit CBA design requires 4-FA circuits. The input buses would be a 4-bit A and 4-bit B with a carry-in ( $C_{in}$ ) signal. The output would be a 4-bit bus X and Carry-out ( $C_{out}$ ) signal. The first 2-FAs would add the first 2-bits together. The  $C_{out}$  signal from the second-FA ( $C_1$ ) would drive the selected signal for three 2:1 multiplexers. Fig. 4 shows the block diagram of the carry bypass adder. The 4-bit RCA is shown in Fig. 5.

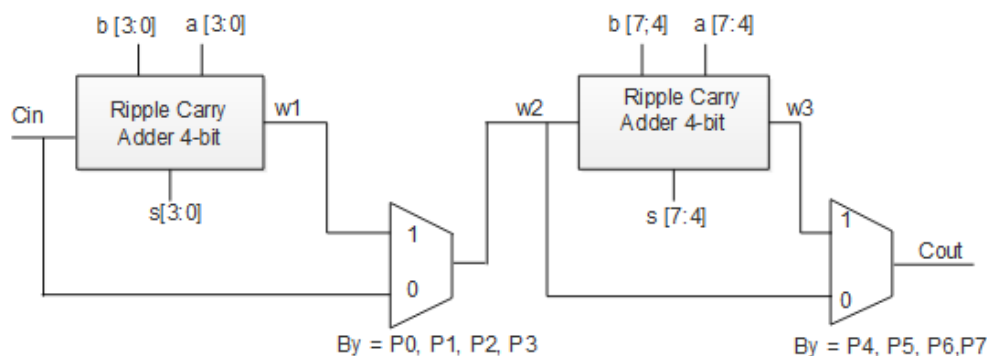


Figure.4 Block diagram of the carry bypass adder

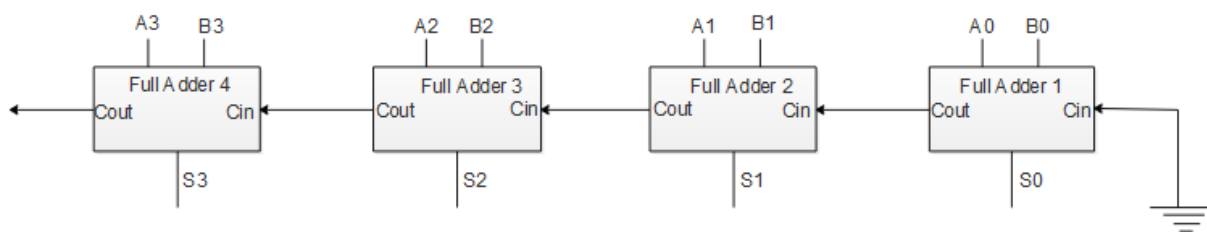


Figure.5 4-bit ripple carry adder

Multiple FA are cascaded in parallel to add an N-bit. For N-bit parallel adder, there are N-number of FA circuits. An RCA is a logic circuit, in which  $C_{out}$  of each FA is  $C_{in}$  of the succeeding next significant FA that is known as RCA due to each carry bit gets rippled into the next stage. In RCA the sum and carry-out bits of any half adder stage are not valid until  $C_{in}$  of that stage occurs. Propagation delay is the time elapsed between the application of input and occurrence of the corresponding output.

For example, for a NOT gate, when the input is “zero” the output will be “one”. The time taken for the NOT gate’s output to become “zero” after the application of logic “one” to NOT gate’s input is the propagation delay. Similarly the carry propagation delay is the time elapsed between the application of the carry in signal and the occurrence of the  $C_{out}$  signal. Sum out  $S_0$  and  $C_{out}$  count of the FA “one” is valid only after the delay of the 1-bit FA. In the same way, sum out  $S_3$  of the 4-bit FA is valid only after the joint propagation delays of 1-bit FA to 4-bit FA. The final outcome of the RCA is valid only after the joint propagation delay of the FA circuit inside it.

#### 4. Result and discussion

The LC-CBA-RFIR design timing diagram was verified in Modelsim 10.1c using Verilog code. RTL schematic was taken from Synplify pro tool. FPGA performance was analyzed for different devices of Virtex-4, Virtex-5, and Virtex-6 by using Xilinx ISE tool. In LC-CBA-RFIR work, ASIC implementation

of RFIR filter algorithm was verified by using Cadence tools in 180nm as well as 45nm technology.

#### 4.1 ASIC synthesis

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

##### 4.1.1. Area

With shrinking system size, ASIC should be able to accommodate maximum functionality in minimum area. The designer specifies area constraint and cadence tool is used to optimize the area performance. The area is optimized by having 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 a reduction of battery size and hence providing low power consuming systems. Low power consumption has become a basic requirement for a lot of designers.

##### 4.1.3. Delay

The designer specifies the maximum delay between primary input and a primary output. This is taken as maximum delay across any critical path.

Table 1. The performance of area, power and delay the proposed method for 180nm and 45nm technology

Technology	Method	Bits & Taps	Area (um <sup>2</sup> )	Power (nW)	Delay (ps)	APP (um <sup>2</sup> * nW )	ADP (um <sup>2</sup> * ps )
180nm	Existing -I [7]	8 B & 3T	282242	2649825.18	292	747907806600	82414664
		8 B & 7T	325521	8858665.45	298	2883681489465	97005258
	Existing-II [13]	8B & 3T	265478	2458978.13	285	652804561484	75661230
		8B & 7T	306852	8654255.56	295	2655575455260	90521340
	Existing [8]	8B & 3T	256478	2418971.12	279	679402784138	71557362
		8B & 7T	296221	8545246.69	287	2651558764566	85015427
LC-CBA-RFIR	8 B & 3T	<b>234789</b>	<b>2241023.1</b>	<b>271</b>	<b>526167549147</b>	<b>63627819</b>	
	8 B & 7T	<b>271420</b>	<b>8345131.5</b>	<b>280</b>	<b>2265035456020</b>	<b>75997600</b>	
45nm	Existing [7]	8 B & 3T	15182	99848.24	199	1515892336	3021218
		8 B & 7T	18546	59895.26	220	1110812670	4080120
	Existing [13]	8B & 3T	13343	96466.32	197	1287145838	2628571
		8B & 7T	16598	56206.95	217	932907188	3601766
	Existing [8]	8B & 3T	12403	98400.23	198	1220455200	2455794
		8B & 7T	15489	55402.65	215	858121578	3330135
	LC-CBA-RFIR	8 B & 3T	<b>10428</b>	<b>89452.43</b>	<b>189</b>	<b>932805456</b>	<b>1970892</b>
		8 B & 7T	<b>12605</b>	<b>48648.88</b>	<b>205</b>	<b>613208040</b>	<b>2584025</b>

The comparison of the area, power, delay, APP, and ADP for different technologies such as 180nm and 45nm presented in Table 1. Additionally, this table presents a comparison of Existing-I, Existing-II, Existing-III and LC-CBA-RFIR. These four methods developed using Verilog and the output are tabulated. In existing [7], the RFIR filter was implemented for various taps. But this technique does not focus on FPGA. In existing [13], FIR designed by using full adder that occupy large area. In existing method [8], the normal digital adder performed the accumulation operation, which occupies more area. In the LC-CBA-RFIR method, carry select adder was used in the accumulator, which required less space to operate the shifting and accumulation. Due to this CBA adder,

the area, power, delay, APP, and ADP have minimized in LC-CBA-RFIR architecture than conventional RFIR filter.

The comparison graph of the area, power, area power product, and area-delay product is shown in Figs.6, 7, 8, and 9. These results are drawn by using 180nm technology for different kind of bits and tabs such as 8 B & 3T, and 8 B & 7T. According to that graph, the blue line represents existing and the orange line represents as an LC-CBA-RFIR based method. From this graph, it is clear that LC-CBA-RFIR method consumes less area, less power, less area power product and less area-delay product than conventional methods.

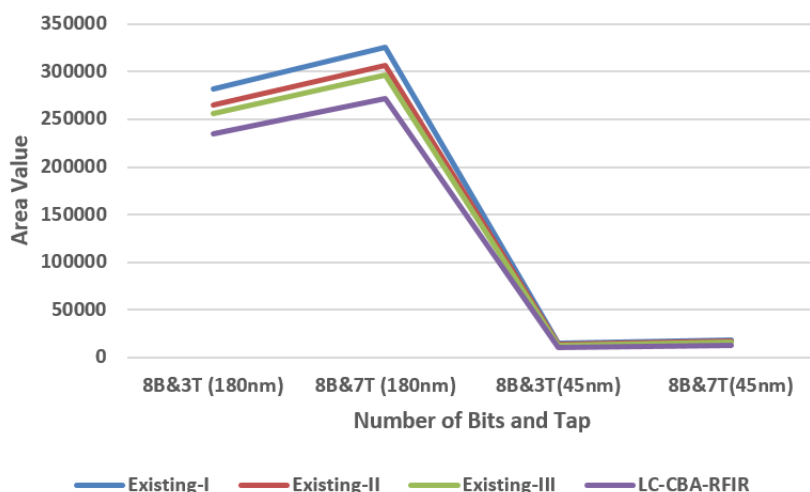


Figure.6 Area performance of different bits and tabs for 180nm and 45nm technology

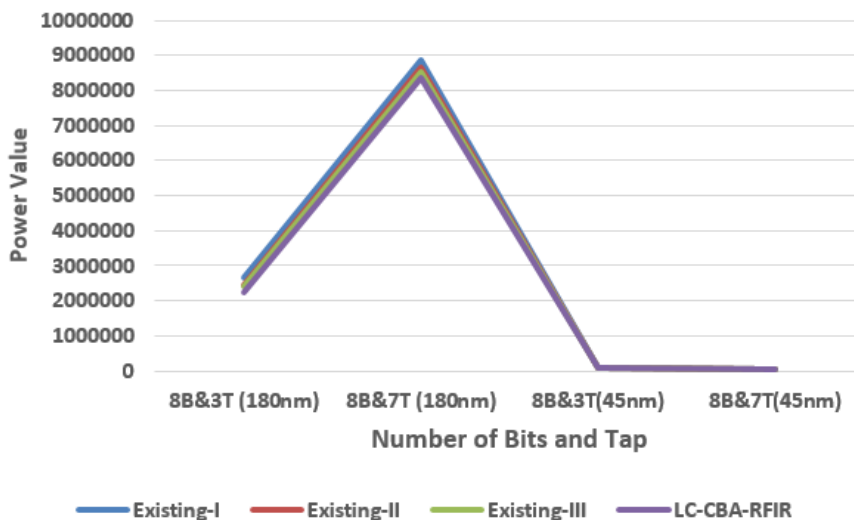


Figure.7 Power performance of different bits and tabs for 180nm and 45nm technology

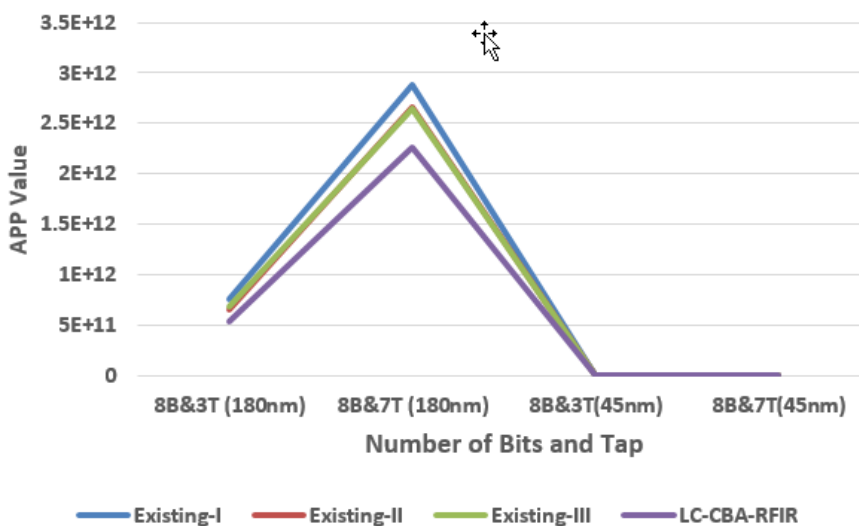


Figure.8 APP performance of different bits and tabs for 180nm and 45nm technology

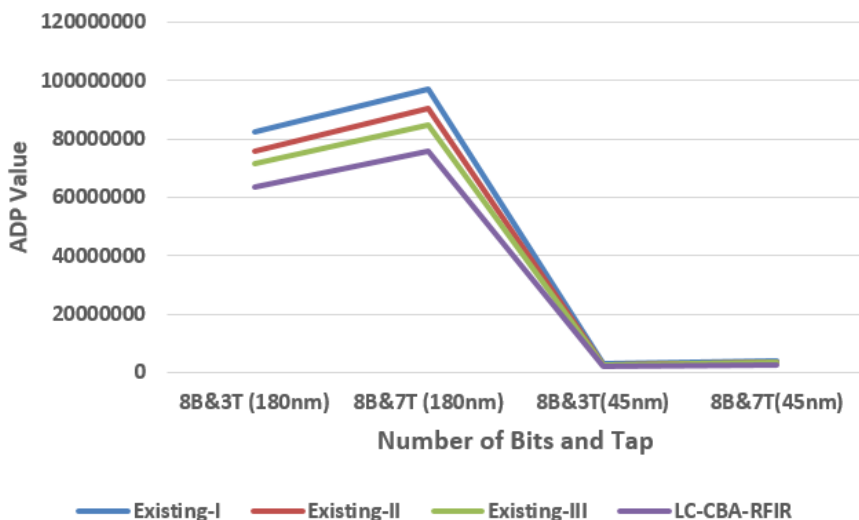


Figure.9 ADP performance of different bits and tabs for 180nm and 45nm technology

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

Technology	Window	Reduced % of Area	Reduced % of power	Reduced % of APP	Reduced % of ADP
180nm	8 B & 3T	8.45	15.40	22.52	11.08
	8 B & 7T	8.43	6.70	14.54	10.66
	<b>Average</b>	<b>8.44</b>	<b>11.05</b>	<b>18.53</b>	<b>10.87</b>
45nm	8 B & 3T	15.92	9.09	29.68	19.74
	8 B & 7T	18.61	12.19	28.54	22.40
	<b>Average</b>	<b>17.26</b>	<b>10.64</b>	<b>29.11</b>	<b>21.07</b>

The reduction percentage of area, power, delay, APP, and ADP for different bits and taps like 8 B & 3T, and 8 B & 7T are given in Tab 2. This architecture result has been taken in both 180nm and 45nm technology. In 180nm technology, 8.44 % of area, 11.05% of power, 18.53% of APP, and 10.87% of ADP is minimized in LC-CBA-RFIR as well as 45nm technology, 17.26% of area, 10.64% of power, 29.11% of APP, and 21.07% of ADP is reduced in LC-CBA-RFIR method when compared to the 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 parameter like LUT, flip-flop, Slices, and Frequency has been calculated.

##### 4.2.1. LUT

A LUT, which stands for LUT, in general terms it is basically a table that determines what is the output for any given input(s). In the context of combinational logic, it is the truth table. This truth table effectively defines how combinational 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 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 is the comparison of the 8-bit input sample to analyze performance parameters such as LUTs, the number of flip-flops, slices, and operating frequency for different FPGA devices such as vertex 4, vertex 5 and vertex 6. This result has been taken for different bits and taps like 8 B & 3T, and 8 B & 7T. From this table, it is concluded that the LUT, flip-flop, slices reduced and operating frequency is increased in LC-CBA-RFIR method than the existing RFID method. Due to the reduction of those parameters, the area has been minimized in filter architecture.

FPGA performance of Virtex-4 devices for 8-bit different taps is shown in Fig.10. In that graph, LUT, Flip-flop, slices, and Frequency have been analyzed of the Virtex-4 device for different bits and tabs like 8 B & 3T, and 8 B & 7T. From this graph, it clears that all the FPGA performance is improved in LC-CBA-RFIR design than conventional design.

The RTL schematic of FIR filter is shown in Fig.11, which is taken from Synplify pro software using Verilog code. This architecture is having a separate code for each block such as a counter, reg\_bank, GRPPG, and an accumulator. Input is stored in a registered bank in the form of bitwise. That register bank input is performed DRPPG operation, which contains MUX and DRAM. In DRAM, the coefficient value is stored to perform FIR filter operation. DRPPG output is performed on the accumulate operation then it will give RFIR filter output in “y”.



Table 3. Implemented on different Xilinx FPGA devices for various tap of 8 bit FIR filter

8- bit input						
Target FPGA	Circuit		LUT	Flip-flop	Slice	Frequency (MHz)
Virtex4 xc4vfx12	Existing [4]	8B & 3T	161/10944	110/10944	115/5472	85.213
		8B & 7T	210/10944	155/10944	190/5472	93.562
	Existing [12]	8B & 3T	150/10944	102/10944	103/5472	98.253
		8B & 7T	195/1094	142/10944	175/5472	105.124
	Existing [8]	8B & 3T	142/10944	98/10944	94/5472	108.312
		8B & 7T	189/10944	136/10944	162/5492	114.902
LC-CBA- RFIR	<b>8B &amp; 3T</b>	<b>115/10944</b>	<b>87/10944</b>	<b>78/5472</b>	<b>110.012</b>	
	<b>8B &amp; 7T</b>	<b>156/10944</b>	<b>114/10944</b>	<b>140/5472</b>	<b>117.410</b>	
Virtex5 xc5vlx20 T	Existing [4]	8B & 3T	180/12480	105/12480	112/3120	120.442
		8B & 7T	198/12480	122/12480	98/3120	67.223
	Existing [12]	8B & 3T	172/12480	95/12480	98/3120	133.342
		8B & 7T	183/12480	102/12480	85/3120	75.862
	Existing [8]	8B & 3T	160/12480	75/12480	89/3120	141.417
		8B & 7T	178/12480	98/12480	76/3120	89.432
LC-CBA- RFIR	8B & 3T	<b>115/12480</b>	<b>59/12480</b>	<b>81/3120</b>	<b>145.624</b>	
	8B & 7T	<b>146/12480</b>	<b>84/12480</b>	<b>72/3120</b>	<b>94.145</b>	
Virtex6 xc6vcx75 t	Existing [4]	8B & 3T	194/46560	115/93120	99/11640	98.152
		8B & 7T	170/46560	82/93120	88/11640	38.245
	Existing [12]	8B & 3T	186/46560	108/93120	82/11640	110.124
		8B & 7T	162/46560	98/93120	72/11640	40.235
	Existing [8]	8B & 3T	165/46560	96/93120	79/11640	136.912
		8B & 7T	149/46560	68/93120	66/11640	48.645
LC-CBA- RFIR	<b>8B &amp; 3T</b>	<b>124/46560</b>	<b>88/93120</b>	<b>64/11640</b>	<b>141.267</b>	
	<b>8B &amp; 7T</b>	<b>126/46560</b>	<b>56/93120</b>	<b>52/11640</b>	<b>54.486</b>	

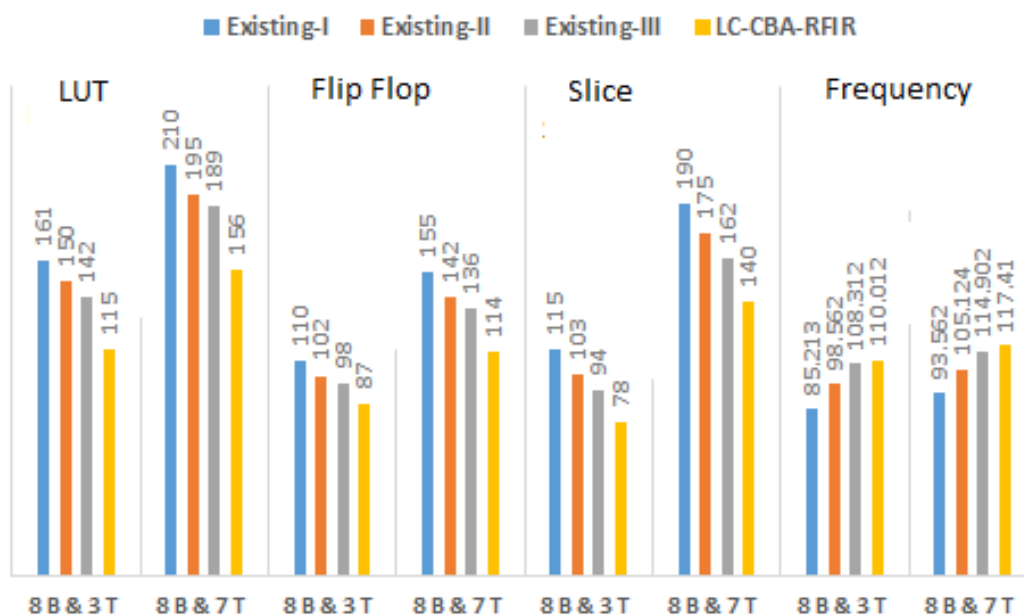


Figure.10 FPGA performance of Virtex-4 device for 8B & 3T and 8B & 7T

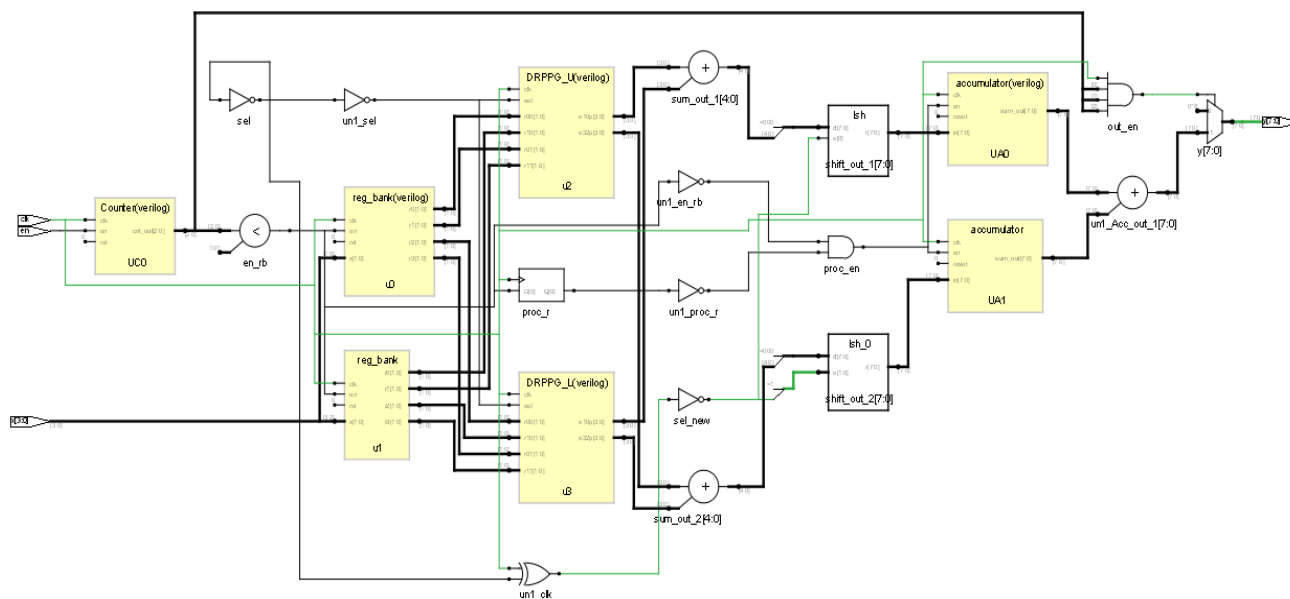


Figure.11 RTL schematic diagram of 8B & 3T

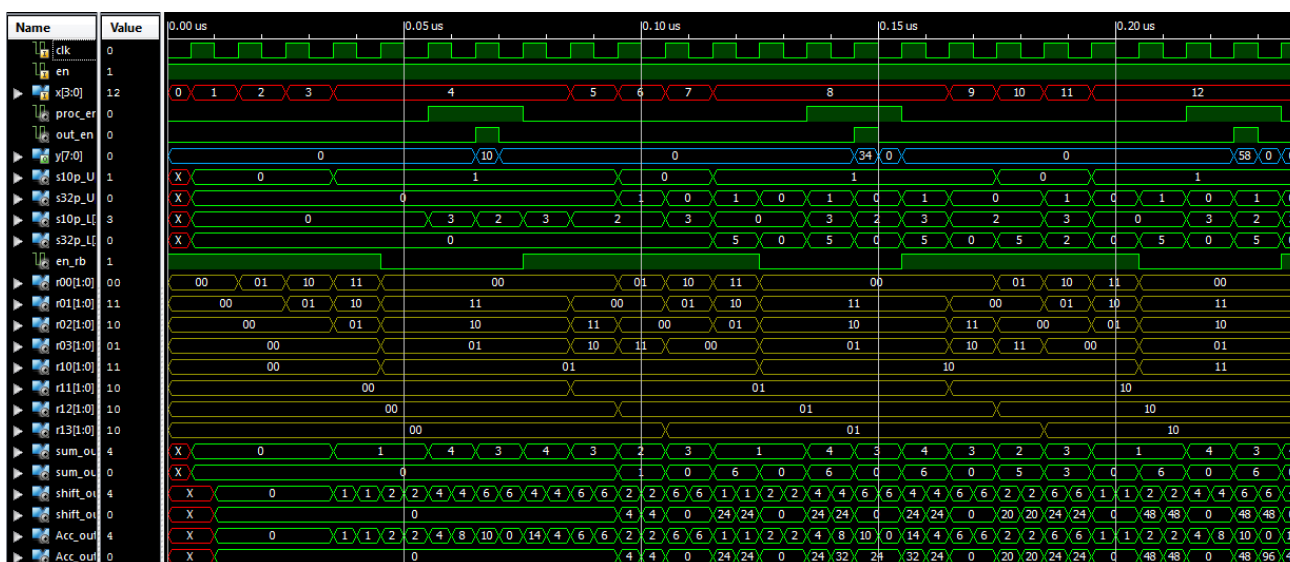


Figure.12 Output wave form of 8B & 3T

The output waveform of 8B & 3T is shown in Fig.12. The input value is represented as a red color waveform. For example, ‘4, 3, 2, and 1’ is input value, which is stored in register bank in the form of r00, r01, r02, r03, r10, r11, r12, and r13. That register bank is represented as a brown color waveform. DRPPG output is denoted as s10p\_U, s32p\_U, s10p\_L, and s32p\_L, which gives the output base on MUX selection line and a DRAM in DRPPG. This input such as ‘4, 3, 2, and 1’ is stored in the registered bank, which performs the filter operation according to Section. 3.3.2. Here, consider co-efficient as ‘0, 1, 2, and 3’ for four different inputs. The output value 10 is stored in ‘y’, which is represented as blue color. When proc\_en and out\_en are in a high state (1), the

output is generated in ‘y’. From this waveform, it is clear that the RFIR architecture is working perfectly.

The RTL schematic of LC-CBA-RFIR design for 8B& 3T is shown in Fig.13, which is taken from cadence tool. For ASIC implementation, same code has been used which is used for the FPGA implementation. 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 are also set into the Tcl file. After synthesizing, Area, Power and Delay, the result is displayed in cadence tool. Finally, total area, total delay, total power, APP and ADP are reduced in LC-CBA-RFIR method when compared to the conventional methods.

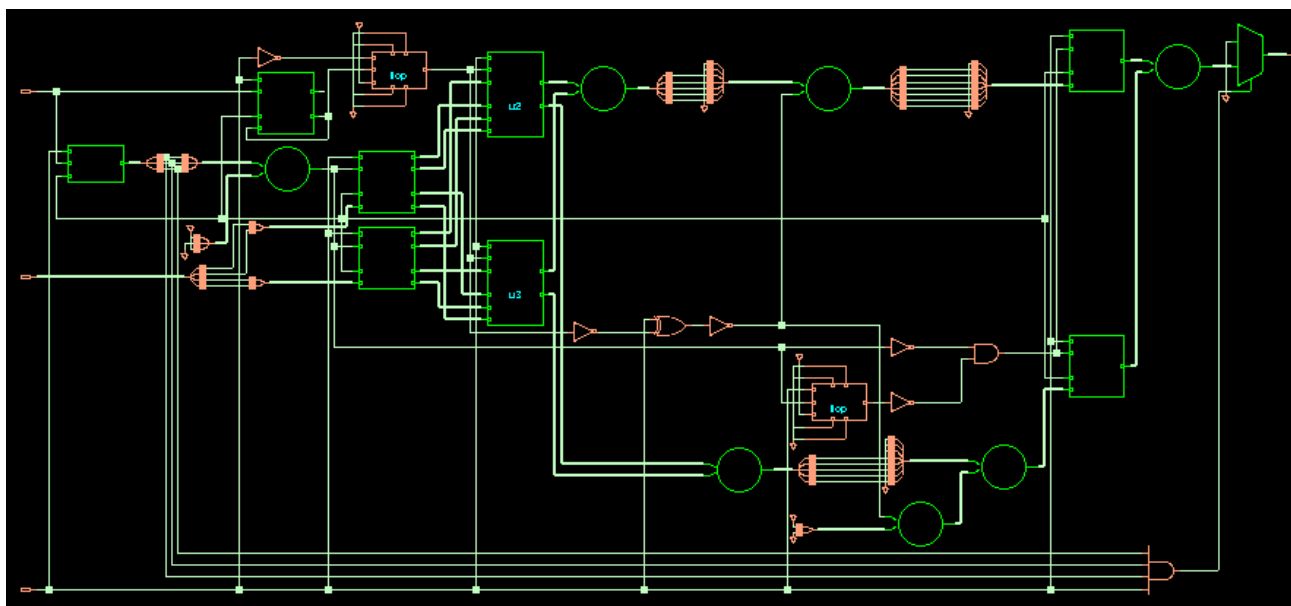


Figure.13 RTL schematic of LC-CBA-RFIR for 8B & 3T in 180nm technology

## 5. Conclusion

In this paper, LC-CBA-RFIR architecture has been implemented in ModelSim software by writing Verilog code. Area, power and the delay parameters are evaluated for different bits and taps like 8 B & 3T, and 8 B & 7T. Using FPGA implementation, LUT's, slices, flip-flops and the frequency improved in LC-CBA-RFIR architecture. In ASIC 180nm technology, 8.44 % of area, 11.05% of power, 18.53% of APP and 10.87% of ADP is minimized in LC-CBA-RFIR where as in the 45nm technology, 17.26% of area, 10.64% of power, 29.11% of APP and 21.07% of ADP is reduced by using LC-CBA-RFIR technique.

In future, this FIR filter design will be performed by using Carry Increment Adder (CIA) to further reduce the hardware utilization like LUT, slices, and flip flop as well as area, power and delay.

## References

- [1] A. Bonetti, A. Teman, P. Flatresse, and A. Burg, "Multipliers-Driven Perturbation of Coefficients for Low-Power Operation in Reconfigurable FIR Filters", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.64, No.9, pp. 2388 – 2400, 2017.
- [2] J. Chen, J. Tan, C.H. Chang, and F. Feng, "A new cost-aware sensitivity-driven algorithm for the design of FIR filters", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.64, No.6, pp.1588-1598, 2017.
- [3] C.Y. Yao, W.C. Hsia, and Y.H. Ho, "Designing hardware-efficient fixed-point FIR filters in an expanding subexpression space", *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol.61, No.1, pp.202-212, 2014.
- [4] N. Bhagyalakshmi, K.R. Rekha, and K.R. Nataraj, "Design and implementation of DA-based reconfigurable FIR digital filter on FPGA", In: *Proc. of International Conf. on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, pp.214-217, 2015.
- [5] A. Liacha, A.K. Oudjida, F. Ferguene, M. Bakiri, and M.L. Berrandja, "Design of high-speed, low-power, and area-efficient FIR filters", *IET Circuits, Devices & Systems*, Vol.12, No.1, pp.1-11, 2017.
- [6] M. Alawad and M. Lin, "Fir filter based on stochastic computing with reconfigurable digital fabric", In: *Proc. of the International Conf. on Field-Programmable Custom Computing Machines (FCCM)*, pp.92-95, 2015.
- [7] A. Rasekh and M.S. Bakhtiar, "Design of Low-Power Low-Area Tunable Active RC Filters", *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.65, No.1, pp.6-10, 2018.
- [8] P.K. Meher and S.Y. Park, "High-throughput pipelined realization of adaptive FIR filter based on distributed arithmetic", In: *Proc. of the 19th International Conf. on VLSI and System-on-Chip (VLSI-SoC)*, pp.428-433, 2011.
- [9] C.S.V. Patnam and E. Chitra, "Efficient FPGA Realization of DA-Based Reconfigurable FIR Digital Filter", *IJRECE*, Vol.3, No.2, pp.24-28, 2015.
- [10] 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.
- [11] J.L.M. Iqbal and S. Varadarajan, “High Performance Reconfigurable FIR Filter Architecture Using Optimized Multiplier”, *Circuits, Systems, and Signal Processing*, Vol.32, No.2, pp.663-682, 2013.
- [12] R. Thakur and K. Khare, “High Speed FPGA Implementation of FIR Filter for DSP Applications”, *International Journal of Modeling and Optimization*, Vol.3, No.1, pp.92, 2013.
- [13] S.J. Lee, J.W. Choi, S.W. Kim, and J. Park, “A reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption”, *IEEE transactions on very large scale integration (VLSI) systems*, Vol.19, No.12, pp.2221-2228, 2011.
- [14] P. Stalin and C. Arun, “A Study on Low Power Reconfigurable FIR Filters with Dynamic Change in Filter Order”, *Journal of Theoretical & Applied Information Technology*, Vol.54, No.2, pp.230-237, 2013.
- [15] B. Rashidi, B. Rashidi, and M. Pourormazd, “Design and Implementation of Low Power Digital FIR Filter based on low power multipliers and adders on xilinx FPGA”, In: *Proc. of the 3rd International Conf. on Electronics Computer Technology (ICECT)*, pp.18-22, 2011.
- [16] S.Y. Park, and P.K. Meher, “Low-power, high-throughput, and low-area adaptive FIR filter based on distributed arithmetic”, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol.60, No.6, pp.346-350, 2013.
- [17] S. Ramanathan, G. Anand, P. Reddy, and S.A. Sridevi, “Low Power Adaptive FIR Filter Based on Distributed Arithmetic”, *Int. Journal of Engineering Research and Applications*, Vol.6, No.5, pp.47-51, 2016.
- [18] N. Sriram and J. Selvakumar, “A Reconfigurable FIR Filter Architecture to Trade Off Filter Performance for Dynamic Power Consumption”, *Int. J. Adv. Comput. Theor. Eng.(IJACTE)*, Vol.2, No.1, pp.112-119, 2013.
- [19] K.M. Basant, P.K.Meher, S.K. Singhal, and M.N.S. Swamy, “A high-performance VLSI architecture for reconfigurable FIR using distributed arithmetic”, *Integration, the VLSI Journal*, Vol.54, pp.37-46, 2016.
- [20] R. Jia, H.G. Yang, C.Y. Lin, R. Chen, X.G. Wang, and Z.H. Guo, “A Computationally Efficient Reconfigurable FIR Filter Architecture Based on Coefficient Occurrence Probability”, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, Vol.35, No.8, pp.1297-1308, 2016.