



# High speed low area OBC DA based decimation filter for hearing aids application

C. Obulesu<sup>1</sup>, Pamisetty Chandana<sup>2</sup>, N.Sravani<sup>3</sup>

<sup>1</sup>Assistant Professor, <sup>2</sup>Assistant Professor, <sup>3</sup>Assistant Professor, ECE Department, Anantha Lakshmi Institute of Technology and Sciences, Ananthapuramu, Andhra Pradesh, India.

## Abstract

*This brief introduces a decimation filter for hearing aid application using distributed arithmetic (DA) approach. In this paper, we propose a reconfigurable offset-binary code (OBC) DA based finite impulse response (FIR) filter with a shared look-up table (LUT) updating scheme. The size of the LUTs in DA increases exponentially with the order of filters. Shared LUT based DA structure is a solution to reduce this large memory requirement for higher order filters. The proposed shared LUT updating scheme uses LUT partitioning in which coefficients are spilt into small length vectors and it ensures a drastic reduction in the size of LUTs. The proposed DA filter is synthesized on CMOS 90 nm technology using Synapsis ASIC Design Compiler. The proposed design achieves high speed at a reduced area-delay product (ADP) when compared with recent designs. The proposed architecture is implemented and tested on Virtex 5vsx95-1ff1136 FPGA and the results show that the proposed design involves less number of slices and offers high speed than existing designs. A three-stage decimation filter of hearing aids is designed with the proposed FIR filter and is implemented on the target device by Matlab simulink and Xilinx system generator.*

## Key words

Distributed arithmetic, Decimation filter, Hearing aid, Look-up-table (LUT), Offset binary code.

## Introduction

Hearing Aid is an acoustic gadget used to provide clear, amplified and distortion free audio signal which synchronizes with the normal hearing audiogram signal. Abundant signal processing algorithms have been introduced in the hearing aids to get rid of the noise. Hearing aids are of two types. They are analog hearing aid and digital hearing aid. Analog hearing aids convert the input signal to an electrical signal and the digital hearing aids convert the input signal to numerical binary code. Digital hearing aids are more vantage than analog hearing aids because



they are more flexible and are self adjustable. The basic operation performed by the digital hearing aid is amplification, filtering, and output limiting [1]. Fig.1 (a) shows the block diagram of advanced the digital hearing aid. Digital Hearing aid comprises of preamplifier, sigma to delta analog to digital (ADC) converter, digital decimation filter and sigma to delta digital to analog (DAC) converter. Digital decimation filter plays a paramount role in the hearing aids to scale down the sampling rate and increase the efficiency of the speech signal [2]. Normally decimation filters are accomplished by using finite impulse response (FIR) filters due to their linear phase and stability in nature. The Decimation filter consists of a combination of cascaded integrated comb (CIC) filter and two FIR filters and is shown in Fig. 1(b). The complexity of filters is contributed mainly by the complexity of coefficient multipliers. In the decimation filter of Fig.1b, CIC is implemented by using adders and delays and is a multiplier less architecture. Hence the decimation filter can be optimized further by implementing the half band and corrector FIR filters using multiplier less designs [3]. In recent years, several multiplier-less FIR filter designs have been presented. Conversion based design and Memory based design are prominent among them. In conversion based design, input signals are represented in a non-binary form which occupies more area and is suitable for limited applications.

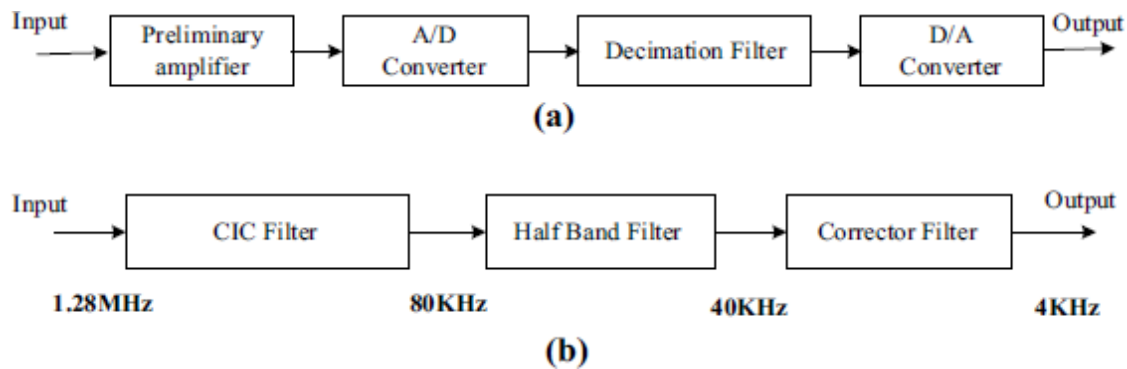


Fig.1(a). Hearing aids 1(b). Digital decimation filter

Distributed Arithmetic (DA) is one of the omnipotent memory based multiplier-less architecture used in FIR filters. Here, the filter partial products are precomputed and saved in the look-up-table (LUT). Later they are shift accumulated to produce the convolution sum. DA based architectures offer high throughput and possess better regularity [4]. Generally, in basic DA based FIR filters [5], the filter coefficients cannot be modified during run-time. To change the filter coefficients during run-time, reconfigurability is introduced in the FIR filters [6] [7][8]. Croisier et al introduced DA for the first time [9]. Later White proposed mathematical calculation and application of DA [10] in DSP. Anderson et al proposed offset binary code (OBC) based DA [11] in the adaptive FIR filter. The author [12] uses two separate LUTs in design, one LUT is for calculating filter output and another LUT is for filter updating. Meher [13] proposed systolic DA based FIR filter which is suitable for higher order filters. Also, Meher developed re-writable look-up-table (LUT) based design in which filter coefficients can be changed during run-time [14]. Ozalevli et al. [15] presented an architecture for DA, where the data signals are in analog form. Das [16] proposed decimation filter for hearing aids using basic DA based FIR filter.

Haw-Jing developed a [17] reusable DA FIR architecture for affine projection algorithm. Waelchli et al. implemented DA for efficient baseband processing in software receiver. Chitra discussed the analysis and implementation of high performance reconfigurable FIR filter using DA. In the context of aforesaid observations, in this article, we propose a shared LUT OBC DA based FIR filter design in decimation filter. In this article, LUTs are shared by various weights according to the DA input bits. Here the DA input samples are in offset binary code format. The proposed architecture is area and power efficient when compared with existing architectures. The



rest of this paper begins with Section 2 which reveals the mathematical calculation of OBC DA based FIR filter. Section 3 describes the implementation of proposed reconfigurable shared LUT OBC DA based FIR filters. The FPGA and ASIC results of proposed and existing designs is discussed in Section 4. Section 5 explains the application of OBC DA based FIR filter in hearing aids and implementation of digital decimation filter using Matlab Simulink and Xilinx system generator tool. Finally, Section 6 concludes the paper.

## 2. Formulation of OBC DA algorithm

Distributed Arithmetic (DA) is a bit serial operation which is used to compute correlation, convolution and inner products, which are imperative operations present in FIR filters. The significance of DA is its high efficiency of mechanization. In the case of basic DA implementation, when the filter order is increased, the LUT size increases, which affects both area and performance. By using LUT concept size of DA is reduced and performance is increased. Here, instead of binary data (1, 0), a signed data (-1, 1) is used.

The output of an FIR filter  $y(n)$  with 'N' order is given as follows.

Let us consider the inner product of two vectors  $d$  and  $x$  given by

$$y = \sum_{k=1}^K d_k x_k \quad (1)$$

where  $d_k$  is fixed coefficient,  $x_k$  is the input signal and  $K$  is the number of input words.  $x_k$  is an  $N$ -bit scaled 2's complement binary number such that  $|x_k| < 1$ , and  $x_k$  can be expressed as:

$$x_k = -b_{k0} \sum_{n=1}^{N-1} b_{kn} 2^{-n} \quad (2)$$

$$x_k = \{b_{k0}, \dots, b_{k(N-1)}\}$$

where  $b_{kn}$  denotes  $N^{\text{th}}$  bit of  $x_k$ .

By substituting equation (2) in (1),  $y$  is given in expanded form as



$$y = \sum_{k=1}^K d_k [-b_{k0} + \sum_{n=1}^{N-1} b_{kn} 2^{-n}] \quad (3)$$

Equation (3) is the conventional form of inner product expression. By interchanging the summation order, finally we get equation as shown below

$$y = \sum_{n=1}^{N-1} [\sum_{k=1}^K d_k b_{kn}] 2^{-n} + \sum_{k=1}^K d_k (-b_{k0}) \quad (4)$$

Let us write  $x_k$  as

$$x_k = \frac{1}{2} [x_k - (-x_k)] \quad (5)$$

The two's complement form of  $-x_k$  can be written as

$$-x_k = -\bar{b}_{k0} + \sum_{n=1}^{N-1} b_{kn} \bar{b}_{kn} 2^{-n} + 2^{-(N-1)}$$

On substituting  $x_k$  and  $-x_k$  in equation (1), we get

$$x_k = \frac{1}{2} [-(b_{k0} - \bar{b}_{k0}) + \sum_{n=1}^{N-1} (b_{kn} - \bar{b}_{kn}) 2^{-n} - 2^{-(N-1)}]$$

For simple notation, let us assume

$$s_{kn} = \begin{cases} b_{kn} - \bar{b}_{kn}, n \neq 0 \\ -(b_{kn} - \bar{b}_{kn}), n = 0 \end{cases} \text{ where } s_{kn} \in \{-1, 1\} \quad (6)$$

On substituting these values we get

$$y = \frac{1}{2} \sum_{k=1}^K d_k [\sum_{n=0}^{N-1} s_{kn} 2^{-n} - 2^{-(N-1)}] = \sum_{n=0}^{N-1} P(b_n) 2^{-n} + 2^{-(N-1)} P(0)$$

where

$$P(b_n) = \sum_{k=1}^K \frac{d_k}{2} s_{kn}$$

$$P(0) = - \sum_{k=1}^K \frac{d_k}{2} \tag{7}$$

$P(b_n)$  has  $2^{(k-1)}$  possible values and  $P(0)$  is the one word-initial condition register. Hence the size of the memory in OBC DA is reduced to  $2^{k-1}$  instead of  $2^k$ .

### 3. Proposed shared LUT updating OBC DA based FIR filter:

For large values of ‘N’ the memory size of DA will be increased eventually. To get rid of this, let the filter order of ‘N’ is spitted into ‘R’ vectors such that  $N = RM$ . Here ‘R’ and ‘M’ can be any two positive integers.

$$y = \sum_{l=0}^{L-1} 2^{-l} \left[ \sum_{r=0}^{R-1} \sum_{m=0}^{M-1} [d_{(m+rM)l} [b_{(m+rM)l} - \bar{b}_{(m+rM)l}]] \right] \tag{8}$$

Let  $S_{(m+rM)l} = b_{(m+rM)l} - \bar{b}_{(m+rM)l}$

From eq. 8, we analyze that  $S_{(m+rM)l}$  has the input bit sequence which will act as a selection line to the filter coefficient present in the shared LUT register bank. The proposed shared LUT updating OBC DA (SLU OBC DA) architecture consists of offset-binary code unit, reconfigurable shared LUT updating unit, pipeline shift adder cell, and pipeline adder cell. Fig. 2 illustrates the block diagram of shared LUT OBC DA based FIR filter. For ‘N’ tap filter, let  $d(n)$  be the filter coefficient and  $x(n)$  be the input sample.

The block diagram for the OBC unit is shown in Fig. 3. For every clock pulse, the input sample  $x(n)$  of each word length  $L$  is passed to SIPO shift register of size  $N$ . Let decomposed samples of SIPO be  $x_1x_2x_3, \dots \dots \dots x_N$ . The decomposed samples  $x_2x_3, \dots \dots \dots x_N$  are XORed with  $x_1$  to obtain the offset binary code. The ‘N’ decomposed output samples of OBC unit are divided to ‘R’ sample vectors with each  $(M - 1)$  bit length, which are chosen according to the relation  $N = RM$  and are passed to the ‘R’ reconfigurable shared LUT updating units (RSAU) to compute the inner product. Fig.4 shows the internal architecture of the RSAU. Each RSAU contains one shared register bank and  $L (2^{M-1} \times 1)$  multiplexers. Each shared register bank contains  $2^{M-1}$  registers and stores all the computed filter coefficient values. The word length of the filter coefficient is  $Z$  stored outputs from the register bank are given as data signal to the

multiplexer and the outputs from the OBC unit are used as select lines for the multiplexer. Then RSAU will generate L partial outputs from the multiplexers. The memory occupied by the architecture is drastically decreased by LUT sharing through L-bit slices. To access the LUT in parallel, the register arrays are used rather than memory based LUT. Moreover, the data present in this LUT can be updated concurrently to get the required filter coefficient. The partial outputs from multiplexers of the RSAU are passed to the L pipeline adder cells (PACs). The output from multiplexer of every RSAU is fed as input to the first pipeline adder. Similarly, the output from any nth multiplexer from every RSAU is fed to nth pipeline adder. The circuit diagram of PAC is shown in Fig.5. The output of each PAC cell will have a bit length of L. The outputs from PACs are given to the PSAC where they will do shift and accumulate to provide the final sum.

Table 1 shows the hardware requirement for the rewritable DA architecture and the proposed architecture. The total number of registers required for rewritable DA architecture is  $960(15 \times 64)$  where a total number of registers required for the proposed design is  $512(8 \times 64)$ .

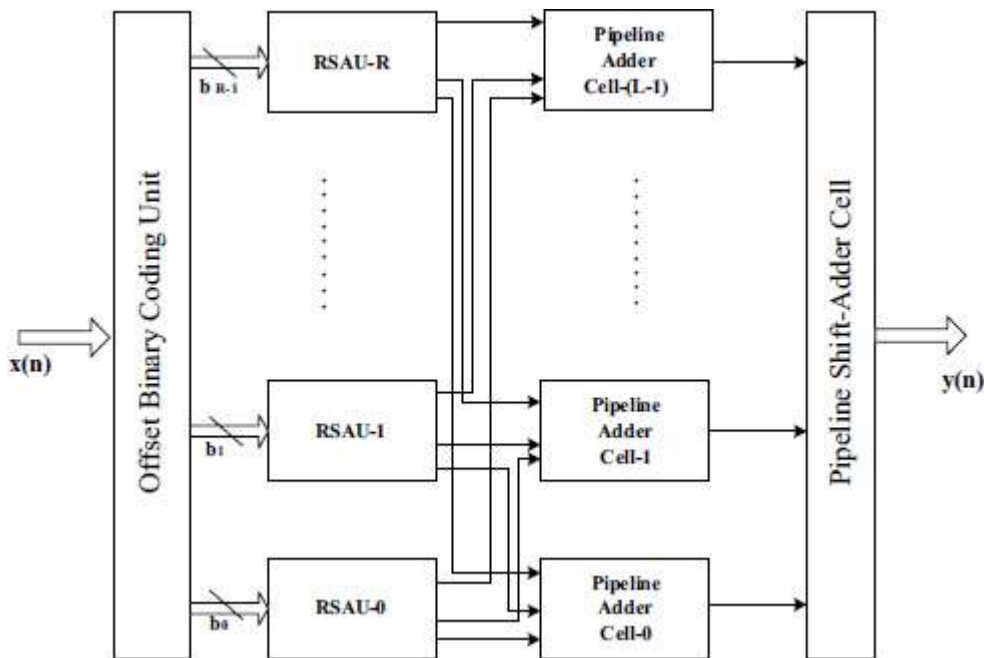


Fig.2. SLU OBC DA based FIR Filter

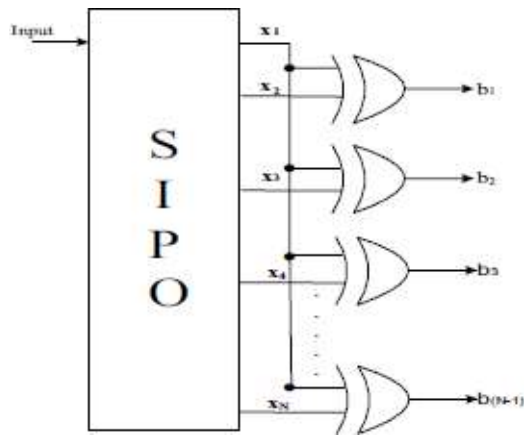


Fig.3. Offset binary code unit

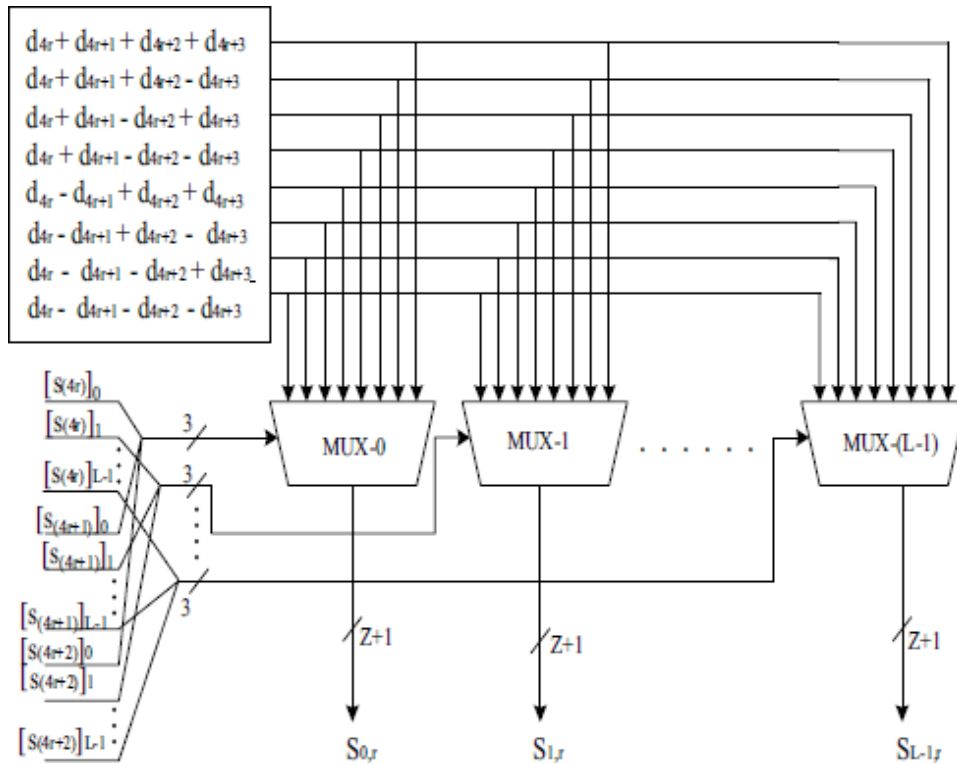


Fig.4. RSAU architecture for R = 4



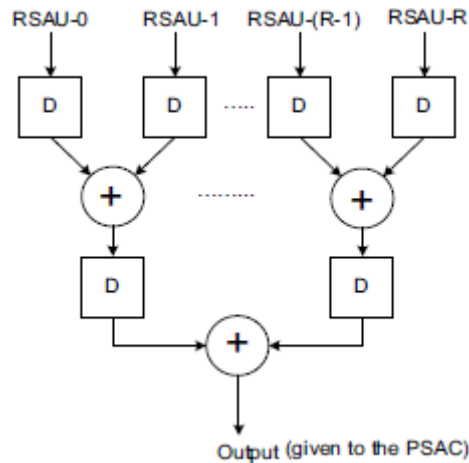


Fig.5. Pipeline adder tree

Design	Architecture of Park and Meher (2014)	Proposed architecture
Number of RSAU	64	64
Number of register banks	64	64
Type of Multiplexer	16X1	8X1
Number of multiplexers in each RSAU	16	16
Number of registers(In one register bank)	15	8
Number of pipeline adders	16	16
Total registers	960	512

Table 1: Comparison summary of proposed SLU OBC DA based FIR filter for N=256

#### 4. Experimental results and discussion

The proposed work is coded in Verilog Hardware Description Language and synthesized by Synopsys ASIC Design Compiler using 90 nm CMOS library. For the 16 tap FIR filter, an 8-bit input data and eight-bit filter coefficients are considered. Area, power delay product (PDP),



area-delay product (ADP), maximum sampling frequency (MSF), minimum sample period (MSP) are evaluated and tabulated in Table 2. The synthesis results of the proposed SLU OBC DA based FIR filter architecture is compared with the existing DA based pipelined architecture, DA based systolic architecture and rewritable DA based FIR filter architectures. From the synthesis results, it can be noticed that the area occupied by the proposed architecture is very less when compared with the other architectures. The proposed SLU OBC DA based FIR filter occupies 45% less area when compared with the work of Park and Meher (2014), 15% less area

Design	Area	ADP	MSP	MSF	PDP
Meher (2006)	71195	127439	1.79	558	60.76
Park and Meher (2014)	25163	39757	1.58	630	13.03
Meher and Park (2011)	18257	94936	5.20	192	32.38
Synposys (2012.06-sp2)	51994	172620	3.32	301	13.63
Proposed architecture	16688	23060	1.38	724	13.00

Units : Area:  $\mu^2m$ , Area Delay Product (ADP):  $\mu^2m \times ns$ , Minimum Sampling Period (MSP) :  $ns$ , Maximum Sampling Frequency (MSF):  $MHz$ , Power Delay Product (PDP):  $mW \times ns$

Table 2: Comparison of ASIC synthesis results for a 16-tap FIR filter.

when compared to the results of Meher and Park (2011) and 95% less area when compared with the results of architecture (Meher 2006). The power consumption of the proposed design is less when compared with other mentioned architectures. The proposed design has less ADP and PDP when compared with that of Meher (2006) and Synopsis (2012). Figure.6 shows the physical

layout design for the SLU OBC DA based FIR filter architecture. The hardware implementation of the SLU OBC based DA FIR filter is carried out on the Xilinx Virtex 5vsx95t1ff1136 FPGA device. The no. of slice registers (SREG), the no. of slice LUTs (SLUT), the slice delay product (SDP), MSP, MSF, number of slices (NOS) are tabulated in Table 3. The proposed design has 50% less number of SREG when compared with the architecture of Meher (2006) and 10% lesser number of SREG than that of Xilinx (2010). The proposed design supports up to 350 MHz input frequency.

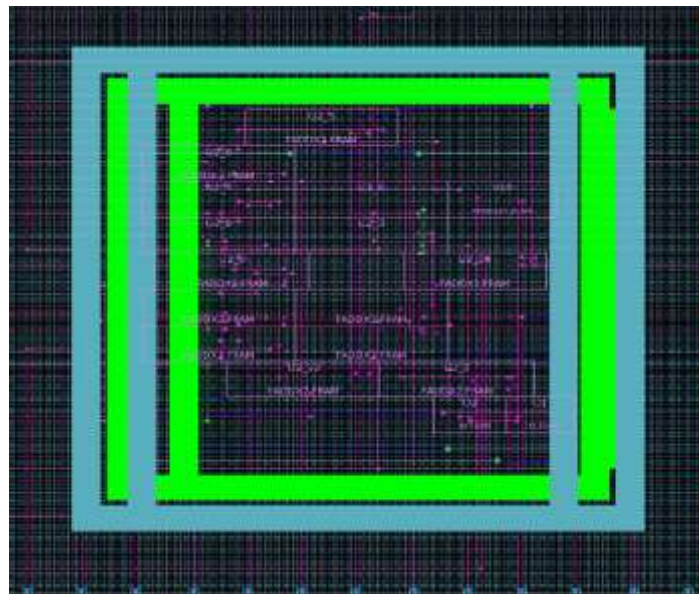


Fig.6. Layout for SLU OBC DA based FIR filter

Design	SREG	SLUT	MSP	MSF	NOS	SDP
Meher (2006)	688	833	4.17	239	275	1148
Meher and Park (2011)	412	267	17.35	57	178	3088
Xilinx (2010)	970	806	3.96	252	368	1458
Proposed architecture	323	268	2.814	355.366	134	377

Table 3: Performance comparison of xilinx virtex(5VVSX95T-1FF1136) for N=Z=8, and K=16

## 5. Application of distributed arithmetic in hearing aids

In this section, the application of SLU OBC DA in decimation FIR filter for digital hearing aids is illustrated. The hearing aid is an electronic gadget worn by people with hearing challenges to hear the sound signal properly. Normally human ear can hear the signal ranging from 20 Hz to 20 KHz and is more sensitive to hear the sound signal when the frequency range is in between 1 KHz and 4 KHz. So the hearing aids need to be designed to meet these specifications. In hearing aids, we use sigma-delta ADC modulator to convert the analog signal to sampled digital signal with high frequency. This high-frequency signal is passed to the digital decimation filter to provide the low-frequency signal. Decimation filter consists of a cascade of CIC filters, half-band filter and corrector filter as shown in Fig. 1(b).

CIC shown in Fig. 7 is the combination of integration section, decimation section, and the comb stage. The high frequency signal from the sigma-delta modulator is passed to CIC and

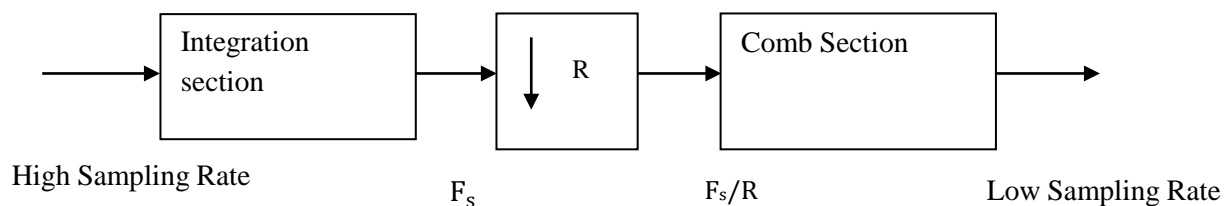


Fig.7. CIC

band FIR filter. The half band low pass FIR filter reduces the bandwidth (BW) of the input data by a factor of 2. Hence the frequency is further reduced and passed to the corrector filter. In the corrector filter, undesired signals are removed. As CIC filters do not involve power hungry multipliers in its architecture, the decimation filter can be optimized by implementing the half-band and corrector FIR filters with multiplier-less architecture. In the present work, we propose a decimation filter with SLU OBC DA based half band and corrector FIR filter. The proposed decimation filter for hearing aids is designed by using Matlab Simulink and Xilinx system generator tool and is shown in Fig.8.

**Implementation of decimation filters for hearing aids:**

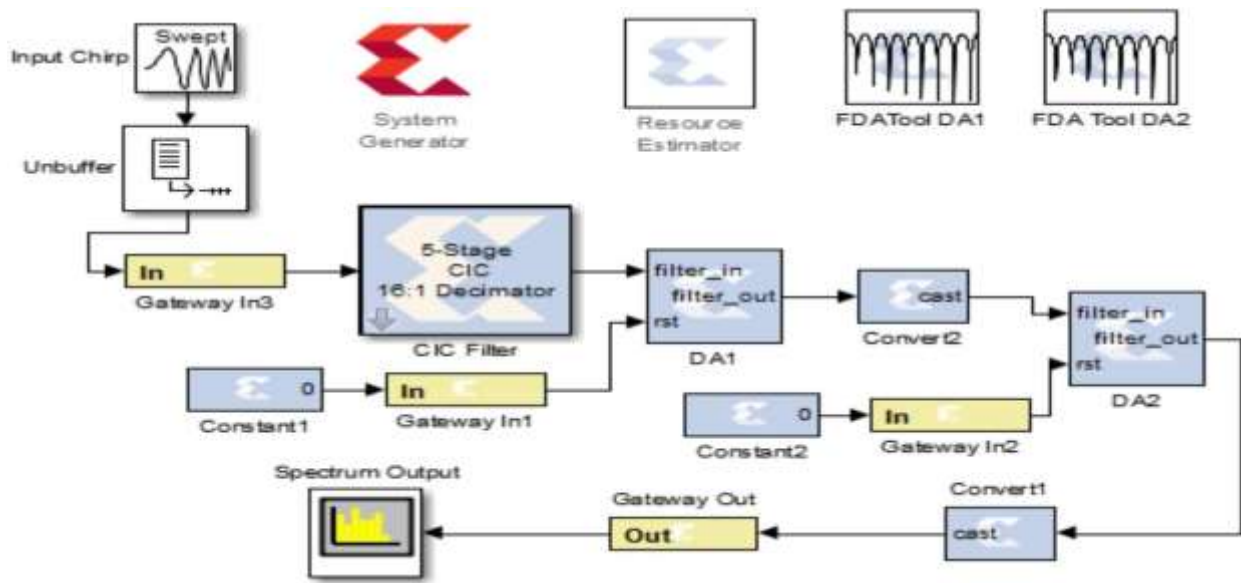


Fig.8. Implementation of digital decimation filter for hearing aids

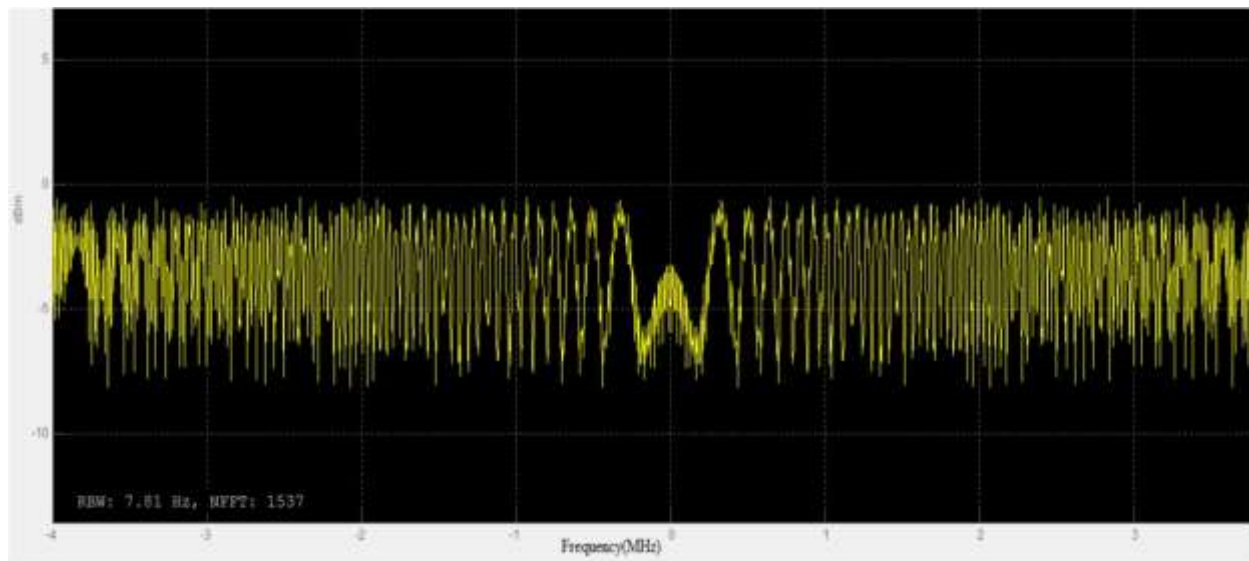


Fig.9. Input signal to the decimation filter

An input bit stream of 1.28 MHz is given to the decimator and is shown in Fig.9. The first stage of decimator which is CIC (5-stage comb filter) divides the input signal by 16 times and outputs a signal of 80KHz. Half band filter and corrector filter stages decimate the signal to

40 KHz and 4 KHz respectively. The Proposed SLU OBC DA based FIR filter is used for implementing half band as well as corrector FIR filters. Half band filter designed here has a pass-band (PB) frequency of 20 KHz and 35 KHz cut-off frequency. Its magnitude output response is shown in Fig. 10. The corrector filter removes unwanted noise signals and the filter is designed with sampling frequency 40 KHz, PB frequency of 4 KHz and 15 KHz cutoff frequency. Fig. 11 shows the magnitude response of the corrector filter. The designed corrector filter has PB frequency of 4 KHz to match human ear characteristics. The final output magnitude response of the decimation filter is shown in Fig.12.

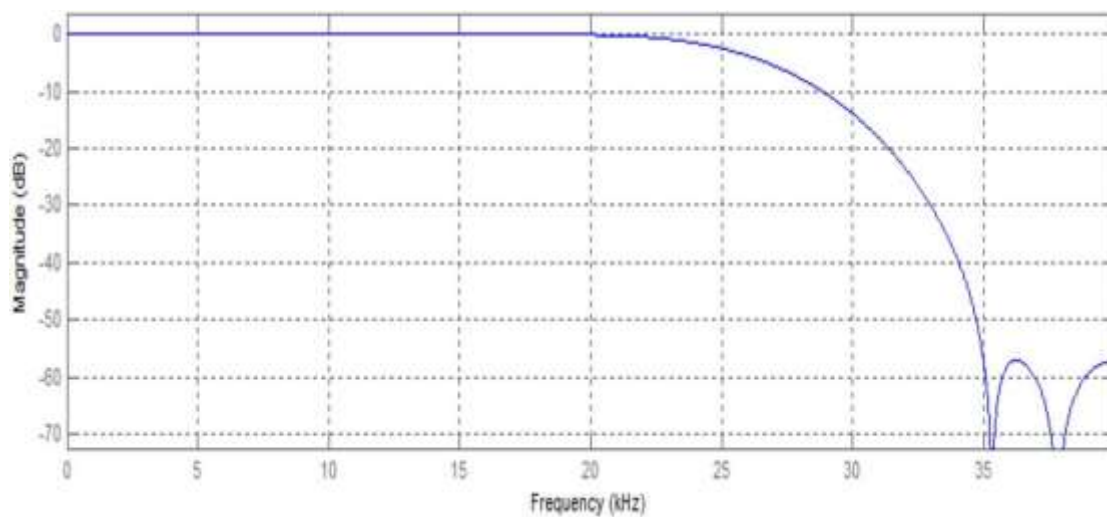


Fig.10. Frequency response waveform of half band filter

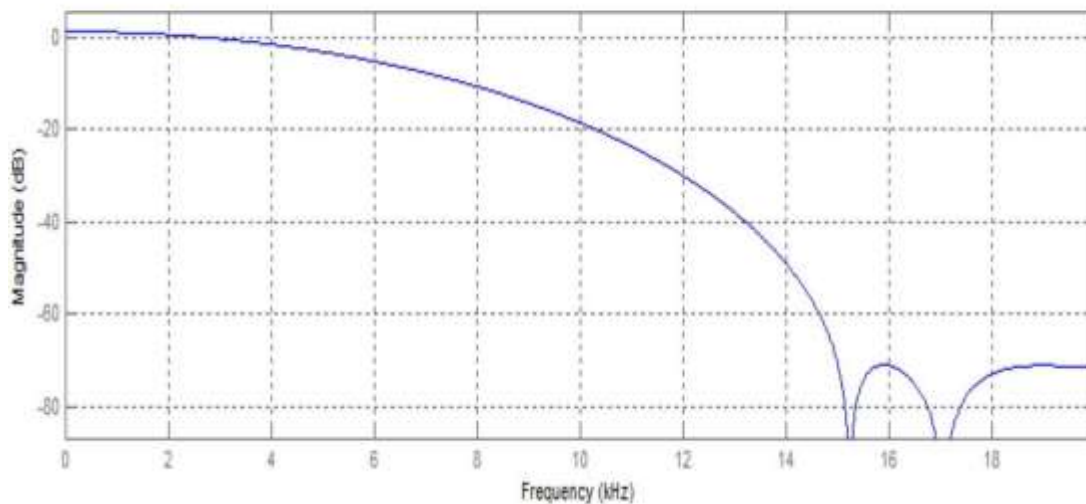


Fig.11. Frequency response of corrector filter



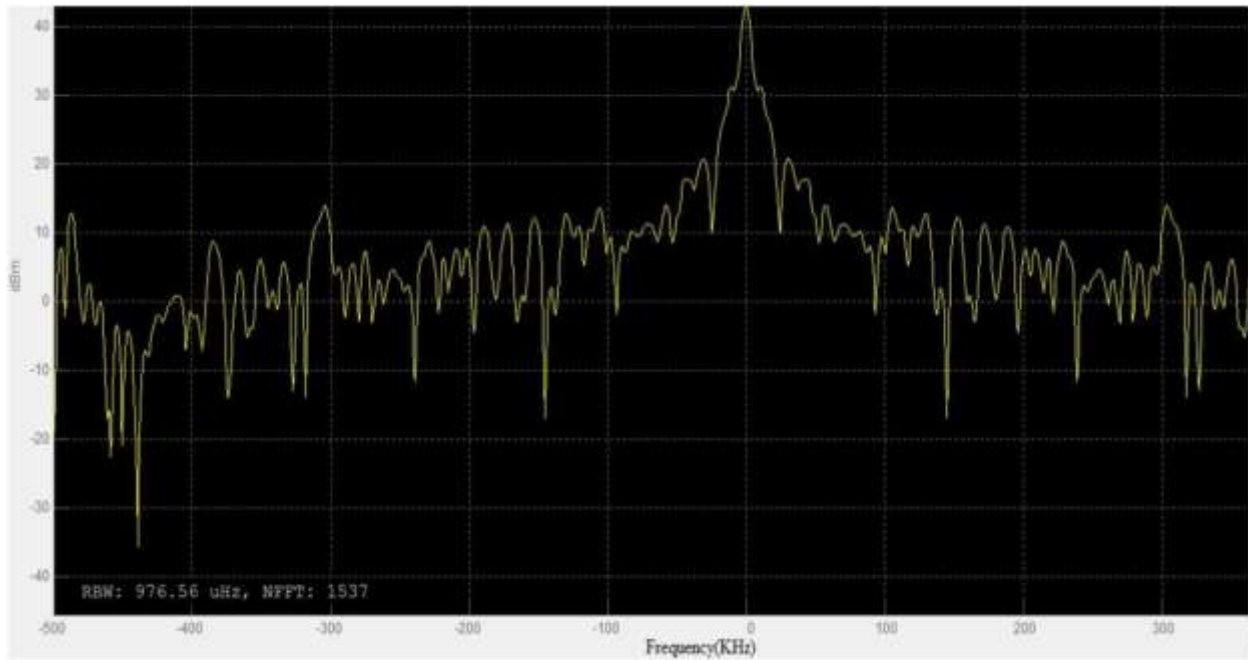


Fig.12.Frequency response of decimation filter

## 6. Conclusion

In this paper, a reconfigurable SLU OBC DA based FIR decimation filter for digital hearing aid application has been proposed. By using the OBC concept and SLU updating scheme, the throughput rate of the proposed architecture is improved. The area complexity of the proposed design is drastically reduced by SLU technique, where the coefficients in the register bank are divided into vectors of smaller bit length for different bit slices. The proposed design has 40% less ADP when compared with the existing designs. It is implemented on FPGA Virtex 5vsx95t-1ff1136 and these results show that the design utilizes 15% less number of slices than the existing designs. From the results, it is evident that the proposed decimation filter will give a cost effective solution for the hardware implementation of the hearing aids.

## References

1. Kuo YT, Lin TJ, Li YT, Liu CW. Design and implementation of low-power ANSI S1. 11 filter bank for digital hearing aids. *IEEE Transactions on Circuits and Systems I: Regular Papers.*2010; 57(7):1684-96.



2. *Levitt H. Digital hearing aids: a tutorial review. J Rehab Res Dev. 1987 Sep; 24(4):7-20.*
3. *Xin J, Qi Y. Mathematical Modeling and Signal Processing in Speech and Hearing Sciences. Springer Science and Business Media; 2014 Apr 14.*
4. *Reed JH. Software radio: a modern approach to radio engineering. Prentice Hall Professional; 2002.*
5. *NagaJyothi G, SriDevi S. Distributed arithmetic architectures for FIR filters-a comparative review. In: IEEE 2017 International Conference on Wireless Communication, Signal Processing and Networking(WiSPNET);chennai,TamilNadu,India; 2017.pp. 2684-2690.*
6. *Longa P, Miri A. Area-efficient FIR filter design on FPGAs using distributed arithmetic. In: IEEE 2006 International21 Symposium on Signal Processing and Information Technology;Vancouver,BC,Canada; 2006.pp. 248-252.*
7. *Kumm M, Moller K, Zipf P. Reconfigurable FIR filter using distributed arithmetic on FPGAs. In: IEEE 2013 International Symposium on Circuits and Systems (ISCAS2013); 2013. pp. 2058-2061.*
8. *Grande NJ, Sridevi S. Asic implementation of shared lut based distributed arithmetic in FIR filter. In: IEEE 20172 International conference on Microelectronic Devices, Circuits and Systems; Vellore, Tamil Nadu, India; 2017. 3 pp.1-4.*
9. *Croisier A, Esteban D, Levilion M, Riso V, inventors; International Business Machines Corp, assignee. Digital filter for PCM encoded signals. United States patent US 3,777,130. 1973 Dec 4.*
10. *Yoo H, Anderson DV. Hardware-efficient distributed arithmetic architecture for high-order digital filters. In: IEEE 2005 International Conference on Acoustics, Speech, and Signal Processing; Philadelphia, PA, USA; 2005. pp. v-125*
11. *White S. High-speed distributed-arithmetic realization of a second-order normal-form digital filter. IEEE Transactions on Circuits and Systems. 1986;33(10):1036-8.*
12. *White SA. Applications of distributed arithmetic to digital signal processing: A tutorial review. IEEE Assp Magazine. 1989 Jul;6(3):4-19.*





13. Meher PK. *Hardware-efficient systolization of DA-based calculation of finite digital convolution. IEEE Transactions on Circuits and Systems II: Express Briefs.* 2006; 53(8):707-11.
14. Park SY, Meher PK. *E\_icient FPGA and ASIC realizations of a DA-based reconfigurable FIR digital filter. IEEE Transactions on Circuits and Systems II: Express Briefs.* 2014;61(7):511-5.
15. Ozalevli E, Huang W, Hasler PE, Anderson DV. *A reconfigurable mixed-signal VLSI implementation of distributed arithmetic used for finite-impulse response filtering. IEEE Transactions on Circuits and Systems I: Regular Papers.* 18 2008; 55(2):510-21.
16. Das JK. *Low Power Digital Filter Implementation in FPGA.* PhD, National Institute of Technology Rourkela, India, 2009
17. Meher PK, Park SY. *High-throughput pipelined realization of adaptive FIR \_lter based on distributed arithmetic.*