ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

System Generator based area Reduction of IIR Decimation Filter using Merged Delay Transformation

Smitha G L [1], Meharunnisa S P [2]

P.G. Student, Department of E&IE, Dayananda Sagar College of Engineering, Bengaluru, India[1]

Assistant Professor, Department of E&IE, Dayananda Sagar College of Engineering, Bengaluru, India[2]

ABSTRACT : Decimation filters plays an important role in the field of electronics and communication system. The Infinite Impulse Response (IIR) decimation filter is designed by using Merged Delay Transformation (MDT) and then it is implemented by using system generator. The proposed method is effective in terms of area and provides better stability for coefficient quantization. By decomposing into parallel first order and second order sections, the proposed method is applied to higher order filters. The performance of the proposed method is checked through MATLAB Simulink simulations and then it is implemented using system generator.

KEYWORDS: Decimation filter, Merged delay transformation, Infinite Impulse Response, Recursive Filter

I. INTRODUCTION

In signal processing the function of a filter is to remove unwanted parts of the signal, such as random noise or to extract useful parts of the signal, such as the components lying within a certain frequency range. The transfer function of Infinite Impulse Response (IIR) filter consists of both poles and zeros. Decimation is the process of reducing the sampling rate. The decimation factor is the ratio of the input rate to the output rate. Decimation filters are having applications in the field of electronics and communication such as digital subscriber line (ADSL), broad band and video applications [1],[2]. There have been continuous efforts to reduce the computational complexity of decimation filters. The conventional realizations of decimation filters consists of integrators, decimators and a number of differentiator. The larger size and enhanced complexity of calculations are the drawbacks of multistage realizations. These IIR filters are well known for their reduced computational complexity[3].

Computationally IIR filters are much better than FIR filters, which is having superior processing flexibility for IIR filters to reduce the pre-echoes and keep flat frequency phase response for all sample rates. IIR filters are preferred over FIR filter in terms of latency characteristics [8]. The higher order FIR filter requires long critical path delay which demands faster hardware for practical applications. By using IIR filters instead of FIR filters, one can also reduce the hardware complexity and the critical path delay[9]. IIR filters are recursive in nature and feedback is also involved in the process of calculating output sample values.

An improved MDT method for designing IIR filters is designed, which directly computes the current value of the output without computing the intermediate outputs[4]. It means, the new (current) output sample becomes dependent only on the M^{th} old output sample and M input samples. In order to decrease the output data rate, M delay elements are integrated in the recursive path. An N^{th} order IIR filter can be decomposed into N parallel first order sections with complex co-efficients. This is not a problem as two first order sections having complex conjugate co-efficients produce real output for a real input sample. The second order sections are realized with reduced complexity. The input and output data rates are same in case of conventional IIR fiters, where as in decimation filters the input data rate is greater than the output data rate, due to decimation factor. To generate output of decimation with the factor M, it requires all the M-1 output intermediate samples. In recursive system, it is very difficult to have M-1 intermediate output while for non-recursive system it is not a problem. The only possible way to determine the output of the IIR recursive filter

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

directly from M[th] output without calculating the intermediate output is transformation. For example, for M=4, the output can be directly determined by y[0] not requiring y[1], y[2] and y[3]
Paper is organized as follows. Section II talks about literature survey on previous works. The proposed method is given in Section III. Section IV describes Software implementation. Section V provides simulation results , Section VI concludes the work.

II. RELATED WORK

In [1] authors discussed about the area efficient decimation filter based on merged delay transformation for the wireless applications. Decimation filters are the important part of transceivers that are used in wireless technologies like Global system for mobile (GSM), Digital enhanced cordless telecommunication (DECT) and Wi-Fi. The existing merged delay transformation consumes more power. The area and power consumption are done with the proposed method by using FPGA kit. Hence this method is highly suitable for the wireless applications. In [2] during designing of an embedded system, power consumption is an important constraint. In multi-standard receiver, the design of decimation filter is having a deep impact on the power consumption. RF filter, Analog to digital converter and FIR decimation filters are the blocks of general communication chain in a multi-standard receiver. In [3] presents the methodology for next generation wireless standard for synthesizing optimized power decimation filters for wide band delta sigma analog to digital converters. This Delta-sigma Analog to digital filter shapes the quantization noise and also the sampling rate is reduced to the Nyquist rate. These decimation filters are realized by using sinc which are realized as accumulators and differentiators. Here the decimation filters is implemented by using 45nm CMOS technology. In [4] Several techniques are used to reduce the power consumption of decimation filters. In this paper a new design of decimation filter is proposed with reduced hardware complexity, low power consumption and approximate linear phase. The proposed digital decimation filter consists of comb filter, one third band filter and halfband filter. In order to reduce the computational complexity these halfband filters are used. The software implementation is done by using Simulink and DSP blockset and Matlab simulations are performed to verify the magnitude and phase response. Paper [8] gives a brief about rise and fall of the recursive digital filter. In the decade of the 1960s there was an explosive growth in the field of digital signal processing. Digital filters are used in the applications like frequency selectivity, matched filtering as well as adaptive filtering. For these techniques closed loop IIR filter design didn't work, therefore FIR filters were used. The theoretical as well as practical designs of adaptive filtering were suitable only for FIR filters. Linear phase also plays an important role which preserves the waveshape of the desired output.

III. MERGED DELAY TRANSFORMATION (MDT)

A first-order recursive Difference equation of a first-order recursive can be written as follows:

$$y[n] = b1\ y\ [n-1] + a_0\ x\ [n] \quad (1)$$

Here, $x[n]$ is the input data sample and $y[n]$ is the output data samples respectively, where b_1 and a_0 real constants. The input and output data rates are same in case of conventional IIR filters. Due to decimation factor M the input data rate is greater than output data rate in case of decimation filters.
The values of y[n-1], y[n-2], y[n-3],…….y[n-M+1] can be replaced successively in (1) and the following general equation can be derived:

$$y[n] = b_1^{M}\ y[n-M] + \sum_{k=0}^{M-1} b_1^{k}\ a_0\ x[n-k] \quad (2)$$

Equation (2) is called as merged delay transformation For second- and higher order IIR filters . such a direct relationship is not possible. Equation (2) computes current output $y[n]$ from single M^{th} previous output and M inputs. The values of intermediate outputs are not required to calculate for merged delay transformation. The input sampling rate can be M-times higher than the output sampling rate. With the help of this transformation, an IIR filter can be transformed into an M-fold decimation filter. The filter structure to realize is shown in Figure1. In the figure, y[n] is the output and it is fed back after passing through M number of unit delay elements. The output sampling rate can be reduced by merging M number of delay elements. In this way, we obtain one output sample at every M^{th} sample of the input data that realizes down sampling of factor M. Intermediate values of output are not computed. This structure

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

performs M + 1 multiplications as compared to 2M multiplications performed in the IIR decimation filter where all output samples are to be computed.

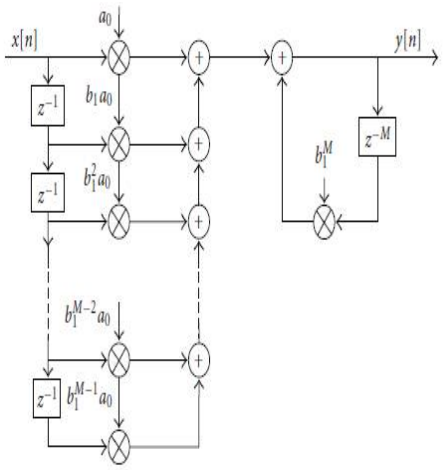

Fig 1: Realization of MDT based M fold decimation in first order IIR filter

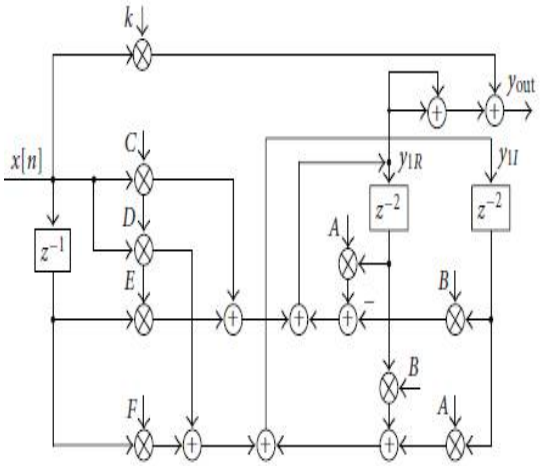

Fig 2: Realization of second order IIR decimation filter with M=2

Direct application of merged delay transformation is not possible for higher-order IIR filters. However, we have proposed a simple technique for the higher-order cases, too. We decompose the Nth order filter into N parallel first-order sections. The coefficients of individual first-order sections may be complex, but they are complex conjugate for a pair of sections. Hence in the implementation, two first-order sections can be combined to form a second-order section to give a real output. The second-order transfer function $H(z)$ can be written in parallel form as follows:

$$H(z) = k + H_1(z) + H_2(z)$$

where

$$H_1(z) = \frac{r_1}{1 - p_1 z^{-1}}$$

$$H_2(z) = \frac{r_2}{1 - p_2 z^{-1}} \qquad (3)$$

Here, $r_1 = r_{1r} + jr_{1i}$ and $p_1 = p_{1r} + j\, p_{1i}$ are complex conjugates of r_2 and p_2, respectively. The symbol "j" denotes imaginary operator.

Let $y_1[n]$ and $y_2[n]$ represent the outputs from $H_1(z)$ and $H_2(z)$, respectively. We can apply MDT on each section and obtain the following results for M = 2

$$y_1[n] = (A+jB)\, y_1[n-2] + (C+jD)\, x[n] + (E+jF)\, x[n-1]$$
$$y_2[n] = (A-jB)\, y_2[n-2] + (C-jD)\, x[n] + (E-jF)\, x[n-1] \qquad (4)$$

$$A = p_{1r}^2 - p_{1i}^2 \qquad B = 2p_{1r}p_{1i}$$
$$C = r_{1r} \qquad D = r_{1i}$$
$$E = r_{1r}p_{1r} - r_{1i}p_{1i} \qquad F = r_{1i}p_{1r} + r_{1r}p_{1i}$$

$$y_{1R}[n] = A\, y_{1R}[n-2] - B\, y_{1I}[n-2] + C\, x[n] + E\, x[n-1]$$
$$y_{1I}[n] = A\, y_{1I}[n-2] + B\, y_{1R}[n-2] + D\, x[n] + F\, x[n-1] \qquad (5)$$

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Here, $y_{1R}[n]$ is the real part of $y_1[n]$ and $y_{1I}[n]$ is the imaginary part of $y_1[n]$ respectively. Similarly expressions for $y_{2R}[n]$ and $y_{2I}[n]$ are obtained. Equation (5) , shows that y[2] can be computed from y[0], y[2], y[4] and so on. Thus $y[M_n]$ can be computed without intermediate outputs and this leads to sample rate reduction by M.

It is observed that

$$y_{1R}[n] = y_{2R}[n]$$

$$y_{1I}[n] = - y_{2I}[n] \quad (6)$$

Since the imaginary parts are equal and opposite, they cancel out at the output. Real parts are equal so computation of only one real part is sufficient to get the output from second-order section. This results in reduction of computational complexity. The output from a second-order section, $y_{out}[n]$ be obtained as follows:

$$y_{out}[n] = k\, x[n] + 2\, y_{1R}[n] \quad (7)$$

The second-order section of MDT- based IIR decimation filter with $M = 2$ is realized as shown in Figure 2 The filter structure can be drawn for any value of M. The number of multipliers for this structure is equal to $2M + 6$. With the help of above procedure, an Nth order IIR decimation filter with any integer decimation factor M can be realized.

IV. SOFTWARE IMPLEMENTATION

This section describes the software implementation of decimation filter in MATLAB tool box and on Xilinx platform. The IIR filter used for the purpose of research is of Butterwoth filter. Here we have considered two orders of filters namely first and second order. The specifications of the filters are sampling frequency 44.2kHz , cut-off frequency 20kHz and Decimation factor M = 4. The butterworth is a type of signal processing filter designed to have a flat frequency response as possible in the passband. The magnitude response of the butterworth filter decreases with increase in frequency from zero infinity. The width of the transition band is more in butterworth filters are used in applications where maximum passband flatness is required. The key steps for the software implementation is given in the fig 3.The output of proposed IIR is compared with actual filter equation output to find out the error.

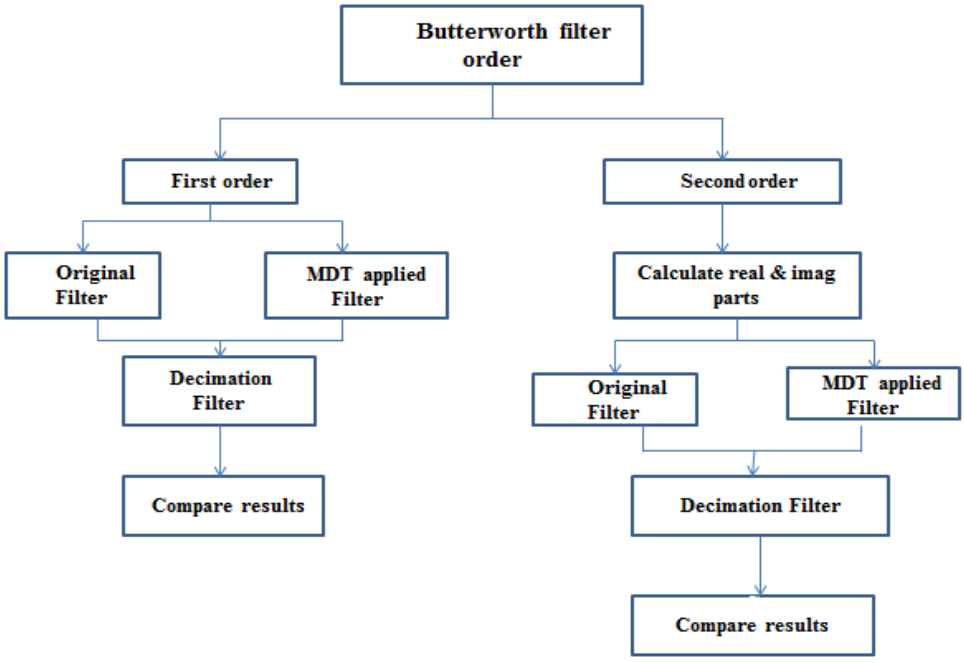

Fig 3: Key steps for Matlab implementation

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

System Generator for DSP is a highly productive design environment for the development and prototyping of DSP systems using FPGAs. The fig 4 shows the steps involved during implementation in Xilinx platforms. Here the DSP blocks are converted into Xilinx bocks by using System generator. After that HDL code is generated for those Xilinx bocks. Synthesis is done by using ISE project navigator and power analysis is done by using X-power analyzer. Then the resultsa are compared.

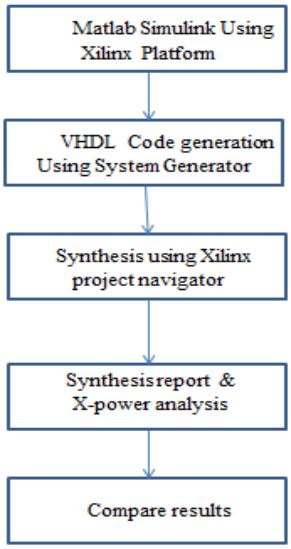

Fig 4: Key steps for implementation in Xilinx platform

V. SIMULATION RESULTS

Simulink is a graphical extension to MATLAB for modeling and simulation of systems. Fig 5 shows the design of the first order IIR Butterworth filter using Simulink blocks. Here the systems are drawn on screen as block diagrams. Simulink is integrated with MATLAB and data can be easily transferred between the programs. Scope window shows the output obtained.

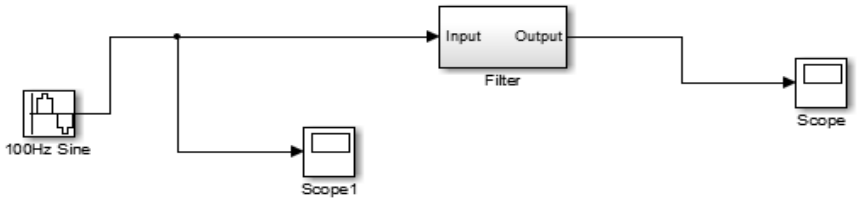

Fig 5: First order filter design using Simulink blocks

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Fig 6 shows the internal structure of first order filter with MDT. The first order Butterworth filter doesn't consists of complex roots. By using MDT the number of multiplications used can be reduced.

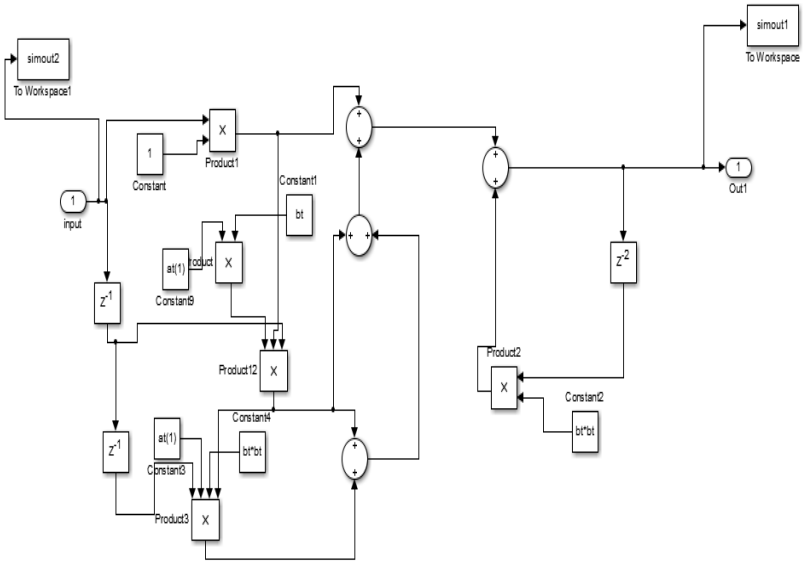

Fig 6: Internal structure of first order filter with MDT

The Second order Butterworth filter consists complex roots which includes real parts and imaginary parts. Here we need to calculate the output from real part and imaginary part. The Fig 7 shows the internal structure of the second order IIR Butterworth filter with MDT is designed by using Simulink bocks.

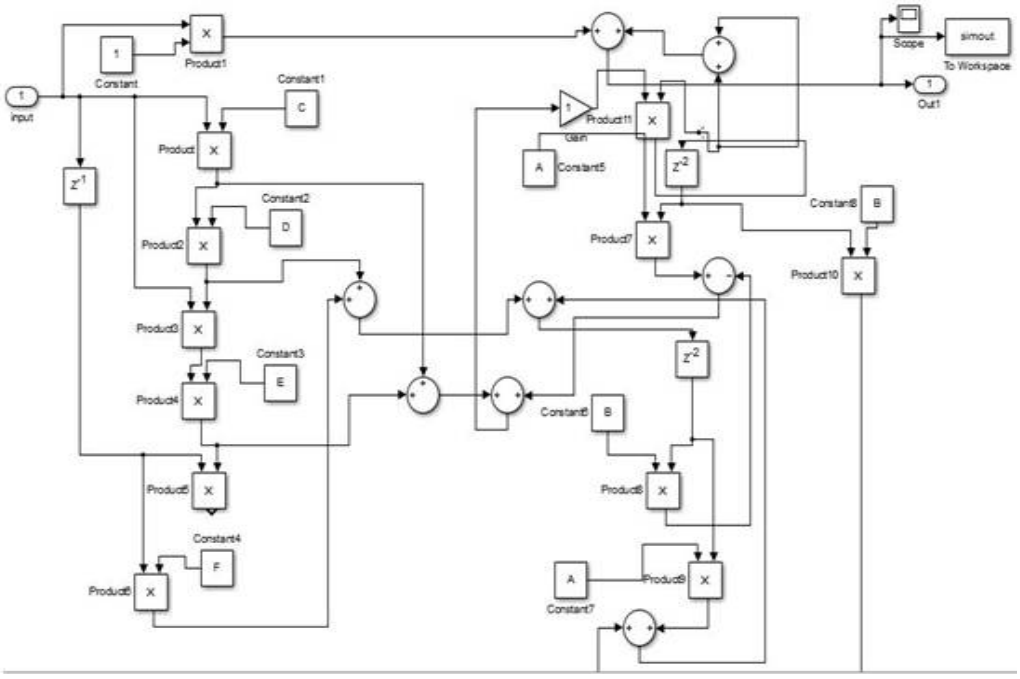

Fig 7: Internal structure of Second order filter with MDT

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Fig 8 and 9 illustrates the simulation results of transformed filter for first and second order with M=4. Fig 8(a) and 9(a) shows the input sine wave. The decimated signal is shown in fig 8(b) and 9(b). The original filter output is depicted in fig 8(c) and 9(c). The output from the proposed transformed IIR decimation filter is shown in the fig 8(d) and 9(d)

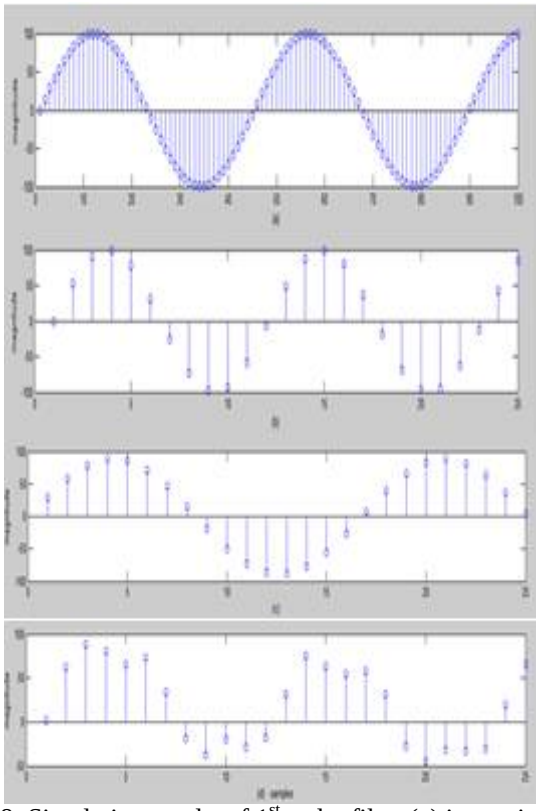

Fig 8: Simulation results of 1st order filter (a) input signal (b) Decimated signal (c) Original filter output (d) Transformed filter output

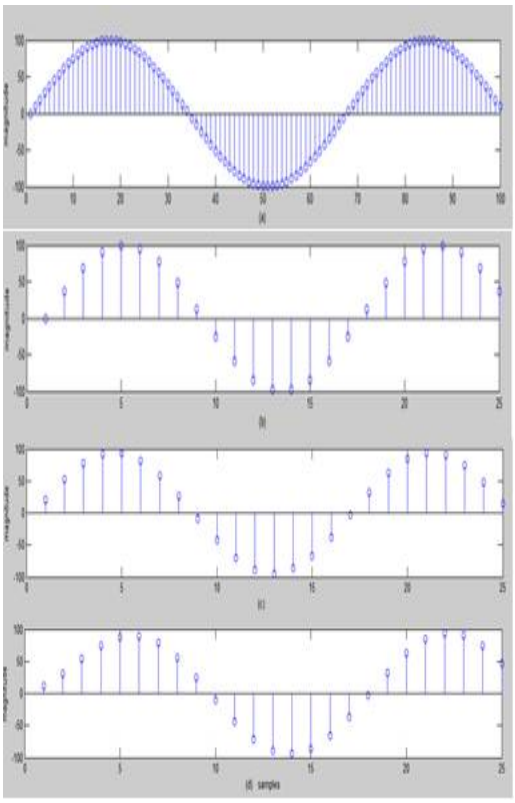

Fig 9: Simulation results of 2nd order filter (a) input signal (b) Decimated signal (c) Original filter output (d) Transformed filter output

The synthesis report obtained from the Xilinx System generator for the filter using MDT and without MDT is given in the table 1. Hence it is revealed in the table that by using MDT we can reduce area as well delay of the filter.

Table 1: Xilinx Implementation results (a) First order filter without MDT (b) First order filter with MDT (c) Second order filter without MDT (d) Second order filter with MDT

Parameters	Values
Number of Slice Register	8205 out of 184305 4%
Number of Fully used LUT FF pairs	2458 out of 139 17%
Number of Bonded IOB's	69 out of 396 17%
Maximum frequency	223.537MHz
Maximum Combinational path delay	13.115ns

(a)

Parameters	Values
Number of Slice Register	61 out of 184305 0%
Number of Fully used LUT FF pairs	16 out of 139 11%
Number of Bonded IOB's	33 out of 396 8%
Maximum frequency	706.71MHz
Maximum Combinational path delay	8.414ns

(b)

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Parameters	Values
Number of Slice Register	4109 out of 184305 2%
Number of Fully used LUT FF pairs	2236 out of 139 22%
Number of Bonded IOB's	33 out of 396 8%
Maximum frequency	12.445MHz
Maximum Combinational path delay	10.743ns

(c)

Parameters	Values
Number of Slice Register	561 out of 184305 0%
Number of Fully used LUT FF pairs	317 out of 139 37%
Number of Bonded IOB's	33 out of 396 8%
Maximum frequency	67.078MHz
Maximum Combinational path delay	6.407ns

(d)

VI. CONCLUSION

A new approach is proposed in this paper to effectively deal with area and speed of IIR decimation filters. First order recursive equation is transformed by merged delay transformation. Drawbacks associated with the cascade of integrators, down samplers, and differentiators are avoided. This method reduces the number of multiplicatin involved in conventional IIR filters. The performance of the proposed method is checked through MATLAB Simulink simulations and then it is implemented using system generator. The results obtained from the Xilinx platform are tabulated and it shows that by using MDT we can reduce the area of the decimation filter.

REFERENCES

[1] U. Rashid, F. Siddiq, T. Muhammad, and H. Jamal, "Area efficient decimation filter based on merged delay transformation for wireless applications," *The Nucleus*, vol. 50, no. 4, pp. 301–310, 2013.

[2] K. Grati, N. Khouja, B. Le Gal, and A. Ghazel, "Power consumption models for decimation FIR filters in multistandard receivers," *VLSI Des.*, vol. 2012, 2012.

[3] H. Johansson and L. Wanhammar, " High-speed recursive filter structures composed of identical all pass sub filters for interpolation, decimation, and qmf banks with perfect magnitude reconstruction," IEEE Transactions on circuits and systems II: Analog and Digital Processing , vol.46,no. 1, pp.16-28,1999

[4] R. M. R. Koppula, S. Balagopal, and V. Saxena, "Efficient design and synthesis of decimation filters for wideband delta-sigma ADCs," in 2011 *IEEE International SOC Conference*, pp. 380–385, 2011.

[5] R. Lagadec and T. G. Stockham, "Dispersive models for A-to-D and D-to-A conversion systems," in Audio Engineering Society Convention 75, 1984.

[6] S. Green, "A new perspective on decimation and interpolation filters," Cirrus Logic Inc., 2004.

[7] S. B. Nerurkar and K. H. Abed, "Low-power decimator design using approximated linear-phase N-band IIR filter," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1550–1553, 2006.

[8] A. Fernandez-Vazquez and G. Jovanovic-Dolecek, "A new method for the design of IIR filters with flat magnitude response," *IEEE Trans. Circuits Syst. I Regul. Pap.*, vol. 53, no. 8, pp. 1761–1771, 2006.

[9] K. Surma-aho and T. Saramaki, "A systematic technique for designing approximately linear phase recursive digital filters," in *ISCAS '98. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, vol. 5, pp. 399–403, 1998.

[10] C. Rader, "dsp History - The rise and fall of recursive digital filters," *IEEE Signal Process. Mag.*, vol. 23, no. 6, pp. 46–49, 2006.