Area, Delay & Power Analysis of IIR Decimation Filter Using Merged Delay Transformation

Smitha G L
Dept. of EIE, MECS
DSCE
Bengaluru, India
smithapari@gmail.com

Meharunnisa S P
Assistant Professor, Dept. of EIE
DSCE
Bengaluru, India
meharbhakshi2013@gmail.com

***Abstract*— Area, delay and power consumption are the important constraints during designing an efficient decimation filter. In this paper Merged Delay Transformation (MDT) has been proposed for the realization of IIR decimation filters. By using MDT current output can be directly computed from M^{th} old output, so that the computational efficiency can be enhanced. By decomposing higher order IIR filters into parallel first order and second order, this transformation can be applied. The proposed method establishes better stability for co-efficient quantization and also reduces the requirement on processing clock, for samples, rate reduction. The number of multiplications are reduced , as compared to the conventional IIR filters. The performance of the proposed method is checked through MATLAB Simulink simulations and then it is implemented using System Generator.**

***Index Terms*—** *Decimation filter, Merged delay transformation, Mean square error, Peak signal to noise ratio*

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[5].

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.

This paper presents an improved MDT method for designing IIR filters, 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 filters, 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 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].

This paper is divided as follows. Section II provides a brief and general overview of MDT. Section III describes the software implementation of the proposed method. Section IV provides results and discussion. Section V concludes the paper.

II MERGED DELAY TRANSFORMATION (MDT)

A first-order recursive difference equation can be written as follows:

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

Here, $x[n]$ and $y[n]$ are the input and output data samples, respectively, where b_1 and a_0 real constants.

The values of $y[n-1], y[n-2], y[n-3], \ldots, y[n-M+1]$ can be replaced successively in (1) and the following general equation can be derived:

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

Equation (2) is called as merged delay transformation. Such a direct relationship is not possible for second- and higher order IIR filters. Equation (2) computes current output $y[n]$ from single Mth previous output and M inputs. The values of intermediate outputs are not required. 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 this figure, output $y[n]$ is fed back after passing through "M" number of unit delay elements. Merging M number of delay elements, output sampling rate can be reduced by M. 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 performs $M + 1$ multiplications as compared to $2M$ multiplications performed in the IIR decimation filter where all output samples are to be computed.

For higher-order IIR filters, direct application of merged delay transformation is not possible. 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 $r2$ and $p2$, 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 $M = 2$

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

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

$$A = p_{1r}^2 - p_{1i}^2 \qquad B = 2 p_{1r}\, p_{1i}$$

$$C = r_{1r} \qquad D = r_{1i} \qquad (5)$$

$$E = r_{1r}\, p_{1r} - r_{1i}\, p_{1i} \qquad F = r_{1i}\, p_{1r} + r_{1r}\, p_{1i}$$

$$Y_{1R}[n] = A\, y_{1R}[n-2] - B\, y_{1I}[n-2] + C\, x[n] + E\, x[n-1]$$

$$Y_{1I}[n] = A\, y_{1I}[n-2] + B\, y_{1R}[n-2] + D\, x[n] + F\, x[n-1]$$

Here, $Y_{1R}[n]$ and $Y_{1I}[n]$ are the real and imaginary parts 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

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

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

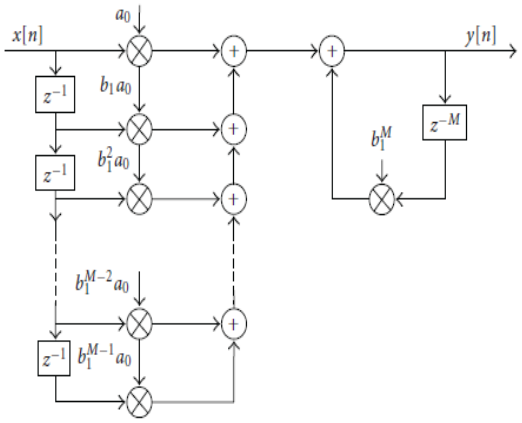

Fig 1: Realization of MDT based M fold decimation in first order IIR filter

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:

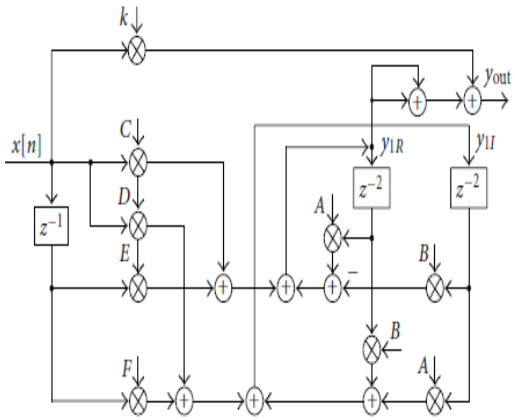

Fig 2: Realization of second order IIR decimation filter with M=2

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

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 N^{th} order IIR decimation filter with any integer decimation factor M can be realized.

III SOFTWARE IMPLEMENTATION

This section describes the software implementation of decimation filter using System generator and key steps for designing are shown fig 3.

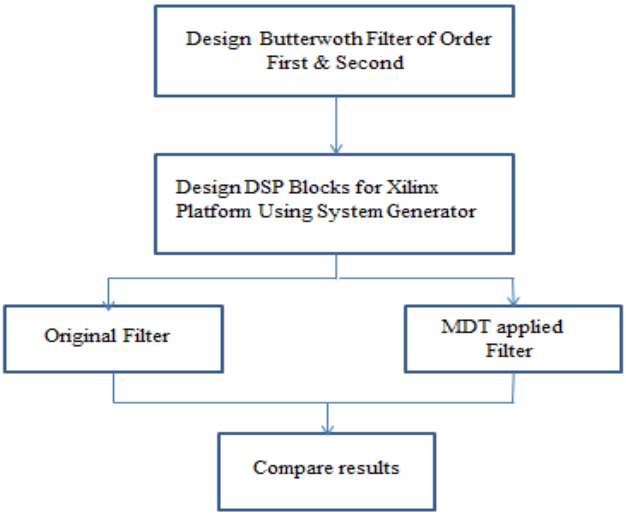

Fig 3: Key steps for software implementation

The IIR filter used in this research is of Butterworth type. The designing parameters of the filter are Sampling frequency 44.2kHz, Cut-off frequency 20kHz, Decimation factor 4.The magnitude response of the butterworth filter decreases with increase in frequency from zero infinity[3]. The width of the transition band is more in butterworth filters are used in applications where maximum passband flatness is required. The output of proposed IIR is compared with actual filter equation output to find out the error.

IV RESULTS AND DISCUSSIONS

The proposed technique is verified using matlab simulations for IIR butterworth filters. Simulink is a graphical extension to Matlab for modeling and simulation of systems. Simulink is integrated with matlab and data can be easily transferred between the programs. The DSP blocks are converted into Xilinx blocks by using system generator. Fig 4 and 5 shows First order filter design without and with MDT using Xilinx blocks. Table 1 and 2 shows the implementation results of First order filter without and with MDT . Fig 6 and 7 shows Second order filter design without and with MDT using Xilinx blocks. Table 3 and 4shows the implementation results of Second order order filter without and with MDT. Fig8 shows the design of First order Chebyshev type I filter using Xilinx blocks and table 5 shows its implementation results.

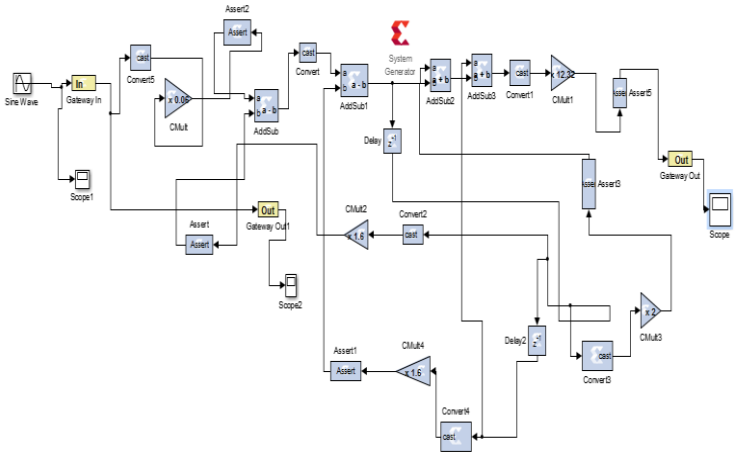

Fig 4: First order filter design using Xilinx blocks

Table 1: Synthesis results of First order filter without MDT

Parameters	Values
Number of Multipliers used	1
Adders/Subtractors	4
Registers	8205
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
Total power	0.158W

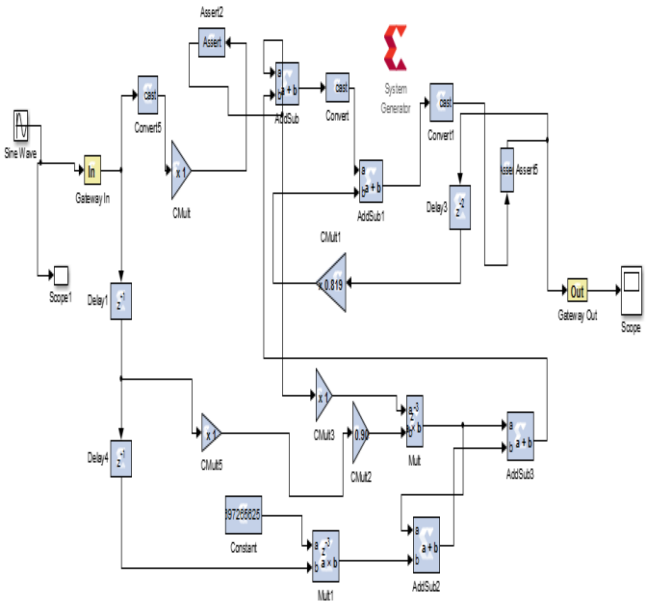

Fig 5: First order filter design with MDT using Xilinx blocks

Table 2: Synthesis results of First order filter with MDT

Parameters	Values
Adders/Subtractors	4
Registers	61
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
Total power	0.160W

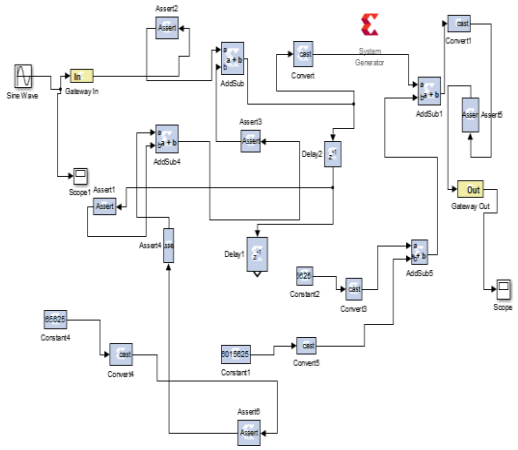

Fig 6: Second order filter design using Xilinx blocks

Table 3: Synthesis results of Second order filter without MDT

Parameters	Values
Adders/Subtractors	4
Registers	4109
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
Total Power	0.121W

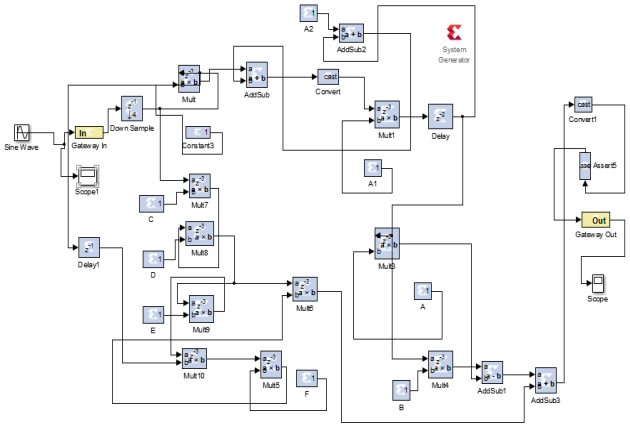

Fig 7: Second order filter design with MDT using Xilinx blocks

Table 4: Synthesis results of Second order filter with MDT

Parameters	Values
Multipliers	2
Adders/Subtractors	4
Counters	1
Registers	88
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
Total Power	0.129W

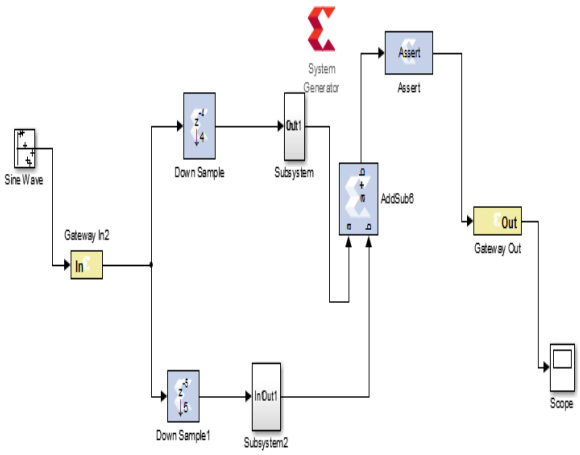

Fig 8: Design of First order Chebyshev type I filter using Xilinx blocks

Table 5: Synthesis results of First order Chebyshev type I filter without MDT

Parameters	Values
Counters	2
Registers	563
Number of Slice Register	563 out of 184305 0%
Number of Fully used LUT FF pairs	68 out of 568 11%
Number of Bonded IOB's	61 out of 396 12%
Maximum frequency	387.621MHz
Maximum Combinational path delay	3.150ns
Total Power	0.199W

V CONCLUSION

A novel technique to efficiently realize IIR decimation filter is presented. First-order recursive equation is transformed by merged delay transformation. Drawbacks associated with cascade of integrators, down samplers, and differentiators are avoided. The DSP blocks are designed into

Xilinx blocks by using system generator. Area, delay and power consumption of Butterworth filter and Chebyshev type I filter are obtained. Overall performance of the filter can be improved by using Merged Delay Transformation.

REFERENCES

[1] 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.

[2] H. Aboushady, Y. Dumonteix, M.-M. Lou¨erat, and H. Mehrez "Efficient polyphase decomposition of comb decimation filters in ΣΔ analog-to-digital converters," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 48, no. 10, pp. 898–903, 2001.

[3] M. G. Bellanger, G. Bonnerot, and M. Coudreuse, "Digital filtering by polyphase network: application to sample-rate alteration and filter banks," *IEEE Transactions on Acoustics,Speech,and Signal Processing*, vol. 24, no. 2, pp. 109–114, 1976.

[4] U. Farooq, H. Jamal, and S. A. Khan, "Realization of IIR decimation filters based on merged delay transformation," *Res. Lett. Commun*., vol. 2007, no. 2, pp. 1–3, 2007.

[5] 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.

[6] A. V., Oppenheim, R. W. Buck, and J. R., Discrete-Time Signal Processing, Second. Pearson Education Asia Pte Ltd. India, 2000.

[7] C. M. Rader, "The rise and fall of recursive digital filters," IEEE Signal Processing Magazine, vol. 23, no. 6, pp. 46–49, 2006.

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

[9] 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.