

Multistage implementation of multirate CIC filters

Anil Singh, Poonam Singhal and Rajeev Ratan¹
 ECE, DCRUST, Campus Murthal, Sonapat, Haryana-131001, India
¹ECE, MVN Education City, Palwal- 121102, Haryana, India
 duhan1986123@gmail.com

Abstract

In this paper a multistage multirate CIC filter has been compared with a FIR Filter. FIR filters are widely used in implementation of Digital Filters, but with some disadvantages like higher number of computational coefficients requirement and less computational speed. In this work we have shown that CIC filters are better choice than the single stage CIC or FIR Filters for multirate filtering structures and further improvement can be made by cascading different stages of CIC Filters.

Keywords: CIC Filters, FIR Filters, Sample Rate Conversion (SRC), Decimator, Interpolators, MATLAB

Introduction

The Digital filter can be used to remove undesired signal, such as noise and hence to extract valuable information of the signal (Fig.1). Basic Fourier transform theory states that the linear convolution of two sequences in the time domain is the same as multiplication of two corresponding spectral sequences in the frequency domain. Filtering is in essence the multiplication of the signal spectrum by the frequency domain impulse response of the filter. For an ideal lowpass filter, the pass band part of the signal spectrum is multiplied by one and the stopband part of the signal by zero. A digital filter takes a digital input, gives a digital output, and consists of digital components. In a typical digital filtering application, software running on a digital signal processor (DSP)

Fig.1. Block of a basic filter

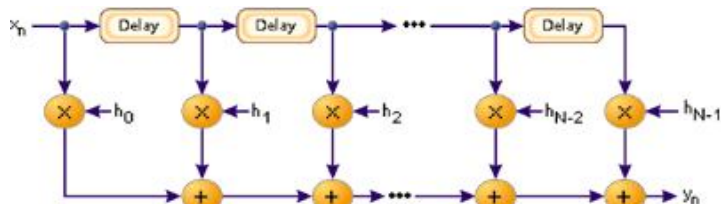


reads input samples from an A/D converter, performs the mathematical manipulations dictated by theory for the required filter type, and outputs the result via a D/A converter (Ricardo, 2008). An analog filter, by contrast, operates directly on the analog inputs and is built entirely with analog components, such as resistors, capacitors, and inductors.

FIR filters

A finite impulse response (FIR) filter is a filter structure that can be used to implement almost any sort of frequency response digitally. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the filter's output. Fig.2 shows the basic

Fig.2. The logical structure of an FIR filter



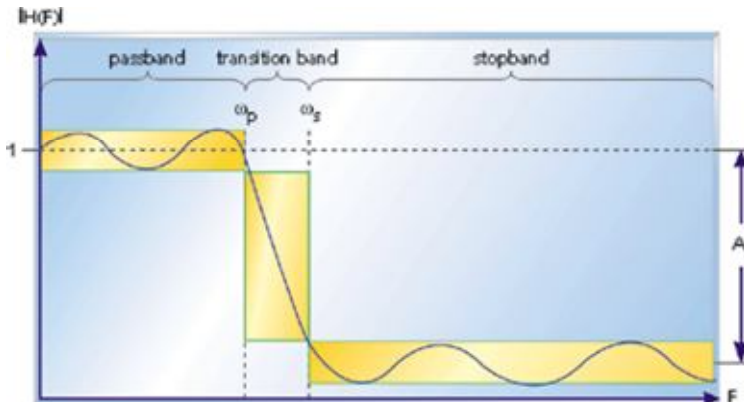
block diagram for an FIR filter of length N. The delays result in operating on prior input samples. The h_k values are the coefficients used for multiplication, so that the output at time n is the summation of all the delayed samples multiplied by the appropriate coefficients. The process of selecting the filter's length and coefficients is called filter design. The goal is to set those parameters such that certain desired stopband and passband parameters will result from running the filter. Most engineers utilize a program such as MATLAB to do their filter design. But whatever tool is used, the results of the design effort should be the same: The longer the filter (more taps), the more finely the response can be tuned (Meyer-Bease, 2007).

Frequency response of FIR Filter

Simple filters are usually defined by their responses to the individual frequency components that constitute the input signal. There are three different types of responses. A filter's response to different frequencies is characterized as passband, transition band, or stopband. The passband response is the filter's effect on frequency components that are passed through (mostly) unchanged. Frequencies within a filter's stopband are, by contrast, highly attenuated. The transition band represents frequencies in the middle, which may receive some attenuation but are not removed completely from the output signal.

Fig.3 shows the frequency response of a lowpass filter, ω_p is the passband ending frequency, ω_s is the stopband beginning frequency, and A_s is the amount of attenuation in the stopband. Frequencies between ω_p and ω_s fall within the transition band and are attenuated to some lesser degree. Given these individual filter parameters, one of numerous filter design software packages can generate the required signal processing equations and coefficients for implementation on a DSP. Before we can talk about specific implementations, however, some additional terms need to be introduced (Hentschel & Fettweis, 2000). Ripple is usually specified as a peak-to-peak level in decibels. It describes how little or how much the filter's amplitude varies within a band.

Fig.3. The response of a lowpass filter to various input frequencies



Smaller amounts of ripple represent more consistent response and are generally preferable. Transition bandwidth describes how quickly a filter transitions from a passband to a stopband, or vice versa. The more rapid this transition, the higher the transition bandwidth; and the more difficult the filter is to achieve. Though an almost instantaneous transition to full attenuation is typically desired, real-world filters don't often have such ideal frequency response curves. There is, however, a tradeoff between ripple and transition bandwidth.

Fig.4. Block diagram of three-stage CIC decimation and interpolation filters



CIC Decimation Filter



CIC Interpolation Filter



CIC filters

A Cascaded Integrator Comb (CIC) filter is a special class of linear phase, finite impulse response (FIR) filter. CIC filters do not require multipliers and use a limited amount of storage. Therefore, CIC filters are more efficient than conventional FIR filters, especially in fixed-point applications (Hogenauer,

1981). CIC filters generally find use in multirate systems with large sampling frequency conversion factors, such as digital down converters (DDC) and digital up converters (DUC) in communication systems (Babic & Renfors, 2005). CIC filter also finds applications in interpolation and decimation. Unlike most FIR filters, it has a decimator or interpolator built into the architecture. The system function for the composite CIC filter referenced to the high sampling rate, f_s is:

$$H(z) = \left[\sum_{k=0}^{RM-1} z^{-k} \right]^N = \left(\frac{1 - z^{-RM}}{1 - z^{-1}} \right)^N$$

Where:

R = decimation or interpolation ratio

M = number of samples per stage (usually 1 but sometimes 2)

N = number of stages in filter

Characteristics of CIC Filters are that it gives linear phase response and utilizes only delay and addition and subtraction; that is, it requires no multiplication operations (Ljiljana, 2009).

CIC Filter Structure

The basic elements of a CIC filter are integrator filters and comb filters, as shown in Fig. 4.

The CIC filter's difference equation is:

$$y(n) = x(n) - x(n - D) + y(n - 1)$$

Building a CIC filter

In CIC filters, the comb section can precede, or follow, the integrator section. It's sensible, however, to put the comb section on the side of the filter operating at the lower sample rate to reduce the storage requirements in the delay (Crochiere & Rabiner, 1983). Swapping the comb filters with the rate-change operations results in the most common implementation of CIC filters. Notice the decimation filter's comb section now has a delay length (differential delay) of $N = D/R$. That's because an N -sample delay after decimation by R is equivalent to a D -sample delay before decimation by R . Likewise for the interpolation filter; an N -sample delay before interpolation by R is equivalent to a D -sample delay after interpolation by R which yields two major benefits: first, the comb

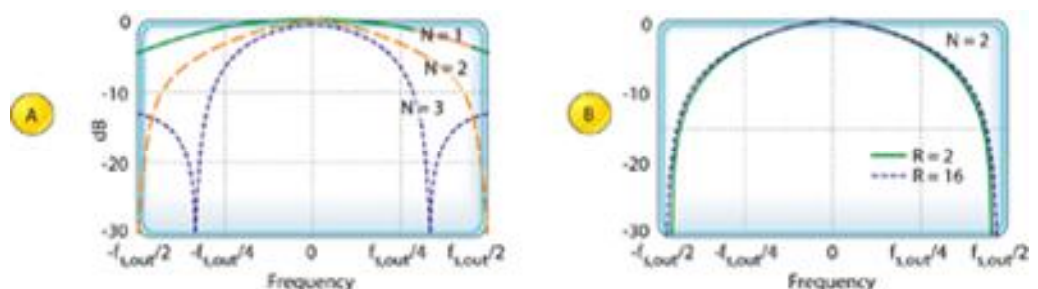
Fig. 5. CIC decimation filter responses: for various values of differential delay N , when $R = 8$; for two decimation factors when $N = 2$


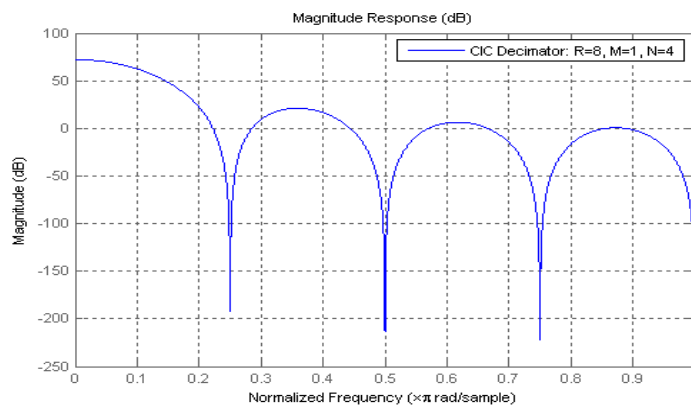
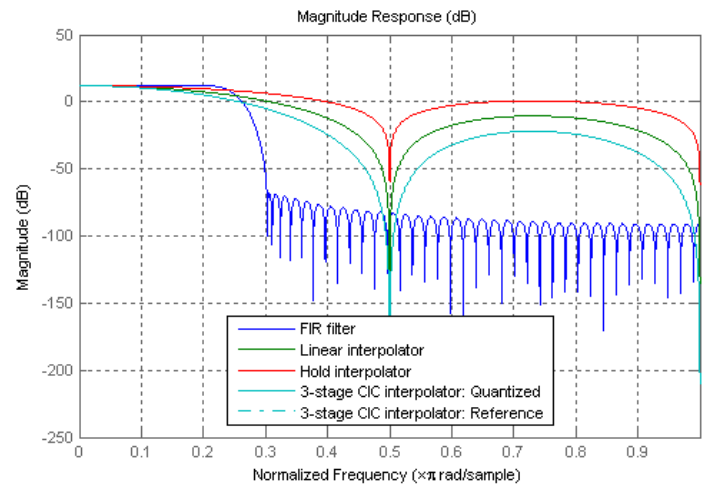
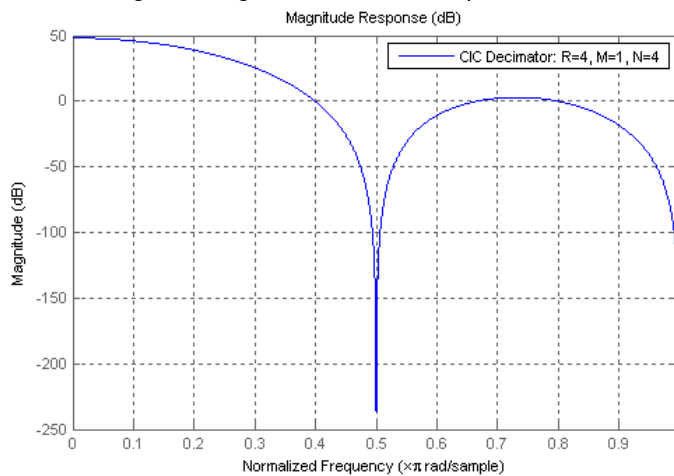
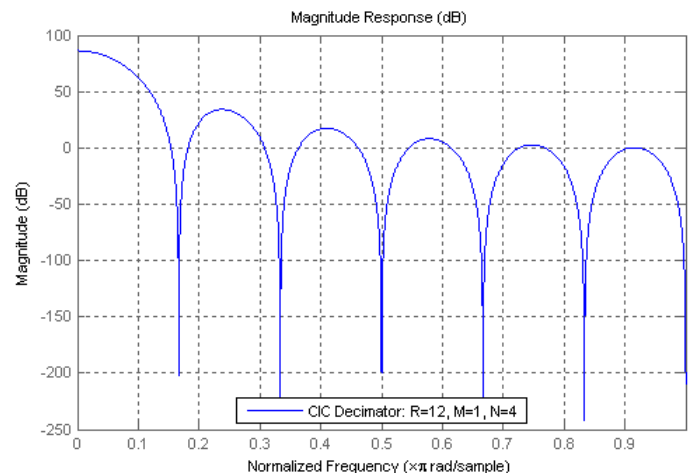
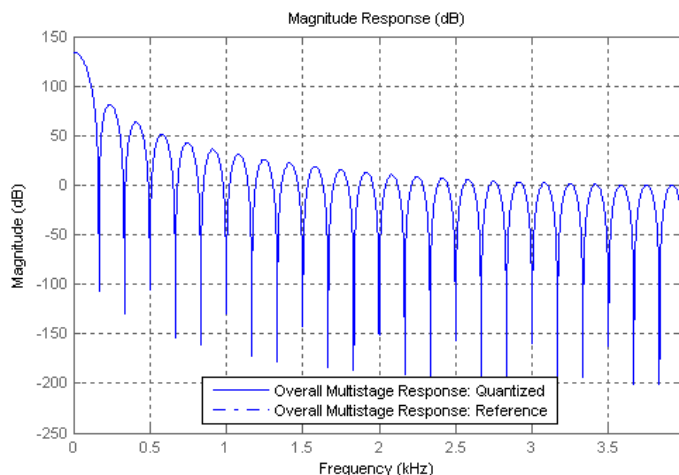
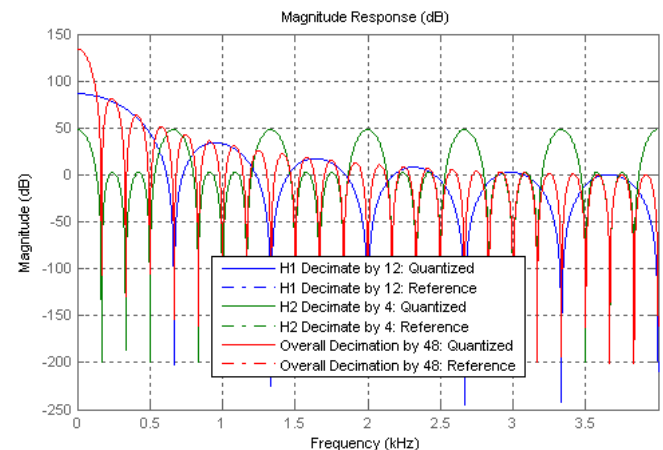
Fig. 6. 4 stage CIC Decimator response with $R=8$


Fig. 7. Comparison of CIC and FIR filter


Fig. 9. 4 stage CIC Decimator response with $R=4$

Fig. 8. 4 stage CIC Decimator response with $R=12$

Fig.10. Overall multistage response of CIC filter with $R=12$ & $R=4$

Fig.11. Overall Decimation by $R=12$, $R=4$ and their cascade


section's new differential delay is decreased to $N = D/R$ reducing data storage requirements; second, the comb section now operates at a reduced clock rate. Both of these effects reduce hardware power consumption (Vesma, 1999).

The comb section's differential delay design parameter N is typically 1 or 2 for high sample-rate ratios as is often used in up/down-converters. N effectively sets the number of nulls in the frequency response of a decimation filter, as shown in Fig. 5a. An important characteristic of a CIC decimator is that the shape of the filter response changes very little, as shown in Fig. 5b, as a function of the decimation ratio. For values of R larger than roughly 16, the change in the filter shape is negligible. This allows the same compensation FIR filter to be used for variable-decimation ratio systems (Farrow, 1998), (Babic *et al.*, 2002).

The CIC filter suffers from register overflow because of the unity feedback at each integrator stage. The overflow is of no consequence as long as the two conditions are met: 1. the range of the number system is greater than or equal to the maximum value expected at the output, and 2. the filter is implemented with two's complement (non-saturating) arithmetic.

Materials and methods

The CIC Filters have been compared with their counterpart FIR Filter. All the algorithms have been developed in MATLAB 7.0 and the procedure adopted is presented.

Single-rate filter design

Consider the design of a lowpass filter that preserves all frequencies below 70 Hz, removes all frequencies above 80 Hz, and runs at an input sample rate of 8 kHz. The filter cutoff frequency is therefore 80 Hz, or in terms of normalized frequency, $80/4000$ or 0.02 , where the normalization is done with respect to half the sample rate. In order to completely specify a CIC filter, we must also specify the amount of desired maximum passband attenuation (for this example, 0.15 dB or 0.01 linear) and minimum stopband attenuation (80 dB or 10^{-4} linear) that we wish to achieve.

Multirate narrowband filter design

The bandwidth of the signal output from the single-rate filter designed above is quite narrow with respect to the sample rate. The minimum sample rate necessary to reconstruct a signal with 80 Hz bandwidth is 160 Hz, and this requires only $160/8000$ or $1/50$ the rate at which the output is presently sampled. This implies that we could *decimate* the output by 50 (i.e., remove 49 out of every 50 samples) without aliasing the output signal. Pursuing this further, the cascade of a CIC filter with a decimator is most efficiently performed with a *poly-phase CIC decimation filter* which reduces the computational rate by the decimation factor, M . This achievement is most easily understood by considering that only one out of every M input samples is output by the decimator; all others are "thrown away." Thus, one could consider

computing one out of every M output samples. The polyphase CIC decimation filter is structured in such a way as to balance the computation of this one output sample over M consecutive inputs, thereby obtaining the noted reduction in computational rate. However, decimation is only half the story. For general multirate filtering, we assume that the output rate is equal to the input rate. Thus, we need to interpolate the output of the polyphase CIC decimation filter back up to the original input rate. This may be performed by the cascade of an upsample operation (in which $L-1$ zeros are inserted after every sample) followed by an CIC filter which is most efficiently performed with a *polyphase CIC interpolation filter*. Similar to the decimator, this implementation achieves the interpolation while reducing the computational rate by the factor L . This outcome is most easily understood by considering that only one out of every L interpolated samples is non-zero prior to entering the interpolation filter. This means we may effectively skip the filter computation for the zero-valued samples, achieving a reduction in the rate of computation.

Single-stage multirate filter design

The cascade of polyphase CIC decimation and interpolation filters forms an efficient, multirate filter. A single-stage multirate filter doesn't ease the filter length requirement, but it reduces the computational load. Returning to the lowpass filter example, we choose to decimate the filter output by a factor of 48 (slightly smaller than the maximum decimation factor of $8000/160$, or 50 as a margin of safety).

Multistage multirate filter design

Additional design efficiency can be achieved by using several cascaded multirate stages, which reduces the total number of filter coefficients. For example, instead of implementing a design based on a single decimation factor of 48 , we could consider two stages of decimation using 12 and 4 , or 24 and 2 , or any other combination of factors that multiply to 48 . The CICst stage decimates by a factor of 12 ; thus, the corresponding CICst-stage filter must cut off before $1/12$ of the normalized sampling frequency (approximately 0.083) to prevent aliasing. Note that this requirement is less constraining than our overall specification, which requires an 80 Hz cutoff. (Recall that an 80 Hz cutoff translates to $80/4000$ or 0.02 in normalized frequency, and is smaller and thus more constraining than 0.083 .) This can lead to a shorter (CICst stage) filter design. The second stage filter is then responsible for attaining our final filter requirements (70 Hz passband, 80 Hz stopband), but now operates at $1/12$ the original sample rate. Thus, the design constraints placed on the second stage are eased, resulting in a shorter second-stage filter. However, since we are now cascading *two* filters, the passband ripple requirement must be halved for each filter stage, and reducing the ripple specification tends to increase overall filter length. Choice of decimation factor for each stage allows a degree of optimization across these design factors, and

can lead to a design that is more efficient than single-stage designs.

Implementation and results

Fig.6 represents the response of a simple CIC Decimator ($R=8$, $M=1$, $N=4$). Now the response of a CIC Filter is compared with the FIR filter, which is shown as Fig.7. A CIC Decimator with $R=12$ is shown as Fig.8. A 4 stage CIC Filter is shown as Fig.9. Now a multistage CIC filter is designed by cascading a four stage CIC filter and 12 stage CIC filter and the response of cascade of the above 2 filters is shown in Fig.10. The comparison of single stage and multistage CIC filter is presented in Fig.11.

Table 1. Comparative summary of single stage, two stage and multistage design

Design technique	Total CIC coefficients	Computational rate (KMACs/sec)
Single rate	2585	20680
Two stage multirate	1752	584
Multis stage multirate	390	101

Conclusion

CIC filters are better choice than the single stage CIC or FIR Filters for multirate filtering structures and their response can be further improved by cascading different stages of CIC Filters. If a CIC filter (Decimation Ratio 12) is cascaded with the CIC filter (Decimation Ratio 4), the overall response of CIC filter is 48 times decimated. Thus, it results in drastic cost reduction of designing CIC filters. A comparative summary of the single stage, two stage and multistage design examples are shown in Table 1.

References

1. Babic D and Renfors M (2005) Power efficient structures for conversion between arbitrary sampling rates. *IEEE Signal Processing Lett.* 12(1), 1-4.
2. Babic D, Saramaki T and Renfors M (2002) Sampling rate conversion between arbitrary sampling rates using polynomial-based interpolation filter. The 2nd Intl. Workshop on Spectral Methods & Multirate Signal Processing. Toulouse, France. pp: 57-64.
3. Crochiere RE and Rabiner LR (1983) Multirate digital signal processing. Englewood Cliffs, NJ: Prentice-Hall.
4. Farrow C (1998) A continuously variable digital delay element. Proc. IEEE Intl. Symp. on Circuits and Systems (ISCAS88). pp: 2642-2645.
5. Hentschel T and Fettweis G (2000) Continuous-time digital filter for sample rate conversion in reconfigurable radio terminals. Proc. of European Wireless Conf. pp: 55-59.
6. Hogenauer EB (1981) An economical class of digital filters for decimation and interpolation. *IEEE Transact. on Acoustic, Speech and Signal Proc.* ASSP29(2). pp:155-162.
7. Ljiljana M (2009) Multirate filtering for digital signal processing: MATLAB applications. Hershey, PA: Information Science Reference.

Meyer-Bease U (2007) Digital signal processing with field programmable gate arrays. Springer, Third Edition.

Ricardo L (2008) Digital Filters with MATLAB. [Online]. Domain Site: www.mathworks.com/matlabcentral/fileexchange/19880.

Vesma J (1999) Optimization and applications of polynomial based interpolation filters. *Doctoral Thesis, Tampere University of Technology*, Publ. pp: 254.