

Performance Analysis of Compensated CIC Filter in Efficient Computing Using Signed Digit Number System

Vishal Awasthi¹ & Krishna Raj²

¹Member, IETE, ECE Deptt., CSJM. University, Kanpur, U.P., India

²Fellow Member, IETE, ECE Deptt., H.B.T.I., Kanpur, U.P., India

E-mails: awasthiv@rediffmail.com, kraj_biet@yahoo.com

Abstract: This paper presents efficient compensated Cascaded Integrator Comb (CIC) decimation filter to improve the passband of interest using redundant signed digit arithmetic. In redundant representations, addition can be carried out in a constant time independent of the word length of the operands. Most of the research in the last decades has concentrated on reducing the delay of addition. A hybrid adder can add an unsigned number to a signed-digit number and hence their efficient performance greatly determines the quality of the final output of the concerned circuit. The proposed structure consists of compensated section cascade with CIC decimation sections, each down-sampled by a specific down-sampling factor. The number of sections depends on the decimation factor of the original comb decimator and the number of cascaded filters for different stages. The magnitude response is improved by using FIR prefilters. The coefficients of the compensator filter are presented in a signed digits (SD) form, and can be implemented using only adders and shifts. Consequently, the resulting filter is a multiplier free filter and exhibits a high attenuation in the stopband, as well as a low passband droop.

Keywords: Multirate filtering, Decimation filter, Compensator, CIC filter, Redundant signed digit arithmetic, Carry free addition, Fast adders.

1. INTRODUCTION

Multirate signal processing techniques are widely used in many areas of modern engineering such as communications, image processing, digital audio, and multimedia. The main advantage of a multirate system is the substantial decrease of computational complexity, and consequently, the cost reduction. The computational efficiency of multirate algorithms is based on the ability to use simultaneously different sampling rates in the different parts of the system. Multirate DSP is the process of converting data sampled at one rate (F_{s1}) to data sampled at another rate (F_{s2}). The process of decimation (if $F_{s1} > F_{s2}$) is used to transform 1-bit data stream which has very high sampling rate into the required bit resolution signal with lower sampling rate. Thus decimation has two functions to perform simultaneously, one is averaging filter function and the other is rate reduction. Hogenauer [1] presented a commonly used decimation filter known as cascaded-integrator-comb (CIC) filter that consists of cascaded integrators and differentiators section, separated by a down-sampler. A pictorial representation of the decimation process is shown in below figure:

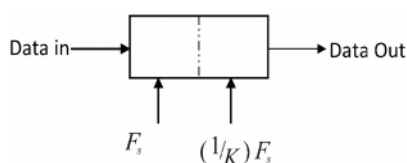


Figure 1: Conceptual View of a Decimation Filter

For decimation by a factor of K , the original data must reside in a bandwidth given by $F_s / (2K)$, where F_s is the rate at which the original data was sampled. Thus, if the original data contains valid information in the portion of the spectrum beyond $F_s / (2K)$, decimation is not possible. The three basic tasks performed by this decimation filter are:

Removing quantization noise: Reducing the base band quantization noise is equivalent to increasing the effective resolution of the digital output.

Decimation (sample rate reduction): Desirable to bring the sampling rate down to the Nyquist rate which minimizes the amount of information for subsequent transmission, storage, or digital signal processing.

Anti-aliasing: Due to the high sampling rate of the modulator the digital decimation filter must perform computationally intensive signal processing algorithms in real time. Higher order modulators produce highly shaped noise and hence the decimation filter should be very efficient to remove this excess quantization noise and to regain the original characteristics of the signal in the base band.

This filter has very low complexity but exhibits two main problems: (a) The integrator section works at the higher input data rate while the differentiator section operates at the lower data rate and therefore require higher chip area with higher power dissipation and (b) A high passband droop and a low stopband attenuation in its magnitude characteristic.

To overcome these problems, various methods have been introduced that uses non-recursive structure of a comb filter to reduce the power consumption as well as to increase the circuit speed [2]. Kwentus *et. al.* [3] outlined a method that uses the sharpening technique to decrease the passband droop and to increase the stopband attenuation but it requires sharpening to be performed at the high input rate and hence resulting higher power consumption. Jovanovic *et. al.* [4] and Presti L. L. [5] discussed some methods to attain the desired low stopband attenuation by allowing the sharpening section to operate at the lower rate with the cost of the introduction of two multipliers working at a high rate. In [6] a new multistage comb-rotated sinc (RS) decimator is introduced which permit both multipliers to work at the lower rate, with no filtering at the high input rate. A cascading method [7] uses to reduce the passband droop by cascading sharpened comb filter with the Rotated sinc (RS) filter. G. Jovanovic, Dolecek *et. al.* [8] is introduced a new multiplier-free CIC-cosine decimation filter with no filtering at the high input rate. In [9], G. Javanovic Dolecek *et. al.* proposed a efficient modification of the CIC cosine decimation filter using canonical signed digits (CSD). In its proposed structure, a second order compensator filter is introduced at low rate to improve the passband and then the compensator filter coefficient are presented in a canonical signed digits (CSD) form but limiting with speed.

Arithmetic operators designed using redundant number system achieves considerable speed improvement compared with operators designed using conventional number system. The execution speed of an arithmetic operation is directly related to chosen architecture and the number system employed to implement architecture. These redundant arithmetic operations employ a signed digit representation where each digit of a number can be positive or negative. The use of a redundant number systems leads to carry free addition, where the carry propagates only through two or three stages, independent of the word length. In recent years carry free arithmetic operations (such as multiplication, division, square root etc.) employing redundant number systems [10] have received considerable attention. Ripple carry adder (RCA) had a long carry propagation paths extending from the least significant bit to the most significant bit position. Its computation time is directly proportional to the word length of the operands. Carry select addition scheme reduces the computation time by pre-computing some positions of the results for all possible carry bit values (i.e. 1 & 0) and using the carry from the previous bit position (after it becomes available) to choose the proper result. Weinberger and Smith [11] have proposed Carry Look-ahead adder scheme to speed-up additions by using parallelism to propagate carries. Robertson [12] and Avizienis [13] suggested a set of arithmetic rules for redundant signed digit numbers. Takaji and Yazima [14] proposed a high-speed algorithm suitable for VLSI implementation using RBSD numbers. A carry select

addition technique was presented [15] by O. J. Bedrij. Changes at all levels are required to have a higher performance design since the clock frequency and power consumption doubles every two years.

The efforts to increase the speed of computing depends on the choice of logic design style (arrangement of gates and number system representation which is capable of elimination length of carry propagation chains during addition. In this paper, we have proposed a simple and efficient compensated CIC decimation filter using signed digit number arithmetic to achieve fast decimation response and desired low passband droop in comparison to CIC filter.

2. CASCADED INTEGRATOR COMB (CIC) FILTER

Multirate multistage signal processing is important in modern telecommunication applications. Cascaded integrator-comb (CIC), or Hogenauer filters, are multirate filters used for realizing large sample rate changes in digital systems. CIC filter structure, suggested by E. B. Hogenauer, consist of an integrator block working at the oversampled frequency F_s , a clock divider for rate reduction and a differentiator block working at F_s/k , where k is the decimation ratio. CIC filters are multiplierless structures, consisting of only adders and delay elements which is a great advantage when aiming at low power consumption. The frequency response of CIC filters has a poor in band frequency response and are mostly used to reduce the the decimation factor by large ratio while suitable response sharpening filters work at the reduced rate to get the desired characteristics. CIC filters are frequently used in digital down-converters (DDCs) and digital up-converters.

The cascaded integrator-comb (CIC) filter is a class of hardware-efficient linear phase finite impulse response (FIR) digital filters, consists of an equal number of stages of ideal integrator filters and comb filters. Its frequency response may be tuned by selecting the appropriate number of cascaded integrator and comb filter pairs. However, the disadvantage of a CIC filter is that its pass band is not flat, which is undesirable in many applications. Fortunately, this problem can be alleviated by a compensation filter. CIC filters achieve sampling rate decrease (decimation) and sampling rate increase (interpolation) without using multipliers. The CIC filter first performs the averaging operation then follows it with the decimation. The transfer function of the CIC filter in z -domain is given in equation (1).

$$H[z] = H[z]_I^p \cdot H[z]_C^p = \frac{(1 - z^{-K})^p}{(1 - z^{-1})^p} = \left(\frac{1 - z^{-K}}{1 - z^{-1}} \right)^p \quad (1)$$

In equation (1), K is the oversampling ratio and p is the order of the filter. The numerator $(1 - z^{-K})^p$ represents the transfer function of a differentiator and the denominator $1/(1 - z^{-1})^p$ indicates the transfer function of an integrator.

A simple block diagram of a first order CIC filter is shown in Figure 2. In a CIC filter, the integrators operate at high sampling frequency (f_s), and the comb filters operate at low frequency ($\frac{f_s}{K}$). The clock divider circuit divides the oversampling clock signal by the oversampling ratio, K after the integrator stage. By operating the differentiator at lower frequencies, a reduction in the power consumption is achieved.

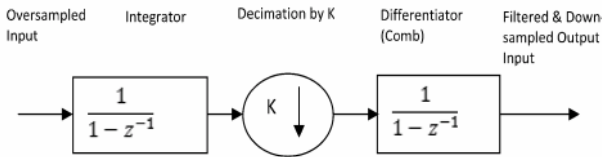


Figure 2: Block Diagram of First Order CIC Filter

3. DIFFERENT COMPENSATION TECHNIQUES

When the number of stages is large, CIC filter introduces a droop in the passband and this droop is dependent on the CIC decimation ratio. To overcome the magnitude droop, a FIR filter that has a magnitude response that is the inverse of the CIC filter can be applied to achieve frequency response correction. Usually the CIC filter is followed by a second decimating lowpass filter stage and its decimation ratio is significantly smaller than that of the CIC stage (typically 16 or less). The decimation factor of this second stage will determine the frequency at which the worst-case aliasing will occur and will also determine the edge frequency of the passband of interest, where the worst-case passband distortion will occur. Several schemes have been proposed to design the compensation of CIC filter's passband droop, mainly in the narrow pass band. The motivation behind the compensation methods is to appropriately modify the original CIC characteristic in the pass band such that the compensator filter has as low complexity as possible. Various methods used for compensation of CIC decimation filter are as follows:

3.1. CIC Roll-Off Compensation Filter

The CIC roll-off compensation filter is like a channel selective filter with symmetric characteristics in frequency response. This method compensated the roll off of the CIC filter in pass band by letting the CIC filter followed by a symmetric FIR filter with a minimum order. CIC roll off compensation filter can be written as:

$$c(n) = \frac{-v}{1-2v} \delta(n+1) + \frac{1}{1-2v} \delta(n) + \frac{-v}{1-2v} \delta(n-1) \quad (2)$$

Where $\left[\frac{-v}{1-2v}, \frac{1}{1-2v}, \frac{-v}{1-2v} \right]$ are the compensation filter coefficient and $C(w) = \frac{1-2v \cos w}{1-2v}$ is its frequency response.

The performance of the compensation filter depends on the value of v , which is obtained by minimizing the corresponding error function. $C(w)$ can work as a roll off compensation filter as it shows opposite frequency characteristics of CIC filter in frequency domain. Let the frequency response of the CIC filter as $f(w)$, P_e is pass band edge of the received signal and the frequency response of an ideal filter as $D(w)$, then error function is defined by-

$$E_{uv(v)} = \int_0^{P_e} (D(w) - C(w)f(w))^2 dw \quad (3)$$

Roll off phenomenon of the CIC filter can exactly compensated if the frequency response characteristics of the received signal are used as a weighting function. It slightly improves the flatness of the pass band. This method focused on compensating the slope of the pass band, which is already fixed in the digital receiver, by letting the CIC filter followed by the compensation filter with a minimum computational load.

3.2. Compensated CIC-Cosine Decimation Filter

This filter is a modified and efficient version of the CIC Cosine decimation filter. In order to improve the passband of interest of the overall filter, a second order compensator filter is introduced at low rate. The compensator filter coefficients are presented in a canonical signed digits (CSD) form, and can be implemented using only adders and shifts. Transfer function and magnitude response of compensation filter is given by-

$$H_{comp}(Z^M) = v + uz^{-M} + vz^{-2K} \quad (4)$$

$$\left| H_{comp}(e^{jKw}) \right| = |2v \cos(Kw) + u| \quad (5)$$

Where v & u real valued constant and K are is decimation factor. Worst pass band distortion occurs at $w = 0$ & $w = w_c$ where $w_c = \frac{\pi}{KR}$ and R is the decimation factor of next decimation stage. In order to compensate the pass band droop (δ_c) at the frequency w_c then $2v + u = 1$ and $2\cos(Kw_c) + u = \frac{1}{\delta_c} \cdot v$ and u can be calculate by-

$$\begin{bmatrix} v \\ u \end{bmatrix} = \begin{bmatrix} -1 & 1 \\ \frac{2(\cos(Kw_c) - 1)}{(\cos(Kw_c) - 1)} & \frac{2(\cos(Kw_c) - 1)}{(\cos(Kw_c) - 1)} \end{bmatrix} \begin{bmatrix} 1 \\ \delta_{comp} \end{bmatrix} \quad (6)$$

$$v = \frac{-1 + \delta_{comp}}{2(\cos(Kw_c) - 1)} \quad (7)$$

$$u = \frac{\cos(Mw_c) - \delta_{comp}}{\cos(Kw_c) - 1} \quad (8)$$

Where $\delta_{comp} = \frac{1}{\delta_c}$ and δ_c should be less than 0.01 dB. If the passband droop is within the desired limit then the transfer function of compensated filter can be represented in canonical signed digit (CSD) as:

$$H_{comp-CSD}(Z^K) = x_{CSD} + y_{CSD}z^{-K} + x_{CSD}z^{-2K} \quad (9)$$

Where x_{CSD} and y_{CSD} are the CSD representations of the quantized coefficients x_q and y_q of the proposed compensation filter that satisfied the relation $2x_q + y_q = 1$ and given by-

$$x_q = 2^{-p} \left(\frac{v}{2^{-p}} \right) \quad (10)$$

$$y_q = 2^{-p} \left(\frac{u}{2^{-p}} \right) \quad (11)$$

The procedure is continued until the desired Pass band compensation is obtained. There is a trade off between the desired compensation of the pass band droop and filter coefficients can control the desired pass band droop of the overall decimation filter.

4. TECHNOLOGY DEVELOPMENT OF FAST ADDERS AND THEIR ARITHMETIC

A fast and energy-efficient adder is essential for a high-performance processor. In computing signed digit number representation is required to encode negative numbers in binary number systems. In mathematics, negative numbers in any base are represented by prefixing them with a – sign. However, in computer hardware, numbers are represented in binary only without extra symbols, requiring a method of encoding the minus sign. The signed digit number representation makes it possible to perform addition without carry propagation chains that are used to speed up arithmetic operations. Ripple Carry adder is the first and the most fundamental adder that is capable of performing binary number additions.

4.1. Ripple Carry Adders (RCA)

The ripple carry adder is composed of a chain of full adders with length n, where n is the length of the input operands. The most straightforward implementation of a parallel adder for two-operands $A_{n-1}, A_{n-2}, \dots, A_0$ and $B_{n-1}, B_{n-2}, \dots, B_0$ is through the use of n basic units called full adders. At the i^{th} bit position, the i^{th} bits of operands A and B and a carry signal C_i from the preceding adder stage are used to generate the i^{th} bit of the sum, S_i , and a carry, C_{i+1} , to the next adder stage.

In a parallel arithmetics units, all 2n inputs bits are usually available to the adder at the same time. However, the carries have to propagate from the FA in position 0 to position i in order to produce the correct sum and carry-out

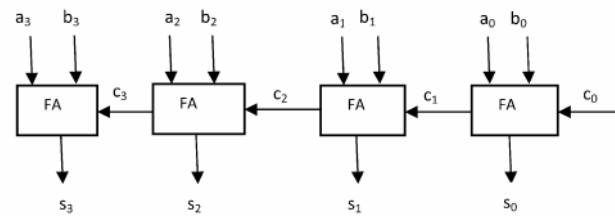


Figure 3: A 4-bit Ripple-Carry Adder

bits. In other words, we need to wait until the carries ripple through all n FAs.

4.2. Carry Look Ahead Adder (CLA)

The carry propagation can be speed-up by three ways. First, to use a faster logic circuit technology, second, to generate carries by means of forecasting logic that does not rely on the carry signal being rippled from stage to stage of the adder and third, by carry free addition algorithm. The acceleration of the computation uses speed-up techniques (Carry Look Ahead, Carry Select and Carry Skip). Finally, a combination of these techniques can prove to be an optimum for large adders.

Weinberger and Smith [11] uses look ahead carry technique rather than carry rippling technique to speed up the carry propagation by Calculating values for each digit position whether that position is going to propagate a carry if one comes in from the right and deduce quickly whether that group is going to propagate a carry that comes in from the right.

4.3. Carry Select Adder (CSA)

Carry select adder scheme [15] divides adders into blocks of ripple carry adder each with two replicas, one replica evaluates with carry- in of 0, the other one with carry- in of 1. In this scheme the carry out is from less significant block which then conditionally selects the output of succeeding blocks.

4.4. Carry-Skip Adders

Carry-skip scheme is proposed by Kilburn et al. [16] to accelerate the carry propagation. A carry-skip adder reduces the time needed to propagate the carry by skipping over groups of consecutive adder stages. In VLSI technology the carry-skip adder is compatible in speed to the carry look-ahead technique but it requires less chip area and consumes less power. Carry Skip Adders take advantage both of the generation or the propagation of the carry signal. They are divided into blocks, where a special circuit detects quickly if all the bits to be added are different ($P_i = 1$ in the entire block). It adds additional group carry bypass paths to its ripple path and the carries can bypass the ripple path when the group propagate signal is high.

4.5. Ling Adders

Adder proposed by Ling [17] is an improved version of conventional carry-look ahead adder. This approach is faster,

less expensive and is designed using H. Ling’s equations and generally implemented in BiCMOS. These adders replaces the conventional propagate operator from a XOR with an OR gate which results in a much cheaper operation. In Ling’s modification, “*p*” signal is replaced by “*t*” signal, which is called transfer bit and the group generated carry G_i by H_i .

$$p_i = a_i \oplus b_i \rightarrow t_i = a_i + b_i \tag{12}$$

$$H_i = G_i + G_{i-1} \tag{13}$$

If there is a carry-in or a carry-out at i^{th} bit then H_i is high. Similarly the generate and group generate operation can be reformulated as:

$$G_i = g_i + g_i G_{i-1} + p_i G_{i-1} = g_i + (g_i + p_i) G_{i-1} = g_i + t_i G_{i-1}$$

$$H_i = g_i + t_{i-1} H_{i-1} \text{ (Ling)} \tag{14}$$

$$G^* = g_3 + t_3 g_2 + t_3 t_2 g_1 + t_3 t_2 t_1 g_0 \text{ (conventional)}$$

$$H^* = g_3 + t_2 g_2 + t_2 t_1 g_1 + t_2 t_1 t_0 g_0 \rightarrow g_3 + g_2 + t_2 g_1 + t_2 t_1 g_0 \text{ (Ling)} \tag{15}$$

Comparing G^* and H^* , the maximum fan-in of Ling adder is 3 as compared to 4 of CLA, which means the pull-down path is shortened by one and thus the speed is improved. The advantage of this scheme is its cheaper carry propagation at the cost of more complex sum generation stage. There are also a large group of Hybrid adders that uses a combination of two or more of the previously described methods for addition to achieve the desired performance. A common approach to the design of hybrid adders is to choose one method for carry propagation and another method for sum calculation.

5. SIGNED BINARY DIGIT (SBD) ARITHMETIC

In the binary signed digit number system, each digit can assume any one of three values $\{-1, 0, 1\}$. As a result redundancy is introduced in a system i.e. a number can be represented in more than one way. Due to the presence of redundancy one can perform carry-propagation free addition and hence parallel addition of two redundant numbers can be performed in a constant time independent of the word length of operands.

Signed-digit (SD) number representation makes it possible to perform “Carry free addition” which means that carry propagation is limited to a single digit position i.e. the carry propagation length is fixed irrespective of the word length and has been used to speed up arithmetic operations. This redundancy can be exploited to limit the length of carry propagation chains to only one digit position, making it possible to add two numbers in fixed time, irrespective of the word length. The above algorithm can be clearly understood with the help of following example:

Radix (r) = 10, α = 6		Radix (r) = 2, α = 1	
$\bar{4}15$ (-385) Addend		$\bar{1}01\bar{1}$ (-7) Addend	
$3\bar{1}2$ (292) Augend		$1\bar{1}\bar{1}0$ (2) Augend	
$\bar{1}07$		$0\bar{1}0\bar{1}$	
$\bar{1}0\bar{3}$	Intermediate sum	$010\bar{1}$	Intermediate sum
010	Intermediate carry	$0\bar{1}000$	Intermediate carry
$\bar{1}1\bar{3}$ (-93) Final sum		$0\bar{1}10\bar{1}$ (-5) Final sum	

5.1. Algorithm of Signed Digit Number Representation

For a given radix r , each digit mi in an sign digit number system is typically in the range,

$$-\alpha \leq x_i, y_i \leq +\alpha, \text{ where } \left[r - \frac{1}{2} \right] \leq \alpha \leq r - 1 \tag{16}$$

The addition here is done in two steps. In the first step, an intermediate sum s_i and a carry c_i is generated parallely for all digit position based on the operand digits m_i and n_i at each digit position i . In the second step, the summation $p_i = s_i + c_{i-1}$ is carried out to produce the final sum digit p_i the most important fact is that it is always possible to select the intermediate sum s_i and c_{i-1} carry such that the summation in the second step does not generate a carry. If the selected value of b in the equation 1 satisfies the condition:

$$\left[r + \frac{1}{2} \right] \leq \alpha \leq r - 1 \tag{17}$$

Then the intermediate sum s_i and c_i depend only on the input operands i.e. on m_i and n_i . The interim sum is-

$$s_i = x_i + y_i - rc_i \tag{18}$$

$$\text{where } c_i = \begin{cases} 1 & \text{if } x_i + y_i \leq +b \\ -1 & \text{if } x_i + y_i \leq -b \\ 0 & \text{if } x_i + y_i < b \end{cases}$$

Let $\alpha_i = x_i + y_i$ & $\alpha_{i-1} = x_{i-1} + y_{i-1}$ denote the sums of the input digits at the two positions respectively. Given in table, the rules for generating the intermediate sum s_i & carry c_i . In the table, the symbol X indicates a don’t care i.e. the value of α_{i-1} doesn’t matter.

Table 1
Signed Digit (SD) Rules for Selecting Carry c_i and Intermediate Sum s_i Based on α_i for Radix r

α_i	α_{i-1}	c_i	s_i
$\alpha_i \leq 2b$	X	-1	0
$-2b < \alpha_i < -b$	X	-1	$\alpha_i - r$
$\alpha_i = -b$	$\alpha_{i-1} \leq -b$	-1	b
	$\alpha_{i-1} > -b$	0	-b
$-b < \alpha_i < b$	X	0	α_i
$\alpha_i = b$	$\alpha_{i-1} < b$	0	b
	$\alpha_{i-1} \geq b$	1	-b
$b < \alpha_i < 2b$	X	1	$\alpha_i - r$
$\alpha_i \geq 2b$	X	1	0

6. DESIGN ISSUE AND SIMULATION RESULTS

CIC decimation filter have a poor low pass response, however they are easy to implement and not require multiplications in real-time. Low pass magnitude response can be improved by compensating the passband droop and then by using redundant signed digit number arithmetic, the compensated response can realize more faster than other compensation method.

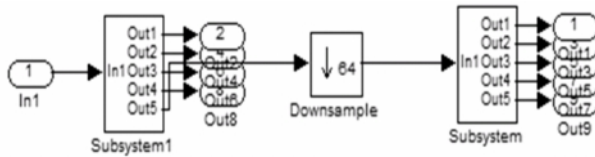


Figure 4: Sub-System Model for Down Sampling

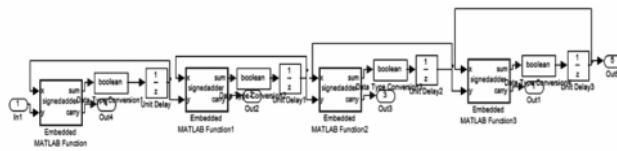


Figure 5: Inner-System Model for Down Sampling with Signed Digit Adder

The presented CIC decimation filter and the cascade equivalence are used to build an efficient compensated CIC decimation filter structure. we analyzed the performance of Compensated CIC decimation filter using different fast adder algorithm. The resultant decimation filter using signed digit arithmetic shows faster and more efficient performance in terms of frequency response, speed and area.

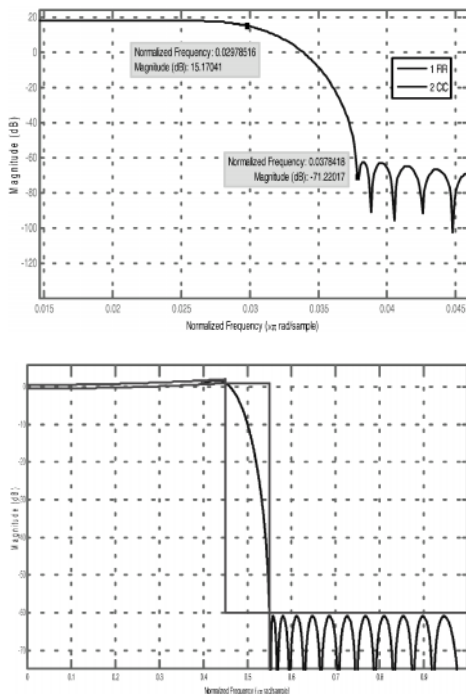


Figure 6: (a) Two CIC Filters with one FIR Filter with K = (8,2) X4 (b) Compensated CIC Decimation Filter with K = 64

Filter Coefficient

-0.005022 -0.01366 -0.013297 0.0011859 0.011467 -0.00013418

Response Time

Number of Bits	Decimated CIC Filter	Compensated CIC Filter	Compensated CIC Filter with RCA	Compensated CIC Filter with SD-8	Compensated CIC Filter with SD-16
2	8.541ns	8.623ns	7.935ns	6.216ns	6.217ns
4	8.686ns	8.725ns	7.852ns	6.215ns	6.211ns
8	8.746ns	8.819ns	7.763ns	6.203ns	6.201ns
16	8.873ns	8.965ns	7.602ns	6.118ns	6.103ns
32	8.951ns	9.128ns	7.511ns	6.067ns	6.058ns
64	9.113ns	9.298ns	7.385ns	6.006ns	6.017ns
128	9.265ns	9.378ns	7.218ns	5.982ns	5.972ns
256	9.329ns	9.504ns	7.119ns	5.866ns	5.827ns
512	9.471ns	9.717ns	7.097ns	5.582ns	5.561ns
1024	9.523ns	9.831ns	7.019ns	5.579ns	5.534ns

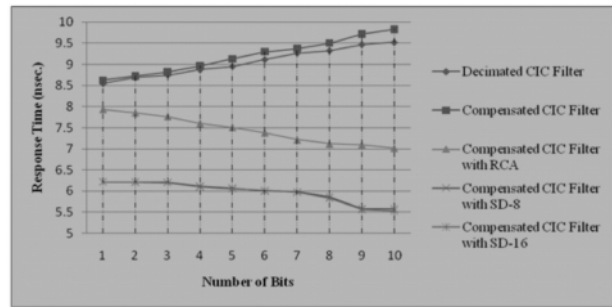


Figure 7: Comparison of Response Time of Different Addition Algorithms

7. CONCLUSIONS

This paper presented an efficient and compensated CIC decimation filter using different fast adder algorithms to increase the speed of down sampling. Area wise CIC filter is a better option than FIR due to its multiplier less architecture and less circuitry and hence results a trade-off between the desired compensation of the passband droop and area. Large rate changes require fast multipliers and very long filters and therefore by using the Signed digit numbers system at bit level, the speed of operation can be increased. Through examples and MATLAB programming for radix-8 and 16, we conclude that algorithm for signed digit numbers is more flexible, where it is a generalized form & very well includes case of hybrid numbers. After the proper analysis and comparison of compensated CIC filter structure with fast adder algorithm for radix - 8 and 16, it is found that the proposed compensated filter stucture using signed digit arithmetic is more efficient in terms of the desired passband droop of the overall decimation factor, response time and area. The width of the passband and the frequency characteristics outside the passband are severely

limited. Additionally, using the polyphase decomposition, the filters at the first stage can be moved at the lower rate and by increasing the number of stages, the amount of passband aliasing or imaging error can be brought within required ranges.

REFERENCES

- [1] E. B. Hogenauer, "An Economical Class of Digital Filters for Decimation and Interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-29, No. 2, pp. 155-162, April 1981.
- [2] H. Aboushady, Y. Dumonteix, M. M. Loerat, and H. Mehrez, "Efficient Polyphase Decomposition of Comb Decimation Filters in Σ - Δ Analog-to-digital Converters," *IEEE Transactions on Circuits & Systems– II: Analog and Digital Signal Processing*, Vol. 48, pp. 898-903, October 2001.
- [3] A. J. Kwentus, Z. Jiang and A. N. Willson, Jr., "Application of Filter Sharpening to Cascaded Integrator-comb Decimation Filters," *IEEE Transactions on Signal Processing*, Vol. 45, pp.457-467, February 1997.
- [4] G. Jovanovic-Dolecek and S. K. Mitra, "Sharpened Comb Decimator with Improved Magnitude Response," Proc. 2004 International Conference on Acoustics, Speech, and Signal Processing, Montreal, Canada, Proc. Vol. 2, pp. 393-396, May 2004.
- [5] L. L. Presti, "Efficient Modified-sinc Filters for Sigma-delta A/D Converters," *IEEE Transactions on Circuits & Systems– II: Analog and Digital Signal Processing*, Vol. 47, pp. 1204-1213, November 2000.
- [6] G. Jovanovic-Dolecek and S. K. Mitra, "Efficient Multistage Comb-modified Rotated Sinc (RS) Decimator," Proc. EUSPCO-2004, Vienna, Austria, pp.1425-1427, September, 2004.
- [7] F. Daneshgaran and M. Laddomada, "A Novel Class of Decimation Filters for $\Sigma\Delta$ A/D Converters," *Wireless Communications and Mobile Computing*, Vol. 2, No. 8, pp. 867-882, December 2002.
- [8] G. Jovanovic-Dolecek and J. Diaz-Carmona, "A New Cascaded Modified CIC-Cosine Decimation Filter," Accepted for Intern. Conference ISCAS-2005, Kobe, Japon.
- [9] Gordana Jovanovic Dolecek and Fernando Javier Trejo Torres, "Compensated CIC-Cosine Decimation Filter," *ECTI Transactions on Computer And Information Technology* Vol.4, No.1, pp. 8-13, May 2010.
- [10] K. Hwang, "*Computer Arithmetic: Principles, Architecture and Design*", John Wiley and Sons, 1979.
- [11] Weinberger, J. L. Smith, "*A Logic for High-Speed Addition*", National Bureau of Standards, Circulation 591, p. 3-12, 1958.
- [12] J. E. Robertson, "A Deterministic Procedure for the Design of Carry-Save Adders and Borrow-Save Subtractors," University of Illinois, Urbana-Champaign, Dept. of Computer Science, Report No. 235, July 1967.
- [13] A. Avizienis, "Signed-digit Number Representation for Fast Parallel Arithmetics", *IRE Transactions on Electronic Computers*, p. 389, 1961.
- [14] N. Takagi, H. Yasuura and S. Yajima, "High Speed VLSI Multiplication Algorithm with a Redundant Binary Addition Tree," *ZEEE Trans. Comp.* Vol. 34 pp. 789-796, Sept. 1985.
- [15] O. J. Bedrij, "Carry-Select Adder", *IRE Transactions on Electronic Computers*, p. 340, June 1962.
- [16] T. Kilburn, D. B. G. Edwards and D. Aspinall, Parallel Addition in Digital Computers: A New Fast "Carry" Circuit, *Proc. IEE*, Vol. 106, Pt. B. p. 464, September 1959.
- [17] Ling, "High Speed Binary Parallel Adder", *IEEE Transactions on Electronic Computers*, EC-15, p. 799-809, October, 1966.

