

# Polyphase Implementation of Non-recursive Comb Decimators for Sigma-Delta A/D Converters

Shahana T. K., Rekha K. James, Babita R. Jose, K. Poulouse Jacob and Sreela Sasi

**Abstract** – In a sigma-delta analog to digital (A/D) converter, the most computationally intensive block is the decimation filter and its hardware implementation may require millions of transistors. Since these converters are now targeted for a portable application, a hardware efficient design is an implicit requirement. In this effect, this paper presents a computationally efficient polyphase implementation of non-recursive cascaded integrator comb (CIC) decimators for Sigma-Delta Converters (SDCs). The SDCs are operating at high oversampling frequencies and hence require large sampling rate conversions. The filtering and rate reduction are performed in several stages to reduce hardware complexity and power dissipation. The CIC filters are widely adopted as the first stage of decimation due to its multiplier free structure. In this research, the performance of polyphase structure is compared with the CICs using recursive and non-recursive algorithms in terms of power, speed and area. This polyphase implementation offers high speed operation and low power consumption. The polyphase implementation of 4<sup>th</sup> order CIC filter with a decimation factor of ‘64’ and input word length of ‘4-bits’ offers about 70% and 37% of power saving compared to the corresponding recursive and non-recursive implementations respectively. The same polyphase CIC filter can operate about 7 times faster than the recursive and about 3.7 times faster than the non-recursive CIC filters.

## I. INTRODUCTION

Sigma-delta analog to digital converter (ADC) is the most widely used oversampling ADC suitable for high precision operation. It has an important role in today’s mostly digital mixed mode systems as interface circuit. A sigma-delta ADC consists of two main building blocks, which are an analog sigma-delta modulator and a digital decimator. The analog part of the ADC modulates an analog input to an oversampled low resolution digital signal. The digital decimator part consists of a lowpass filter and a downsampler that is responsible for transforming the low resolution oversampled signal into high resolution signal sampled at Nyquist rate [1].

Shahana T. K., Rekha K. James, Babita R. Jose and K. Poulouse are with Cochin University of Science and Technology, India and Sreela Sasi is with Gannon University, Pennsylvania, USA. E-mail: [shahanatk@cusat.ac.in](mailto:shahanatk@cusat.ac.in)

As most of the sigma-delta ADC applications require decimation filters with linear phase characteristics, symmetric Finite Impulse Response (FIR) filters are widely used for implementation. But the number of FIR filter coefficients will be quite large for implementing a narrow band decimation filter. Implementing decimation filter in several stages reduces the total number of filter coefficients, and hence reduces the hardware complexity and power consumption [2].

The first stage of decimation filter can be implemented very efficiently using a cascade of integrators and comb filters which do not require multiplication or coefficient storage. The remaining filtering is performed either in single stage or in two stages with more complex FIR or infinite impulse response (IIR) filters according to the requirements. The amount of passband aliasing or imaging error can be brought within prescribed bounds by increasing the number of stages in the CIC filter. The width of the passband and the frequency characteristics outside the passband are severely limited. So, CIC filters are used to make the transition between high and low sampling rates. Conventional filters operating at low sampling rate are used to attain the required transition bandwidth and stopband attenuation.

Several papers are available in literature that deals with different implementations of decimation filter architecture for sigma-delta ADCs. Hogenauer has described the design procedures for decimation and interpolation CIC filters with emphasis on frequency response and register width [3]. A power efficient Sinc<sup>4</sup> filter for decimation is proposed in [4] and is further optimized in [5] by removing the pipelining registers between the adders. Another FIR-Sinc architecture is given in [6] for low-power consumption by taking the advantage of the low number of bits at input and use of multiple  $V_{DD}$  logic. The non-recursive algorithm for comb decimators is investigated in [7]. The comparison results with recursive CIC structure show that the non-recursive implementation provides reduced power consumption and increased circuit speed. The use of a combination of sharpened filter cells and modified-comb cells which diminishes the filter passband drop and increases the quantization noise rejection is presented in [8].

To reduce power consumption in a circuit either the clock rate or the operating voltage has to be decreased. But sigma-delta ADCs utilize oversampling at high clock rates, and hence power consumption will increase. Lowering the operating voltage increases the circuit delay that will put a

bound on operating frequency. One solution to this problem is to use parallel processing. Polyphase decomposition has been traditionally used to implement parallel structures in digital signal processing. A polyphase CIC implementation is done in [9] for high speed operation, but the complete rate reduction is achieved by using another CIC which is again a recursive structure. In this paper, a polyphase CIC decimator based on non-recursive algorithm is presented. The proposed polyphase decimator is having lesser computational effort, and hence reduces dynamic power dissipation compared to the classical recursive and non-recursive CIC decimators.

The organization of the paper is as follows: First, the classical CIC filters based on recursive and non-recursive algorithms are illustrated in section II. In section III, the polyphase decomposition of non-recursive architecture is presented. The simulation results and performance comparison of recursive, non-recursive and polyphase implementations are provided in section IV. The dynamic power dissipation, speed of operation and area requirement for CIC filters with different decimation factors obtained using Leonardo spectrum and Synopsys synthesis tools are given in this section. The conclusion of the paper is given in section V.

## II. CLASSICAL CIC FILTERS

Hogenuauer devised a flexible, multiplier free filter suitable for hardware implementation that can handle large sampling rate changes [3]. The basic structure of the Hogenuauer CIC is shown in Fig. 1. This consists of an integrator and a comb filter as two basic building blocks. So, it is an infinite impulse response (IIR) filter followed by a finite impulse response (FIR) filter. In a CIC filter of order  $k$ , the integrator section consists of a cascade of ' $k$ ' digital integrators operating at the high sampling rate  $f_s$ . Each integrator is a one-pole filter with unity feed back coefficient and the transfer function is

$$H_I(z) = \frac{1}{1 - z^{-1}} \quad (1)$$

The comb section consists of ' $k$ ' comb stages with a differential delay of ' $D$ ' and operates at the low sampling rate  $f_s/N$ , where ' $N$ ' is the rate change or decimation factor. The transfer function of a comb stage referenced to high sampling rate is

$$H_C(z) = 1 - z^{-ND} \quad (2)$$

The rate change switch between the two filter sections subsamples the output of the integrator stage reducing the sample rate from  $f_s$  to  $f_s/N$ . In practice, the differential delay,  $D$  is usually held equal to 1 or 2. Using (1) and (2), the system transfer function of the CIC filter with respect to the high sampling rate  $f_s$  is given by

$$\begin{aligned} H(z) &= H_I^k(z) H_C^k(z) \\ &= \frac{(1 - z^{-ND})^k}{(1 - z^{-1})^k} = \left[ \sum_{i=0}^{ND-1} z^{-i} \right]^k \end{aligned} \quad (3)$$

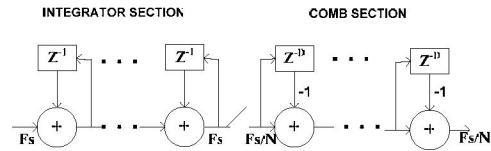


Fig. 1. CIC decimation filter

The major problems encountered with CIC filters include the following. The first problem is that the register widths can become large for large rate change factors. The register growth is considered in filter design process to ensure that no data are lost due to register overflow. The maximum register growth  $G_{\max}$  from the first stage up to and including the last stage is approximated as

$$G_{\max} = (ND)^k \quad (4)$$

If the number of bits in the input data stream is  $B_{in}$ , then the register growth can be used to calculate  $B_{\max}$ , the most significant bit at the filter output. It is given by  $B_{\max} = \lceil k \log_2 ND + B_{in} - 1 \rceil$ , where the least significant bit of the input register is considered to be bit number 'zero'. Since the first ' $N$ ' stages of the filter are integrators with unity feedback, the integrator outputs grow without bound for uncorrelated input data. It can be concluded that  $B_{\max}$  is the most significant bit not only for the integrators, but also for the combs that follow.  $B_{\max}$  is large for many practical cases, and can result in large register widths. So, truncation or rounding has to be used at each filter stage to reduce the register widths.

Second problem with the recursive CIC filter is the higher power consumption since the integrator stage works at the highest oversampling rate with a large internal word length. As the decimation ratio and filter order are increasing, power consumption increases significantly. Another problem is that the circuit speed will be limited by the large word length and recursive loop of the integrator stage.

The non-recursive CIC filter demonstrated in [7] reduces power dissipation and increases speed of operation by avoiding the IIR part in the recursive structure. The difference between the non-recursive and recursive algorithms is that they use different VLSI structures to implement the transfer function in (3). Taking differential delay  $D = 1$  and rate change factor,  $N = 2^M$ , the transfer function in (3) can be rewritten as

$$\begin{aligned} H(z) &= \left( \frac{1 - z^{-N}}{1 - z^{-1}} \right)^k = \left( \sum_{i=0}^{N-1} z^{-i} \right)^k \\ &= \left( \sum_{i=0}^{2^M-1} z^{-i} \right)^k = \prod_{i=0}^{M-1} (1 + z^{-2^i})^k \end{aligned} \quad (5)$$

The non-recursive CIC architecture is shown in Fig. 2. Every stage is a FIR filter but operates at different sampling rate. After each stage, the sampling rate is reduced by a factor of 2. The output from a sigma-delta modulator of word length  $B_m$  is given as input to the filter. The word length increases through every stage by ' $k$ ' bits, but the sampling rate decreases through every stage by a factor of 2 starting from the oversampling rate  $f_s$ . Thus the word length is short when the sampling rate is high, and

when the word length increases the sampling rate decreases. In the recursive algorithm, the IIR part has to operate with the oversampling rate and has a word length of  $\lceil k \log_2 N + B_m \rceil$  bits. In the non-recursive algorithm, the first stage works at the oversampling rate but has only a word length of  $(B_m + k)$  bits. This helps to reduce the power consumption and to increase the maximum speed of operation for non-recursive decimator.

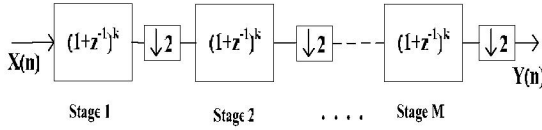


Fig. 2. Non-recursive comb decimator

### III. POLYPHASE NON-RECURSIVE CIC ARCHITECTURE

The average power consumption of a digital signal processing system is determined by the number of computations performed per sample, the word length and the sampling frequency. Parallel processing through polyphase decomposition is a wise way to achieve high speed and lower power consumption [10]. In the proposed implementation, polyphase decomposition is done for each FIR filter stage of the non-recursive decimator as shown in Fig. 3. Here, decimation occurs at the input of each filter reducing the sampling frequency by 2. So the number of computations per sample is also reduced to half of that for non-recursive implementation leading to low power consumption. As in non-recursive structure, polyphase implementation is also not having any register overflow problems, and the word length of initial stages is limited to a few bits. Since the use of Polyphase decomposition has reduced the operating frequency of the filters significantly at the last stages, the critical path is no longer a problem. So, the CIC filter can operate at a higher speed.

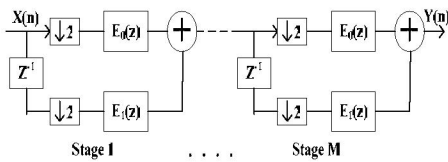


Fig. 3. Polyphase realization of non-recursive comb decimator

In general, an  $L$ -branch polyphase decomposition of the transfer function of FIR filter of order  $N$  is of the form

$$H(z) = \sum_{k=0}^N h(k)z^{-k} = \sum_{m=0}^{L-1} z^{-m} E_m(z^L) \quad (6)$$

$$\text{where } E_m(z) = \sum_{n=0}^{\lfloor (N+1)/L \rfloor} h(Ln+m)z^{-n}, 0 \leq m \leq L-1,$$

$$\text{with } h(n) = 0, \text{ for } n > N. \quad (7)$$

Performing two-branch polyphase decomposition of each FIR block of the non-recursive comb decimator, the transfer function in (5) can be rewritten as

$$H(z) = \sum_{m=0}^1 z^{-m} E_m(z^2) \quad (8)$$

Consider the comb decimator with decimation factor  $N = 64$ , and order  $k = 4$ , so that the polyphase filter equations are  $E_0(z) = h(0) + h(2)z^{-1} + h(4)z^{-2} = 1 + 6z^{-1} + z^{-2}$  and  $E_1(z) = h(1) + h(3)z^{-1} = 4 + 4z^{-1}$ .

The corresponding polyphase CIC filter architecture is shown in Fig. 4. The multiplier in the polyphase filter can be implemented using shift and add method, which requires only adder circuit as shifting can be achieved by properly routing the input bits.

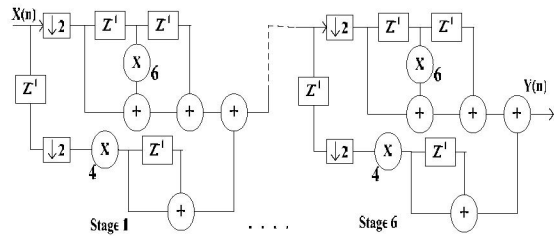


Fig. 4. Polyphase realization of non-recursive comb decimator for  $N=64$ ,  $k=4$ .

### IV. SIMULATION RESULTS

The proposed polyphase implementation is compared mainly with the recursive CIC or Hogenauer CIC in [3] and the non-recursive implementation in [7]. The filter architectures are defined using VHDL codes and functional simulation is performed using Modelsim. The filter responses obtained with the three implementations are identical.

Power, speed and area analysis is done with the Synopsys Design compiler using  $0.18 \mu\text{m}$ ,  $1.8\text{V}$  CMOS technology. Synthesis is done for 4<sup>th</sup> order CIC filter ( $k = 4$ ) using three different architectures, and a comparison of performance is done for different decimation factors as  $N = 64, 128$  and  $256$ . In all the implementations differential delay, 'D' is assumed as '1'. Table I shows the power comparison for the three implementations with an input word length,  $B_m = 4$ . The synthesis results show that for a 4<sup>th</sup> order CIC with decimation factor of  $N = 64$ , the polyphase implementation has about 70.02 % and 36.93% of power saving compared to the corresponding Hogenauer CIC and non-recursive implementations respectively. The power comparison plot is given in Fig. 5.

TABLE I. TOTAL DYNAMIC POWER CONSUMPTION FOR CIC ARCHITECTURES

Filter type	Dynamic power consumption in mW		
	N = 64	N = 128	N = 256
Hogenauer CIC	4.6950	5.3245	5.9968
Non-recursive CIC	2.2318	2.5398	2.8412
Polyphase CIC	1.4077	1.6129	1.815

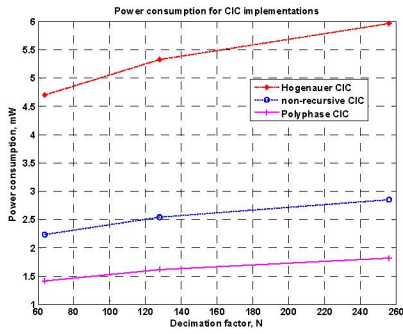


Fig. 5. Power consumption for CIC architectures with  $k=4$  and  $B_{in} = 4$ .

TABLE II. HIGHEST OPERATING FREQUENCY FOR CIC ARCHITECTURES WITH  $K = 4$ ,  $N = 64$  AND  $B_{IN} = 4$

Filter type	Highest operating frequency, MHz
Hogenauer CIC	64.7
Non-recursive CIC	122.6
Polyphase CIC	463.1

The total area required for filter implementation seems to be more for polyphase CIC and non-recursive implementation than that for recursive structure. The increase in area requirement for polyphase decomposition is due to the additional adder required for multiplication and for the extra decimator switch required in each stage. The synthesis results obtained for 4<sup>th</sup> order CIC filter with  $B_{in} = 4$  are given in Table III. It shows the area requirement of each CIC structure for different decimation factors as  $N = 64, 128$  and  $256$ . The area comparison plot is given in Fig. 6.

TABLE III. AREA REQUIREMENT FOR CIC ARCHITECTURES

Filter type	Total area occupied, $\mu\text{m}^2$		
	$N = 64$	$N = 128$	$N = 256$
Hogenauer CIC	30946	44832	56554
Non-recursive CIC	35367	59053	75004
Polyphase CIC	39788	75202	96079

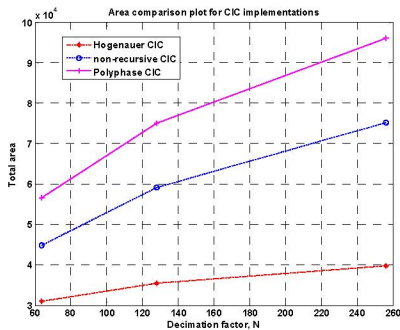


Fig. 6. Area requirement for CIC architectures with  $k = 4$  and  $B_{in} = 4$

## V. CONCLUSION

This paper presents a computationally efficient polyphase implementation of non-recursive cascaded

integrator comb (CIC) decimators for Sigma-Delta Converters (SDCs). The polyphase decomposition of non-recursive structure has the advantages of low power consumption and high speed operation compared to the recursive, non-recursive implementations. Low power consumption is achieved due to the fact that the word length is small for the initial stages which operate at high sampling rate, and as the word length increases for the subsequent stages the sampling rate is decreasing. Also, the computational complexity per input sample is reduced for each stage of polyphase structure than that for the non-recursive implementation. The maximum speed of operation of the polyphase structure is improved by the smaller word length of the first stage compared with that for recursive algorithm. As the first stage is operating at half the sampling rate and as parallel processing is done with polyphase decomposition, further speed improvement is obtained compared to the non-recursive implementation. The area requirement seems to be high for polyphase and non-recursive CICs than that for recursive structure. So the designer has to select the architecture of CIC based on the system requirements of power consumption, speed of operation and silicon area. Due to the low power and high speed operation, the polyphase CIC filters find applications in digital radio receivers, wireless communication systems, digital RF/IF signal processing and many others.

## REFERENCES

- [1] Philip E. Allen and Douglas R. Holberg, *CMOS Analog circuit design*, second edition, New York Oxford, Oxford University Press, 2002.
- [2] S. R. Norsworthy, R. Schreier and G. C. Temes, *Delta-Sigma Data Converters, Theory, Design, and Simulation*, Piscataway, NJ: IEEE Press, 1997.
- [3] E.B. Hogenauer, "An economical class of digital filters for decimation and interpolation", *IEEE Transactions on Acoustic, Speech and Signal Processing*, Vol. ASSP-29, No. 2, Apr. 1981, pp. 155-162.
- [4] J.C. Candy, "Decimation for Sigma Delta Modulation", *IEEE transactions on Communications*, Vol. COM-34, No. 1, Jan. 1986, pp. 72-76.
- [5] O.Gursoy, O. Saglamdemir, M. Aktan, S. Talay and G. Dunder, "Low-power decimation filter architectures for sigma-delta ADCs", 4th International Conference on Electrical and Electronics Engineering, Bursa, Turkey, December 2005.
- [6] S. F. Li and J. Wetherrell, "A compact low-power decimation filter for sigma-delta modulators", *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Turkey (ICASSP '00)*, Vol. 6, June 2000, pp. 3223-3226.
- [7] Y. Gao, L. Jia, J. Isoaho and H. Tenhunen, "A comparison design of comb decimators for sigma-delta analog-to-digital converters", *Analog Integrated Circuits and Signal Processing*, 22, Kluwer academic publishers, 1999, pp. 51-60.
- [8] M. Laddomada, "Comb-based decimation filters for sigma-delta A/D converters: Novel schemes and comparisons", *IEEE Trans. on signal processing*, Vol. 55, No. 5, May 2007, pp. 1769-1779.
- [9] H. K. Yang and W. M. Snelgrove, "High speed polyphase CIC decimation filters", *IEEE International Symposium on Circuits and Systems (ISCAS '96)*, GA, USA, Vol. 2, May 1996, pp. 229-232.
- [10] U. Meyer-Baese, *Digital signal processing with field programmable gate arrays*, Springer-verlag Berlin Heidelberg, New York, 2001.