

# Low Power Approach for Decimation Filter Hardware Realization

Kar Foo Chong, Pradeep K. Gopalakrishnan and T. Hui Teo

**Abstract**—There are multiple ways to implement a decimator filter. This paper addresses usage of CIC (cascaded-integrator-comb) filter and HB (half band) filter as the decimator filter to reduce the frequency sample rate by factor of 64 and detail of the implementation step to realize this design in hardware. Low power design approach for CIC filter and half band filter will be discussed. The filter design is implemented through MATLAB system modeling, ASIC (application specific integrated circuit) design flow and verified using a FPGA (field programmable gate array) board and MATLAB analysis.

**Keywords**—CIC filter, decimation filter, half-band filter, low power.

## I. INTRODUCTION

DECIMATION filter has wide application in both the analog and digital system for data rate conversion as well as filtering. One of the most popular applications of the decimation filter is sigma-delta ADC (analog-to-digital converter) [1]. Sigma-delta ADC is widely used in audio application for very high resolution, such as CD (compact disc) player.

Conventionally sigma-delta modulator is sampled at significantly higher frequency compare to the actual frequency band where the interested signal located in. In most of the communication device, we will implement a decimator filter to reduce the data rate in order to reduce the dynamic power consumption. There are many approaches in the decimation filter implementation. i.e cascade CIC – FIR (finite impulse response) filter, multiple stage HB filter and cascade CIC – HB filter [2].

**Fig. 1** delineates the basic building blocks of a decimator filter. In this design, a  $F_s = 19.2$  kHz input sampling frequency is first reduce to 1.2 kHz ( $F_s/16$ ) with a third order CIC filter, followed by further reduction to 300 Hz using two HB filters. This decimation filter is designed for sigma-delta ADC in ECG (electrocardiograph) application, which has a signal band from 0.05 to 150 Hz.

In this paper, a cascade CIC – HB filter implementation is addressed in detail. This paper is organized as follows. Section II highlights the design specifications and show

couple of the overall filter design frequency response results from FPGA verification. The CIC filter supporting theory and detail digital circuit and results are summarized in Section III. The HB supporting theory and detail digital circuit and results are summarized in Section IV. Conclusion is drawn in Section V.

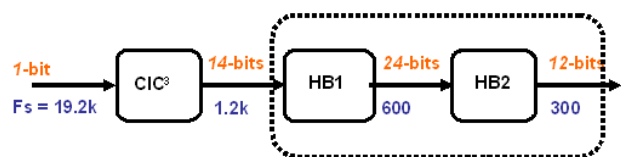


Fig. 1 Building blocks of a decimator filter

## II. PROCEDURE FOR PAPER SUBMISSION OVERALL DESIGN SPECIFICATIONS AND RESULTS

This featured decimator filter is targeting on operating at  $1\text{-V} \pm 10\%$  supply, across room temperature. The power consumption of the overall decimator filter is to keep below  $20\text{ }\mu\text{W}$ . The cut off frequency is at 150 Hz and the attenuation at 80 dB. Overall gate count should be below 120 k. This design is targeted for low power usage. That is the reason the size of the design is slightly larger. Parallel CSD (canonical signed digit) multiplier is adopted to design the HB filter to reduce the dynamic power consumption.

### A. Frequency Response of the design

The filters were designed and verified using FPGA. Their frequency responses are plotted using MATLAB. **Fig. 2** show the spectrum of the decimation filter with 150 Hz input signal.

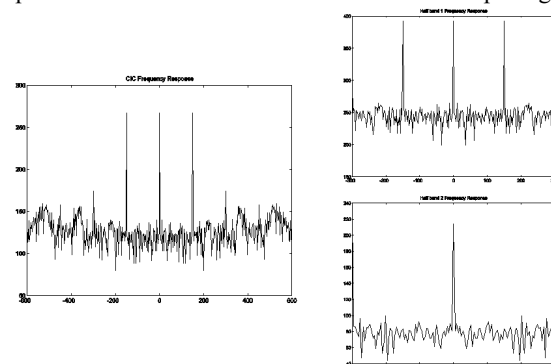


Fig. 2 Frequency response for 150 Hz input signal

Kar Foo Chong, Pradeep K. Gopalakrishnan, and T. Hui Teo are with the Institute of Microelectronics, A\*STAR (Agency for Science, Technology and Research), Singapore (phone: 65-67705323; fax: 65-67745754; e-mail: teehui@ime.a-star.edu.sg).

### B. Area and Power Consumption

The decimation filter's gate count and dynamic power were estimated with Synopsys DC (Design Compiler) tool. Table I show the area and power result. The total gate count is less than 40 k with less than 20  $\mu\text{W}$  power consumption.

TABLE I AREA AND POWER ESTIMATION OF THE DECIMATION FILTER

| Block   | Gate Count | Dynamic Power       |
|---------|------------|---------------------|
| CIC     | 3 k        | 1.19 $\mu\text{W}$  |
| HB 1    | 12 k       | 6.19 $\mu\text{W}$  |
| HB 2    | 21 k       | 12.43 $\mu\text{W}$ |
| Overall | 36 k       | 19.81 $\mu\text{W}$ |

### III. CIC FILTER

A 3<sup>rd</sup> order CIC filter is selected for a better attenuation. The CIC filter will reduce the sampling frequency by a factor of 16 and differential delay used is 1. The frequency of the input signal is 19.2 kHz and the frequency of the output signal is 1.2 kHz. Two CIC filter design approaches were implemented to estimate their area and power performances.

The first CIC filter design approach is to cascade three stages of accumulator, followed by a decimator and another three stages of differentiator. Second approach is to implement the CIC filter transfer function through the polynomial formula with zeros and poles. Table II shows the area and power performance for the two design approaches. It is obvious that the cascade of accumulator and differentiator is a better approach in designing low power CIC filter.

TABLE II PERFORMANCE SUMMARY OF CIC FILTER DESIGNS

| Design Approach                         | Gate Count | Dynamic Power      |
|---|------------|--------------------|
| Cascade of Accumulator - Differentiator | 3 k        | 1.19 $\mu\text{W}$ |
| Polynomial                              | 8 k        | 8.82 $\mu\text{W}$ |

Fig. 3 shows the structure of the CIC filter using accumulator and differentiator. Block I represents the accumulator, R represents the decimator in the design and C represents the differentiator.

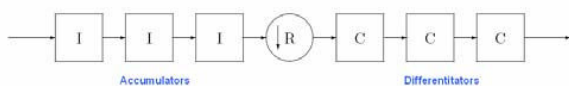


Fig. 3 CIC filter realization with cascade of accumulators and differentiators

Fig. 4 shows the basic accumulator (Integrator) in z-transforms and the digital circuit representation.

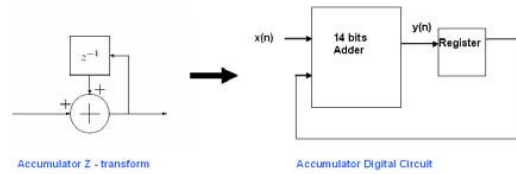


Fig. 4 Accumulator in Z-transform and digital circuit implementation

Fig. 5 shows the basic differentiator (Comb) in z-transforms and the digital circuit representation. M in the figure is the differential delay which is 1 in this design.

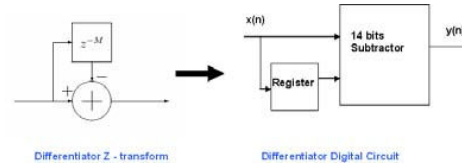


Fig. 5 Differentiator in Z-transform and digital circuit implementation

Internal word width needed to ensure not run time overflow is estimated from (1).

$$W = (1 \text{ Sign Bit}) + (\text{Number of Input Bit}) + (\text{Number of Stages}) \cdot \log_2(\text{Decimator Factor}) \quad (1)$$

In this design,  $W = 14$  bits.

$$W = 1 + 1 + 3 \cdot \log_2(16) \quad (2)$$

$$W = 14$$

Fig. 6 represents the digital circuit for the CIC filter design using the cascaded accumulator and differentiator approach. Each single bit input from the sigma-delta modulator is mapped to a 14 bits input. This data format used is Q-13 format where the first bit represents the sign bit and the rest of the bits represent the value of the data. Bit 1 is represented by  $1.22 \times 10^4$  and bit 0 is represented by  $-1.22 \times 10^4$ . The accumulators (Integrators) operate on 19.2 kHz and differentiators (Combs) operate on 1.2 kHz.

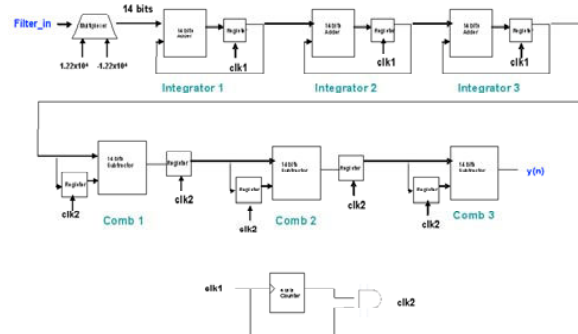


Fig. 6 CIC filter implementation using cascaded accumulator-differentiator approach

Fig. 7 shows the block diagram of the polynomial CIC filter. The output for each time sample is determined by the current input, previous input, and previous output. In order to realize

this design, registers are used to store the previous input sample and output sample.

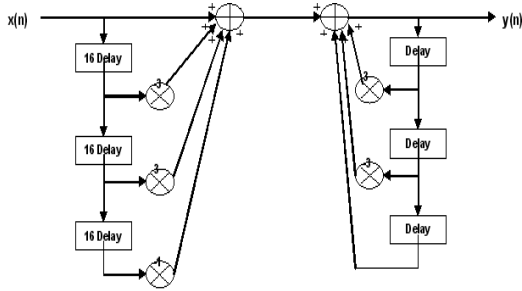


Fig. 7 CIC filter in polynomial representation

Transfer functions for the CIC filter is

$$H(z) = \frac{(1 - z^{-16})^3}{(1 - z^{-1})^3} \quad (3)$$

When the formula above (3) is expanded, it can be expressed as

$$H(z) = \frac{(1 - 3z^{-16} + 3z^{-32} - z^{-48})}{(1 - 3z^{-1} + 3z^{-2} - z^{-3})} \quad (4)$$

Inverse Z-transform of (4) gives the hardware implementation of the filter,

$$y(n) = x(n) - 3x(n-16) + 3x(n-32) - x(n-48) + 3y(n-1) - 3y(n-2) + y(n-3) \quad (5)$$

Fig. 8 shows the digital circuit to implement (5). Each of the multiplication is implemented using a CSD multiplier where each multiplier is implemented using shift register and adder [3].

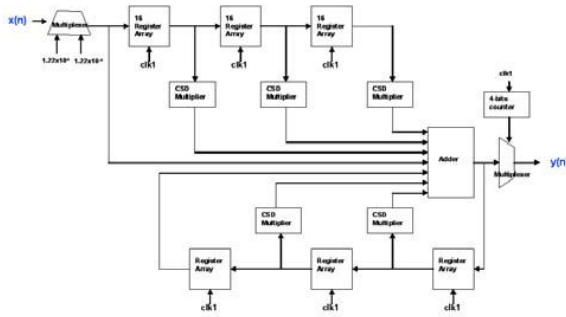


Fig. 8 CIC filter implementation using polynomial expansion

#### IV. HALF BAND FILTER

Two HB filters are required as a filter as well as a decimator. The first HB filter acts as a filter to improve the attenuation of the low frequency signal as well as reduce the sampling frequency from 1.2 kHz to 600 Hz. The first HB filter is a 25 tap filter with 13 non-zero coefficients. The second HB filter is a higher order filter to improve the attenuation. In this design, the second HB filter is a 45 tap

filter with 23 non-zero coefficients and it reduces the sampling frequency from 600 Hz to 300 Hz. CSD multiplier is adopted instead of a normal digital multiplier in designing HB filter to reduce the gate count. Figure 10 shows the design structure of the half band filter.

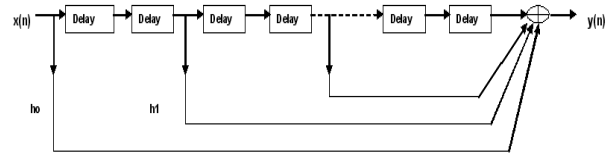


Fig. 9 Structure of the HB filter

Serial and parallel CSD multiplications were implemented to estimate the power consumption and area usage of the HB filters. The key component in a FIR HB filter is the multiplier used between the coefficient and the input sample. Their performances in power and area consumption are summarized in Table III. From this table, we understand that the parallel CSD multiplication will help us to reduce the dynamic power consumption.

TABLE III PERFORMANCE SUMMARY OF HB FILTER DESIGNS

| Design Approach | Gate Count | Dynamic Power |
|-----------------|------------|---------------|
| Serial CSD      | 10 k       | 7.26 $\mu$ W  |
| Parallel CSD    | 12 k       | 6.19 $\mu$ W  |

MATLAB is use to generate the coefficient based on the filter design specifications. A few MATLAB commands can be used to get the coefficient through MATLAB GUI such as “filterbuilder” and “fdatool”. These coefficients are converted to the CSD format for CSD multiplication. An effective CSD should have very minimum number of bit 1.

Parallel CSD approach requires only one clock signal in the operation. All CSD multiplications and other operations in the filter design are executed on the same clock at the sampling frequency. Fig. 10 shows the digital circuit for the HB filter using parallel CSD approach.

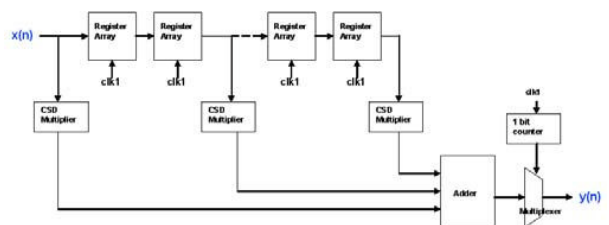


Fig. 10 HB filter implementation using parallel CSD multiplication

Serial CSD approach requires two clock signals in the operation. CSD multiplication operates at a much faster clock frequency and input samples are shifted in the filter at the

clock which operates at the sampling frequency. Figure 12 shows the digital circuit for the HB filter using serial CSD approach.  $clk2$  is the faster clock and  $clk1$  is the clock at the sampling frequency.

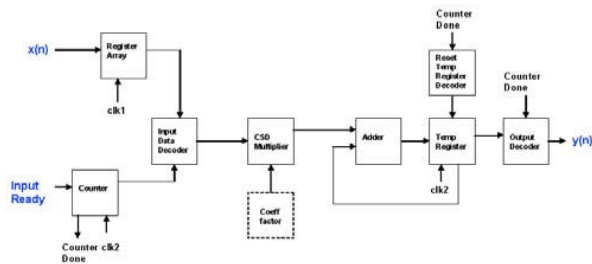


Fig. 11 HB filter implementation using series CSD multiplication

## V. CONCLUSION

In conclusion, different design approaches were implemented to realize a low power decimator filter. The performance in term of power consumption and gate count were also compared. In order to achieve low power consumption, the operating clock frequency and hardware reduction concept were implemented. For example, in CIC filter realization using cascaded accumulator-differentiator, the differentiators operate on lower frequency compare to the accumulators. CIC filter which is realized using polynomial involve multipliers consumes higher power. In HB filter realization, by avoiding higher clock frequency in the parallel CSD approach dynamic power can be reduced.

## ACKNOWLEDGMENT

The authors would like to thank staff of ICS Laboratory, IME, A\*STAR for various technical discussion and support.

## REFERENCES

- [1] S. R. Norsworthy, R. Schreier, and G. C. Themes, *Delta-Sigma Data Converters: Theory, Design, and Simulation*, IEEE Press, 1997.
- [2] 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, 1981.
- [3] Y. C. Lim, and B. R. Parker, "FIR filter design over a discrete powers-of-two coefficients space," *IEEE Transaction on Acoustics, Speech and Signal Processing*, vol. ASSP-31 pp. 583-591, 1983.