DESIGN OF A DECIMATION FILTER FOR NOVEL SIGMA-DELTA MODULATOR

Anna S. Kuncheva*, Thibault Mougel*, Lukas Fujcik**, Blagomir Donchev*, Marin Hristov***

*Department of Microelectronics, ECAD Laboratory, FEET, Technical University of Sofia, 8, Kliment Ohridski St, 1000 Sofia, Bulgaria, e-mail: anika@ecad.tu-sofia.bg;
***Université Toulouse-Le Mirail, France, thibaultmougel@yahoo.fr;

**Department of Microelectronics, FEEC, Brno University of Technology, Udolni 53, CZ – 602 00, Brno, Czech Republic; e-mail: fujcik@feec.vutbr.cz;

The steps involved in the design of decimation filter for high-resolution sigma-delta (ΣΔ) A/D converter are described. The design of a decimation filter is proposed that employs three stage – one Cascaded Integrator Comb filter (CIC) followed by two finite impulse response (FIR) filters. This approach eliminates the need for multiplication, requires a maximum clock frequency equal to the sampling. Specifications of decimation filter are dependent upon the overall specification from ΣΔ A/D converter with sampling frequency 16 MHz. The design implements a decimation ratio of 64, allows a maximum resolution of 14 bits in the output of the filter. Values of quantize coefficients are obtained using the tool Matlab. The decimation filter deign is written in VHDL, using "top-down" design methodology. The verification is made with test vectors generated in VHDL test bench module. Finally, it explains how to implement design in Xilinx Spartan-3 FPGA.

Keywords: ΣΔ modulation, decimation filter, A/D conversion, oversampling, FPGA, VHDL

1. INTRODUCTION

The oversampling sigma-delta modulation is a proven method to realize high- and very high-resolution analog to digital converters. Because of the use of oversampling in sigma-delta modulators, the need arises for changing the sampling rate of signal, decreasing it to Nyquist rate in A/D-converters. Thus, the high resolution can be achieved by the decimation (sample reduction). Such sample reduction can be achieved employing high precision FIR filters, usually in cascaded structures. This paper describes the steps of practical design of decimation filter for high-resolution ΣΔ A/D converter. The design of a decimation filter is proposed that employs three stage – one Cascaded Integration Comb filter (CIC) followed by two FIR filters. This approach eliminates the need for multiplication, requires a maximum clock frequency equal to the sampling. Finally, it explains how to implement the filter efficiently in hardware.

2. SPECIFICATION OF THE DECIMATION FILTER FOR SIGMA DELTA CONVERTER

A sigma-delta A/D-converter consists of a sigma-delta modulator, which produces the bitstream at the sampling rate, which can be in the megahertz range. The figure 1 shows a digital side of sigma-delta modulator. The specifications of the decimation filter are dependant upon the overall specification from the sigma-delta converter. These are summarized in Table 1.

The output of sigma-delta modulator is a 1-bit data bitsteam at the sampling rate, which is 16 MHz. The purpose of digital-and-decimation filter (Fig.1) is to extract information from this data bitstream and reduce the data rate to a more useful value.

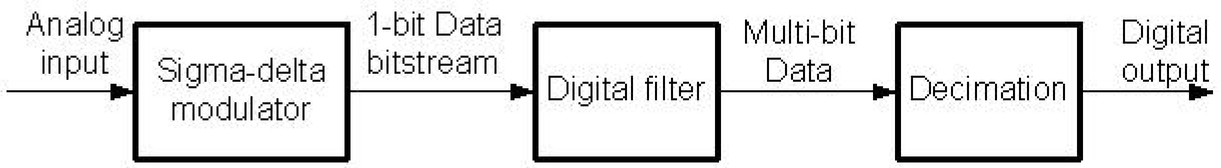


Fig.1. Digital side of sigma-delta modulator

Parameter	***Symbol***	***Value***
Signal bandwidth:	BW	100 kHz
Sampling Frequency:	Fs	16 MHz
Over Sampling Ratio:	OSR	64
Modulator order:	M	2
Number of bits in modulator bit stream:	Bmod	1
Number of bits in output of filter:	B	14

Table 1: Delta-sigma converter specification

Using specification given in Table 1, in sigma-delta ADC, the digital filter average the 1-bit data stream, improves the ADC resolution and removes quantization noise that is outside the band of interest. The purpose of decimation filter is to remove the noise shifted by modulator to high frequencies, reduce the sampling frequency of 16 MHz and increase the number of bits in the output. With 14 bits in the output of the filter Signal to Noise Ratio (SNR) is 86 dB. The overall characteristics of the filer are summarized in Table 2 below:

Filter parameter	***Value***
Sampling frequency:	Fs = 16 MHz
Over sampling ratio :	64
Passband frequency:	Fpass = 100 kHz
Stopband frequency:	Fstop = 125 kHz
Passband Max ripple:	Apass = 0.0005 dB
Stopband attenuation:	Astop = 90 dB

Table 2: Specification of whole decimation filter

F_{stop}: This is equals to Fs/(2*OSR)
A_{stop}: It is necessary to attenuate the signal to level of at least the quantization noise of the filter (in dB).
F_{pass}: This is a maximum frequency of the signal to be converted, and from given specification, is equal to the bandwidth of the signal, 100 kHz.
Decimation factor: This is to OSR of 64

3. DESIGN OF THE DECIMATION FILTER

Practically it is not possible to implement a single filter that would meet the characteristics of Table 2. The order of such of filter would be close to 3400. It is almost impossible to implement such a filter in hardware [1]. It is therefore necessary to use a multi-stage approach, whereby the decimation is performed in several stages. The proposed decimation filter architecture that consists of three stages – one Cascaded Integrator Comb filter followed by two (FIR) filters is shown in Figure 2.

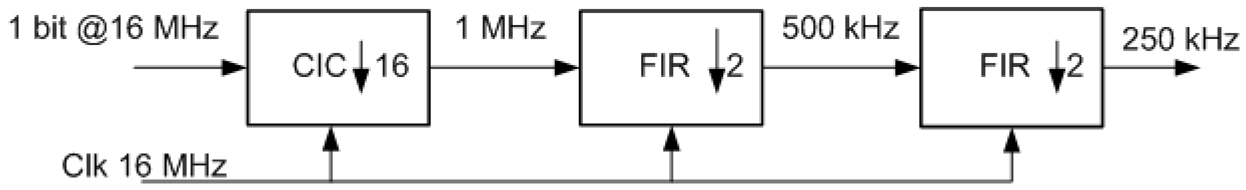


Fig 2. Decimation filter architecture

3.1 First stage, CIC filter

It is very efficient to use a Cascaded Integrated Comb filter (CIC) for the first stage [3]. Such CIC filter can be easily implemented in hardware, requiring no multiplication. Furthermore, it can be used to decimate the data by a large factor, allowing easier implementation of the following stages. The drawback is the drop in the pass-band due to the sin(x)/x response of the filter. This needs to be compensated for in the following stage of FIR filtering. In this design is used CIC filter to decimate data by decimation ratio 16 (Fig. 2). After simulation with the tool Matlab, is found that, in order to meet the overall specification of the filter, are needed 5 stages of CIC filtering with a differential order of 1.

3.2 Two stages of FIR filtering:

The numbers of FIR stages required by looking at computational effort required for different solution and selecting the best one. The frequency band attenuation for each stage can be determined using the following formula:

$$\textit{Passband: } 0 \le f \le F_p$$
$$\textit{Stopband: } F_i - F_s/2M < f < F_{i-1}/2$$
$$\textit{Ripple}_{passband}: \delta_p/I$$
$$\textit{Ripple}_{Stopband}: \delta_s$$

where:

Fs: Sampling frequency of filter

I: Total number of stage

$Fi = F_{i-1}/Mi$; *i is number of stage, (i=1, 2, ...I);*

The requirements for such a filter are highlighted in Table 3:

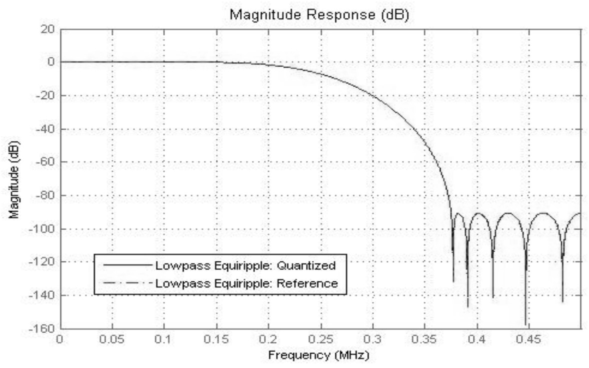
Filter parameter	***First FIR filter***	***Second FIR filter***
Sampling frequency:	Fs = 1 MHz	Fs= 500 kHz
Passband frequency:	Fpass = 100 kHz	Fpass = 100 kHz
Stopband frequency:	Fstop = 375 kHz	Fstop = 125 kHz
Passband Max ripple:	Apass = 0.00025 dB	Apass = 0.00025 dB
Stopband attenuation:	Astop = 90 dB	Astop = 90 dB

Table 3: Requirements for two stages of FIR filtering

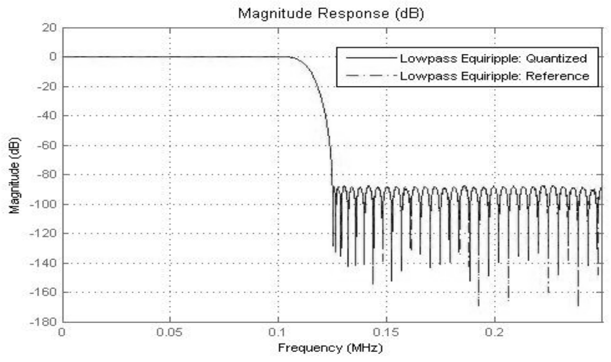
In this case, the filter orders of the first and second filter to be 18 and 111 respectively are found. Clearly, this requires less computational effort, than the one stage. As will be explain later on, it would be possible to design each of these filters in hardware using only one multiplier. Therefore, increasing the number of stage any further would turn out to be less efficient since it would require an extra multiplier for each further stage of FIR filtering. Therefore are chosen to use two stages of FIR filtering.

3.3 Quantization of coefficients:

Next step in this design is to select the number of bits used to represent each coefficient. It is obvious that quantization of the filter coefficient reduces the filter performance. Once again, Matlab tool is used to evaluate the effect that of quantization of the coefficient has on the response of each filter. When using 20 bits for quantization, the response of the quantized filter matches closely the reference filter, both in the passband and stopband. It is shown in figure 3. Therefore, 20 bits for the coefficients, both of first stage FIR filter and second stage FIR have been selected.



a. First stage FIR filter



b. Second stage FIR filter

Fig. 3. Frequency response of FIR filters with and without coefficients quantization

3.4 Efficient hardware implementation:

The goal is to implement a filter that meets all specifications, while making the most efficient use of hardware available. Since fast multipliers for data with relatively large number of bits are difficult to implement, and require a large amount of resources, it is important to try to reduce the number of multiplier required. Several methods can be used for this. A first step toward reducing the number of multiplication is making use of the fact that, for linear-phase response filter, coefficients are symmetrical. This is will half the number of multiplications required.

Further reduction in the number of multiplier required can be achieved by taking advantage of the fact that, in a multi-stage decimation filter, as to progress in stage, the input data rate is reduced. In this design, the first FIR filter is preceded by a comb filter that decimates the data by 16. This means input data to this FIR will only be at a rate equals to the over-sampling frequency divided by 16.

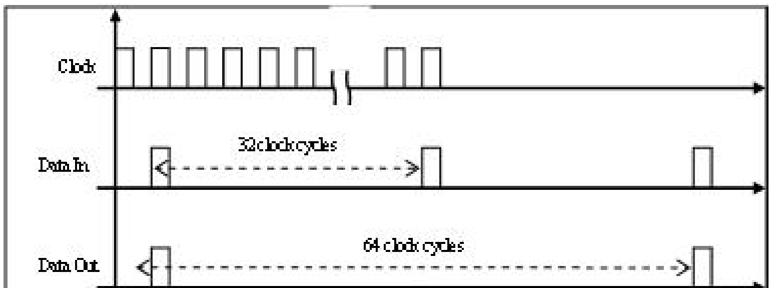


Fig. 4. Taking advantage of decimation Second FIR filter

Assuming this, the filter operates at the over-sampled frequency, and to perform one multiplication per clock cycle, it is possible to do 16 multiplications between samples. Things get even better with the second FIR, where data rate is further divided by 2.

This is advantageous since this last stage requires a shaper response in the transition band, therefore more coefficients, so more multiplications. In this case, for this second FIR, 32 multiplications can be performed between samples. Decimation also means that not all outputs values need to be computed. In this design, each filter decimates the data by 2. The time available to compute the output value of the second FIR filter is now 64 clock cycles, as shown in figure 4.

3.4.1 Summary:

- First FIR filter: 32 multiplications can be performed between outputs. Using the symmetry of coefficient, it can have a maximum number of 64 coefficients if using only one multiplier.
- Second FIR filter: 64 multiplications can be performed between outputs. Using the symmetry of coefficient, it can have a maximum number of 128 coefficients if using only one multiplier.

Both of these figures are less than the number of coefficients required to meet the specification of each stage. Therefore each FIR filter is implemented using only one multiplier.

4. FPGA IMPLEMENTATION

Two data corresponding to symmetrical coefficients are first added together, and then multiplied by the corresponding coefficient. The result is then stored in an accumulator. Once all multiplications for a given output have been performed, the content of the accumulator is send to the output, and its own content is reset for the next calculation. To reduce latency on the output is need to start by multiplying and accumulating all stored data samples corresponding to the next input sample.

The general structure of the FIR filter is showed in figure 5. This way, when the sample is received, there is only one multiplication left before the results can be sent to the output. This approach greatly reduces the latency on the output.

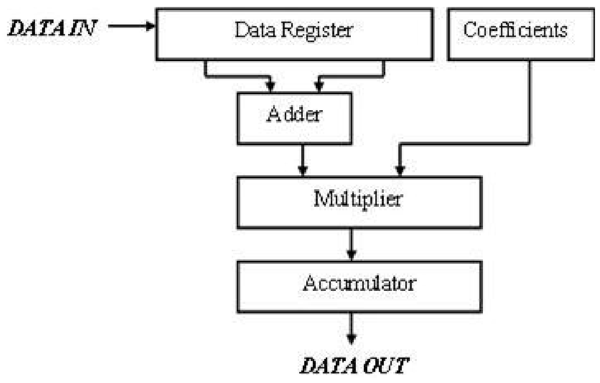


Fig. 5. Structure of FIR

The decimation filter was implemented in a Xilinx Spartan-3 XC3S200-4FT256 FPGA as an intermediate step in effort to design a high resolution A/D converter. The structure of FIR filter from Figure 5 is implemented in an FPGA using VHDL.

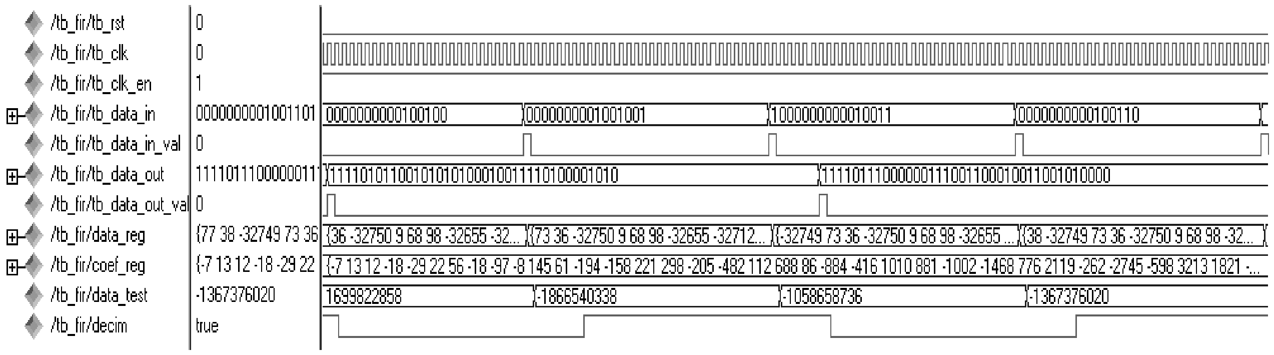


Fig. 6. Filter simulation waveforms

Value of quantize coefficients are obtained using the tool Matlab. A program written in VHDL then converts the coefficients to a constant array. BRAM are used to store coefficient values and data samples. The presented filter was tested by applying test vectors using ModelTech ModelSim™ simulator.

It is shown on Figure 6 above. The top level of digital circuit is shown on figure 7 below:

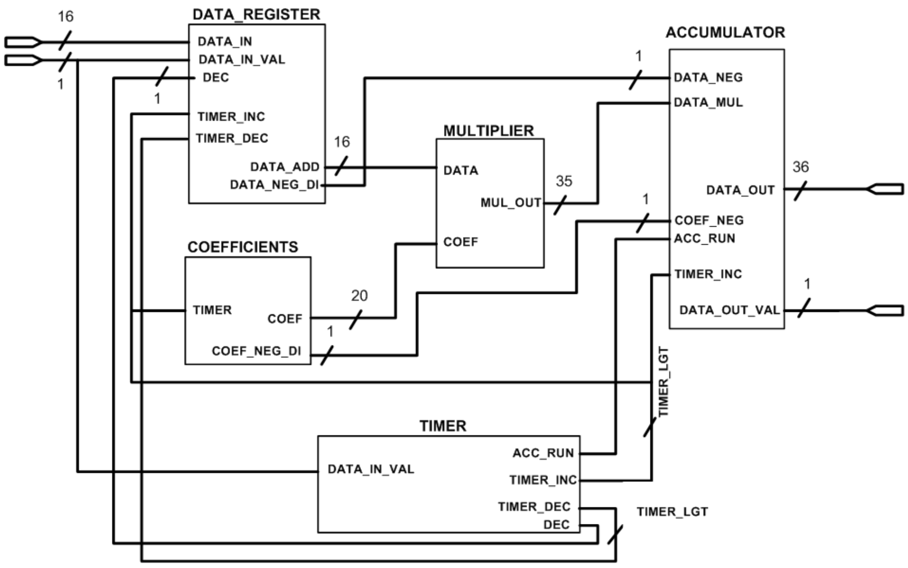


Fig. 7. Top level of the filter

After computer simulations the design was implemented in a Xilinx FPGA (XC3S200-4FT256). Implementation results are shown in Table 4 below:

Number of BUFGMUXs	1	(8)	12 %
Number of External IOBs	16	(173)	9 %
Number of LOCed IOBs	16	(16)	100 %
Number of MULT18X18s	2	(12)	16 %
Number of RAMB16s:	3	(12)	25 %
Number of Slices	451	(1920)	23 %
Number of SLICEMs	0	(960)	0 %

Table 4

5. CONCLUSION:

Specifications of decimation filter are derived from the specifications of the overall sigma delta modulator. The design of novel decimation architecture for ΣΔ converter has been presented. The architecture is composed of employs three stage – one Cascaded Integrator Comb filter (CIC) followed by two finite impulse response (FIR) filters. Using tool such as Matlab or other, it is possible to quickly find the filer order, the required quantization level for the coefficients and their values. Finally, by analyzing the design, is finds an efficient way to implement the filter in hardware. The use of top-down design methodology decreased the total design time. The high level hardware description language VHDL fully supports arithmetic and binary manipulations that are specific for this design.

6. REFERENCES:

[1] Emmanuel C. Ifeachor, Barrie W. Jervis, *Digital Signal Processing – A practical approach*

[2] http://www.beis.de/Elektronik/DeltaSigma/DeltaSigma.html "An introduction to Delta-Sigma converters";

[3]http://www.embedded.com/showArticle.jhtml?articleID=160400592 "Understanding cascaded integrated-comb filters", article from Embedded Systems Programming by Richard Lyons