Realization of Multistage FIR Filters Using Pipelining-Interleaving

Milenko Ciric and Vojkan Radonjic

***Abstract* — Multistage digital filters can be one of the solutions for the realization of filters with a narrow transition zone. If requirements for the width of transition zone are too strict, then they are the only alternative, and the decimation/interpolation must be performed in several steps. Combining decimation/interpolation operations related to the implementation of multi-channel filters in the PI (pipelining/interleaving) technique can give an efficient structure of multichannel multistage filter. Using the advantages offered by newer generations of FPGA chips in terms of digital design structure, it is possible to realize such filters with considerable savings of hardware resources and reduce the effect of finite length codeword. This paper proposes such an efficient implementation and presents the results of such a realization with FPGA components.**

***Keywords* — Critical loop limitation, digital filtering, effect of finite length codeword, multistage filter, pipelining/interleaving technique.**

I. Introduction

MULTISTAGE filters are suitable for the realization of filters where the bandwidth is less than a quarter of the sampling frequency. In such cases, multistage filters are actually the only solution and enforceable in practice. The general structure of these filters is shown in Fig. 1.

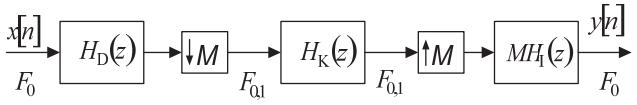

Fig. 1. The general structure of the multistage filter.

As we see from Fig. 1, a multistage filter consists of a block for reducing the sampling frequency, kernel filter, and a block for increasing the sampling frequency. We first reduce the sampling frequency to a lower value, and filtering with a kernel filter is performed at a lower frequency.

From the above we see that the realization of bandpass filters using a multi-stage filter has the following advantages:

1. Filtering requirements and problems can be divided into several lower orders;
2. The influence of finite word length on the overall performance of the filter is significantly reduced;
3. Arithmetic operations are performed in filters with a reduced sampling frequency.

M. P. Ciric, Tehnicko Remontni Zavod Čačak, Dr. Dragiše Mišovića 167, 32 000 Cacak, Serbia; (e-mail: cciro@blunet.rs).

V. M. Radonjić, Tehnicko Remontni Zavod Čačak, Dr. Dragiše Mišovića 167, 32 000 Čačak, Serbia; (e-mail: trzca@ptt.rs).

II. Multistage multirate filters in PI technique

In cases of very narrow filters, where a decimation/interpolation factor is significant (for example M>10), it is efficient to use multiple decimation/interpolation filters, as shown in Fig. 2 for a decimation side. In this way, instead of one decimation/interpolation filter, more of these filters (as many as we have degrees) are used but with a significantly lower order.

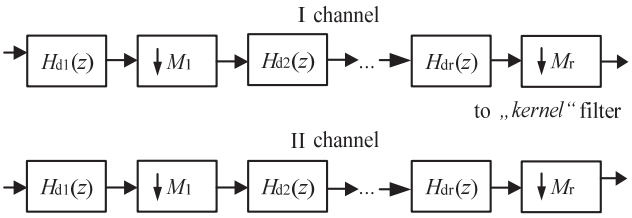

Fig. 2. Multistage decimation.

Suppose now that we need to realize a number of identical narrowband multistage filters. One of the most effective solutions would be the use of PI procedure (Fig. 3) which enables using only one digital filter instead of using K digital filters for each channel. Except for this realization, it is possible to realize a cascade connection of identical filters by introducing feedback loops from the output of one channel to the input of another. In our case we will take the variant from Fig. 3 (implementation without feedback) and replace K filters from the first decimation stage with one second-stage filter, kernel filters with third-stage and so on. This will be the total number of filters K times smaller than the number required for a standard implementation.

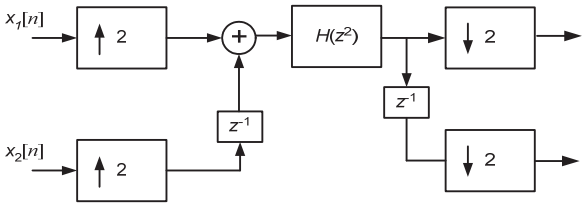

Fig. 3. Principle of PI techniques (two channels).

We see that in this way signals are processed in parallel and thereby we avoid the realization in which signal processing is performed sequentially. In practice it rarely happens that the sampling frequency and speed of signal processing are equal, especially when implementations of digital filters are with programmable hardware. If the sampling frequency is much lower than the clock

frequency at which the system elements work, then the use of parallel hardware unnecessarily consumes resources available chip elements. In this case, the application of parallel processing allows using the same hardware resources, at specific intervals of time, for varius signal processing operations.

III. Modification of Multistage Filters Using PI Technique

Consider now the use of PI technique on a multichannel multistage filter. As we said, the decimacion part of such a filter will contain several downsampler blocks and several decimation (as in Fig. 2), then the kernel filter and an interpolation part made analogously to the decimation part. Both channels should be identical.

Our filter will be realized by merging pairs of identical filters with PI technique of both channels (the first decimation filter of the first channel and the first decimation filter of the second channel, the second decimation filter of the first channel and the second decimation filter of the second channel, etc.).

If we look at the structure between two adjacent filters resulting from operations related to the PI process and operation of reducing the frequency (Fig. 4) , we notice that the algorithm that describes the decimation part of multistage filter (we will refer to it as an equivalent decimator) is:

- Save two consecutive samples,
- Remove the following 2*(M_k-1) samples,
- Repeat the process until the last sample expires.

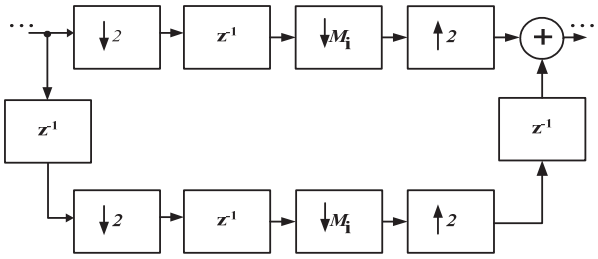

Fig. 4. One degree of multistage filter (equivalent decimator).

Analogously to this, the algorithm for the multistage interpolation part of the filter (we will refer to it as an equivalent interpolator) would read:

- Save two consecutive samples,
- Insert 2*(M_k-1) zeros,
- Repeat the process until the last sample expires.

These and similar structures are ideal candidates for the realization of programmable hardware. We can realize the entire structure as a single block, and then use it several times for each stage (equivalent decimator block and equivalent interpolator block, Fig. 5).

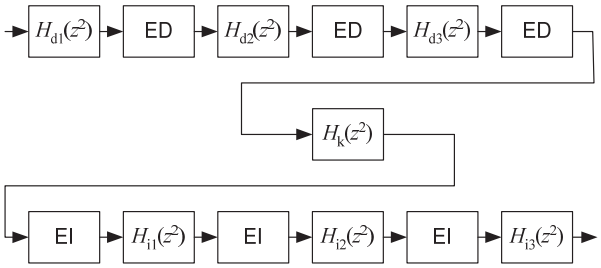

Fig. 5. The final structure of the filter H(z2).

IV. Multistage Filters with FPGA Komponents

Software tools for designing structures provided by digital programmable hardware manufacturers to users in recent years have become more complete and offer more and more freedom in design. The user has the option to choose whether to use one of the solutions from a wide range of forms for a specific digital structure, or start the design from the beginning, using the original manufacturer tool, or use some of the standard tools (including Matlab) and then again, via software manufacturers, make a compilation in VHDL, Verilog HDL, or the Verilog code.

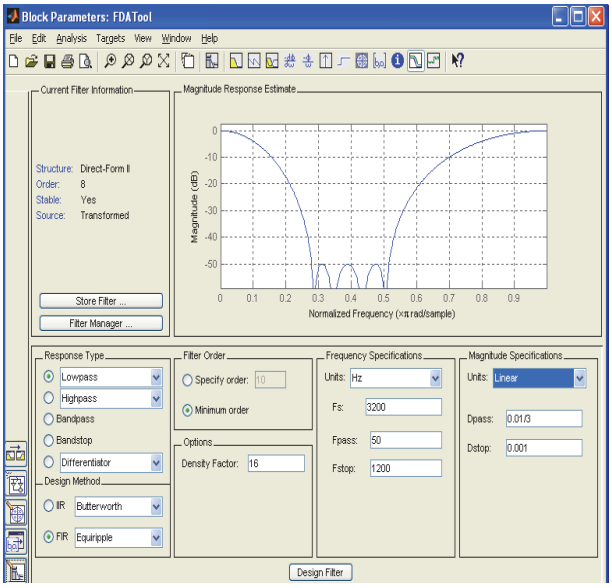

Fig. 6. Kernel filter design using FDA tool.

Let's see the realization of one multichannel narrowband low-pass filter with the following characteristics:

F_o=2 000 Hz, F_p=50 Hz, F_s=100 Hz;

δ_p=0,01, δ_s=0,001.

If we try to realize a filter implemented with standard methods, the filter order will obtain an unrealistic value (N> 100). In contrast, with the multistage realization of a filter with five stages, the kernel filter would now be of

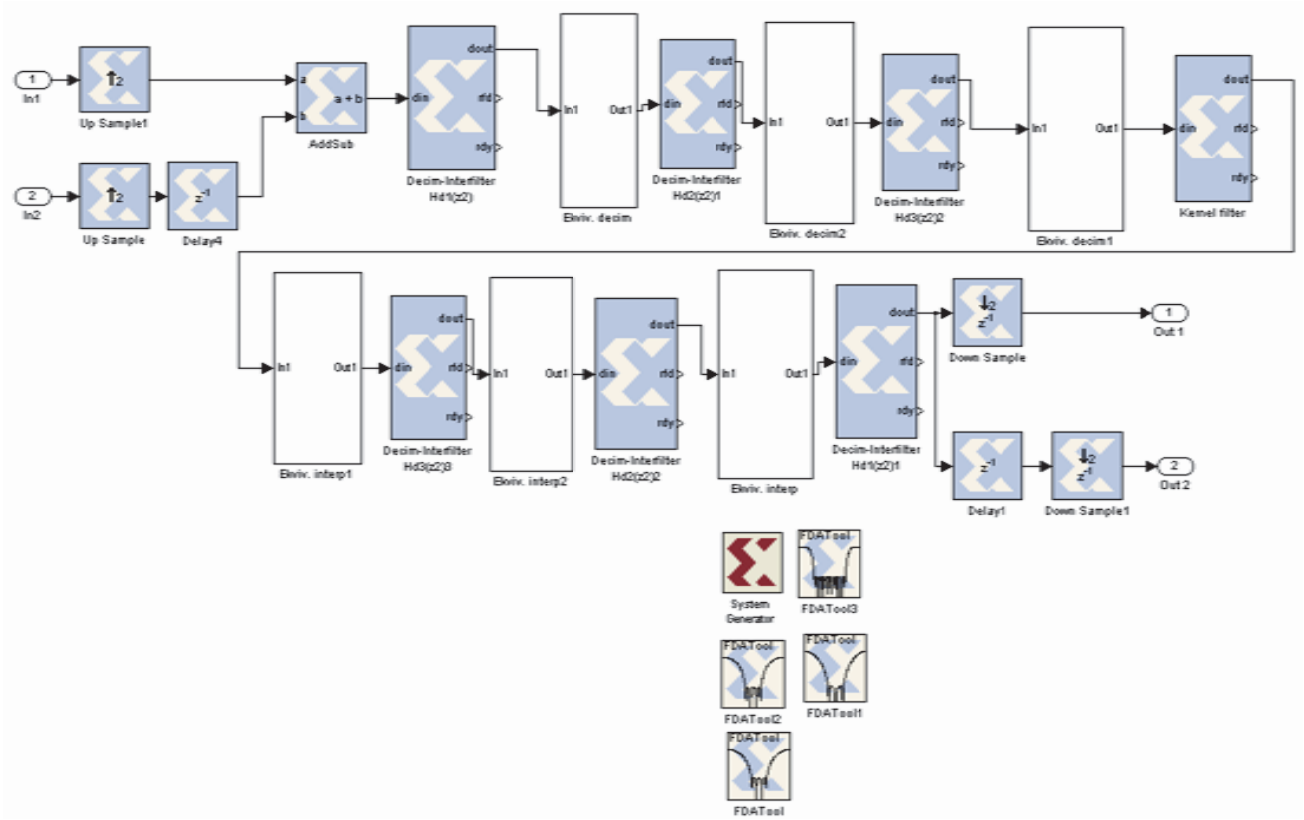

Fig. 7. FPGA realization of multistage filter.

the 17-th order. Let us go one step forward. If we change the requirements of the filter so that Fo=3200 Hz under the same conditions, it will not be possible to design the filter as a multistage filter with only one change of sampling frequency. Therefore, we will implement the filter in three stages and two channels with the same frequency change ratio M = 2, as shown in Fig. 5. The proposed structure represents one of the most effective solutions and also is an ideal candidate for the realization of programmable hardware (multiple use of identical blocks).

The filter will be implemented using Xilinx's System Generator. We will define blocks of equivalent decimators and equivalent interpolator as subsystems and make multiple use of them. All filters (decimation, interpolation and kernel) will be implemented as FIR filters with the use of multiplying coefficients from the Look-up tables (*distributed arithmetic*), instead of the usual procedure of multiplying samples with coefficients and their successive addition (*Systolic Multiply Accumulate*) and thus will reduce the accumulation of errors due to the impact effect of finite length codeword and also errors due to multiple signals passing through the same path (of course the fundamental error due to the effect of finite length codeword cannot be avoided). Fig. 6 shows the process of selecting kernel filter with its magnitude response, using the tools FDA tool, and simulation model of the entire filter is shown in Fig. 7. It should be noted that the implementation of filters must verify whether the proposed solutions in FPGA technology with a fixed point are possible.

Amplitude characteristics of the structure in Fig. 7, recorded in one channel are shown in Fig. 8, while the characteristics of an ideal FIR filter are shown in Fig. 9. We see that the ideal filter order is N = 163 and such a filter would be a pure fiction.

Calculating the points of amplitude characteristics is done by the principle sample by sample. The whole new structure is defined as a single block (*Xilinx*'s tool *AccelDSP* provides a connection between *Matlab* and *System generator*). At the input port of the new structure we have brought a sinusoidal signal whose frequency is changed in the range from zero to π and we record point by point features.

If we look at the characteristics we can conclude that with this realization we can successfully implement multistage multirate filters with stricter requirements and a narrow transition zone. The characteristic deviation is caused by imperfections of decimation/interpolation filters (aliasing and imaging) and recording metod of amplitude charasteristics (the principle of sample by sample).

It should be noted that the size of the error which occurs as a result of "*aliasing*" and "*imaging*" increases with the number of degrees to which we have divided the multistage filter because thus we increase the number of decimation/interpolation filters. This means that the decision in choosing the filter should be the result of a compromise between the requirements for the width of the transition zone filter, the amount of available hardware resources, as well as the size of the deviations that result from increasing the number of stages.

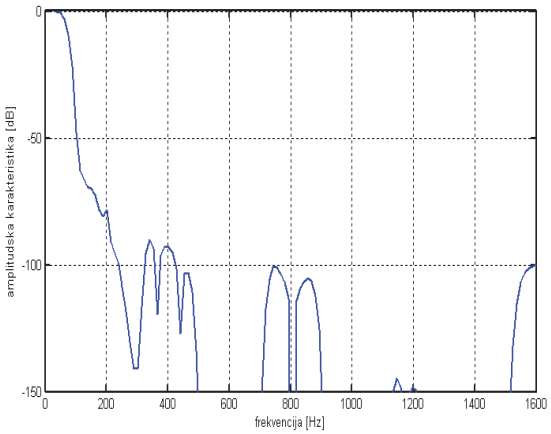

Fig. 8. Filter characteristics recorded sample by sample.

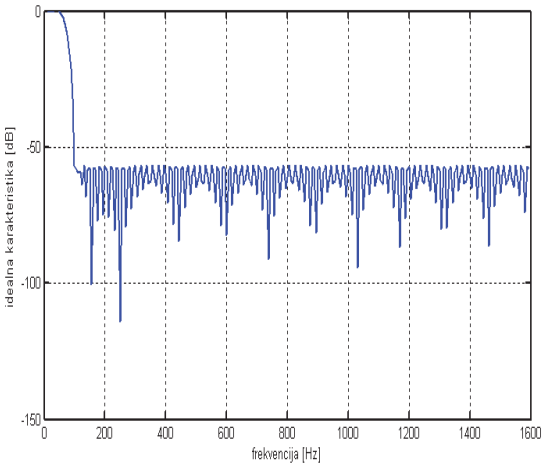

Fig. 9. Characteristics of an ideal filter.

Although the accumulation of errors resulting from multiple passing of a signal due to the effect of finite length codeword decreases by selecting the distributed arithmetic method for performing a filter, it is still necessary to check the influence of this effect because of the increased complexity of the structure and increased number of arithmetic operations.

V. Conclusion

The application of PI technique provides a solution for the rationalization of hardware resources for applications that involve or require the application of the same filter several times or hardware structure can be rearranged so that the signal processing is performed in parallel. Combining the operations related to the implementation of PI procedure together with the operation of the filter can be made by additional improvements of filters and the rationalization of hardware resources. Especially effective implementation can be done with multirate systems of signal processing. One of such examples is multistage multichannel filters with narrow transition zone, where the decimation/interpolation must be done in several steps. These filter structures are suitable for implementation with FPGA components because improvements of the software tools for designing filters that the manufacturers provide open opportunities for effective implementation of such filters.

References

[1] M. Ciric, "Realization of FIR digital filters using Pipelining/Interleaving," *ETRAN*, Herceg Novi, EK 2.2, 2007.

[2] M. Ciric, "Realization of FIR digital filters using Pipelining/Interleaving," *TELFOR* Beograd, pp 326-329, 2007.

[3] Z. Jiang and A. N. Willson, "Efficient digital filtering architectures using pipelining/interleaving," *IEEE Trans. Circuits Syst.*, vol. 44, no. 2, pp. 110-118, February 1994.

[4] J. F. Kaiser and R. W. Hamming, "Sharpening the response of a symmetric nonrecursive filter by multiple use of the same filter," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSIP-25, pp. 415-422, Oct. 1977.

[5] R. A. Valenzuela and A. G. Constantinides, "Digital signal processing shemes for efficient interpolation and decimation," *IEEE Proceadings, Pt. G.,* vol. 130, no. 6, pp. 225-235, December 1983.

[6] K. K. Parhi and D. G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters-Part I: Pipelining using scattered look-ahead and decomposition," *IEEE Trans. Acoust., Speech*, Signal Processing, vol. 37 no. 7, pp. 1099-1117, July 1989.

[7] V. Poučki, A. Žemva, M. Lutovac, and T. Karčnik, "Chebyshev IIR filter sharpening implemented on FPGA," in: *16th Telecommunication Forum Telfor,* 2008, pp. 432–435.

[8] System Generator for DSP, release 10.1, March, 2010, www.xilinx.com.

[9] Spartan 6 family overview, DS160 (v 1.5*)* August 2, 2010, Advanced product specification www.xilinx.com.

[10] F. J. Harris, Multirate Signal Processing for Communication Systems, *Prentice Hall PTR,* 2004, ch. 11.