Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

journal homepage: www.elsevier.com/locate/nima

FPGA implementation of digital filters for nuclear detectors

D. Alberto [a,b,*], E. Falletti [b], L. Ferrero [a], R. Garello [b], M. Greco [a], M. Maggiora [a]

[a] Dipartimento di Fisica Generale, Università di Torino, INFN-Sezione di Torino, Via Giuria 1-10125, Torino, Italy
[b] Dipartimento di Elettronica, Politecnico di Torino, Torino, Italy

ARTICLE INFO

Article history:
Received 25 July 2009
Accepted 18 September 2009
Available online 1 October 2009

Keywords:
FPGA
Digital filter
LMS
Nuclear detectors
Hadron physics

ABSTRACT

Innovative filtering structures have been implemented on Field Programmable Gate Arrays (FPGA) for processing signals generated in nuclear particle detectors. These digital filters will contribute to develop new triggerless data acquisition systems. In this paper, we describe the structures of these filters and compare their hardware-implemented performances with the results of software simulations. Particular attention has been paid to the consumption of FPGA internal components and to the maximum work frequency.

1. Introduction

In future Data Acquisition (DAQ) systems of nuclear particle experiments, all detectors will be self-triggering entities selecting the events of interest in order to overcome the limits of the traditional trigger system architectures. Thus, appropriate filtering becomes a key issue and both their theory and implementation are receiving a wide attention. New algorithms for the elaboration and the digital filtering of signals generated in nuclear particle detectors have been recently studied and optimized in [1]. These algorithms will enhance the feature extraction in DAQ systems of the next generation of physics experiments as the Panda experiment at FAIR [2,3].

When a charged particle is detected, a voltage (or a current) pulse is generated. Great attention should be paid to all those phenomena that can modify time and amplitude measurement, like baseline shift, pile-up effect, ballistic deficit and noise. In our analysis, the parameters to be optimized are the peak distortion (PD) and the *Signal-to-Noise Ratio* (SNR) correlated with the deposition energy and the detector efficiency, respectively.

Several classes of standard and adaptive digital filtering algorithms have been compared by simulation [1]. Standard filters were of Butterworth, Bessel and Chebyshev families with transfer functions from II to V order, whereas the adaptive one was the Finite Impulse Response (FIR) Least Mean Square (LMS) filter. After proper design and simulation, in order to perform digital signal processing on real-time signals, filtering must be implemented on an Application Specific Integrated Circuit (ASIC) or on a programmable board, as a Field Programmable Gate Array (FPGA). (A Xilinx Virtex 4 ML402 FPGA [4] will be considered in this paper). For this purpose, filter structures suitable for hardware implementation must be developed and optimized.

The paper is organized as follows. In Section 2, the simulated filtering structures and the hardware-implemented ones are introduced and compared. The implementation on FPGA is presented in Section 3. The paper is concluded with a discussion on the obtained results and an outlook on the further developments in Section 4.

2. Filtering structures

The FPGA-oriented structure of some digital filters for nuclear detectors will be presented in the following subsections.

Among the standard filtering algorithms, the Infinite Impulse Response (IIR) Low Pass (LP) Butterworth III order one has reached the best performances under the point of view of PD and SNR. Comparing by simulation the Butterworth III order filter with an adaptive four-coefficient LMS filter, this last one has presented a better behaviour in the evaluation of PD reduction and SNR enhancement. The Matlab and Simulink [5] simulation of these filtering algorithms has been discussed in [1].

However, the direct translation of those structures in VHDL leads to board consumptions higher than an FPGA-oriented implementation. For this reason, some changes were introduced and will be discussed and compared with the original simulated schematics.

[*] Corresponding author at: Dipartimento di Fisica Generale, Università di Torino, INFN-Sezione di Torino, Via Giuria 1-10125, Torino, Italy. Tel./fax: +39 011 670 7482.
E-mail address: diego.alberto@to.infn.it (D. Alberto).

0168-9002/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2009.09.049

2.1. Standard LP digital filters

The Butterworth III order normalized transfer function in the continuous complex frequency domain is

$$H(s)=\frac{1}{s^3+2s^2+2s+1}$$

To convert this analog transfer function in the digital domain, a digital transfer function $H(z)$ was used, obtained from bilinear transform of $H(s)$, with a cut-off frequency according to the bandwidth of the desired input signal (20 MHz)

$$H(z)=\frac{0.075+0.226z^{-1}+0.226z^{-2}+0.075z^{-3}}{1-0.827z^{-1}+0.515z^{-2}-0.087z^{-3}}$$

Once evaluated, the transfer function coefficients are fixed and time independent by definition of standard filter. In our simulations, $H(z)$ was implemented with a Direct Form (DF) II structure, as shown in Fig. 1. The numerator coefficients are the multipliers of the rightern subchain, the denominator ones of the leftern. This second half is retroacted and its contribution is added to the input signal.

Translating this structure in VHDL language for the implementation on the Virtex 4, signals coming from the lowest register, passing into the retroaction (leftern) subchain, encounter 2 multipliers and 4 adders (Fig. 1). This path is the so-called combinatorial path and represents the distance that a signal has to cover between two consecutive registers, or between one register and the filter output [6]. Long combinatorial paths take more time to execute; so they limit the maximum compile rate of the FPGA. The maximum work frequency is the inverse of this time.

As an example, in the case of the Butterworth III order filter, the implementation with a DF II structure leads to a combinatorial path that imposes a maximum work frequency of 34 MHz. Since this rate could not be enough to cope with the clock rate of some detectors of a nuclear experiment (as expected for Panda [3]), the filtering structures must be optimized to increase the FPGA maximum work frequency.

To improve this feature we have to break the longest combinatorial path. This was done by moving to a DF I transposed structure for translating the same transfer function (Fig. 2). In this case, the maximum combinatorial path is composed of only 1 multiplier and 2 adders (Fig. 2), thus the maximum work frequency rises up to 63 MHz.

2.2. Adaptive LMS digital filter

By definition, an adaptive filter evaluates and adapts its coefficients during a noisy signal acquisition. This feature is then very useful for non-stationary processes like nuclear particle signals. If a noisy process $x(n)$ is observed, it can be thought as the sum of a desired component $d(n)$ and a corrupting noise $v(n)$

$$x(n)=d(n)+v(n)$$

where $d(n)$ has to be estimated. If nothing is known about $d(n)$ or $v(n)$, it is not possible to separate the signal from the noise, except the case in which a reference signal is at disposal (e.g. through an active transducer in noise control applications for engines [7]). In nuclear detector applications, we cannot have a reference signal, so it is necessary to follow a different approach. The process $x(n)$ is delayed of n_0 samples and considered as the reference signal to estimate $d(n)$ through an adaptive algorithm. The parameter n_0 is chosen to obtain an uncorrelation of the noise component but a correlation of the desired one.

In the case of the LMS algorithm, the update coefficients equation assumes a simple form known as

$$\overline{w}_M(n+1)=\overline{w}_M(n)+\mu e(n)\overline{x}^*_M(n-n_0)$$

where $\bar{w}_M(n)$ and $\bar{w}_M(n+1)$ are vectors of the filter coefficients at time n and $n+1$, respectively; $\bar{x}^*{}_M(n-n_0)$ is the complex conjugate of the measurement at time $n-n_0$; $e(n)$ is the error at time n and represents the difference between the estimated filter output and the noisy input signal and n_0 is the introduced delay. The parameter μ is called the stepsize; it is a positive number that affects the rate at which the weight vector $\bar{w}_M(n)$ moves down towards a stable solution. This parameter is structure-dependent and has to be optimized according to our PD and SNR requirements.

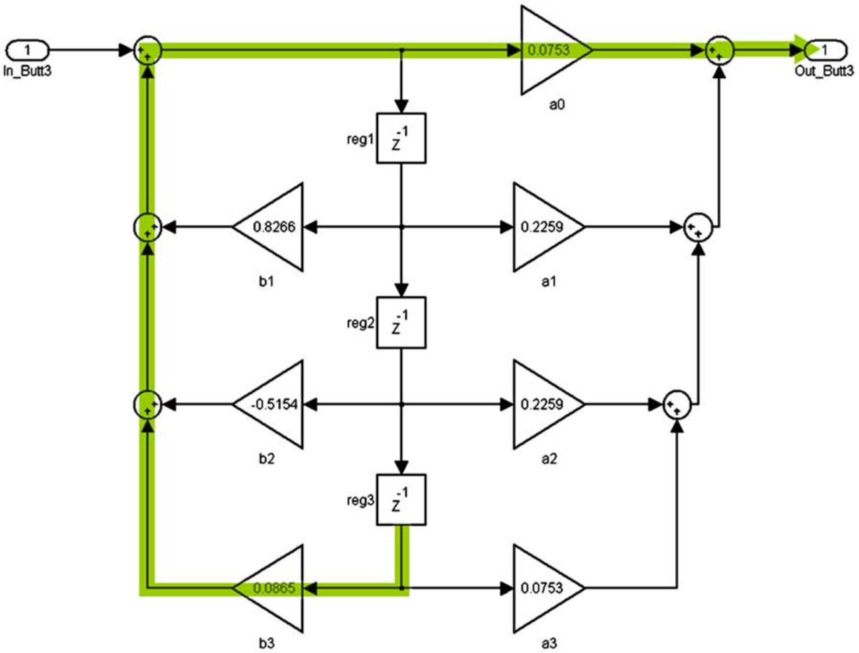

Fig. 1. IIR Digital Butterworth III order filter implemented with Direct Form II structure, Simulink schematics. The longest combinatorial path is highlighted in dark green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

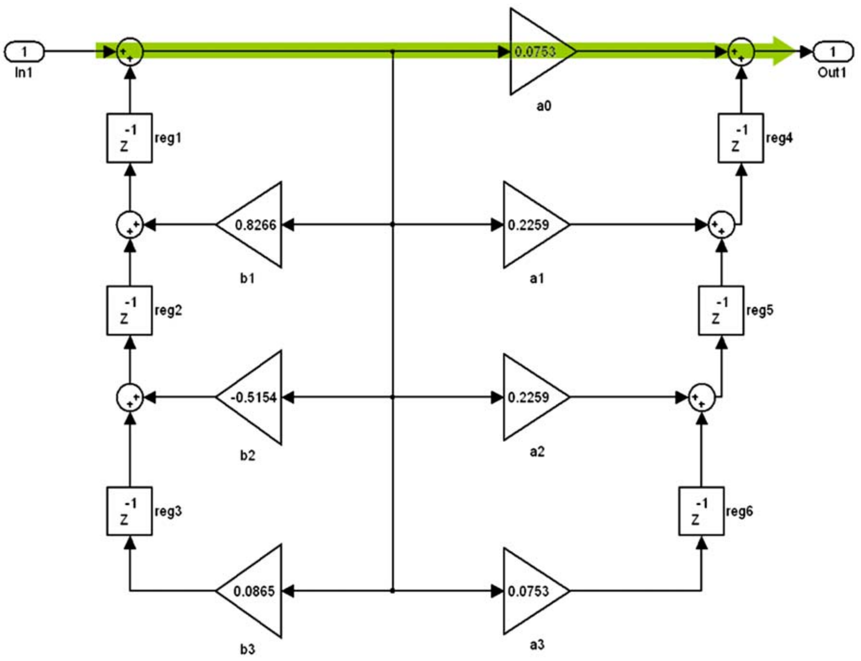

Fig. 2. IIR Digital Butterworth III order filter implemented with Direct Form I transposed structure, Simulink schematics. The longest combinatorial path is highlighted in dark green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

For a complete description of the LMS algorithm, the reader is referred to Ref. [8].

The structure used for the LMS simulation, translated in VHDL, presents a very long maximum combinatorial path (3 multipliers and 5 adders) leading to a maximum work frequency of 22 MHz (LMS1, Fig. 3a). Also in this case, it is possible to optimize the hardware structure and enhance this rate. In order to break the longest combinatorial path, some extra registers were added. The optimized adaptive filter structure (LMS2) is shown in Fig. 3b. With these changes in the structure, the maximum combinatorial path starts after the register under the first coefficient subchain, passes through 4 adders and the multiplier with the stepsize μ as second input. This path is shorter, thus the maximum work frequency increases to 58 MHz.

Since the LMS filter structure was changed, we found, as expected, a different value for the stepsize μ that optimizes our requirements on SNR maximization and PD reduction.

It is worth noting that for both LMS1 and LMS2 structures every adaptive coefficient is implemented with a dedicated subchain, fed by the multiplication of the stepsize μ with the difference between the noisy signal and the adaptive filter output (Fig. 3a and b).

3. Results and discussion

In this section, the performance of the FPGA digital filtering implementation is discussed with reference to nuclear detector requirements. For the sake of comparison with the simulation results [1], the same noisy signal has been used for the FPGA filtering. The information pulses are modelled with a series of successive finite support waveforms, with very narrow time duration (0.5 ns) and random times of arrival. No specific assumptions had been made about the mathematical model of the times of arrival, since it was intended to perform a general analysis not referred to a specific detector. An additive White Gaussian Noise (WGN) signal had been added to the desired signal. This represents a realistic situation where the noise has a significantly wider bandwidth than the signal of interest.

3.1. Filter performance

In Fig. 4 the simulated noisy digital signal and the superimposed desired one are shown.

In Fig. 5 and in Table 1, the most relevant simulation results are presented. As reported, the LMS filter introduces a much lower PD and a slightly higher SNR than Butterworth III order filter. The PD is evaluated for the filtered peak featuring the highest distortion (the first in this analysis) [1].

The direct VHDL implementation of these structures leads to the same values of SNR and PD of Matlab-simulated filtering algorithms. The plot of the filtered signals is presented in Fig. 6. In the VHDL translation both Butterworth and LMS filtered signals present a delay of 2 sampling periods with respect to the Matlab-simulated ones. Indeed, in every structure of the previous section, the two registers for input and output storages are not shown. This corresponds to a usual FPGA implementing rule chosen to avoid a wrong numerical representation of signals when high work frequencies are involved.

As pointed out in the previous section, a direct VHDL translation of the Butterworth DF II structure (Fig. 1) leads to a low maximum work frequency. Thus, we adopted the translation of the DF I transposed structure (Fig. 2). The processing and the performances of PD and SNR do not change for the Butterworth filter because the same transfer function has been translated. A different consideration has to be made for the VHDL translation of the LMS filter. Since the LMS1 structure worked at low frequency, some registers were added to break the too long combinatorial path. This insertion changed the structure (LMS2) giving higher work frequency. Therefore, the LMS2 structure leads to a different best stepsize value, but also to a higher PD and a lower SNR than the Matlab-simulated LMS1 structure (Table 1). This is the price that must be paid to work at a frequency two times and a half higher. Anyway, LMS filter continues to match better the requirements related to the energy resolution and

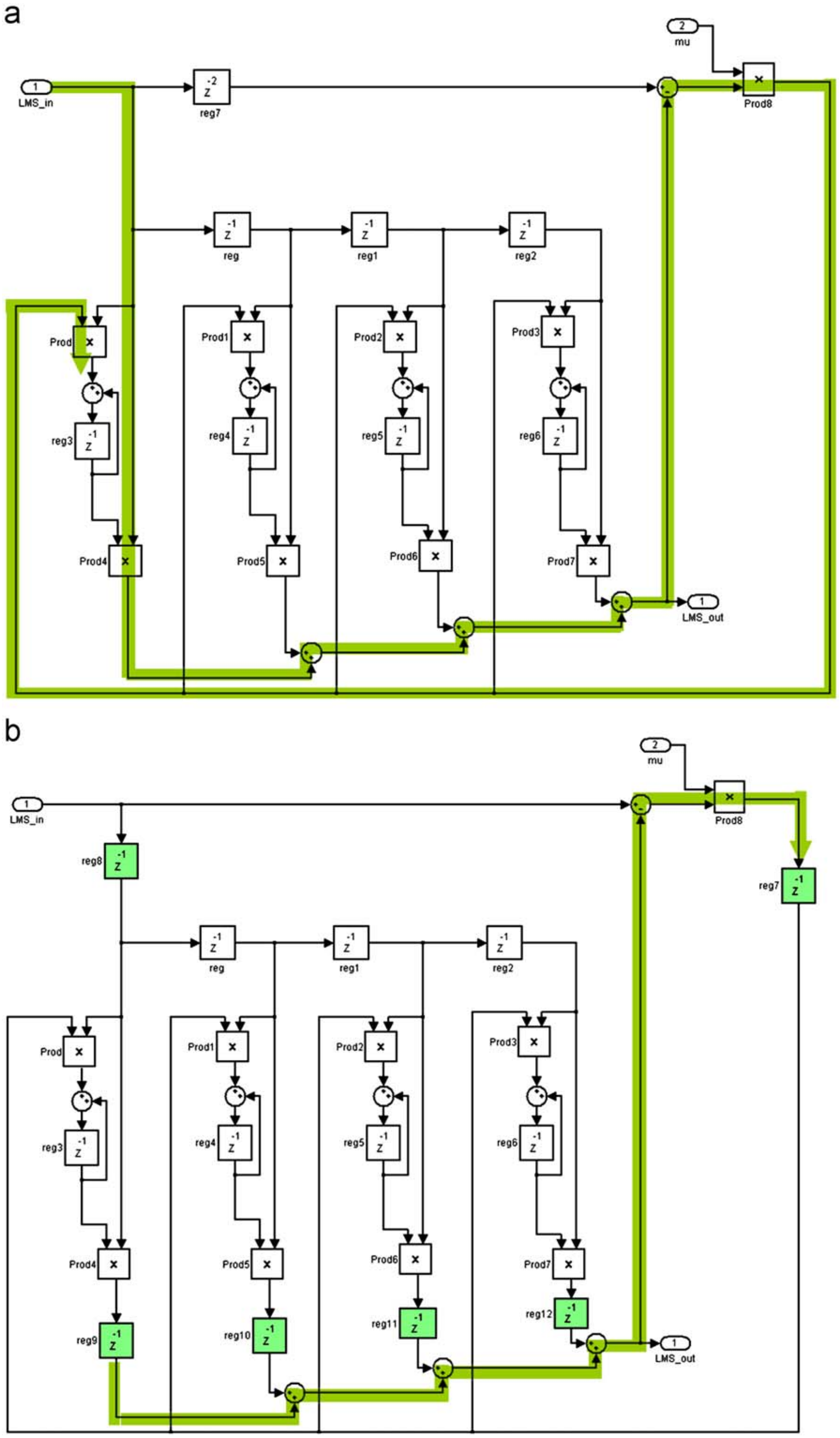

Fig. 3. (a) FIR LMS 4-coefficients structure Matlab-simulated (LMS1), Simulink schematics. The longest combinatorial path is highlighted in dark green, mu is the stepsize and (b) FIR LMS 4-coefficients structure implemented for FPGA board (LMS2), Simulink schematics. The longest combinatorial path is highlighted in dark green, mu is the stepsize, added registers are highlighted in light green. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

detector efficiency than the Butterworth one also in the implemented FPGA form (Table 2).

In Fig. 7 the optimized Butterworth (DF I transposed) and LMS (LMS2) filter outputs are shown. The LMS2 filter shows a delay of 8 sampling periods, with respect to the LMS1 structure (Fig. 6), due to the added registers that shorten the combinatorial path. The Butterworth FPGA filtered signal has no delay because no register is added.

3.2. Power consumption

In the implementation of filtering structure on FPGA, it is very important to evaluate the consumption of the board internal components (e.g. the percentage of utilized memory slices and digital signal processors and the number of needed logic levels). This analysis is indeed critical if we consider the total number of boards that should be used in a nuclear particle experiment. We

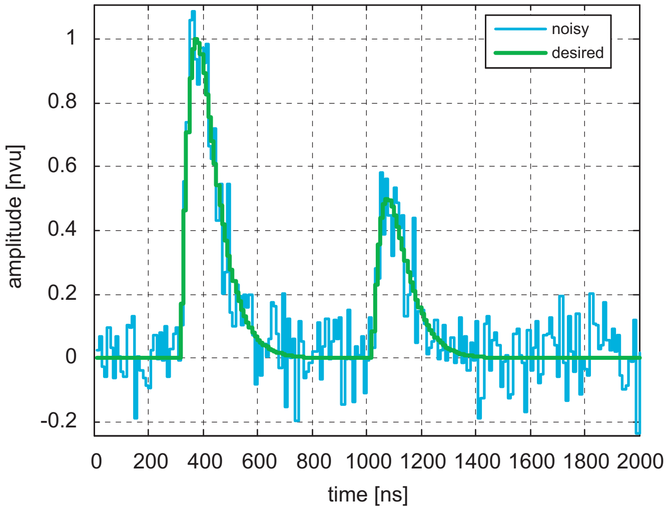

Fig. 4. Noisy vs desired digital signal, amplitude in normalized voltage unit (nvu).

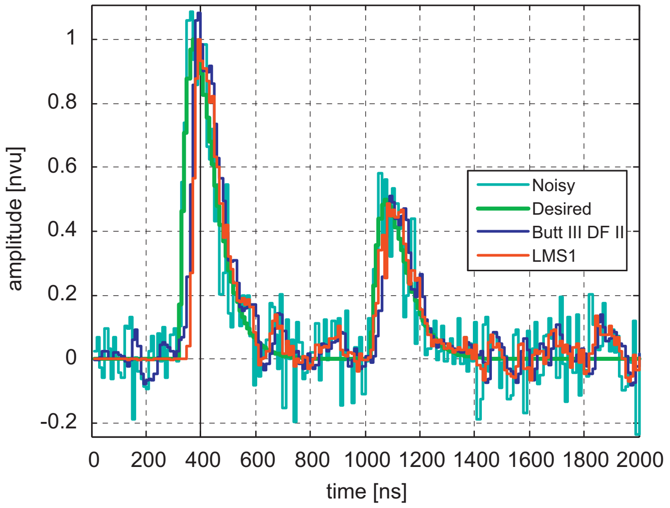

Fig. 5. Matlab-simulated filter comparison.

Table 1
Performance of Matlab-simulated Butterworth and LMS filters.

Filter type	Stepsize *μ*	*PD (%)*	*SNR improvement (dB)*
Butt III DF II		8.34	5.76
LMS1	0.18	0.06	6.57

have compared the different filtering structures described in the previous section also under this point of view.

As reported in Tables 3 and 4, the DF I transposed and the LMS2 structures optimized for FPGA reach not only higher maximum work frequency but also need lower components for their implementation than the simulated and directly VHDL-translated filtering structures (DF II and LMS1).

3.3. *Maximum work frequency*

The structures were optimized to obtain the maximum work frequency with Xilinx Virtex 4 board, doubling at least the rate.

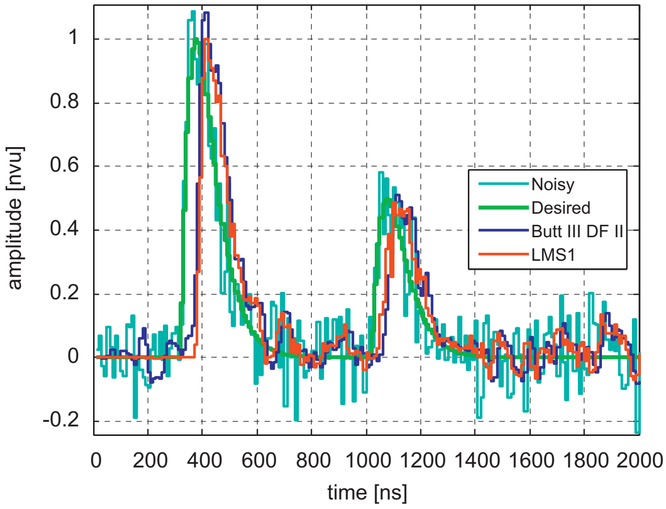

Fig. 6. Matlab-simulated structures implemented on FPGA.

Table 2
Performance of FPGA-optimized Butterworth and LMS filtering structures.

Filter type	Stepsize *μ*	*PD (%)*	*SNR improvement (dB)*
Butt III DF I		8.36	5.76
LMS2	0.27	0.49	5.91

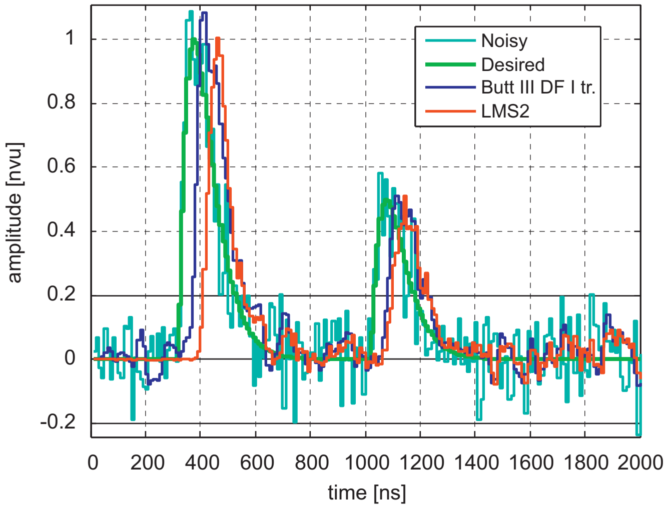

Fig. 7. FPGA-optimized filtering structures.

Table 3
Compared performance for different structures of Butterworth filter implemented on a Xilinx Virtex 4 ML402.

Butterworth	*# Memory slices (%)*	*# DSP (%)*	*Logic levels*	*MAX work freq. (MHz)*
DF II	< 1	14	30	34
DF I transp.	< 1	10	24	63

However, some detectors can work at a frequency higher than 60 MHz. The same structures were then implemented on a Virtex 5 FPGA using a Xilinx ISE tool [6] to verify the possibility to increase the FPGA maximum work frequency. The results are

Table 4
Compared performance for different structures of LMS filter implemented on a Xilinx Virtex 4 ML402.

LMS	*Stepsize* μ	# Memory slices (%)	# DSP (%)	*Logic levels*	MAX work freq. (MHz)
LMS1	0.18	<1	18	37	23
LMS2	0.27	<1	18	24	58

Table 5
FPGA consumptions on different boards: Xilinx Virtex 4 XC4VSX3 −10 (V4) and Virtex 5 X5VSX35T-3 (V5).

Filter	*FPGA*	*# Memory slices (%)*	# DSP (%)	*Logic levels*	MAX work freq. (MHz)
Butt III DF I tr	**V4**	<1	10	24	63
	V5	<1	5	40	130
LMS2	**V4**	<1	18	24	58
	V5	<1	18	28	116

summarized in Table 5, for comparison. The maximum rate can be easily doubled using the Virtex 5, a board that has already been in FPGA marketplace for some years and is relatively low cost. This means that the FPGA-optimized structures are ready to be implemented and work at rates higher than 100 MHz.

4. Conclusions and outlook

We have translated and implemented standard and adaptive digital filters on an FPGA for real-time processing of nuclear detector signals. A direct VHDL translation of filtering structures previously simulated [1] leads to a high FPGA consumption and a low maximum work frequency. We optimized the implementations to enhance the work frequency in order to cope with the foreseen high-rate data acquisition of nuclear detectors. These different implementations introduce no significant variations in terms of PD reduction and SNR enhancement for the standard IIR Butterworth III order LP filter, while the adaptive LMS with our new structure (LMS2, Fig. 3b) presents slightly lower performances but a higher maximum work frequency than the simulated implementation (LMS1, Fig. 3a). We can conclude that the requirements on PD and SNR are matched better both in simulation and in FPGA implementation by the LMS filter.

The LMS FPGA-oriented structure will be used in a filtering system for the data acquisition of Si detector signals in a real data stream at LNS-INFN [9] to verify the hardware-implemented foreseen behaviour.

Acknowledgements

The authors would like to thank Alberto Dassatti, Guido Masera and Luca Toscano for the useful discussions during the VHDL translation of the filtering structures.

References

[1] D. Alberto, et al., Nucl. Instr. and Meth. A 594 (2008) 382.
[2] http://www.fair-center.org.
[3] Physics Performance Report for PANDA: Strong Interaction Studies with Antiprotons, arXiv:0903.3905v1 [hep-ex].
[4] http://www.xilinx.com/.
[5] MATLAB, The language of technical computing, 〈http://www.mathworks.com〉.
[6] L. Scheffer, L. Lavagno, G. Martin, Electronic Design Automation for Integrated Circuits Handbook, Taylor & Francis, 2006.
[7] S.J. Elliot, P.A. Nelson, Sig. Proc. Mag. 10 (4) (1993) 12.
[8] M.H. Hayes, Statistical Digital Signal Processing and Modelling, John Wiley & Sons, 1996.
[9] Laboratori Nazionali del Sud (LNS)-INFN 〈http://www.lns.infn.it/〉.