Nuclear Instruments and Methods in Physics Research A 735 (2014) 297–303

Contents lists available at ScienceDirect

Nuclear Instruments and Methods in Physics Research A

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

Implementation of a real-time adaptive digital shaping for nuclear spectroscopy

Alberto Regadío [a,b,*], Sebastián Sánchez-Prieto [a], Manuel Prieto [a], Jesús Tabero [b]

[a] Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain
[b] Electronic Technology Area, Instituto Nacional de Técnica Aeroespacial, 28850 Torrejón de Ardoz, Spain

ARTICLE INFO

Article history:
Received 4 January 2013
Received in revised form
21 August 2013
Accepted 10 September 2013
Available online 4 October 2013

Keywords:
Spectroscopy
Shaping
Adaptive
Signal processing
FPGA
Nuclear detectors

ABSTRACT

This paper presents the structure, design and implementation of a new adaptive digital shaper for processing the pulses generated in nuclear particle detectors. The proposed adaptive algorithm has the capacity to automatically adjust the coefficients for shaping an input signal with a desired profile in real-time. Typical shapers such as triangular, trapezoidal or cusp-like ones can be generated, but more exotic unipolar shaping could also be performed. A practical prototype was designed, implemented and tested in a Field Programmable Gate Array (FPGA). Particular attention was paid to the amount of internal FPGA resources required and to the sampling rate, making the design as simple as possible in order to minimize power consumption. Lastly, its performance and capabilities were measured using simulations and a real benchmark.

1. Introduction

Nuclear spectroscopy is the term used to describe the electronic systems employed to study elemental particles and nuclear physics. These techniques have become extensively used in many fields, such as radiation detection or space particle telescopes. Typical elements of these devices include high-speed detectors to measure rates and energy, preamplifiers, shapers, discriminators, counters and pulse height analyzers [1]. From the point of view of signal transformation, the weak and fast pulse provided by the detectors is amplified and filtered to optimize detection. This process has traditionally been implemented by means of analog electronic elements such as operational amplifiers, comparators and peak detectors [2].

The basic detection chain for a particle detector using analog electronics is shown in Fig. 1(a). The pulse provided by the detector is integrated into the preamplifier to obtain a fast slope signal followed by a long tail. This signal is typically filtered to obtain a quasi-Gaussian shaped output, a procedure usually known as pulse shaping, the purpose of which is to increase the Signal-to-Noise Ratio (SNR) in order to optimize detection [3]. In several detectors, the pulse charge is proportional to the energy of the detected particle. Also, the charge of the pulse is proportional to its amplitude. In these situations, a peak detector may be used to obtain a value proportional to the charge.

Nowadays, digital signal processing is also used in particle detectors [4,5]. The proposed detection chain for a particle detector using digital signal processing is shown in Fig. 1(b). When the same detector and preamplifier from Fig. 1(a) are used, an Analog-to-Digital Converter (ADC) should be added after these stages to digitalize the signal. In addition, an optional analog filtering stage before the ADC could also be added. One objective of this study was to use this algorithm in a future satellite payload, an environment in which power requirements are very restrictive. Thus, one main objective was to reduce sampling frequency as much as possible in order to reduce power consumption. With this goal in mind, a passive low-pass filter was included between the preamplifier and the ADC. This was an anti-aliasing filter, which is typically used before an ADC, and its inclusion made it possible to fulfill the Nyquist criteria. The pole generates a low-pass filter that avoids the aliasing effect and thus, the sampling frequency could be decreased [6]. After the ADC, all processing, including shaping and peak detection, is performed using digital elements.

The ideal shaping for a detector depends on the input signal and the predominant noise [7]. Thus, specific techniques are used to synthesize various pulse shapers and optimize detection by maximizing the SNR [8,9]. However, the noise spectrum can vary

* Corresponding author at: Department of Computer Engineering, Space Research Group, Universidad de Alcalá, 28805 Alcalá de Henares, Spain. Tel.: +34 669013583.
E-mail addresses: aregadio@srg.aut.uah.es, alberto.regadio@edu.uah.es, aregadio@isdefe.es (A. Regadío), ssanchez@srg.aut.uah.es (S. Sánchez-Prieto), mprieto@srg.aut.uah.es (M. Prieto), taberogj@inta.es (J. Tabero).

0168-9002/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.nima.2013.09.063

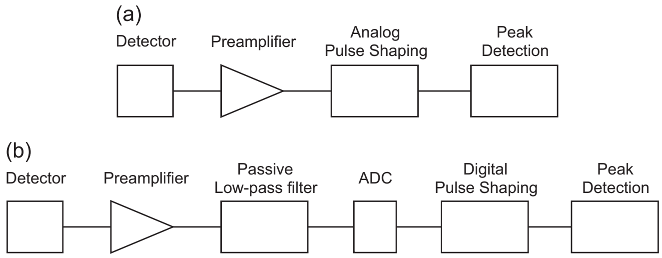

Fig. 1. (a) Typical particle detection chain based on analog electronics. (b) Particle detector chain based on mixed analog-and-digital electronics.

over time for several reasons (e.g. accumulated radiation on the detector and damage to the detector structure [1]), whereas these techniques do not usually consider noise variations in real-time [4,5]. To avoid this drawback and allow the use of self-adjusting transfer functions, the use of adaptive shapers was proposed.

Several algorithms exist for performing adaptive shapers (e.g. Least-Mean-Square (LMS), Digital Penalized LMS (DPLMS), Wiener algorithm and Discrete Fourier Transform (DFT) method [10,11]). The LMS algorithm finds the filter coefficients associated with producing the least mean squares of the difference between the desired and the actual signal. The DPLMS is an improvement on the LMS algorithm proposed in Ref. [10]. The Wiener algorithm minimizes the error by equating each partial derivative to zero. Finally, the DFT method is based on dividing the desired signal by a given input signal. The shape can also be adapted by modifying any of its parameters (e.g. the algorithm presented in Ref. [12] searches for the optimum shaping time).

In this paper, we propose a technique for automatic synthesis of digital pulse shapers for high-resolution spectroscopy using an adaptive algorithm based on the DFT and LMS methods. The proposed algorithm is suitable for real-time implementation and only requires simple hardware and an example pattern for training. The main advantage of this technique is that it has the capacity to transform any input signal into any output signal with the desired features (including typical shapes such as trapezoidal, triangular or cusp-like outputs).

In Ref. [10], algorithms for adaptive digital shaping with a good performance were presented; however, implementation of most of these algorithms is quite complex. This complexity would also affect power consumption in a real implementation, which should be kept as low as possible. In addition, in the majority of these algorithms, the entire input signal must be known before calculating the filter coefficients, whereas the algorithm proposed here has the capacity to start adjusting the shaper coefficients as soon as the first pulse sample arrives. To sum up, the proposed design is a synthesizable algorithm that takes into account area, power consumption and above all, functionality.

2. Proposed adaptive algorithm

As explained in the Introduction, the utilization of different shaping techniques makes it possible to improve resolution in particle detectors [13]. Consequently, the aim of this study was threefold: to develop an automatic desired pulse shaping operation, to reduce input noise and to solve the pile-up effect. Of the design possibilities for obtaining the desired results, adaptive algorithms were selected because they have the capacity to automatically adjust the coefficients of a linear system and because they are easy to implement in digital hardware.

The proposed shaper works as an adaptive digital Finite Impulse Response (FIR) filter. The output of a FIR filter is given by the following expression:

$$y[n] = \sum_{k=0}^{N-1} a_k x[n-k] = h[n] * x[n] \tag{1}$$

where $x[n]$ is the input signal, $y[n]$ is the output signal, a_k are the coefficients of the filter, $h[n]$ its impulse response and N its order.

Two parameters related to digital filters are the shaping time τ_s, which is defined as the duration of the shaper output signal $y[n]$, and the ADC sampling period T_s. These are related to N as follows:

$$N = \frac{\tau_s}{T_s} \tag{2}$$

One method to adjust the a_k coefficients, which is described in Ref. [10], uses an iteration method to search for the solution that minimizes the error according to this formula:

$$h(z) = \frac{x(z)}{d(z)} \tag{3}$$

where $x(z)$ and $h(z)$ are the z-transform of $x[n]$ and $h[n]$, respectively, and $d(z)$ is the z-transform of the desired output signal $d[z]$. This is one of the simplest methods for adapting a shaper. Thus, implementing it in an FPGA yields a low occupation area and low power consumption, one of the objectives of this study. However, several drawbacks of this method are described in Ref. [10], including output signal undershoots and the possibility that the filter order is infinite (i.e. it does not belong to the FIR class). To minimize the effects of these drawbacks and obtain better results, the lengths of $x[n]$ and $d[n]$ should be equal or the length of $d[n]$ should be slightly smaller than the length of $x[n]$. This is because when $x[n]$ is equal to zero during several cycles at the end of the pulse, the shaper needs a_k coefficients with high values to obtain $d[n]$ at the output. Therefore, the coefficients at the end are much greater than those at the beginning, producing an increase in the undershoot and a poor adaptation depending on $x[n]$.

One method to mitigate noise and phase shift is to replace $x[n]$ by the mean value of Λ input signals; however, this does not work properly unless the input pulses have the same shape and duration. Furthermore, the sampling frequency should be sufficiently high to ensure a negligible phase shift between pulses. Thus, there are many applications in spectroscopy where this procedure cannot be applied, and Λ shall be equal to one.

3. Design

As indicated in the Introduction, the proposed design was conceived with two goals in mind: simplicity and low power consumption. Therefore, it was important to reduce the functional units and working frequency as much as possible. Furthermore, both the shaper and peak detector were implemented in the same FPGA.

In this design, the output pulse $y[n]$ is converted into a desired pulse $d[n]$ by adjusting the shaper coefficients in real-time as a function of an input signal $x[n]$.

Both $x[n]$ and $d[n]$ are introduced into the shaper at the same time. According to Eq. (2), both N and T_s must be set manually before the shaper adjusts its coefficients. The correct T_s depends on the duration of the input pulses.

In order to synchronize $x[n]$ and $d[n]$, this last signal is provided when $x[n]$ is greater than a predefined learning threshold, which should be equal to or greater than the pulse-height analysis threshold. Thus, the shaper only "learns" when a desired input pulse arrives, avoiding possible false triggers due to noise and undesired pulses.

Fig. 2 shows a detailed diagram of the shaper. The *den* signal is used to enable the learning process. The error $e[n]$ is obtained by subtracting the reference signal $d[n]$ from the output signal $y[n]$