Hardware Implementation for Pileup Correction Algorithms in Gamma Ray Spectroscopy

Manar M. Ouda, Mohamed S. El_Tokhy,
Sherief Hashima, Imbaby I. Mahmoud
Engineering Department, Nuclear Research
Center, Egyptian Atomic Energy Authority,
Inshas, Cairo, Egypt

Mohamed Amal-Eldin, Nesreen I. Ziedan
Computer and Systems Engineering Department,
Faculty of Engineering,
Zagazig University,Zagazig, Egypt

ABSTRACT

One of the main detection duties in spectroscopy system is pileup detection of the location of the maximum peaks utilizing a local extreme method. This paper presents the hardware implementation of two pile up recovery algorithms. The first algorithm utilizes a peak detection algorithm, while the second one utilizes a peak sensitivity algorithm. The two algorithms are designed and evaluated by compiling MATLAB into field programmable gate array (FPGA) using Xilinx system generator (XSG). The tested signal is captured by analogue to-digital converter (ADC) with sampling rate of 16MS/s. The results confirm that the first algorithm is better than the second one due to its simple architecture, which leads to faster processing speed.

Keywords

Pulse pile up, Digital signal processing, XSG, FPGA

1. INTRODUCTION

One of the key problems in gamma-ray spectroscopy (GRS), regards pulse processing, is the lost data due to pulse pileup occurrence. Pileup happens when two pulses overlap at high count rate, particularly for crystal NaI(Tl) scintillator detectors. The overlapped pulses data contents are either lost or add noise to the system, when no actions are taken to separate these pulses [1], [2]. In hardware design, pulse pileup recovering is achieved using XSG, which is a system-level modelling tool that support FPGA hardware design [3].
Pulse pileup recovery algorithms are implemented mainly either by hardware or software approaches. Most of recent pileup recovery algorithms are based on FPGA technology as they offer several advantages compared with their analogue counter parts. These advantages include simple and fast implementation of complex pile up recovery algorithms used for signal processing. The authors in [4], [5] discussed three different approaches for peak pileup recovery problem in GRS. The first approach depends on the inverse matrix algorithm and it is more accurate than other algorithms. The second approach is called direct search algorithm. It presents the best recovered pulse width result. The least square algorithm is the third one, which acquires the highest pulse width error than the other algorithms. The algorithms are evaluated by means of parameters error and fitting error calculations. In [6], the authors studied a digital pulse pile-up correction algorithm that can recover overlapped pulses up to 80% at high count rates. In [2], [7], the authors discussed the principle of operation and the development of a completely digital spectrometer system with a NaI(TI) detector suitable for high-quality GRS at very high counting rates. The central part of the system, the high-speed parallel pulse processor, is realized using an FPGA device. In [8], the authors proposed high yield pileup event recovery (HYPER) method that corrects multiple pulse pileup events. The correction is done by computing a weighted sum of an overlapping pulse and subtracting the weighted sum of the previous pulses, decreased by a time-decay term. The digital pulse processing for radiation energy measurement is discussed and implemented using FPGA including shaping algorithm, trigger generation for radiation event, pile-up detection and rejection of pileup events and baseline restoration method [9]. An FPGA does not have any hardwired logic blocks that would defeat the field programmable aspect of it. However, microprocessors have a settled set of orders, which the programmers want to learn with a specific target to make the fitting working program. The primary distinction amongst FPGAs and software is the complexity. Microprocessors have a tendency to be more complicated than FPGAs.

The principle objective of this work is to develop a digital FPGA based signal processing algorithms for a pulse pile-up correction. This will enable us to discover experiments with higher count rates [6]. Hardware modules with reconfigurable devices, are capable of transferring and processing data in real time. FPGAs combine the inherent parallel nature of hardware with the flexibility of software in that their functionality can be reprogrammed or reconfigured. Early FPGAs were quite small in terms of low memory, so they tended to be used primarily for providing flexible interconnect and interface logic. The problem of FPGA hardware implementation may be overcome by utilizing efficient design skills such as parallelism and pipelining concepts. In addition, FPGA minimizes the time-to-market cost, enables rapid prototyping of complex algorithms, and simplifies debugging and verification.

This paper is divided into five Sections as follows. In Section II, the system components and its main responsibilities are explained in details. Section III explains the implementation of the two pileup recovery algorithms using FPGA with XSG. The two algorithms test results are verified in Section IV. Finally, the conclusions are presented in Section V.

2. SYSTEM COMPONENTS

A scheme of the framework is depicted in Fig. 1. The system main components are as follows: a scintillation detector NaI (TI), which used to recognize the input signal from radiation source ^{137}Cs; the main block of the digital data acquisition system (DAS), which consists of: amplifier, high speed ADC digitizer, and monitoring or personal computer (PC) for data reconstruction and display [4], [10].

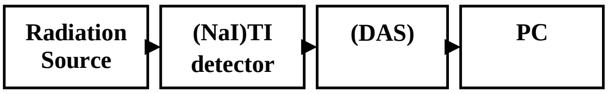

Fig 1: The schema of (DAS) for pulse height analysis of GRS system

3. DESIGN AND IMPLEMENTATION OF THE PROPOSED APPROACH USING FPGA

The well-organized quick prototyping system needs an improved environment for the hardware design platform. The applied tools are MATLAB Simulink, XSG 11.4 [11]. The design methodology with XSG is shown in Fig. 2 [12].

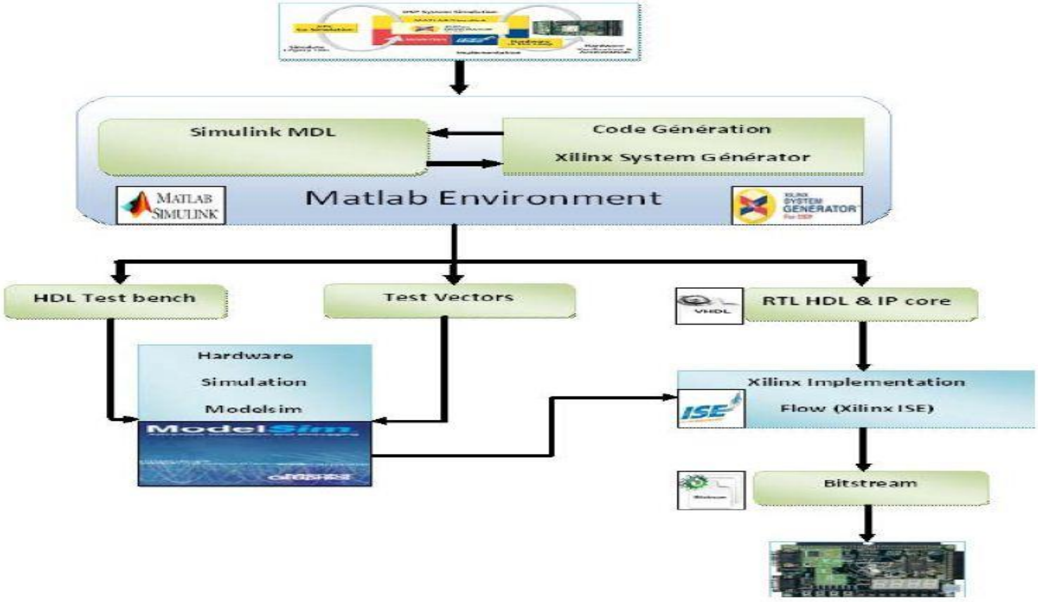

Fig 2: Design Methodology with XSG [12]

The XSG condition allows for the Xilinx line of FPGAs to be interfaced with Simulink. By using XSG, the two algorithms are implemented in an FPGA target, which determines the position and amplitude of the peaks. Due to using delay blocks in XSG design, a time interval between simulation and response are found which is called latency [11].The first algorithm locates the applicable peaks in crude information and the second algorithm finds the max peak of information (local_extrema) [13].

3.1 Peak detection method implementation using FPGA with XSG

The peak detection method main target is to find the relevant peaks in raw data. Signals passing from Simulink blocks to Xilinx blocks must go through Xilinx gateway, which enforces these constraints. The gateway-in block exposes the controller, which defines the signal type as shown Fig. 3. In this algorithm, each three consecutive values are compared. The peak is detected when the value in the middle is greater than the previous and the following. The sample time must be a positive number, indicating the rate at which the input signal is sampled. The amount of FPGA hardware used is directly related to the data width, so for reasons of cost, circuit speed, and power dissipation, it is better to request only the precision required for a particular application. The algorithm can be easily constructed using a multiplier and accumulator block as shown in Fig. 4. According to its configuration, the gateway-out block can either be defined as an output port in the top level of the HDL design generated by SG, or as a test point that will be trimmed from the hardware representation.

The advantages of this algorithm are simplicity, less time consumption, easy to program, and it has small number of logical elements or components.

a) ***Capture radiation signals***
b) ***Read signals from Matlab workspace***
c) ***Data type Conversion for inputs of type Simulink integer to Xilinx fixed point type***
d) ***Adjust the i/p gate from Matlab workspace that control the next stages***
e) ***Make two relation blocks to compare the three consecutive values together, when (1st value < 2nd value >3rd value) a peak is detected.***
f) ***Connect the o/ps of two relation blocks of step (e) by inputs of AND logical operators***
g) ***The o/ps of AND gate is considered as selector of 2 to 1 multiplexer (MUX), if this o/p is 1 then this position is a peak and is selected by (MUX) otherwise the o/p of (MUX)***
h) ***Get Peak Values***

Fig 3: The proposed peak detection algorithm using XSG

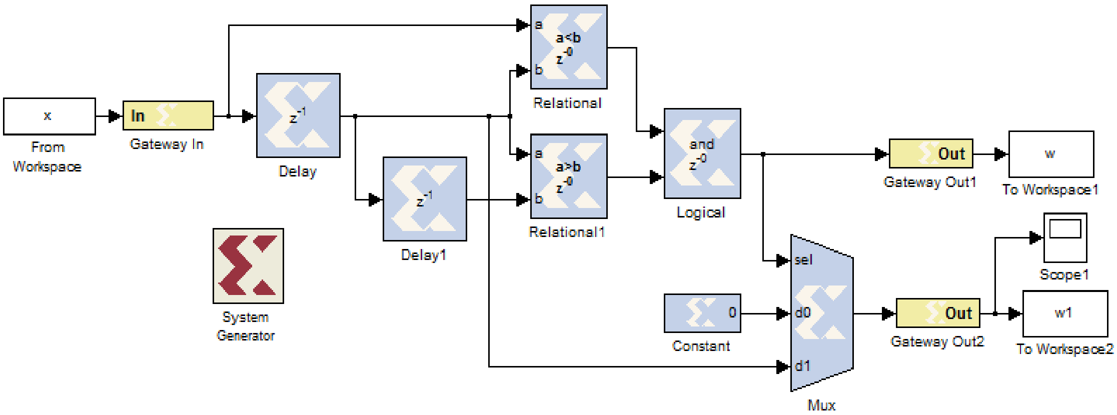

Fig 4. The implementation of peak detection algorithm using XSG

3.2 Peak sensitivity algorithm using XSG implementation

a) ***Capture radiation signals***
b) ***Read signals from Matlab workspace***
c) ***Determine No. of data bits***
d) ***Convert i/ps of type Simulink integer to Xilinx fixed point type***
e) ***Subtract each two adjacent peaks using delay of i/p***
f) ***Multiply each two adjacent difference values in e) using a delay of difference***
Multi value = [def (1: end-1).* dif(2:end)]
g) ***Create empty matrix and if***
1) ***multi value ≤ 0 set 1 in matrix***
2) ***else set 0 in matrix***
i) ***Multiply each two value in matrix by difference of data***
j) ***Extract max peaks that be***
Max peak =1 + find (diff >1)

Fig 5: The proposed peak sensitivity method using XSG

The first step of this algorithm is to capture the radiated signal and read it from a Matlab workspace. Secondly, the number of data bits is determined. After that the inputs of type Simulink integer are converted into Xilinx fixed point type as shown in Fig. 5.

The proposed peak sensitivity method using XSG are shown in Fig. 6. The algorithm is more flexible, and its implementation is reconfigurable by the user or other applications. Moreover, it provides potential for high control performance. The main disadvantages of this algorithm are the computational complexity due to multiplication, division, and the very high number of used logical elements.

4. RESULTS AND DISCUSSION

The number of latency occurs when two coincidental pulses calculation. Coincidence checker is a separate FPGA unit that allows events from all detectors and denotes pulses, which are not in the same place of the pile-up window. The experimental results of the two methods are achieved by using XSG.

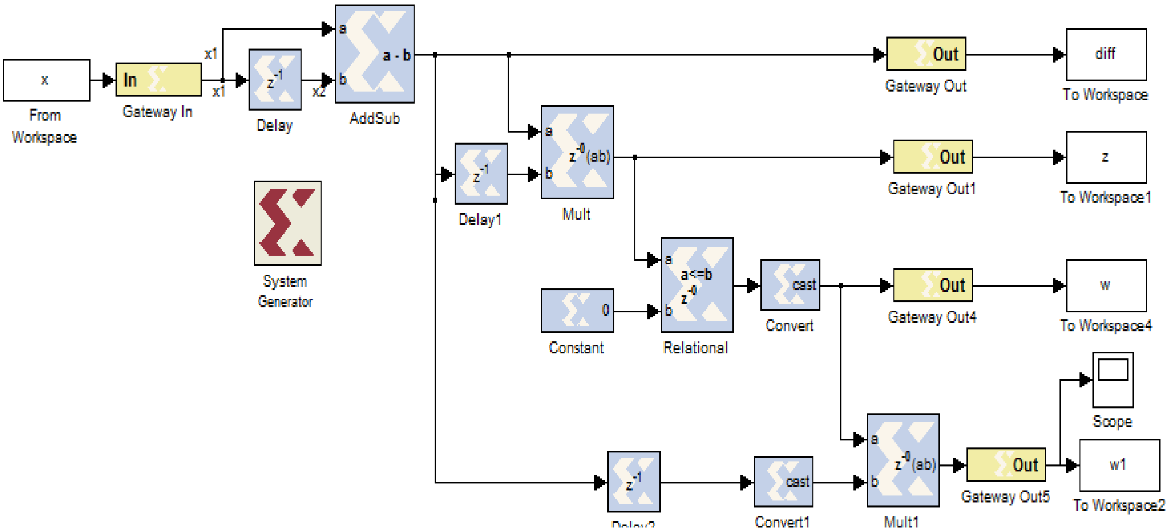

Fig 6. The implementation of peak sensitivity method using XSG

Table 3. Shows values ratio for the peak and position of the 2nd algorithm

K	X (i/p from workspace)	M (dif data)	Z	W	M1 (dif1 data)	M1>0 (W1= M1* W)	Position= 1+findW1
1	130	0	0	1	70	70	2
2	200	130	0	1	-10	0	
3	190	70	9100	0	-50	0	
4	140	-10	-700	1	40	40	5
5	180	-50	500	0	-20	0	
6	160	40	-2000	1	10	10	7
7	170	-20	-800	1	-20	0	
8	150	10	-200	1	80	80	9
9	230	-20	-200	1	-20	0	
10	210	80	-1600	1	0	0	

4.1 Experimental results of the first algorithm

The first algorithm compares each of three consecutive values together, so that the value in the middle is greater than the previous one and the following one. The sample time must be a positive number, indicating the rate at which the input signal is sampled. The algorithm starts by getting the maximum values of peaks, at the start time, for multiple peaks as shown in Fig. 4. From the results, the third sample is considered as the start of processing. The previous two samples, which are defined as latency samples are neglected. This is because they appear due to using two delay blocks, ie. system delays using two samples as listed in Table 1. The parameters of the peaks are calculated and listed in Table 2. The output scop1 in Fig. 4 is processed using XSG to determine the maximum of recovered peaks as shown in Fig. 7. The output of scop1 is converted into MATLAB as shown in Fig. 8.

Table 1. Estimated amplitude and position of the recovered peaks of the 1st algorithm by using XSG

Time (Position(μs))	i/p data	w	w1 (Max. Peak(V))
1	124	0	0
2	149	0	0
3	139	0	0
4	125	1	149
5	102	0	0
6	89	0	0
7	106	0	0
8	115	0	0
9	95	0	0
10	110	1	115
11	124	0	0
12	106	0	0
13	112	1	124
14	111	0	0
15	107	1	112
16	108	0	0
17	112	0	0

Table 2. Estimated parameters of recovered peaks

Evaluated Parameters of i/p data	Max. Peak(V)	Time Position(μs)
1st Peak	149	2
2nd Peak	115	8
3rd Peak	124	11
4th Peak	112	13

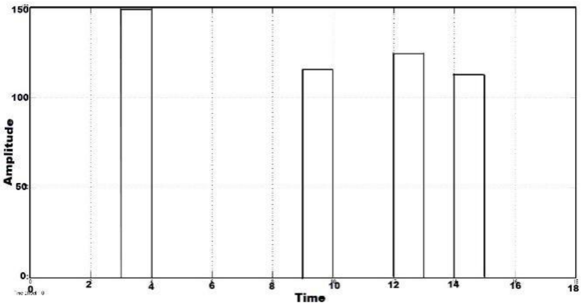

Fig 7: O/p of the scope maximum of the recovered overlapping peaks by using XSG for 1st algorithm

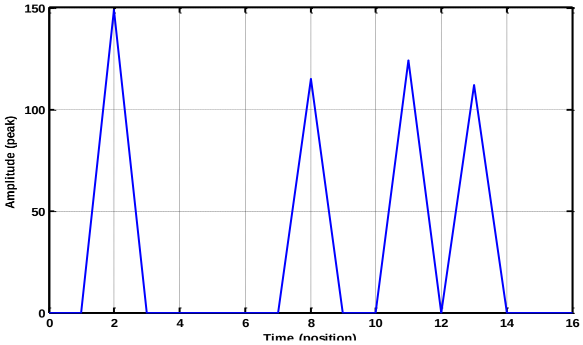

Fig 8: the recovered overlapping peaks by using 1st algorithm

4.2 Experimental results of the second algorithm

The results of the second algorithm are presented. This algorithm is a peak sensitivity method, which is implemented using XSG that determines the location of the maximum peaks. The resultant peaks correspond to peak location. A local extreme is a local maximum that is extremely fast and depends on the definition of the derivative as shown in Fig. 5, and Fig. 6. This algorithm works in several steps. The first step calculates the difference between each two nearby points (M). It can be calculated as in Eq. (1). M1 can be estimated from the first value of M, which is the first value of the previous end value as in Eq. (2). M2 can be estimated from the second value of M to the end value as in eq. (3). Each two equations (2), (3) multiplied and determined as in Eq. (4), as shown in Fig. 9 and Fig. 10.

The difference can be calculated by this equation

$$\boldsymbol{M} = [A_2 - A_1 \quad A_3 - A_2 \quad . \quad . \quad . \quad A_H - A_{H-1}\ ] \qquad \textbf{(1)}$$

$$\boldsymbol{M_1} = [A_2 - A_1 \quad A_3 - A_2 \quad . \quad . \quad . \quad A_{H-1} - A_{H-2}] \qquad \textbf{(2)}$$

$$\boldsymbol{M_2} = [A_3 - A_2 \quad A_4 - A_3 \quad . \quad . \quad . \quad A_H - A_{H-1}\ ] \qquad \textbf{(3)}$$

$$\boldsymbol{Z} = \begin{bmatrix} A_2 - A_1 * A_3 - A_2 & A_3 - A_2 * A_4 - A_3 & . \\ . \quad . & A_{H-1} - A_{H-2} * A_H - A_{H-1} & \end{bmatrix} \qquad \textbf{(4)}$$

From this equation, Z can be obtained. A check is made over each element of Z value. If this element <= 0 then it is replaced with value one (W). The new vector will be called W1 as in Eq. (5).

$$\mathbf{W_1} = \mathbf{M1} * \mathbf{W} \qquad \textbf{(5)}$$

The first two delays in (M) dif. data (0,130) are neglected because of their initial delay nature, hence the first dif1 (M1) is 70. W_1 represents the amplitude of i/p signal as well as estimated $M_{1 > 0}$. However, the peak position corresponds to the position of the estimated peak pulse one. The values ratio for the peak and position of the second algorithm are listed in Table 3. Moreover, the parameters of peaks are calculated and listed in Table 4.

Table 4. Estimated parameters of recovered peaks

Evaluated Parameters of i/p data	Max. Peak(V)	Time Position(μs)
1st Peak	200	2
2nd Peak	180	5
3rd Peak	170	7
4th Peak	230	9

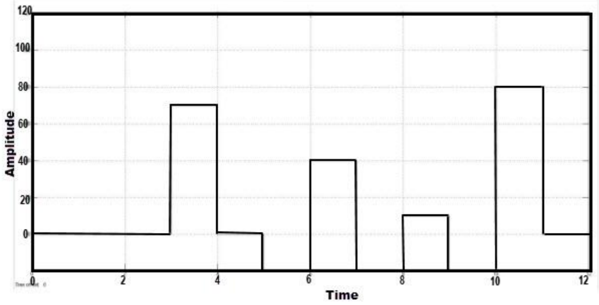

Fig 9: O/p of the scope maximum of the recovered overlapping peaks by using XSG for 2nd algorithm

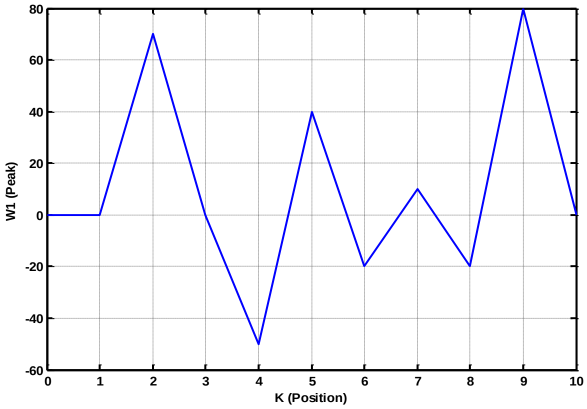

Fig 10: The recovered overlapping peaks by using 2nd algorithm

5. CONCLUSIONS

In this paper, two pile up correction algorithms are developed and implemented using XSG in gamma-ray spectroscopy. These algorithms help in processing higher count rates from higher tracer dosages or from architectures that have higher count rates per channel. They also enable us to remove unwanted pileup events and provide an accurate pulse count rate. Their designs are performed using XSG. System Generator "gateway" blocks convert between Simulink's double-precision floating point signals and System Generator's fixed-point arithmetic type. A powerful method for separating the pile-up events has been presented in scintillator detector. The simulation and experiment results assure high accuracy of recovering pile-up events. The first algorithm ties in very well with small timing and can easily be implemented in an FPGA with a reasonable amount of resources. Therefore, the obtained results confirm that the first algorithm is better than the second one due to its speed and smaller logical components. This paper has characterized several innovative procedures developed and executed to a great success for the detection and correction pile up algorithms of GRS using FPGA. Even though these algorithms has succeeded and achieved good results. More algorithms will be developed for dealing of GRS problems. There are still many aspects which can be advanced in order to increase the efficiency of the pulse height analysis using other sources and applying the same algorithms for neutron, alpha and X-ray spectroscopy. Moreover, that research will continue with other detectors to be suitable for gamma radiation detection.

6. REFERENCES

[1] G. F. Knoll, "Radiation detection and measurement." Third Edition, John Wiley and Sons, NY, 2000.

[2] M. Bolic and V. Drndarevic, "Digital gamma-ray spectroscopy based on FPGA technology," Nuclear Instruments and Methods in Physics Research, Vol. 482, pp.761–766, 2002.

[3] "System Generator for DSP User Guide UG 640 (v11.4)," December 2, 2009

[4] I.I. Mahmoud, M.S. El Tokhy, and H. A. Konber, "Pileup recovery algorithms for digital gamma ray spectroscopy," Journal of Instrumentation, IOP, pp. 1-19, 2012.

[5] I.I. Mahmoud, M.S. El Tokhy, "Development of coincidence summing for digital gamma ray spectroscopy," J. Anal. At. Spectrom, Vol. 29, pp. 1459–1466, 2014.

[6] M. D. Haselman, J. Pasko et al., "FPGA-Based Pulse Pile-Up Correction With Energy and Timing Recovery, " Nuclear Science, IEEE Nuclear and Plasma Sciences Society, Vol. 59, 2011.

[7] M. Bolic, V. Drndarevic, and W. Gueaieb, "Pileup Correction Algorithms for Very-High-Count-Rate -Ray Spectrometry With NaI(Tl) Detectors," Senior Member, IEEE Transactions On Instrumentation And Measurement, Vol. 59, NO. 1, January 2010.

[8] J. Liu et al., "Real time digital implementation of the high-yield pile- up-event-recovery (HYPER) method," in 2007 IEEE Nuclear Science Symposium Conference Record, vol. M26-4, pp. 4230–4232.

[9] M. D. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Simulation of algorithms for pulse timing in FPGAs," in IEEE Nuclear Science Symp. Conf. Record, 2007, pp. 3161–3165.

[10] I.I. Mahmoud, M.S. El Tokhy, "Advanced signal separation and recovery algorithms for digital x-ray spectroscopy," Nuclear Instruments and Methods in Physics Research A773, pp. 104–113, 2015.

[11] "Xilinx, Spartan-3E FPGA Starter Kit Board User Guide," UG230, vol. 1.2, 2011.

[12] T. Saidani , D. Dia, W. Elhamzi, M. Atri and R. Tourki,,"Hardware Co-simulation For Video Processing Using Xilinx System Generator," Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.

[13] M. M. Ouda, M. S. El_Tokhy, M. Amal-Eldin , Sh. Hashima, N.. Ziedan, and I. I. Mahmoud, "Development of Pileup Recovery Algorithms by Peak Detection Method of Digital Gamma Ray Spectroscopy," IEEE National Radio Science Conf. (NRSC). P.487 – 491, 2017.