Online Trigger Processing for a Small-Animal RPC-PET Camera

Filomena M. C. Clemêncio, Custódio F. M. Loureiro, and Jorge Landeck

***Abstract*—A complex task for positron emission tomography (PET) cameras is the design of an appropriate coincidence-detection trigger system as it usually encompasses coincidences in a large number of channels and requires tight time specifications. Those requirements are even greater for a resistive plate chamber (RPC)-based detector technology since the time window specification is quite small (in the order of a few hundred picoseconds) and the number of coincidence-channels can be quite large. Previous work showed that the time resolution for gamma photon pairs of a detector based on resistive plate chamber technology is under 300 ps full width at half maximum. This allows for a very tight time window for coincidence detection, with the corresponding benefits in reducing the number of random coincidences observed and, hence, the overall noise on the acquired image. In this paper, we first show experimental results of a coincidence-detection algorithm implemented inside a Xilinx Virtex-5 field-programmable gate array for a small-animal RPC-PET camera being built.**

***Index Terms*—Coincidence trigger, coincidence validation, resistive plate chamber–positron emission tomography (RPC-PET) camera.**

I. Introduction

PREVIOUS work showed that the time resolution for gamma photon pairs of a detector based on resistive plate chamber (RPC) technology is under 300 ps full width at half maximum (FWHM) [1], [2]. This allows a very small time window for coincidence detection, with the corresponding benefits of reducing the number of random coincidences and, hence, the overall noise in the acquired image and the total amount of data to save and process. Improved RPC technology can allow even better time resolution allowing implementing an effective time-of-flight (TOF) RPC-based camera.

To benefit from these improvements, a good coincidence-detection system must be developed. The use of traditional discrete schemes encompassing discriminators and gates is area and power consuming, especially in the case of a large number of channels, and does not easily provide the desired time resolution [3], [4]. The offline coincidence detection methods put pressure on the data acquisition system, since all events must be time-stamped and saved for later processing [5], [6]. Some systems use free-running fast analog-to-digital converters (ADCs) and extract time stamps directly from the digitized data without the use of time-to-digital converters (TDCs) [7]–[10], with fairly complex data acquisition systems, and coincidence windows in the order of several nanoseconds. A better approach might be the use of TDCs and a completely digital, programmable, real-time coincidence detection algorithm. This would allow acquiring data just for the coincidence events, significantly lowering the cost and complexity of the data acquisition systems.

In this paper, we discuss the first experimental results of a real-time coincidence-detection algorithm implemented inside a Xilinx Virtex-5 field-programmable gate array (FPGA) that will be used in the small-animal RPC-positron emission tomography (PET) camera being developed at our institute.

Manuscript received June 15, 2010; revised March 06, 2011; accepted May 26, 2011. Date of publication July 14, 2011; date of current version August 17, 2011. This work was supported by the Instituto de Investigação Interdisciplinar, Universidade de Coimbra, Portugal.

F. M. C. Clemêncio is with the Escola Superior de Tecnologia da Saúde do Porto—IPP, Vila Nova de Gaia 4400-330, Portugal (e-mail: fcc@estsp.ipp.pt).

C. F. M. Loureiro and J. Landeck are with the Centro de Instrumentação, Department of Physics, University of Coimbra, Coimbra P-3004-516, Portugal (e-mail: custodio@fis.uc.pt).

Digital Object Identifier 10.1109/TNS.2011.2159124

II. General Benefits of the Digital Approach

The inclusion of the necessary logic inside the FPGA greatly eases the problems associated with the distribution and synchronization of time signals through the data acquisition system, a major concern in large systems, allowing for easy implementation of a unique time base for all logic. This can also help build synchronous solutions that are metastability free.

Another major advantage of the FPGA is the actual abundance of resources (when compared to discrete or analog solutions) and the low power consumption. This allows the development of complex algorithms and the use of pipelined architectures that might be able to answer the needs of real-time processing of events, even in the presence of a large number of channels.

A quick inspection of modern FPGA data sheets shows ever-increasing supported clock speeds and sophisticated clock management, a myriad of high-speed interface protocols supported, and very high-speed transceivers. Many of these developments are driven by the need to effectively implement the high-speed serial protocols of these days. This results in the availability of a large number of very high speed one-bit digital samplers, with clock rates over 1 GHz, thus facilitating the implementation of TDCs with hundreds of picosecond resolution, as is often needed in PET applications.

Using FPGA technology might also provide a path for higher speed and higher time resolution without much future re-design as a side effect of the technology development curve. An additional benefit of FPGA is that coincidence sorting algorithms can easily be updated and/or changed.

III. Design Architecture

A. Specifications

The small-animal RPC-PET camera being developed at our institute will have the geometry of a cube, with one RPC plate

forming each face of the cube. The overall characteristics of the camera are: an event rate close to 4 MHz (maximum estimated number of disintegrations/second inside the camera); a relatively high number of data acquisition channels involved (16 channels for each RPC plate, totalizing 96 channels); a data acquisition rate of 100 MHz; a maximum expected coincidence rate near 100 kHz (considering a coincidence window of 1 ns); and a data acquisition time (for image acquisition) that can reach 5 min.

The coincidence detection system should be able to detect coincidences in the six RPC plates and send a trigger signal to the involved data acquisition channels to acquire just the interesting information. In fact, each RPC plate generates a fast hit signal (the time channel) when a photon is detected that is used to test for coincidences.

B. TDCs

Modern high-speed serial protocols are based on the availability of very high-speed serializers-deserializers (serdes), working with clock speeds of 1 GHz and above. Many FPGA providers, such as Xilinx, include these options at the I/O (pad) level. It is then possible to implement high-speed deserialization at the input level and work at an internal lower and manageable clock speed.

The parallel word obtained by deserialization can be seen as the time sampling of the incoming signal. This is the base of a TDC: a digital, time-sampled word showing the status of the event signal in time. An implementation of this principle can be found in [11], where time sampling was made using Gigabit Communications devices integrated in an FPGA. Time sampling also produces the raw data used by the coincidence detection algorithm.

In many cases, the maximum sampling clock speed of the FPGA does not provide the intended temporal resolution. A sampling scheme using more than one input pad per channel and out-of-phase clocks can then be used to achieve higher sampling rates.

C. Edge Processing

The fast hit signal coming from each RPC plate (the time channel) is sampled by the FPGA using a 500 MHz double-data-rate (DDR) clock and three pins, effectively multiplying the sampling rate by 3. The precise timing of occurrence of the event is determined by the low to high transition, or edge, of the stream of ordered bits arriving from the three sampling pins. This stream of data is optionally filtered for noise removal and an edge detection algorithm is applied. The information of the time location of edges is kept to be used by the coincidence detection algorithm. The signals arriving from the six RPC plates are all acquired in parallel using the same 500 MHz DDR clock.

D. Coincidence Detection and Validation

When an edge is detected, its time information is compared to the time information of edges in the other channels. In the actual implementation, the coincidence window can be selected in increments of approximately 0.33 ns up to $\approx$1.3 ns. The coincidences are detected in parallel in all time channels resulting in a 3×3 matrix of coincidences (for our system with six RPC plates organized in two groups, the north and the south groups, each group comprising three time channels).

An algorithm to validate the coincidences globally can then be applied. We implemented an algorithm rejecting multiple coincidences (more than one hit signal in each of the north and south time channels in the specified time window). If a coincidence is validated, a trigger signal is generated and sent to the data acquisition channels of the involved RPCs. Data present in these channels can then be saved and/or processed. These data are used to determine the line of response and, hence, to construct the image.

E. Overall Architecture of the Coincidence System

Fig. 1 shows the overall architecture of the coincidence system. The RPCs time channels are organized into two groups (the north and south channels). Each group contains three time channels.

Every time channel is sampled (sample block) with the high-speed e_bit_clk (a 500 MHz DDR clock) and the resulting parallel words (n_bits and s_bits) are ordered and sent to processing. Further processing of these parallel words is made using the lower-speed e_word_clk (a 250 MHz clock in our implementation).

An edge detection algorithm is then run (Edge Detection block). An edge filtering mechanism that filters out bit transitions that lasts less than a given amount of time (specified by e_filter) is available. We now have the n_e_bits and s_e_bits words containing the edges found.

The coincidence unit (Coincidence Matrix block) looks for edges within a given time window (specified by cw_size) in the n_e_bits and s_e_bits words. In this process, a matrix of coincidences is built (the ec_trigger matrix). This matrix contains one line per each north channel and one column per each south channel. It is then possible to implement a coincidence validation scheme (Coincidence Validation block) to reject unwanted coincidences.

It should be noted that in this architecture, the sampling block is the only one that depends on the FPGA used. All other blocks are completely hardware-independent.

The design is completely synchronous and was written in VHDL in a parameterized fashion. The entities corresponding to the Edge Detection and Coincidence Matrix blocks have two different architectures, named "programmable" and "hard-wired." The first one allows the real-time changing of operating parameters (the edge filter width, e_filter, and the time coincidence window, cw_size), while the second one reaps out unneeded logic (but parameters cannot be changed in real time).

IV. Simulation Results

Fig. 2 shows an example of simulation results obtained with ModelSim [12] for several events. The coincidence window is $\approx$0.66 ns (cw_size is set to 2) and no edge filtering was applied (e_filter is set to 1). Events 1 and 2 are 1 ns apart, and so are outside the detection window, and no coincidence signal is generated (n_coinc_o and s_coinc_o remain low). Events 3 and 4 are 0.5 ns apart, and so are inside the coincidence window, and the corresponding trigger signals are generated less than 30 ns after

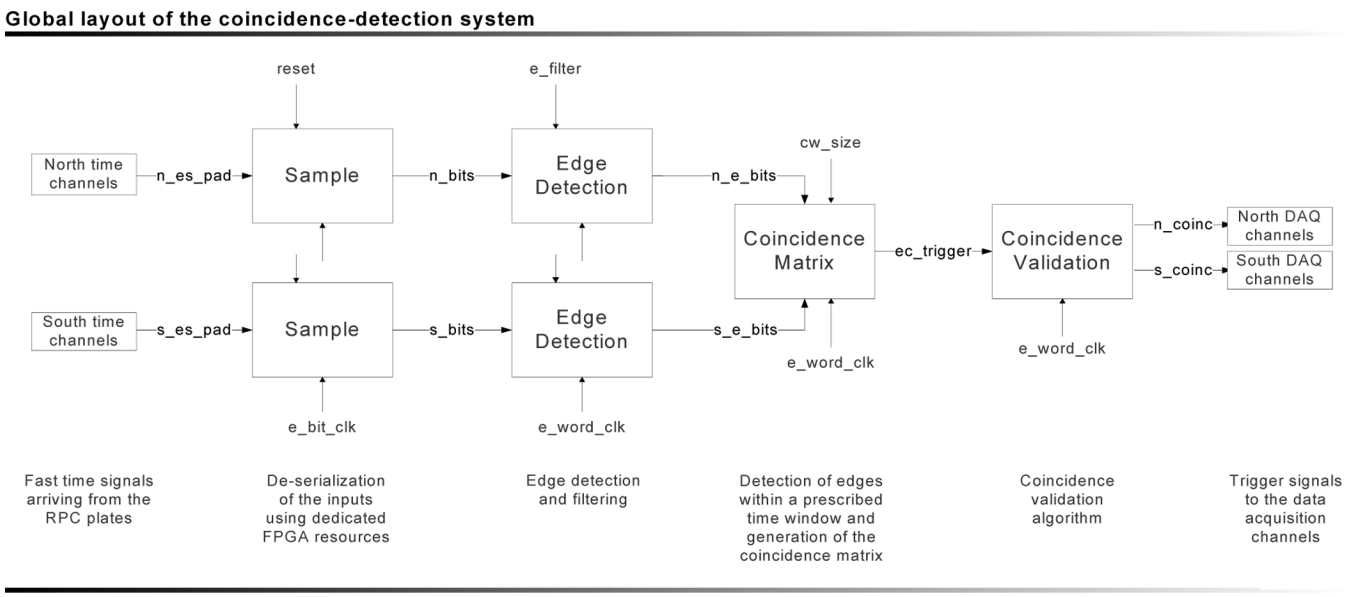

Fig. 1. Overall architecture of the coincidence algorithm.

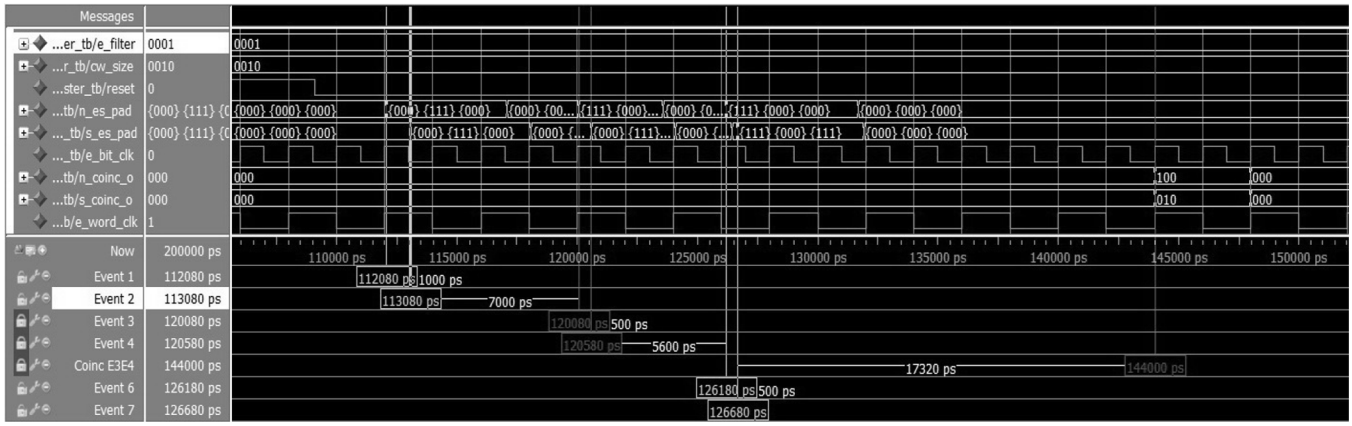

Fig. 2. ModelSim simulation example. Events 1 and 2 are outside the coincidence window. Events 3 and 4 are inside the coincidence window and generate coincidence-trigger signals. The events occurring between Event 6 and Event 7, although inside the coincidence window, are vetoed (multiple coincidences are rejected).

the events occur: coincidence trigger signals are sent to north channel 3 and to south channel 2. A nonmarked event occurs between Events 6 and Event 7: these three events occur inside the coincidence window, but are rejected by the coincidence validation algorithm, as multiple coincidences are rejected.

An interesting point of the developed methods is that they are virtually free of dead time. Edges and coincidences are detected at the rate they are produced without any time-processing penalty: the only request is that the input signal return to zero (low) before a new edge can be detected.

V. Experimental Results

Experimental tests were carried out using a Tektronix AWG240 arbitrary waveform generator, enabling the testing of the system under controlled conditions. The waveform generator was initially programmed to generate a 9.990 MHz rectangular wave ranging from 0 V to approximately 2 V. This signal was fed through two subminiature version A (SMA) connectors to a Virtex-5 XC5VLXT-1 FPGA [13] in a prototype module developed inhouse. This module (Fig. 3) contains six input SMA connectors, organized in rows of three, that are used to receive the north and south time channels from the RPC-PET camera. The other three SMA connectors are used to send the coincidence trigger signals to RPCs in opposing faces of the detector cube.

The module also contains several data acquisition channels built around 12-b 125 MHz ADCs that use LEMO connectors. During development, the FPGA was programmed using Xilinx ISE 12.3 [14] and Platform Cable USB II [15]. Communications with the host PC were made using the USB 2.0 interface.

The 4-ns-long coincidence trigger signals internal to the FPGA were stretched to obtain an approximate 20 ns pulse at the SMA output. The coincidence trigger output to be sent to north 1 channel was then visualized with the help of a Tektronix TDS 3052 digital oscilloscope.

To produce test signals with a known temporal relationship, the output of the waveform generator was split in two. Delay

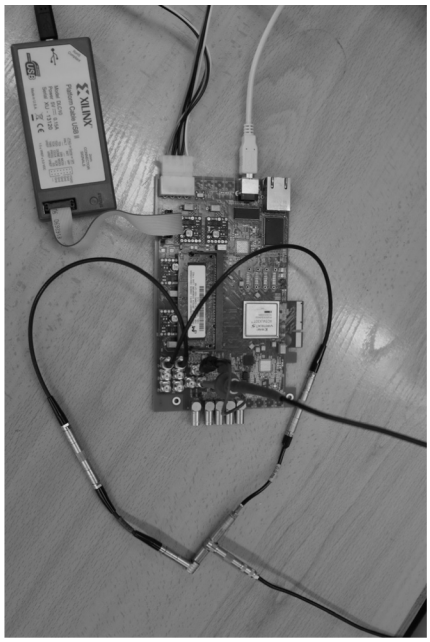

Fig. 3. Module developed for testing, data acquisition, and trigger generation. The left branch of the cable introduces a delay of Δ.

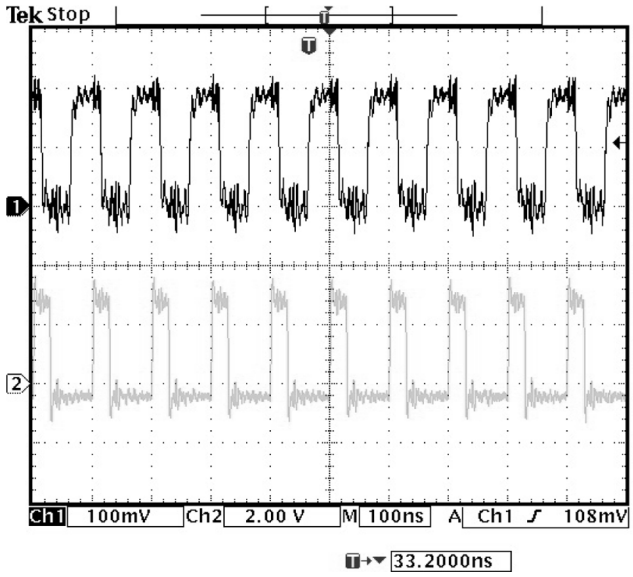

Fig. 4. Coincidence triggers (bottom signal) using a coincidence window $\approx$ 0.33 ns and a cable delay $\approx$0 ns. They are generated within 35 ns after the input edges are detected. Ch1 (top signal) is using a divide-by-10 probe.

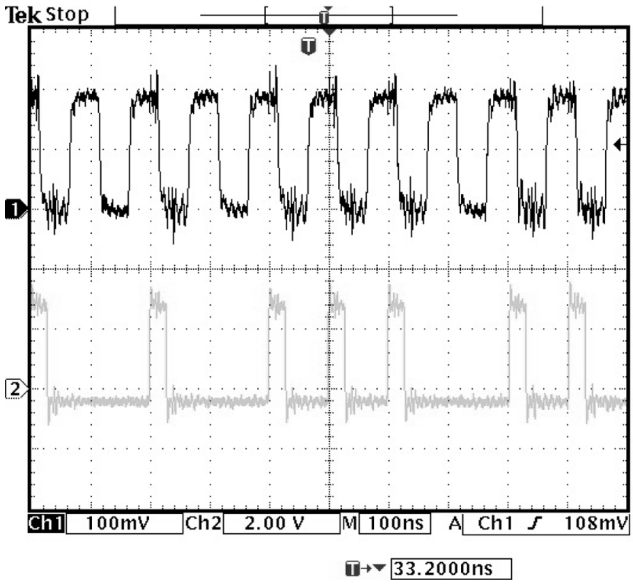

Fig. 5. Coincidence window $\approx$0.66 ns, delay 0.5 ns, same channel layout as in Fig. 4. Not all coincidences are detected.

cables were used to eventually delay one of the branches. These are the signals that are used to test for coincidences. We used a set of cables with delays of 0.5, 1, and 2 ns to introduce delays up to 1.5 ns in one of the branches. Intermediate delays were created adding elbow (90°) LEMO plugs. In the following text, we refer to these delays by Δ. Fig. 3 shows a delay of Δ introduced in one of the branches.

Fig. 4 shows the input signal (top signal) and the output coincidence trigger (bottom signal) generated when no delay is introduced between the two cables (delay is $\approx$0 ns), measured at the SMA connectors. No edge filtering is used. The coincidence window is $\approx$0.33 ns. Fig. 4 shows the coincidences being detected at the rate of the input waveform (9.990 MHz).

Using the same parameters but introducing a delay of 0.5 ns in one of the cables, no coincidence triggers are generated, and the bottom display of the oscilloscope becomes a line (not shown).

The situation changes in Fig. 5, where we programmed a double-width coincidence window size (we set cw_size to 2) and kept using the delay of 0.5 ns. Now, the coincidence window is $\approx$0.66 ns, and ideally all coincidences should be detected. In fact, coincidences are detected, but not all of them. This is due to noise present in the system and to small differences due to routing inside the FPGA.

To better characterize the system, we programmed the FPGA to count every edge detected in the north and south channels, and the number of coincidences detected. All counters start counting simultaneously, and all counters are stopped when the first of them reaches the value 32767. A first result was that all edges are detected, as both north and south counters always reach the value of 32767. The number of coincidences detected varies with the programming of the coincidence window size and the delay introduced in one of the cables. Results in percentage are shown in Table I for the 9.990 MHz rectangular wave. Measures with other frequencies showed similar results. No edge filtering was applied (e_filter is set to 1). The coincidence window size, cw_size, with a granularity of $\approx$0.33 ns, was varied as shown. Each value in the table was calculated from the mean of a set of 10 consecutive measures. The average of the absolute deviations is also shown in percentage, unless its value is zero.

The results in Table I show that 100% of the coincidences are detected when the coincidence window is sufficiently large. They also show that the actual implementation suffers from noise, although subnanosecond coincidence windows can be used.

VI. Conclusion

We implemented, in an FPGA, online trigger processing for a small-animal RPC-PET camera, with subnanosecond temporal

TABLE I
PERCENTAGE OF COINCIDENCES DETECTED FOR THE 9.990 MHz RECTANGULAR WAVE AND DIFFERENT COMBINATIONS OF COINCIDENCE WINDOW SIZES AND CABLE DELAYS

delay (ns)	0	0+Δ	0+2Δ	0+3Δ	0.5	0.5+Δ	0.5+2Δ	0.5+3Δ	1	1+Δ	1+2Δ	1+3Δ	1.5
cw_size													
1	96.7±0.1	69.3±0.3	35.0±0.1	8.8±0.1	0	0	0	0	0	0	0	0	0
2	100	100	100	99.7±0.1	68.6±0.2	42.5±0.2	11.0±0.5	0	0	0	0	0	0
3	100	100	100	100	100	100	99.2±0.1	82.4±0.2	31.2±0.1	8.5±0.1	0	0	0
4	100	100	100	100	100	100	100	100	100	97.1±0.1	73.6±0.2	38.3±0.2	0

resolution and the ability to implement dedicated coincidence-validation algorithms.

The method used is quite versatile and fast. It copes very well with high event rates since it is based on the detection of the edges of the fast detector signals (time channels), and should be useful whenever time information can be derived from the edges of signals. The simulations show that event rates over 100 MHz can be processed in real time.

We show experimental results that validate the simulation results. They also show that the actual implementation suffers from noise, although subnanosecond coincidence windows can be used. The coincidence-trigger signals are generated within 35 ns after the fast detector signals are generated, enabling easy acquisition of only the interesting data. This is a huge advantage of the presented method.

The concept is also very interesting for larger cameras with a greater number of time channels to test for coincidences. We are considering the extension of the method to a larger RPC-PET camera that is in the design phase [16].

We are also working on achieving better time resolution in future implementations.

REFERENCES

[1] P. Fonte *et al.*, "A new high-resolution TOF technology," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 443, pp. 201–204, 2000.

[2] A. Blanco *et al.*, "RPC-PET: A new very high resolution PET technology," *IEEE Trans. Nuclear Sci.*, vol. 53, no. 5, pt. 1, pp. 2489–2494, Oct. 2006.

[3] B. Collinge *et al.*, "A fast multiple coincidence circuit," *J. Scientif, Instrum.*, vol. 33, pp. 72–72, 1956.

[4] H. M. Dent, W. F. Jones, and M. E. Casey, "A real time digital coincidence processor for positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. NS-33, no. 1, pp. 556–559, Feb. 1986.

[5] T. W. Leadbeater and D. J. Parker, "A high speed PC-based data acquisition and control system for positron imaging," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 604, pp. 355–358, 2009.

[6] D. P. McElroy *et al.*, "A true singles list-mode data acquisition system for a small animal PET scanner with independent crystal readout," *Phys. Med. Biol.*, vol. 50, pp. 3323–3323, 2005.

[7] M. Streun *et al.*, "A PET system with free running ADCs," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 486, pp. 18–21, 2002.

[8] E. Auffray *et al.*, "The clearPET project," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 527, pp. 171–174, 2004.

[9] J. D. Martínez *et al.*, "Design of a coincidence processing board for a dual-head PET scanner for breast imaging," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 546, pp. 28–32, 2005.

[10] A. Mann, B. Grube, I. Konorov, S. Paul, L. Schmitt, D. P. McElroy, and S. I. Ziegler, "A sampling ADC data acquisition system for positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 1, pt. 2, pp. 297–303, 2006.

[11] J. Sousa *et al.*, "A PCI time digitizer for the new JET time-of-flight neutron spectrometer," *Fusion Eng. Design*, vol. 71, pp. 101–106, 2004.

[12] Mentor Graphics Corporation, ModelSim SE Plus 6.4a, 2008.

[13] Xilinx, Inc., Virtex-5 FPGA User Guide v5.3, 2010. [Online]. Available: http://www.xilinx.com/support/documentation/virtex-5.htm

[14] Xilinx, Inc., ISE Design Suite 12.3. 2010. [Online]. Available: http://www.xilinx.com/support/documentation/dt_ise.htm

[15] Xilinx, Inc., Platform Cable USB II. 2011. [Online]. Available: http://www.xilinx.com/support/documentation/data_sheets/ds593.pdf

[16] A. Blanco *et al.*, "Efficiency of RPC detectors for whole-body human TOF-PET," *Nucl. Instrum. Meth. Phys. Res. A: Accelerators, Spectrometers, Detectors Associated Equip.*, vol. 602, pp. 780–783, 2009.