An All-Digital Coincidence-Selection and Coincidence-Trigger Generation for a Small Animal RPC-PET Camera

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

***Abstract*–The time resolution for gamma photon pairs of a detector based on resistive plate chamber technology is close to 300 ps FWHM. This allows a tight time window for coincidence detection, reducing the number of random coincidences observed and the overall noise of the acquired image. We implemented, inside a Xilinx Virtex-5 field programmable gate array, a fully programmable coincidence-trigger system with sub-nanosecond resolution and low latency for a small animal RPC-PET camera being built.**

An automatic procedure to construct a histogram of coincidences was implemented that helps characterizing the camera and allows compensating for different delays in the time channels. The time window for coincidences can then be placed at the optimal position helping rejecting coincidences from undesired sources.

I. Introduction

RESISTIVE plate chamber (RPC)-based detector technology provides a very interesting path towards high-resolution, low-cost positron emission tomography (PET) cameras with high sensivity [1, 2]. Such cameras make use of the high temporal resolution of RPC detectors [3] to achieve a time resolution close to 300 ps full width at half maximum (FWHM) for the detection of gamma photon pairs. This allows a tight time window for coincidence detection, reducing the number of random coincidences observed and the overall noise of the acquired image.

In this paper we show experimental results of a coincidence-trigger algorithm implemented inside a Xilinx Virtex-5 field programmable gate array (FPGA) for a small animal RPC-PET camera being built. This camera has the shape of a cube, with an RPC detector in each face. Each RPC detector generates a fast hit signal when a photon is detected that is used to test for coincidences. This allows, conceptually, just using a high-speed comparator to generate a digital signal from it that can be time-sampled and used to check for coincidences in an all-digital fashion.

Manuscript received October 4, 2011. This work was supported in part 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, Rua Valente Perfeito, 322, 4400-330 Vila Nova de Gaia, Portugal (e-mail: fcc@ estsp.ipp.pt).

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

J. Landeck is with the Centro de Instrumentação, Department of Physics, University of Coimbra, P-3004-516 Coimbra, Portugal (e-mail: jlandeck@fis.uc.pt).

A first implementation of such a coincidence system, along with several experimental results helping characterizing it, obtained through the use of a Tektronix AWG240 arbitrary waveform generator, can be found in [4]. However, while trying to use this coincidence system with real signals arriving from an experimental two-heads RPC-PET camera we were faced with several difficulties. The more challenging was a time offset depending on the RPC detector and associated electronics behavior that can be described roughly as the introduction of different time delays between the time channels pairs tested for coincidences. In other words, the events resulting from the positron annihilation were arriving at quite different time instants when measured by a unique system clock. It became clear that our original coincidence system should be enhanced to deal with this problem in a practical way.

We also wanted to enhance the algorithm of coincidence validation to implement more directly the field of view (FOV) concept.

II. Coincidence System Architecture

In the following the requirements of the coincidence system are outlined and the selected architecture is shown.

A. The Small-Animal RPC-PET Camera

The small-animal RPC-PET camera being developed has the approximate geometry of a cube. The RPC detector in each face of the cube generates a fast hit signal that is used to test for coincidences. Although the annihilation source can be roughly considered at the geometrical center of the cube, in practice, the hit signals arriving from one positron annihilation can be away in time by several ns due to detector peculiarities and differing electronics and cable delays, as was verified experimentally. The coincidence system should be able to compensate for such delays.

The building of the RPC-PET camera is a permanently evolving project. The actual 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 (48 channels for each RPC head, totalizing 268 channels for a full six heads camera); a data acquisition rate near 10 MHz; a maximum expected coincidence rate of 100 kHz (considering a coincidence window of 1 ns); a data acquisition time (for image acquisition) that can reach 5 minutes.

978-1-4673-0120-6/11/$26.00 ©2011 IEEE

B. Time Channels and Coincidence Requirements

Although the usual PET camera has the form of a ring the approach to design the coincidence algorithm for a camera with the shape of a cube is quite similar. Fig. 1 shows a drawing of a ring PET composed by six detectors to mimic the number of channels in the cube PET. The original algorithm implemented a coincidence validation stage. We shall enhance it to implement a field of view (FOV) validated coincidence trigger. We use a new parameter, *nFOV*, to distinguish between the different possibilities seen in Fig. 1. Using the first channel as an example we have: if *nFOV* = 0 then channel 1 is tested for coincidences with channel 4; if *nFOV* = 1 channel 1 is tested for coincidences with channels 3, 4 and 5; finally if *nFOV* = 2 channel 1 is tested for coincidences with channels 2, 3, 4, 5 and 6. The same scheme is implemented for channels 2 and 3, resulting in 15 pairs of channels to be tested for coincidences. This scheme can be easily generalized to a system comprising N time channels (N even).

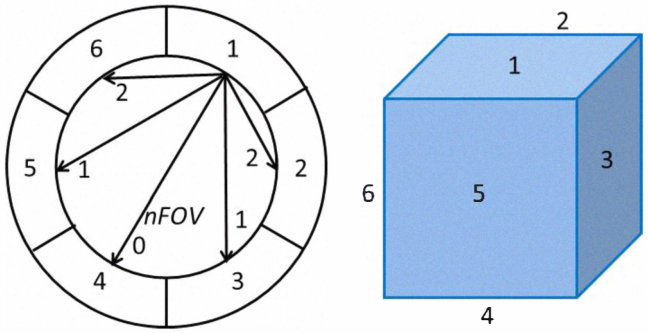

Fig. 1. Ring and cube PET-cameras with time channels identified by numbers. Using channel 1 as an example and the parameter *nFOV* = 2 the pairs of channels shown are tested for coincidences.

In the case of the cube the test for coincidences follow exactly the same pattern, with the channels numbers as identified in Fig. 1.

Coincidences outside the selected FOV and multiple coincidences are rejected by the coincidence validation stage (Fig. 3).

C. Time Synchronization

Time synchronization is a major problem of large experimental setups as usually it's not easy to build a clock distribution system with the desired accuracy. This is especially true for large PET cameras, where a large number of detectors and data acquisition channels are involved, and the acceptable error margins are small, maybe of the order of a few tens of picoseconds or even lower.

Several ingenious solutions have been proposed to deal with this problem. The usual delay-line solution involves calibration of the system using physical delays, eventually using cables with different lengths to compensate for the delays, is very cumbersome and cannot adapt easily to changing system conditions. More modern solutions have been proposed that make use of logic programmable devices, like the field programmable gate array (FPGA) and its high-speed serial links [5].

In our case the clock synchronization problem is not an issue due to the small number of time channels in our RPC-PET camera, but the availability of a global system clock is mandatory.

D. The Role of the FPGA

Using FPGA technology allows developing complex, fast logic algorithms that can be upgraded or changed easily. This by itself is an enormous advantage, as the present work shows through the improvements introduced in the existing algorithms.

Another practical and important advantage, from the point of view of designing trigger systems, is that the FPGA naturally provides a unique system clock for all logic. This clock is used to sample the time signals arriving form the PET camera and is used internally implementing a fully synchronous design.

E. Digitized Time Channels

All time signals arriving from the RPC-PET camera are digitized at the pads of the FPGA using a 500 MHz double-data-rate (DDR) clock. Three pads are used for each time signal, effectively providing an equivalent sampling rate of 3 GHz. To lower the internal processing clock input de-serializers are used (the ISERDES components of the Virtex-5 FPGA, [6]), so that internally all information is processed at the more manageable speed of 250 MHz. The digitized data are inspected to find transitions from low to high (edges). These transitions mark the detection of one event (a hit) in the time channel.

Previously an edge-filtering mechanism was implemented [4] but experience showed that this filtering was not needed as the digital time signals had clean low to high transitions.

F. Coincidence Window

Coincidence detection is based on using the 3 GHz equivalent sampling clock and finding edges within a certain time window. The smallest available coincidence window spans two clock periods, or approximately 660 ps, as can be seen in Fig. 2, where several events are shown that are less than one clock period T apart (T = 330 ps) but that might be sampled by consecutive clock edges (events in the time interval $T < \Delta t < 2T$ are always sampled by two consecutive clock transitions).

Coincidences windows of sizes 2T, 3T and 4T can be selected. Nevertheless, in the case of the small-animal RPC-PET camera, the preferred coincidence window is the smallest one (660 ps).

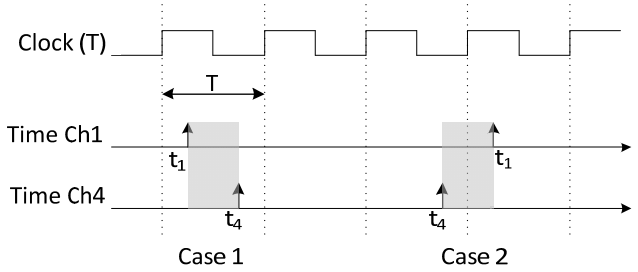

Fig. 2. Digitized time channels sampled with a clock of period T. For hits occurring less than one clock period apart two different cases are possible, showing the minimum available coincidence window spans two clock periods to guarantee detection of all coincidences. Note that the detection order of the events (first t_1 then t_4, or first t_4 then t_1) is irrelevant.

G. Delay Compensation

Due to the peculiarities of the detectors and associated electronics different time channels can have different systematic propagation delays. From the point of view of detecting coincidences these different delays introduce a systematic time interval Δt_d that is added to the real time interval Δt due to the detection of the gamma photon pair. The time interval of true coincidences becomes $\Delta t_p = \Delta t_d + \Delta t$. This would require using a much larger time coincidence window of size Δt_p to detect coincidences, with the disadvantage of augmenting the number of random coincidences and resulting eventually on a noisier image.

To keep the coincidence window small a method to compensate the Δt_d delay is needed. First Δt_d must be measured. This can be accomplished using an extension of the delay method usually used to estimate the random coincidence rate. Instead of simply measuring the random coincidences using a delayed coincidence window, we build a histogram of coincidences by sweeping the delay for a given time range in increments of the sampling clock period T. Ideally the histogram would show a more or less constant random coincidence floor and a peak where the true coincidences are found. This way we have a good estimate of the random coincidence rate and at the same time we can keep using a small coincidence window. In fact, after finding the position of the true coincidences the digital delay line can be programmed with the corresponding delay, in practice making the true coincidences coincident in time.

The delays were easily introduced using programmable shift registers inside the FPGA. The actual implementation allows sweeping a 19.47 ns range in increments of 330 ps.

H. Coincidence System Architecture

The conceptual coincidence system architecture can be seen in Fig. 3. Each time channel is sampled using the equivalent 3 GHz sampling rate. The stream of bits produced from sampling can be delayed according to the *delay* parameter. The delay unit is the sampling clock period, or 330 ps. After delaying the signals edges are found. These edges are the time-mark of the detection of an event. If an event in another channel happens within a prescribed time window (parameter *cw_size*) the coincidence is registered. This is made in parallel for all pairs of channels in the FOV. In the case of a six-channel system and *nFOv* = 2 this results in the construction of a matrix of coincidences of size 3x5. The matrix of coincidences is then checked for allowed coincidences according to the parameter *nFOV*. This way the concept of field of view is implemented at the trigger system level. It is also possible to reject multiple coincidences by selecting the *m_reject* parameter.

The design was written in VHDL in a device-independent fashion, although the Sample block uses specific components of the FPGA used (Xilinx Virtex-5). No time constraints were used.

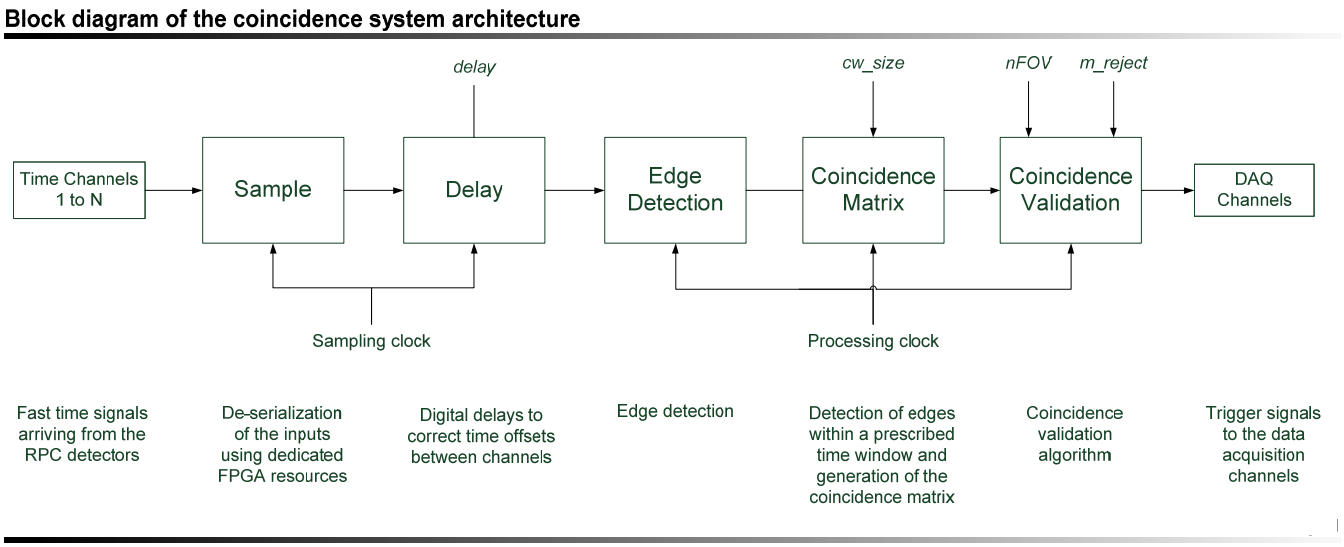

Fig. 3. Conceptual organization of the coincidence trigger system. Time channels are digitized at the equivalent 3 GHz sampling rate. A digital delay can then be applied to compensate for time offsets between channels. Processing is done at the much lower speed of 250 MHz.

III. Experimental Results

A. Using a Function Generator

First experimental tests were carried out using a function generator (Wavetek 50 MHz Pulse/Function generator model 81) and physical delay lines. The output of the function generator was split by using a T-shape connector and the two branches were delayed using different cable lengths. These delayed signals were used to test the algorithm. The signals were connected to the time channels FPGA inputs and a set of histograms of coincidences was built.

The results shown in Fig. 4 are for an input rectangular wave of frequency 19.91 MHz and a data acquisition time of approximately 0.1 seconds. Five overlaid histograms are shown. The first one has a small delay due to the use of cables of similar, but no equal, lengths, and is labeled as the 0 ns series. The other histograms were obtained adding cables with the approximate delays shown to one of the branches. A total sweep time of 19.47 ns was used, in increments of 0.33 ns (although, for clearness, only the first 5.94 ns of the sweeps are show, as all other readings are zero). The added cable delays are clearly visible in the histogram as the histogram peaks shift accordingly.

Measuring the FWHM of the peaks in Fig. 4 show a coincidence time window width of approximately 0.99 ns instead of the nominal 0.66 ns expected. This can happen if the events considered in coincidence have a real time offset smaller than the coincidence window but bigger than the sampling clock period (actually 330 ps), or if a similar spreading effect happens, eventually due to noise.

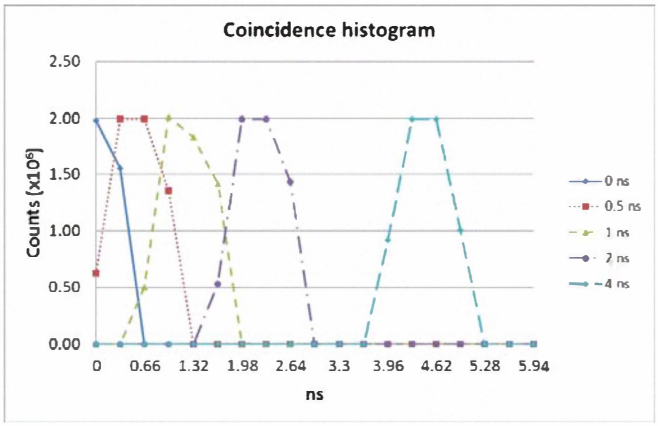

Fig. 4. Coincidence histograms obtained by splitting the output of a function generator, a 19.91 MHz rectangular wave, and using it as inputs to the two channels tested for coincidence. The data acquisition time for each bin was 0.1 s. Due to a fixed time offset bigger than 0.33 ns in the events considered in coincidence or to a similar spreading effect due to noise the 0.66 ns coincidence window selected looks like a larger 0.99 ns FWHM coincidence window.

Introducing a delay in the time channels incurs a penalty in the latency of the trigger system. Fig. 5 shows the coincidence histogram of a system with a large time offset (a much longer cable was used in one of the branches of the signal arriving from the function generator). The histogram shows a peak around the delay of 11.22 ns.

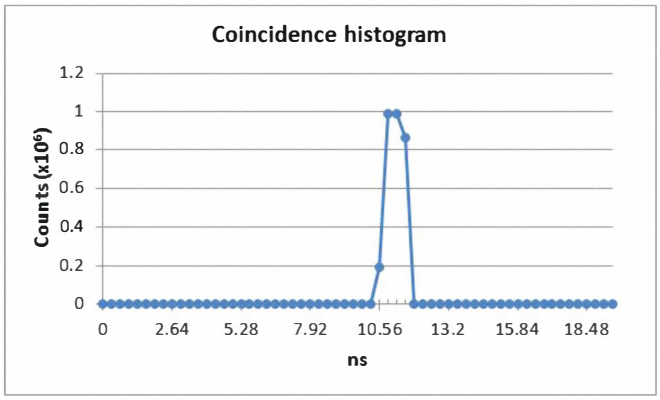

Fig. 5. Coincidence histogram measured while using a large time offset between the channels in coincidence. It shows the time offset between the two channels is around 11.22 ns.

Selecting the time offset of 11.22 ns we now measure the latency of the trigger system with the help on an oscilloscope. In Fig. 6 we can see the input time signal (top 9.91 MHz rectangular wave) and the output trigger signal that is generated. As it can be seen, the latency is now of approximately 50 ns, to be compared to the original latency of approximately 35 ns.

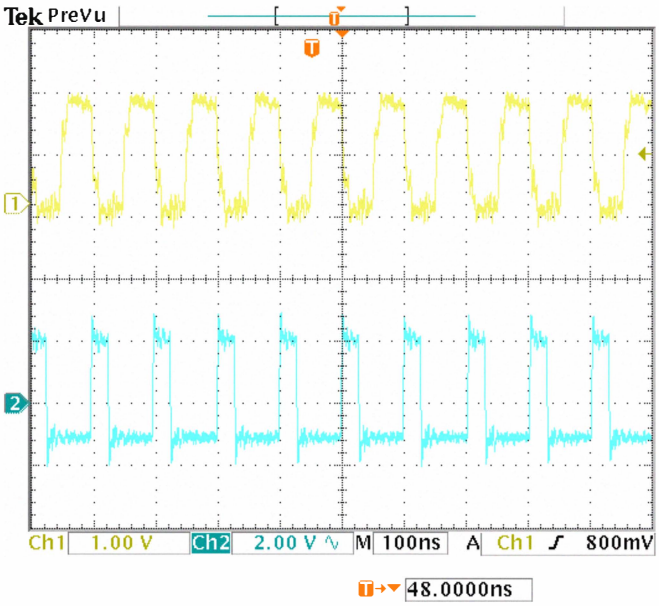

Fig. 6. Latency of the coincidence trigger system when one of the channels has a time offset of 11.22 ns. The top signal is the undelayed input signal, while the bottom signal shows the coincidence trigger output generated. The measured latency is under 50 ns.

B. Using signals From a RPC-PET camera

Experiments were carried out using a prototype two-heads RPC-PET camera. Fig. 7 shows the experimental setup, with the camera inside the metal box.

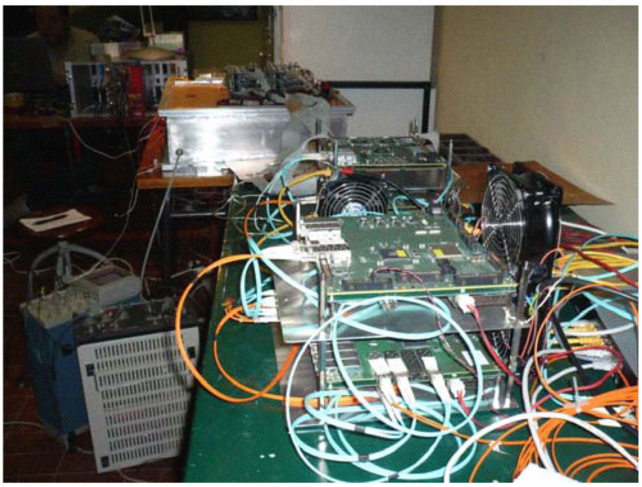

Fig. 7. Experimental setup for the prototype two heads RPC-PET camera.

A positron source was placed in the middle of the camera and a coincidence histogram was built. Although one would expect to have just one peak in the coincidence histogram the measurements showed differently. Fig. 8 shows the histogram obtained while acquiring 300 s for each bin. There are clearly three distinct peaks, showing some kind of noise source is present in the system.

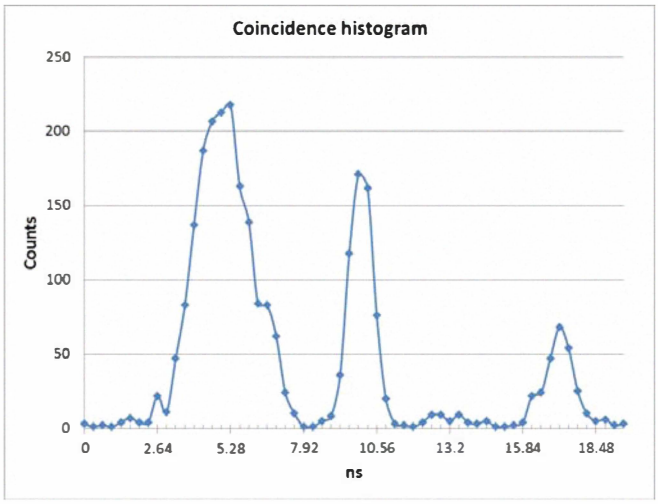

Fig. 8. Coincidence histogram obtained experimentally using the prototype two-heads RPC-PET camera. Each bin was acquired during 300 s. The histogram has several peaks, showing some kind of noise source is present in the system.

IV. Conclusions

We implemented, inside a Xilinx Virtex-5 field programmable gate array, a fully programmable coincidence-trigger system with sub-nanosecond resolution and low latency for a small animal RPC-PET camera being built. First experimental tests showed the trigger system works as expected.

Using a completely digital approach it was possible to develop an automatic procedure to construct a histogram of coincidences that helps studying and characterizing the camera.

A method was implemented to place the coincidence window at the correct position in time, easily compensating the time offsets due to differing electronics and cable delays, while allowing to use a small coincidence window.

The coincidence trigger system showed a very small latency (under 50 ns while delaying a time signal for approximately 11 ns) and is virtually dead-time free.

References

[1] A. Blanco, N. Carolino, C. M. B. A. Correia, L. A. F. L. Fazendeiro, N. C. A. F. N. C. Ferreira, M. F. F. A. M. M. F. F. Marques, R. F. A. M. R. F. Marques, P. A. F. P. Fonte, C. A. G. C. Gil, and M. P. A. M. M. P. Macedo, "RPC-PET: A New Very High Resolution PET Technology," *Nuclear Science, IEEE Transactions on,* vol. 53, pp. 2489-2494, 2006.

[2] A. Blanco, M. Couceiro, P. Crespo, N. C. Ferreira, R. Ferreira Marques, P. Fonte, L. Lopes, and J. A. Neves, "Efficiency of RPC detectors for whole-body human TOF-PET," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* vol. 602, pp. 780-783, 2009.

[3] P. Fonte, A. Smirnitski, and M. C. S. Williams, "A new high-resolution TOF technology," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment,* vol. 443, pp. 201-204, 2000.

[4] F. M. C. Clemencio, C. F. M. Loureiro, and J. Landeck, "Online Trigger Processing for a Small-Animal RPC-PET Camera," *Ieee Transactions on Nuclear Science,* vol. 58, pp. 1766-1770, Aug 2011.

[5] R. J. Aliaga, J. M. Monzo, M. Spaggiari, N. Ferrando, R. Gadea, and R. J. Colom, "PET System Synchronization and Timing Resolution Using High-Speed Data Links," *Ieee Transactions on Nuclear Science,* vol. 58, pp. 1596-1605, Aug 2011.

[6] "Xilinx Virtex-5 documentation," *http://www.xilinx.com/support/documentation/virtex-5.htm.*