The Design of a Digital Coincidence-Detection Trigger System for a RPC-Based Pet

Filomena M. C. Clemêncio, Custódio F. M. Loureiro, Carlos M. B. A. Correia

***Abstract*–A complex task for pet cameras is the design of an appropriate coincidence-detection trigger as it usually encompasses coincidences in a large number of channels and tight time specifications. Those requirements are even greater for a resistive plate chamber (RPC)-based detector technology as 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 (more than 100 in a large camera).**

In this work we discuss on-going work aimed at implementing the coincidence-detection algorithm inside a FPGA along with the necessary TDCs. The use of the developed solution in a first prototype of an RPC-based small animal pet and its scalability to the case of a human, all-body, pet, is discussed.

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 FWHM [1] [2]. This allows a very small 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 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 does not easily allow the implementation of a TOF pet and is area and power consuming, specially in the case of a large number of channels. A better approach might be the use of time-to-digital converters (TDCs) and a completely digital time-coincidence detection algorithm.

In this paper we discuss on-going work aimed at implementing the coincidence-detection algorithm inside a Xilinx Virtex-4 field programmable gate array (FPGA).

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 easily implementing a unique time base for all logic. This can also help building synchronous solutions that are meta-stability 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.

Using FPGA technology might also provide a path for higher-speed and higher-resolution TDCs without much future re-design as a side effect of the technology development curve.

III. Design architecture

A. Introduction

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 of implementing effectively the high-speed serial protocols of these days. But this means that we have easily available a large number of very high speed one-bit digital samplers, with clock rates over 1 GHz. This is the key to implementing digital TDCs with hundreds of pico-second resolution, as it is needed for our pet camera.

B. TDCs

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

Obviously the parallel word obtained by deserialization can be seen as the time sampling of the incoming signal. This is

Manuscript received November 23, 2007. 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, Rua João de Oliveira Ramos, 87, 4000-294 Porto (e-mail: fcc@estsp.ipp.pt).

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

C. M. B. A. Correia is with the CEI, Department of Physics, University of Coimbra, P-3004-516 Coimbra, Portugal (e-mail: correia@fis.uc.pt).

1-4244-0923-3/07/$25.00 ©2007 IEEE.

the base of our TDC: a digital time-sampled word showing the status of the event signal in time. If a higher sampling rate than the directly provided by the FPGA is needed, and if resources are available, a sampling scheme with out-of-phase clocks, maybe using more than one input pad per channel, can be used.

C. Edge Processing

Each event in the detector generates a fast electrical signal that marks its occurrence. This is the signal that is sampled and converted to a parallel word. The precise timing of occurrence is defined by its low to high transition, or edge. After sampling and deserialization an algorithm is run in each word looking for transitions. The transitions in the word are then saved along with their time of occurrence for further processing. The information of time of occurrence includes a coarse time (we implemented a 40-bit, 250 MHz, counter) and a fine time, derived from the sampled word, with a global resolution of approximately 300 ps.

D. Coincidence Detection

When an edge is detected the time information is compared to the time information of edges in the other channels. In case there is an edge in the specified time window a coincidence trigger is generated and sent to the corresponding channels. Data present in these channels can then be saved. This data, eventually in conjunction with the time information, is then used to determine the spatial origin inside the camera of the detected photon pair (TOF pet), and hence to construct the image.

E. Overall architecture of a coincidence channel

Fig. 1 shows the overall architecture of a coincidence channel composed by two input channels (the north and south channels). Each channel is time-sampled (edge_sample) with the high-speed clock and the resulting words are sent to processing. The processing unit (edge_process) looks for edges and adds a coarse time (generated by edge_time) stamp to each one detected, and also increments a counter counting the edges. There is also provision for an edge filtering mechanism that filters out transitions with less than a specified amount of time. The coincidence unit (edge_coinc) finally looks for edges found within a specified time window.

In this architecture the sampling procedure is the only one that depends on the underlining FPGA used. All others are completely hardware-independent. The hardware is completely described using VHDL.

The entities corresponding to the processing and coincidence detection blocks have two different architectures, named programmable and hard-wired. The first one allows the real-time changing of operating parameters (the edge filter with and the time coincidence window), while the second one reaps out unneeded logic (but parameters cannot be changed in real-time).

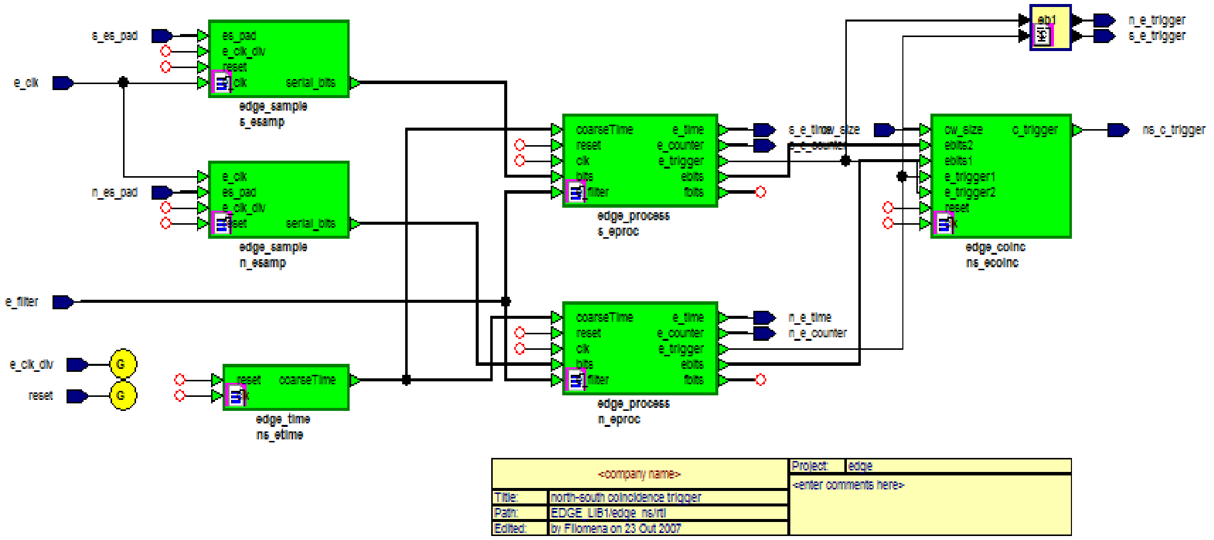
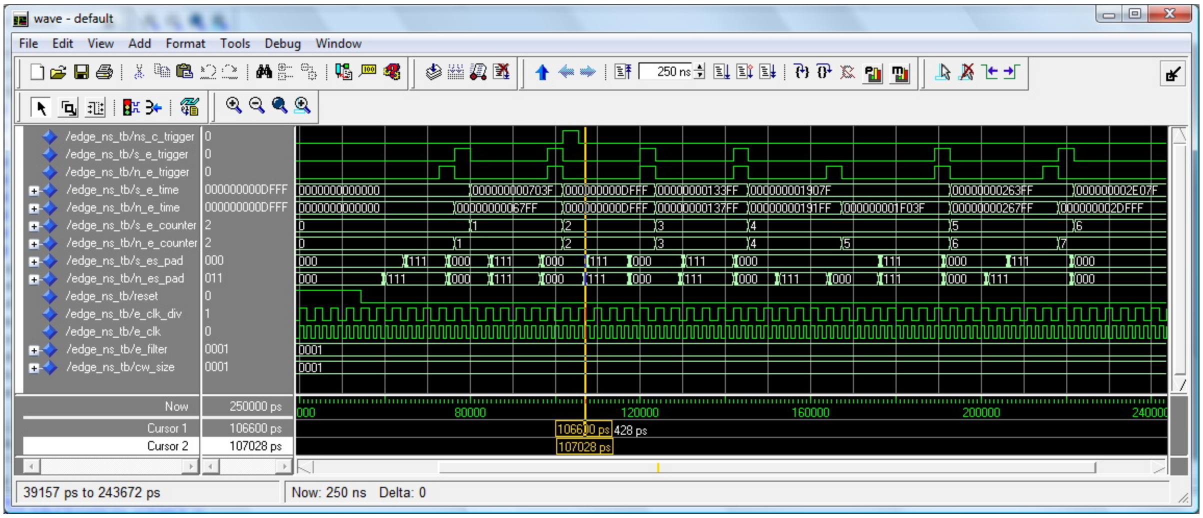


Fig. 1. The overall architecture of the north-south coincidence channel.

IV. Simulation Results

Figures below show the simulation results obtained with ModelSim for the north-south coincidence channel of the small-animal pet being developed. These figures are for the same input and different time coincidence windows. Please note that as the time coincidence window is increased the number of coincidences detected (ns_c_trigger) also increases. The figures show (using time cursors) the time interval between two events (428 ps) that obviously are not detected as a coincidence when using the time window of 300 ps, although, as expected, with a time window of 600 ps, a coincidence is detected.

A small description of the information flow during the coincidence trigger generation appears below.

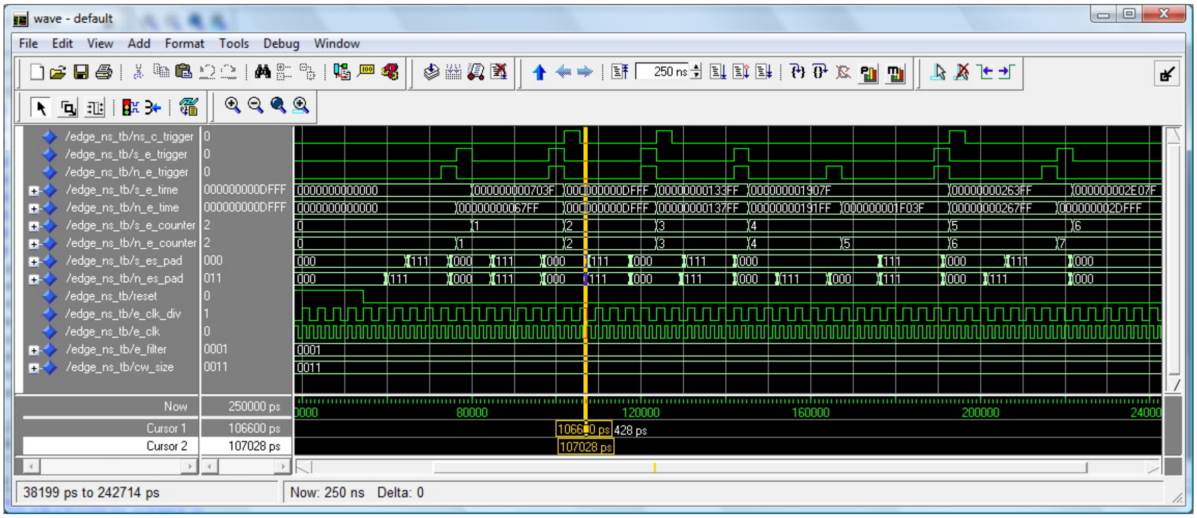


```
/edge_ns_tb/ns_c_trigger   0
/edge_ns_tb/s_e_trigger    0
/edge_ns_tb/n_e_trigger    0
/edge_ns_tb/s_e_time       000000000DFFF
/edge_ns_tb/n_e_time       000000000DFFF
/edge_ns_tb/s_e_counter    2
/edge_ns_tb/n_e_counter    2
/edge_ns_tb/s_es_pad       000
/edge_ns_tb/n_es_pad       011
/edge_ns_tb/reset          0
/edge_ns_tb/e_clk_div      1
/edge_ns_tb/e_clk          0
/edge_ns_tb/e_filter       0001
/edge_ns_tb/cw_size        0001

Now       250000 ps
Cursor 1  106600 ps
Cursor 2  107028 ps

39157 ps to 243672 ps    Now: 250 ns   Delta: 0
```

Fig. 2 Time coincidence window ≈ 300 ps.

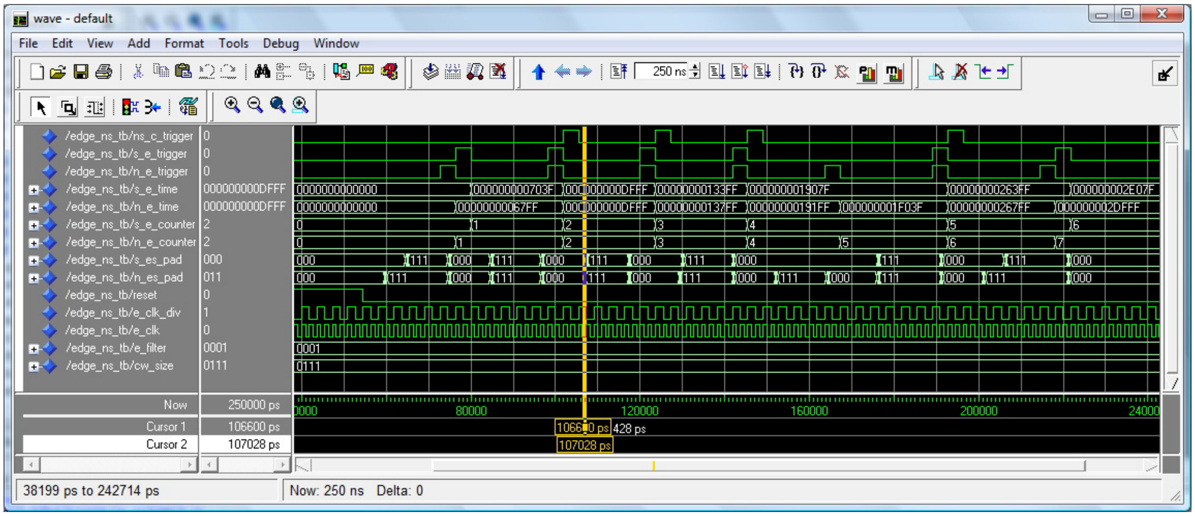


```
/edge_ns_tb/ns_c_trigger   0
/edge_ns_tb/s_e_trigger    0
/edge_ns_tb/n_e_trigger    0
/edge_ns_tb/s_e_time       000000000DFFF
/edge_ns_tb/n_e_time       000000000DFFF
/edge_ns_tb/s_e_counter    2
/edge_ns_tb/n_e_counter    2
/edge_ns_tb/s_es_pad       000
/edge_ns_tb/n_es_pad       011
/edge_ns_tb/reset          0
/edge_ns_tb/e_clk_div      1
/edge_ns_tb/e_clk          0
/edge_ns_tb/e_filter       0001
/edge_ns_tb/cw_size        0011

Now       250000 ps
Cursor 1  106600 ps
Cursor 2  107028 ps

38199 ps to 242714 ps    Now: 250 ns   Delta: 0
```

Fig. 3 Time coincidence window ≈ 600 ps.



```
/edge_ns_tb/ns_c_trigger   0
/edge_ns_tb/s_e_trigger    0
/edge_ns_tb/n_e_trigger    0
/edge_ns_tb/s_e_time       000000000DFFF
/edge_ns_tb/n_e_time       000000000DFFF
/edge_ns_tb/s_e_counter    2
/edge_ns_tb/n_e_counter    2
/edge_ns_tb/s_es_pad       000
/edge_ns_tb/n_es_pad       011
/edge_ns_tb/reset          0
/edge_ns_tb/e_clk_div      1
/edge_ns_tb/e_clk          0
/edge_ns_tb/e_filter       0001
/edge_ns_tb/cw_size        0111

Now       250000 ps
Cursor 1  106600 ps
Cursor 2  107028 ps

38199 ps to 242714 ps    Now: 250 ns   Delta: 0
```

Fig. 4 Time coincidence window ≈ 900 ps.

The fast signals n_es_pad (north channel) and s_es_pad (south channel) arriving at the FPGA pads from the north and south detector plates are time-sampled (Fig. 1) and produce the corresponding time words to be processed. To increase the sampling speed three pads with appropriate delays are used for each channel, effectively increasing the sampling rate (using DDR) to 3.3 GHz.

When an edge is detected the edge processing unit (e.g. for the north channel) generates the signals n_e_trigger (can be used as a trigger signal by the corresponding data acquisition channel as it is generated when a valid transition (edge) is detected), n_e_time (composed by a 40 bit coarse time plus the fine time identifying the instant of occurrence of the edge) and n_e_counter (a 32 bit counter counting the number of edges detected in the channel since the last reset).

Finally a coincidence signal is generated if the detected edges are within a specified time window. This signal can be used to validate the data acquisition in the corresponding channels, allowing collecting just the interesting data, which contributes significantly to reduce the total amount of data to save. This also reflects, in the image construction phase, in a marked reduction in the time taken to produce the pet image.

An interesting point of the developed algorithms 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 must return to zero (low) before a new edge can be detected.

V. Conclusions

The technique used to generate the coincidence signals is quite versatile and fast if a small number of channels are concerned. It may not scale well if used directly with a large number of channels, as the number of coincidences to check increases in a quadratic way (for a system divided in 50 north and 50 south channels there would be 2500 individual coincidences to check). Nevertheless many different alternatives based on the concept shown are possible and are being actively pursued for systems with a larger number of channels.

References

[1] P. Fonte, “High resolution timing of MIs with RPCs - a model”, *Nuclear Instruments and Methods in Physics Research A* 456 (2000) 6-10.

[2] A. Blanco, N. Carolino, C. M. B. A. Correia, L. Fazendeiro, N. C. Ferreira, M. F. F. Marques, R. F. Marques, P. Fonte, C. Gil, M. P. Macedo, “RPC-PET: a new very high resolution pet technology”, *IEEE Trans. Nuc. Sci.*, vol. 53, no. 5, pp. 2489-2494, Oct. 2006.