HHS Public Access

Author manuscript

Phys Med Biol. Author manuscript; available in PMC 2021 April 15.

Published in final edited form as:

Phys Med Biol. ; 66(5): 055009. doi:10.1088/1361-6560/abde85.

Field-programable-gate-array-based distributed coincidence processor for high count-rate online positron emission tomography coincidence data acquisition

Xinyi Cheng, **Kun Hu**, **Dongxu Yang**, **Yiping Shao**

Department of Radiation Oncology, University of Texas Southwestern Medical Center, Dallas, TX 75057, United States of America

Abstract

For positron emission tomography (PET) online data acquisition, a centralized coincidence processor (CCP) with single-thread data processing has been used to select coincidence events for many PET scanners. A CCP has the advantages of highly integrated circuit, compact connection between detector front-end and system electronics and centralized control of data process and decision making. However, it also has the drawbacks of data process delay, difficulty in handling very high count-rates of single and coincidence events and complicated algorithms to implement. These problems are exacerbated when implementing a CCP on a field-programable-gate-array (FPGA) due to increased routing congestion and reduced data throughput. Industry companies have applied non-centralized or distributed data processing to solve these problems, but those solutions remain either proprietary or lack full disclosure of technical details that make the techniques unclear and difficult to adapt for most research communities. In this study, we investigated the use of a set of distributed coincidence processors (DCP) that can address the CCP problems and be implemented relatively easily. Each coincidence processor exclusively connects one detector pair and selects coincidence events from this detector pair only, which breaks a centralized coincidence process to a collection of independent and parallel processes. DCP can significantly minimize the data process delay, maximize count-rates of coincidence events and simplify implementation by implementing a single coincidence processor with one detector pair and replicating it to the rest. A prototype DCP with 42 coincidence processors was implemented on an off-the-shelf FPGA development board for a small PET with 12 detectors configured with 42 detector pairs. DCP performances were tested with both pulsed signals and gamma ray interactions. There was no coincidence data loss up to the detector's maximum singles count-rate (250 k s^{-1}). Approximately 1.2 k registers were utilized for each coincidence processor and the FPGA resource utilization was proportional to the number of coincidence processors. Coincidence timing spectra showed the results from accurately acquired coincidence events. In conclusion: complementary to CCP, DCP can provide high count-rate capability, with a simplified algorithm for implementation and potentially a practical solution for online acquisition of a PET with a larger number of detector pairs or for ultrahigh-throughput imaging.

yiping.shao@utsouthwestern.edu.

Keywords

PET; coincidence processor; data acquisition; count-rate; FPGA

1. Introduction

Positron emission tomography (PET) relies on a coincidence processor (CP) to select valid coincidence events from acquired gamma interactions for reconstructing an image of a positron emitting radioisotope distribution (Bailey et al 2005, Cherry et al 2012). For an online CP that selects and outputs coincidence events as part of the data acquisition, a centralized CP (CCP) has been extensively used in many modern commercial and research PET scanners. The basic data processing of a typical CCP is shown in figure 1(a), where one single CCP consists of a timing sorter (sorter) to sort all detected 'singles' events in sequential timing order and a coincidence event selector (selector) to select coincidence events from these singles events for all valid detector pairs (e.g. defined detector pairs within the imaging field of view (FOV)). Such CCPs work very similarly to a computer central processing unit (CPU) and have the advantages of highly integrated circuits and compact signal connection, simplified data transfer between the detector front-end readout electronics and the CCP and centralized control of event selection processing and decision making (Yamada et al 2008, Tetrault et al 2010, Wang et al 2010, Njejimana et al 2013, Lewellen et al 2014, Shao et al 2014). On the other hand, with a single processor, CCP has several drawbacks that include the evitable processing delay from a single CP, which could lead to considerable lagging time and demand for large memory capacity for input data buffering and the implementation of complicated algorithms to compare and sort event timings of a large data of singles events (figure 2) (Song et al 2016). These drawbacks can be exacerbated if implementing a CCP solely on a field-program-gate-array (FPGA) as it can lead to further increased FPGA routing congestion and reduced data throughput capability. Mixed hardware and software online approaches have been used to mitigate this problem but can only partially remedy these CCP performance issues (Lewellen et al 2014, Joost and Salomon 2015, Peng et al 2016). In short, although a CCP works for most conventional PET imaging applications, the nature of its single-thread sequential processing can limit the imaging performance of a PET that has a large number of detector pairs or requires an ultrahigh data throughput. Additionally, it is usually a challenging task to implement a comprehensive high-performance FPGA-based CCP by most research groups due to their limited expertise and resource, which could impede the progress of PET instrumentation technology development.

There have been technological developments by commercial companies that use multiple coincidence processors to circumvent these problems. For example, a circular topology of electronics and data processing was implemented that uses a coincidence processor at each detector event processing level to compare the timing difference between the singles events from two detectors (Newport et al 2006, Hu et al 2011). Singles events are stored and transferred among detectors in a loop pattern for coincidence processing. This 'daisy-chain' based ring data processing and transmission topology provides a flexible, extensible and high-speed coincidence data processing. On the other hand, the technology is also

complicated and difficult for most academic groups to implement. For example, it uses a multiple-gigabit transmission interface on each detector for massive data routing.

We explored a different approach that uses of a set of distributed coincidence processors (DCP) which consist of Selectors only, with N Selectors for a PET with N defined detector pairs (figure 1(b)). No data transferring among different detector level data processing in a loop pattern is required since each selector exclusively connects to one detector pair and selects the coincident events from this detector pair only. In comparison to a CCP, the DCP breaks the centralized sorting and selection process into many individual detector-pair level selection processes, which works very similarly as a graphics processing unit (GPU) that uses the hardware and many-threads parallel process to divide the computation tasks and combine them into a final result to gain the acceleration of overall computations (Harris 2005). Each DCP Selector has an identical and relatively simple task. Without using a sorter that is process-consuming, the DCP has the advantages of minimal processing delay and therefore potentially ultrahigh data throughput even for a PET with a large number of detector pairs, a simplified selection algorithm that can be easily implemented for one detector pair and replicated to the rest and flexible modification for PET scanners with different detector configurations and sizes. The main caveat of this DCP architecture is the high demand of hardware resource for implementing many CPs. Nevertheless, the recent commercially available advanced FPGAs have substantially elevated real-time processing power and resource that make it feasible to enable the implementation of a DCP.

In this paper, we firstly introduce the hierarchy structure and components of a DCP, followed by the description of an initial implementation and performance test of a prototype DCP on a small animal PET scanner and then completed by the discussion of the DCP design related to the FPGA resource utilization and scalability, the advantages and limitations and potential advanced applications.

2. Materials and methods

2.1. Structure and component of DCP

The proposed DCP adopts a simple scatter/process/gather data processing structure that the input data are ‘scattered’ by design to local memories, processed by many computational cores and gathered from them as the final result (Shibata 2010). One example is the modern general-purpose GPU that trades hardware resources (e.g. memories and computation cores) to provide parallel processes and accelerate the computation. (Harris 2005, He et al 2007) A DCP with such data processing architecture can be implemented at a system-level FPGA with scatter, select and gather stages (figure 3).

Scatter-stage.—This stage is intended to receive the data of singles events transferred from all detectors, and multiplicate each singles event from a single detector and distribute them to selectors at the next select-stage according to a predefined map of valid detector pairs. Specifically, for a PET with M detectors and N valid detector pairs with N depending on the scanner configuration and FOV size, M inputs to the scatter-stage will be multiplicated to 2N and paired into N inputs to N selectors at the select-stage. Such a design provides a tree topology of event processing and data transfer that each selector exclusively

processes singles events from one designated valid pair of detectors with a deterministic latency of data processing and transmission, which eliminates the need for a sorter and simplifies the subsequent event processing.

Select-stage.—This stage is composed of N selectors that work in parallel. Each selector compares the timing difference between the two singles events from its designated single detector pair and selects a coincident event if the timing difference is within the predetermined coincidence timing window.

Gather-stage.—This stage collects the information of pairs of singles events that are in coincidence and packs and transfers these coincidence event data to the computer. Output of coincidence data from each selector is usually collected with equal probability.

In brief, a DCP has a sequential, modular event timing process through the three stages for each detector pair, and all timing processes with different detector pairs are independent and parallel. Compared to a CCP, this leads to distributed resources usage when implementing a DCP solely on a FPGA, with a simplified CP algorithm, minimized workload for each CP, and significantly reduced FPGA routing congestion.

2.2. Implementation of a DCP

We implemented a DCP on a small animal PET that has 12 detector panels and 42 detector pairs.

Detector.—The PET has 12 depth-of-interaction (DOI) measurable detector panels in a dodecagon configuration (figure 4). Each detector panel contains four closely tiled detectors and each detector consists of a 30 × 30 array of 1 × 1 × 20 mm^3 LYSO scintillator crystals, with each crystal end optically coupled to an 8 × 8 Silicon Photomultiplier array (model MPPC S13361-2050-08, Hamamatsu Photonics K.K.) for a dual-end-scintillator readout (Bircher and Shao 2012). Each MPPC active pixel size is 2 × 2 mm^2. Output electrodes of the combined 4 MPPC arrays on each detector end are connected into position-sensitive 16 rows of anodes and 16 columns of cathodes to reduce the total number of detector output channels from 512 to 64. The imaging FOV is ~10 cm diameter trans-axially and ~3.5 cm axially. There are a total of 42 defined detector-panel pairs for acquiring coincidence events within the FOV. The coincidence timing resolution from a pair of detector panels was measured around ~3.6 ns.

The detector front-end readout and processing electronics (front-end electronics for short) includes a dual-polarity sigma-delta circuit and carry-line based FPGA time-digital-convertor (TDC) with online self-calibration ability and ~70 ps (RMS) timing resolution (Wang et al 2015, Wang et al 2017, Cheng et al 2019). The output digitized information of each detected singles event includes the measured event energy, arrival timing, and the interaction position. The maximal detected singles event rate for each detector panel is 250 K s^{-1}. A mini display port (miniDP) cable connects the front-end electronics of a detector panel to the system data processing and acquisition electronics board (system electronics for short). Three differential signal pairs were used within each miniDP cable. Two down-streaming pairs were used to provide a 50 MHz clock and a synchronization signal from

system electronics to each detector and one up-streaming pair was used to provide 400-Mbps low-voltage-differential-signals (LVDS) for data transmission from each detector to the system electronics.

The front-end electronics can also input and process signals from a single-channel signal generator for debugging and testing. In this 'testing' mode, the test signal that mimicked the detected signal of a singles event was inputted through an onboard general-purpose-input-output (GPIO) pin to the selected detector front-end electronics, with the equivalent energy and arrival timing of each mimicked singles event being processed, transmitted and stored the same way as processing a detected singles event from a detector panel.

Hardware of the DCP.—As shown in figure 5, a commercial FPGA development board (DE4 development kit manufactured by Terasic Inc, that includes a Stratix IV GX EP4SGX230 FPGA manufactured by Intel Corporation) was used as the system electronics board to connect to all detector front-end electronics with 12 miniDP cables through a customized adapter board. The output coincidence events were transferred to a PC computer through a PCI-e cable.

Firmware of the DCP.—There are 12 scatter-stages that receive singles signals from 12 detector panels. Each scatter-stage contains several functional modules that include: (1) LVDS receiver that receives serially transferred signals from a detector panel and converts them into 8-bit formatted data. (2) Event decoder that unpacks and extracts the information of interaction position, energy and timing of a singles event. The event timing is recorded as a digital time stamp. (3) Timing compensator that compensates the different timing delays among detector panels due to different wiring, cable lengths and TDC timing delays. (4) Event multiplicator that multiplicates the data of timing-delay-compensated a singles event and distributes them to a predefined list of seven other detector panels that are paired with this detector panel, as shown in figure 4(b).

There are total 42 independent and parallel select-stages for 42 detector-panel pairs. Each select-stage contains two first-in-first-out (FIFO)-based data buffers for storing the data of singles events from the two detector panels and a selector for selecting coincidence events. Each selector compares the time stamps between the two singles events and outputs a coincidence event if the difference between the two timestamps is within the coincidence timing window. It takes four clock cycles for the selector to complete these coincidence event selection process for each pair of singles events. With a 200 MHz clock that was multiplicated from the 50 MHz system clock for coincidence processing, the maximal rate of coincidence event selection for each select-stage can theoretically reach 50 M event s^{-1}.

There are a total of 42 gather-stages that consist of 42 shift-register arrays and a high-speed PCI-e interface to transfer the data. Each shift-register array downloads exclusively the data of coincidence events from its designated select-stage. All 42 shift-register arrays download the data in parallel with 1 clock cycle, however, these data need to be sequentially shifted and transferred through the PCI-e interface to the computer with 42 clock cycles. Both download and shift processes of all shift-register arrays are synchronized with the 200 MHz

clock. Clearly, the initially implemented gather-stage is the most process-consuming stage of the DCP. The maximal data transfer rate of this gather-stage is~200 M event s^{-1}.

2.3. Performance of DCP

The following studies were conducted to test the functionality and evaluate the performance of the DCP:

Resource usage.—If implemented appropriately, the FPGA resource used by a DCP is linearly proportional to the number of detector pairs involved as the characteristics of the scatter/process/gather data processing structure. To validate this, different versions of DCP firmware with 6, 12, 18, 24 and 42 detector-panel pairs were compiled and tested and their usage of the adaptive look-up-table (ALUT), register and on-ship memory were recorded and compared.

Coincidence event count-rate performance.—Two studies were conducted. In the first study, rectangular pulsed signals generated by the waveform signal generator were split and fed to 2 detector front-end electronics boards in 'testing' mode. The pulse width was 300 ns. The event rate increased from 1 K s^{-1} to 250 K s^{-1} that is the maximum data processing rate of the detector front-end electronics. At each event rate, coincidence events were acquired with a 40 ns coincidence timing window for one minute.

Each coincidence event had its unique serial id number that was inserted to the paired pulse signals by the front-end electronics. These serial id number increased continuously from one coincidence event to the next before sending to the system-level FPGA for coincidence processing. If there is no coincidence event data loss from the coincidence processing, the difference between the id numbers from any two consecutively acquired coincidence events should equal to 1; if there is a data loss, the difference will be more than 1, and the quantity above 1 equal to the number of coincidence events lost.

In the second study, we expand the first study from 2 detectors to all 12 detectors. The rectangular pulsed signals generated by the waveform signal generator were replicated and fed to 12 detector front-end electronics boards in 'testing' mode and the same coincidence event measurement and data analyses procedures were followed for all 42 detector pairs. This study emulated a 'stress test' of the overall DCP count-rate performance with simultaneous acquisitions of all 42 distributed coincidence processors at the count rate up to the maximum limited by the detector front-end electronics.

PET acquisition with a ^{22}Na point source.—Coincidence events were acquired by the DCP with a 0.25 diameter and 277.9 KBq activity ^{22}Na point source at the center of the PET FOV. For this study, the minimal energy threshold was set at ~380 KeV. The online coincidence timing window width was set at 40 ns in order to acquire and test wide range true and random coincident events. The coincidence time spectrum was generated for measuring the coincidence time resolution.

With each acquired coincidence event, we drew a line between the two detected interaction positions. In principle, all such lines with the true coincidence events should pass through the FOV center, which let us visualize the accuracy of the coincidence event selection.

3. Results

3.1. Resource utilization

Table 1 shows the utilization of FPGA resources of ALUT, register and on-chip memory at different stages of the DCP implemented with 42 detector panel pairs. Not surprisingly, the select-stage uses the most resources (27.0%), followed by the gGather-stage (9.2%) and then the scatter-stage (2.7%). Among the three resources used, register is the dominating one (27.7%), followed by ALUT (11.0%) and memory (0.4%).

Figure 6(a) shows the utilization of registers and routing wires with the DCP were evenly distributed over the FPGA available resources, indicating evenly distributed resource usage and minimal routing congestion. As a comparison, figure 6(b) shows the utilization of registers and routing wires with the CCP were not evenly distributed over the FPGA available resources, indicating routing congestion and potential data processing delay.

Figure 7 shows the register utilization as a function of the number of detector-panel pairs. As expected, it shows that the register utilization with the DCP linearly increases with the number of detector-panel pairs.

3.2. Coincidence event count-rate performance

Figure 8(a) shows that a total of three coincidence events were lost for a pair of detectors as measured with 250 k s^{-1} pulsed signals over 1 s acquisition time. In fact, they were all lost during the acquisition deadtime due to the signal synchronization between the front-end and system electronics, as shown in figure 8(b). If neglecting these acquisition deadtimes, the coincidence event data loss due to the coincidence processing for each detector pair was 0 under the 250 k s^{-1} count-rate. Figure 8(c) shows that the measured count rates from all 42 distributed coincidence processors matched very well with the inputted emulated signal count rates. At the maximum input count rate of 250 K s^{-1}, the mean and standard deviation of measured count rates from all 42 coincidence processors were 249.997 K s^{-1} and 0.00034 K s^{-1} respectively.

3.3. PET acquisition with a ^{22}Na point source

Coincidence time spectrum.—Figures 9(a) and (b) show the coincidence time spectrum acquired from one typical and all detector-panel pairs. The coincidence time resolution (FWHM) measured from one and all detector-panel pairs were 3.6 ns and 4.1 ns, respectively. As a comparison, figure 9(c) shows an 89.1 ps coincidence time resolution measured from the rectangular pulsed signals with one detector-panel pair, indicating excellent coincidence timing process by the front-end readout electronics and coincidence event selection. The random coincidence events were mainly contributed from the LYSO background activity.

Projection data.—A 10 ns post-acquisition software timing window was applied to the data acquired from each individual detector-panel pair to separate the 'true' and 'random' coincidence events. Figure 10(a) shows the count distributions of all coincidence events, where the first 6 histogram bins were the ones associated with the 6 detector-panel pairs that were expected in coincidence when the ^{22}Na point source was at the center of FOV. Figures 10(b) and (c) show that most 'true' coincidence events (91.2%) were indeed within the expected six detector-panel pairs, while the random coincidence events were more or less uniformly distributed across all detector-panel pairs. It worth to mention that the above measured were raw counts without detectors' normalization.

Figure 11 shows total 1000 lines that were drawn from the selected true or random coincidence events, which visually validate the accuracy of the coincident event selection performed by the implemented DCP.

4. Discussion

The initial distributed CP was implemented solely on a low-cost, off-the-shelf FPGA development board that has been commercially available since 2008. The firmware programing was relatively straightforward by populating a relatively simple single CP algorithm. For the purpose of this initial demonstration, the gather-stage used a simple readily available shift-register method to transfer the coincidence events to the PC computer, which is sufficient for testing the small animal PET acquisition although it is a low-efficient single threaded process and can be enhanced with one of many high-efficiency data collection and transfer methods, such as a tree-like data merger or FPGA's high-speed data buses (e.g. Avalon by Intel Corporation, AXI by Xilinx Company), that can be used to significantly improve the efficiency of data gathering processing.

As demonstrated by this study, the resource utilization is linearly proportional to the number of detector pairs. Figure 12 shows the extrapolation of this linear relationship up to the level when implementing a DCP with several off-the-shelf FPGA boards, if the same firmware scheme will be applied. This conservative estimation indicates that the current DCP can be expanded to a PET with ~10 K detector pairs by using an existing FPGA board. No routing congestion or signal processing delay is expected since all CPs work in parallel and independently. Thus, the overall count-rate of a DCP can be in theory linearly proportional to the number of detector pairs.

We can further substantially extend the DCP capability beyond what the CCP can achieve. In contrast to a CCP that is difficult to be implemented beyond a single FPGA, CPU or mixed FPGA/CPU board, one unique advantage of the DCP is that it can be easily implemented by distributing CPs over several FPGA boards without concern over the complicated cross-board processing and coordination, which can be easily seen from figures 1 and 3. This scalability is really valuable in implementing an online CP for, as an example, a large axial FOV PET scanner with its number of detector pairs likely exceeding the capability of a single board based online CP. For example, due to far higher rates of singles and coincidence events at the system level, the latest long axial FOV total-body human PET has to use offline software to sort acquired singles events to select coincidence events

(Cherry et al 2018, Leung et al 2018). However, with the ratio of singles to coincidence events can reach more than 50:1, this alternative offline CP requires very large data storage and extended data processing time and is severely inefficient. More importantly, it prevents any opportunity of interventional study within the imaging session that could be invaluable for investigating some important subjects that are otherwise uniquely available with the total-body FOV and ultrahigh sensitivity, such as image-based brain-body dynamic responses to multiple external stimulations. With DCP, or other non-centralized multiple coincidence processors' approach, it is feasible to enable online total-body PET acquisitions so that dynamic images can be interpretated and used to guide such interventional studies and possibly new discoveries.

Overall, the conventional CCP and the investigated DCP are complementary in terms of implementation and resource use. While both can provide comparable performance for most PET systems with moderate number of detector pairs and standard imaging applications, DCP provides solutions beyond the CCP's performance limit when a targeted application involving a large number of detector pairs, high-throughput imaging, real-time data acquisition, etc, although with the caveat of requiring substantially more FPGA resources.

It is worth noting that many different versions of distributed coincidence processors can be implemented. The conventional CCP and the investigated DCP are two extremes of centralized and parallelized coincidence processing approaches. In practice, particularly for a PET with large number of detector pairs, it may be more practical, economic, or necessary to have non-centralized coincidence processing with multiple CPs, where each CP handles several detector pairs. Such an approach with combined centralized and distributed coincidence processes may provide a balance between technical complexity, performance, cost and implementation requirement for different applications.

5. Conclusion

More than an alternative to the conventional CCP, DCP replaces a single-thread centralized coincidence process with many-thread, independent and parallel coincidence processes to provide very attractive advantages over CCP, such as easy implementation, ultrahigh-throughput acquisition, extended scalability beyond a conventional single-board CP. With recently available advanced and affordable FPGA boards, a practical DCP can be implemented with much less technical challenges than implementing a CCP and can be applied to enhance the PET imaging capability and performance that are limited by CCP, as well as to accelerate the PET instrumentation and imaging technology development with substantially reduced technical complexity, effort and expense in implementing a comprehensive and sophisticated PET coincidence processor.

Acknowledgments

This study is supported in part by grants from National Institute of Health (1R01EB019438, 1R01EB01943804S, 1R01CA218402) and a career development grant from the Department of Radiation Oncology, University of Texas Southwestern Medical Center.

References

Bailey DL, Maisey MN, Townsend DW and Valk PE 2005 Positron Emission Tomography (Berlin: Springer)

Bircher C and Shao Y 2012 Use of internal scintillator radioactivity to calibrate DOI function of a PET detector with a dual-ended-scintillator readout Med. Phys 39 777–87 [PubMed: 22320787]

Cheng X, Hu K and Shao Y 2019 Dual-polarity SiPM readout electronics based on 1-bit sigma-delta modulation circuit for PET detector applications IEEE Trans. Nucl. Sci 66 2107–13 [PubMed: 33746246]

Cherry SR, Jones T, Karp JS, Qi J, Moses WW and Badawi RD 2018 Total-body PET: maximizing sensitivity to create new opportunities for clinical research and patient care J. Nucl. Med 59 3–12 [PubMed: 28935835]

Cherry SR, Sorenson JA and Phelps ME 2012 Physics in Nuclear Medicine (Philadelphia: Saunders)

Harris M 2005 Mapping computational concepts to GPUS ACM SIGGRAPH 2005 Courses, SIGGRAPH 2005 50–72

He B, Govindaraju NK, Luo Q and Smith B 2007 Efficient gather and scatter operations on graphics processors Proc. of the 2007 ACM/IEEE Conf. on Supercomputing, SC'07

Hu D, Siegel S B, Wu H, Wen J, Ravindranath B and Tai Y 2011 Extending QuickSilver™ electronics for PET insert integration 2011 IEEE Nuclear Science Symp. Conf. Record 3247–9

Joost R and Salomon R 2015 CDL, a precise, low-cost coincidence detector latch Electronics 4 1018–32

Leung EK, Judenhofer MS, Cherry SR and Badawi RD 2018 Performance assessment of a software-based coincidence processor for the EXPLORER total-body PET scanner Phys. Med. Biol 63 18NT01

Lewellen TK, Dewitt D, Miyaoka RS and Hauck S 2014 A building block for nuclear medicine imaging systems data acquisition IEEE Trans. Nucl. Sci 61 79–87

Newport DF, Siegel SB, Swann BK, Atkins BE, McFarland AR, Pressley DR, Lenox MW and Nutt RE 2006 QuickSilver: a flexible, extensible, and high-speed architecture for multi-modality imaging 2006 IEEE Nuclear Science Symp. Conf. Record 2333–4

Njejimana L et al. 2013 Design of a real-time fpga-based data acquisition architecture for the labpet ii: an apd-based scanner dedicated to small animal pet imaging IEEE Trans. Nucl. Sci 60 3633–8

Peng Q, Netscher GM, Huber JS, Vu C, Moses WW and Choong WS 2016 Firmware and software framework of OpenPET electronics system for radiotracer imaging 2014 IEEE Nuclear Science Symp. and Medical Imaging Conf., NSS/MIC 2014 31–3

Shao Y, Sun X, Lan KA, Bircher C, Lou K and Deng Z 2014 Development of a prototype PET scanner with depth-of-interaction measurement using solid-state photomultiplier arrays and parallel readout electronics Phys. Med. Biol 59 1223–38 [PubMed: 24556629]

Shibata N 2010 Efficient evaluation methods of elementary functions suitable for SIMD computation Computer Science—Research and Development 25 25–32

Song W, Koch D, Lujan M and Garside J 2016 Parallel hardware merge sorter Proc.—24th IEEE Int. Symp. on Field-Programmable Custom Computing Machines, FCCM 2016 95–102

Tetrault MA, Oliver JF, Bergeron M, Lecomte R and Fontaine R 2010 Real time coincidence detection engine for high count rate timestamp based PET IEEE Trans. Nucl. Sci 57 117–24

Wang C, Li H, Ramirez RA, Zhang Y, Baghaei H, Liu S, An S and Wong W-H 2010 A real time coincidence system for high count-rate TOF or non-TOF PET cameras using hybrid method combining AND-logic and time-mark technology IEEE Trans. Nucl. Sci 57 708–14 [PubMed: 20676230]

Wang Y, Liu C, Cheng X and Li D 2015 Spartan-6 FPGA based 8-channel time-to-digital converters for TOF-PET systems 2015 IEEE Nuclear Science Symp. and Medical Imaging Conf. (NSS/MIC) 1–3

Wang Y, Kuang J, Liu C and Cao Q 2017 A3.9-ps RMS precision time-to-digital converter using ones-counter encoding scheme in a Kintex-7 FPGA IEEE Trans. Nucl. Sci 64 2713–8

Yamada R, Watanabe M, Omura T, Sato N, Shimizu K, Takahashi M, Ote K, Katabe A, Moriya T and Sakai K 2008 Development of a small animal PET scanner using DOI detectors IEEE Trans. Nucl. Sci 55 906–11

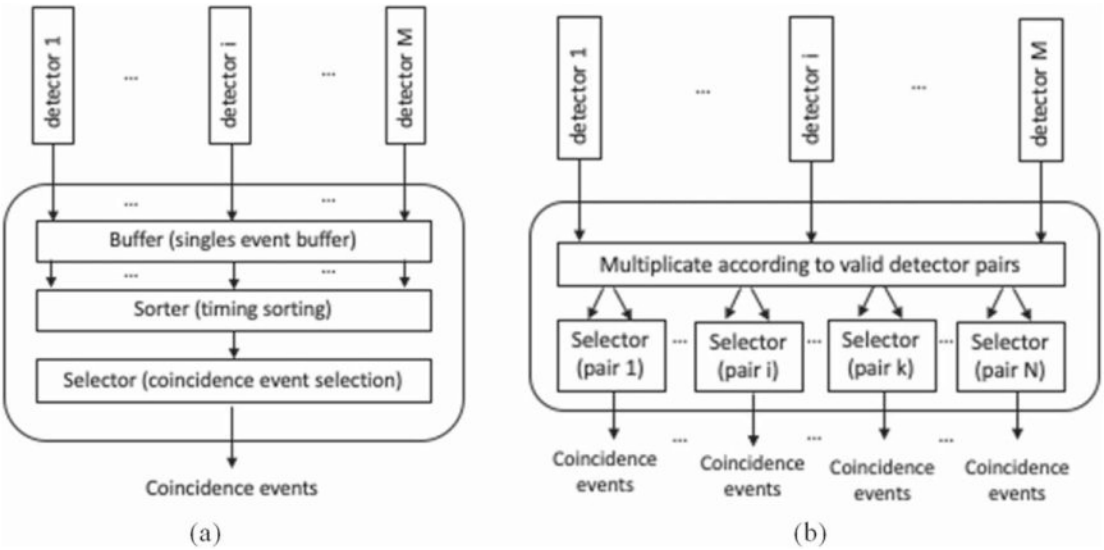

Figure 1.

(a) Schematics of centralized coincidence processor with a single-tread coincidence event selection process. (b) Schematics of distributed coincidence processors with many-tread coincidence event selection processes.

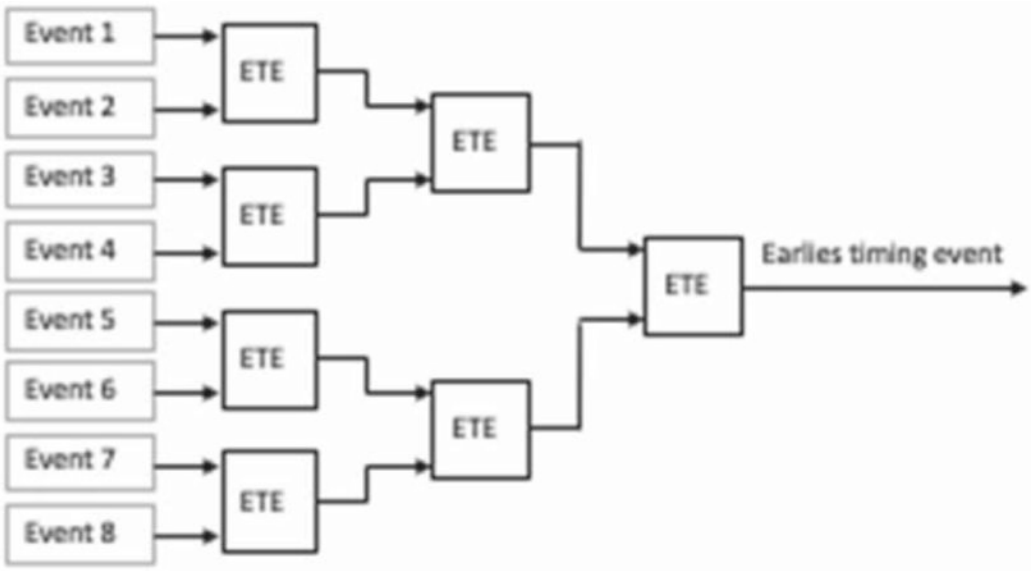

Figure 2.

One implementation of ‘tree’ timing sorter for CCP. Input with the earliest singles event (with lowest time stamp) is selected by a series of earliest event extractors (ETE), which leads to FPGA routing congestion and signal process delay when with large number of input singles events or during high count-rate acquisition.

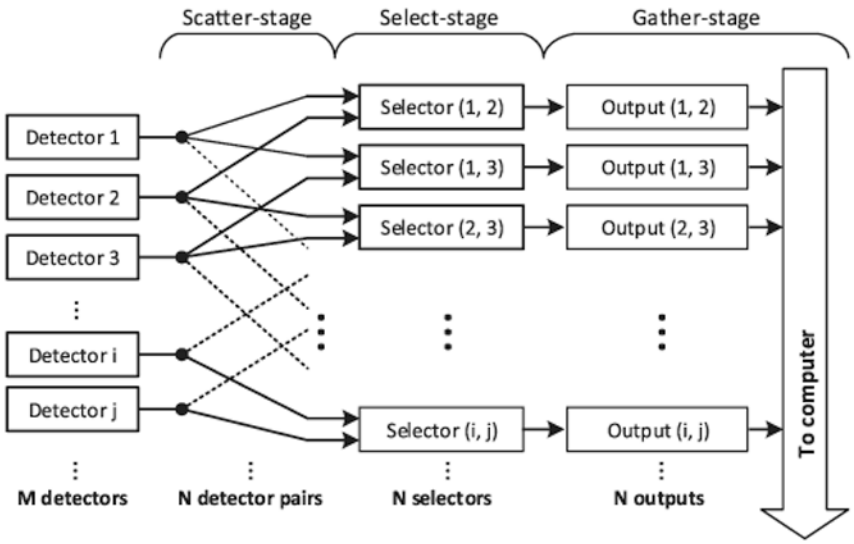

Figure 3.

Schematics of a 3-staged DCP event processing architecture in a tree topology with M detectors and N possible detector pairs.

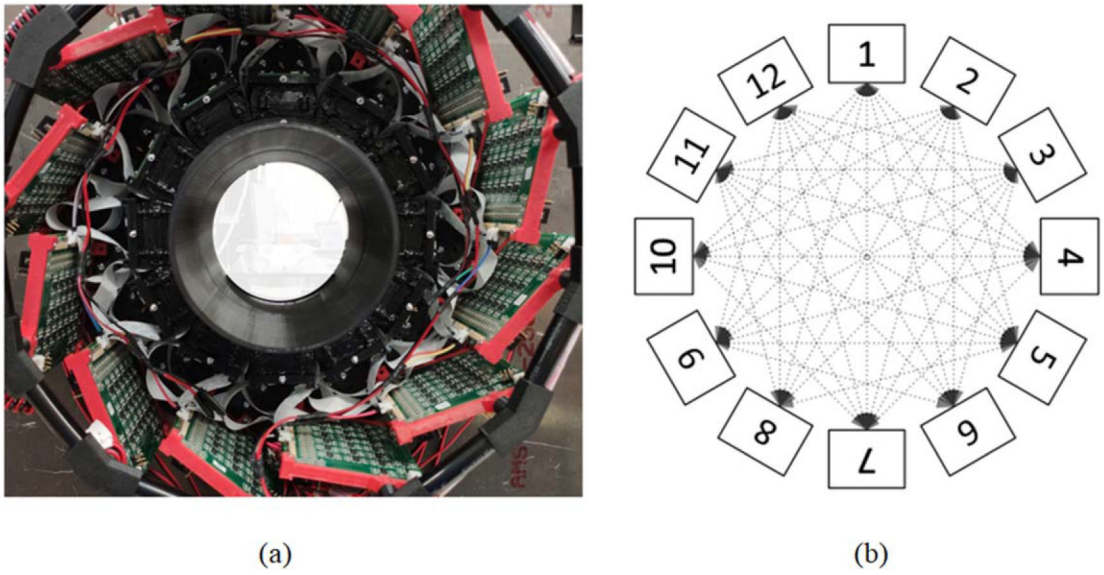

Figure 4.
(a) The small animal PET scanner with 12 detector panels. (b) Configuration of valid detector-panel pairs for coincidence data acquisition.

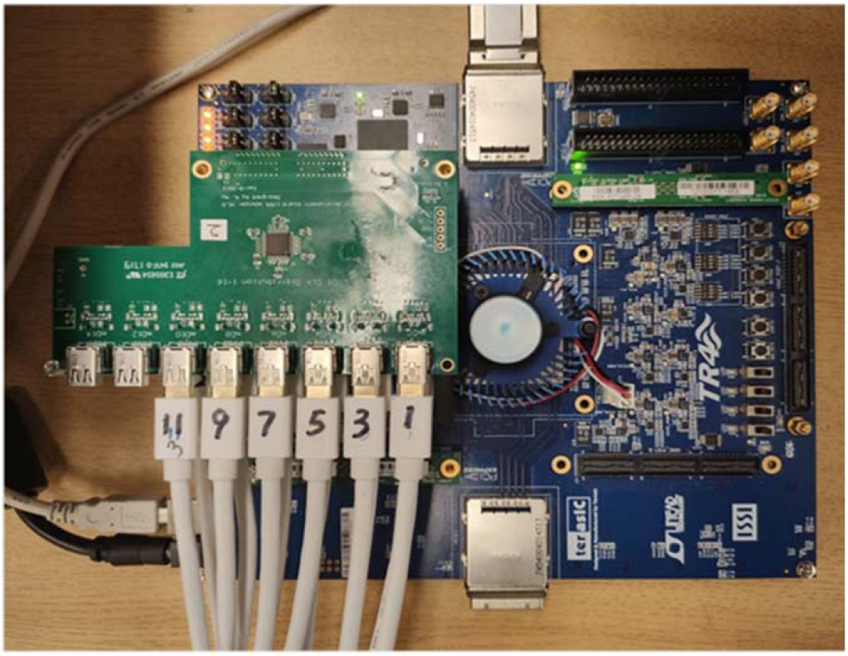

Figure 5.

The event signals from 12 detector panels were transferred to system-level FPGA by 12 miniDP cables and processed with the DCP. The coincidence events were transferred to a PC computer through a PCI-e cable.

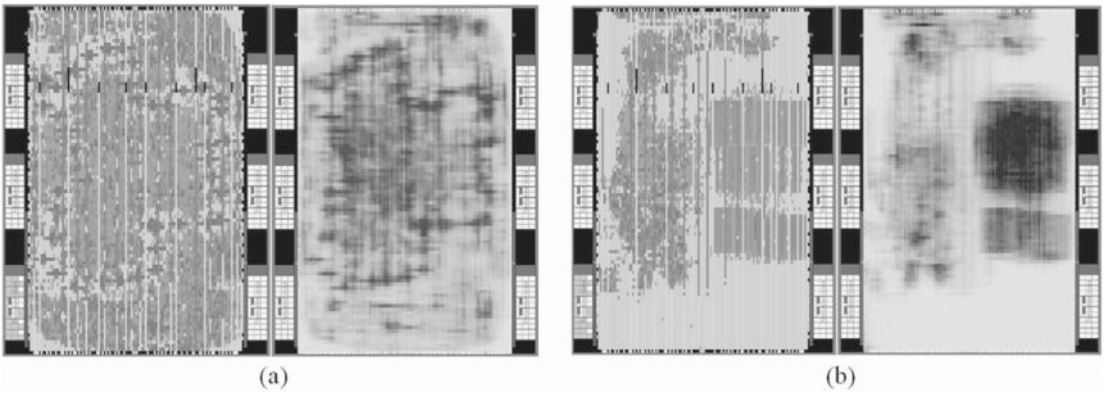

Figure 6.

(a) Reported resource utilization of the DCP with 42 CPs for 42 detector pairs exported from Intel Quartus II's chip planner. Intensity of the gray scale is proportional to the percentage utilizations of registers (left) and routing wires (right). (b) Similar reported resource utilization of the CCP with 1 CP for 42 detector pairs.

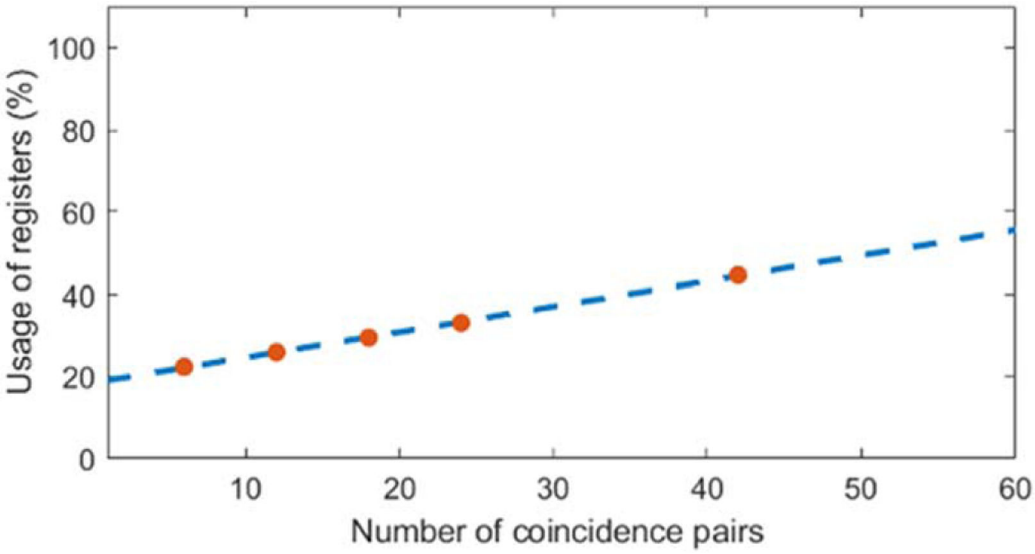

Figure 7.
Utilization of registers with the DCP as a function of number of coincidence processors. A linear function fits them very well.

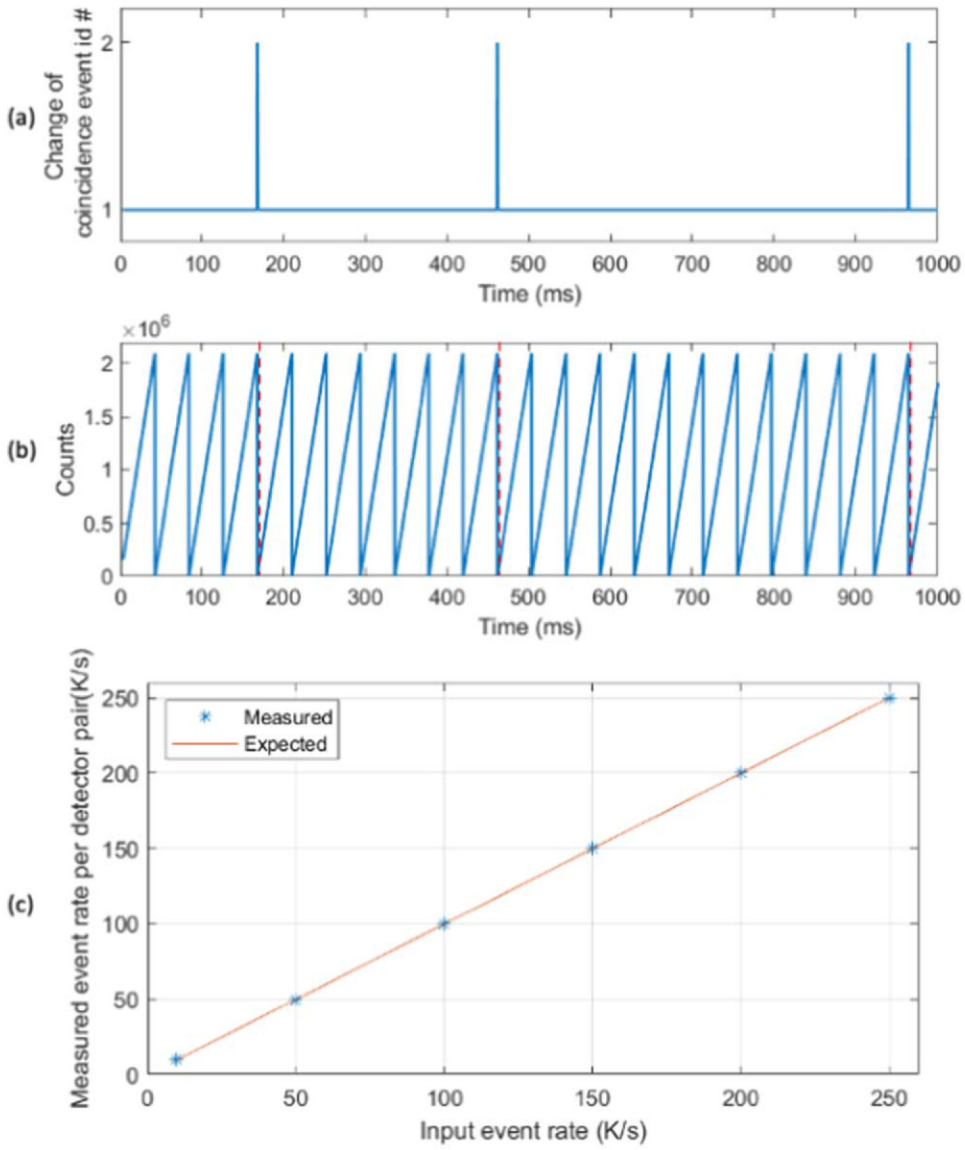

Figure 8.

(a) Measured change of coincidence event id numbers over the 1 s coincidence event acquisition time. It shows that three coincidence events were lost due to coinciding with three acquisition system resets. (b) Counts of 50 MHz system clocks over the same 1 s acquisition time, which repeated in each 41.943 ms with the periodical signal synchronizations between the front-end and system electronics. The acquisition deadtime due to the signal synchronization was 20 ns (1 clock cycle). Three red dashed lines indicate the times when coincidence events were lost due to the acquisition deadtime. (c) The measured count rates from all 42 coincidence processors matched very well with the input count rates.

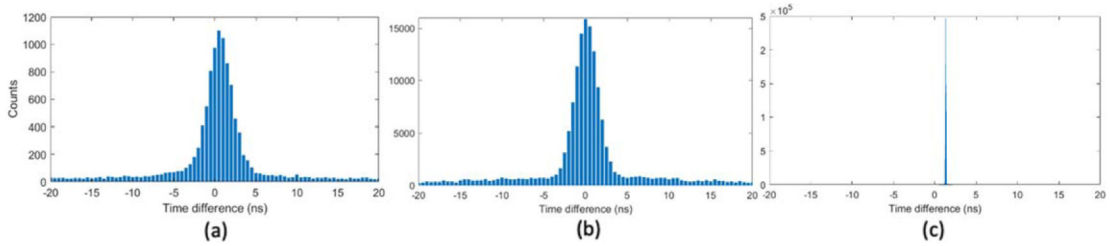

Figure 9.

(a) Coincidence timing spectrum acquired with the ^{22}Na point source from one detector-panel pair, and (b) from all detector-panel pairs. (c) Coincidence timing spectrum acquired with pulsed signals.

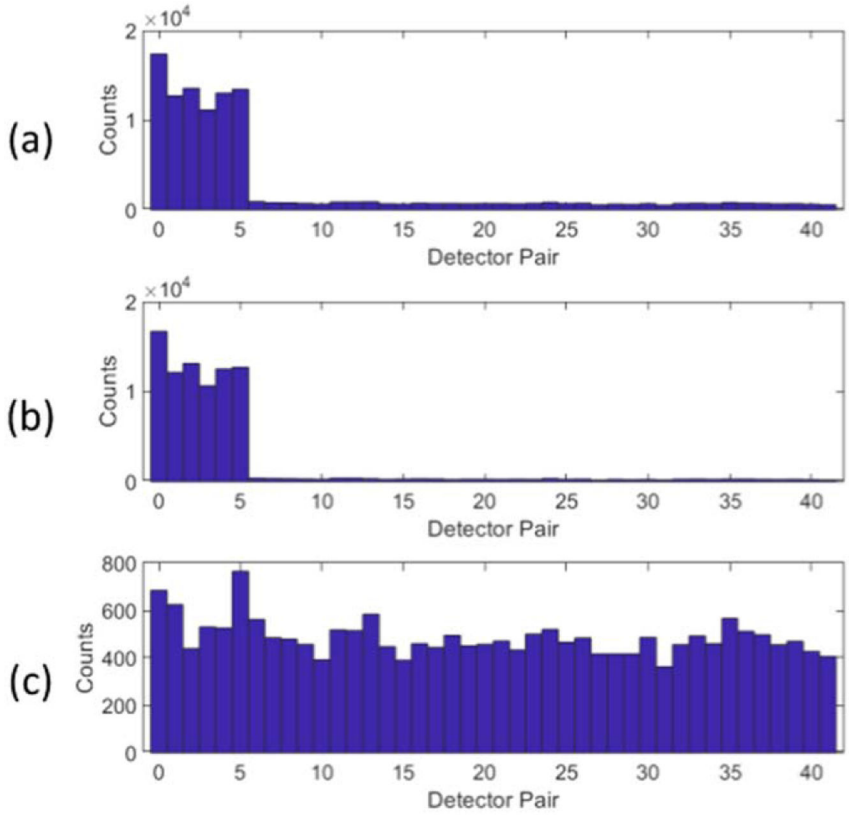

Figure 10.

Histogram of coincidence events acquired from all 42 detector pairs. (a) With all events acquired. (b) With true coincidence events only. (c) With random coincidence events only.

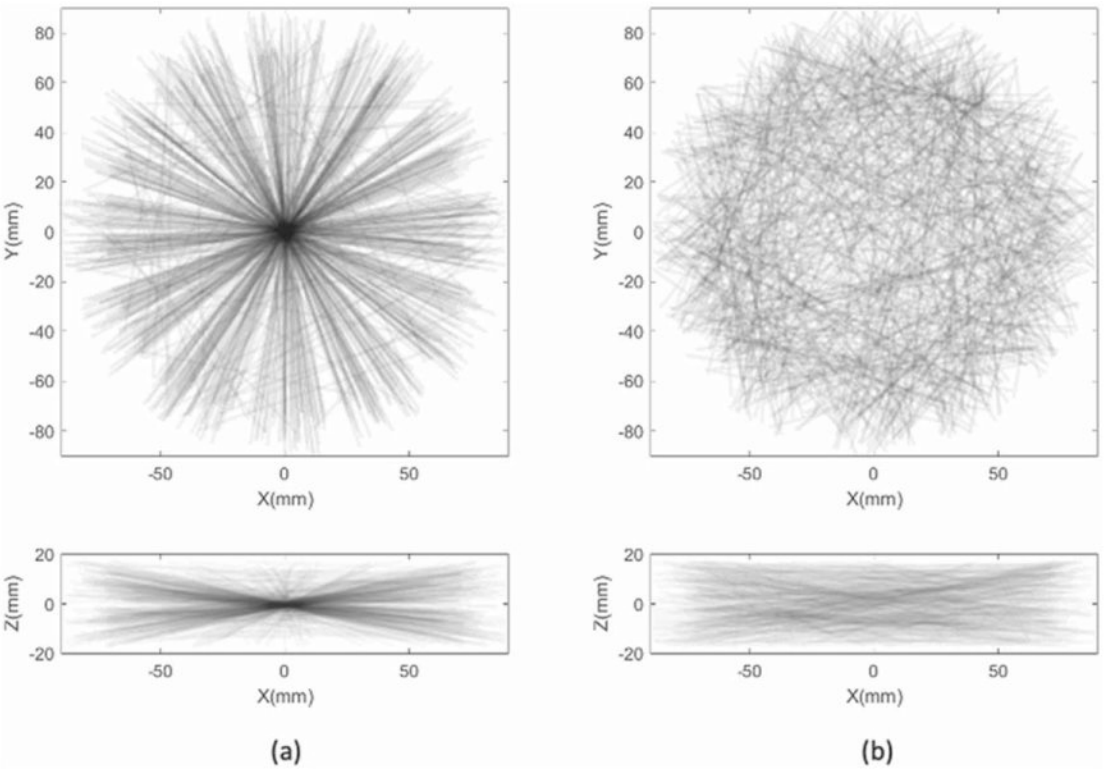

Figure 11.

Lines of responses drawn from the acquired (a) true and (b) random coincidence events. Top and bottom panels show the trans-axial and axial views.

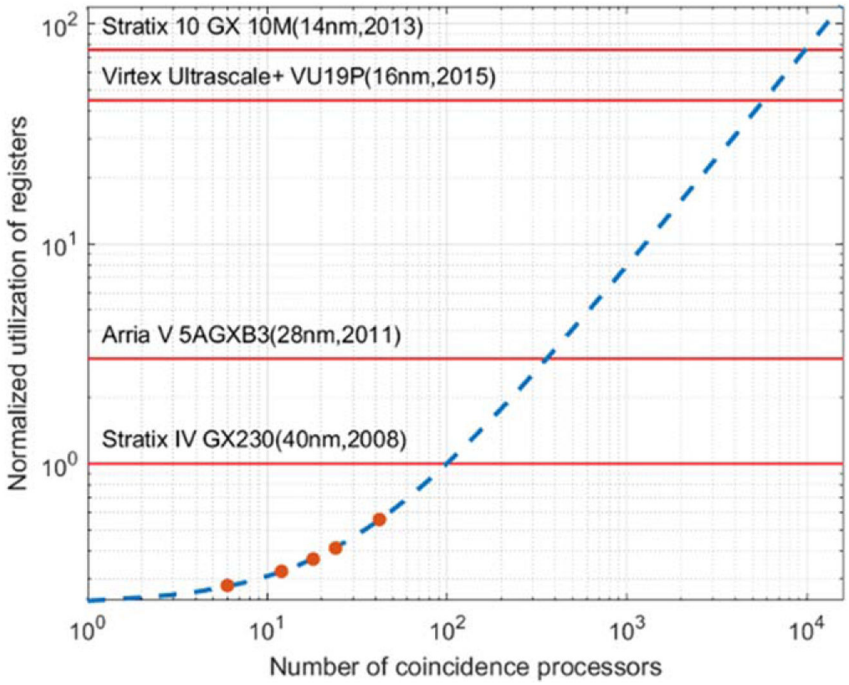

Figure 12.

Utilization of FPGA registers as a function of number of coincidence processors. The normalized one represents the available registers (80% of the total register capacity) that can be utilized for DCP based on the same FPGA board used for the prototype DCP implementation. Several common off-the-shelve FPGA boards with their listed models, chip fabrication technology and the year on the market are also shown with their available registers that can be utilized for DCP.

Table 1.

FPGA resource utilization by different DCP processing stages.

Distributed Coincidence Processor	ALUT	Register	Memory (Bits)
Scatter-stage	0.7 K (0.7%)	3.7 K (2.0%)	6.1 K (0.0%)
Select-stage	8.1 K (8.9%)	32.8 K (18.0%)	21.5 K (0.1%)
Gather-stage	1.1 K (1.2%)	14.1 K (7.7%)	44.3 K (0.3%)
Sum	**9.9 K (11.0%)**	**50.6 K (27.7%)**	**71.9 K (0.4%)**
Resource capacity	91.2 K (100.0%)	182.4 K (100%)	14.6 M (100%)