

A Sensor Network Architecture for Digital SiPM-Based PET Systems

Claudio Bruschini, *Senior Member, IEEE*, Chockalingam Veerappan, *Member, IEEE*,
 Francesco Gramuglia^{id}, *Student Member, IEEE*, Martijn Bijwaard, Zoltan Papp,
 and Edoardo Charbon^{id}, *Fellow, IEEE*

Abstract—Digital silicon photo-multipliers (SiPMs) have emerged in the recent past as a viable low cost alternative to photomultiplier tubes in positron emission tomography systems, providing multiple timestamps, energy and scintillation coordinates at high spatial granularity as well as MRI compatibility. The rich but large datasets generated by digital SiPM sensors have posed a data preprocessing and acquisition challenge, at the sensor, module and system level when a multitude of such sensors are to be used. In this paper, we present a sensor network-based approach for data acquisition, scalable to multiring configurations, whereby each module acts as an autonomous sensing and computing unit, capable of determining in real time basic information for each scintillation event and communicating it to its peers. The proposed architecture is equally applicable to modules based on analog SiPMs with local digitization. Coincidence detection can then take place in the ring itself, in a deferred and distributed manner to ensure scalability and allow to fully process only the fraction of the total events which corresponds to true coincidences. Simulations and experimental results show that it is indeed possible to handle the system level challenges associated with digital SiPMs at data rates compatible with realistic configurations, including event packet transfers and real-time coincidence detection, using Gb/s serial communication links for internode communication. The downside of the proposed architecture is represented by the need, at module level, for additional connectivity and processing power. We also address possible solutions for network-based clock synchronization, in particular a hybrid scheme, combining a hard-wired ring clock distribution network with a network-based clock offset estimator. The latter was tested in an 8-node system, performing synchronization in real time with a worst-case phase estimator stability (standard deviation of the clock phase offset estimation

between two nodes) of about 160 ps, and substantial room for improvement (at least 5× according to simulations).

Index Terms—Medical imaging, photon detection, positron emission tomography (PET), synchronization, time-of-flight.

I. INTRODUCTION

IMAGE sensing and reconstruction techniques play a key role in modern medical diagnostics, enabling in particular the early detection of several diseases, such as cancer. Positron emission tomography (PET) [1]–[3] is a prime example of a noninvasive medical technique, which has enabled the functional imaging in 3-D of oncological, neurological, and cardiovascular diseases. PET systems map the *in vivo* distribution of positron emitting radioactive isotopes, such as ¹⁸F, ¹⁵O, or ¹¹C via the detection of the 511 keV gamma photons resulting from the positron’s annihilation, and which are emitted in pairs back-to-back.

Broadly speaking, a PET system is composed of three main components: 1) gamma-ray detection modules (generally organized in a ring fashion surrounding the organ under examination); 2) a data acquisition (DAQ) system coupled to one or more coincidence processing units, and an image reconstruction engine.

Nowadays, the detection modules rely on inorganic scintillating crystals, such as LSO/LYSO, in which the gamma photons deposit their energy when they interact inside the crystal via photo-effect or Compton scattering. In this process, they transfer their energy fully or partially to an electron, which in turn interacts with the crystal producing scintillation photons, in the visible/near UV range and over a typical timescale of a few hundred nanoseconds. The latter are then detected by fast and sensitive devices, either photomultiplier tubes (PMTs) or, increasingly in modern systems, by solid-state photodetectors, such as avalanche photodiodes or silicon photo-multipliers (SiPMs), which provide a processable analog or digital electrical output signal. Solid-state photodetectors also address the need for higher pixel granularity and timing response, while ensuring MRI compatibility (creating the possibility of dual PET-MRI systems [4]–[6]) and reduced size.

From the engineering perspective, there are three main parameters which characterize a scintillation event, and which need to be provided to the image reconstruction unit: 1) the deposited energy; 2) the time-of-arrival of the 511 keV

Manuscript received February 13, 2018; revised June 21, 2018; accepted August 2, 2018. Date of publication August 23, 2018; date of current version November 1, 2018. This work was supported in part by the European Community within the Seventh Framework Programme ICT Photonics through SPADnet project, and in part by the Swiss National Fund Project under Grant 200021-166289. (Claudio Bruschini and Chockalingam Veerappan contributed equally to this work.) (Corresponding author: Francesco Gramuglia.)

C. Bruschini, F. Gramuglia, and E. Charbon are with IMT, École Polytechnique Fédérale de Lausanne, 2002 Neuchâtel, Switzerland (e-mail: claudio.bruschini@epfl.ch; francesco.gramuglia@epfl.ch; edoardo.charbon@epfl.ch).

C. Veerappan and M. Bijwaard were with IMT, Delft University of Technology, 2628 Delft, The Netherlands. They are now with the Dialog Semiconductors, Reading, U.K., and also with Datawell B.V., Haarlem, The Netherlands (e-mail: vr.chockalingam@gmail.com; martijn.bijwaard@gmail.com).

Z. Papp is with the Mediso Medical Imaging Systems, Budapest, Hungary (e-mail: zoltan.papp@mediso.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TRPMS.2018.2866953

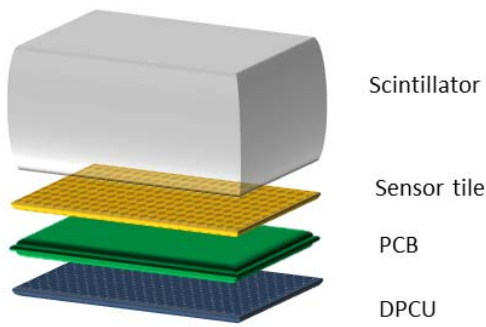


Fig. 1. Schematic representation of a photonic module, embedding four components: a scintillating crystal (such as LYSO, LSO, or BGO) that converts the incoming gamma radiation in optical photons; an SPAD-based sensor tile, where the optical photons are detected and counted/timed; a PCB to interface the sensor to the rest of the module; a motherboard embedding an FPGA to preprocess the data, hosting a DPCU.

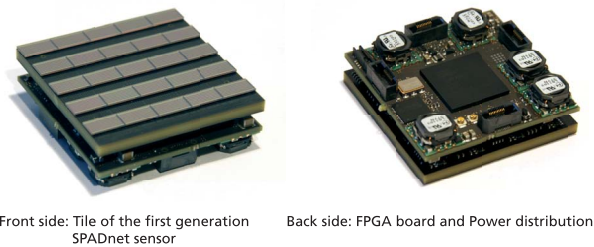


Fig. 2. SPADnet photonic module: left, SPADnet1 sensor tile assembly; right, FPGA processing and communication board. Actual size: $5 \times 5 \text{ cm}^2$.

gamma; and 3) the spatial scintillation coordinates over the (x, y) focal plane, and possibly in z (depth of interaction) as well to correct for parallax problems.

The timing information is of crucial importance to enable the separation of the real coincidences from singles or random events. This task is carried out by the coincidence processing unit(s), which is/are implemented either as a dedicated electronic circuit [7]–[9], in the firmware of a field-programmable gate array (FPGA), or in software [2], [10], [11]. Furthermore, accurate timestamps are used to constrain the position of the annihilation along the line-of-response between the two modules which have registered a true coincidence event. For high accuracy time-of-flight systems (featuring a timing precision of a few hundred ps), this leads to significant improvements of a system’s noise equivalent count rate (NECR) and thus of the signal-to-noise ratio of the reconstructed images [3], [4].

Modern solid-state-based detection modules enable the possibility of moving the data processing layer closer to the detectors, thereby creating compact detection units, or photonic modules (Fig. 1), usable in a stand-alone and modular fashion, and capable of high data rates. This can be done either by employing analog SiPMs, whereby a separate ASIC is needed to process the sensor outputs, or by means of arrays of CMOS-integrated single-photon avalanche diodes (SPADs), also termed digital SiPMs [12]–[14] (Fig. 2). This approach has the potential of leading to compact and flexible detector stacks, enabling full use of the inherently digital properties of SPADs through their migration to deep submicron CMOS processes, possibly with built-in processing capabilities, such as

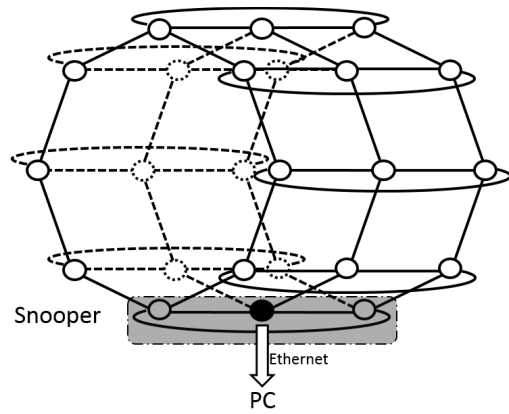


Fig. 3. Proposed SPADnet cylindrical mesh topology. The network is partitioned in rings, with each photonic module (represented by a white circle) connected to its nearest neighbors. Each ring (3 in this example) includes a functional node called snooper, with one of them designated to function as a master, in charge of managing the communication with a host PC.

timestamping and additional intelligence [15]–[17]. We called the approach “SPADnet.” In SPADnet, CMOS SPADs, along with modern microelectronics 3-D integration capabilities, can thus lead to highly granular pixel arrays capable of timestamping individual photons in a flexible and programmable way. They need, however, to be developed together with the appropriate system architectures in order to handle the data volume generated for each potential gamma event in an efficient and accurate manner.

In addition, in a scenario, where local timestamps are generated, the distribution of precise and globally valid timing information becomes necessary. This can in principle be obtained by distributing an accurate global reference clock to all photonic modules, e.g., employing timing modules, such as the IDT ICS8534-01. Such a clock signal can, however, suffer from routing delays due to the signal cables and traces to its destinations, which can be compensated for if fixed, and from jitter, which can deteriorate the timing resolution [10]. It would therefore be advisable to include a reliable and scalable clock distribution scheme as well when considering the overall system architecture.

In SPADnet, we propose a modular and scalable PET system based on networked, digital photonic modules, aimed at handling the challenges associated with DAQ when hundreds of digital SiPMs are used, or equally, analog SiPMs with local digitization. The photonic modules building up the detector are interfaced in a ring-like fashion to their nearest neighbors (i.e., including a scalable multiple axial ring topology, Fig. 3). The communication protocol is organized in such a way that coincidence detection takes place in real time in the ring itself, in a deferred and distributed manner [15], [18], rather than in *ad hoc* coincidence-processing units. Each photonic module comprises a sensor tile, matched to an equally sized, FPGA-based digital module containing a data processing and communication unit (DPCU). The entire photonic module (Fig. 2) has additional in-built gamma detection, processing, and communication intelligence so as to generate compact data packets, comprising the estimated values of energy, timing, and spatial coordinates of the scintillation. These packets are then

used, in a two-stage process, to perform coincidence detection in the network, in a module's field of view, followed by the transfer of only the true coincidences in real time to an external (reconstruction) computer through a dedicated (snooper) module and a Gigabit Ethernet link. The first stage consists of only small packets with basic timing information, whereas the second relies on larger packets but only for the detected coincidence events (true events).

The network was implemented on a Spartan-6/Virtex-6 backbone, and featured a data rate of over 2 Gb/s in module-to-module communication for a maximum processing capability of up to 3.3 million events per second per photonic module.

Finally, we address possible solutions for network-based clock synchronization. The combination of timing resolution requirements and of the configurability of a network-based PET makes the problem harder and the stability of the synchronization very difficult to maintain [19], compared to solutions traditionally employed in sensor networks. We opted for a hybrid scheme, combining a hard-wired ring clock distribution network with a network-based clock offset estimator. This allows to preserve scalability while maintaining high precision.

This paper is organized as follows. Section II reviews the main DAQ challenges of PET systems, comparing typical network topologies to that of SPADnet. Section III details the SPADnet photonic module, while Section IV provides a system level description, before entering into the details of packet handling techniques (Section V) and the main processing and communication units (CUs) (Sections VI and VII). Section VIII then describes the clock distribution and synchronization, and Section IX the main experiments and corresponding results.

II. DAQ CHALLENGES AND NETWORK TOPOLOGIES

Table I summarizes some of the key parameters that characterize the performance of typical PET systems [20]. As an example, a preclinical PET system, when designed using 5×5 cm² SPADnet sensor tiles, would generate a theoretical maximum of around 420 Gb/s of raw data, with similar figures for other PET modalities. The implementation of a data filtering techniques in PET systems, such as pile-up reduction, energy windowing, and real-time coincidence detection [21], is therefore mandatory to reduce the data transfer rates to manageable levels.

Several PET DAQ system topologies have been developed over the years, as well as combinations thereof, to address the design tradeoffs inherent in PET systems [20] (Fig. 4).

- 1) *Tree Topology*: The boards are connected in a hierarchical structure. The tree topology represents the first topology used in PET. Its main advantage is the small signal delay and a high throughput rate, however, at the price of an inefficient network usage [22]–[26].
- 2) *Daisy-Chain Topology*: Each DAQ board is connected to its neighbors in a chain-like fashion. This architecture allows the easy scaling of the system and makes it quite flexible, at the price of a long signal delay and the need for a high-speed network for each link [27], [28].

TABLE I
TYPICAL SPECIFICATIONS OF PET SYSTEMS

Characteristics	Small animal PET	Brain PET	Whole-body PET
Bore diameter (cm)	6	30	60-80
Axial field-of-view (cm)	10-12	2-18	15-20
Detection area (cm ²)	200	1400	4000
Crystal pixel size (mm)	1.2-3	2.5-2.5	4-5
Number of crystals	10k-40k	10k-120k	10k-40k
Radioactivity inside bore (mCi)	0.5	2	20
Sensitivity (%)	<10%	<7%	<2%
Spatial resolution (mm)	0.5-1	1.5-2	3-5
Measurement time (min)	3-10	20-40	25-50
Number of channels	400-600	4500	400-1200
Coincidence timing window (ns)	2.6	6	<6
Timing resolution (ns)	1.3	3	0.25-3
Energy resolution (%)	12-18	12-14	10-16
Singles event input rate (count/s)	10M	20M	<60M
Coincidence event throughput (count/s)	500k	1M	<1M
# of coincidences for an image (events)	100M	200M	200M

(Adapted from [20].)

- 3) *Bus Topology*: The event packets are transmitted through a bus that is shared by all the DAQ boards. This topology has the advantage, with respect to the daisy-chain one, of a small propagation delay provided by the use of a shared bus to broadcast the data. The main limitation is given by the need for a high network speed [29]–[31].

The chosen network architecture should enable sufficient speed and bandwidth to process and transmit events up to the peak NECR. Considering the steady performance increase of high-speed communication devices available on the market, a migration to architectures, such as daisy-chain and bus seems reasonable. Another increasingly crucial point is related to the scalability of the system as well as its flexibility, all the more so with the appearance of novel whole-body PET paradigms, which call for digitally interfaced and network-enabled photonic modules, due to the exponential growth of the system size and complexity. Modularizing the system, including for example FPGAs and exploiting their embedded transceivers for interface implementation, can certainly help in this direction.

In light of the aforementioned DAQ challenges, we propose to employ a cylindrical multiring mesh topology, whereby each photonic module acts like a sensor network element connected to its neighbors. In addition, a functional node acting as a snooper is included in every ring, whereby one of the snooper nodes, designated as a master, acts as bridge between the network and an external host PC. A high speed bidirectional serial communication link is used for inter node communication, due to its compactness and high rate. The snooper-to-PC communication is established by means of gigabit Ethernet connectivity.

One of the main advantages of this system architecture is that the processing is distributed over the entire network and that it does not need to be performed in full for all the events.

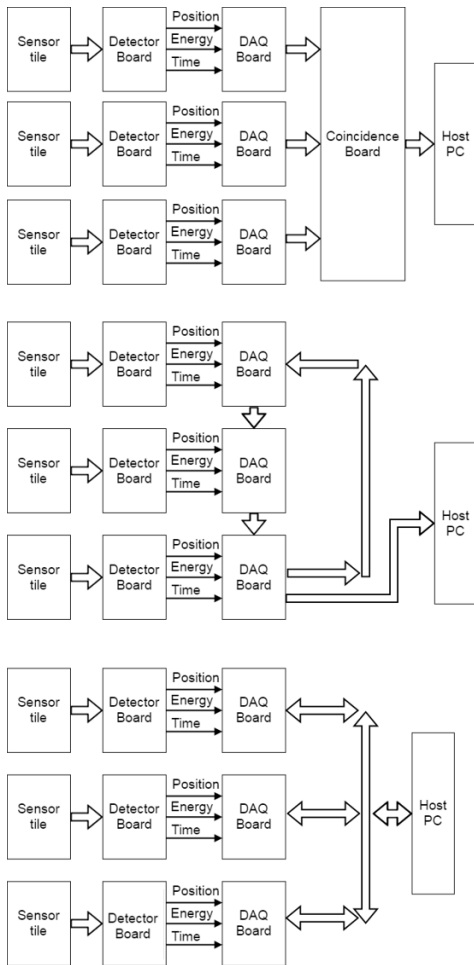


Fig. 4. Three typical PET DAQ system topologies: tree (top), daisy-chain (center), and bus (bottom). The structure of a DAQ system can be decomposed in four parts: multiplexing, event generation, DAQ board interconnection to detect coincidences, and transmission to a host computer. Every single part needs to be designed carefully in order not to degrade the performance of the overall system. Adapted from [20].

Indeed, if a coincidence is not detected, the corresponding singles packet can be simply discarded. This approach extends beyond previously proposed DAQ systems [27], [32], [33] by providing scalability and flexibility both in axial and in radial directions in terms of PET system construction.

III. SPADNET PHOTONIC MODULE

The SPADnet1 photosensor (Fig. 5) consists of an array of 16×8 fully digital CMOS pixels, each composed of mini-SiPMs; its characteristics are summarized in Table II [12]. Each pixel is capable of individual photon timestamping, using 12-bit, 64 ps time-to-digital converters (TDCs), and energy accumulation, whereas the sensor as a whole provides a real-time output of the total detected energy at up to 10 ns time intervals, including on-chip discrimination of gamma events. The system relies on a standard 130 nm, through-silicon vias (TSV)-enabled CMOS image process (STMicroelectronics). The SPADnet photonic module (Fig. 2) embeds 25 arrays of tightly abutted SPADnet photosensors on a single PCB to form a sensor tile [13]. TSVs allow to replace conventional wire

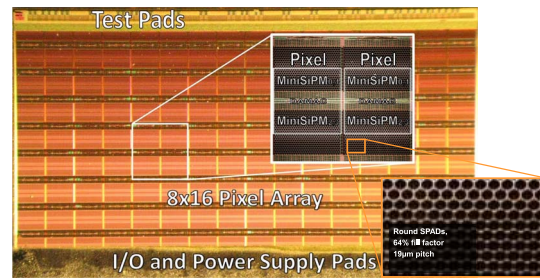


Fig. 5. SPADnet1 chip micrograph [12].

TABLE II
SPADNET1 SENSOR CHARACTERISTICS

Process technology	CMOS 1P4M 0.13 μm imaging with TSV
Array size	8×16 pixels
Array fill factor	42.6%
Chip size	$9.850 \times 5.452 \text{ mm}^2$
Maximum clock freq.	100 MHz
Output data rate @100 MHz	1.6 Gb/s
Supply voltage	2.5 V for I/O; 1.2 V for core
Chip power consumption @100 MHz	120 mW (dark)
Pixel pitch	$610.5 \mu\text{m} \times 571.2 \mu\text{m}^2$
Number of cells	720 (divided in 4 mini-SiPMs)
Cell diameter	$16.27/19.27 \mu\text{m}$ (active/total)
Number of TDCs per pixel	2 (1 always ready)
TDC resolution	64 ps
TDC range	12 bits

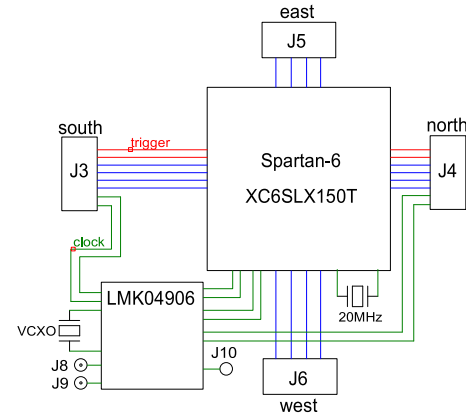


Fig. 6. Simplified schematic of the SPADnet FPGA modules. The core of the DPCU is a Spartan-6 FPGA (Xilinx). The connectors J3–J6 allow to network a node to its nearest neighbors in the SPADnet network. Connectors J8 and J9 allow to interface to external clock sources.

bonding with backside connection [13]. The sensor tile is then interfaced to an FPGA-based PCB on its back, schematically illustrated in Fig. 6, where the DPCU is designed to reside. The set of sensor tile and DPCU constitutes the SPADnet photonic module that functions as an autonomous sensing, computing, and CU.

The SPADnet1 photosensor has been followed up by the SPADnet2 design [17], based on the same technology and containing 16×16 pixels in an area of $9.85 \times 9.85 \text{ mm}^2$, with an enhanced pixel fill factor of 55% and 672 SPADs per pixel.

IV. SYSTEM LEVEL DESCRIPTION

We will describe in the following the main SPADnet DAQ phases and operating modes.

A. True Event Acquisition

Coincidence detection [21] is performed by correlating two gamma events which are detected within a short coincidence time window, typically in the nanosecond range. In order to tag them as a true event pair, the identified gamma events have to be within the field-of-view of the photonic modules which have detected them. Coincidence detection is generally performed using a dedicated processor because of its complexity, whereas the approach which we selected to perform the search operation is a distributed one. In this strategy, a search algorithm needs to be implemented in every node, in order to identify whether an incoming event packet is in coincidence with any of the events detected in that node. To avoid searching across the full event history, the network was designed so as to provide a low, well-defined packet latency (or low variance in packet latency). A low variance in packet latency will indeed ensure that the packets arrive at a specific node within a certain time period after having been detected. Thus, it is possible to update the event search space continuously by erasing the events that have been detected before the expected packet latency.

In practice, the true event pair is formed in two stages in the network: first, coincidence detection is performed, then, the detected coincidence events are paired to form a true event pair. The network is partitioned into two unidirectional channels (coincidence detection channel and data transfer channel) to aid its two-stage operation. The first stage [Fig. 7(a)], coincidence detection, aims to reduce packet latency by using a smaller data packet of 32 bits [Fig. 8(a)], which carries only the gamma event timing information and the node ID and is circulated in the coincidence channel. When the packet traverses a node's field-of-view, a copy of the packet is injected as well in the axial direction. The distributed coincidence detection units located in every DPCU perform coincidence detection by comparing the received information on the gamma event's timing with their own event history. Upon successful detection of the coincidence pairs, the event present in history is tagged accordingly; the tagged true event is then communicated to the data transfer channel.

In the second stage [Fig. 7(b)], coincidence pair formation, the data transfer channel, upon receiving the tagged event, transfers a coincidence detected packet (CDP) [Fig. 8(b)] to its pair or holds it until its pair arrives. The timing information of every packet is used by the network logic to arbitrate on which CDP has to be transferred and which one has to wait. This ensures equal utilization of the network resources across all nodes, randomizing the selection of the node that transfers its coincidence event packet. Once a pair has been formed (using the data transfer channel), it is sent to the external PC via the snoopers. It is important to note that the data transfer channel is designed to act in the opposite direction with respect to the coincidence channel (Fig. 7). This allows the nodes to monitor the status of their neighbors using node status packets.

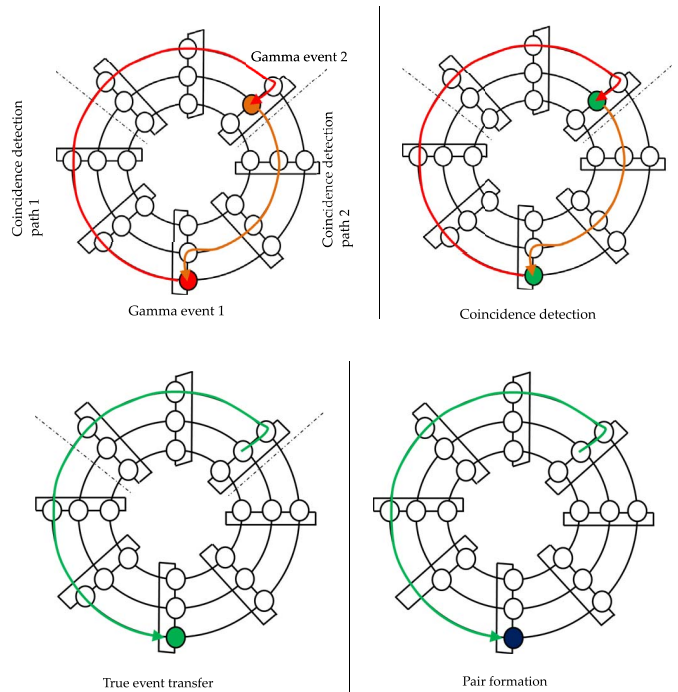


Fig. 7. Example of two-stage coincidence detection in a system composed of 3 rings of 8 photonic modules each (same system as in Fig. 3 but seen from an axial perspective). (a) Stage 1—Coincidence detection and data transfer, performed through the detection channel (clockwise): two nodes (red dots) detect a gamma event, transfer the information along the network to the respective field-of-view, and eventually detect a coincidence (green dots). (b) Stage 2—Coincidence pair formation, using the data transfer channel (counterclockwise): one node transfers the true event packet to the other, and a coincidence pair is formed, ready to be sent to the host PC via the snoopers.

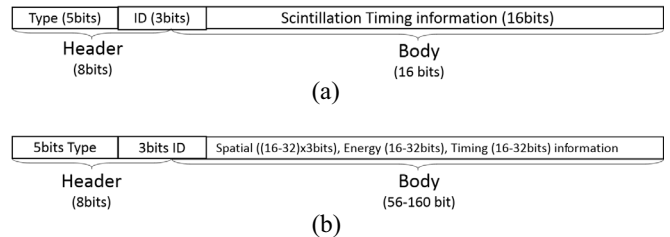


Fig. 8. (a) SPADnet singles event packet and (b) coincidence event packet.

B. Singles Acquisition

The network is also designed to perform singles acquisitions. A singles packet comprising the estimated values of energy, timing and scintillation coordinates of the gamma event is transferred to the PC via the snoopers. In this mode of operation, a data transfer channel is used for communication. Since the singles data rate can be higher than the Ethernet capacity, there could be a situation whereby the snoopers are not able to transfer all data packets to the host PC. In such situation, some events need to be dropped. This is, however, not deemed critical, given that singles acquisition is primarily included to ease PET system development as well as for more research oriented work, rather than to perform routine scans.

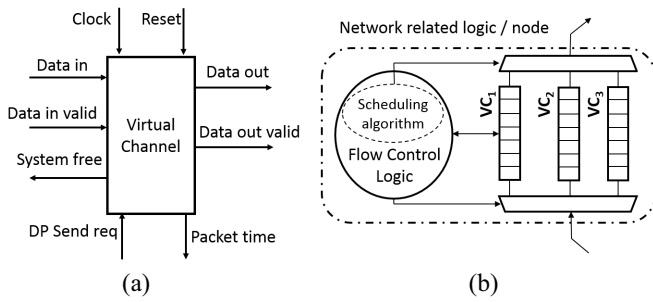


Fig. 9. (a) VC block diagram. (b) VC_1 : receives data packets from the coincidence packet generator; VC_2 and VC_3 : receive data packets from the IPC. The outputs of all three VCs are connected to a multiplexer that selects a VC output to be connected to the physical CU.

C. Raw Data Acquisition

The network also features the capability to acquire raw data from every photonic module individually for test and characterization purposes. Raw data, packetized into smaller data packets, are transferred to the snooper using the data transfer channel. The snooper collects the network packets, packetizes them into a larger raw data packet, and then transfers it to the computer for further processing.

D. Sensor/Node Configuration

A set of registers, with read and write access, are included in every node and controlled by the host PC via the network using the data transfer channel. These registers are used to facilitate the node configuration.

V. PACKET HANDLING TECHNIQUES

The network, partitioned in two data channels, handles six different packet types. The coincidence channel handles the node status and the coincidence packets. The data transfer channel handles the node status, configuration, coincidence detected, true event, singles and raw data packet. Since the two channels do not interfere with each other, they are treated as two standalone networks from the perspective of the routing, flow-control, and scheduling schemes.

A. Routing Strategy

For the network presented, static routing techniques are used for all data packet types. Thus, a packet entering the network will always follow a fixed path, irrespective of the network status. The static routing scheme was chosen for its simplicity. For the cylindrical mesh topology, the packets are routed first along the radial axis until they reach the axial ring of their destination node, and then they travel in the radial direction to their final destination. The used routing strategy is similar to the conventional X–Y routing scheme [34].

B. Scheduling Algorithm

The scheduling algorithm is used to decide if a packet needs to be transferred to the following node. It is designed to function in two stages: in the first stage, the packet type that needs to be transferred is selected. In the second stage, a specific

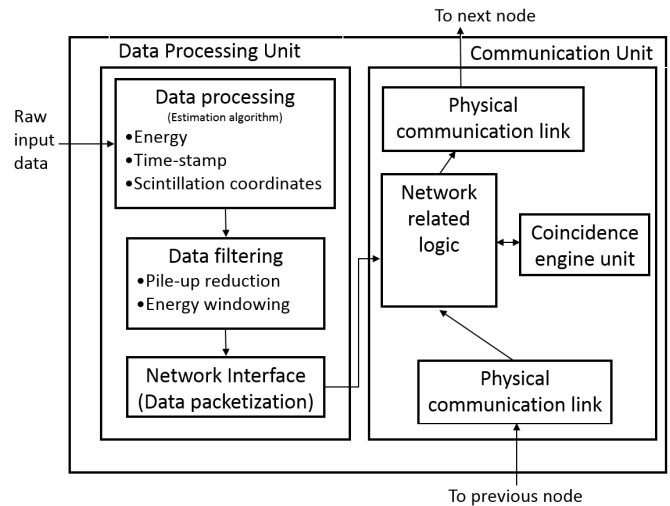


Fig. 10. Schematic of the DPCU hosted in each photonic module.

packet is chosen from a selected list. For the data channel which handles six different packet types, the packet priority is assigned in the following order.

- 1) Status packet.
- 2) Configuration packet (CP).
- 3) CDP.
- 4) True event, raw data, and singles packet.

The status packet is given the highest priority as it dictates the flow control of the other packet types. The CP is given the second highest priority to facilitate node control, even when the network is congested by other packet types. The CDP is given higher priority when compared to the DAQ packets (true event, raw data, and singles), to improve latency performance when forming a true event pair. Finally, true event, raw data, and singles packets are given the same priority because at any point in time the network will be configured for only one mode of operation.

For status, configuration, true event, raw data, and singles packets, a first-come-first-served policy is used in selecting a packet for transfer. For CDPs, an oldest-packet-first scheduling scheme is used. To aid the design a virtual channel (VC)-based approach is employed [35], [36]. A VC is a first-in first-out buffer, placed along the packet flow [Fig. 9(a)]. Every packet type is assigned at least to one VC. In the design, multiple VCs are assigned for the coincidence packet and for the CDP types, to reduce the design complexity in finding the oldest packet. The incoming packets are assigned to various VCs in a round-Robin fashion. In this configuration, the oldest packet is determined by comparing the timing of the packet present at the top of every VC.

C. Virtual Channel

A VC receives data packets from the downstream module and transfers them to the physical CU on receiving a command from the output packet controller (OPC). In addition, a VC continuously keeps updating the OPC with the timing information of the packet present on the top of the stack.

In the system, a three VC-based flow control technique was implemented [Fig. 9(b)]: the first, VC_1 , is used to store the

CP generated in the same node, whereas VC2 is used to store the CP received from other nodes, and VC3 is used to store the true event packet. The three VCs are identical in their functionality. The outputs of all the three VCs are connected to a multiplexer that selects a VC output to be connected to the physical CU.

D. Flow Control

The network flow control logic decides when to transfer a packet to the next node. For the proposed network, an adaptive flow control technique is used, which helps in achieving almost equal utilization of the buffer (VC) occupancy in every node. In the case of PET, it is indeed critical to ensure that all nodes maintain equal packet dropping probability during the entire DAQ. This ensures that at any given point in time, no gamma event is lost while its pair is being processed in the network. In the proposed network, DAQ packets are transferred to the next node only when the receiving node's network resource utilization level is less than that of the current node. This information is provided thanks to a periodical status packet transfer between neighboring nodes.

Based on this information, CPs and CDPs are transferred only if the receiving nodes' VC occupancy is less than 80%. When this is not the case, a status packet is broadcast to all nodes to stop acquiring any new gamma event. Under such a situation, the node that raised the stop flag needs to send a restart command to all nodes to start acquiring new gamma events. The restart command is sent only when the VC occupancy of the involved node decreases below a given threshold (typically around 10%). Furthermore, for intranode communication a protocol was devised to allow transfer of data packets only when the receiving module within a node is free to receive it. This protocol aids the internode data flow in shutting down the gamma event acquisition when the network is not able to accept any more events.

VI. DATA PROCESSING AND COMMUNICATION UNIT

The DPCU embeds three different modules, as depicted in Fig. 10, whereby the sensor control and the data processing units (DPUs) depend on the actual sensor used.

- 1) Sensor control unit.
- 2) DPU.
 - a) Energy estimation.
 - b) Timing estimation.
 - c) Scintillation coordinates estimation.
- 3) CU.
 - a) Coincidence channel unit.
 - b) Coincidence engine unit (CEU).
 - c) Data transfer channel unit.

A more detailed block diagram of the overall DPCU architecture is shown in Fig. 11.

A. Data Processing Unit

The availability of processing resources close to or possibly in the photosensing units, allows to carry out a number of preprocessing steps, data filtering, and/or low level data corrections already early on in the processing chain, thereby

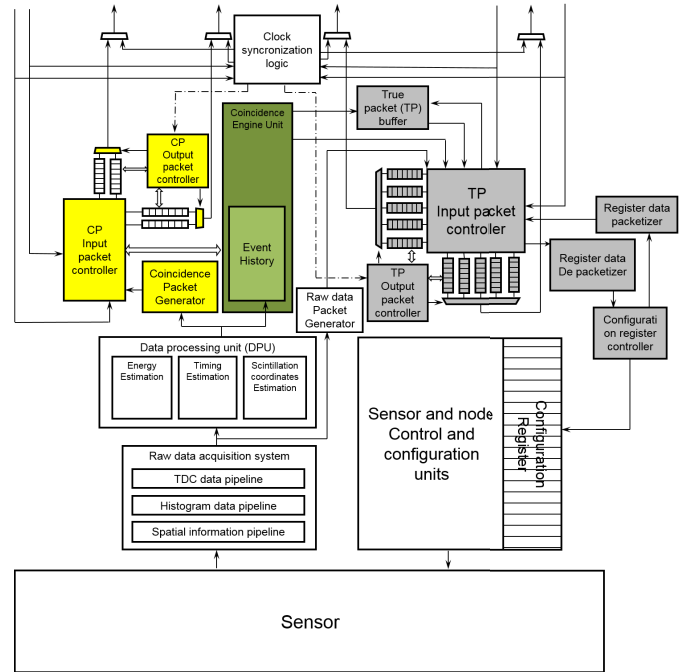


Fig. 11. DPCU architecture. CP: coincidence packet, TP: true packet.

helping in reducing the overall bandwidth requirements. This does come at the cost of architectural and implementation challenges to ensure smooth real-time operation.

In the case of SPADnet, the real-time output of the total detected energy can be easily processed by the FPGA for real-time energy windowing. Likewise, the spatial coordinates of the scintillation can be reconstructed from the pixel hit maps, e.g., via a modified center-of-gravity algorithm [37], thus enabling high probability crystal pin identification for SPADnet1 thanks to the sensors' high spatial granularity; the corresponding position estimation algorithms can then be tested at FPGA level [38] and, if promising, the same register-transfer level modules can be even implemented on the sensor chip periphery, due to the use of standard CMOS combined with low silicon real estate needs, as was indeed the case for SPADnet2 [17].

Furthermore, the use of digital SiPMs with on-chip multiple timing elements (typically TDCs) makes it possible to combine multiple timestamps, usually of the first N photons (with N up to several tens), in hardware-friendly time-of-arrival estimators [16]. Finally, additional (digital) functionalities can be added as well on-chip [17], such as frame buffers to decouple pixel array readout from data transmission, single/multiple threshold as well as post-event triggering, and on-chip event centroid calculation.

B. Communication Unit

1) *Coincidence Channel Unit*: The data packet generated by the DPU is communicated to the coincidence channel unit (Fig. 12). Here, the data packet is stored in the event history, while its copy is used to generate a coincidence packet. The coincidence packet received from the network and from the coincidence packet generator is arbitrated into various VCs

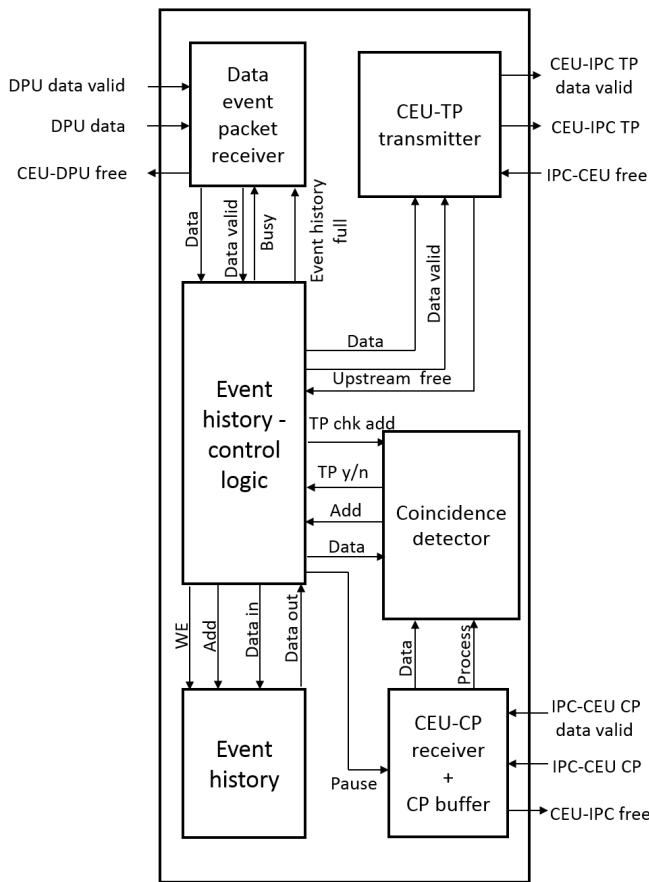


Fig. 12. CEU architecture (details of the green block in Fig. 11). The coincidence detection in the CEU is performed by comparing the data packets present in the event history with the CPs received from the IPC. CP: coincidence packet, TP: true packet.

adhering to the packet routing scheme in the input packet controller (IPC). The OPC, which houses the scheduling and flow control algorithms, selects a packet from the VC outputs and transfers it to the communication link.

2) *Coincidence Engine Unit*: In a multiring system the expected rate of CPs arriving at the CEU is much higher than that of a single-ring system, due to the fact that the coincidence detection needs to be performed across rings. In case of a clinical system, the expected packet rate could be as high as 125 million packets per second. To handle such a high packet rate, a high throughput design is required. Hence a novel CEU design, capable of performing coincidence detection in one clock cycle, was developed (Fig. 12). An 18kb RAM is used to store timing information, whereby each bit represents a time unit. Upon detection of a gamma event, the RAM bit addressed by its timing information is flagged using the RAM controller, and after a certain time (i.e., the packet latency) the flagged bit is nullified. Thus, the timing information of the entire gamma event history is stored in a user friendly and efficient format. The coincidence detection block (Fig. 12) performs then the actual coincidence detection and transfers the result to the results buffers. Finally, the true event identifier tags the gamma event packets as a singles or true or multiple event, depending on the coincidence detection results.

3) *Data Transfer Channel Unit*: The functionality of the input and the OPC are the same as described for the coincidence channel unit. The data transfer network, on the other side, is designed to operate on six different data packet types. For every packet type, at least one specifically dedicated VC is included in the design for both axial and radial communication. For this reason, the number of VCs present in the true packet network is higher than that in the coincidence packet network.

VII. MASTER SNOOPER MODULE

We already described the main functionalities of the snooper nodes, and will now dwell on some of the characteristics of the master snooper node.

A. Data Transfer Channel Unit

In case of the master snooper node (Fig. 13), the data transfer unit is enhanced with additional logic to divert the DAQ packets circulating in the network to an accumulator connected to the Ethernet controller. The diversion of the DAQ packets takes only place when the accumulator is free to receive them. If this is not the case, the true-event packets are recirculated into the network. This will, however, not impact the dead-time even when operating at high data rates, due on one hand to the intrinsically very low rate of true event packets, and on the other to the use of dedicated, independent VCs for each packet type. Moreover, the master snooper node also includes additional logic to handle the CPs being communicated to/from the PC through an interface module.

B. Ethernet-PC Communication Unit

The Ethernet-PC CU is a module that is included only in the master snooper node. This module is designed to transfer the data collected in the accumulator to the host PC, and to handle the configuration data transfer to/from it. The implemented design achieves a data communication rate, as high as 105MB/s, which is very close to the theoretical maximum for Gigabit Ethernet connectivity.

VIII. CLOCK DISTRIBUTION AND SYNCHRONIZATION

When designing time-of-flight PET systems, a key point is represented by their synchronization. Typically a hard-wired clock distribution network is used in modern PET scanners, e.g., in a “ring” or “star” topology. In the first case, the clock is daisy-chained from each node to the next, and locally “cleaned” using a module, such as TI’s LMK04906 (ultra low noise clock jitter cleaner/multiplier). This entails local clock offsets, which can be compensated with a precision of 25 ps by the LMK04906 module itself, provided that the user has measured the phase offset and programmed the module correspondingly. In the second case, a clock distribution board is required, such as the IDT ICS8534-01 low skew buffer, whose 1-to-22 fanout buffer introduces a maximum phase offset error of 100 ps. Depending on the quality of the coaxial cables, the phase needs to be measured and compensated.

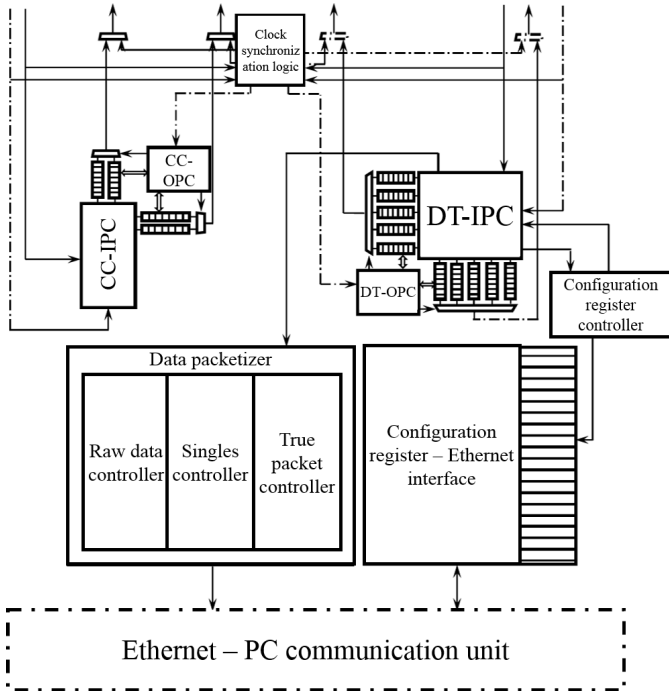


Fig. 13. Master snooper architecture. CC: coincidence channel, DT: data transfer, IPC/OPC: input/output packet controller.

All in all, hard-wired clock distribution solutions are not always feasible or practical in large sensor networks (e.g., the SPADnet system or future whole-body scanners). In addition, some applications demand a synchronization precision in the picosecond range and scalability. For these reasons, effort has been spent in designing network-based clock synchronization solutions, which are based on the exchange of message between sensor nodes, while recording transmission and arrival times.

A. Clock Synthesizer Model

To model the short-term relation between local time and global time, the first order model reported in [39] was used

$$t_i = \omega_i t + \phi_i \quad (1)$$

where t is the reference clock, t_i is the local time at i th node, ω_i the local clock skew, and ϕ_i the local clock offset. Converting the expression to global time, it can be written as follows:

$$t = \alpha_i t_i + \beta_i \quad (2)$$

where $\alpha = \omega_i^{-1}$, $\beta = -\omega_i^{-1}\phi_i$ are the parameters used to correct the local time in the i th node.

B. Two-Way Timestamp Exchanges

The two-way timestamp exchange represents the basis of most clock synchronization algorithms. The algorithm, as shown in Fig. 14, acts between pairs of nodes, whereby node 1 starts the timestamp exchange by sending a message to node 2. When the message, marked with the timestamp $T_{1,2}^{(1)}$, is received by node 2, the reception time is marked $R_{2,1}^{(1)}$.

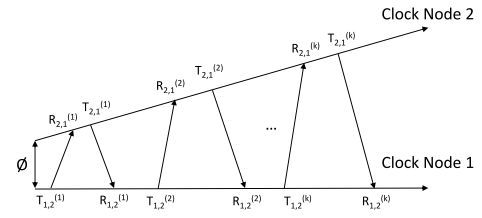


Fig. 14. Conventional two-way timestamp exchanges between a pair of nodes; ϕ represents the clock offset. After k two-way timestamp exchanges, the clock correction parameters can be estimated.

Rx/Tx (1 bit)	Coarse value (22 bits)	Fine value (9 bits)
---------------	------------------------	---------------------

Fig. 15. Two-way timestamp packet construction.

Then, node 2 replies and records the transmission and reception times, and so on. After k two-way timestamp exchanges, the clock correction parameters can be estimated.

C. Pairwise Least Square

The selected solution to synchronize the clocks is based on the pairwise least squares (PLS) scheme [39], which is well suitable for a hardware implementation, thanks to its low complexity compared to other approaches. This solution efficiently estimates the first order clock parameters. When the network contains $N > 2$ nodes, pairs of nodes are successively synchronized using one and the same node as reference. The algorithm has to be applied $(N-1)$ times in total, leading to the following set of equations:

$$\hat{\theta}_j = (A_{ji}^T A_{ji})^{-1} A_{ji}^T t_{ij} \quad (3)$$

with

$$A_{ji} = [t_{ji} \mathbf{1}_{2k} \mathbf{e}]^T \in \mathbb{R}^{2k \times 3} \quad (4)$$

$$\theta_j = [\alpha_j \beta_j \tau_{ij}]^T \in \mathbb{R}^{3 \times 1} \quad (5)$$

where t_{ij} , $t_{ji} \in \mathbb{R}^{2k \times 1}$ are the timestamps recorded at node i and j for all the k two-way communications, respectively, τ_{ij} is the pair-wise distance between node i and j , $\mathbf{e} = [-1, +1, -1, \dots, +1]^T \in \mathbb{R}^{2k \times 1}$, $\mathbf{1}_{2k} \in \mathbb{R}^{2k \times 1}$ is a vector of ones of length $2k$, and α_j , β_j are the correction parameters for the clock skew ω and clock offset ϕ of node j . For the presented algorithm, the most complex operation is the inversion of a 3×3 matrix, resulting in a simple hardware implementation.

D. Phase Synchronization Protocol

In a practical implementation, each ring of SPADnet modules features a reference node, and all nodes of that ring synchronize with it by performing a series of two-way timestamp exchanges and using the PLS algorithm to estimate their phase difference. The packets exchanged between nodes (Fig. 15) consist of 32 bits: the first eight LSBs contain the timestamp's fine value, whereas the following 22 bits contain its coarse value. The MSB indicates the type of packet (transmission or reception). In the case of multiring systems, the individual nodes in a given ring will synchronize, in a first

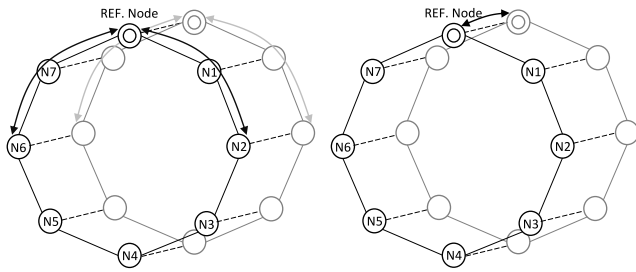


Fig. 16. Stages of the proposed clock synchronization protocol for a multiring system. (left) The individual nodes in a given ring will first synchronize to a reference node in that ring. (right) The reference nodes will then synchronize with each other.

stage, to a reference node in that ring (Fig. 16), in parallel over all the rings. Once done, the reference nodes will synchronize, in a second stage, with each other. The procedure takes a time $(R/2 + N/2) \cdot t$, where N is the number of nodes, R the number of rings, and t the average time needed to synchronize a pair of nodes.

IX. EXPERIMENTS AND RESULTS

A. Simulation Results

Two simulators, one emulating the coincidence detection channel and the other the data transfer channel, were designed and optimized for speed and memory usage [40], [41]. The former was designed to monitor the transfer of packets from one node to the next, employing a discrete time simulation technique, and to evaluate packet latency and its variation with time. The latter was aimed at studying the node buffer VC occupancy as well as bandwidth requirements, employing the rate-of-change concept; this led to substantial speed-ups, at the price of a higher implementation complexity. The simulators allowed the study of various packet flow control and routing algorithms.

The simulators were then modified so as to incorporate measurements from the single ring network prototype model, detailed below, to ensure realistic scenarios in terms of the packet latency and data bandwidth variations introduced by the communication protocol, and coupled with GATE to emulate a real PET system testbed. This allowed us to carry out a scalability study, testing the proposed concept for various PET configurations (preclinical, brain, and human) [15], [42]. Equal distributions of gamma events, following Poisson statistics, were injected in each node. Fig. 17 shows the simulation results for preclinical dimensions (5 rings of 10 photonic modules each). For a node-to-node communication speed of 1 Gb/s (2 Gb/s), the packet latency diverges when an incident gamma event rate of around 1.6 (3.2) million events per second is reached, due to bandwidth saturation. Similar studies were carried out with clinical (5 rings of 50 photonic modules each) and brain PET (5 rings of 25 photonic modules each) configurations. In the latter case, which is particularly demanding, bandwidth saturation is reached earlier (around 0.4 and 0.6 million events per second at 1 and 2 Gb/s, respectively) due to the wider field-of-view requirements. This calls for a data

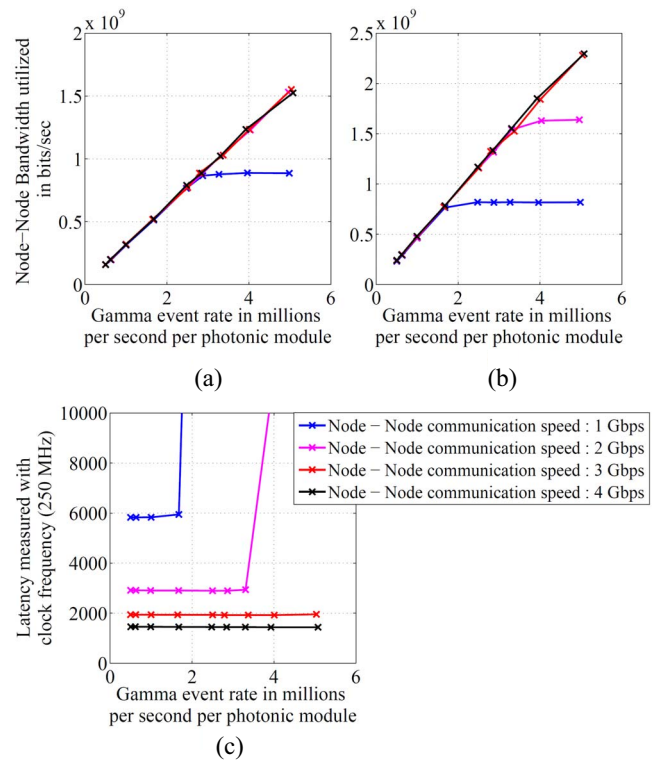


Fig. 17. Preclinical PET coincidence simulation results (5 rings of 10 photonic modules each). (a) Radial ring node-node bandwidth utilization, (b) axial ring node-node bandwidth utilization, and (c) maximum coincidence packet latency.

communication rate of around 3 Gb/s to meet the envisaged specifications (maximum singles rates of 5 Mcps, 500 kcps, and 500 kcps for preclinical, clinical, and brain PET, respectively), which has been experimentally demonstrated to be feasible.

Finally, transient simulations were carried out by artificially injecting a large number of packets in a given node when the network had reached a steady-state status. The simulation results showed that the network was capable of adjusting to the imbalance, at a rate which depended on the data communication bandwidth.

B. Single Ring System

The network built to validate initially a single ring configuration comprised ten DPCU nodes and one master snooper node. The DPCU was ported to the custom designed Spartan-6-based FPGA board shown in Fig. 2, right, and Fig. 6, whereas the snooper was implemented in a Virtex-6-based board (ML605, Xilinx Inc.). In this setup, a 2 Gb/s serial communication link was established for internode communication using the GTP (for Spartan-6) and GTX (for Virtex-6) transceivers. The Aurora link-layer protocol was selected to provide a 32 bit interface, along with an 8/10 bit encoding scheme for forward error corrections. To facilitate the exchange of packets of varying sizes via the Aurora interface, a packetizer and a de-packetizer were designed as an interface between the Aurora interface and the network core. In the implemented design both the DPCU and the snooper,

where operated at 62.5 MHz to match the data communication rate and the Aurora data interface.

Using this setup, all node configuration and DAQ modes were tested successfully, together with coincidence detection, using artificially injected data packets generated within every node (no sensor tiles were employed). The data packets were generated periodically in all/selected nodes every few clock cycles, enabling to test the network operation and to obtain the real time parameters for simulations, as mentioned above.

C. Multiring System

The multiring system was tested using another prototype model based on the Xilinx ML605 development kit, which embeds a Virtex-6 FPGA. In this configuration, each ML605 board houses two nodes from adjacent rings, emulating a network of 2 rings of 5 nodes each. In addition, to facilitate the synchronization of the timestamp generation for gamma events across various nodes, a clock was distributed to all the nodes from a centralized source based on the LMK00301 board (Texas Instruments). Using such a setup, the network was successfully operated at an internode communication speed of 3.2 Gb/s, demonstrating multiring operation.

D. Clock Synchronization and Phase Estimator Hardware Implementation

Multiple solutions for synchronization in wireless sensor networks are readily available in the literature. We initially focused on a class of purely network-based algorithms [39], [43], which did, however, not enable to reach a sufficient precision level by themselves. Indeed, although the tested network-based clock synchronization algorithms did reach their theoretical limits, the overall performance was still bound by the “long term” (>1 s) stability of the local clock synthesizers. In the case of SPADnet, the best stability which could be reached was measured to be 22 ns, which is obviously insufficient [19]. Alternative clock synthesizer solutions do exist, but are impractical.

We therefore focused on a hybrid scheme, whereby a hardwired ring clock distribution network is combined with a network-based clock offset estimator. In this configuration, only clock phase offset corrections are needed at each node. (All modules run on the same clock; the long-term clock stability is therefore ensured.) Thus, the clock correction algorithm can be used to estimate the phases, and by means of a clock jitter cleaner module, such as the aforementioned TI LMK04906, the phase can be physically corrected [19], [44]. This combination allows to preserve scalability while maintaining high precision. The presented hybrid solution enables the option to regularly monitor the phase offsets, allowing to compensate for temperature changes, aging of the electronics and other sources that might affect the phase offsets in the system.

The phase estimator is based on the PLS algorithm described in Section VIII, which finds an estimate for the clock parameters between pairs of nodes. The algorithm was

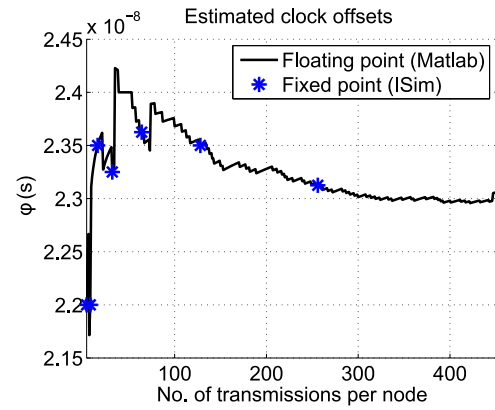


Fig. 18. Verification of the fixed point phase estimator implementation (corresponding to the FPGA implementation), comparing ISim, and MATLAB simulation results (offline implementation). The stars correspond to some of the discrete solutions of the fixed point implementation.

optimized and implemented in Spartan-6 FPGAs, wherein a delayline FPGA TDC was deployed to enable precise timestamping resolution, and therefore high resolution phase estimation.

Given the absence of clock drift in this approach, it is actually unnecessary to estimate the α_j (clock skew) parameters (2). Setting $\alpha_j = 1$, the expression (3) is thus simplified and all matrix elements A'_{ji} reduce to constants. In terms of operations, for a given number $k = 2^i$ of timestamp exchanges, solving the least squares problem reduces to only $2k$ logical shift, $3k$ additions, and $2k$ subtractions [19], which can be easily implemented in hardware. In terms of efficient hardware implementation, the parameters need to be converted to fixed point. The smallest value which needs to be represented is $1/2k$, which requires ($k = 8192$) at least 13 fractional bits in two’s complement fixed point format. The largest number depends on the communication latency between two nodes, expressed in clock ticks, and amounts to about 100 in the presented system (for a 100 MHz timestamping clock and a 1.25 Gb/s line rate). We have therefore implemented phase estimator registers of maximum 48 bits, noting that 25 integer bits would allow to handle a network of more than 500 nodes. Details of the actual implementation are available in [19].

1) *Phase Estimator Simulation Results:* The phase estimator has been simulated first with ISim simulator v.14.7 (fixed point), employing real timestamps collected by two-way timestamp exchanges using a two node network. The results are compared in Fig. 18 with a floating point PLS MATLAB simulation. The comparison shows no difference between the two simulations, hence enough fraction bits are used.

2) *Phase Estimator Experimental Results:* The final resolution of the phase estimator does ultimately depend on the timestamping precision. We used a single delay line FPGA TDC, along the lines of what detailed in [45], achieving a resolution of 18.4 ps. Because of the structure of the FPGA, the TDC has large linearity deviations, mostly due to the gaps in the carry4 elements. The resulting integral (INL) and differential (DNL) nonlinearity are shown in Fig. 19. They will increase the noise on the timestamping of the two-way messages, which will in turn affect the performance of

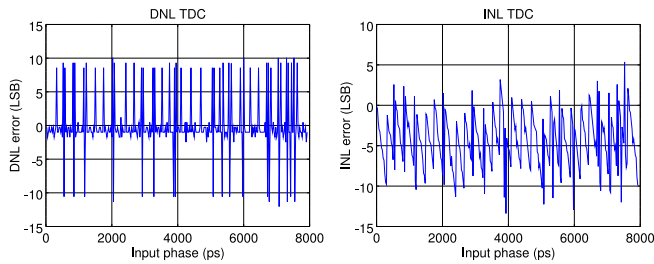


Fig. 19. FPGA TDC uncalibrated DNL and INL nonlinearities.

the estimation. However, this noise can be compensated by careful calibration and increasing the number of timestamp exchanges.

The estimator was first tested over 2000 runs with a fixed phase difference between two adjacent nodes (Fig. 20). The estimation was performed using 4096 two-way timestamp exchanges and a fixed observation window of 328 ms, resulting in a standard deviation of 124 ps. We then evaluated the influence of the observation window, varying it from 0.8 to 328 ms while exchanging 2048 timestamps. As expected, the estimator did reach the lowest value (182 ps) for the longest observation window. The smallest phase offset that can be estimated was measured to be 18.4 ps, matching the time resolution of the TDC (Fig. 21).

The proposed synchronization scheme was then tested in an 8 node system, using the actual FPGA boards of the SPADnet modules, which allows to perform the synchronization in real time, on the very same FPGA which is taking care of data preprocessing and communication to neighboring nodes. A worst-case stability of 157 ps (standard deviation) was reached for $k = 4096$ two-way message exchanges between two adjacent nodes (Fig. 22). Simulations indicate that an improvement by at least a factor of 5 is easily attainable by further increasing the number of exchanged messages, within a still acceptable measurement time, even for a large system (using the protocol detailed at the end of Section VIII), as well as through better TDC calibration.

X. CONCLUSION

Digital photonic modules designed using digital SiPMs in conjunction with DPCUs provide *in situ* DAQ, processing, and communication capability. In this paper, we demonstrate that localized processing and CUs, such as DPCUs, together with a sensor network-based design, make it possible to handle the data preprocessing and acquisition challenges associated with digital SiPMs, or equally, analog SiPMs with local digitization. The scalability and flexibility of the proposed approach were first verified by extensively simulating various PET system configurations. The detailed study carried out at various abstraction levels has shown that the presented approach could indeed be used for multiring-based preclinical, clinical, and brain PET. The downside of the proposed architecture is represented by the need, at module level, for additional connectivity and processing power.

In addition, the presented hybrid clock synchronization solution uses a ring hard-wired clock distribution network in

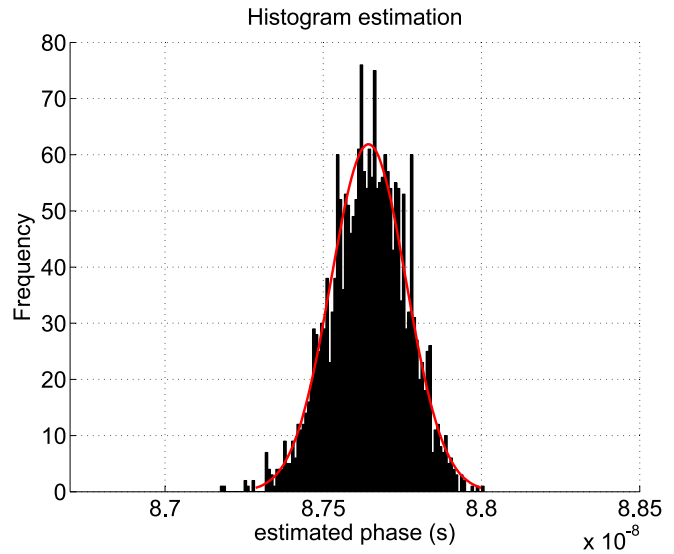


Fig. 20. Two thousand executions of the hardware implemented phase estimation between two adjacent nodes and a fixed phase difference, 4096 two-way timestamp exchanges per execution, 328 ms observation window. The resulting standard deviation was of 124 ps.

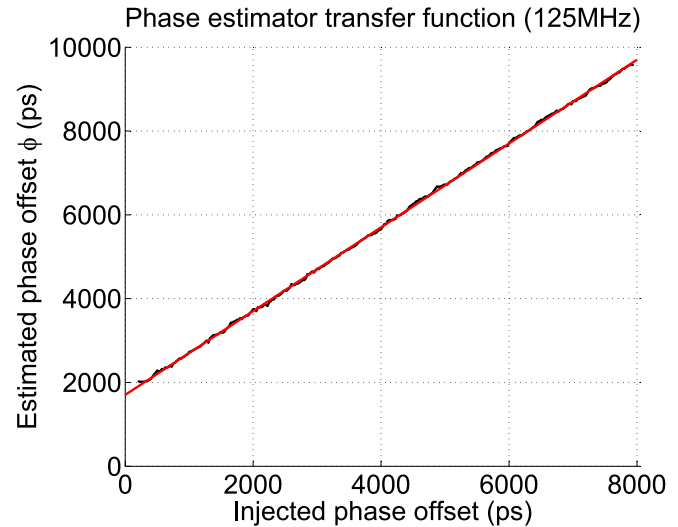


Fig. 21. Phase estimator transfer function applied to two adjacent nodes, showing the average of 25 estimations per injected phase, with $k = 4096$ two-way timestamp exchanges over an observation window of 328 ms.

combination with a network-based phase estimator. The scalability is ensured maintaining high precision and reducing the user interventions. The phase estimator, based on a least squares estimation algorithm, was optimized and implemented in Spartan-6 FPGAs to run in real time. The presented solution reaches a resolution of 18.4 ps also thanks to a high-resolution delay line TDC. The worst-case stability of the phase estimator is at the 160 ps level (standard deviation) for an eight nodes system, with substantial room for improvement (at least $5\times$ according to simulations) by increasing the sample size parameter of the estimator, as well as through better TDC calibration. The presented hybrid solution enables the option to regularly

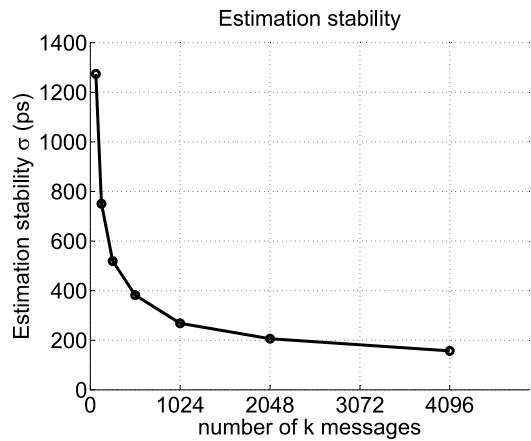


Fig. 22. Eight node phase estimation stability while increasing message exchanges.

monitor the phase offsets, which allows to compensate for temperature changes, aging of the electronics and other sources that might affect the phase offsets in the system.

ACKNOWLEDGMENT

The authors would like to thank the Mediso staff for precious support throughout the whole SPADnet project. They also would like to thank Xilinx Inc. for FPGA donation.

REFERENCES

- [1] S. R. Cherry and M. Dahlbom, *PET: Physics, Instrumentation, and Scanners*. New York, NY, USA: Springer, 2006.
- [2] M. Pizzichemi, "Positron emission tomography: State of the art and future developments," *J. Instrum.*, vol. 11, Aug. 2016, Art. no. C08004.
- [3] S. Vandenberghe and P. K. Marsden, "PET-MRI: A review of challenges and solutions in the development of integrated multimodality imaging," *Phys. Med. Biol.*, vol. 60, no. 4, p. R115, May 2015.
- [4] M. S. Judenhofer *et al.*, "Simultaneous PET-MRI: A new approach for functional and morphological imaging," *Nat. Med.*, vol. 14, no. 4, pp. 459–465, 2008.
- [5] H. Zaidi and A. D. Guerra, "An outlook on future design of hybrid PET/MRI systems," *Med. Phys.*, vol. 38, no. 10, pp. 5667–5689, 2011.
- [6] B. Weissler *et al.*, "A digital preclinical PET/MRI insert and initial results," *IEEE Trans. Med. Imag.*, vol. 34, no. 11, pp. 2258–2270, Nov. 2015.
- [7] 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.
- [8] D. F. Newport, H. M. Dent, M. E. Casey, and D. W. Bouldin, "Coincidence detection and selection in positron emission tomography using VLSI," *IEEE Trans. Nucl. Sci.*, vol. 36, no. 1, pp. 1052–1055, Feb. 1989.
- [9] M.-A. Tetrault *et al.*, "Real time coincidence detection system for digital high resolution APD-based animal PET scanner," in *Proc. Nucl. Sci. Symp. Conf. Rec.*, vol. 5, Oct. 2005, pp. 2849–2853.
- [10] D. Schug, B. Weissler, P. Gebhardt, and V. Schulz, "Crystal delay and time walk correction methods for coincidence resolving time improvements of a digital-silicon-photomultiplier-based PET/MRI insert," *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 1, no. 2, pp. 178–190, Mar. 2017.
- [11] B. Goldschmidt *et al.*, "Software-based real-time acquisition and processing of PET detector raw data," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 2, pp. 316–327, Feb. 2016.
- [12] L. H. C. Braga *et al.*, "A fully digital 8×16 SiPM array for PET applications with per-pixel TDCs and real-time energy output," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 301–314, Jan. 2014.
- [13] C. Bruschini *et al.*, "SPADnet: Embedded coincidence in a smart sensor network for PET applications," *Nucl. Instrum. Methods Phys. Res. A Accelerators Spectrometers Detectors Assoc. Equip.*, vol. 734, pp. 122–126, Jan. 2014.
- [14] D. R. Schaart, E. Charbon, T. Frach, and V. Schulz, "Advances in digital SiPMs and their application in biomedical imaging," *Nucl. Instrum. Methods Phys. Res. A Accelerators Spectrometers Detectors Assoc. Equip.*, vol. 809, pp. 31–52, Feb. 2016.
- [15] C. Veerappan, C. Bruschini, and E. Charbon, "Sensor network architecture for a fully digital and scalable SPAD based PET system," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Anaheim, CA, USA, 2012, pp. 1115–1118.
- [16] L. Gasparini, D. Mariz, R. Passerone, and D. Stoppa, "A comparison of FPGA architectures to extract gamma arrival times from multiplexed digital SiPM PET detectors," in *Proc. Nucl. Sci. Symp. Conf. Rec.*, Oct. 2015, pp. 1–3.
- [17] E. Gros-Daillon *et al.*, "First characterization of the SPADnet-II sensor: A smart digital silicon photomultiplier for ToF-PET applications," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2016.
- [18] E. Charbon *et al.*, "SPADnet: A fully digital, networked approach to MRI compatible PET systems based on deep-submicron CMOS technology," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2013, pp. 1–5.
- [19] M. Bijwaard, "Scalable network based clock synchronization for digital PET system," M.S. thesis, Circuits Syst., Delft Univ. Technol., Delft, The Netherlands, 2015.
- [20] E. Kim, K. J. Hong, J. Y. Yeom, P. D. Olcott, and C. S. Levin, "Trends of data path topologies for data acquisition systems in positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 60, no. 5, pp. 3746–3757, Oct. 2013.
- [21] P. E. Valk, D. L. Bailey, D. W. Townsend, and M. N. Maisey, Eds., *Positron Emission Tomography: Basic Science and Clinical Practice. Part IV: Oncology*. London, U.K.: Springer, 2003, pp. 481–688.
- [22] Y. C. Tai *et al.*, "MicroPET II: Design, development and initial performance of an improved microPET scanner for small-animal imaging," *Phys. Med. Biol.*, vol. 48, no. 11, pp. 1519–1537, Jun. 2003.
- [23] H. W. de Jong *et al.*, "Performance evaluation of the ECAT HRRT: An LSO-LYSO double layer high resolution, high sensitivity scanner," *Phys. Med. Biol.*, vol. 52, no. 5, pp. 1505–1526, 2007.
- [24] R. Fontaine *et al.*, "The hardware and signal processing architecture of LabPETTM, a small animal APD-based digital PET scanner," *IEEE Trans. Nucl. Sci.*, vol. 56, no. 1, pp. 3–9, Feb. 2009.
- [25] W. W. Moses *et al.*, "OpenPET: A flexible electronics system for radio-tracer imaging," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2009, pp. 3491–3495.
- [26] M. Morrocchi *et al.*, "Timing performances of a data acquisition system for time of flight PET," *Nucl. Instrum. Methods Phys. Res. A Accelerators Spectrometers Detectors Assoc. Equip.*, vol. 695, 2012, pp. 210–212.
- [27] D. F. Newport *et al.*, "Quicksilver: A flexible, extensible, and high-speed architecture for multimodality imaging," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, vol. 4, Oct. 2006, pp. 2333–2334.
- [28] E. Kim, P. Olcott, and C. Levin, "Optical network-based PET DAQ system: One fiber optical connection," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2010, pp. 2020–2025.
- [29] A. Douraghy, F. R. Rannou, R. W. Silverman, and A. F. Chatzioannou, "FPGA electronics for OPET: A dual-modality optical and positron emission tomograph," *IEEE Trans. Nucl. Sci.*, vol. 55, no. 5, pp. 2541–2545, Oct. 2008.
- [30] G. Sportelli *et al.*, "Reprogrammable acquisition architecture for dedicated positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 58, no. 3, pp. 695–702, Jun. 2011.
- [31] E. Fysikopoulos *et al.*, "Fully digital FPGA-based data acquisition system for dual head PET detectors," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 5, pp. 2764–2770, Oct. 2014.
- [32] B. E. Atkins *et al.*, "A data acquisition, event processing and coincidence determination module for a distributed parallel processing architecture for PET and SPECT imaging," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, vol. 4, Oct. 2006, pp. 2439–2442.
- [33] E. Kim, P. Olcott, and C. Levin, "A new data path design for a PET data acquisition system: A packet based approach," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Oct. 2011, pp. 3871–3873.
- [34] S. D. Chawade, M. A. Gaikwad, and R. M. Patrikar, "Review of XY routing algorithm for network-on-chip architecture," *Int. J. Comput. Appl.*, vol. 43, no. 21, pp. 20–23, Apr. 2012.
- [35] W. J. Dally, "Virtual-channel flow control," *IEEE Trans. Parallel Distrib. Syst.*, vol. 3, no. 2, pp. 194–205, Mar. 1992.

- [36] W. Dally and C. Seitz, "Deadlock-free message routing in multiprocessor interconnection networks," *IEEE Trans. Comput.*, vol. C-36, no. 5, pp. 547–553, May 1987.
- [37] B. Játékos, E. Lőrincz, F. Ujhelyi, and G. Erdei, "High probability crystal pin identification in scintillator matrix-based PET detector with a prototype digital SiPM," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Oct. 2013, pp. 1–4.
- [38] A. Kufcsák, "Maximum likelihood based determination of the position of annihilations in PET detectors," Faculty Elect. Eng. Informat., Dept. Meas. Inf. Syst., Budapest Univ. Technol. Econ., Budapest, Hungary, TDK Rep., 2014.
- [39] R. T. Rajan and A.-J. van der Veen, "Joint ranging and clock synchronization for a wireless network," in *Proc. Comput. Adv. Multi Sensor Adapt. Process. (CAMSAP)*, 2011, pp. 297–300.
- [40] C. Veerappan, E. Venialgo, C. Bruschini, and E. Charbon, "SPADnet network modeling, simulation and emulation," in *Proc. 19th IEEE-NPSS Real Time Conf.*, May 2014, pp. 1–2.
- [41] C. Veerappan, C. Bruschini, and E. Charbon, "Distributed coincidence detection for multi-ring based pet systems," in *Proc. 19th IEEE-NPSS Real Time Conf.*, May 2014, pp. 1–2.
- [42] C. Veerappan, "Single-photon avalanche diodes for cancer diagnosis," Ph.D. dissertation, Quant. Technol., Delft Univ. Technol., Delft, The Netherlands, 2015.
- [43] D. W. Allan, "Time and frequency (time-domain) characterization, estimation, and prediction of precision clocks and oscillators," *IEEE Trans. Ultrason., Ferroelect., Freq. Control*, vol. 34, no. 6, pp. 647–654, Nov. 2011.
- [44] M. Bijwaard, C. Veerappan, C. Bruschini, and E. Charbon, "Fundamentals of a scalable network in SPADnet-based PET systems," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, 2015, pp. 1–3.
- [45] H. Homulle, F. Regazzoni, and E. Charbon, "200 MS/s ADC implemented in a FPGA employing TDCs," in *Proc. ACM/SIGDA Int. Symp. Field Program. Gate Arrays (FPGA)*, Monterey, CA, USA, 2015, pp. 228–235.