# A Building Block for Nuclear Medicine Imaging Systems Data Acquisition

Tom K. Lewellen, *Fellow, IEEE*, Don DeWitt, Robert S. Miyaoka, *Senior Member, IEEE*, and Scott Hauck, *Senior Member, IEEE*

***Abstract*—Adapting acquisition electronics to new detector designs has often led to complications and compromises. As we developed depth-of-interaction detector designs based on both discrete crystal arrays (dMiCE) and monolithic crystals (cMiCE) concepts, we found that our previous electronics design was inadequate to the task and launched a design effort we have termed our Phase II electronics. The system is based on a basic card design (the Phase II board) that has a large field programmable gate array (FPGA) with sufficient static RAM to support a variety of pulse processing algorithms our group has developed–including timing estimation, pulse integration with pileup correction, and statistical estimation of the event location in the detector. Here we report on the initial development and testing of the Phase II digital board as a basic building block for data acquisition systems.**

***Index Terms*—Biomedical applications of radiation, nuclear electronics, nuclear medicine.**

## I. Introduction

DEVELOPING new detector designs for PET and SPECT imaging systems often leads to challenges in finding acquisition electronics that are optimized for the application. This is a problem we have had several times as we worked on new detector concepts. As a result, we launched a design effort for an electronics system that can be adapted to our many needs - what we term our Phase II electronics board [1].

Figs. 1–4 depict examples of the four major detector designs currently being utilized in our laboratory. The original cMiCE [2] module is based on a multi-anode photomultiplier tube (PMT) coupled to a monolithic crystal. It requires 64 channels of readout along with a timing pickoff channel. In addition, the module is designed to be used with statistical based position algorithms to estimate the x, y, and z coordinates of the event. Our designs call for those algorithms to be implemented at the detector module level to further reduce the amount of data required to be sent to the host computer. Those algorithms have been developed and implemented in FPGAs [3].

Manuscript received May 06, 2013; revised September 05, 2013; accepted October 16, 2013. Date of publication January 29, 2014; date of current version February 06, 2014. This work was supported in part by NIH NIBIB EB001563, NIH NIBIB EB002117, NIH NCI CA136569, DOE DE-FG02-05ER15709, Altera Corporation, and Zecotek Photonics.

T. K. Lewellen, D. DeWitt, and R. S. Miyaoka are with the University of Washington Department of Radiology, Seattle, WA USA (e-mail: tkldog@u.washington.edu; dewitdq@u.washington.edu; rmiyaoka@u.washington.edu).

S. Hauck is with the University of Washington Department of Electrical Engineering, Seattle, WA USA (e-mail: hauck@ee.washington.edu).

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/TNS.2013.2295037

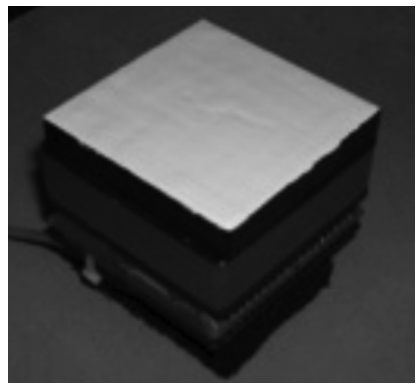

Fig. 1. cMiCE detector module requires 64 readout channels plus a timing pickoff channel.

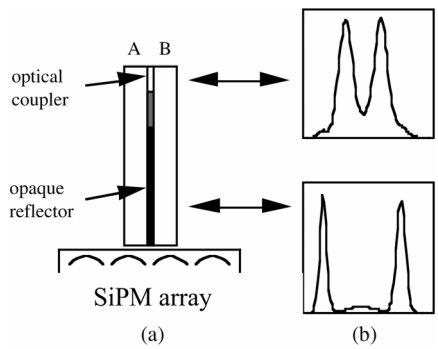


Fig. 2. dMiCE detector concept. (a) DOI detector unit. (b) PMT ratio plots [A/(A+B)]. A significant amount of light is shared when an event is detected near the entrance face of the detector unit. Less sharing occurs for interactions near the PMT interface.

The dMiCE [4] approach (using controlled light sharing between crystal pairs) is best realized with individual readouts for each crystal (Fig. 2). The current module designs are based on arrays of $10 \times 10$ to $20 \times 20$, and will use silicon photomultiplier (SiPM) devices for the light collection. To reduce the number of readout channels, a row/column/diagonal summing ASIC [5] is under development that would reduce the number of channels to be digitized for a $20 \times 20$ array to 60 plus a fast timing pick off channel that is a sum of all the array elements.

A variant on the cMiCE approach is shown in Fig. 3. For this design (termed SES) the sensor array is put on the entrance surface of the monolithic crystal, closer to where the majority of first interactions occur. The result is in an improvement in the estimation of the position of the event. The initial versions of this design use an $8 \times 8$ array of SiPMs mounted on a flex circuit to allow routing of the signals to the backside of the de-

Fig. 3. SES cMiCE detector module (sensors on front surface of the monolithic crystal) with flex circuit for routing signals to the backside of the module.

tector module where the connectors and impedance matching networks are located. As with the original cMiCE module, 64 read out channels and one timing channel will be utilized. However, future versions may go to a larger SiPM array to obtain better sampling of the light spread functions, in which case we plan to use the row/column ASIC [5] to reduce the number of signals to be processed.

Our newest detector design is shown in Fig. 4 where segmented crystals (tapered) in the transaxial direction and continuous decoding in the axial direction are utilized. The current module design requires 16 channels of readout, again with fast timing pickoff.

Before starting the Phase II design effort, we investigated other projects such as the OpenPet initiative [6]. At the time we began this project the alternatives did not offer the FPGA capacity to support the estimation algorithm requirements for the cMiCE and SES detector designs. Thus we began the Phase II board project focused on performing all of the pulse processing (timing, integration, baseline correction, pileup correction, and position estimation) in a large FPGA.

## II. Phase II System Design and Testing

### A. System Design

Our basic approach was to minimize analog processing and implement as much pulse processing as feasible into an FPGA. To allow easy adaptation to different topologies and detectors, we designed the system around a digital processing board (the Phase II board) that is then connected to the detectors with a separate analog adapter board. The analog board provides power and any needed local control to the detector and sends differential signals to the Phase II board.

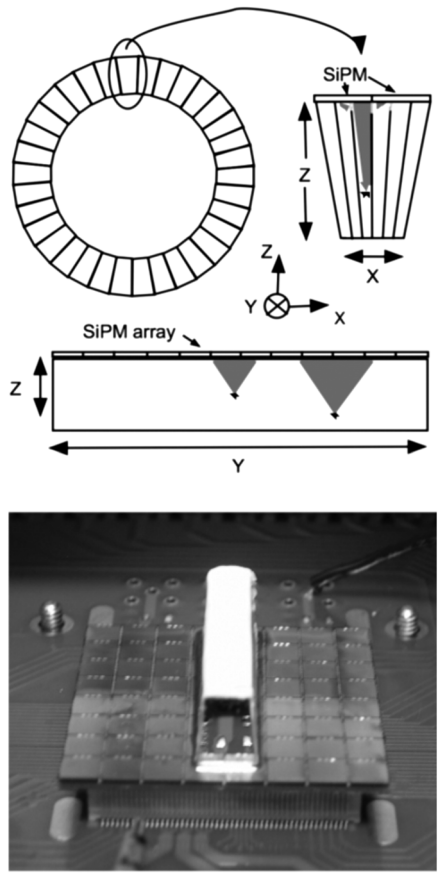


Fig. 4. TSC detector module prototype (2x8) requiring 16 channels of data per detector.

Given our limited resources, we also decided to put our effort into designing a single board that can take on many different roles in a data acquisition system depending on which components are installed and the FPGA code utilized. During the design phase, one engineer did both the schematic design and board layout in parallel. In that way, we were able to optimize the design to maintain needed bandwidth of signal traces, minimize any skewing issues for parallel data paths, and keep the number of layers down to twelve. To address heating issues there are large ground planes to “pipe” heat to the mounting hardware. The design is built around a family of Altera FPGAs that share the same footprint so that different cost/performance targets can be met by choosing which FPGA to mount on the board.

Fig. 5 is a block diagram of the Phase II board. Fig. 6 depicts the top of the Phase II board, as well as an image of the trace layout of the card. The current printed circuit board (PCB) is 216 mm long and 104 mm wide and consists of 12 layers. The schematic layout (in particular the pin assignments for the FPGA) was done iteratively with the PCB layout to reduce the number of layers. Such a process allows for minimization of path lengths and reduces the number of required layers in part by reducing the number of traces that need to cross over each other. Part of the process is a measurement of the trace lengths and calculation of both time skew and bandwidth capacity (which

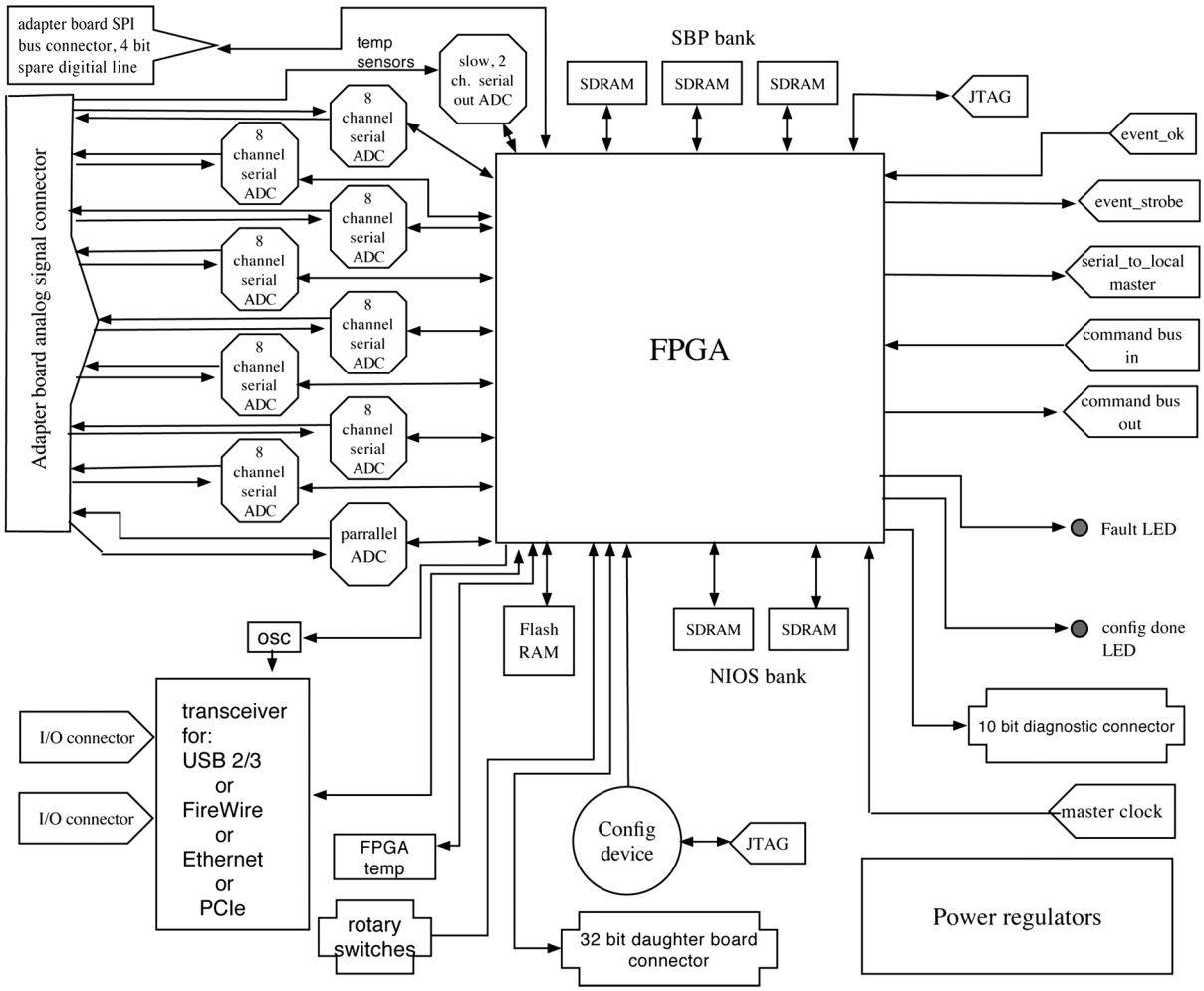


Fig. 5. Block diagram of the Phase II digital board.

is also driven by the trace geometry) to assure that the performance targets for the board can be maintained. For example, each serial ADC integrated circuit contains eight discrete ADCs, each digitizing at rates up to 70 MHz from a common clock. Each IC then connects to the FPGA with eight serial data lines synchronized to a common clock. The data rate on each serial line when digitizing at 70 MHz is 840 Mbps.

The card is designed to accept either 64 channels of differential analog signals (which are then digitized with up to 70 MHz ADCs) or 64 channels of serial signals from a detector adaptor board or digital detector. The board also has one high speed ($> 300$ MHz) ADC channel for those detectors that provide a fast timing pickoff signal. To support the board we have developed a library of Verilog code for pulse integration, timing, and position processing algorithms for the device [7]–[10].

As is indicated in Fig. 5, there are a variety of input/output options for communicating between boards, to the analog adapter board(s), and to the host computer(s). Control and configuration commands can be sent/received from the analog card(s) via an SPI bus connector.

The board provides room for two connectors that can be mounted for FireWire, USB (2 or 3), Ethernet, or other future communication options. While the original electronics designed in our laboratory utilized FireWire [1], the current Phase II card revision we are using is configured to use USB with a Cypress CYUSB3014 transceiver that supports both USB 2 and 3 protocols. For this implementation, we normally only

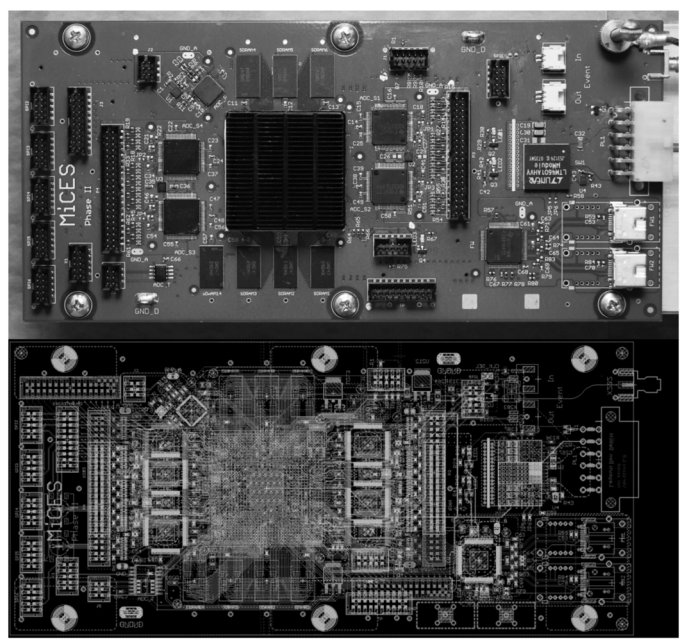


Fig. 6. Phase II board top view and the X-ray image (generated by the Eagle layout software) showing the 12 layers of traces.

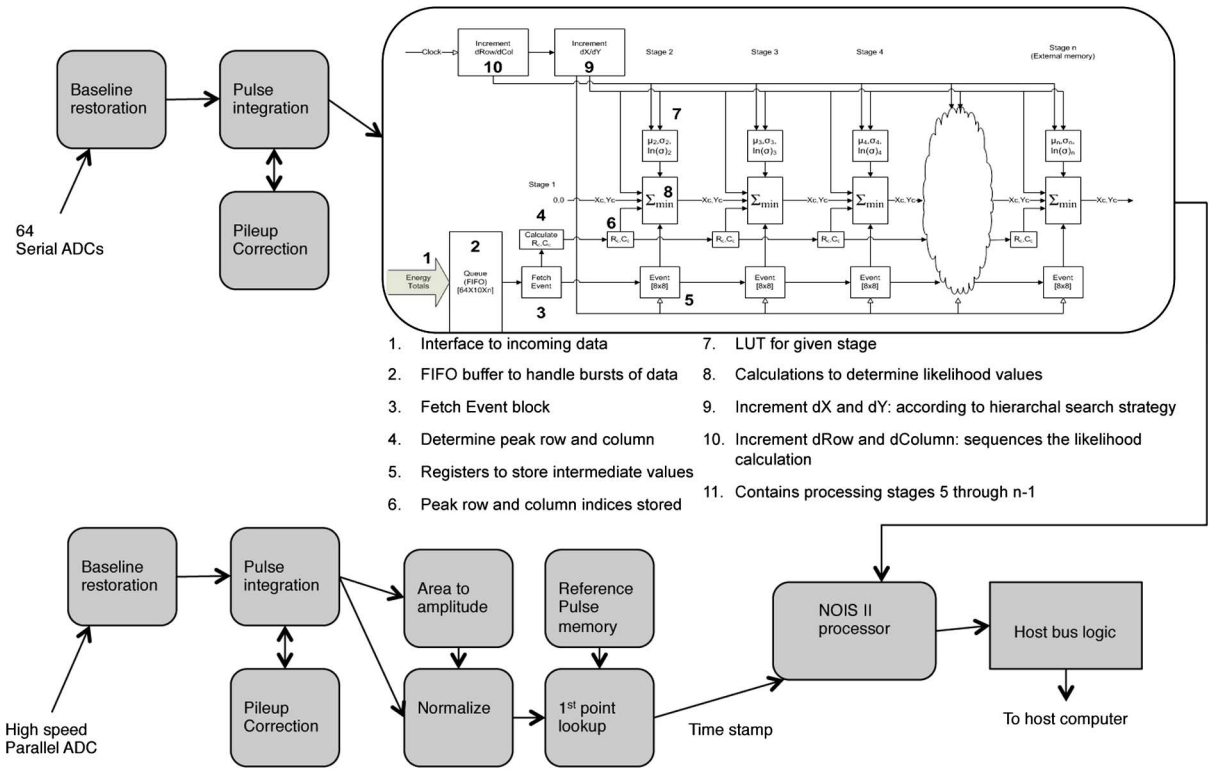


Fig. 7. Diagram of the FPGA logic used for the cMiCE detector module.

mount a single connector and rely on standard USB hubs for connecting the Phase II card to the host computer. For communication with other cards (Phase II or other devices) we provide a general high-speed serial port with control lines (which can be configured to support many different protocols such as $I^2C$ and SPI as well simple point to point serial communications such as RS232 and RS422). We also provide two additional serial ports with control lines that we utilize as a local, daisy chained command bus connecting the Phase II cards to a master or coincidence controller if they are being used in the system. These controllers can be used to issue master start/stop commands, synchronize timers, or other system functions.

The FPGA is configured with the NIOS II software based embedded processor for general control, processing of commands from the host, and communications between the cards. A megabyte of FLASH ROM is included for NIOS software and other data needed to be loaded into the static random access memory (SRAM) supporting the FPGA. The FPGA logic and NIOS II processor are loaded at power up from an Altera load device (currently 16 MByte capacity). An additional FLASH ROM (currently a 128 MByte device) is used to store data such as lookup tables which need to be reloaded after a power cycle. JTAG connectors are provided for programming either the FPGA or the load device as well as for general testing.

Other features of the board include temperature monitoring connectors, as well as expansion connectors for daughter boards (16 bits each) for future requirements. An additional 10 bit "diagnostic connector" is included for general-purpose use. Rotary switches allow additional information to be provided to the FPGA code such as an ID number for the board when it is part of a scanner system. Due to the requirements for the statistical estimator for the cMiCE detector [3], [9], the board also includes 380 Mbytes of SRAM for the FPGA general logic and 64 Mbytes of SRAM for use by the NIOS II embedded processor.

For those designs with a coincidence controller that looks at all acquisition nodes and imposes criteria for acceptance of an event (such as a coarse timing window or a geometric criterion for which detectors can be in coincidence in a PET scanner), we also provide some control signal connections (event_strobe and event_ok in Fig. 5). The use of a coincidence controller with these additional control lines was part of our original MiCES electronics [11] and we have carried over that basic concept in our current scanner designs. There are many options for implementing a coincidence controller, including configuring a Phase II card to provide that function or using a less expensive commercial device such as an Altera DE0-Nano Development and Education Board that has sufficient capacity for the relatively limited functions required for the task.

Fig. 7 is a diagram of the basic FPGA logic structure that supports our cMiCE detectors and represents the most demanding use of the Phase II board resources we have implemented to date. The FPGA code provides baseline restoration and pulse pile-up correction for all 65 ADC channels. The 64 "slow" channels are then used as inputs to the event estimator that locates the event in three dimensions using the estimation algorithms developed in our laboratory [3], [9]. In parallel, the high-speed pickoff-timing signal is processed and the time stamp determined. The data are then combined and sent to the host using the NIOS II soft processor and the host bus interface.

This connectivity flexibility allows supporting different acquisition topologies. Some of the possibilities are diagramed in Fig. 8. The 'Bus' example is the approach used in the original

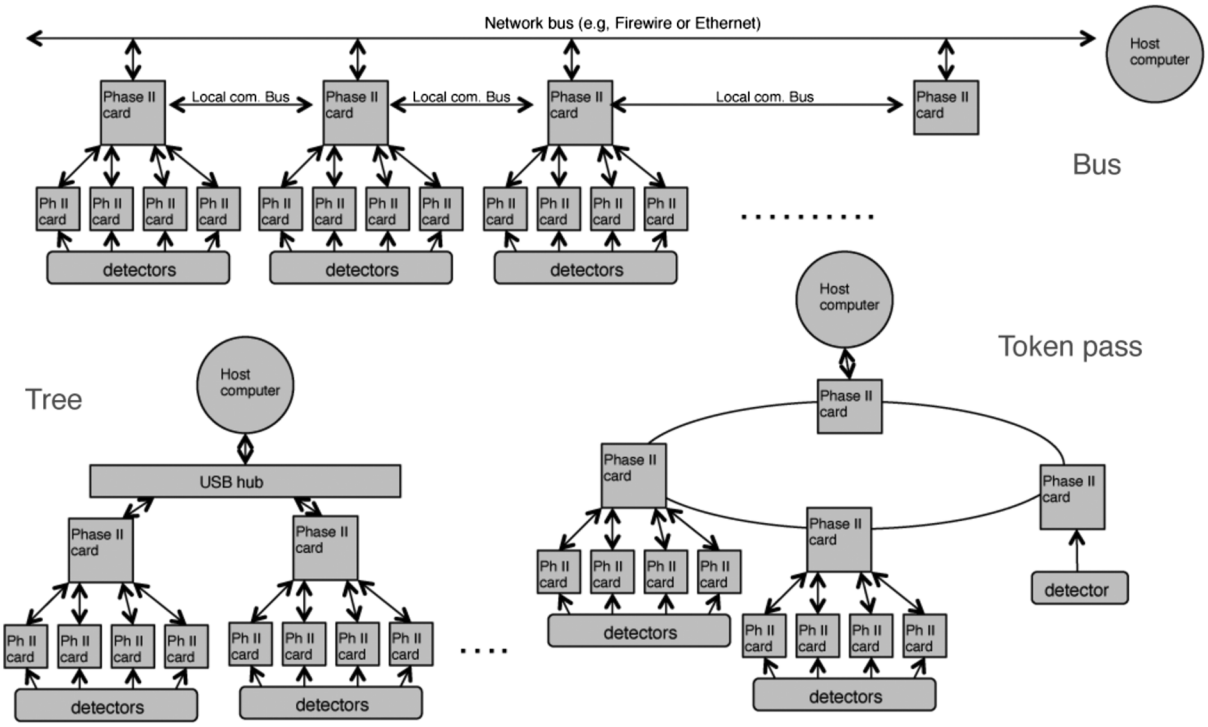


Fig. 8. Examples of some of the bus topologies that can be supported with the Phase II card.

MiCES scanner and is appropriate for a serial bus system such as FireWire that supports independent nodes on the bus connected to the host (the nodes can communicate without the host interacting). The 'Tree' example is the configuration we are using for the USB host connection option. In this case, using the USB 2/3 protocol, there is a bus master (the host) that controls the traffic flow. Hubs are used to connect multiple devices to the host. One could have each USB device be a single Phase II board or, as shown, you can stack Phase II boards so that one acts as a bus node (what we term a master node) and the boards connected to this master node pass data to it. This stack of boards can be extended to many levels to allow a larger number of detector channels to be supported by each bus node (or master node in our general terminology). The 'token pass' example is one we have not implemented in our laboratory, but have allowed enough flexibility in the Phase II board to support such a configuration. In this case, the data is passed between Phase II boards (which could be boards connected directly to detectors or boards using our master node approach to connect phase II boards to a single node on the ring). Typically the data is passed around the ring until criteria to accept or reject an event is reached. If the event is accepted, then one of the nodes passes it to the host. A discussion of many of these topology options can be found in [12], [13].

### *B. Initial Testing*

Fig. 9 shows the configuration for testing with a cMiCE detector. Once all of the components were confirmed to be operating properly, simple Verilog code was written and combined with C code in the NIOS II soft embedded processor to begin testing the card with real data. For these first tests, a prototype detector adapter board was fabricated to provide amplification and conversion of signals to differential pairs from a cMiCE detector utilizing a 64 channel Hamamatsu H8500 PMT. This setup allowed testing of the revision 1.0 Phase II board by recording event by event data digitized from all 64 analog inputs.

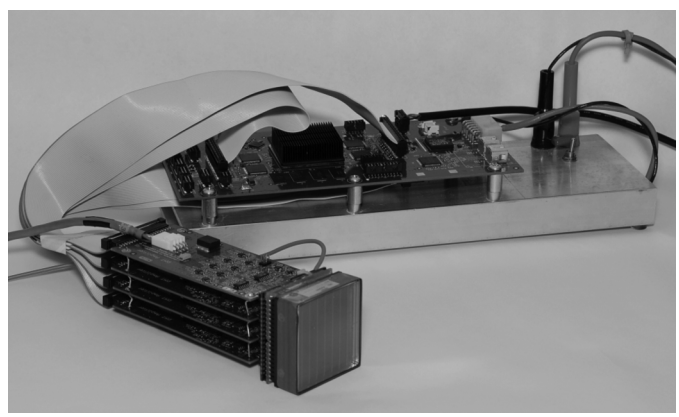

Fig. 9. Phase II board connected to a cMiCE detector with analog cMiCE adapter boards.

Fig. 10 depicts one of the first tests done with signals from a cMiCE PMT based detector. A single event is shown with the digitized outputs from each of the 64 anodes (the grid of small plots) and a zoomed plot of the sum of all the signals. Fig. 11 depicts the result of summing all of the 64 channel "slow" ADCs and comparing it to the high speed ADC that is connected to the timing pickoff signal from the cMiCE detector. The data show the similarity of the two different input channels in terms

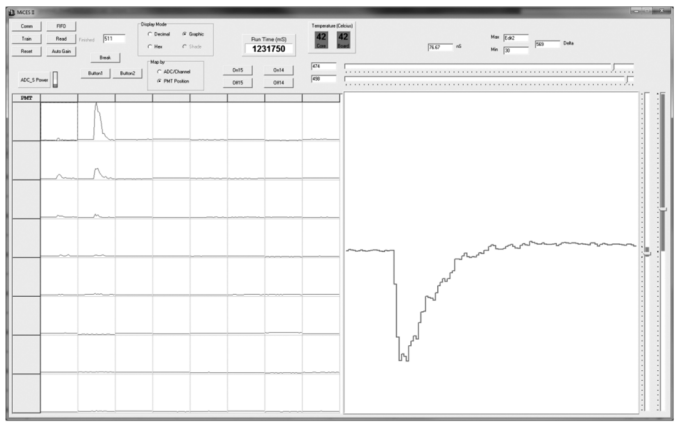

Fig. 10. Example of a single event in the cMiCE detector (grid of 64 channels on the left) and the timing pulse (plot on the right) as recorded by the Phase II digital card.

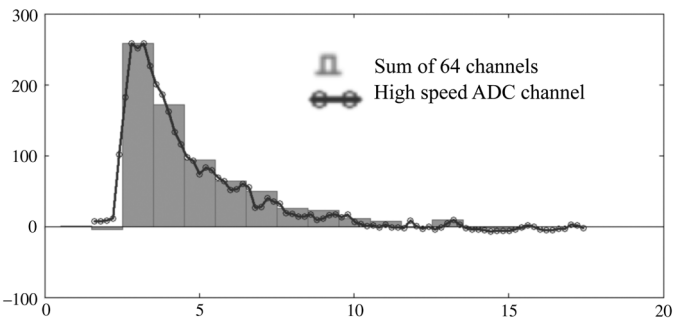


Fig. 11. Comparison of the summed slow ADC channels and the high speed timing ADC for a single event from the cMiCE detector as recorded by the Phase II digital card. The vertical axis are arbitrary units. The 64 channel ADC data are sampled at 64 MHz (15.4 ns intervals). The high speed data were sampled at 300 MHz (3.3 ns intervals).

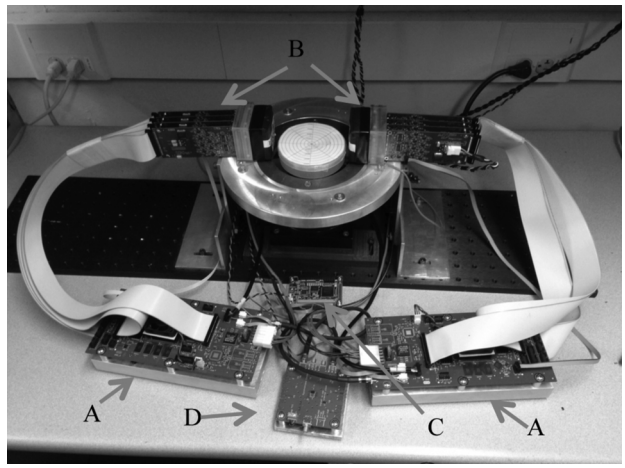


Fig. 12. Two Phase II cards (A) supporting two cMiCE detectors (B) for testing. Also shown is an Altera De0 Nano FPGA card used as the coincidence controller (C) and a clock fan out board (D) developed in our laboratory.

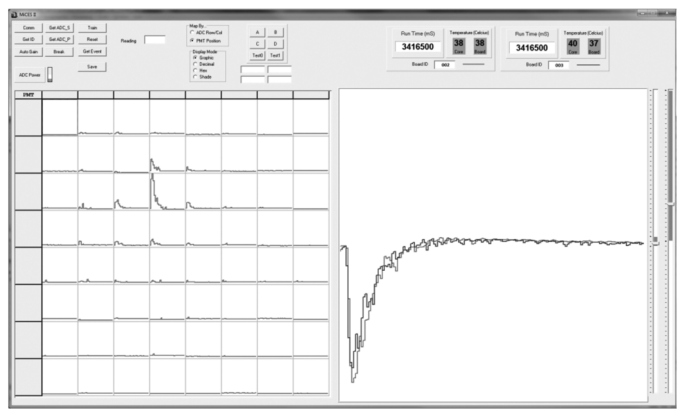

Fig. 13. Data from a single event for the test configuration of Fig. 8. The 64 channels of the left PMT are shown on the left and both timing pickoff channels are shown on the right. In this case, the FPGA logic utilized the Event_out/Event_in signals to notify the paired boards that a coincidence event occurred. The 64 channel ADC data are sampled at 64 MHz (15.4 ns intervals). The high speed data were sampled at 300 MHz (3.3 ns intervals).

of pulse shape, and confirms that possible time skewing issues for the entire chain are all properly compensated for in the basic board design.

Fig. 12 depicts a coincidence test configuration with two cMiCE detectors. Both detectors are connected to Phase II boards using a full set of analog adapter boards. The large flat cables carry the 64 anode signals to the "slow" ADC inputs while the smaller flat cables connect the timing pickoff signal to the "fast" ADC inputs. The clock fan out board (item D) supplies the clocks to the Phase II boards with minimal time skewing. An Altera DE0-Nano development board (item C) is connected to the event control lines of the two Phase II boards and is programmed to provide a rough coincidence window for acceptance of events using the event_strobe and event_ok control line option provided on the Phase II board. One of the results is shown in Fig. 13 where we depict the 64 slow channels processed by the one Phase II board and overlay the fast timing ADC results from both boards for a coincident event. Again, we see that the time skews for the entire processing chain are minimal.

More typical performance measurements (e.g., energy resolution, timing, deadtime, event position accuracy) depend on the exact application the board is to be used for and the FPGA code the user develops/downloads into the card. There are many algorithms that can be implemented on the system and we have published details on the several we have developed in our laboratory [e.g., 3, 7, 8, 9, 10]. Here we will review the performance of the algorithms we have implemented for timing, pulse pileup correction, baseline restoration, and position estimation in monolithic detectors that we have previously published.

The first step in a typical processing task is that of pulse integration and timing. While we have one parallel, high speed ADC on the Phase II card for those detectors that offer a timing pickoff signal, we also had to support detectors (such as SiPM arrays) that did not have an analog summing channel to provide a timing signal. For that case we use the slower, serial ADCs and bandwidth limit the incoming analog signal such is illustrated in Fig. 14.

In our typical implementation, the bandwidth limited pulse is processed by fitting a reference pulse as illustrated in Fig. 15. The incoming pulse is sampled and then integrated. The area is then used to normalize a reference pulse. The reference pulse, which is usually determined with 40 ps resolution, is then used to determine the time stamp for the event. We have also been

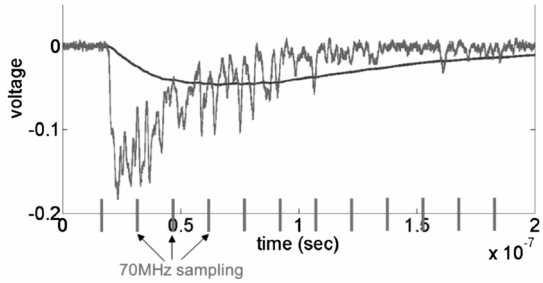


Fig. 14. Example of bandwith limiting a typical SiPM pulse to 33 MHz and the sampling intervals for a 70 MHz ADC.

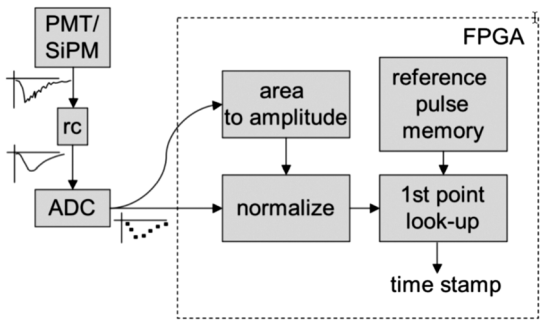


Fig. 15. The pulse integration and timing algorithm developed for the Phase II board.

able to develop a calibration routine for the FPGA that can derive the reference pulse from a large sampling of pulses by the slow ADCs and get the same result as digitizing the pulse at a high rate (e.g. 25 GHz) [8].

Measurements at slower ADC rates were compared to simulation results and found to be in good agreement [10]. Fig. 16 depicts a timing measurement using the detector configuration of Fig. 12 with Hamamatsu PMTs. We note that the existing analog boards did not have bandwidth limiting of the analog pulses optimized for the Phase II board modeled pulse timing algorithm

For most of our current applications, high singles rates in the detectors is not a major issue. However, as rates are increased, the issue of pulse pileup arises and implementation of methods to correct for it. Algorithms for pileup correction and baseline restoration have been developed for the Phase II board and details were presented in [7] which also provides an overview of many of the approaches that have been previously implemented by others, as well as the rational for the approach implemented for the Phase II card. This paper presented results obtained with pulses from one particular SiPM device that was not optimized for fast timing. The main goal was to preserve energy resolution and timing over a wide range of counting rates. Here we reproduce one of the result plots for the timing resolution with the 300 MHz ADC used on the current Phase II card (Fig. 17) as well as the results for energy resolution for different ADC sampling rates versus count rate.

While we have not yet used the Phase II card with a discrete crystal block, we have used it with monolithic detectors since that represents the most complex pulse processing we have implemented in the FPGA to date. We note that for our laboratory needs, we will use discrete crystal blocks with one-on-one readout (one SiPM for each crystal) for our dMiCE detectors. Thus for those detectors the event positioning will all be done by the host computer during the image reconstruction process [4]. Fig. 19 depicts a cMiCE detector data set acquired in a calibration apparatus and then processed with the algorithm was developed for FPGAs [3], [7] and implemented on the Phase II card. We have not yet used a Phase II board directly with our calibration system, but have determined that the Phase II board integrated pulse values for detector pulse trains in general are the same as those obtained with our older CAMAC system.

We have found that the algorithm works well with either the original cMiCE detector or the SES variant as described above. Fig. 19 depicts a plot of the FWHM contours for a grid of point sources (acquired in coincidence) with the detector arrangement of Fig. 12. The detector consisted of a 15 mm thick LYSO $50 \times 50$ mm coupled to a Hamamatsu 8x8 multi-anode PMT using existing analog electronics (a cMiCE detector module).

## III. Discussion and Conclusion

This project was undertaken initially to support the cMiCE and dMiCE detector designs in our laboratory. At the time the project started, we were unable to find commercial products that would support the estimation tasks at the detector level required by our cMiCE approach. Similarly, the OpenPet initiative (http://openpet.lbl.gov/) was in the early stages of development and did not provide the capabilities we wished for the data reduction for the cMiCE detector. Many other groups have explored the use of FPGAs to implement many different pulse processing approaches and incorporate them into acquisition systems [e.g., 12-28]. But again, we did not feel that those projects were an optimal fit for our needs, but they did provide a high level of confidence that our goals could be achieved with modern FPGAs.

The board is meeting its design specifications and is now being used to develop a pre-clinical scanner and also a human breast scanner based on the cMiCE detector concept. The card is also being used by one commercial vendor for development of a single photon detection system. The next revision of the board (expected in the Fall of 2013) is expected to add a dedicated daughter board architecture to allow converting any Phase II board between the different host bus options (USB, FireWire, Ethernet, PCIe, etc). The current restrictions of such an adapter board are dictated by the current transceiver to FPGA data path (based on an 8 bit parallel bus with several control lines) and preserving the bandwidth of the different bus standards. Thus far, all of the transceivers we have considered can easily be adapted to the current restrictions. However, as we make modifications for the next revision of the board, our intention is to add enough control lines to the dedicated daughter board to support a 16 bit wide data path and additional control lines.

Along with the card, we have a reconfigurable software tool for list mode acquisition that will support the bus types the Phase II card is configured for (currently FireWire or USB). Standard command and control code has also been written for the embedded processor on the Phase II board to allow easy adaptation to many different acquisition topologies–a few examples of which are shown in Fig. 8. We are working on refining our library of Verilog and C code modules for the board to allow

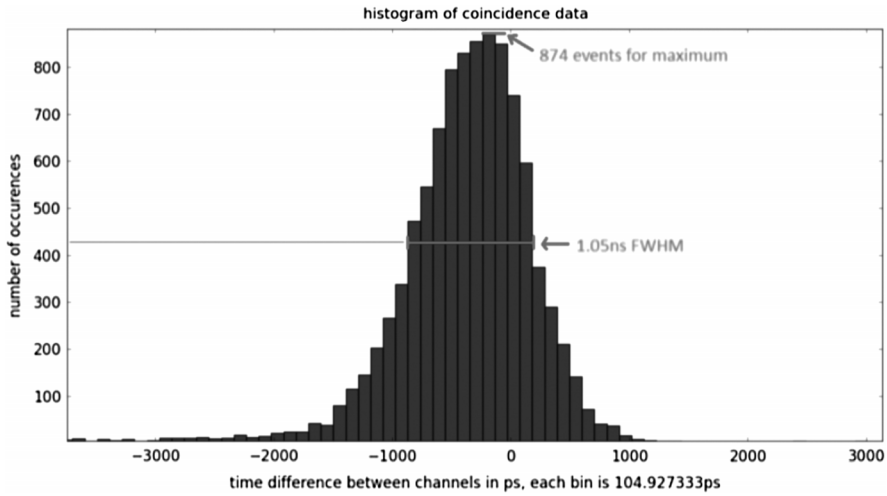


Fig. 16. Timing spectrum between two cMiCE detectors using the 300 MHz ADCs on the Phase II board and the model reference pulse approach for generating time stamps. FWHM was measured to be 1.05 ns without matching the analog pulse bandwidth to one optimized for the Phase II board timing algorithm.

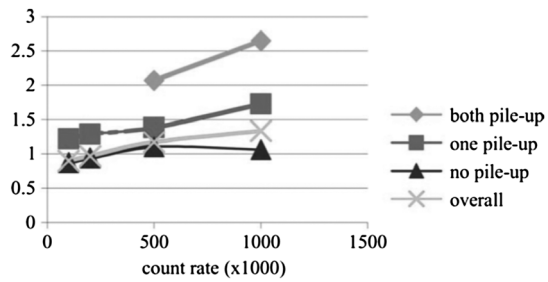


Fig. 17. Plot of timing resolution for a 300 MHz ADC sampling for different count rates with the pileup correction algorithm. The timing resolution is shown for coincidental pulses where neither of the pulse is in a pile-up event, where one of the pulses is piled up and where both of the pulses were in a pile-up event.

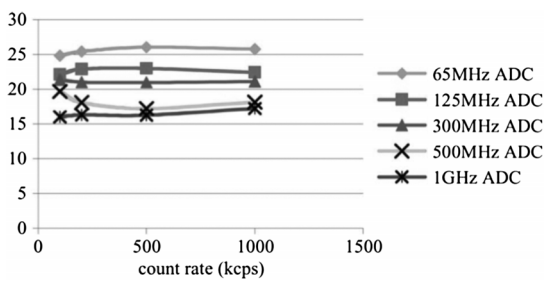


Fig. 18. Plot of the energy resolution with the pulse pileup algorithm versus count rate for different ADC sampling rates.

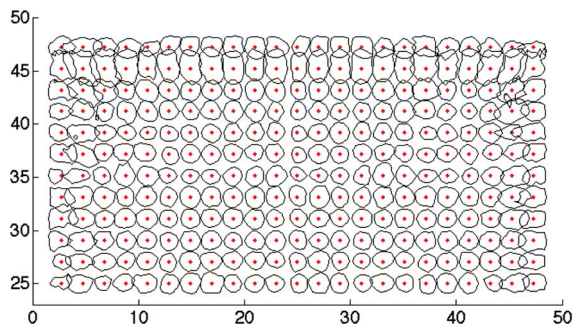

Fig. 19. Intrinsic spatial resolution positioning results for 15 mm thick cMiCE detector using a CAMAC based 128 channel ADC system and an 8x8 MA-PMT using the event positioning algorithm implemented on the Phase II board.

more rapid development of new applications for the board that can use the various logical blocks we have developed.

In this paper we have not presented deadtime measurements since that is highly dependent on the detector used and the choice of FPGA algorithms and the number of buffers the user decides to implement within the FPGA to hold pulse data and results of pulse processing. The current board has a sufficiently large FPGA that we are limited in count rate capability by the pulse pileup. Issues on bus bandwidth to the host computer are handled by designing large enough buffers to hold burst data within the card until it can be transferred to the host. This is possible since each event is uniquely time stamped. With our pulse fitting approach using a pulse bandwidth limit of 30 MHz, the time stamp resolution of the current card is 40 ps. The maximum transfer rate from a card to the host depends on the system topology and bus type used. As stated earlier, the current card is now configured to use USB 2 or 3. USB 3 has a maximum bus transfer rate of 640 megabytes per second (MBps), but bus overhead and the needed transfer of commands limits how much bus bandwidth can actually be used for transfer of the detector data. In our tests, we are currently seeing maximum transfer rates from the Phase II card to the host of 144 MBps - essentially the same rates one sees with a solid state disk drive. As more nodes are added to the system, the amount of bus time for each node is reduced which is then compensated by the system designer with the addition of more data buffers in the FPGA and USB integrated circuit. In the end, the major contributor to dead time other than pulse pileup will be the deadtime in the host computer to process incoming data packets.

We have not included host performance in this paper since it is a function of which computer is selected and the design of the host software. In our own laboratory, we have extended the software we developed for our original MiCES scanner with Apple Macintosh computers [1] and will be benchmarking the software with the new Macintosh Pro multiprocessor systems with Thunderbolt 2 disk drives in the Fall of 2013. This code treats

each node on the bus as an independent data source and allocates a list mode buffer for each one. At the end of the acquisition, the individual list mode files are sorted into a single file based on the event time stamps. To assure that the node clocks are synchronized, we use a master clock fan out with equal length cables connected to the external clock in on each of the Phase II cards. We also use the same scheme as for the original MiCES scanner [1] and implement a master controller that generates the master stop/start signals (synced to the master clock). This master controller also handles events on an event bus (event_strobe and event_ok in Fig. 5) to reject or accept events between cards that are within a coarse timing window (normally 40 ns) and are within the geometric acceptance window between detector modules for any given scanner.

As a result of this development, we now have a basic module that we can configure for any of the scanner/detector designs ongoing in our laboratory with considerable options for refinement of processing algorithms and support for future detector systems not yet envisioned.

## REFERENCES


[1] T. K. Lewellen, R. S. Miyoka, L. R. MacDonald, D. DeWitt, and S. Hauck, "Evolution of the design of a second generation firewire based data acquisition system," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2011, pp. 3994–3998.

[2] X. Li, W. C. J. Hunter, T. K. Lewellen, and R. S. Miyaoka, "Spatial resolution performance evaluation for a monolithic crystal PET detector with Cramer-Rao lower bound," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2010, pp. 2202–2205.

[3] D. DeWitt, R. S. Miyaoka, L. Xiaoli, C. Lockhart, T. K. Lewellen, and S. Hauck, "Design of an FPGA based algorithm for real-time solutions of Statistics-Based Positioning," *Trans. Nucl. Sci.*, vol. 57, no. 1, pp. 2769–2776, 2010.

[4] K. M. Champley, T. K. Lewellen, L. R. MacDonald, R. S. Miyaoka, and P. E. Kinahan, "Statistical LOR estimation for High-Resolution dMiCE PET detector," *Phys. Med. Bio.*, vol. 54, pp. 6369–6382, 2009.

[5] S. Dey, L. Banks, S. Chen, X. Wenbin, T. K. Lewellen, R. S. Miyaoka, and J. Rudell, "A CMOS ASIC design for SiPM arrays," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2011, pp. 732–7737.

[6] W. W. Moses, S. Buckley, C. Vu, Q. Peng, N. Pavlov, W. S. Choong, J. Wu, and C. Jackson, "OpenPET: A flexible electronics system for radiotracer imaging," *IEEE Trans. Nucl. Sci.*, vol. 57, pp. 2532–2537, 2010.

[7] M. D. Haseleman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "FPGA-Based pulse pileup correction with energy and timing recovery," *IEEE Trans. Nucl. Sci.*, vol. 59, pp. 1823–1830, 2012.

[8] M. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "FPGA-Based pulse parameter discovery for positron emission tomography," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2009, pp. 2956–2961.

[9] N. G. Johnson-Williams, R. S. Miyaoka, L. Xiaoli, T. K. Lewellen, and S. Hauck, "Design of a real time FPGA-Based three dimensional positioning algorithm," *IEEE Trans. Nucl. Sci.*, vol. 58, pp. 26–233, 2011.

[10] M. D. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Simulation of algorithms for pulse timing in FPGAs," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2007, pp. 3161–3165.

[11] T. K. Lewellen, M. Janes, R. S. Miyaoka, S. B. Gillespie, B. Park, K. S. Lee, and P. Kinahan, "System integration of the MiCES small animal PET scanner," in *Proc. IEEE Nul. Sci. Symp. Med. Imag. Conf.*, 2004, vol. 5, pp. 3316–3320.

[12] K. Ealgoo, J. H. Key, Y. Y. Jung, P. D. Olcott, and C. S. Levin, "The trend of data path structures for data acquisition systems in positron emission tomography," in *Proc. 18th IEEE-NPSS Real Time Conf. (RT)*, 2012, pp. 1–8.

[13] K. Ealgoo, 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.*, 2011, pp. 3871–3873.

[14] L. Jiguo, L. Hongdi, L. Shitao, W. Yu, K. Soonseok, Z. Yuxuan, H. Baghaei, R. Ramirez, and W. Wai-Hoi, "A low dead time full digital pulse-shape-discriminator (PSD) for DOI PET," in *Proc. IEEE Nucl. Sci. Symp. Conf. Rec.*, 2007, pp. 3314–3317.

[15] M. D. Fries and J. L. Willaims, "High-precision TDC in an FPGA using a 192-MHz qaudrature clock," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2002, pp. 580–584.

[16] B. Von Herzen, "Signal processing at 250 MHz using high-performance FPGAs," in *Proc. ACM Int. Symp. PFGAs*, 1997, pp. 62–68.

[17] R. Fontaine, F. Belanger, N. Viscogliosi, M. A. Terrault, P. Berad, J. Cadorette, J. D. Leroux, J. B. Michaud, C. Pepin, J. F. Pratte, S. Robert, and R. Lecomte, "Preliminary results of a data acquisition sub-system for distributed, digital, computational, APD-based, dual modality PET/CT architecture for small animal imaging," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2004, in press.

[18] P. Guerra, J. E. Ortuno, G. Kontaxakis, M. J. Ledesma-Carbayo, J. J. Vaquero, M. Desco, and A. Santos, "Real-Time digital timing in positron emission tomography," *IEEE Trans. Nucl. Sci.*, vol. 55, pp. 2531–2540, 2008.

[19] G. Tambave, M. Kavatsyuk, E. Guliyev, F. Schreuder, H. Moeini, and H. Lohner, "Pulse pile-up recovery for the front-end electronics of the PANDA electromagnetic calorimeter," *J. Instrum.*, vol. 7, pp. P11001–11024, 2012, (23 pp.).

[20] H. Wei, C. Yong, J. H. Key, K. Jihoon, H. J. Jin, S. H. Youn, K. L. Hyun, S. K. Sang, K. Byung-Tae, and C. Yonghyun, "Free-running ADC- and FPGA-based signal processing method for brain PET using GAPD arrays," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, pp. 370–375, 2012.

[21] J. H. Key, K. Ealgoo, Y. Y. Jung, P. D. Olcott, and C. S. Levin, "FPGA-based time-to-digital converter for time-of-flight PET detector," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. Rec.*, 2012, pp. 2463–2465.

[22] H. J. Jin, C. Yong, J. H. Key, K. Jihoon, H. Wei, K. L. Hyun, H. Yoonsuk, K. Sangsu, J. Jiwoong, and B. K. Kyu, "Development of brain PET using GAPD arrays," *Med. Phys.*, vol. 39, pp. 1227–1233, 2012.

[23] W. J. Ashmanskas, Z. S. Davidson, B. C. LeGeyt, F. M. Newcomer, J. V. Panetta, W. A. Ryan, B. R. Van, R. I. Wiener, and J. S. Karp, "Combined analog/digital approach to performance optimization for the LAPET whole-body TOF PET scanner," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. Rec.*, 2012, pp. 3496–3500.

[24] L. H. C. Braga, L. Gasparini, and D. Stoppa, "A time of arrival estimator based on multiple timestamps for digital PET detectors," in *IEEE Nuc. Sci. Symp. Med. Imag. Conf. Rec.*, 2012, pp. 1250–1252.

[25] J. M. Cook, J. M. Palmer, E. C. S. Rabin, L. C. Stonehill, D. C. Thompson, S. R. Whittemore, and M. D. Ulibarri, "A photon-counting camera system developed from a crossed-strip detector," *Proc. SPIE, Int. Soc. Opt. Eng.*, vol. 8460, pp. 84601–84611, 2012.

[26] W. Gao, D. Gao, B. Gan, L. Wang, Q. Zheng, F. Xue, T. Wei, and Y. Hu, "A novel data acquisition scheme based on a low-noise front-end ASIC and a high-speed ADC for CZT-based PET imaging," in *Proc. IEEE-NPSS Real Time Conf.*, 2012, pp. 4–8.

[27] J. Ohi, Y. Yamakawa, M. Satoh, M. Furuta, and K. Kitamura, "Development of a digital baseline restorer for high-resolution PET detectors," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf.*, 2011, pp. 3602–3604.

[28] R. T. Schiffer, M. Flaska, S. A. Pozzi, S. Carney, and D. D. Wentzloff, "A scalable FPGA-based digitizing platform for radiation data acquisition," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 652, pp. 491–493, 2011.