FPGA-Based Front-End Electronics for Positron Emission Tomography

Michael Haselman[1], Don DeWitt[1], Wendy McDougald[2], Thomas K. Lewellen[2], Robert Miyaoka[2], Scott Hauck[1]
[1]Department of Electrical Engineering, [2]Department of Radiology
University of Washington, Seattle, WA
{haselman, dewitdq, hauck}@ee.washington.edu,
{wam2, tkldog, rmiyaoka}@u.washington.edu

Abstract

Modern Field Programmable Gate Arrays (FPGAs) are capable of performing complex discrete signal processing algorithms with clock rates above 100MHz. This combined with FPGA's low expense, ease of use, and selected dedicated hardware make them an ideal technology for a data acquisition system for positron emission tomography (PET) scanners. Our laboratory is producing a high-resolution, small-animal PET scanner that utilizes FPGAs as the core of the front-end electronics. For this next generation scanner, functions that are typically performed in dedicated circuits, or offline, are being migrated to the FPGA. This will not only simplify the electronics, but the features of modern FPGAs can be utilizes to add significant signal processing power to produce higher resolution images. In this paper two such processes, sub-clock rate pulse timing and event localization, will be discussed in detail. We show that timing performed in the FPGA can achieve a resolution that is suitable for small-animal scanners, and will outperform the analog version given a low enough sampling period for the ADC. We will also show that the position of events in the scanner can be determined in real time using a statistical positioning based algorithm.

Categories and Subject Descriptors

B.7.1 [Integrated Circuits]: Types and Design Styles—*Gate arrays, Algorithms implemented in hardware*

General Terms

Algorithms, Measurement, Design, Verification

Keywords

PET, FPGA, Digital Signal Processing, Pulse Timing, Event Localization

1. Introduction

The ability to produce images of the inside of a living organism without invasive surgery has been a major advancement in medicine over the last 100 years. Imaging techniques such as X-ray computer tomography (CT) and magnetic resonance imaging (MRI) have given doctors and scientists the ability to view high-resolution images of the anatomical structures inside the body.

While this has led to advancements in disease diagnosis and treatment, there is a large set of diseases that only manifest as changes in anatomical structure in the late stages of the disease, or never at all. This has given rise to another branch of medical imaging, called functional imaging, which is able to capture certain metabolic activities inside a living body. Positron emission tomography (PET) is the most advanced form of this imaging modality.

Traditionally, the front-end electronics of PET scanners have consisted of many discrete parts that perform the necessary data acquisition and pulse processing functions. There are current efforts to utilize FPGAs for all or most of the data acquisition [3,4] because they provide the necessary speed to process the incoming data without the complexity of an application-specific circuit. Technology advances have also yielded FPGAs with the logic capacity to fit all of the necessary task as well as adding tasks that where not possible before. Most current scanners use the FPGAs to collect data and do some simple filtering. This paper discusses our use of FPGAs in a PET scanner and how we utilize the capabilities of modern FPGAs to help produce higher resolution images.

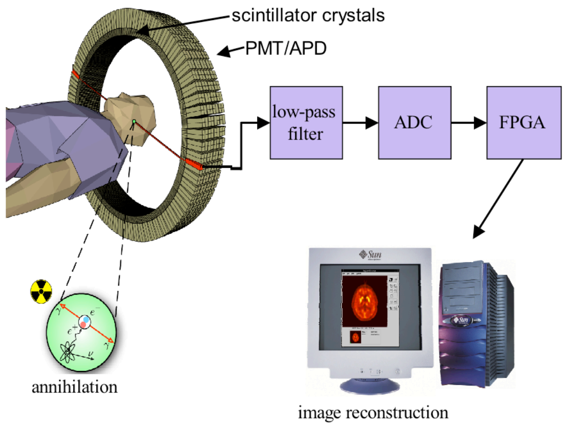

Figure 1. Drawing of a PET scanner ring and attached electronics.

2. Positron Emission Tomography

PET is a medical imaging modality that uses radioactive decays to measure certain metabolic activities inside living organisms. It does this through three main components (see Figure 1). The first step is to generate and administer the radioactive tracer. A tracer is made up of a radioactive isotope attached to a metabolically active molecule. The tracer is then injected into the body to be scanned. After enough time has lapsed for the tracer to distribute and concentrate in certain tissues, the subject is placed inside the scanner. The radioactive decay event for tracers used in PET studies produces positrons. The positrons will travel a short distance in tissue (as illustrated in Figure 2) and eventually annihilate with an electron to produce two 511KeV antiparallel photons. These photons are what the scanner captures. The scanner, the second component of PET, consists of a ring of sensors that detect the photons and electronics that process the signals arising from the sensors. The sensors are made up of scintillator crystals and photodetectors. The scintillator converts the 511KeV photon into many visible light photons, and the photodetector converts the visible light into an electrical pulse. These pulses are processed by the front-end electronics to determine the parameters of each pulse (i.e. energy, timing). Finally, the data is sent to a host computer that performs tomographic image reconstruction to turn the data into a 3-D image.

2.1 Radiopharmaceutical

The first step to produce a PET scan is to generate a radiopharmaceutical (often called a tracer). To synthesize the tracer, a short-lived radioactive isotope is attached to a metabolically active molecule. The short half-life reduces the exposure to ionizing radiation, but it also means that the tracer has to be produced close to the scanners, as it does not store for very long. The most commonly used tracer is fluorine-18 flourodeoxyglucose ([F-18]FDG). [F-18]FDG is an analog of glucose that has a half-life of 110 minutes. [F-18]FDG is similar enough to glucose that it is phosphorylated by cells that utilize glucose, but the modifications do not allow it to undergo glycolysis. This means that the radioactive portion of the tracer is trapped in the consuming tissue. This means that cells that consume lots of glucose, such as cancers and brain cells, accumulate more [F-18]FDG over time relative to other tissues.

2.2 Decay Event

After sufficient time has passed for the tissue of interest to uptake enough tracer, the scanner can be used to detect the radioactive decay events, as shown in Figure 2. The isotopes used in PET are positron emitters. When a positron is emitted, it travels a few millimeters in tissue before it annihilates with an electron. The annihilation results in the generation of two 511KeV photons that travel away from the annihilation at 180° ± .23° from one another.

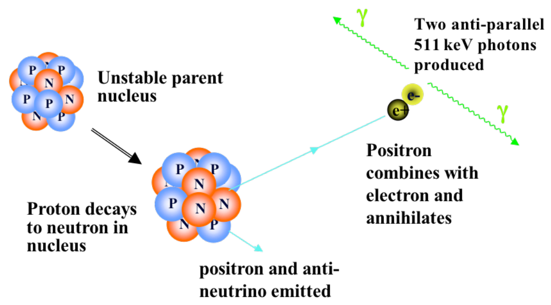
Unstable parent nucleus
Proton decays to neutron in nucleus
positron and anti-neutrino emitted
Positron combines with electron and annihilates
Two anti-parallel 511 keV photons produced
γ
γ
e+
e-

Figure 2. Physics of a positron decay and electron annihilation [2].

2.3 Photon Scintillation

A 511KeV photon has a substantial amount of energy and will pass through many materials, including body tissue. While this is beneficial for observing the photon outside the body, it makes it difficult to detect the photon. Photon detection is the task of the scintillator. A scintillator is a crystal that absorbs high-energy photons and emits light in the visible spectrum. Scintillators can be made up of many different materials, including plastics, organic and inorganic crystals, and organic liquids. The scintillator that we use in our experiments is $Lu_2SiO_5(Ce)$, or LSO, which is an inorganic crystal.

2.4 Photodetectors

Attached to the scintillator are photodetectors that convert the visible light photons into electronic pulses. The two most commonly used devices are photomultiplier tubes (PMT) and micro-pixel avalanche photo diodes (MAPD). The PMT is a vacuum tube with a photocathode, several dynodes, and an anode that has high gain to allow very low levels of light to be detected. MAPDs are the semiconductor version of the PMT that operate in Geiger mode so that when a photon interacts and generates a carrier, a short pulse of current is generated. The array consists of about 103 diodes per mm2. All of the diodes are connected to a common silicon substrate so the output of the array is a sum of all of the diodes. The output can then range from one diode firing to all of them firing. This gives theses devices a linear output even though they are made up of digital devices.

2.5 Front-end electronics

The analog pulses arising from the PMTs/MAPDs (shown in Figure 3), along with their origin, contain the information needed to create a PET image. These pulses have to be processed to extract start time, location, and total energy. This is done with the front-end electronics (filter, ADC, and FPGA in Figure 1). The pulse is filtered with a low pass filter to remove some of the high frequency noise (shown in Figure 3). Before the analog pulse is then sent to the FPGA, it is digitized with an analog to digital converter (ADC). Even though there are ADCs that can sample up to 400 mega samples per second (MSPS), we have chosen to use an ADC that samples at 70MSPS. This will greatly reduce the complexity, cost and power consumption of the design. Another consideration is the number of inputs to the FPGA. The very fast ADCs have a parallel output which would require 10-12 bits per channel, with tens to hundreds of channels per FPGA. The number of inputs would thus outnumber the amount that even modern FPGAs can support. The solution is to use serial output ADC, which also limits the sampling rate to around 100MSPS for current technologies.

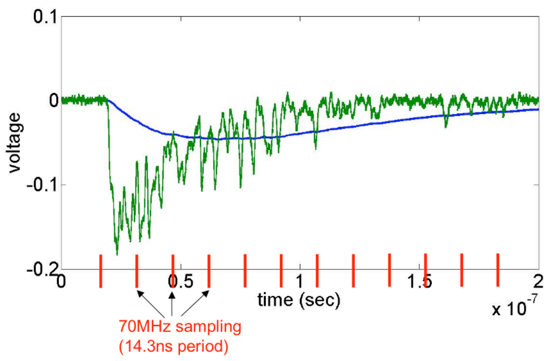

Figure 3. MATLAB plot of a pulse from a LSO scintillator crystal attached to a PMT. A simulated low-pass filter with a 33.3MHz cutoff is also shown. The wide lines on the x-axis indicate the interval at which a 70MHz ADC would sample.

Once the data is digitized, the needed parameters can be extracted in the FPGA. The required data is the energy of the pulse, the start time of each pulse, and the location of each interaction in the detector.

The energy is important, because it indicates if the interaction in the scintillator was a scatter (only part of the energy of the 511KeV photon was deposited), a photoelectric interaction (all of the energy is deposited), or the interaction was from a photon from another source.

The event location refers to where in the detector the interaction occurred. The detectors are about 20mm square, so just knowing what detector the interaction occurred in would result in very poor resolution. In order to improve the resolution, techniques are used to determine where in the detector the interaction occurred within a couple of millimeters. These techniques are discussed in detail in section 6.

The start time of the pulse is important for determining coincidence pairs (timing will be discussed extensively in section 5). Coincidence pairs refer to the two photons that arise from a single annihilation event. Many of the photons from a radioactive event don't reach the scanner because they are either absorbed or scattered by body tissue, they don't hit the scanner, or the photon passes through the scintillator without interacting with it. The scanner works by detecting both photons of an event and essentially draws a line that represents the path of the photons. If only one of the two emitted photons hits the scanner, there is no way to determine where the event occurred. To determine if two photons are from the same event, they have to occur within a certain time of each other and they need to be within the field of view (FOV), as shown in Figure 4. Considering the photons travel at essentially the speed of light, the timing portion requires a very precise time stamp be placed on each event. The better the precision of the time stamp, the lower the probability that two separate random events will be paired together. The FOV for one detector (detector is defined as a scintillator/PMT pair) consists of a set of detectors on the opposite side of the scanner. For example, in our scanner, the FOV for a detector consists of the detector on the opposite side of the scanner and two detectors on either side for a total of five detectors. The FOV is set to detect signals that originate at the edge of the object being scanned (detectors A and B in Figure 4). If the FOV is set too large, events that have one photon scattered could be erroneously determined to be coincidental, (detectors B and C in Figure 4).

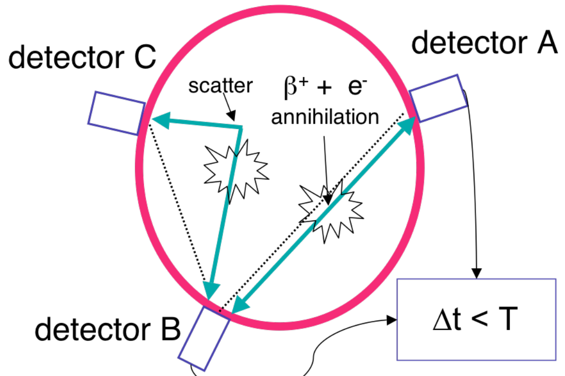

Figure 4. Coincidence detection in the PET scanner. If events are detected in detectors A and B within a certain time period, then the events are saved as coincidental pairs and are considered to have arisen from the same event. If events are detected in detectors B and C, then the event is ignored because it is out of the imaging field of view (FOV). [7]

2.6 Image Reconstruction

When enough coincidental events have been sent to the host computer, image reconstruction can begin. The details of image reconstruction are beyond the scope of this paper. Essentially lines of response are drawn between each interaction of a coincidental pair. Where more lines cross, it is deduced that more radioactive activity is present. The result of reconstruction is shown in Figure 5.

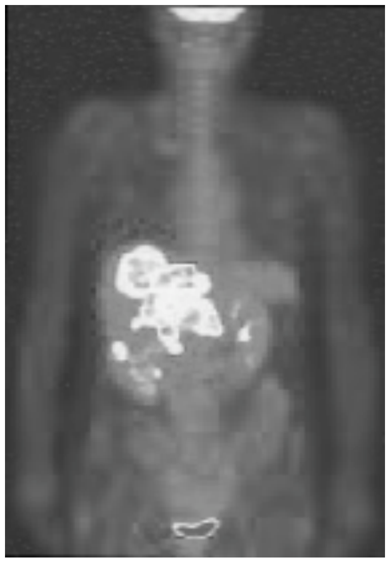

Figure 5. Whole body PET scan image using [F-18]FDG. Notice all of the body has some radioactivity because glucose is utilized in many cells. Notable "hot" spots are in the abdomen, the brain, and the bladder near the bottom of the image (excess glucose is excreted in the urine).

2.7 Uses of PET

While PET, magnetic resonance imaging (MRI), and computed tomography (CT) are all common medical imaging techniques, the information obtained from them is quite different. MRI and CT give anatomical or structural information. That is, they produce a picture of the inside of the body. This is great for problems such as broken bones, torn ligaments or anything else that presents as abnormal structure. However, it doesn't give any indication of metabolic activity. This is the domain of PET. The use of metabolically active tracers means that the images produced by PET provide functional or biochemical information.

Oncology (study of cancer) is currently the largest use of PET. Certain cancerous tissues metabolize more glucose than normal tissue. [F-18]FDG is close enough to glucose that cancerous cells readily absorb it, and therefore they have high radioactive activity relative to background tissue during a scan. This enables a PET scan to detect some cancers before they are large enough to be seen on a MRI scan. They are also very useful for treatment progression, as the quantity of tracer uptake can be tracked over the progression of the therapy [8]. In other words, the number of events detected from cancerous tissue is directly related to metabolic rate of the tissue. So, if a scan indicates lower activity in the same cancerous tissue after therapy, it indicates the therapy is working. There are other uses for PET in neurology (study of the nervous system) and cardiology (study of the heart).

4. Scanner Architecture

We are developing new front-end electronics that are based on an Altera StatixIII. This board is targeted towards the solid state MAPDs (see section 2.4), which has one output per crystal in an 8x8 array. These 64 channels will be summed by row and column to produce 16 channels. Each FPGA will process four detectors for a total of 64 ADC channels from 8 chips (8 channels per ADC chip). Each ADC channel will have 10-bits of resolution. If we used parallel ADCs, the FPGA would have 640 inputs for ADCs alone. To reduce the I/O count to a reasonable amount, we will use serial ADCs (we are using a 70MHz serial ADC), which can easily communicate with the Stratix III using the dedicated serializer/deserializer and phase lock loops. We have discovered that the bottleneck in our design is the built in phase-lock loops (PLLs) on the FPGA. Unfortunately, because each ADC chip requires a separate PLL for dynamic phase alignment, 8 PLLs are needed for the ADCs. This requires us to use StratixIII EPSL200 1517 pin package to have extra PLLs for clock management event though we don't anticipate the need for the logic provided by this large FPGA. There are other parts of a modern FPGA that make them perfectly suited to our application. A NiosII microprocessor will be used to communicate with the host computer to control the scan. A FireWire core from Mindready [1] eliminates the need for a separate TI FireWire chip, as the link layer will be in the reconfigurable fabric of the FPGA and the transactional layer can be implemented in the Nios II processor. Finally, the signal processing requires the use of the dedicated memories and multipliers.

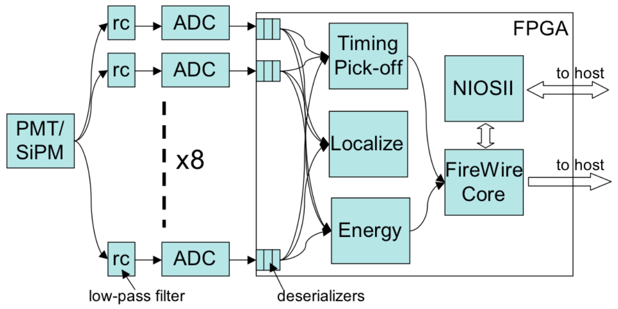

Figure 6. Front-end electronics for the next generation scanner in an Altera Stratix III FPGA. Note there will be 8-8 channel ADCs.

5. Timing

A consequence of having so many channels is that the cost and board space required to have a timing ASIC for each of the channels would be prohibitive. This has led us to develop an algorithm to perform the complete timing inside the FPGA.

A crucial piece in producing high-quality PET images is accurately determining the timing of the photon interacting with the scintillator crystal. This is because the timing resolution is directly correlated to the number of non-coincidental events that are accepted as good events, and thus add to the noise of the final image. We show that this timing can be done in the FPGA, eliminating the need for the per-channel ASICs.

The figure of merit for timing pick-off is the distribution of the times stamps. In other words, for a setup with two detectors and a source exactly centered between them (so coincidental photons arrive at both detectors at the same time), what is the distribution of differences between the time stamps for each detector? Literature generally reports the full width at half maximum (FWHM) of the distribution histogram. For an ideal system, the difference between the time stamps would be zero, but noise in the system will introduce errors.

To determine the timing of a photon interacting with the scintillator, a timing pick-off circuit is used. A timing pick-off circuit assigns a time stamp to a certain feature of a pulse. For the pulse in Figure 3, this could be the time that the pulse started, or the time of the peak value, or the time it crossed a certain voltage. The two traditional techniques for timing pick-off are leading edge and constant fraction discriminators (CFD). Leading edge is simply determining when the pulse has crossed a certain fixed threshold voltage. This requires an analog circuit that detects the crossing. The drawback of this technique is that the time to reach the threshold is dependent on the amplitude of the pulse. This effect, known as time walk, gets worse as the trigger level is set higher. Since the threshold must be well above the noise margins to avoid false triggers caused by noise, leading edge suffers greatly from time walk.

Current state of the art timing pick-off for PET systems is performed with analog CFDs [5] because CFD automatically compensates for pulse amplitude variance. A CFD implements a circuit for equation 1.

$$h(t) = \delta(t - D) - CF \cdot \delta(t) \tag{1}$$

$\delta(t)$ is the incoming signal. Equation 1 is computed by splitting the analog pulse into two copies. One copy is attenuated by the constant CF (typically ~0.2), while the other copy is delayed by D and inverted. The two altered copies are added to produce a pulse with a zero crossing that can be detected and time stamped. This is shown in Figure 7. The zero crossing occurs at a constant fraction of the pulse amplitude for pulses with the same shape. Both CFD and leading edge must be done in dedicated ASICs, and require a circuit to convert the trigger to a time stamp.

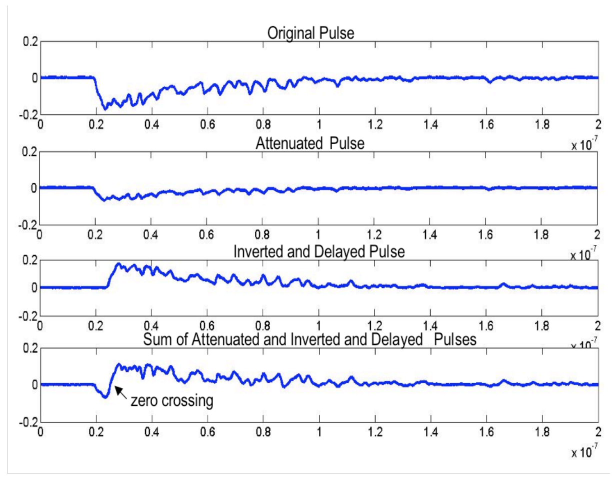

Figure 7. Steps of a constant fraction discriminator (CFD). The original analog pulse is split. One copy is attenuated while the other is inverted and delayed. The sum of these two copies creates a zero crossing that is a good point for pulse timing.

While CFDs can achieve sub-nanosecond timing resolution [5], FPGAs have made advancements in computing power and I/O sophistication that may allow them to achieve similar timing results. There have been previous efforts to perform the timing pickoff in the FPGA. One way is to utilize the increasing clock frequencies to perform a time-to-digital conversion [6]. This method still requires an analog comparator, and may be limited by the complexity of using fast clocks on FPGAs. Another method is to use signal processing to achieve precisions below the sampling time interval [11]. While this method is more complex, it has the advantage of using lower frequency components, which are cheaper, lower power, and make printed circuit board design simpler.

The pulses from the detectors have fairly uniform shapes. Using these known characteristics of the pulses to compute the start of the pulse is one method for achieving sub-sampling timing resolution. We assume that the rise and fall times (rise refers to the first part of the pulse and fall is the second part that decays back to zero, even though the "rise" on our photodetector is a drop in voltage) of the PMT/MAPD pulses are constants and the variability in the pulses is from the pulse amplitude and white noise as shown in (2).

$$V[n] = A\left(\exp^{\frac{-n*T_s}{\tau_R}} - \exp^{\frac{-n*T_s}{\tau_F}}\right) \qquad (2)$$

Based on these assumptions, the start time of the pulse can be determined by fitting an ideal pulse to the sampled pulse and using the ideal pulse to interpolate the starting point of the pulse.

In order to develop our timing algorithms on real data, we used a 25Gs/s oscilloscope to sample pulses from a PMT that was coupled to a LSO crystal. A 511 KeV (22Na) source was used to generate the pulses. The data from the oscilloscope was then imported into MATLAB to determine the parameters of (2) that gave the minimum sum of absolute errors for all of the pulses in the data set.

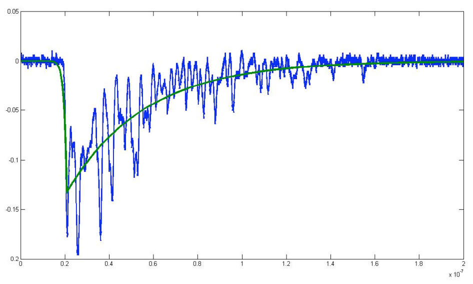

Figure 8: Sample pulse from a PMT coupled to an LSO scintillator, overlaid with the best least squares fit of a curve with exponential rise and fall.

Unfortunately, finding the minimum of the sum of absolute errors between the ideal reference pulse and the data samples requires the search of a large space that varies the amplitude (amplitude can vary by 2x) and start time of the reference pulse. The search space is too large to get good timing results while handling the count rate of the scanner (hundreds of thousands of events per second). Therefore, we developed a fitting method that doesn't require a search. To eliminate the search for amplitude, we discovered that there is a direct linear correlation between the area and amplitude of the pulse. This makes it simple to normalize the data pulse to the reference pulse. To normalize the data pulse, the pulse is multiplied by the ratio of the area of the reference pulse to the area of the data pulse.

To eliminate the time search, we created a lookup table that is indexed by the voltage of each sample as shown in Figure 9. We pre-calculate for each possible input voltage (for 10-bits of resolution), the time it occurs on the reference pulse in relation to the start of the reference pulse. The input voltage (v_n) then becomes the address to the lookup table and the data is the time elapsed from the start of the reference pulse (Δt_n). To use this table, the voltage of each sample from the detector pulse is converted to a Δt. The time stamp is composed of a coarse time (counter of 70MHz sampling clock) and the fine timing (Δt). To find the interpolated start of the pulse, Δt is subtracted from the corresponding coarse time.

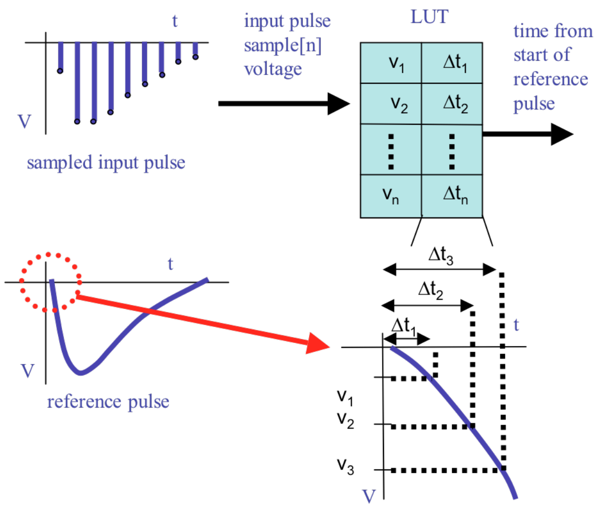

Figure 9. Voltage to time lookup method employed to eliminate searching for start time.

This was initially done for all samples of a detector pulse so each sample had a "vote" on where the start of the pulse was. Each of the estimated start times from all of the samples were then averaged to reach a consensus starting point. After a close inspection of the results from our look-up method, it became apparent that some of the sample points give much better results than others. This is shown in Figure 10, which plots the standard deviation of each of the samples of a pulse. In other words, all possible samplings (different starting points) of the 25Gs/s data was sampled at 70MHz. The inverse lookup was then applied to all of the first samples for the different sampling starting points and the standard deviation of the time stamps were calculated. This was repeated for all of the other samples. Notice that the deviation of the "voted" time stamp for each pulse is correlated with both the slope of filtered pulse, and distance from the pulse start. Points at the peak (points 4 and 5) have a low slope, and thus a small change in voltage results in a large time shift. The tail of the pulse also has a large deviation.

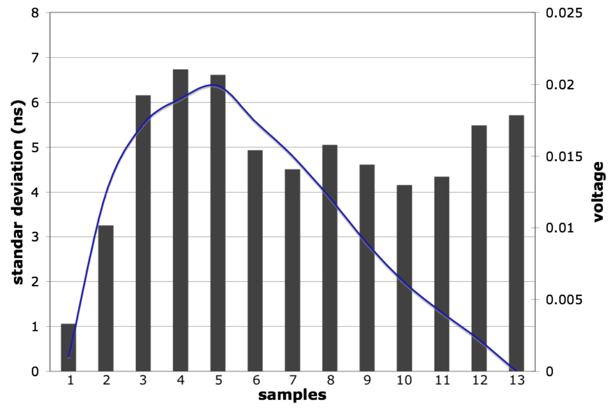

Figure 10. Plot of the standard deviation of the points of a filtered pulse that is sampled with a 70MHz ADC. The line is a filtered pulse (inverted), to give a reference for each point's position on the pulse.

Looking at Figure 8, this makes sense since the rise of the unfiltered pulse has much less noise than the rest of the pulse. Using this information, the look-up was changed to only use the first point above the baseline noise on the pulse. This is similar to a leading-edge detector, but automatically eliminates the effects of amplitude variation. It also has better noise immunity, since it can test very close to the signal start (where results are most accurate), while eliminating false positives by referencing back only from strong peaks.

The final timing algorithm uses one decay constant, and one rising constant, calculates the pulse amplitude from the area, and uses the voltage-to-time look-up for the first sample. The architecture of our final timing algorithm is shown in Figure 11.

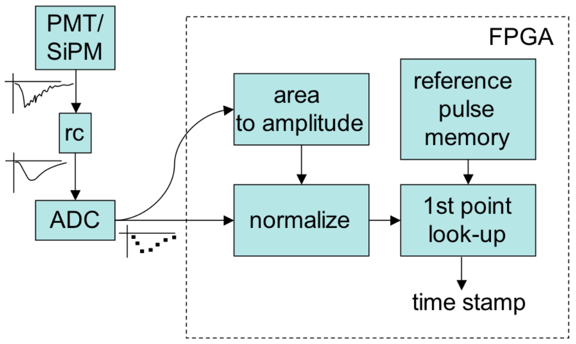

Figure 11. The architecture of the timing pick-off circuit implemented in the FPGA.

This timing algorithm has been implemented on a StratixII DSP Development board as shown in Figure 12. This board has two 12-bit ADCs (125MHz maximum sampling rate) so the detectors can be set up in coincidence to simulate two detectors on opposite sides of the scanner. A detector was attached to each ADC, which feeds into a timing circuit in the FPGA. The time stamp outputs of the two timing circuits are then compared and their difference is binned to form a histogram. The source was essentially halfway between the two detectors so in theory all coincidental time stamps should be identical. In reality though, there is a slight difference in the arrival times of the two pulses, but that just moves the center of the distribution of time stamp difference and does not affect the resolution. Since this experiment only used two detectors 180º from one another, the field of view determination is not required.

The design was place and routed using QuartusII 8.0. One timing core uses 250ALMs (out of 71,760), 114 kbits of memory (out of 9.4 Mbits), 1 PLL (out of 8), and 36 I/Os (out of 743). In the scanner, there will need to be more than one timing circuit though as there will be 64 channels supported by each FPGA. Each timing circuit can handle multiple channels, so the number of timing circuit instances will depend on the count rate of each channel.

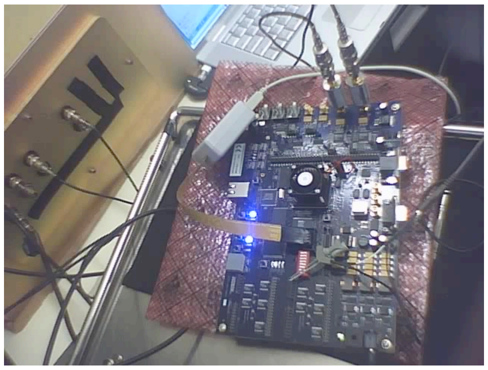

Figure 12. Timing test setup. The two ADCs on the Altera StratixII DSP development board were connected to two photodetectors that had a radioactive source placed between them (gold light-tight box).

Figure 13 gives the results for the timing experiment with the FPGA and real detectors for sampling rates of 60MHz and 125MHz. The same scintillator/detector setup was also tested with a CFD for comparison. To determine how the timing algorithm will improve as ADC technology advances, MATLAB simulation results for our timing algorithm are also included in Figure 13. To simulate the timing algorithm, data was collected from the setup in Figure 12 before the signal enters to the ADC. The data was imported into MATLAB and the timing algorithm was then run on the data for simulated sampling rates from 60MHz to 1GHz. This is reported in Figure 13 as simulated unfiltered. One issue with the experimental timing setup was the low-pass filter. In the experimental setup, only a 50MHz low pass filter was available, while in simulation a 10MHz cutoff produced the best timing results. As seen in Figure 3, these signals have a substantial amount of high frequency noise that needs to be filtered out. We believe that with a lower cutoff on the filter, the experimental results will improve. To test this, the same data was filtered again in MATLAB before timing. The result for this is shown in Figure 13 as simulated filtered. Notice that with appropriate filtering, our algorithm will achieve resolution similar to that of the CFD given a fast enough ADC. Given that the resolution of a CFD does not scale with technology (CFD performance has remained fairly constant over the last decade or more), our algorithm outperforms the CFD with a 500MHz ADC (available now in parallel ADCs, and expected soon in serial ADCs). Note that even in situations where our technique does not match CFDs in timing resolution, CFDs require per-channel custom logic, in fixed ASICs. Our all-digital processing avoids this cost, which is substantial in future PET scanners with 64 channels per FPGA. Another interesting result shown in Figure 13 is that we are able to achieve a timing resolution well below the ADC sampling rates.

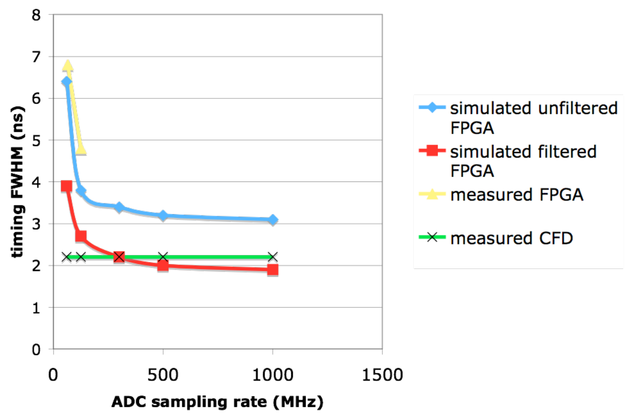

Figure 13. Timing results for simulated and experimental coincidental timing. Note that CFDs do not use ADCs, but the data is included to give a reference.

One issue with the current implementation is the use of a reference pulse that is unique to each photodetector and scintillator technology, and possibly each separate detector even if they are the same technology. For this implementation, the reference pulse was formed by capturing pulses with an oscilloscope just before the ADC and creating a reference pulse in MATLAB. While this works for experimentation, this would not be practical for a scanner with hundreds of photodetectors and thousands of scintillator crystals. To remedy this, we are currently developing a routine by which the FPGA can form its own reference pulse from the ADC sampled data. Periodically the FPGA would be reprogrammed with a self-tuning circuit, and a reference source put into the scanner. The FPGA would collect data to automatically tune the timing parameters to all detectors, circuits, and other variations through observation data. Then, the normal timing circuit could run with the tuned parameters for each sensor.

6. Interaction Localization

In addition to performing timing in the FPGA, we have developed a function to determine the location of the interaction in the detector. The better the localization, the more accurate the lines of response (LOR). Lines of response are the lines that are drawn between two interactions during image reconstruction. In other words, they are the tomography algorithms estimation of the lines of travel of the two photons from one event. Obviously, the better the LOR, the better the final image quality will be. Interaction localization is typically done offline in the processor before image reconstruction, but it is a computationally intensive step and adds a lot of time to the image reconstruction. Performing localization in the FPGA can reduce the image reconstruction time without affecting the final quality of the image.

One method currently being used to increase the resolution and the effective area of the sensor involves installing a large number of small scintillator crystals in a reflective matrix. This configuration, shown in Figure 14, is known as a discrete crystal array [9]. This causes the visible light photons to be constrained to a smaller area of the electronic sensor, making the determination of the location of the event more accurate than using the same algorithm in the continuous crystal. This also reduces the edge effect, allowing a larger useful imaging area of the electronic sensor.

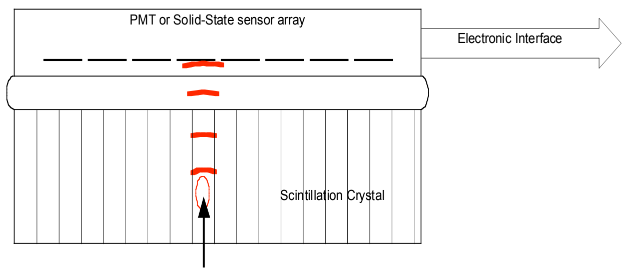

Figure 14. Concentrated output from a "discrete" scintillator crystal array.

Although the discrete array makes localization an easy problem, new problems arise. As the name suggests, this scintillator array consists of hundreds of sub-millimeter rectangular crystals arranged in a reflective grid. Each of these crystal slivers have to be cut to shape and polished, then placed by hand in the reflective matrix. This is a labor intensive and costly process requiring many hours per sensor for each of the sensor positions in the system. The fabrication of the sensors in this system constitutes a significant increase to the overall system costs. A secondary problem with the discrete array is that the reflective matrix occupies a part of the exposed area of the sensor without contributing to the scintillator area. This increases the likelihood that a photon will pass through the scintillator without interacting.

An alternative to discrete crystal arrays that is being researched in our lab is a continuous slab crystal [10]. This monolithic slab crystal has the benefit of cheaper production, but the interaction localization is more complicated. As can be seen in Figure 15 the light spreads out substantially from the point of interaction, and many of the sensors on the photodetector receive light.

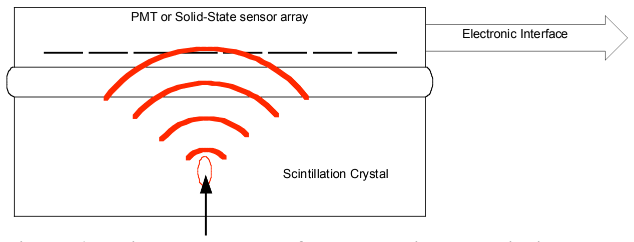

Figure 15. Dispersed output from a continuous scintillator crystal.

The photodetector in this study has and 8x8 array of sensors. For an event in the slab crystal, a distribution of light will result like the one shown in Figure 16. Notice that all of the 64 sensors receive some light, and there is an obvious peak, but achieving resolution beyond the 8x8 will require additional processing. Interactions in different locations of the crystal will produce different distributions. If the different distributions can be "recognized" through training, then the location of interaction may be inferred.

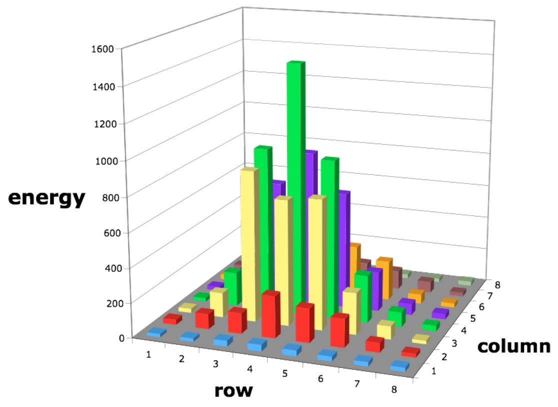

Figure 16. A representative event, showing the response of an 8x8 PMT array.

The method we have developed for increasing the resolution of the photon sensors is known as Statistics Based Positioning (SBP) [10]. SBP is able to improve the overall detection characteristics of a continuous crystal detector. In an SBP system, each sensor array requires an initial characterization step. This involves positioning a source in the field of view at a known X-Y location, collecting data from each of the 64 electronic sensors in the array, and saving the light response characteristics for each X-Y location in a table for future reference. The two statistical characteristics that are stored are the mean (μ) and variance (s2) of the light distribution function of each sensor for a given X-Y location. In other words, the same location of the crystal is hit with photons many times and the mean and variance of the energy deposited on each of the 64 sensors for all photon interactions is calculated and stored. This is done for 127x127 locations in the crystal. The result of this characterization is a 127x127 table that has 64 means and variances (one for each sensor in the 8x8 array) in each location. Later, when data is collected from an unknown location, it is compared with the previously collected data table using (3), and the coordinates of the characterization data that has the maximum value for (3) are taken to be the position of the unknown source. Using this method, position resolution much finer than the spacing of the individual detectors within the 8x8 arrays is achieved.

To calculate (3) in the FPGA, the following modifications were made for the equation inside the summation.

$$R = \frac{(E-\mu)^2}{2\cdot\sigma^2} + \ln(\sigma) = \left[(E-\mu)\cdot\frac{1}{\sqrt{2\cdot\sigma}}\right]^2 + \ln(\sigma)$$

let : $A = \mu$ $\qquad B = \frac{1}{\sqrt{2\cdot\sigma}}$ $\qquad C = \ln(\sigma)$

$$R = [(E-A)\cdot B]^2 + C$$

In this simplification the values for A, B and C are precalculated and stored in memory. To further simplify this, the likelihood equation was also converted to fixed point. An analysis of the

$$\ln(P(\mathrm{event}, X, Y)) = -\sum_{\mathrm{row}=1}^{8}\sum_{\mathrm{col}=1}^{8}\left[\frac{\left\lceil \mathrm{event}_{\mathrm{row,col}} - \left(\mu_{X,Y}\right)_{\mathrm{row,col}}\right\rceil^{2}}{2\cdot\left\lceil\left(\sigma_{X,Y}\right)_{\mathrm{row,col}}\right\rceil^{2}} + \ln\left\lceil\left(\sigma_{X,Y}\right)_{\mathrm{row,col}}\right\rceil\right] \tag{3}$$

algorithm indicated that the memories for A, B, and C need to be 16 bits while the data path required 24 bits.

The first implementations of SBP used an exhaustive search to find the best statistical fit of the data for each unknown point. In the exhaustive search, the likelihood function is computed for every potential location in the characterization data set. This resulted in a simple algorithm that was always able to find an optimum solution but put a heavy load on processing resources. The solution for each event's possible position requires 8x8 statistical computations, each containing approximately 6 floating-point operations. The desired 127x127 possible positions for each sensor results in 6x8x8x127x127 ≈ 6 Million Floating Point Operations (MFLOP) per detected event, for a PET scan that may contain millions of events. This resulted in a computation time that was unrealistic.

In an effort to shorten the computation time, a method termed "Localized Exhaustive" was employed. In this method, it is assumed that the solution is near the location of the detector channel with the maximum response. In the 8x8 solution to the likelihood equation it was found that the channels with low values added little to the solution. A solution that ignores the channels that are far from the channel with the peak energy yields results that do not introduce any significant errors to the solution set (this result will be shown further below). However, solving and summing only the values closest to the peak could significantly reduce the processing load. Experiments showed that using only the 21 values closest to the peak, as shown in Figure 17, produced for results that closely matched those found with the exhaustive solution. Recall that these 21 values are calculated for each of the 127x127 locations on the crystal.

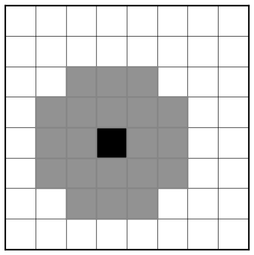
Figure 17. Likelihood function using only 21 channels adjacent to the peak energy.

This reduction in processing is not adequate to achieve real time SBP processing in the FPGA. The search of all 127x127 positions is too much processing to complete in the time required for the desired count rate of the scanner. Fortunately, the solution space is convex and thus lends itself to a smarter search. The algorithm that we implemented is a modified hierarchal search.

The hierarchal search starts with sampling 9 points at multiple locations (see Figure 18), uniformly scattered across the solution space. The sample with the lowest value is assumed to be closer to the final solution than any point with a higher value. The lowest valued sampled point is then used as the center of a new solution space that is ¼ the size of the previous solution space. When the size of the solution space is reduced to one, the solution has been reached.

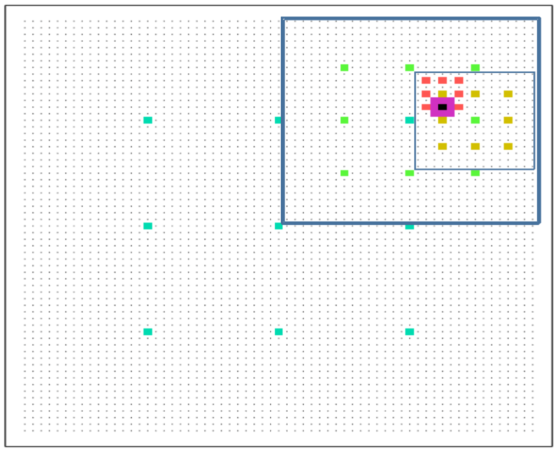
z,Z

Figure 18. Five iterations of a 3x3 search on a 63x63 solution space.

The regularity of this search allows each level of the search to consist of an almost identical process, where each iteration consists of a center point for the search and the spacing between points to be tested at this iteration. For the 3x3 search shown, 9 sets of characterization data table values are needed for the first stage, one for each position to be tested. The next iteration will consist of potential sample positions at one-half the spacing, with sets of points centered on each of the previous stage's points, with the first stage defaulting to being centered on the center of the entire data table. This results in each stages' data table having approximately four times the number of testable locations as the previous stage, but still only a fraction of the overall table, until the last stage is reached. The last stage is the only stage that needs access to the entire table. Each iteration will begin with the best solution from the previous stage as the center point of its search and will test all points adjacent to it at the current resolution. As each stage passes its best result to the next stage, eventually the last stage is reached where the spacing between adjacent pixels is one. The results of this stage represent the best solution available for the data table used.

Structuring the search in this way allows points sampled at each iteration to be independent of calculations currently underway in the other iterations. This allows pipelining of the calculations. Different events can be underway at each of the stages simultaneously. A second advantage of this method is that the storage requirement for each stage's data table is reduced for each iteration prior to the last.

This algorithm was also implemented on the StratixII DSP board but because the memory available in the prototype is not sufficient for a seven-stage system, only four stages could be created and tested. The results of this system were compared, on an operation-by-operation basis with the results of an integer-based simulation written in "C". The prototype was found to properly compute Xc and Yc locations at each stage and transfer this information to the next stage. Each stage requires 21*9+3=192 clock cycles. This represents a 21-cell solution for each of 9 X-Y locations at each stage, plus 3 cycles for final calculation of the coordinates of the minimum value and transferring the results to the next stage. Although each event will take 192 clocks at each of seven stages, for a total of 1344 clock cycles, as each stage completes its calculation it is immediately available for the next sequential event. This results in 7 events being solved simultaneously, with each event at a different stage of its solution. This allows the system throughput to be based on the 192 clock-cycle count, well below the 300 clock-cycle target to meet the incoming event rate. The implementation utilizes only 3.6% of the StratixII S180 and 35.9% of the memory (stages 1-5). Additionally, the results were only on average, .01mm worse in resolution than the software version which achieve a resolution of ~1.4mm [10]. This is below the sensor pitch of ~ 2.5mm.

7. Conclusion

PET is an application well suited to FPGAs. Certainly, FPGAs are ideal as we develop algorithms such as digital timing, and interaction localization, but they also provide most of the pieces needed for an advanced data acquisition and processing system for PET. We use the sophisticated I/O to interface with fast serial ADCs, which allows us to process more channels. We can utilize microprocessor for control and interfacing with the host computer, the DSPs for signal processing, the memories for reference pulse and characterization tables, and the reconfigurability for tuning and algorithm development.

This paper has presented application for our new generation of PET scanner, which leverages modern FPGAs in very aggressive ways. In this work we show proof of concept of two algorithms on a development board as we are currently engineering a printed circuit board for a full small animal PET scanner. We demonstrate a new, all-digital timing pickoff mechanism that demonstrates better timing resolution than current state-of-the-art approaches when coupled with current and future ADC technologies. We also present the conversion of a software based localization technique to the FPGA. These demonstrate how many of the features of modern FPGAs can be harnessed to support a complete, complex signal processing system in an important electronics domain.

Acknowledgements

This work is supported by Altera, NIBIB R21/R33-EB0001563, NIH grant EB002117, NSF, and Zecotek.

References

[1] http://www.mindready.com/eng/index.asp
[2] T. K. Lewellen, J. Karp, *Emission Tomography,* San Diego: Elsevier Inc., 2004, pp.180.
[3] C. M. Laymon et al., "Simplified FPGA-Based Data Acquisition System for PET," *IEEE Trans. Nuclear Science,* vol. 50, no. 5, 2003, pp. 1483-1486.
[4] J. Imrek et al., "Development of an FPGA-Based Data Acquisition Module for Small Animal PET," *IEEE Trans. Nuclear Science,* vol. 53, no. 5, 2006, pp. 2698-2703.
[5] W. W. Moses, M. Ullish, "Factors Influencing Timing Resolution in a Commercial LSO PET Scanner," *IEEE Trans. Nuclear Science,* vol. 43, no. 1, 2006, p. 78-85.
[6] M. D. Fries, J. J. Williams, "High-Precision TDC in an FPGA using a 192-MHz Quadrature Clock," *IEEE Nuclear Science Symp. Conf. Recond,* vol. 1, 2002, pp. 580-584.
[7] A. Alessio, unpublished presentation.
[8] A. B. Brill, R. N. Beck, *Emission Tomography,* San Diego: Elsevier Inc., 2004, pp.25.
[9] Kisung Lee et al., "Detector characteristics of the micro crystal element scanner (MiCES)," *Nuclear Science, IEEE Transactions on*, vol. 52, 2005, pp. 1428-1433.
[10] J. Joung et al., "cMiCE:a high resolution animal PET using continuous LSO with a statistics based positioning scheme," *Nuclear Science Symposium Conference Record, 2001 IEEE*, 2001, vol.2, pp. 1137-1143.
[11] R. Fontaine et al., "Timing Improvement by Low-Pass Filterering and Linear Interpolation for the LabPET Scanner", *IEEE Trans. Nuclear Science,* 2008, vol. 55, no. 1, pp. 34-39.