

FPGA-Based Data Acquisition System for a Positron Emission Tomography (PET) Scanner

Michael Haselman¹, Robert Miyaoka², Thomas K. Lewellen², Scott Hauck¹

¹Department of Electrical Engineering, ²Department of Radiology
University of Washington, Seattle, WA

{haselman, hauck}@ee.washington.edu, {tkldog, rmiyaoka}@u.washington.edu

Abstract: Modern Field Programmable Gate Arrays (FPGAs) are capable of performing complex discrete signal processing algorithms with clock rates of above 100MHz. This combined with FPGAs low expense, ease of use, and selected dedicated hardware make them an ideal technology for a data acquisition system for positron emission tomography (PET) scanner. Our laboratory is producing a high-resolution, small-animal PET scanner that utilizes FPGAs as the core of the front-end electronics. While this scanner uses an Altera ACEX1k and has limited complexity, we are also developing a new set of front-end electronics based on an Altera StratixII. This next generation scanner utilizes many of the features of modern FPGAs to add significant signal processing to produce higher resolution images. One such process we discuss is sub-clock rate pulse timing. 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.

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 that is able to capture certain metabolic activities inside a living body. Positron emission tomography (PET) is in this class of medical imaging.

Many current PET systems already utilize FPGAs for data acquisition [3,4] because they provide the necessary speed to process the incoming data without the complexity of an application-specific circuit. Most current scanners use the FPGAs to collect data and do some simple filtering. This paper discusses our use of FPGAs in our current PET scanner as well as work on our next generation scanner which better utilizes the capabilities of modern FPGAs to help produce higher resolution images.

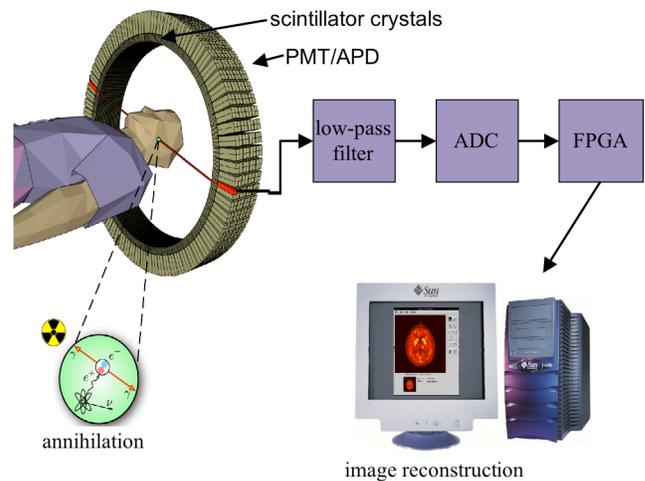


Figure 1. Drawing of a PET scanner ring and attached electronics.

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 and 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 is positron emission. The positron 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 photomultiplier tubes (PMT) or avalanche photo diodes (APD). The scintillator converts the 511KeV photon into many visible light photons, while the PMT or APD generate an electrical pulse from the visible light. 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 create the radiopharmaceutical (the 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 fluorodeoxyglucose ([F-18]FDG). [F-18]FDG is an analog of glucose that has a half-life of 110 minutes (plutonium-239 has a half life of 88 years). [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. Thus the radioactive portion of the molecule is trapped in the 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.

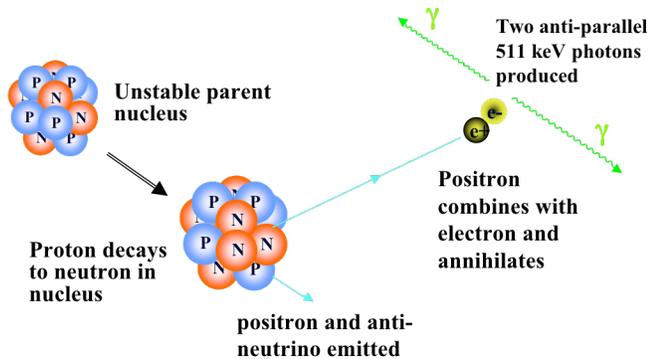


Figure 2. Physics of a positron decay and electron annihilation [2].

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 $180^\circ \pm .23^\circ$ from one another.

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 visible

light. Scintillators can be made up of many different materials, including plastics, organic and inorganic crystals, and organic liquids. Each type of scintillator has a different density, index of refraction, timing characteristics, and wavelength of maximum emission. The density determines how well the material stops the photons. The index of refraction of the crystal and the wavelength of the light affect how easily light can be collected from the crystal. The wavelength also needs to be matched with the devices that will turn the light into an electrical pulse in order to increase the efficiency. Finally, the timing characteristics is related to how long it takes for the visible light to reach its maximum output (rise time) and how long it takes for the light pulse to decay (decay time). This is important because the longer the sum (the sum is dominated by the decay time) of these two times are, the lower the number of events a scintillator can handle per given time unit, and thus the longer the scan will take to get the same number of counts. This is because the longer the sum, the greater the likelihood that two events will overlap (pile-up) and the data will be lost. The scintillator that we use in our experiments is $\text{Lu}_2\text{SiO}_5(\text{Ce})$, or LSO, which is an inorganic crystal. It has a reported rise constant of 30ps and a decay constant of 40ns [2]. These documented times can vary slightly due to the geometry of the crystal and the electronics that are attached to it. LSO is a newer scintillator material that exhibits fast response and good light output.

2.4 Photomultiplier Tubes

Attached to the scintillator are electronic devices that convert the visible light photons into electronic pulses. The two most commonly used devices are photomultiplier tubes (PMT) and avalanche photo diodes (APD). The PMT is a vacuum tube with a photocathode, several dynodes, and an anode that has high gains to allow very low levels of light to be detected. APDs are the semiconductor version of the PMT. An alternate technology that is currently being studied for use in PET scanners is silicon photomultipliers (SiPM). SiPMs are an array of semiconducting photodiodes 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 10^3 diodes per mm^2 . 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 these devices a linear output even though they are made up of digital devices.

Our system (discussed in the next section) currently uses a PMT that has 12 channels: six in the 'x' direction and six in the 'y' direction, as depicted in Figure 3. The separate channels allow for determining the location of an event. For example, if an event occurred in the upper left hand corner of Figure 3, then channels Y1 and X1 will have a

large signal, with progressively smaller signals at each successively larger channel number. Channels Y6 and X6 will have virtually no signal.

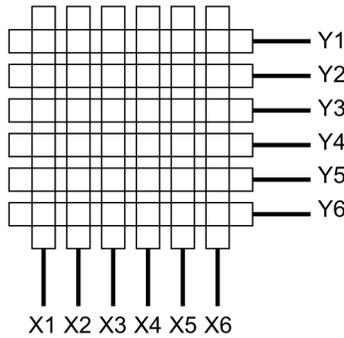


Figure 3. Illustration of the output channels of a 12-channel PMT, with 6 channels in the ‘x’ direction and 6 in the ‘y’ direction.

2.5 Front-end electronics

The analog pulses arising from the PMTs contain the information used 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 to remove some of the noise (shown in Figure 4). This is done with a low pass filter. 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 handle. The solution is to use serial output ADC, which also limits the sampling rate to around 100MSPS.

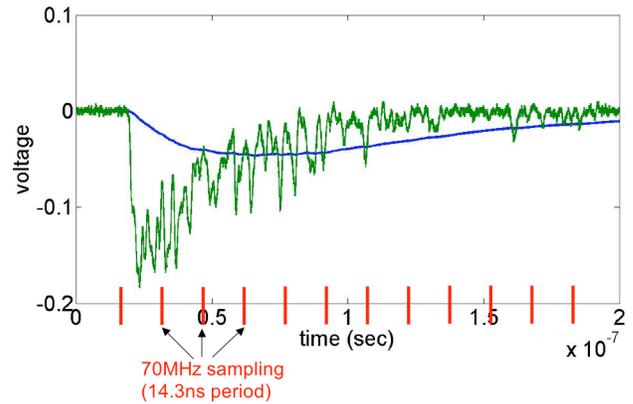


Figure 4. MATLAB plot of a pulse from a LSO scintillator crystal attached to a PMT. A simulated low-pass filter with a 33.3MHz cutoff is also shown. The wide lines on the x-axis indicate the interval at which a 70MHz ADC would sample.

Now that the data is digitized, the needed parameters can be extracted in the FPGA. The easiest data point to extract is the total pulse energy. It is obtained by simply summing up the samples of the pulse values and subtracting out the baseline (the output value of the ADC without an input pulse).

The start time of the pulse is important for determining coincidence pairs (timing will be discussed extensively in section 3). 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 scintillator crystal does not detect them. 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 5.

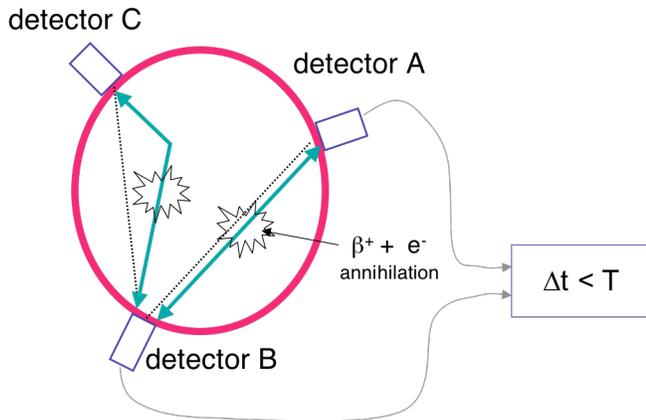


Figure 5. Coincidence detection in the PET scanner. If events are detected in detectors A and B within a certain time period, then the events are saved as coincidental pairs and are considered to have arisen from the same event. If events are detected in detectors B and C, then the event is ignored because 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 all of the events are separated into parallel lines of response (interpreted path of photon pair). These lines can then be used to create a 3-D image using computer tomography.

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.



Figure 6. 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).

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 with the tracer uptake 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). An interesting application in neurology is the early diagnosis of Parkinson's disease. Tracers have been developed that concentrate in the cells in the brain that produce dopamine [2], a neurotransmitter. In patients with Parkinson's disease, neurons that produce dopamine reduce in number. So, a scan of a Parkinson's patient would have less activity than a healthy patient. This can lead to early diagnosis, since many of the other early signs of Parkinson's are similar to other diseases. It is also a useful to see if therapies are progressing. If a therapy results in more cells that produce dopamine, then there will subsequently be more activity than previous scans for the same patient.

3. Current System (MiCEs)

We are developing a high-resolution small-animal PET scanner [3]. Small animal scanners serve two purposes in research. The first is that they serve as pre-clinical imaging systems that allow the investigation of new drugs and therapies in small animals (i.e., mice) before they move on to human trials. Second, because the imaging systems are much smaller than is needed for humans, new detector technologies can be investigated at a reduced expense compared to constructing a human system.

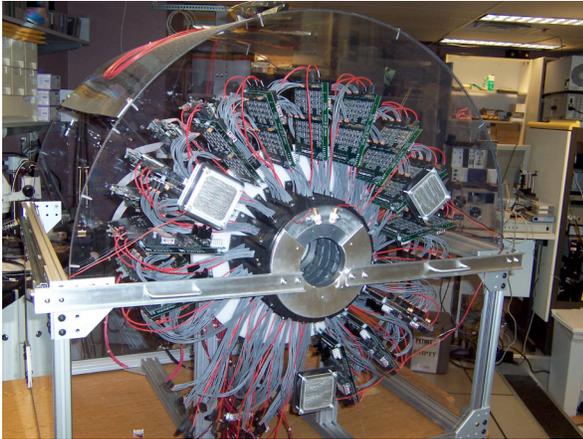


Figure 7. Our current small-animal PET scanner.

Our scanner, shown in Figure 7, utilizes PMTs for the detection technology and older generation FPGAs for the front-end data acquisition. The front-end electronics use a board from CTI PET Systems, Inc. to process the pulses. The scanner consists of a ring of 18 detector cassettes, with each cassette consisting of four scintillator arrays attached to four PMTs. Each detector set is connected to its own set of front-end electronics, as shown in Figure 8. The 18 different boards each have two “nodes” (a microprocessor FPGA pair), where each node supports two PMTs. All 36 nodes are daisy chained together by FireWire (IEEE 1394a) to create one connection to the host computer, which contains software that collects data from all of the nodes. After all of the data is acquired, it is processed off-line to produce an image.

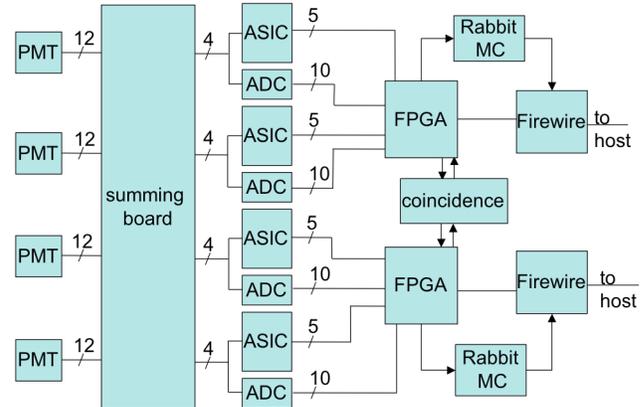


Figure 8. Architecture of our current small-animal, high-resolution PET scanner.

As shown in Figure 8, there are many discrete parts to the front-end electronics. The PMTs are the sensors that turn the light into analog pulses (see section 2.4). Each PMT outputs twelve signals, six for the x direction and six for the y directions. To reduce I/O counts, the twelve signals are reduced to four, by a summing board. The ASICs implement an algorithm to determine the time of the event. The four signals are then digitized by 62.5MHz ADCs, and then sent to the FPGAs. In addition to the FPGA, there is a Rabbit microcontroller and FireWire physical layer chip for communication to the host computer.

The FPGA and Rabbit microprocessor play a central role in the data acquisition. The microprocessor is used for general control of each individual node. This includes configuring the FPGAs, communicating with the host computer, initializing the system, and working with the FPGA to tune the system. While the microprocessors handle the control of the system, the FPGAs make up the bulk of the datapath. During normal operation, the FPGAs have two main tasks: pulse processing and data packing for the FireWire system. The pulse processing for the current generation scanner is fairly trivial. The first step is to determine if a given event is in coincidence with another event on the other side of the scanner. If a matching coincidence event is detected, the pulse is integrated to determine the energy, and a coarse resolution time stamp is put on the data. The energy and time (coarse grain from the FPGA and fine grain from the ASIC) are put in a packet to be sent to the host computer over FireWire.

Occasionally, the scanner needs to be tuned to set the amplifier gains in the ASICs. This is required because the crystals have different light output and light collection efficiencies and the PMTs all have different gain characteristics. The tuning normalizes the differences between the sensors and corrects for drift over time. To tune the system, the microprocessor reconfigures the FPGAs with the tuning algorithm and initializes them to bin up the energy values for each pulse. Unfortunately, the periphery of the crystal array and PMT has edge effects that

can introduce errors into the tuning algorithm, so they must be ignored. In order to filter out the events that hit the edge of the crystal array, the position of the event must be decoded. The four signals that come from the summing board (see Figure 8) contain enough information to determine the position of the event in the crystal array. The four signals, x^+ , x^- , y^+ and y^- , indicate the position by the energy distribution.

$$x = \frac{x^+}{(x^+) + (x^-)} * 256, \quad (1)$$

$$y = \frac{y^+}{(y^+) + (y^-)} * 256, \quad (2)$$

By computing the result of equations 1 and 2, the x and y location of the event in the crystal array can be differentiated into a 256 by 256 array. Once the event is determined to be in the interesting area of the array, the total energy of the event is calculated and it is placed into 256 different energy bins. This produces an energy histogram as shown in Figure 9. The peak at about 90 is called the photo peak because it represents the energy when all of the energy from the 511KeV photon is deposited into the scintillator crystal. Counts below the photopeak represent scattered photons, where either only part of the energy of the photon is deposited in the crystal or the photon Compton scatters in the object being imaged before it reaches the crystal.

If the system is tuned, the photo peaks should line up for all of the detectors. If the peaks have some variations, the microprocessor changes the gains and reruns the tuning algorithm. This is all automated, so once the operator instructs the machine to tune itself, the microprocessor will instruct the FPGA to run the tuning algorithm. Once the FPGA signals that the routine has completed, the microprocessor reads the histogram out of memory and locates the photo peak. If the photo peak is shifted, the microprocessor adjusts the gains in the ASIC and iterates until the photo peaks for all of the sensors line up.

4. Next Generation Scanner

While the current design (tuning and scanning circuits) occupies 92% of the logic elements of the Altera ACEX1k100, it only occupies 6% of the much more modern Stratix II EPS60. Obviously, we have a lot more logic to utilize in our next generation scanner. We are currently developing algorithms and designing prototyping boards for our next generation front-end electronics that will make use of newer generation FPGAs to integrate many of the parts. We are also doing more advanced signal processing on the pulses to achieve greater resolution of the images.

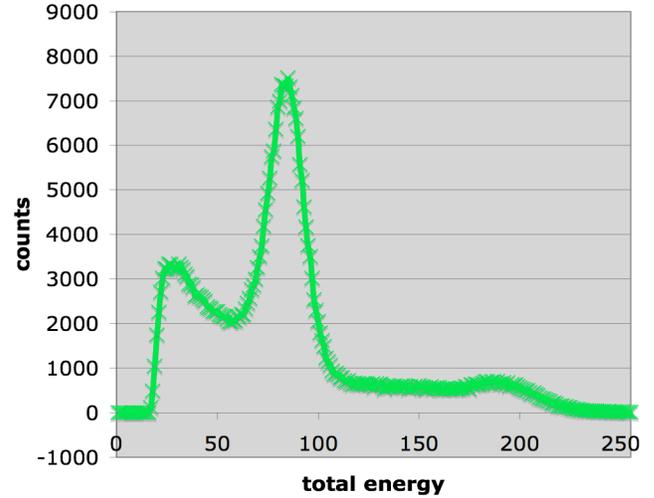


Figure 9. The output from the energy tuning routine produces this energy histogram. This plot is a histogram of the energy, so the y values represent the number of events captured that had certain energy. For example, at the energy of 150, the routine recorded about 800 events.

The greatest difference between our current generation scanner and the next generation is the number of channels that will be used per sensor. This is mostly because we are transitioning to the solid state SiPM (see section 2.4), which has one output per crystal. For the crystal array for our current system, that would be 484 outputs per crystal array, instead of 12 from the PMT. The 484 outputs will be reduced to approximately 44 by row/column summing, but this still represents many more channels (i.e. 44x72 for the same detector geometry) that will have to be digitized. This requires the use of serial ADCs (we are using a 70MHz serial ADC), which can easily communicate with the Stratix II using the dedicated serializer/deserializer and phase lock loops. There are other parts of a modern FPGA that make them perfectly suited to our application. Foremost, we can replace the Rabbit microprocessor with a Nios II embedded processor and eliminate the slow communications between the FPGA and the microprocessor. A FireWire core from Mindready [1] eliminates the need for the TI FireWire chip, as the link layer will now be in the reconfigurable fabric of the FPGA and the transactional layer can be implemented in the Nios II processor. Finally, the additional signal processing requires the use of the dedicated memories and multipliers. As can be seen in Figure 10 (as compared to Figure 8), the architecture of this scanner is more compact due to the integration of parts into the FPGA.

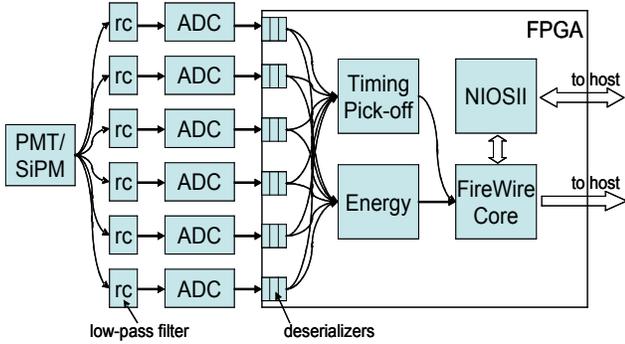


Figure 10. Front-end electronics for the next generation scanner in an Altera Stratix II FPGA.

5. Timing

Another 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 believe that this timing should be done in the FPGA with sampled data, 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 with a source exactly centered between them (so photons arrive both detectors at the same time), what is the distribution of differences between the time stamps for each detector? For an ideal system, the difference between the time stamps would be zero, but the noise in the system will introduce errors. To simulate this, time stamps were calculated for many different samplings of the same pulse. The time stamps for each pulse were then compared to the time stamps of all other 18 pulses to produce a distribution like the one in Figure 15.

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 4, 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.

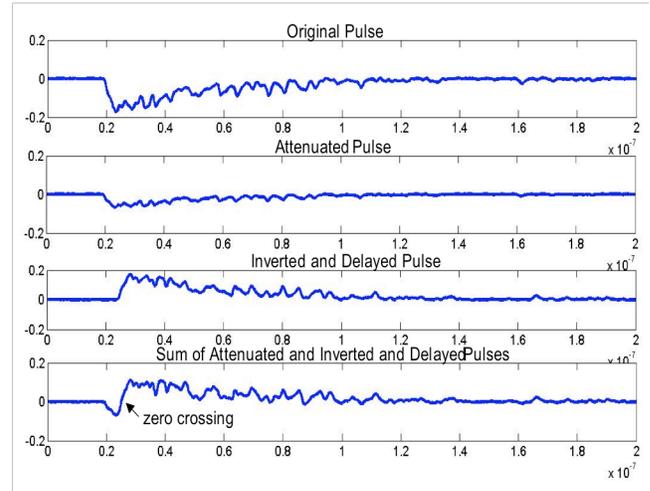


Figure 11. Step of a constant fraction discriminator (CFD). The original analog pulse is split. One copy is attenuated while the other is inverted and delayed. The sum of these two copies creates a zero crossing that is a good point for pulse timing.

Current state of the art timing pick-off for PET systems is performed with analog CFDs [5] because they are immune to pulse amplitude variance. A CFD implements a circuit for equation 3.

$$h(t) = \delta(t - D) - CF \cdot \delta(t) \quad (3)$$

$\delta(t)$ is the incoming signal. Equation 3 is computed by splitting the analog pulse into two copies, and delaying one copy by D . The other copy is inverted and attenuated by the constant CF (typically ~ 0.2). Finally, 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 11. 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.

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 [6]. 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.

Using the known characteristics of 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 crystals is a drop in voltage) of the PMT pulses are constants and the variability in the pulses is from the pulse amplitude and white noise. For LSO, the rise time is dominated by the response of the PMT, while the decay time is a function of the scintillation crystal. 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 test our timing algorithms on real data, we used a 25Gs/s oscilloscope to sample 19 pulses from a PMT that was coupled to a LSO crystal. A 511 KeV (^{22}Na) source was used to generate the pulses. The data from the oscilloscope was then imported into MATLAB. We have chosen to start with simulations in MATLAB for many reasons. Simulations allow the low-pass filter, ADC sampling rate, and fitting algorithms to be quickly investigated before we commit to an implementation.

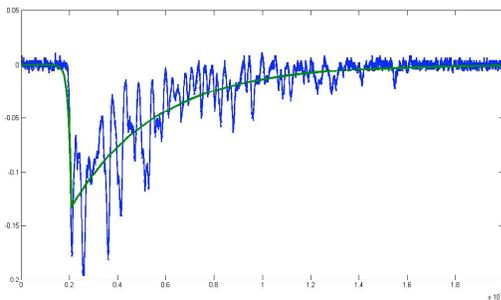


Figure 12: Sample pulse from a PMT coupled to an LSO scintillator, overlaid with the best least squares fit of a curve with exponential rise and fall.

We hypothesized that if we created a pulse with two exponentials (one for the rising edge and one for the falling edge) and found the amplitude, time shift, decaying exponential and rising exponentials that produced the best least squares fit, we could use that ideal pulse to interpolate the starting point of the pulse. Figure 12 shows an example plot of this method. Using this “brute force” method, the standard deviation of the timing pick-off was 1.0ns with a 70MHz ADC. While this is good timing resolution, the search space is far too large for an FPGA to compute in real time. From the “brute force” method, we found that the rise time ranged from .1-.5ns, the decay times ranged from 28-38ns, and the amplitude ranged from .082-.185V. To cover these ranges for a reasonable time step (~40ps) would require the least squares fit to be calculated and compared at least 215,000 times for each pulse (11 decay time steps, 5 rise time steps, 11 amplitude steps and 357 time steps).

Obviously, we must make some compromises to create an efficient FPGA-based algorithm. The first compromise was to use one rise and fall time constant. The rise and decay times that gave the best overall least squares fit for all unfiltered, unsampled data were 310ps and 34.5ns. With fixed rise and fall times, the “brute force” method timing resolution degrades to 1.1ns. However, even after eliminating the time constant searches, almost 4,000 searches would still be required for each event.

To eliminate the search for amplitude, we discovered that there is a direct correlation between the area and amplitude of the pulse, as shown in Figure 13. The function to convert area to amplitude was determined by sampling each of the 19 pulses with many different starting points, and correlating the area obtained for each sampling to the known amplitude for that complete pulse. Using this estimation, the timing resolution is degraded to 1.2ns.

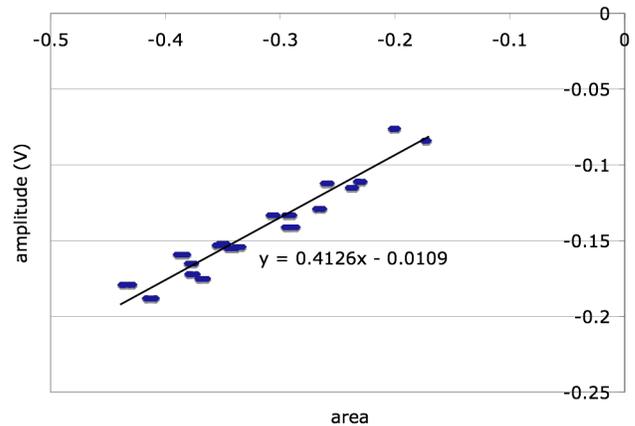


Figure 13. Plot of the area of sampled and filtered pulses versus the amplitude of the pulses. The best linear fit is used to estimate the pulse amplitude from the area of the pulse.

Most dimensions of the brute-force search have been eliminated with a loss of only 20% timing resolution, but this would still require 357 searches for each possible timing offset. However, given that we are fitting the data to a reference curve with known rise and fall times, and computed amplitude, we can convert the brute force search to a reverse-lookup. We pre-calculate for each possible input voltage the time it occurs on the reference pulse. Thus, each incoming voltage can be converted to a timing offset with a simple memory operation. This is done for each pulse so that after the lookup, each of the 13 points (for 70MHz sampling rate) has a time at which it “thinks” the pulse started. If these times are averaged, the timing resolution degrades significantly to 2.84ns.

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 14, which plots the standard deviation of each of the 13 samples. Notice that the deviation is correlated with 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. Notice also that if only the first point is used, the standard deviation is 1.03ns, which essentially equals the “brute force” method.

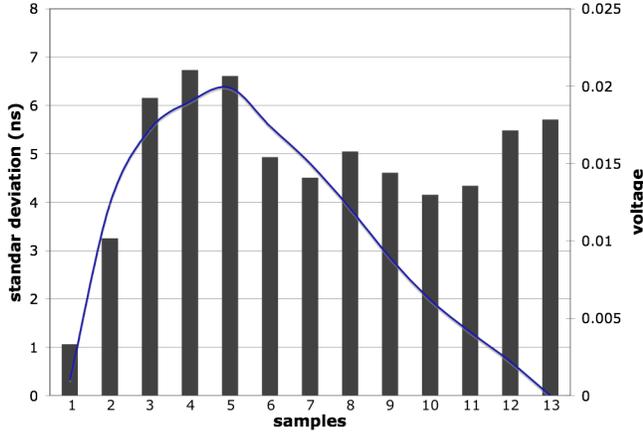


Figure 14. Plot of the standard deviation of the points of a filtered pulse that is sampled with a 70MHz ADC. The line is a filtered pulse (also inverted), to give a reference for each point’s position on the pulse.

Looking at Figure 12, 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 .005V 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.

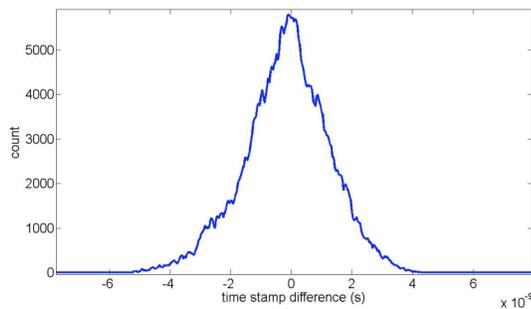


Figure 15. The distribution of difference of time stamps between two pulses for a sampling rate of 70MHz.

The final timing algorithm uses one decay constant, one rising constant, calculates the pulse amplitude from the area, and uses the voltage-to-time look-up for the first sample. It has a standard deviation of 1.03ns. The distribution of the final algorithm is shown in Figure 15 for a 70MHz ADC. The architecture of our final timing algorithm is shown in Figure 16.

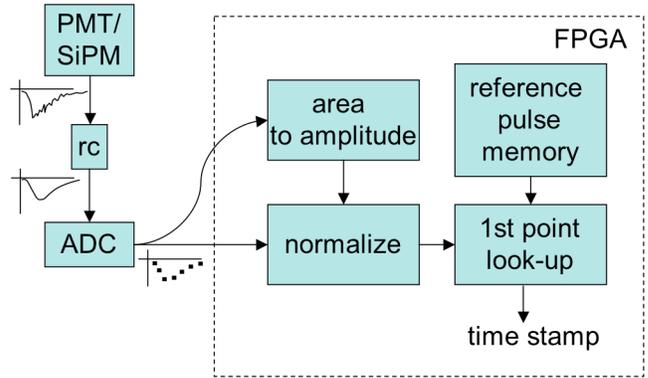


Figure 16. The architecture of the timing pick-off circuit implemented in the FPGA.

With our method, we have standard deviation of 1.03ns using a 70MHz ADC and a 10MHz cutoff low-pass filter, as compared to 390ps for a simulated CFD on the same data. While our current design with a 70MHz ADC has a lower timing resolution than an analog CFD, Table I shows that as ADC technologies improve, the timing resolution will improve. 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 128 channels per FPGA. Another interesting result shown in Table I is that we are able to achieve a timing resolution well below the ADC sampling rates.

Table I. Standard deviations (ns) of the distributions of time stamps for different low-pass filter cutoffs and sampling rates.

| ADC rate (MHz) | RC cutoff (MHz) | | |
|----------------|-----------------|-------|-------|
| | 33.3 | 16.7 | 10 |
| 70 | 1.19 | 1.06 | 1.03 |
| 140 | 0.78 | 0.789 | 0.837 |
| 300 | 0.455 | 0.543 | 0.591 |
| 500 | 0.322 | 0.275 | 0.409 |
| 1000 | 0.198 | 0.196 | 0.198 |

6. Conclusion

PET is an application well suited to FPGAs. Certainly, FPGAs are ideal as we develop algorithms such as digital timing, but they also provide most of the pieces needed for an advanced data acquisition and processing system for PET. We have utilized the reconfigurability to develop a tuning algorithm that, under the control of the microprocessor, can adjust gains and set registers to accommodate for variances in different parts of the

scanner. We use the sophisticated I/O to interface with fast serial ADCs, which allows us to process more channels. These channels can also be processed in parallel in the reconfigurable fabric, which increases the count rate that the scanner can handle. The increase in computing power of modern FPGAs over the earlier generation allows us to implement timing in the FPGA and eliminate the ASICs. Finally, we implement complicated control logic such as the FireWire transaction layer in the embedded microprocessor.

This paper has presented our work on our current generation PET scanner, which uses old FPGAs to implement timing and basic signal processing tasks. It also introduces concepts from our new generation of PET scanner, which leverages more modern FPGAs in very aggressive ways. 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 show 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 Zecotek, Altera, NSF, and NIH grant EB002117.

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.