

Digital Pulse Timing in FPGAs for Positron Emission Tomography.

M.D. Haselman¹, Member IEEE, S. Hauck¹, Senior Member IEEE, T.K. Lewellen², Fellow IEEE,
R.S. Miyaoka², Member IEEE,

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

² University of Washington Department of Radiology, Seattle, WA.

Abstract: Modern Field Programmable Gate Arrays (FPGAs) are capable of performing complex digital signal processing computations with clock rates of above 100MHz. This, combined with their low cost and ease of use, make them an ideal technology for pulse timing, and are a central part of our next generation of electronics for our pre-clinical PET scanner systems. To that end, our laboratory has been developing a pulse timing technique that is based on a digital leading edge discriminator to achieve timing resolution well below the sampling period of the analog to digital converter (ADC). We report our results for timing simulations and initial FPGA implementation.

1. Introduction

Timing pickoff for PET systems is generally performed with analog constant fraction discriminators (CFDs) [1]. While CFDs can achieve sub-nanosecond timing resolution, FPGAs have made advancements in computing power and I/O sophistication that may allow them to achieve similar timing results. Many current PET systems already utilize FPGAs for data acquisition [4, 5], so it is logical to employ the same chips to compute the timing pickoff. This approach is part of the next generation of electronics for our pre-clinical PET scanner development.

There have been previous efforts to perform the timing pickoff in FPGAs. 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. Lower frequency components are cheaper, lower power, and make circuit board design simpler. Also, a constraint for our application is keeping the operating frequency away from the proton resonance frequencies in a 3T MR scanner. Finally, because microprocessor clock rates won't continue to dramatically increase, relying on faster clocks may not be as beneficial as relying on increasing computational ability.

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) of the PMT pulses are constants and the variability in the pulses is from the pulse amplitude and white noise. The rise time is dominated by the response of the PMT, while the decay time is a function of the scintillation crystal. Given these facts, we believe that the most accurate data to use for timing is the beginning of the event pulse. Specifically, we believe the first sample of a pulse gives the

best timing information. However, one problem with the first sample is that its voltage is dependent on the amplitude of the pulse and the time the sample is taken in relation to the start of the pulse. Also, the sampling times of the ADC are asynchronous to the start of any event, so the first sample can occur anytime from the very start of the pulse to one full sampling period after the start of the pulse. To deal with these problems, we have developed methods to normalize the amplitude of the event pulse and a lookup table to correct for the asynchronous sampling.

2. Digital Pulse Timing

In order to test 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. While the data from the oscilloscope is technically in discrete time, we feel that the sampling period of the oscilloscope (40ps) is sufficiently small enough when compared to the ADC sampling period that the scope data can be used as continuous time data.

In our previous work [2], we found the model that best fit the data using a least square error was two exponentials (shown in 1). The rise time (τ_R) and fall time (τ_F) constants that best fit our dataset were 310ps and 34.5ns respectively.

$$V[n] = A \left(\exp^{\frac{-n * T_s}{\tau_R}} - \exp^{\frac{-n * T_s}{\tau_F}} \right) \quad (1)$$

To eliminate the difference in amplitude for the reference pulse and the incoming data pulse, we utilize the direct correlation between the area and amplitude of the pulse. Because the pulses have fixed time constants, the amplitude is a constant fraction of the area under the pulse. To normalize the amplitude of an event to the reference pulse, the ratio of the reference pulse area to the event pulse area is calculated. The event pulse can then be scaled by the ratio to equalize the amplitude of the reference and event pulses.

To determine the start time of the pulse, we use a lookup table that relates the voltages of the reference pulse to the time from the start of the pulse it takes to reach that voltage. In other words, the address to the memory is the voltage of the first sample of the event pulse. The output for that address is the time it takes for the reference pulse to reach that voltage. The final timing architecture is shown in Fig. 1.

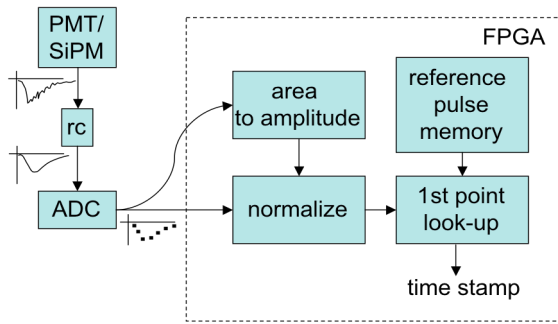


Fig. 1. The architecture of the timing pick-off circuit implemented in the FPGA.

The simulation results for the timing algorithm discussed above are shown in Table I.

Table I. Full width half max (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	2.737	2.438	2.369
140	1.794	1.8147	1.9251
300	1.0465	1.2489	1.3593
500	0.7406	0.6325	0.9407
1000	0.4554	0.4508	0.4554

An initial FPGA implementation of this timing algorithm has been designed. The design utilizes less than 1% of the logic and DSP blocks of an Altera Stratix II EPS60. It uses 2% of the memory, and can run at 100MHz.

3. Reference Pulse Discovery

One complication of this algorithm, as well as others that use a reference pulse [3], is the need to accurately discover a good reference pulse. For simulation, this can be achieved by using LMSE on the oscilloscope data. Unfortunately, there is nothing to indicate that the reference pulse parameters for a real system will be even close to those of the simulated system. This is because the system leading to the FPGA will be different from the system that was used to acquire the data used for simulation. Also, different detectors may need unique reference pulses.

To deal with this, we are currently developing an algorithm that utilizes our amplitude normalization and timing lookup technique to refine an initial reference pulse. The idea is to reconfigure the FPGA to capture and store many event pulses and utilize these pulses to form the reference pulse. Figure 2a shows the data that is stored as it comes into the FPGA. This data is scattered by differing amplitudes and sub-sampling rate time shifts. There is a two-part process to form the composite reference pulse. The first step is to normalize the amplitudes as discussed above. Then, an initial reference pulse is generated by averaging the data. Next, each individual pulse is aligned to this reference pulse by shifting in time. Once all points are aligned (Figure 2b), a final reference pulse can be calculated.

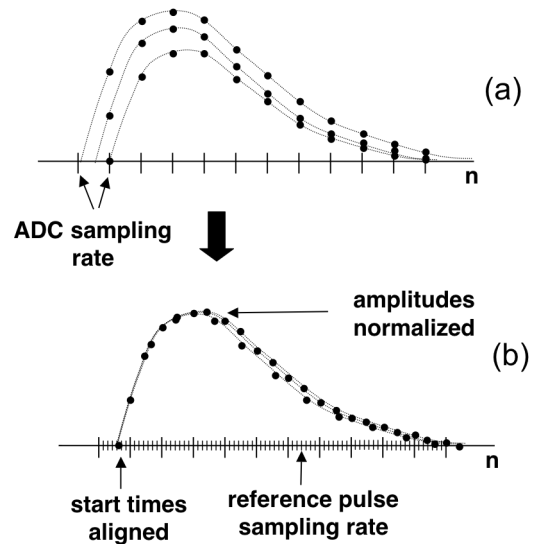


Fig. 2. Pulse discovery technique illustration.

4. Discussion

The initial results indicate that we will be able to achieve timing pickoff precision well below the sampling rate of the ADC, and may approach the precision of an analog CFD as ADC bandwidths increase in the future. Additionally, an initial FPGA implementation indicates that this algorithm utilizes very little resources of the FPGA. Moving forward, we plan on using the FPGA implementation to run experiments with event pulses from a PMT or SiPM that is connected to a Stratix II DSP development board. We also plan to implement our reference pulse discovery routine with the same setup. Questions that need to be explored for the reference pulse discovery are what is the required reference pulse resolution and how much data is needed to fill out the space at that resolution. We expect the majority of these next steps to be complete before the IEEE meeting in October.

Acknowledgments. This work was supported by NIH grant EB002117, Zecotek and Altera.

References

- [1] 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.
- [2] M.D. Haselman, S. Hauck, T.K. Lewellen, and R.S. Miyaoka, "Simulation of Algorithms for Pulse Timing in FPGAs," *Nuclear Science Symposium Conference Record*, 3161 (2007).
- [3] Guerra, P., et al., "Digital timing in positron emission tomography," *IEEE Nuclear Science Symp. Conf. Record*, 1929-1932, 2006.
- [4] 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.
- [5] 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.
- [6] M.D. Fries, J.J. Williams, "High-Precision TDC in an FPGA using a 192-MHz Quadrature Clock," *IEEE Nuclear Science Symp. Conf. Record*, vol. 1, 2002, pp. 580-584.