

Simulation of Algorithms for Pulse Timing in FPGAs.

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

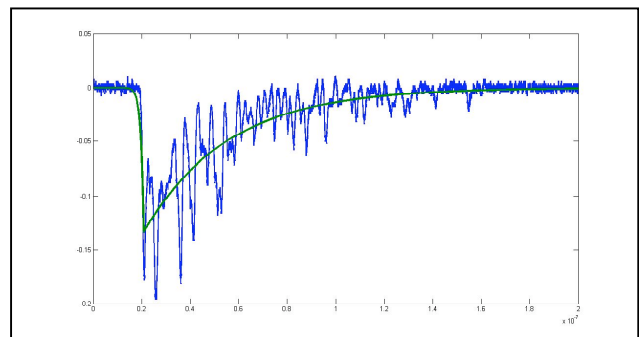
Abstract: Modern Field Programmable Gate Arrays (FPGAs) are capable of performing complex discrete signal processing algorithms with clock rates of above 100MHz. This, combined with FPGA's low expense and ease of use, make them an ideal technology for pulse timing and are a central part of our next generation of electronics for our pre-clinical PET scanner systems. To that end, our laboratory has been developing a pulse timing technique that uses pulse fitting to achieve timing resolution well below the sampling period of the analog to digital converter (ADC). While ADCs with sampling rates in excess of 400MS/s exist, we feel that using ADCs with lowing sampling rates has many advantages for positron emission tomography (PET) scanners. It is with this premise that we have started simulating timing algorithms using MATLAB in order to optimize the parameters before implementing the algorithm in Verilog. MATLAB simulations allow us to quickly investigate filter designs, ADC sampling rates, precisions, and algorithms with real data before implementation in hardware. We report our initial results for a least squares fitting algorithm of PMT pulses.

1. Introduction

Current state of the art timing pickoff for PET systems are 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 already used circuit board area to compute the timing pickoff. This approach is part of our next generation of electronics for our pre-clinical PET scanner development.

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 [Fries 02]. 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. Another constraint for our applications is keeping the operating frequency away from the proton resonance frequencies in a 3T MR scanner. Also, because clock rate 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. If this assumption is true, then the start 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. Unfortunately, the rise time of the pulse shown in Figure 1 is only a few nanoseconds long, so there is no guarantee that the ADC will sample a point on the rising edge. Without a point on the rising edge, it is impossible to distinguish curves with different amplitudes since it has a



constant decay.

Figure 1: Sample pulse from a PMT coupled to an LSO scintillator with the best least squares fitting two exponential curves overlaid.

2. Materials and methods

In order to test timing algorithms on real data, we used a 25Gs/s oscilloscope to sample 19 pulses from a PMT (add PMT mfr and model #) 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 (~16ns), that the scope data can be used as continuous time data. 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, ADC quality, and FPGA precision to be quickly investigated before we commit to an implementation. It also allows us to try many different fitting algorithms.

The first task was to determine the rise and fall times of the pulses. It has been previously reported in [2] that a linear rise sufficiently modeled the rise of the pulse, so we investigated ideal pulses with a linear rise and exponential decay, as well as an exponential rise and decay. The rise and decay times that gave the best overall least squares fit were 1.2 and 34.8ns.

The 19 pulses were processed using these determined times. In order to remove any bias introduced by the oscilloscope, the pulses were first shifted so that the start points of all pulses occurred at the same time. After the pulses are aligned, they are filtered with an ideal low-pass RC filter. The filtered pulses are then sampled to simulate an ADC of a given frequency. Each pulse is sampled with 100+ different starting points to give 100+ different sampled representations of each pulse. Using a least squared fit, each of the representations were then fit with an ideal pulse that was filtered with the same RC filter. This best ideal pulse was then interpolated to determine the starting point. We now had 100+ interpolated starting points for each pulse that have a small distribution due to the noise still present in the filtered pulse. The interpolated starting points from one pulse were then compared to all other points from all other pulses and a histogram of the differences was generated. This process was done for many RC constants and ADC sampling values.

3. Results

We ran MATLAB simulations using both linear and exponential rising ideal curves. Preliminary results show that using two exponentials gives better timing resolution. It is not clear why this is the case and will require further explorations. A sample of the resulting histogram with a 70MHz ADC and a 66.7MHz cutoff low-pass filter is shown in Figure 2. Since the shape of the histogram in Figure 2 is not Gaussian, it is not reasonable to report the FWHM.

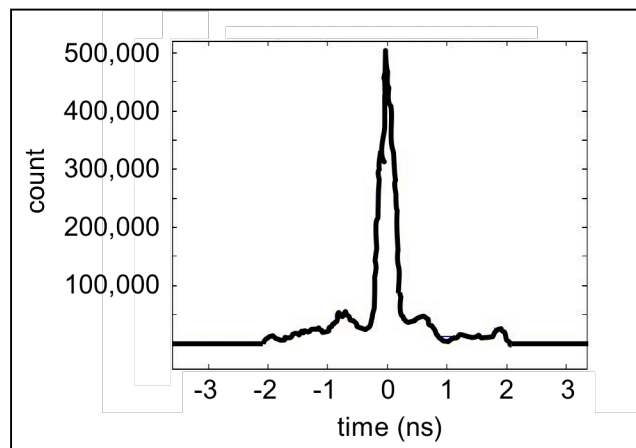


Figure 2: Simulated coincidence timing resolution

In order to give a figure of merit, we report the standard deviation in Table I. 70 MHz represents our current hardware design, while 100MHz and 140MHz give us an idea of how the sampling rate affects the pulse fitting.

Table I

Standard deviation of the simulated coincidence timing for a range of ADC sampling rates and low-pass filter cutoff values.

rc cutoff (MHz)	ADC sampling (MHz)	correlation histogram std dev. (ps)
33.3	70	541
15	70	520
33.3	100	388
15	100	411
33.3	140	312
15	140	359

4. Discussion

The initial results indicate that we will be able to achieve timing pickoff precision well below the sampling rate of the ADC. We plan on simulating on 10x more samples which we believe will help to make the histograms appear more Gaussian. We also will investigate how noise models for the ADC and low-pass filter will affect the results. Simulating a CFD on the sample pulses will also be undertaken to give an indication of accuracy of our simulations when compared to real implementations. Finally, we will investigate the lower bounds of our algorithm similar to the work in [7]. 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. N. Zhang et al., "A Pulse Shape Restore Method for Event Localization in PET Scintillation Detection," *IEEE Nuclear Science Symp. Conf. Record*, vol 7, 2004, pp. 4084-4088.
3. J.-D. Leroux et al., "Time Discrimination Techniques using Artificial Neural Networks for Positron Emission Tomography," *IEEE Nuclear Science Symp. Conf. Record*, vol. 4, 2004, pp. 2301-2305.
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.
7. A.O. Hero, N.H. Clinthorne, W.L. Rogers, "A Lower Bound on PET Timing Estimation with Pulse Pileup," *IEEE Trans. Nuclear Science*, vol. 38, no. 2, 1991, pp. 709-712.