Integrating Timing and Positioning Algorithms onto a Single FPGA Platform for Positron Emission Tomography

Marshal Barrett, Scott Hauck, Robert Miyaoka, Tom Lewellen

***Abstract*—Previous research at the University of Washington has developed methods to accurately extract coincidence timing [1] and positioning information [2] from an incoming digitized pulse stream within an FPGA. Up to this point, the timing and positioning algorithms have not been combined on a single system. We will incorporate both of these algorithms together on our own hardware platform for the purposes of collecting real data, working towards the completion of a full PET system, and refining the timing and positioning algorithms for real world implementation. To overcome obstacles such as scarcity of the FPGA's internal memory, we will use memory sharing or interpolation of memory data to reduce memory requirements without impacting each algorithm's accuracy. Using properly sized buffers and observation of the timing of signals, both timing and positioning algorithms will be incorporated together. We have designed a data acquisition board, complete with an Altera Stratix III FPGA, and 65 analog to digital converters to be used for the complete system. Both timing and positioning algorithms have been successfully tested on a separate development board.**

I. The Hardware Platform

The data acquisition board contains 65 channels total. 64 channels makes use of a 65MSPS, 12-bit analog to digital converter connected via LVDS (low voltage differential signaling) to the FPGA. The FPGA on the data acquisition board is produced by Altera. The board also incorporates flash memory, and five separate SDRAM interfaces, four of which are specifically intended for the three dimensional positioning algorithm described in previous work. The FLASH memory will be used to store statistical information for use in the positioning algorithm. [3]

II. The Timing and Positioning Algorithms

The most complex FPGA code is in support of the cMiCES and SES detectors developed at the University of Washington. For these detectors, we use an algorithm that computes the most likely 3D position of an event within a crystal. Before the position is computed, the summed signal for each column, and row is required. The pulse timing algorithm quantifies energy and pulse timing for each channel of raw data [1]. The positioning algorithm then accepts the energy from each channel, and computes the most likely position [2]. Both algorithms require the use of the FPGA's internal memory. For the timing algorithm, memory is required to store various lookup tables that are pre-computed. The information required for the first four stages of the positioning algorithm, as well as the lookup tables for each timing block, will be stored in the FPGA's internal memory. If the Stratix III device does not contain enough internal memory, options within each algorithm will be explored to more efficiently utilize external or internal memory on the current University of Washington Phase II Electronics Board.

A. Overview of the Timing Algorithm

The general idea of the timing algorithm is to use a previously computed reference pulse waveform to fit to the incoming pulse waveform. This approach provides sub-clock-cycle timing information, and accurate energy calculation should two pulse waveforms be overlapping (pulse pile-up [4]). Timing engines are cascaded together in order to deal with pile-up. The first timing engine will compute the timing and energy information for the first encountered pulse, and then subtract a reference pulse that has been fitted to the incoming pulse from the data stream. The next engine will then compute the timing and energy information for the second pulse from the output data stream of the first engine. The timing algorithm processes data at the same clock rate that data is received from the ADCs. [1]

Before energy and timing information is computed from a channel, the average "DC" signal, or baseline must be subtracted. The baseline, along with a noise threshold is computed for each channel. After the baseline has been subtracted, a pulse has been detected when two consecutive samples from a channel are above the noise threshold plus some small constant. The timing algorithm then calculates energy and timing information for the given data. [5]

B. Overview of the Three Dimensional Positioning Algorithm

The positioning algorithm takes energy from each channel as its input, and computes a measure of probability that an event occurred in a given location using precomputed values of mean and standard deviation signal levels from simulated events. Using a hierarchical search method, the most likely position is selected. Each hierarchy can be thought of as a stage, and compares the probability of nine total points. The most likely position from a stage is then passed on to the next stage as the center point. The next stage will then implement a finer search around the position computed by the previous stage. The increasing granularity of the each subsequent stage requires later stages to use more memory. For example the number of possible positions output from the fourth stage is 31 by 31, while the possible postions output from the third stage is 15 by 15. [2]

III. The Interface Between the Timing and Positioning Algorithms

One previously unexplored aspect of incorporating timing and positioning algorithms is the nature of the interface between them. As previously stated the timing algorithm processes data at the same rate at which pulses are received, while the positioning algorithm cannot process data at the same peak rate. This necessitates the use of a buffer to allow the positioning algorithm to process data as quickly as possible, while not losing data processed at a higher rate from the timing blocks.

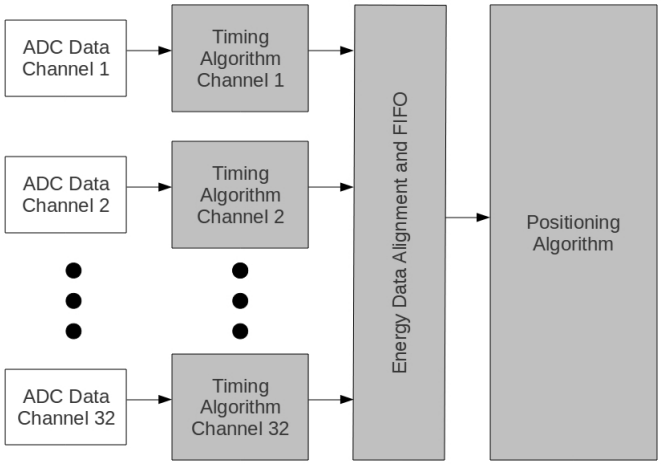

Fig. 1. High Level Block Diagram of FPGA

A. Memory Constraints

Both the positioning algorithm and timing algorithm make use of the FPGA's internal memory, making it a likely candidate for constraining the design. Should this be the case there are many possible avenues for reducing memory requirements. It is possible to run the external memory and memory controller internal to the FPGA at a higher clock speed than the rest of the FPGA to improve throughput. By doing this it is possible for multiple stages within the positioning algorithm to share memory resources and reduce internal memory requirements. Sharing certain look up tables between cascaded timing engines, or interpolation of appropriate data, are a few other ways to reduce memory requirements.

Interpolation is a powerful technique and could provide large savings in memory. Simple linear interpolation would be straight forward to implement especially when restricted to factors of two. More complex interpolation methods such as cubic spline interpolation have promise for even further reductions if such an algorithm can be efficiently implemented. Interpolation of data is a general solution to reducing look-up table size and can be applied to the aforementioned timing and positioning algorithms. We will investigate interpolation methods for use within the FPGA in our system.

IV. Conclusion

With the data acquisition board hardware in hand, we have begun the integration of positioning and timing algorithms for use in a single FPGA platform on our data acquisition board. We have begun designing the interface between the timing and positioning algorithms, and will shortly have the system fully integrated. This will provide a complete hardware system for event timing and positioning in our PET scanner, demonstarting the complete algorithms in a realistic setting. It will also demonstrate the capabilities of our custom data acquisition board. By the time of the conference, we expect to have a complete system implemented in a Phase II board and performance data with a cMiCE detector system to present.

V. Acknowledgements

This work is supported by Zecotek, Altera, and NIH grant EB002117, EB001563, and CA136569.

References

[1] M. D. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Simulations of algorithms for pulse timing in fpgas," *IEEE Nuclear Science Symp. Conf. Record (NSS/MIC)*, pp. 3161–3165, 2007.

[2] N. G. Johnson-Williams, R. S. Miyaoka, X. Li, T. K. Lewellen, and S. Hauck, "Design of a real time fpga-based three dimensional positioning algorithm," *IEEE Transactions on Nuclear Science*, vol. 58, pp. 26–33, February 2011.

[3] T. K. Lewellen, R. S. Miyaoka, L. R. MacDonald, D. DeWitt, and S. Hauck, "Evolution of the design of a second generation firewire based data acquisition sytem," *IEEE Nuclear Science Symposium and Medical Imaging Conference*, 2011.

[4] M. D. Haselman, J. Pasko, S. Hauck, and R. S. Miyaoka, "Fpga-based pulse pile-up correction," To appear in IEEE Transactions on Nuclear Science.

[5] M. D. Haselman, S. Hauck, T. K. Lewellen, and R. S. Miyaoka, "Fpga-based pulse parameter discovery for positron emission tomography," *IEEE Nuclear Science Symposium and Medical Imaging Conference*, pp. 2956–2961, 2009.