An FPGA-Based Array Processor for an Ionospheric-Imaging Radar

Tim Tuan, Miguel Figueroa, Frank Lind, Chucai Zhou, Chris Diorio, John Sahr
University of Washington, Seattle, Washington
timt@eecs.berkeley.edu

Abstract

Atmospheric scientists need to observe fluctuations in the ionosphere, both to probe the underlying atmospheric physics, and to remove the effects of these fluctuations from other measurements. We have built an FPGA-based, pipelined array processor that allows us to make these observations in real-time, using passive radar techniques. Our array processor time-multiplexes 16 multiply–accumulators across 1536 radar ranges, performing a pipelined correlation and integration of the radar signal for each range. A DSP-based postprocessor generates real-time range-Doppler profiles of the ionospheric targets.

1. Introduction

The remote sensing of turbulence in the Earth's ionosphere is a current problem in atmospheric science, with theoretical and practical implications. Observations of atmospheric fluctuations will allow scientists to better understand ionospheric physics. Such observations can also improve systems that communicate through the ionosphere, by allowing the detection and mitigation of degradations due to ionospheric fluctuations.

Conventional radar techniques for ionospheric observation utilize high-power RF sources and large antenna systems, which are prohibitively expensive to install and maintain. Recently, the Radar Remote Sensing Group at the University of Washington proposed a passive radar technique that uses commercial FM radio transmissions as an RF source [1]. The economic benefits of co-opting free commercial radio transmissions are enormous. The primary drawback of this approach is that it requires significant computational power to process the radar data. Even an optimized algorithm [2] requires a throughput of about 5GOPS to operate in real-time.

We have built a complete radar system to demonstrate this passive radar technique. Our system processes signals from local and remote FM receivers: An FPGA-based array processor implements a correlation/integration, and a DSP board performs an autocorrelation on the resulting data. Our array processor is dynamically reconfigurable, allowing end users to tune the system for target-specific parameters such as velocity and range resolution. This paper describes the radar signal processing system, focusing on the FPGA-based array processor.

This work was supported by The University of Washington Royalty Research Fund and by National Science Foundation grant ATM 9816217.

2. System architecture and implementation

The computational core of our radar signal-processing algorithm [2] comprises the following two equations:

$$Q[r/2;\tau] = \sum_{t} z[t;r]z^*[t-\tau;r] \qquad (1)$$

$$z[t;r] = \sum_{s=0}^{T-1} x^*[t+s]y[t+r+s] \qquad (2)$$

where x^* is the complex conjugate of the direct transmission, y is the scattered (remote) signal, T is the averaging window, and r is the range ($0 < r < R$). To simultaneously observe all targets over a range extent of 1000km, our system must computes 1536 ranges, i.e. $R = 1536$, with a typical T of 128 (allowed values of T vary between 32 and 128). Consequently, the cross-ambiguity function of Equation 2 is the bottleneck of the algorithm, and we designed a custom array processor to solve it in real-time. We compute the autocorrelation in Equation 1 on a high-speed DSP.

The top part of Figure 1 shows a block diagram of our radar signal processing system. The receivers sample data at 250kHz, and send it to the system over the Internet. The host performs some preprocessing, and packs the data into streams in the array processor memory. The array processor operates on the data and sends the results to a DSP board that performs the final computational pass.

The bottom part of Figure 1 depicts the architecture of our array processor. We mapped Equation 2 onto a 1536-stage virtual systolic array running at 250kHz. This virtual array is time-multiplexed onto a 16-stage, 25MHz hardware array, realized on FPGAs. The array sequentially computes 96 range blocks, with 16 ranges in each block. The main advantage of this time-multiplexing approach is that all the memory in the architecture is external to the array, greatly facilitating the hardware implementation.

We quantized the local and remote signals at their received SNR; in this case at 6 bits and 1 bit, respectively. Both signals are complex, so the datapath is 12 and 2 bits wide, respectively. Because the remote signal needs only 1-bit quantization, we were able to replace the multiply in Equation 2 with a composed multiplex operation, thereby allowing the array to fit in two FPGAs. To accommodate the maximum decimation factor (T) of 128, we used 13-bit

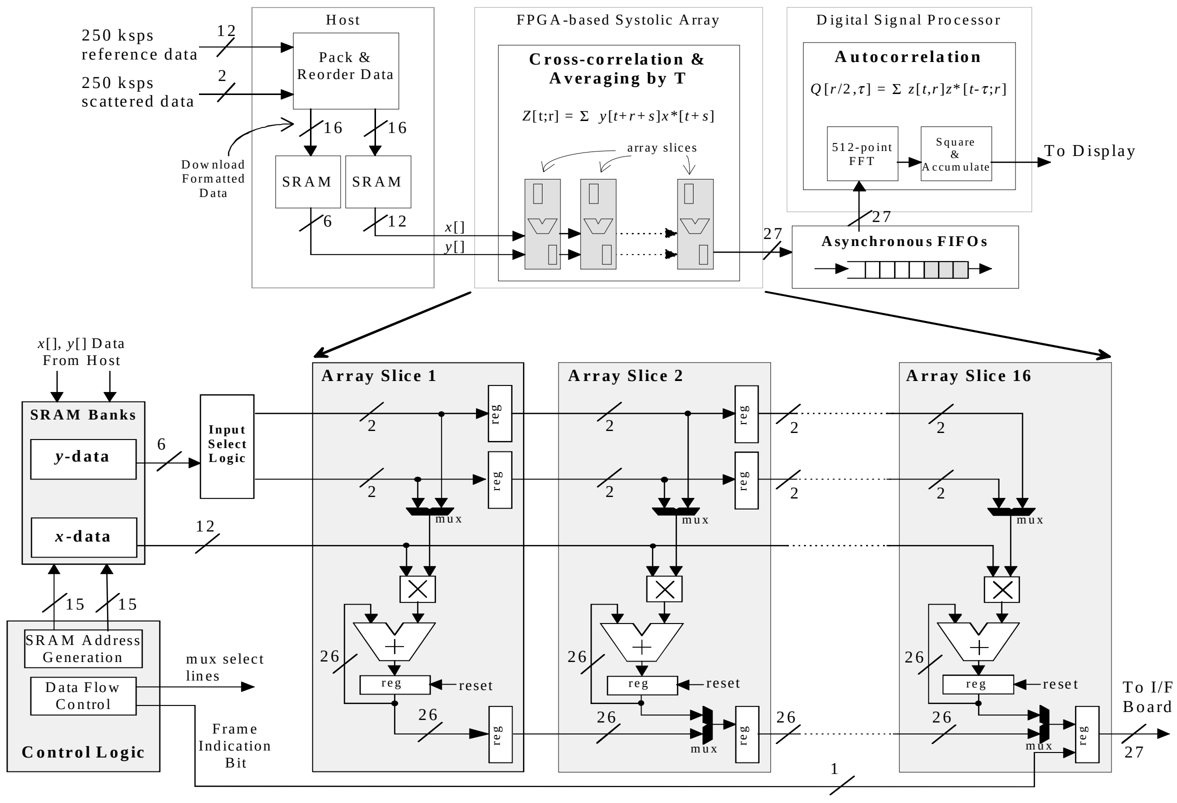

Figure 1. Radar-processor block diagram. The data are preprocessed by the host, followed by correlation and averaging by the systolic array. The array comprises 16 slices. Each slice has one MAC, two input pipeline registers, and one output pipeline register. A bridge board asynchronously interfaces the array to a DSP for computing the necessary post-processing.

complex adders. The local signal (x^*) is broadcast, while the remote signal (y) is pipelined.

To eliminate dead time between each range-block computation, the array preloads data for the next computation. Because only one memory address can be issued at a time, the host is responsible for packing the y-data streams into 6-bit words that include preload data.

We implemented the complete radar signal processor, shown in Figure 1, in a PC. The PC hosts a DEC Pamette board [3], which we use for our array processor, and a TI TMS320C6201 DSP evaluation board that performs the post-processing. The radar receivers sample data at 250kHz (I/Q), and send the data to the radar processor over the Internet. The host formats the input data and writes x and y to the Pamette SRAMs. The array processor solves the cross-ambiguity function (see Equation 2), and passes its output to the DSP board via a custom bridge board, which hosts an asynchronous FIFO interface between the Pamette and the DSP. The DSP computes the autocorrelation function and delivers the results to the host via shared memory.

The Pamette board contains four Xilinx XC4020E chips, each containing approximately 20,000 gates. Mapping the array to the FPGAs was primarily constrained by inter-FPGA bus-width limitations on the Pamette board. Consequently, we used two FPGAs for datapath logic, and the remaining two FPGAs for control logic and data routing. A custom board could host the entire array on two XC4020E chips.

3. Conclusion

This paper has described an FPGA-based system to process signals for a passive radar. The system uses an FPGA-based array processor and a DSP board to compute a cross-ambiguity function and a crosscorrelation on the received data. The system is dynamically reconfigurable: It allows users to modify range, delay and decimation factors, effectively providing the ability to “zoom” into a region of interest. The architecture can be expanded to accommodate additional receivers and enable the observation of additional data, such as target azimuth.

4. References

[1] J. D. Sahr and F. D. Lind, “Passive Radio Remote Sensing of the Atmosphere using Transmitters of Opportunity,” Radio Science Bulletin, no. 284, pp. 4-7, 1998.

[2] J. D. Sahr and F. D. Lind, “The Manastash Ridge radar: A passive bistatic radar for upper atmospheric radio science,” *Radio Science*, vol. 32, pp. 2345-2358, 1997.

[3] M. Shand *PCI Pamette V1* http://www.research.digital.com/SRC/pamette