Accelerating a Software Radio Astronomy Correlator using FPGA co-processors

Andrew Woods,
Michael Inggs and Alan Langman *
drew.woods@gmail.com
Centre for High Performance Computing
www.chpc.ac.za

Index Terms—Correlator, Reconfigurable Computing, DiFX, FPGA, Radio Astronomy.

1 Introduction

This article presents and characterises our work on accelerating a software radio astronomy correlator using reconfigurable computing (RC) hardware. Radio astronomy correlation is an embarrassingly parallel signal processing application, which is used heavily in radio astronomy for imaging and other astronomical measurements. Radio astronomy correlators typically operate on huge data sets and often require real-time processing, as storage of raw data is impractical - resulting in substantial computational requirement. Currently FPGAs are the preferred processing architecture used in modern large radio astronomy correlators [1] and perform well on the types of DSP functions that correlators perform.

In this paper we set out to accelerate the DiFX (Distributed FX) correlator, a software correlator, using FPGA reconfigurable computing hardware. —– hoping to inherit some of the advantages that larger production FPGA correlators have over software.

2 Background

Modern large radio telescopes almost always consist of a number of individual antenna which are used to detect electromagnetic radiation from interstellar objects. Alone, each of these antennas only produces very low resolution results. However, if these individual antennas communicate, their detected signals can be combined, producing much higher resolution results. The process of combining the individual received signals[1] from each of the antennas is the correlator's duty. The correlator is required to calculate the spectral power for each antenna pair, also referred to as a baseline. The final outputs of the correlator are complex visibilities, created from each baseline and represent spectral components of the brightness distribution of the sky [2]. This is a quadratic problem and is one of the most computationally expensive operations of the radio astronomy telescope.

The DiFX correlator, developed at Swinburne University, is a parallel, open-source, software implementation of a fully functional radio astronomy correlator [3]. Designed to work with the less processing intensive very long baseline interferometry (VLBI), the DiFX is an attractive correlator solution for smaller correlator arrays. The DiFX correlator has had a positive response in both astronomy and HPC communities allowing research to be carried out on standard linux compute clusters, without sharing or endangering production correlators.

Although a software correlator has many appealing attributes, it cannot match an equivalent FPGA generation in DSP function performance [4]. The DiFX correlator is an application that is well suited to co-processor acceleration and provided an opportunity to experiment with reconfigurable co-processors while creating something meaningful.

3 Related Work

Currently efforts have been made to port sections of the DiFX to both GPU and Cell BE platforms by the University of Western Australia [5] and Helsinki University of Technology [6] respectively, with encouraging results.

Besides the code porting projects, the National Radio Astronomy Observatory (NRAO) and Max Plank Institute fur Radioastonomie (MPIfR) have adopted the DiFX correlator for the correlation of their Very Long Baseline Array (VLBA) data [7] [8] and have released their own NRAO-DiFX modification [9].

4 FX Correlation

Ignoring the smaller intricacies,[2] radio astronomy correlation is at heart a relatively straight forward problem to describe analytically. The radio astronomy correlator

* *National Research foundation, Karoo Array Telescope*

1. Interferometery/Apature Synthesis

2. the more subtle intricacies are important to the accuracy of the correlator but largely computationally insignificant

computes the correlation for each baseline producing the cross power spectrum. This correlation operation is where the majority of processing time in the DiFX correlator is spent and was the focus for RC acceleration. The DiFX correlator computes the correlation in the frequency domain, which is broken into two separate stages. Firstly, the FFT is computed for each of the K antenna in the array, for a set sample length L (eq. 1); secondly, the transformed output of each antenna is multiplied with every other antenna[3], and accumulated for a period of A samples (eq. 2).

$$S_{a,k}[v] = \sum_{l=0}^{L-1} x_k[l] e^{-i2\pi vl/L} \quad (1)$$

$$\mathbf{C}_{A,i,j}[v] = \sum_{a=0}^{A-1} S_{a,i}[v] S^*_{a,j}[v] \quad (2)$$

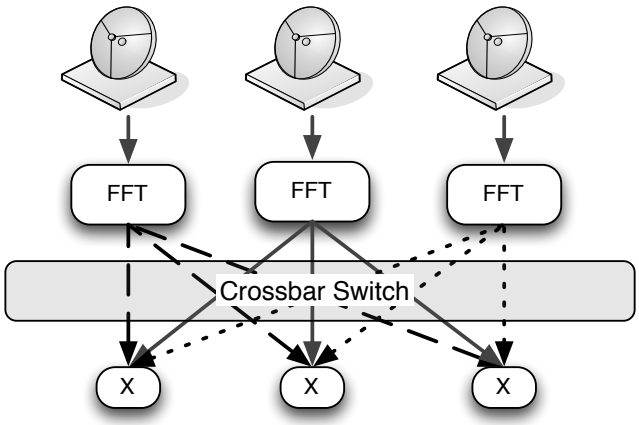

Fig. 1. The correlation operation with 3 antenna, which equates to 6 baselines correlations (including autocorrelation)

The correlation in Equation 2 is performed for each baseline, resulting in $K(K+1)/2$ operations, an $O(N^2)$ problem, while the FFT in Equation 1 is $O(NlogN)$. With Equation 2 being the dominant function for large inputs, combined with the availability of reasonably optimised FPGA FFTs, our RC correlator only implements the conjugate complex multiplication and accumulation stage of the correlation (eq. 2). Another issue supporting this choice is the limited logic gates available in the FPGA.

5 Reconfigurable Computing Hardware

The correlator acceleration was done using 2 local Nallatech H101 FPGA Accelerator boards. The Nallatech H101 consists of a Xilinx Virtex 4 LX100 processing FPGA, 0.5MB Block RAM, 16MB SRAM and 512MB DDR2 SDRAM. Both cards are hosted in an Intel Quad core Xeon 3.0GHz system on a shared PCI-X bus.

Additionally, the Edinburgh Parallel Computing Centre (EPCC) has kindly given us access to their Maxwell machine. Maxwell has 32 Nallatech H101s, but with a larger Virtex 4 LX160 FPGA. Unfortunately, final results from Maxwell were not ready at the time of writing and all benchmarks are done on our 2 local boards.

The Nallatech boards were programmed with Dime-C, Nallatech's C to HDL tool.

6 Approach and Design

Ideally, one would like to port suitable DiFX correlator functions independently to the RC hardware directly. This would allow the RC and software functions to be interchangeable without modifying other parts of the correlator. Unfortunately, we found it difficult to decompose the correlator into discrete stages due to the strong interdependence between sections.

These complications resulted in us creating an RC correlator independent from the DiFX correlator. Custom software code was used to perform the initial FFT stage and control of the RC correlator. However a simplified DiFX correlator was created from the source[4], which was used as an appropriate benchmark to evaluate our reconfigurable computing version.

7 Implementation

The implementation involved three distinguishable aspects, processing, IO management and control. Unfortunately due to page restriction we can only outline these points.

On the Nallatech V4LX100's we were able to implement 96 FPUs per FPGA, which allowed for 12 complex conjugate multipliers. These were configured to compute successive time steps, rather than computing adjacent spectral channels. By computing successive time steps we are able to reduce external memory access by deepening the pipeline.

Using the available memory storage types effectively can be non-trivial depending on the situation. For example, double buffering was important to avoid branching hazards, but consumes limited resources and increases host communication.

Since each antenna in the array needs to be multiplied by one less than its predecessor (the handshakes at a party scenario), the result is a triangular number of operations and shaped domain. This irregular shaped domain is most easily described by two nested loops. However Dime-C is only capable of pipelining the innermost loop and must break the pipeline to iterate through the outer most loop. This is problematic, especially since we have intentionally deepened our pipeline. The two loops were fused into a single loop variable at the expense of a small number of redundant operations.

3. Autocorrelations are also performed

4. The CPU code used Intel's Performance Primitives libraries which make use of SSE vector instructions

8 RESULTS

Figure 2 presents the performance speedup our two Nallatech FPGAs had over optimised software running on a 3.0GHz Quad-Core Xeon (although the software correlator is running on a single thread). The FPGAs outperform the CPU in most situations, except when the array size is small.

The RC correlator implementation is fully pipelined, so the input size has very little effect on its processing efficiency, once data is in onboard memory. Unfortunately, the FPGA is connected via a slow PCI-X bus, resulting in large transfer overheads, affecting performance when the number of input antenna is low[5].

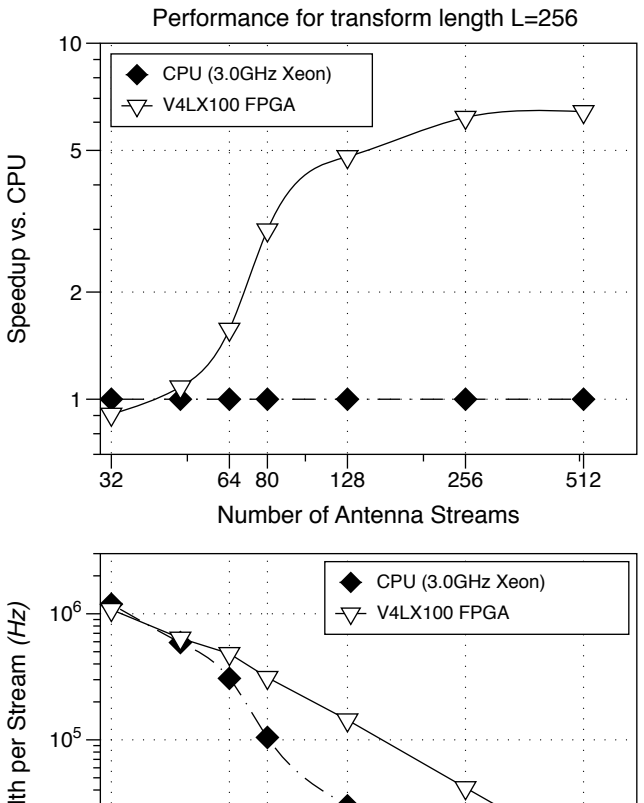

Fig. 2. Results showing the performance of the two Nallatech FPGAs vs. the single threaded Xeon 3.0GHz processor. Correlation results are accumulated over for a period of 1000 time samples, $A = 1000$ and have 256 spectral channels per antenna stream

5. The computation to communication ratio increases steeply as the number of input antenna increase.

9 CONCLUSIONS

Our RC correlator implementation is, for the majority of the time, faster than the optimised CPU code. This is despite the fact that the Virtex 4 FPGAs used are out of date. Newer FPGA generations can have up to 20x the number of onboard DSP blocks, which was a limiting factor to our correlator implementation. Despite these advantages, the software correlator is however running on a single thread, required much shorter development times and does not require expensive FPGA hardware to run.

In the end RC did not provide an easy method of accelerating for the DiFX correlator. However, this is not unique to RC co-processors as [5] had similar complications porting the DiFX correlator to GPUs. This could indicate that the DiFX correlator is not well suited to co-processor acceleration. However, we have shown that radio astronomy correlation can be accelerated using reconfigurable computing.

ACKNOWLEDGMENTS

The author would like to thank the Karoo Array Telescope (KAT) Project for funding my studies.

The Center for High Performance Computing (CHPC) in Cape Town, who have provided my postgrad work with a home.

Adam Deller for his support and involvement with our FPGA work on the DiFX correlator.

Edinburgh Parallel Computing Centre (EPCC) for accommodating me while in Edinburgh and giving us access to Maxwell.

Chris Harris for his advice in his emails.

REFERENCES

[1] J. W. C. Chang and R. B. ., "Bee2: a high-end reconfigurable computing system," *Design & Test of Computers, IEEE*, vol. 22, pp. 114– 125, 2005.

[2] A. Thompson, J.Moran, and G.Swenson, *Interferometry and Synthesis in Radio Astronomy*, 2nd ed. Wiley-VCH, 2004.

[3] M. B. A. Deller, S. Tingay and C. West, "Difx: A software correlator for very long baseline interferometry using multiprocessor computing environments," *The Publications of the Astronomical Society of the Pacific*, vol. 119, pp. 318–336, 2007.

[4] A. M. Fabrizio Stefani, "Fft benchmarking for digital signal processing fft benchmarking for digital signal processing technologies," 2005.

[5] L. S.-S. Chris Harris, Karen Haines, "Gpu accelerated radio astronomy signal convolution," *Experimental Astronomy*, vol. 22, pp. 129–141, 2008.

[6] J. R. Jan Wagner, "Software correlation on the cell processor," in *6th International e-VLBI Workshop*, 2007.

[7] J. Romney, "A vlba upgrade conforming a vlba upgrade conforming to vsop to vsop- a vlba upgrade conforming a vlba upgrade conforming to vsop-2 specifications," in *VSOP-2*, Dec, 2007.

[8] H. R. W. Alef, D. Graham and A. Roy, "Software correlator at mpifr: Status report," in *European VLBI Group for Geodesy and Astrometry Meeting*, 2007.

[9] W. Brisken. (2008, Feb) A guide to software correlation using nrao-difx version 1.0. Tech Report. National Radio Astronomy Observatory.