Application of Reconfigurable Computing Technology to Multi-KiloHertz Micro-Laser Altimeter (MMLA) Data Processing

Wesley Powell, Philip Dabney, Edward Hicks, and Maxime Pinchinat
National Aeronautics and Space Administration
Goddard Space Flight Center
Greenbelt Road, Greenbelt MD, 20771
301-286-1423

Abstract - The Multi-KiloHertz Micro-Laser Altimeter (MMLA) is an aircraft based instrument developed by NASA Goddard Space Flight Center with several potential spaceflight applications. This presentation describes how reconfigurable computing technology was employed to perform MMLA signal extraction in real-time under realistic operating constraints.

The MMLA is a "single-photon-counting" airborne laser altimeter that is used to measure land surface features such as topography and vegetation canopy height. This instrument has to date flown a number of times aboard the NASA P3 aircraft acquiring data at a number of sites in the Mid-Atlantic region.

This instrument pulses a relatively low-powered laser at a very high rate (10kHz) and then measures the time-of-flight of discrete returns from the target surface. The instrument then bins these measurements into a two-dimensional array (vertical height vs. horizontal ground track) and selects the most likely signal path through the array. Return data that does not correspond to the selected signal path are classified as noise returns and are then discarded.

The MMLA signal extraction algorithm is very compute intensive in that a score must be computed for every possible path through the two dimensional array in order to select the most likely signal path. Given a typical array size with 50 x 6, up to 33 arrays must be processed per second. And for each of these arrays, roughly 12,000 individual paths must be scored. Furthermore, the number of paths increases exponentially with the horizontal size of the array, and linearly with the vertical size. Yet, increasing the horizontal and vertical sizes of the array offer science advantages such as improved range, resolution, and noise rejection.

Due to the volume of return data and the compute intensive signal extraction algorithm, the existing PC-based MMLA data system has been unable to perform signal extraction in real-time unless the array is limited in size to one column. This limits the ability of the MMLA to operate in environments with sparse signal returns and a high number of noise returns.

However, under an IR&D project, an FPGA-based, reconfigurable computing data system has been developed that has been demonstrated to perform real-time signal extraction under realistic operating constraints. This reconfigurable data system is based on the commercially available Firebird Board from Annapolis Microsystems. This PCI board consists of a Xilinx Virtex 2000E FPGA along with 36 MB of SRAM arranged in five separately addressable banks. This board is housed in a rackmount PC with dual 850MHz Pentium processors running the Windows 2000 operating system. This data system performs all signal extraction in hardware on the Firebird, but also runs the existing "software based" signal extraction in tandem for comparison purposes.

Using a relatively small amount of the Virtex XCV2000E resources, the reconfigurable data system has demonstrated to improve performance improvement over the existing software based data system by an order of magnitude. Performance could be further improved by employing parallelism. Ground testing and a preliminary engineering test flight aboard the NASA P3 has been performed, during which the reconfigurable data system has been demonstrated to match the results of the existing data system.

TABLE OF CONTENTS

1.0 INTRODUCTION

Reconfigurable technology has for some time been recognized as having potential to both dramatically improve spacecraft onboard data processing and provide flexibility to

modify processing algorithms. Yet with the exception of a planned technology demonstration mission [1], no GSFC flight instrument or subsystem has yet incorporated this technology. There are several reasons for this ranging from the parts issues to lack of flight heritage.

One of the primary reasons in the instrument arena is the fact that in order to be used on a flight instrument, reconfigurable computing must be incorporated into the instrument concept early in its development. First, the instrument scientists must become acquainted with the technology and its benefits. Secondly, data processing algorithms must be adapted to exploit the advantages of reconfigurable computing architecture (i.e. pipelining, parallelism, etc).

The purpose of this IR&D project is to apply reconfigurable computing to a prototype science instrument, the Multi-KiloHertz Microlaser Altimeter (MMLA), early in its development phase. In so doing, this project is intended to (1) develop reconfigurable architectures to serve as a basis for future spaceborne data systems, and (2) enable single-photon-counting laser altimetry on future space missions.

The MMLA is currently an aircraft based instrument that is designed to fly aboard the NASA P-3 research aircraft. Future variants of this instrument are slated to fly on the NASA ER-2 research aircraft and on the Space Shuttle as a GAS (Get Away Special) can experiment. There is also potential to use MMLA technology on upcoming lunar and interplanetary missions.

2.0 MMLA Instrument Concept

The MMLA is an airborne single-photon-counting laser altimeter that is currently being developed at GSFC under the Instrument Incubator Program (IIP). The unique characteristics of this instrument are: (1) its use of a very low output laser; (2) its ability to be flown from high altitudes that don't require special FAA clearances; and (3) its very high laser fire rate of 10 kHz [2]. By comparison, the Multi-Beam Laser Altimeter (MBLA) laser fire rate is about 290 Hz and Geosciences Laser Altimetry System (GLAS) is less than 100 Hz.

As illustrated in Figure 1, this instrument functions by repetitively firing a laser at the ground surface and timing the detected single photon returns reflected from targets of interest in order to determine their height. The Multi-KiloHertz Microlaser altimeter is so named because it uses a very compact, low-energy, sub-nanosecond pulse, solid-state Micro Laser as its source and relatively small (typically 10 to 20 cm in diameter) telescopes. This results in a factor of 100 reduction in telescope weight and volume, as compared to conventional spaceborne laser altimeters. Operating at thousands of pulses per second, the surface sampling rate is approximately 100 times higher than that of prior spaceborne laser altimeters having the same transmitter power-aperture product. The MMLA demonstrates its advantages in developing high spatial resolution digital topographic databases, monitoring biomass and vegetation canopy heights, sea and lake levels, ice sheet thickness, etc.

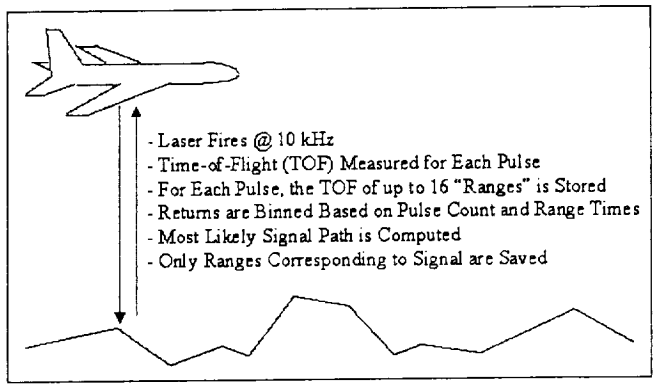

Figure 1 – MMLA Instrument Concept

2.1 MMLA Data Processing

The MMLA uses a passively Q-switched laser that fires at rate of up to 10 KHz. For each laser fire, the MMLA detects and computes the time-of-flight for up to 16 returned photons. A histogram is performed on these returns based on the laser fire time and the measured range of each return.

2.1.1 Plotdata Array Construction

Illustrated in Figure 2, this results in the construction of a two dimensional "plotdata" array. The plotdata array is subdivided in the x dimension. This dimension corresponds to the laser fire time, and hence, the horizontal ground track of the aircraft) into frames. The array is subdivided in the y dimension into range bins. The entire span of the plotdata array in the x dimension is called a "superframe", which consists of an integral number of frames. The span in the y dimension is referred to as the range window.

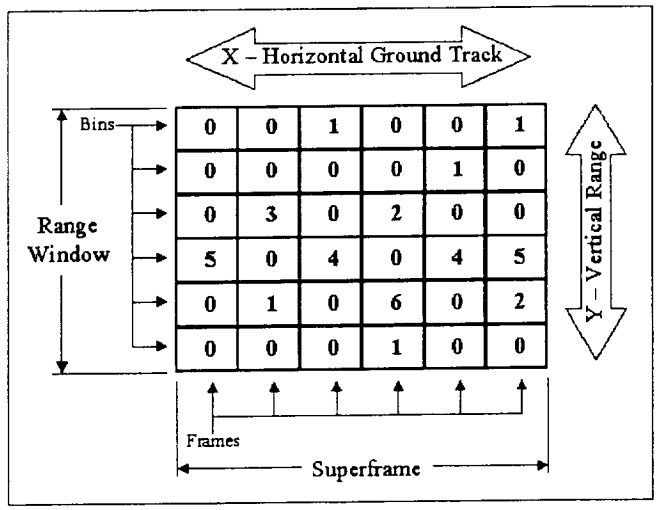

Figure 2 – Plotdata Array

Once the plotdata array is constructed, the MMLA then selects the most likely signal path through the plotdata array. All returns that fall along the selected signal path are then archived for future ground processing. This signal path through successive plotdata arrays is also displayed to provide a real-time indication of signal quality and ground terrain.

2.1.2 Signal Path Selection

The selection of the most likely signal path is by far the most compute intensive step in the MMLA signal processing. This is performed "brute force" through an iterative process where a score is computed for every possible path through the plotdata array (within a set of constraints). The score of each path is computed by accumulating the total number of returns in each cell (array element) along the path. However, a path is considered to be a potential signal path only if at least NN of the MM cells of the path exceed a pre-defined "cell threshold" parameter. The signal path (if one exists) with the highest score is then selected as being the most likely signal path.

A path through the plotdata array is comprised of a sequence of cells traversing the array from the first column to the last column. The paths are constrained by a "cell spread" parameter that determines the set of valid cells in column x that could be transitioned to from a cell in column x-1. Hence, cells in adjacent columns cannot vary in vertical distance by more than (cell spread/2) cells. For example, with the typical cell spread of 3, a cell at location [x, y] can transition to cell [x+1, y-1], [x+1, y], or [x+1, y+1]. Figure 3 illustrates the path from a single starting cell with a cell spread of 3 and a length of 3.

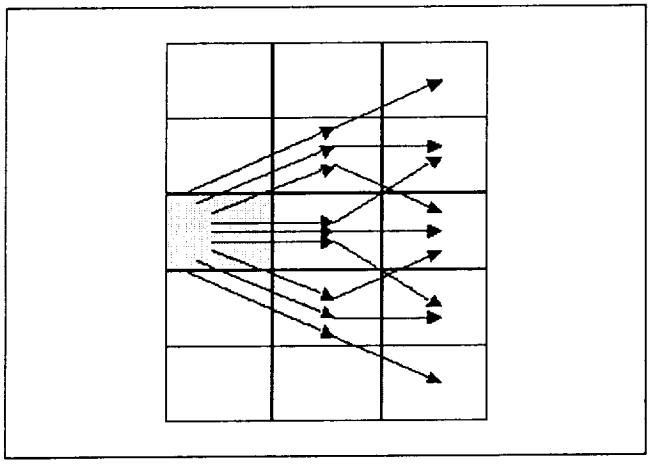

Figure 3 – Paths from Single Cell

The following equation gives the total number of paths through a plotdata array with dimensions [MM x Num_Bins], where MM is the superframe size and Num_Bins is the number of bins in the range window.

$$\text{Number of Paths} = \text{Num\_Bins} \times \text{Cell\_Spread}^{(MM-1)}$$

For a Cell_Spread = 3 and a plotdata array with where MM = 8, Num_Bins = 50, there are a total of 109,350 paths that must be computed.

2.2 MMLA Baseline Configuration

The MMLA instrument consists of two standard equipment racks containing COTS (commercial off-the-shelf) components and a transmitter/receiver assembly containing the laser, telescope, and detector. The baseline MMLA configuration is illustrated in Figure 4.

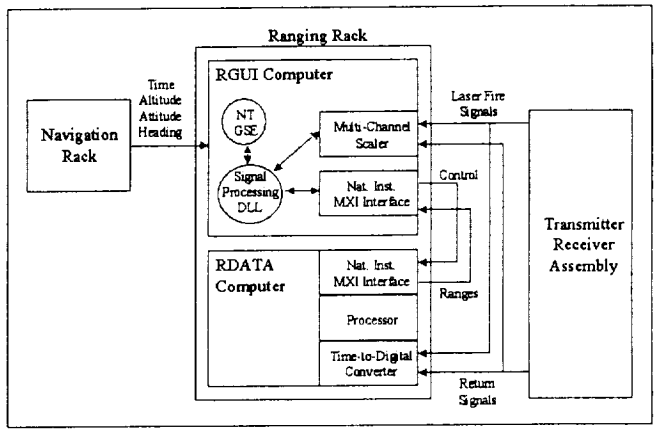

Figure 4 – Baseline MMLA Configuration

The Ranging Rack performs all of the ranging measurements, instrument control, and signal extraction processing. This rack houses two computers, the Ranging System Data Collector (RDATA) Computer and the Ranging System Graphical User Interface (RGUI) Computer. The RDATA is a VXI based computer that collects all ranging data and forwards it to the RGUI. The RGUI performs the instrument control, signal extraction processing, data archiving, and provides the user interface to the instrument. This PC is configured with dual Pentium 850MHz processors and uses the Windows 2000 operating system. The RGUI uses the NTGSE software which provides a generic platform for instrument control and monitoring. The MMLA signal extraction software is implemented as a separate DLL, containing functions called by the NTGSE.

The Navigation Rack houses the Navigation and Camera Control (NAV) Computer, which provides a GPS system time reference and determines the aircraft heading, altitude, and attitude. This data is sent via a shared file interface to the RGUI where it is included with the range data for ground processing.

The Transmitter/Receiver Assembly houses the instrument laser, telescope, and detector. This assembly is mounted to the floor of the aircraft cabin where it transmits the laser pulses and receives the returns through a window in the bottom of the aircraft.

3.0 RECONFIGURABLE COMPUTING (RC) DATA SYSTEM ARCHITECTURE

For this IR&D project, the baseline architecture was enhanced to allow signal extraction to be performed in hardware in parallel with the existing software signal extraction. As illustrated in Figure 5, these enhancements included (1) the addition of an Annapolis Microsystems Firebird FPGA board to the RGUI computer in the Ranging Rack, (2) the development of a reconfigurable computing (RC) application. This RC application consists of an FPGA image to perform the signal processing and a software module to integrate the hardware processing with the existing signal extraction software.

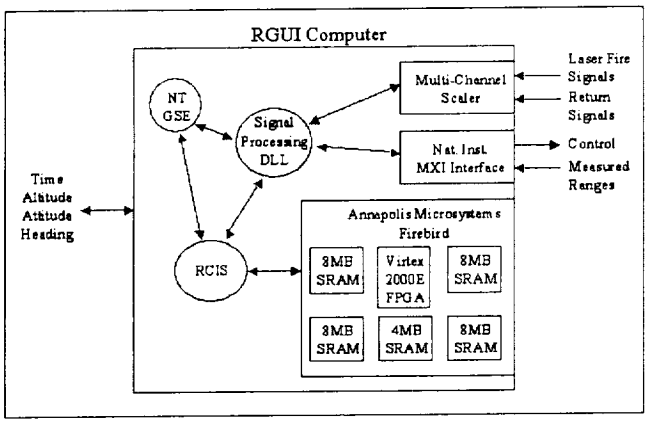

Figure 5 – RC Data System Architecture

The Annapolis Microsystems Firebird, is a commercially available PCI based FPGA computing board. Included on this board is a Xilinx XCV2000E FPGA and 36 MB of SRAM arranged in 5 separately addressable banks. The FPGA image consists of a set of core logic along with several generic modules from Annapolis Microsystems.

Written in C and implemented as a Windows 2000 DLL, the RCIS (Reconfigurable Interface Software) integrates the FPGA processing with the existing signal extraction software. This software (1) initializes the Firebird board and the FPGA image, (2) receives and formats instrument return data for hardware processing, and (3) displays and archives the processed data. The RCIS receives instrument data as a "log record", containing one second of instrument returns, from the existing signal extraction DLL via a shared memory interface.

3.1 Processing Flow

Upon receiving a log record, the RCIS constructs an array of [t, r] pairs corresponding to discrete returns, where "t" is the laser fire time and "r" is the measured range of that return. This array contains the consecutive [t, r] pairs that make up a single superframe. The RCIS then transfers this array to the "firedata" memory on-board the Firebird for hardware processing.

In hardware, three main processing steps occur as shown in Figure 6. In the Build Plotdata step, the [t, r] pairs in the firedata memory are binned to construct a plotdata array. As the plotdata array is constructed, each firedata memory location is annotated with its corresponding frame and bin number. In the Path Score step, all possible paths through the plotdata array are scored and the most likely signal path (if any exist) is stored. In the final Path Trace step, the frame and bin numbers of all [t, r] pairs in the firedata memory are compared against the frame and bin numbers of the cells along the selected path. Those [t, r] pairs that were binned into the cells of the selected path are then annotated with a "z" bit that indicates that it is a signal return.

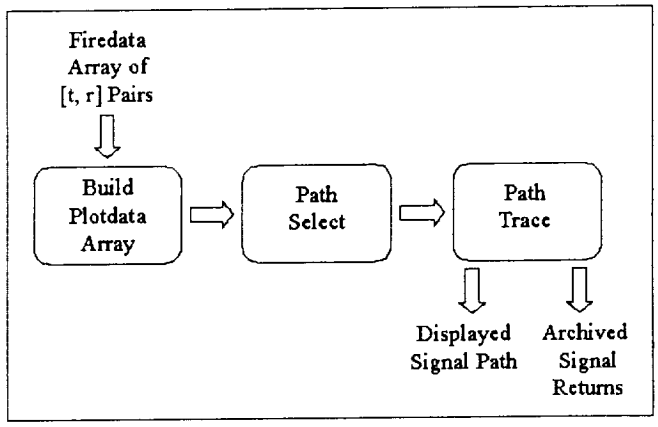

Figure 6 – RC Application Processing Flow

The RCIS then reads the plotdata array from the Firebird and displays it in a window as a scatter plot. This provides the user with a real time display of the detected surface. The RCIS also reads the firedata array from the Firebird and archives to disk the [t, r] pairs that are marked as signal via the z bit.

3.2 FPGA Image Design

The FPGA image consists of a set of core logic that performs the signal processing. This core is divided into three basic modules, the Build Plotdata Module, the Path Select Module, and the Path Trace Module, each corresponding to a specific processing step. The Plotdata Memory is also implemented in the FPGA image as a 4K x 16 dual port BlockRAM. The FPGA image also uses several generic VHDL modules provided by Annapolis Microsystems. These modules perform such functions as clock management, off-chip memory arbitration, and register interfaces to the PCI bus. The FPGA image design is illustrated in Figure 7 (shaded blocks indicate Annapolis Microsystems generic modules)

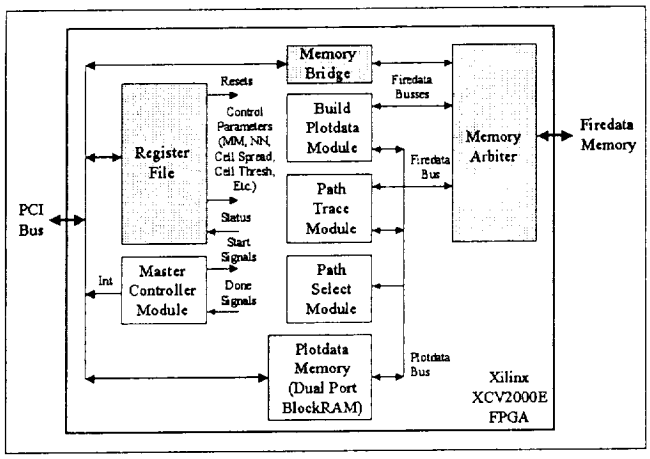

Figure 7 - FPGA Image Design

3.2.1 Build Plotdata Module

This module constructs the plotdata array by binning the [t, r] pairs in the firedata array. To perform this function, this module is further divided into four lower level modules as shown in Figure 8. The Zero Memory Module initially resets the firedata and plotdata arrays to zeros prior to processing a superframe. After this module completes, the RCIS writes the [t, r] pairs into the firedata array. The Find Min/Max Module then performs error checking on the [t, r] pairs and determines the minimum and maximum r values. The Cell Finder Module then reads [t, r] pairs from the firedata memory 16 at a time and computes the (vertical) bin and (horizontal) frame that each [t, r] pair falls into. Once the bins and frames for all 16 pairs are determined, the Update Memory Module annotates each of the 16 firedata array locations with the corresponding bin and frame numbers. For each pair, this module also increments the plotdata array location corresponding to computed bin and frame number.

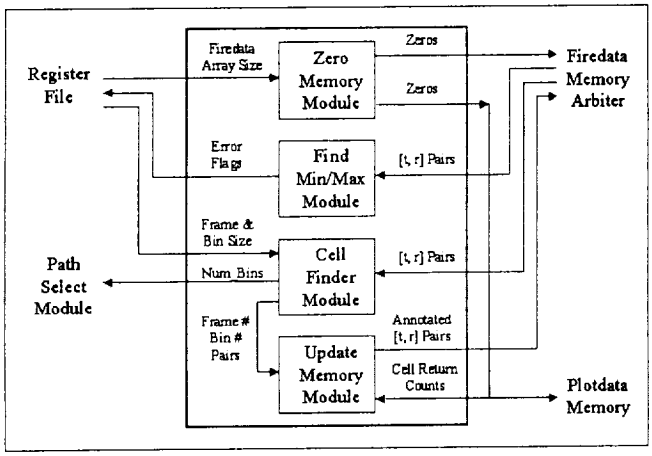

Figure 8 – Build Plotdata Module

3.2.2 Path Select Module

The Path Select Module, which iteratively scores each possible path through the plotdata array, is comprised of three lower level modules as depicted in Figure 9. The Path Counter Module generates "path vectors" that corresponds to each possible path from a single bin of the first column of the plotdata array to the last column of the array. Each 4-bit element of the path vector, consisting of 1 sign bit and 3 increment bits, determines which cell should be transitioned to from the current cell. For example, the following path vector (shown as a sequence of 4-bit hex values) would result in the following path from a given starting bin y:

Path Vector: [1, 0, 9, 9]

Path: [0, y]
[1, y+1]
[2, y+1]
[3, y]
[4, y-1]

The Path Counter Module is implemented as a series of 16 4-bit counters, each of which counts from +Cell_Spread to –Cell_Spread.

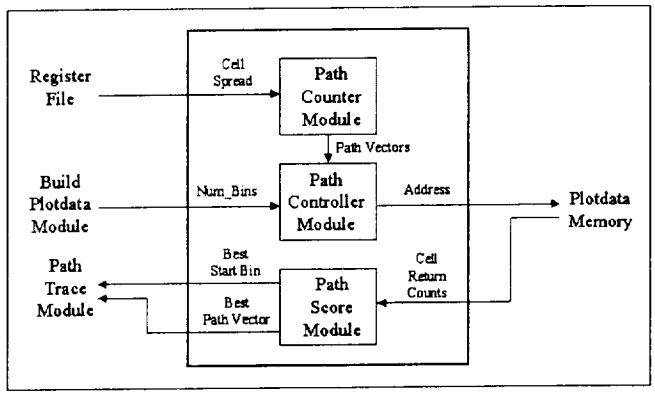

Figure 9 – Path Select Module

The Path Controller Module generates every possible path through the entire plotdata array. This is done by iteratively applying all of the path vectors from the Path Counter Module to each starting bin of the array. For each of the paths, the module then addresses corresponding plotdata memory locations and performs a read operation. This module also detects and aborts paths that exceed the boundaries of the plotdata array.

The values read from the plotdata memory are processed by the Path Score Module. For each path, this module (1) accumulates all of the return values stored in the plotdata array and (2) counts the number of values read that exceed the cell threshold. If a path has at least NN of MM cells above the cell threshold, its accumulated return value is compared against the accumulated return value of the current best path. If it exceeds this value, the current path becomes the new best path and its parameters (start bin, path vector, and accumulated return value) are stored.

After all paths are evaluated, the current best path is the

most likely signal path through the plotdata array. However, it is also possible that no signal path was found through the array. This is the case when none of paths has NN of MM cells exceeding the cell threshold.

3.2.3 Path Trace Module

The Path Select Module provides the best path to the Path Trace Module in the form of the best start bin and the best path vector. The Path Trace Module then annotates the plotdata array to indicate which cells contain signal, and annotates the firedata array to indicate which [t, r] pairs are signal returns. The configuration of this module is shown in Figure 10.

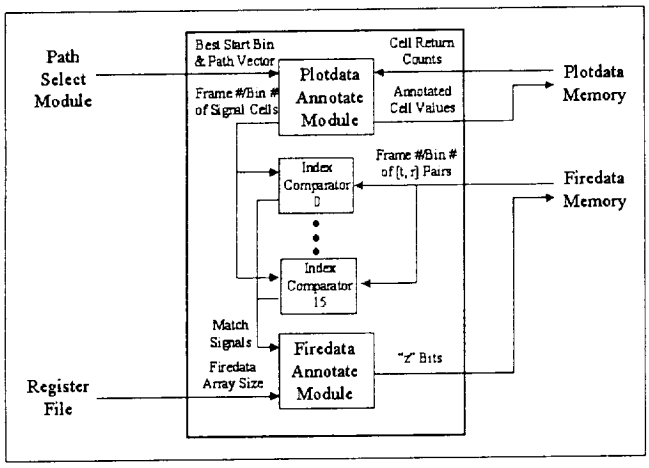

Figure 10 – Path Trace Module

This module contains a lower level module, the Plotdata Annotate Module, which generates (from the best start bin and best path vector) the bin and frame indices of each cell along the best path. These indices are then converted to the addresses of the plotdata memory locations corresponding to the cells of the best path. The module then modifies each of these locations by setting the most significant bit.

The Path Trace Module annotates the firedata memory via the lower level Firedata Annotate Module and a set of 16 Index Comparator Modules. Initially, as the Plotdata Annotate Module generates the bin and frame indices, each set of bin/frame indices are loaded into a separate Index Comparator Module. Once loaded, each Index Comparator corresponds to a cell along the selected path. The Firedata Annotate Module then sequentially reads through all locations in the firedata memory. The bin and frame number of each firedata element (originally stored by the Build Plotdata Module) is then compared in parallel with the bin/frame indices in each of the Index Comparators. If a match is detected, the Firedata Annotate Module sets the z bit in that firedata memory location.

4.0 RESULTS

The RC application has been integrated with the existing MMLA data system has been successfully ground tested. In these ground tests, the RC application operated in parallel with the software processing while ranging to a ground target. Actual flight testing during MMLA engineering check flights and subsequent science flights are upcoming.

The FPGA image currently requires roughly 15% of the resources of the XCV2000E FPGA and operates at a speed of 74 MHz. With this performance, the RC application can process 1,000,000 paths per second. Using test software, the performance of the RC application was compared the performance of software processing on a 1.2 GHz Pentium workstation for varying superframe sizes. As shown in Figure 11, the RC application does not match the performance of the software processing for values of MM less than 6. This is due to the fact that at low values of MM, a higher proportion of time is required to transfer data to the Firebird and to build the plotdata array. However, for larger values of MM, the RC application shows performance gain of 1.5 to 2.3 over the software processing.

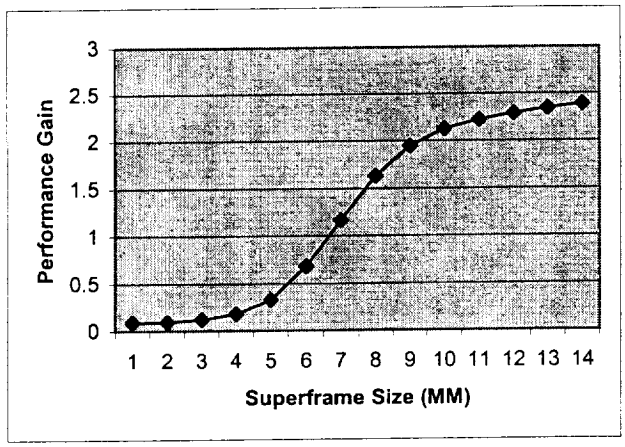

Figure 11 – Performance Gain on 1.2 GHz Pentium Workstation

As a key purpose of this IR&D project is to develop reconfigurable computing for space applications, it would be more pertinent to compare the performance against currently available spaceflight qualified processors such as the RAD6000 and the RAD750. Although the performance of the MMLA signal extraction on these processors has not been measured, a very rough comparison can be inferred by comparing their standard benchmark results. The 1.2 GHz Pentium is rated at 2230 Dhrystone MIPS, where the RAD6000 and RAD750 processors are rated at 35 MIPS and 300 MIPS respectively. Using these figures, it can be inferred that the RC application would have a performance gain of roughly 50 to 150 over the RAD6000 and a gain of 5 to 15 over the RAD750.

For the current airborne MMLA data system, the RC application shows a performance gain over the software

processing. With the laser firing at 8.5 kHz, the RC application can process data using an MM of 6, where the software can only process with an MM of 4. This indicates that the RC application processes 9 times more paths than the software processing.

5.0 CONCLUSIONS

This project has shown that reconfigurable computing offers considerable advantages over conventional software based processing of MMLA data. For the existing aircraft MMLA instrument, the RC application has provided an order of magnitude performance gain. This gain could be further improved by employing parallelism in the FPGA image.

The RC application can also provide a major performance gain for future spaceflight instruments based on the MMLA. Although the RC application currently uses an FPGA that is not flight qualified, the resources required would allow it to be implemented in the smaller, flight qualified XQVR1000 FPGA. It is assumed that parallelism could not be used for a spaceflight FPGA image due to the smaller size of the flight qualified FPGA and the additional resources needed to implement recommended SEU mitigation techniques [3].

REFERENCES

[1]. R. Conde, A. Garrison Darin, C. Dumont, P. Luers, S. Jurcyzk, N. Bergmann, A. Dawood Adaptive Instrument Module – A Reconfigurable Processor for Spacecraft Applications", MAPLD 99 Conference, September, 1999, Laurel, MD.

[2] J. Degnan, J. McGarry, T. Zagwodzki, P. Dabney, J. Geiger, R. Chabot, C. Steggerda, J. Marzouk, A. Chu "Design and Performance of an Airborne Multikilohertz Photon-Counting, Microlaser Altimeter", ISPRS Workshop on Land Surface Mapping and Characterization Using Laser Altimetry, October, 2001, Annapolis, MD.

[3] C. Carmichael "Triple Modular Redundancy Design Techniques for Virtex FPGAs", Xilinx Application Note XAPP197, June, 2001.