Mapping of a 2D SAR Backprojection Algorithm to an SRC Reconfigurable Computing MAP Processor

Peter Buxa, LeRoy Gorham, and Lt. Mathew Lukacs
Air Force Research Laboratory, Sensors Directorate
E-mail Addresses: {Peter.Buxa, Leroy.Gorham, Mathew.Lukacs}@wpafb.af.mil

David Caliga
SRC Computers
E-mail Address: caliga@srccomputers.com

1. Introduction

This paper describes the performance gain of a two-dimensional Synthetic Aperture Radar (2-D SAR) Backprojection algorithm running on a Compact MAP[1] processor compared to a MATLAB and C implementation of the algorithm. The Spotlight Synthetic Aperture Radar (SAR) Backprojection algorithm is considered to be the "gold standard" of the SAR imaging techniques. Originally written in MATLAB, the compute intensive routines of the algorithm were converted into C Language and compiled using SRC's Carte Programming Environment, which targets the MAP reconfigurable hardware. The algorithm was then benchmarked on the Compact MAP processor in order to demonstrate processor performance in a small form factor suitable for man portable or small Unmanned Air Vehicle (UAV) applications.

2. Background

A spotlight SAR image is a two (or three) dimensional mapping of received radar energy. A SAR sensor illuminates a target area with a series of linear frequency modulation pulses. The location of an individual scatterer is determined by measuring the range and doppler (range rate) and comparing this to a central reference point, called the motion compensation point. As more pulses are used, the azimuth (or cross-range) resolution increases.

There are several algorithms that have been developed to form spotlight SAR images. In deciding which algorithm to use, there is a tradeoff between computational efficiency and imaging accuracy. For instance, the simplest algorithm orders the pulses into a rectangular array and performs a two-dimensional Fourier transform. However, the resultant image will not be very accurate, as the algorithm does not compensate for scatterer motion through the synthetic aperture. The most accurate image formation algorithm is the tomographic backprojection. The backprojection algorithm calculates an exact solution for every pixel in the image. However, this approach has very high computational cost. There have been numerous algorithms developed that have acceptable accuracy with much less computational time than the backprojection algorithm. The most popular of these is the polar format algorithm.

The polar format algorithm has low computational cost, but it has some limitations that make the backprojection algorithm more attractive. For instance, when using backprojection, the user can choose any imaging grid while there is only one imaging grid available for the polar format algorithm. Also, the backprojection algorithm intrinsically allows the ability to add or subtract pulses from an image, which is a capability that is unavailable in any other imaging algorithm.

The backprojection algorithm, though highly computational, is desirable since it produces the most accurate images and can be customized depending on mission requirements. As a result, there is a need for a processing architecture that is reconfigurable and capable of real time processing while fulfilling size, weight, and power requirements for man portable or small UAV systems.

3. SRC Compact MAP Processor

In developing the Compact MAP form factor, SRC compressed its standard MAP processor by an order of magnitude down to just 6 cubic inches, while at the same time increasing its processing performance. The resulting Compact MAP contains approximately 60 million gates of user logic and has been tested to sustain a rate of 35 GFlops with 7.6 GBytes/S of sustained external bandwidth. A block diagram of the MAP along with a photograph of the Compact MAP are shown below in Figures 1 and 2.

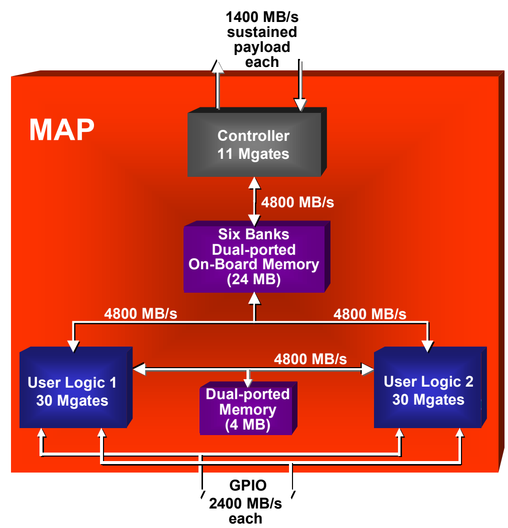

Figure 1: MAP Block Diagram

---

[1] This work was sponsored by Congress under Air Force Contract FA8650-040-C-1618

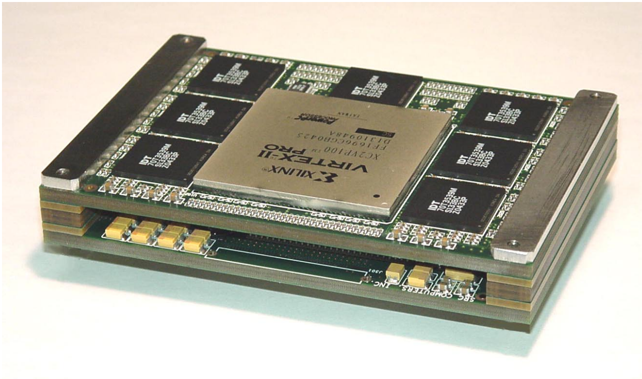

Figure 2: Compact MAP

The compact MAP can be either air-cooled or spray cooled for harsh environments. The air-cooled portable system is shown in Figure 3.

Figure 3: SRC MAPstation

4. Application Implementation

The 2-D SAR Backprojection algorithm written in MATLAB was developed by the Air Force Research Laboratory (AFRL) for prototyping and evaluation with SAR datasets. The dataset was synthetically generated using a simulated wideband (7-13 GHz) complex radar backscatter, 360 degrees around a 3-D CAD model of a backhoe shown in Figure 4.

Figure 4: 3-D CAD model of Backhoe

The 2-D dataset for this study contained one slice at 10 degrees elevation of the full 3-D dataset, resulting in 5,040 1-D projections. After computation by the 2-D SAR Backprojection algorithm, a 2-D image of 1001x1001 pixels was formed and is shown in Figure 5. Using a standard Intel 2.8GHz Pentium 4, one compute run of the full 2D dataset took 1.3 hours running MATLAB code.

The study implemented the original MATLAB code in several forms. The compute performance of each implementation was timed for comparison with the original MATLAB code.

- **MATLAB Only** The MATLAB Only implementation computation time for the imaging routine, image_3d.m, was 99.9% of the total computational time.
- **C Code Only** This implementation converted all of the original MATLAB code into the C Language. The converted imaging routine, image_3d.c, utilized the optimized Intel IPP FFT routine.
- **MATLAB – MAP** The imaging routine, image_3d.c used in the C Code Only implementation was modified to image_3d.mc, and compiled to the MAP using the Carte Programming Environment.
- **C – MAP** This implementation used the main.c from the C Code Only implementation in conjunction with the MAP imaging subroutine, image_3d.mc

Optimizations

The imaging routine, image_3d.mc, implemented on the MAP took advantage of many of the optimization techniques supported by the MAP Compiler. These optimizations included: spreading the computational array across multiple On-Board Memory Banks, using Block RAM arrays, using two User Logic Chips and overlapping DMAs with compute. The initial performance gain was 75x for the MATLAB – MAP version versus the original MATLAB version.

5. Results

The MAP performance of a compute loop when pipelined is one iteration of the loop every clock. The major consumer of computational time was the summation of the contributions of each swath to every pixel in the 2-D image, resulting in many inverse Fourier transforms, multiplies, and summations. The image summation was then split across the two User Logic FPGAs in the MAP giving an additional 2x speedup.

The microprocessor environment used in the study was an Intel 2.8 GHz P4 running Linux. The resulting speedups are shown below.

Implementation	Speedup
MATLAB Only	1x
C Only	4.2x
MATLAB – MAP	151x
C – MAP	153x

The MATLAB – MAP implementation took only 28 seconds to process all 5040 1D lines.

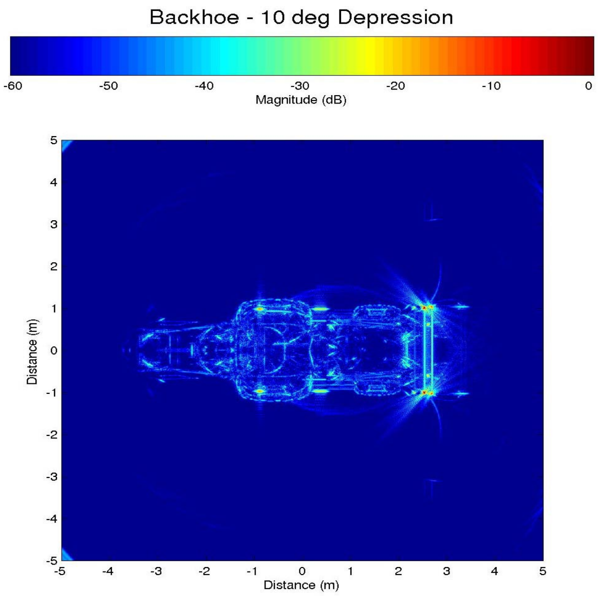

Figure 5: Computed 2D SAR Backprojection Image

Using Two MAPs

If two MAPs were to be utilized, the system would provide a super-linear speedup of the application. The first MAP could compute the scaling, FFT, and post-FFT scaling for ten swaths. These ten swath vectors would then be sent to the second MAP which would perform the image update using all ten vectors concurrently, yielding a speedup of 10x.

5. Conclusion

In this paper, we have shown the performance increase of running the SAR 2-D Backprojection algorithm on the SRC Compact MAP architecture. Until now, use of the backprojection algorithm in a real-time SAR system was difficult due to onboard processing constraints (such as size, weight, and power). With the development of SRC's Compact MAP processor, a real-time backprojection implementation is practical, even onboard small UAV systems.

References

[1] Carrara, G.W., Goodman, R.S., Majewski, R.M. *Spotlight Synthetic Aperture Radar.* p495-499, Norwood, MA: Artech House, Inc., 1995.

[2] Gorham, L., Naidu, K.D., Majumder, U., Minardi, M.A., "Backhoe 3D "Gold Standard" Image". *Algorithms for Synthetic Aperature Radar Imagery XII*, Vol. #5808, SPIE DSS, Orlando, FL, March 28 – 31, 2005.