Implementation of Radar Signal Processor for Stripmap mode of Synthetic Aperture Radar (SAR) on Multiprocessor Hardware

Mahesh Kopp, Akilan Janakirajan, Shayer Dudekula, Peter Joseph Basil Morris, Thiruchelvi M, Pandiri Jhansi Sreeveni, Anju Rao M and Ramakrishnan S

Electronics & Radar Development Establishment, DRDO
mahesh.kopp@lrde.drdo.in

Abstract:

The Synthetic Aperture Radar (SAR) developed for Unmanned Aerial Vehicle (UAV) is medium range radar, for ground imaging and ground moving target detection. The primary mode of operation of SAR is in imaging mode namely Stripmap mode having a coarser resolution of the order of 6m and 3m and Spot mode having a finer resolution of the order of 1m and 0.6m. SAR generates images of ground area in the specified modes and provides display of the area imaged with geo-coded information. The radar is designed and developed in Ku band applicable for range of operation in the order of 10 – 40 Km.

The Stripmap processing involves Azimuth pre-processing, Range Compression, Platform Motion Compensation (MOCO), Azimuth Compression followed by Phase Gradient Autofocus algorithm. This algorithm is implemented on a state of the art Power PC boards with multi-processor architecture.

Key Words: Stripmap, MOCO, RDA, UAV, SAR

I. INTRODUCTION

Synthetic Aperture Radar is all-weather, day-night microwave imaging system. The major advantage of the SAR system is that it provides same resolution irrespective of the illumination range, as the resolution is independent of the range. SAR uses the pulse compression technique (either through matched filtering or deramp-on-receive technique) for achieving the required resolution in the range. In azimuth, it utilizes the motion of the aircraft to synthesize a very long antenna array wherein the azimuth resolution becomes independent of range of the target and wavelength.

SAR signal processing presents a significant computational challenge due to the real-time processing requirements due to high data size at the higher rates. This requires high performance computing platforms like MultiCore processors with AltiVec support.

This paper discusses the adapted signal processing algorithms along with implementation on the hardware.

The processing algorithms for stripmap are explained in section II. The hardware and software architectures used in implementing the algorithm are given in section III and section IV respectively. The implementation strategies and results are given in section IV and Section V respectively.

II. STRIPMAP SIGNAL PROCESSING ALGORITHM

Continuous mapping of a fixed-width land strip, parallel to the aircraft path is performed in Stripmap mode. The strip-map mode is used for large area mapping. In this mode, imaging is performed while Line Of Sight is perpendicular to the flight route. The Signal Processor (SP) receives the radar returns periodically on each Pulse Repetition Interval (PRI) and processes this data to generate the continuous image of strip of the ground. Various algorithms are available for processing strip map mode SAR data. Range Doppler Algorithm (RDA) [1] with 2-stage motion compensation [3] is used in the current implementation due to its inherent advantages. It is developed to generate the 2D SAR images using independent one-dimensional operation (compression) on range and azimuth. The compression in range and azimuth direction is implemented through matched filtering using fast convolution. The major processing modules of RDA are given in [1].

III. OVERVIEW OF MULTIPROCESSOR HARDWARE ARCHITECTURE

The hardware consists of two HCD6220 boards which are interconnected over LAN, SRIO and PCIe. Each board consists of two MPC8640D dual core PowerPC processors which run on VxWorks operating system. All the cores are clustered [4] and are interconnected by SRIO. This

interconnect is intended primarily as board-to-board and chip-to-chip communication at gigabytes per second performance levels. Board 1 carries one data acquisition XMC module and one Memory XMC module. Board 2 carries one 1553B PMC module and one SFPDP XMC module. We have used SAL [2] libraries and C programming language for implementation.

IV. SIGNAL PROCESSOR SOFTWARE ARCHITECTURE

Seven processing cores in the hardware are used for SP software. Out of 7 cores, one core of the 8640D processor on the board containing the ADC-FPGA XMC module is configured as Signal Processor Controller (SPC), which is used for interfacing with DAP firmware to get the Digital Down Converted (DDC) data and interfacing with Radar Controller (RC) to get the control messages and platform parameters. One core of the 8640D processor on the board containing SFPDP XMC module is configured as Signal Processor Dispatcher (SPD), which is used for sending data to Data Recorder as well as sending processed SP output to RC. Rest of the 5 processing cores is used as Compute Elements (CEs) for executing signal processing algorithms. Identical processing chain is running in all the CEs.

The resource allocation for SP is described the Figure 1

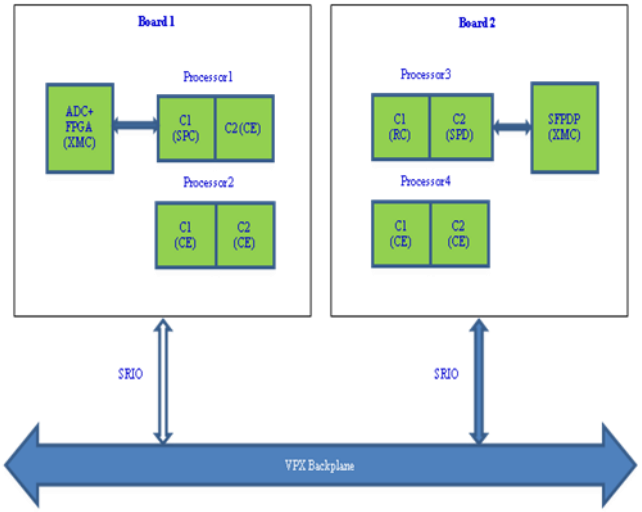

Figure 1: Resource allocation for SP

The data size collected per pulse is fixed for a given mode of SAR operation. DAP firmware continuously collects and DMAs data to SPC using double buffer model to avoid data discrepancy. SPC takes the data and DMAs it to CEs memory for processing. Consecutive batches of data are dispatched to the CEs in round robin fashion. The architecture used for signal processing is pipelined Single Instruction Multiple Data (SIMD) architecture.

The total time available for processing is enhanced by a factor equal to number of CEs available for processing. It is also worthwhile to note that an initial latency equal to the processing time of one batch of data is the price paid for this pipelined architecture design. The processed data outputs are dispatched in intervals of data collection time corresponding to one batch of data. We have used shared memory as Inter Process Communication mechanism between SPC, CE, SPD and semaphores for synchronization between them.

The implemented architecture has the following advantages:

1) *S/W design less complex*: Because there are two categories of processing elements, the communications activities inside the subsystem are virtually between two processors and hence less complicated to design.

2) *Fault tolerance in Compute Elements*: As the SPD keeps track of the CEs it is easily possible to take care of any fault occurring in the CE.

3) *H/W Expansion of CE's easy*: It is easy to expand the computational capability, by adding more cards and just configuring the SPD to utilize them.

4) *Additions of new algorithms do not affect data/control flow:* Any new algorithms to be added do not call for any change in data/control flow.

5) *S/W modules can be reused across processing modes:* All the CEs have the capability to do all the processing functions. Hence any common functions in different processing modes can reuse the code.

V. IMPLEMENTATION

1. Optimization Strategies

1) *Data Organization:* Radar data returns are 2-dimensional. By Organizing data as one dimensional arrays we have enabled faster element access using simple pointer increments

2) *AltiVec operations:* vector signal processing capabilities of the hardware has been leveraged to obtain performance benefits.

3) *Cache Management:* Keeping the data and intermediate results in the L1 and L2 caches as long as possible and use of effective cache management strategies like strip mining wherein a strip of data is pulled into the super-fast L1 cache, operated on till all the required operations have been completed and then finally loading the next strip of data for processing.

2. Implementation

The processing dimension of data is important for obtaining optimal performance for any processor hardware with cache. Radar data is two dimensional viz. fast-time and slow-time. The radar data is arranged in the processing dimension of the algorithm to reduce the number of cache misses. The radar data is processed in batches of pulses. The platform data from RC comes at 100 msec, which is much slower than the PRI. For doing MOCO, the platform data is interpolated to PRI rate for getting better image focusing.

The implementation of the various algorithms is discussed below.

1) Motion Compensation 1

The MOCO1 vector is calculated for each and every pulse with the ΔRref [3] calculated for that pulse. By complex multiplication of the vector with the input pulse data, the amount of error present in the centre range of the swath is corrected for all targets in the swath. The correction is applied for all pulses in the same batch.

2) Range Compression

The function performs matched filtering on the Motion compensated data samples along the fast-time dimension. The algorithm is implemented by fast convolution using the Fast Fourier Transform (FFT) libraries [2]. A weighted array of twiddle factors is setup prior to performing the FFT operation. The same twiddle factor array could be used for consecutive FFT operations to reduce the processing time. The algorithm is performed on every pulse data along the fast time (range) dimension using vector signal processing libraries.

3) Motion Compensation 2

This corrects the residual phase errors [3] remaining in the range bins other than the centre bin (corrected in MOCO1). This correction is different for each range bin. For each range bin, complex phase vector is computed and multiplied with the range compressed data.

4) Azimuth Compression

The azimuth compression is implemented by fast convolution using the FFT libraries [2]. The azimuth reference function is computed based on the range of operation and velocity of the platform. The velocity of the platform can't be uniform across all batches of data. For this reason the azimuth reference function is recomputed for each batch of data by considering the mean velocity of the platform for that particular batch.

5) Phase Gradient Auto-focus (PGA)

PGA is performed on the compressed image to remove the residual phase errors left after MOCO I and MOCO II. The fundamental concept involved is the estimation of the derivative (gradient) of the phase error using only the complex degraded SAR image. The estimation process exploits redundancy of phase error information contained in the degraded image irrespective of the scene content. The output of the algorithm is a phase correction vector, which is used for iterative correction of the image data.

VI. RESULTS

The developed signal processor software is tested for its functionalities and performance in the lab with internal and external BITE. The same is evaluated for its performance by installing the radar on Dornier aircraft. The results of the same are shown below.

The details of the stripmap image for 6m resolution are given table 1. In Figure 2 the satellite image and in the Figure 3 the equivalent SAR image is given.

Table 1: Stripmap image 1 characteristics

Resolution	6m X 6m
Range swath	6 km
Azimuth Swath	6km

The details of the stripmap image for 3m resolution are given table 2. In Figure 4 the satellite image and in the Figure 5 the equivalent SAR image is given.

Table 2: Stripmap image 2 characteristics

Resolution	3m X 3m
Range swath	3 km
Azimuth Swath	3km

Figure 2: Satellite image of the region of interest

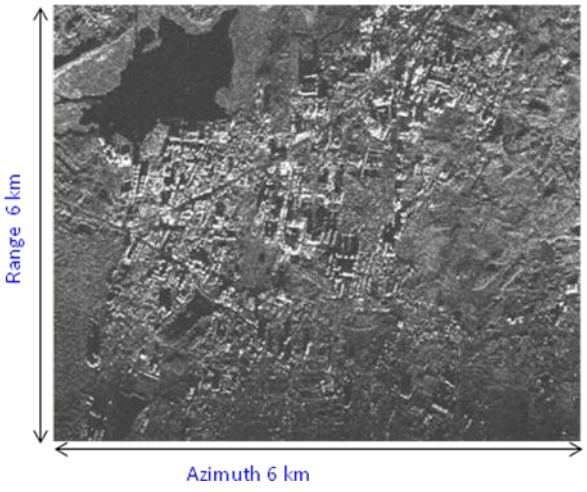

Figure 3: SAR image of the region of interest

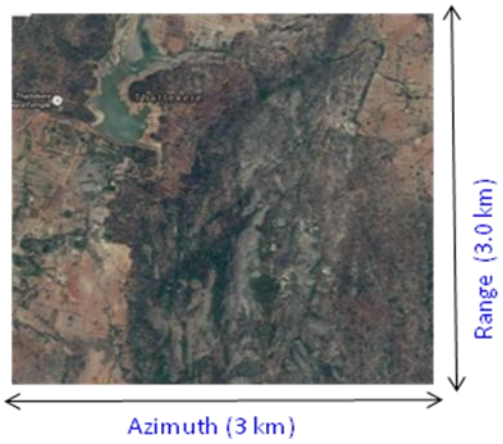

Figure 4: Satellite image of the region of interest

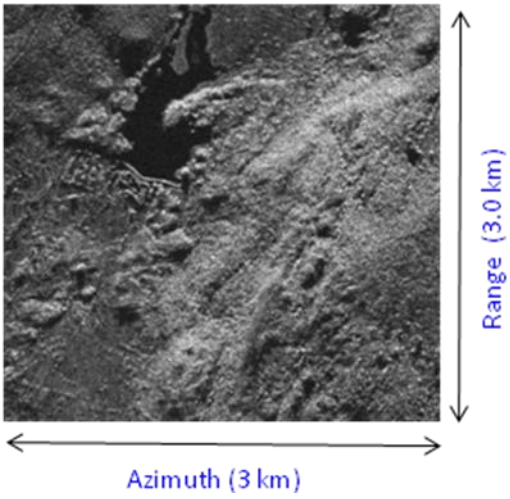

Figure 5: SAR image of the region of interest

VII. CONCLUSION

This paper presented the realization of a radar signal processor for the Stripmap mode of operation of SAR. The paper has primarily focused implementation of the RDA algorithm for Stripmap mode of operation on multiprocessor hardware and the optimization strategies adopted to reduce the execution time and improving the throughput of the system by using pipelined SIMD design. A significant improvement in performance was achieved due to this. Utilization of hardware specific optimization strategies like strip-mining and cache management has further enhanced the throughput of the designed solution. In short, by proper selection of signal processing algorithms and the multiprocessor hardware coupled with the efficient implementation and optimization strategies has led to a

considerable improvement in the performance of the realized signal processor solution.

REFERENCES

[1] Cumming.I.G., Wong.F.H., "Digital Processing of Synthetic Aperture Radar Data Algorithms and implementation" Artech House Publ.

[2] Scientific Algorithm Library (SAL) Reference Manual, Mercury Computer Systems, 2011.

[3] Evan C. Zaugg and David G. Long, "Improved SAR Motion Compensation without Interpolation"

[4] MultiCore Plus ICS Reference, Mercury Computer Systems, 2011.

BIO DATA OF AUTHOR(S)

Mahesh Kopp is Scientist E in LRDE, DRDO Bangalore. He is System Integration Manager for SAR system.

Akilan Janakirajan is Scientist 'C' in ADE, DRDO Bangalore. He has been is working in the field of Radar Signal Processing for Airborne Radar Systems.

Shayer Dudekula is Scientist 'C' in LRDE, DRDO Bangalore. He has been working in the field of software development, integration and testing of Radar Signal Processing for Airborne Radar Systems

Peter Joseph Basil Morris is Scientist 'C' in LRDE, DRDO Bangalore. He has been working in the field of Radar Signal Processing for Airborne Radar Systems.

Thiruchelvi M is Scientist 'D' in LRDE, DRDO Bangalore. She has been working in the field of Radar Signal Processing for Airborne Radar Systems.

Anju Rao M is Scientist 'E' in LRDE, DRDO Bangalore. She has been working in the field of Radar Signal Processing for Airborne Radar Systems.

Pandiri Jhansi Sreeveni is Scientist E in LRDE, DRDO Bangalore. She is working as a System Engineer of SAR.

Ramakrishnan S is Scientist F in LRDE, DRDO Bangalore. Currently he is Project Director for Synthetic Aperture Radar. His main area of interests includes Active Array radars, Radar target Simulator and SAR.