RapidIO-based Space System Architectures for Synthetic Aperture Radar and Ground Moving Target Indicator

David Bueno, Chris Conger, Adam Leko, Ian Troxel, and Alan D. George
{bueno, conger, leko, troxel, george}@hcs.ufl.edu

High-performance Computing and Simulation (HCS) Research Laboratory
Department of Electrical and Computer Engineering, University of Florida

Abstract

The design of space systems capable of performing real-time Synthetic Aperture Radar (SAR) is a significant challenge in HPEC due to the high processor, memory, and network requirements imposed by SAR. However, building a system to support SAR and other Space-Based Radar (SBR) algorithms simultaneously is an even greater challenge. This presentation describes simulation results from a SAR application executing in parallel on an embedded multiprocessor satellite system equipped with a RapidIO interconnection network. We consider several variations on RapidIO network parameters as well as parallel decompositions of the SBR algorithms to design a system to support SAR and Ground Moving Target Indicator (GMTI) simultaneously. In addition, we describe our RapidIO testbed that is used to validate our network models and present results demonstrating the accuracy of our network models under varying traffic conditions.

Introduction

The large dataset sizes and real-time requirements of SBR applications demand a high-performance interconnect architecture capable of meeting throughput requirements on the order of several gigabits per second. In previous work [1], we presented simulation results for systems performing GMTI in parallel on system architectures built around the RapidIO embedded systems interconnect. In this presentation, we significantly extend our previous work with new simulations of a parallel SAR application executing on a RapidIO-based system. RapidIO is an emerging open standard for high-speed, embedded packet-switched interconnection networks which supports data rates up to approximately 60 Gbps. RapidIO is the latest commercial-off-the-shelf (COTS) technology to be considered practical for inclusion in military embedded networks to improve cost-effectiveness and scalability. Moving from bus designs to switched interconnects substantially increases the cost-effectiveness, robustness, and network performance of future embedded systems.

SAR has recently gained popularity for applications such as remote sensing, surveillance, and target recognition [2]. SAR contains some high-level similarities to GMTI, such as the ability to break processing into stages and the repartitioning of data between different stages [3-4]. However, there are also distinct and important differences, which provide an interesting comparison of system performance and limitations for systems built to support both GMTI and SAR. In addition to significant computational changes, examples of major differences of SAR vs. GMTI include much larger memory requirements and therefore associated adjustments in data movement strategies, as well as relaxed time constraints, allowing significantly more time for processing of SAR images relative to GMTI data cubes. For both SAR and GMTI, the processing deadline for a single dataset is referred to as a Coherent Processing Interval (CPI). While our GMTI algorithm mandates a 256 ms CPI, SAR allows up to 16 seconds per CPI. To provide a more fair and thorough comparison of system performance for the different SBR algorithms, we also present results for a new parallel partitioning for GMTI which more closely matches the strategy of our SAR algorithm, which complements the three GMTI partitionings presented in [1] and [5].

Computer-based simulation provides a cost-effective tool for evaluating future satellite systems for SBR and other payload-processing applications. Our discrete-event simulation models are built and simulated using the commercial simulator MLDesigner from MLDesign Technologies. Our virtual prototyping environment for RapidIO systems incorporates moderate-fidelity systems and components including RapidIO switches, end-points, and processor models. To ensure that our simulation results are accurate and instill confidence in our results, we have created a RapidIO testbed using Xilinx RapidIO cores, FPGAs, and development boards.

RapidIO Testbed for Model Validation

Our RapidIO testbed currently consists of two RapidIO endpoints connected via an 8-bit parallel LDVS, 250 MHz RapidIO interface. The RapidIO endpoints are implemented on Xilinx development boards each containing a Virtex-II Pro FPGA, with plans for expansion to include one or more Tundra 4-port, 250 MHz RapidIO switches. We measure RapidIO throughput and packet latencies under varying traffic conditions and calibrate our models to closely match the performance of the RapidIO testbed. Due to space and time limitations, results are not included here. The formal presentation will contain a variety of validation results over a range of RapidIO packet types, sizes, and traffic conditions.

Experimental Setup

Our baseline SAR algorithm processes a 2D data image of size 2 GB, where each data element is a 64-bit complex integer. The high memory requirements of SAR images dictate a different parallelization approach from the three methods used for partitioning of GMTI in [1]. Instead of directly spreading out the entire image over the system nodes in some fashion, SAR requires the entire image be stored in and processed out of a global memory board.

The data image is then inputted in "chunks" by the processing nodes, where each then processes a chunk, writes it back to global memory, reads in another chunk, and repeats. Rather than explicit "corner turns" to redistribute data between stages, data redistribution occurs implicitly as nodes read in data from global memory partitioned along the appropriate dimension for the stage whose chunks are being processed. The disadvantage of this approach is that redundant data communication frequently occurs, as each data item must be written to global memory after each stage and then read in again during the subsequent stage.

A complete description of all system designs and network tradeoffs performed in the course of this study will appear in the full presentation. Due to space limitations here, a condensed version follows. RapidIO system parameters are similar to those used in [1] and [5]. Our baseline system backplane is a composed of four 8-port RapidIO switches and provides non-blocking connections for up to eight boards (including a global memory board), with four dedicated RapidIO links between the backplane and each board. Each processor board has an 8-port switch and four compute nodes.

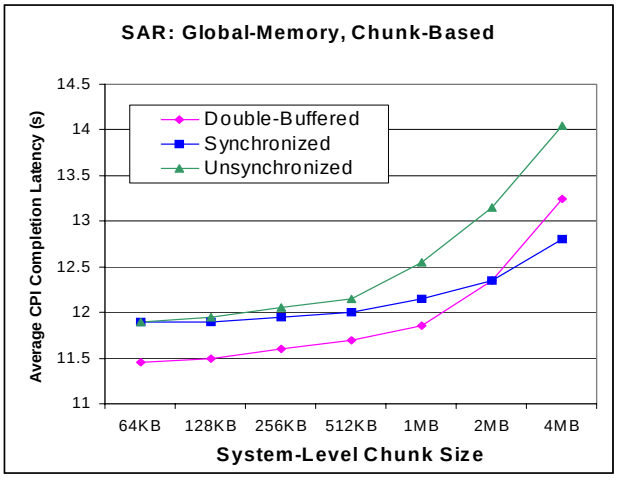

Figure 1- Variants of Parallel SAR Algorithm

Results

Figure 1 shows a summary of CPI latency results for three different variants of the parallel SAR algorithm on this multiprocessor platform. For the unsynchronized case, network contention severely hampers performance as chunk size increases. This contention is largely due to multiple nodes requesting transactions from the same global memory port at the same time. The synchronized case uses a token to only allow one node to attempt to read from global memory at a time, creating more efficient network usage and improving performance, especially for large chunk sizes. The double-buffered case assumes that nodes possess sufficient memory to store and process a chunk in memory while receiving the subsequent chunk. Double buffering achieves the best performance for small chunk sizes, but introduces even higher levels of contention for larger chunk sizes. A broad array of results describing additional algorithm approaches and system design tradeoffs will be included in the final presentation, including SAR vs. GMTI, but are omitted in this abstract.

Conclusions

The incorporation of RapidIO in future satellite payload processing systems is likely to improve performance as well as cost-effectiveness of embedded SBR platforms. This new work builds off of our previous work in modeling and simulation of RapidIO and GMTI, demonstrating that effective, real-time SAR processing can be performed over a RapidIO-based network with HPEC onboard a satellite. Results show the importance of intelligently synchronizing communications over the RapidIO network and demonstrate significant performance gains obtainable by double-buffering of data cube chunks. Our work shows that RapidIO-based switched interconnect designs have the potential to far outperform traditional bus designs in embedded systems, allowing real-time GMTI and SAR processing to occur on-board a satellite. Future directions for this work will include the study of fault-tolerant RapidIO-based space system architectures for SBR and other payload-processing applications. Future experimental and simulative work will also feature hardware-reconfigurable computing devices (e.g. FPGAs) working alongside general-purpose processors and other computing devices connected via a RapidIO fabric.

Acknowledgements

We wish to thank Honeywell Defense and Space Electronic Systems (DSES) in Clearwater, FL for support of this research. We also extend thanks to Xilinx for their generous donation of hardware and IP cores, as well as MLDesign Tech. for donation of simulation software.

References

1. D. Bueno, C. Conger, A. Leko, I. Troxel, and A. George, "Virtual Prototyping and Performance Analysis of RapidIO-based System Architectures for Space-Based Radar," *Proc. High-Performance Embedded Computing (HPEC) Workshop*, MIT Lincoln Lab, Lexington, MA, Sep. 28-30, 2004.
2. C. Cho, "Performance Analysis for Synthetic Aperture Radar Target Classification," MSEE Thesis, Mass. Institute of Technology, Feb. 2001.
3. P. Meisl, M. Ito, and I. Cumming, "Parallel Synthetic Aperture Radar Processing on Workstation Networks," *Proc. 10th International Parallel Processing Symp.*, Apr. 15-29, 1996, pp. 716-723.
4. S. Plimpton, G. Mastin, and D. Ghiglia, "Synthetic Aperture Radar Image Processing on Parallel Supercomputers," *Proc. 1991 ACM/IEEE Conf. on Supercomp.*, Albuquerque, NM, 1991, pp. 446-452.
5. D. Bueno, A. Leko, C. Conger, I. Troxel, and A. George, "Simulative Analysis of the RapidIO Embedded Interconnect Architecture for Real-Time, Network-Intensive Applications," *Proc. 29th IEEE Conf. on Local Computer Networks (LCN)* via the *IEEE Workshop on High-Speed Local Networks (HSLN)*, Tampa, FL, Nov. 16-18, 2004.