# Virtual Prototyping and Performance Analysis of RapidIO-based System Architectures for Space-Based Radar

***David Bueno, Chris Conger, Adam Leko, Ian Troxel*** and ***Alan D. George***
University of Florida
Phone: 352-392-9034
Email Addresses: {bueno, conger, leko, troxel, george}@hcs.ufl.edu

## Abstract

*Space-Based Radar (SBR) processing is a processor- and communication-intensive HPEC application that presents unique design challenges. This talk will concentrate on the presentation of simulation results of mapping a parallel Ground Moving Target Indicator (GMTI) application on an embedded multiprocessor satellite processing system featuring a RapidIO interconnection network. We consider three partitionings of a real-time GMTI algorithm executed on systems of different sizes and topologies. Each partitioning's system performance and algorithm scalability on various RapidIO systems is examined.*

## Introduction

RapidIO is an emerging open standard for high-speed, embedded switched interconnection networks which supports data rates up to approximately 60 Gbps. It is an open standard [1, 2] steered by a non-profit organization known as the RapidIO Trade Association. RapidIO uses Low-Voltage Differential Signaling (LVDS) to minimize power usage at high clock speeds, and therefore is appropriate for use in HPEC systems. 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 will substantially increase the cost-effectiveness, robustness and raw network performance of future embedded systems.

GMTI is an important application in military operations, since moving targets may be laid over a map of a battlefield for strategic planning during a conflict. GMTI works best when combined with some form of airborne radar system. Since space is the ultimate "high ground" for radar systems, having GMTI available in an SBR system is advantageous. The challenge for HPEC systems is to provide real-time data in-system with minimal latency. In traditional air-based GMTI systems, a high-performance cluster of workstations is used to process incoming radar data [3]. As GMTI requires costly adaptive processing (including Space-Time Adaptive Processing or STAP [4]) of high-resolution data, the algorithm imposes severe processing and communication challenges on space-based embedded systems with strict power, size, weight, and radiation constraints.

In order to effectively design RapidIO-based architectures, it is essential to fully understand RapidIO's strengths and weaknesses. A simulation-based testbed provides an ideal environment for performing tradeoff studies on RapidIO's salient features; therefore, we developed a simulation environment to prototype and evaluate RapidIO-based multiprocessor satellite systems for Space-Based Radar (SBR) applications within our discrete-event simulator of choice, Mission-Level Designer [5]. Our RapidIO prototyping environment incorporates moderate-fidelity systems and components including RapidIO switches, end-points and processor models. This prototyping environment has been used to predict the performance of future RapidIO space-based GMTI systems, as well as examine possible power and scalability limitations the technology may impose.

## Experimental Setup

We have started with a baseline GMTI algorithm and have employed three different decomposition strategies, including a “straightforward” approach, a staggered approach, and a parallel-pipelined approach. The straightforward approach maps the incoming data set for processing equally across each of the available processors. Since the GMTI algorithm is typically composed of signal processing procedures with no interprocessor communication during each stage, the algorithm can be considered embarrassingly parallel. The staggered partitioning method is based on the approach described in [6]. This approach is similar to the straightforward mapping approach, except incoming data is sent to groups of processors in a staggered fashion. Under this approach, each processor receives a larger amount of data to process at a time, but receives this data less frequently. Our parallel-pipelined approach is a simplified version of one presented in [7], adapted to fit our vector-based processor models and RapidIO interconnection network. We split the

pipeline into four stages, with specific groups of processors in the system dedicated to each 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 page limitations, a condensed version follows. The sensors created new image data at a rate of 4.6 Gbps. Processors were modeled to perform all or a subsection of the GMTI algorithm as the partitioning warranted. RapidIO-related parameters for endpoints include a 250MHz physical-layer clock rate, input/output buffer sizes of 8 packets, and physical-layer link width of 16 bits, among others. RapidIO switch model parameters include an average memory read/write latency of 72 ns and a central memory size of 10000 bytes, among others.

## Results

Figure 1 shows a summary of the results of each of the three partitioning strategies executed on systems of different sizes. The results show a system of 24 nodes is required to meet the application processing requirements of one coherent processing interval (CPI) per 256ms (denoted by the horizontal bar in the figure). The straightforward partitioning method provides the best raw performance on the RapidIO system, but the pipelining partitioning method may provide a more cost-effective strategy if individual processors for each GMTI step can be produced in a less expensive manner than an all-inclusive design. The staggered approach did not perform as well as the other two due to communication inefficiency. A broad array of additional results describing system design tradeoffs will be included in the final presentation but are omitted here due to space limitations.

## Conclusions

The inclusion of RapidIO in future satellite payload processing systems is likely to improve performance as well as cost effectiveness of embedded SBR platforms. In order to prototype and predict the performance of future RapidIO space-based GMTI systems, simulation models were designed and developed using the Mission-Level Designer discrete-event simulator. Several systems, RapidIO versions, and GMTI decompositions were developed on which a tradeoff analysis was performed. The results showed a 24-processor solution met the algorithm's real-time requirements. The straightforward partitioning method provides the best raw performance on the RapidIO system, but the pipelining partitioning method may provide a more cost-effective strategy for some projects. RapidIO and other COTS-based switched interconnect designs have the potential to outperform traditional bus designs in embedded systems.

Future directions for this work may include mapping other SBR algorithms with different processing characteristics such as Synthetic Aperture Radar (SAR). In addition, now that we have an initial prototyping environment developed, we plan to examine other RapidIO-specific design considerations and system development options.

## Acknowledgements

We wish to thank Honeywell Space Systems in Clearwater, FL for their funding and technical guidance in support of this research.

## References

[1] "RapidIO Interconnect Specification (Parts I-IV)," RapidIO Trade Association, June 2002.

[2] "RapidIO Interconnect Specification, Part VI: Physical Layer 1x/4x LP-Serial Specification." RapidIO Trade Association, June 2002.

[3] M. Linderman and R. Linderman, "Real-Time STAP Demonstration on an Embedded High Performance Computer," Proc. of the IEEE National Radar Conference, Syracuse, NY, May 13-15, 1997.

[4] "Space-Time Adaptive Processing for Airborne Radar," Tech. Rep. 1015, MIT Lincoln Laboratory, 1994.

[5] G. Schorcht, I. Troxel, K. Farhangian, P. Unger, D. Zinn, C. Mick, A. George, and H. Salzwedel, "System-Level Simulation Modeling with MLDesigner," Proc. of 11th IEEE/ACM International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems (MASCOTS), Orlando, FL, October 12-15, 2003.

[6] R. Brown and R. Linderman, "Algorithm Development for an Airborne Real-Time STAP Demonsttration," Proc. of the IEEE National Radar Conference, Syracuse, NY, May 13-15, 1997.

[7] A. Choudhary, W. Liao, D. Weiner, P. Varshney, R. Linderman, M. Linderman, and R. Brown, "Design, Implementation and Evaluation of Parallel Pipelined STAP on Parallel Computers," *IEEE Trans. on Aerospace and Electrical Systems*, vol. 36, pp 528-548, April 2000.

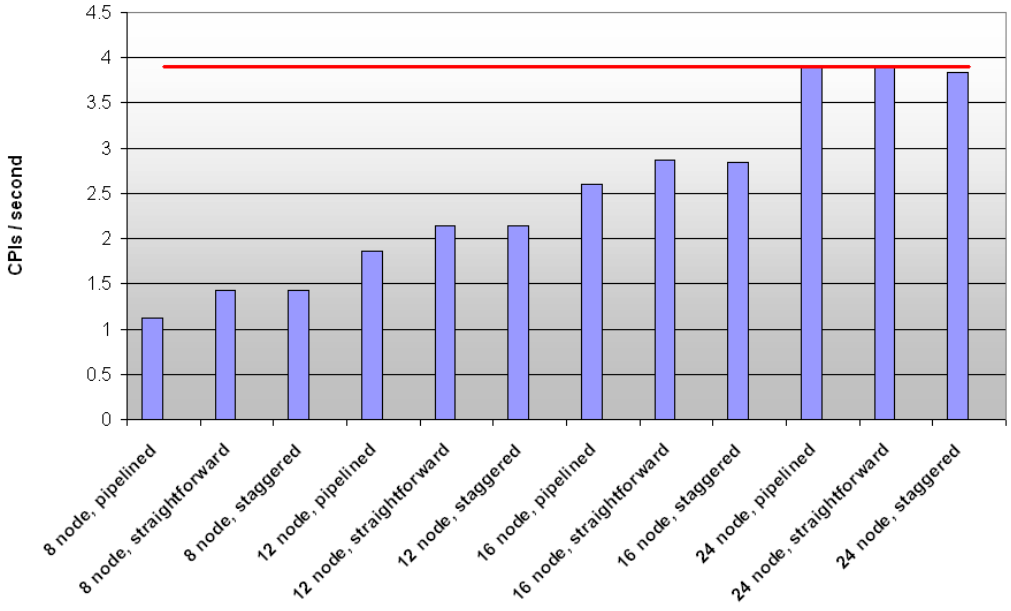


Figure 1: System throughput