A Hybrid FPGA/DSP/GPP Prototype Architecture for SAR and STAP

Jack M. West, Hongping Li, Sirirut Vanichayobon, Jeffrey T. Muehring,
John K. Antonio, and Sudarshan K. Dhall

School of Computer Science
University of Oklahoma
200 Felgar Street
Norman, OK 73019-6151
Phone: 405-325-7859
Fax: 405-325-4044
antonio@ou.edu

HPEC 2000
The Fourth Annual Workshop on High-Performance Embedded Computing

Abstract

A prototype system is described that demonstrates the advantages and trade-offs associated with the combined use of different hardware technologies for two embedded radar processing applications. The primary metrics of interest are size, weight, and power utilizations. The system can be configured with FPGAs (field programmable gate arrays), DSPs (digital signal processors), and/or GPPs (general purpose processors). The two radar applications evaluated are SAR (synthetic aperture radar) and STAP (space-time adaptive processing). Although the prototype system is not evaluated through actual fielded studies, experiments involving continuous input streams at relatively high rates are conducted in the laboratory using stored and unprocessed radar data as input.

The FPGA components of the prototype system are commercially available WildOne and WildForce boards (from Annapolis Microsystems) populated with 4000-series Xilinx parts. The WildForce boards each have four 4085-series FPGAs plus one control FPGA. The DSP/GPP components of the system are within a Mercury Race Multicomputer configured with both SHARC and PowerPC compute nodes. The Mercury system can be configured with up to eight PowerPC nodes and eight SHARC compute nodes (each SHARC compute node actually contains three SHARC DSP chips). An overview of the overall architecture is depicted in Figure 1.

The source PC is responsible for initially loading unprocessed radar data (from disk) into a circular buffer within its main memory. Once the input data is loaded into the circular buffer, the source PC then continuously (and repeatedly) streams this data into the front-end FPGA subsystem, denoted as (F) in Figure 1. It was necessary to locate the input data in a large main memory buffer in order to achieve realistic data throughput rates, which would otherwise not be possible if the data were streamed directly from the disk of the source PC.

All of the Annapolis FPGA boards are PCI-based and reside on the data source and/or data sink PCs. A total of four WildForce boards are available, and zero or more of these may reside on the source and sink PCs. The source and sink PCs also contain one WildOne board each. The WildOne boards are not used for computation, but handle the

data communication (through the PCI bus) between the PCs and the FPGA subsystems. The data communication among all FPGA boards is through two types of 36-bit wide connectors, one called systolic and one called SIMD.

The data communication between the front-end FPGA subsystem (F) and the DSP/GPP subsystem is a custom interface developed using the systolic connector from Annapolis and the RIN-T input device from Mercury. Similarly, the data communication between the DSP/GPP subsystem and the back-end FPGA subsystem (B) is through a custom interface developed using the ROUT-T output device from Mercury and the systolic connector from Annapolis.

Figures 2 and 3 illustrate how the major computational components of the SAR and STAP applications can be mapped onto the prototype system. A candidate mapping is defined by assigning the computations of each major component to one or both of the symbols shown in each block (which correspond to one of the FPGA or DSP/GPP subsystems). Using SAR to illustrate, one mapping would be to perform all of the range compression on the front-end FPGA subsystem (F) and then perform all azimuth processing on the DSP/GPP subsystem. Another possible mapping is defined by using the FPGA subsystems and the DSP/GPP for both components of computation. It is also possible to use only the DSP/GPP subsystem for both components of computations.

The SAR studies were designed by adapting the RASSP (Rapid Prototyping of Application Specific Signal Processors) benchmark developed originally by Lincoln Laboratory at MIT. The benchmark, which was originally implemented in serial C code, was first modified to execute on the parallel DSP/GPP subsystem. A data-streaming component was also added so that input data can be sent continuously from the data source of the prototype system. Core computations from the range compression and azimuth processing components were implemented for the FPGA subsystems.

The STAP studies were designed by adapting the RT_STAP (Real Time STAP) benchmark developed originally at MITRE. This benchmark was already implemented for parallel execution on a PowerPC-based Mercury system. This implementation was expanded to also enable execution on SHARC compute nodes. The same basic data streaming component that was developed for SAR was also adapted to enable the STAP input data to be sent continuously from the data source. Finally, core computations from the range compression and weight computation components from the STAP processing flow were implemented for the FPGA subsystems.

The size, weight, and power utilizations of various mappings and problem instances are under investigation. Initial indications are that heterogeneous configurations, which utilize two or more hardware technologies of the prototype system, are preferred over homogeneous configurations.

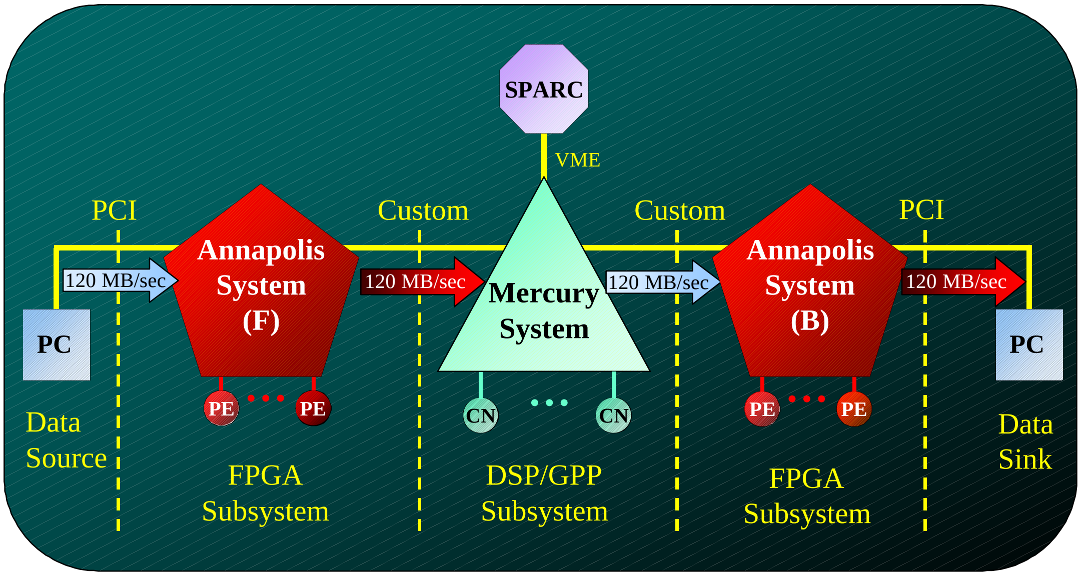

Figure 1. Overview of the architecture of the prototype system.

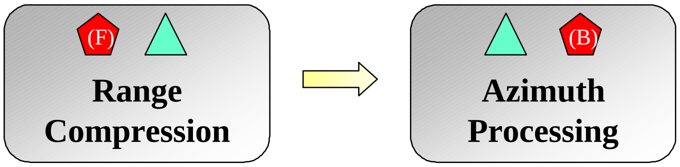

Figure 2. Major computational components of SAR processing flow.

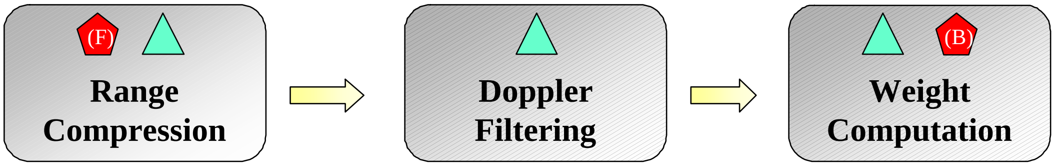

Figure 3. Major computational components of STAP processing flow.