

SRC Computers' MAP® Processors for Airborne Intelligence, Reconnaissance, and Surveillance Applications

SRC Computers, LLC
www.srccomputers.com

This white paper contains information which may be protected by issued and pending patents in the United States and elsewhere. No license under any SRC Computers, LLC intellectual property rights is granted by its publication.

OVERVIEW

SRC Computers has been shipping systems employing its reconfigurable MAP® processors since 2002, initially with its SRC-6 product line and now with the SRC-7 line. These systems have been in use solving problems for the intelligence community, Armed Forces, DoE laboratories and a variety of University programs ever since. In 2004 SRC Computers received an Air Force contract to shrink its desk-side PC-sized MAPstation™ system to allow its use on small surveillance Unmanned Aerial Vehicles (UAVs). The results of this program caught the attention of several prime contractors including Lockheed Martin who ultimately awarded SRC Computers the largest subcontract of the TRACER program.

This paper will present an overview of the SRC® IMPLICIT+EXPLICIT™ Architecture, airborne system implementation and the Carte™ high-level language programming environment as well as the systems' applicability to perform image and Synthetic Aperture Radar processing. Specific airborne program examples of these two major application categories will also be presented.

SYSTEM ARCHITECTURE

SRC Computers has developed a new hardware architecture and programming environment that delivers orders of magnitude more performance per processor than current high performance microprocessors. This new architecture is called the IMPLICIT+EXPLICIT Architecture. Systems built with this architecture execute the user's code, written in ANSI standard high-level languages such as C or Fortran, on a mixture of tightly coupled implicitly controlled microprocessors and explicitly controlled reconfigurable MAP processors. This allows the programmer to utilize both implicitly controlled functions, such as running a standard Linux operating system and executing legacy codes, as well as the explicitly controlled features such as the use of application specific data prefetch, data access, and functional units. This architecture is applicable to systems ranging in size from handheld devices to large multi-rack systems.

The fundamental IMPLICIT+EXPLICIT Architecture is shown in Figure 1. In this architecture, the Explicit and Implicit processors are peers with respect to their ability to access system memory contents. In this fashion, overhead associated with having both types of processors working together on the same program is minimized. This allows the programmer to utilize whichever processor type is best for a given portion of the overall application without concern for control handoff penalties.

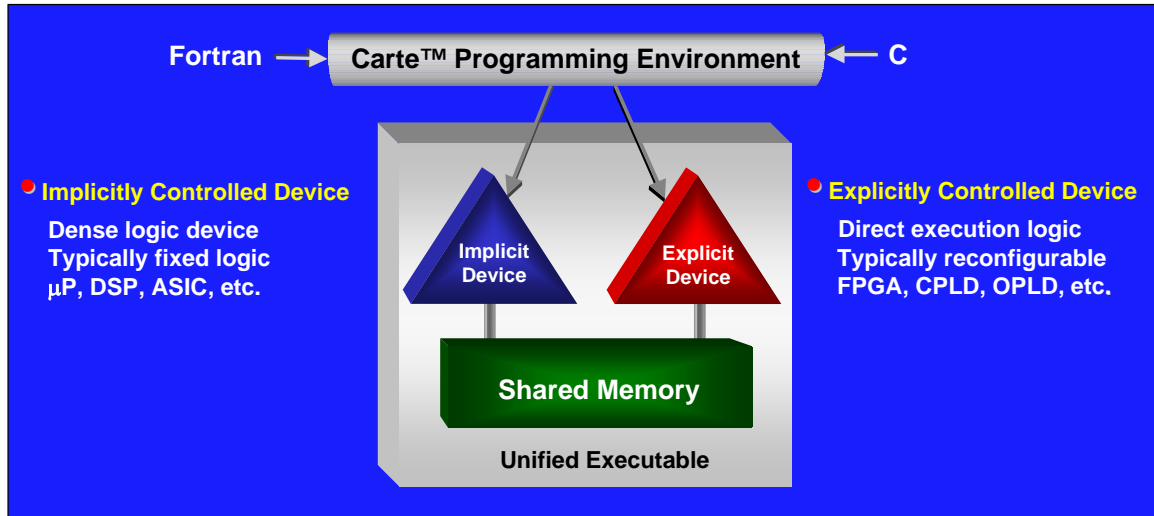


Figure 1 - SRC IMPLICIT+EXPLICIT Architecture

In this architecture, Implicit processors encompass a family of components that includes microprocessors, digital signal processors, as well as some ASICs. These processing elements are all implicitly controlled and are typically made up of fixed logic that is not altered by the user. These devices execute software-directed instructions on a step-by-step basis in fixed logic having predetermined interconnections and functionality. On the other hand, Explicit processors is a family of components that is explicitly controlled and is typically reconfigurable. This includes Field Programmable Gate Arrays (FPGAs), Field Programmable Object Arrays (FPOAs) and Complex Programmable Logic Devices (CPLDs). These devices allow the programmer to establish an optimized configuration of functional units to implement desired computational, prefetch and/or data access functionality for maximizing the parallelism inherent in the particular code. The SRC implementation of an Explicit processor is the MAP processor.

Both Implicit and Explicit processing elements are interconnected as peers to a shared system memory in one fashion or another. It is not required that interconnects support cache coherency since data sharing can be implemented in an explicit fashion.

Computation in the MAP processor uses dynamic logic, which conforms to the application rather than forcing the application into a fixed microprocessor architecture where one size must fit all. This delivers the most efficient circuitry for any particular code in terms of the precision of the functional units and the parallelism that can be found in the code. The result is a dynamic application-specific processor that can evolve along with a given code and can be reprogrammed in a fraction of a second to handle different portions of the code. The MAP processor provides the performance of a special-purpose computer and the economy of a general-purpose machine.

SRC Computers was the first to pioneer the breakthrough that was required to make the incorporation of Explicit devices ready for prime time. SRC Computers tightly coupled the hardware into a standards-based environment using a high-bandwidth and low-latency connection and then made this inherently superior hardware performance accessible to the broadest possible range of Fortran and C application/developers.

THE MAP PROCESSOR

The patented MAP processor uses commodity reconfigurable components to accomplish control, user-defined compute, data prefetch, and data access functions. This compute capability is teamed with very high on and off-module interconnect and memory bandwidths.

The current high performance version of the MAP processor is the Series H as shown in Figure 2. The MAP processor's sixteen logical banks of On-Board SRAM Memory provide 19.2 GBytes/sec of local memory bandwidth. In addition, the MAP processor contains two 1 Gbyte globally shared DDR-2 SDRAM banks. The MAP processor is equipped with two separate input and output ports with each port sustaining a data payload bandwidth of 3.6 GBytes/sec. This allows the MAP processor to simultaneously sustain two input and two output DMAs at an aggregate bandwidth of 14.4 GBytes/sec.

Each MAP processor also has general-purpose I/O expansion (GPIOX) ports sustaining an additional data payload of up to 12 GBytes/sec for direct MAP-to-MAP connections or sensor data I/O. This port is designed to accommodate GPIOX mezzanine cards that can perform such functions as A/D conversion, cameralink connection or Serial FPDP data interfacing.

The MAP processor is a 5"x7" module typically housed in a 5.25" drive bay enclosure along with its cooling solution and power converters. Each MAP processor is powered by +12v from a standard disk drive connector, which allows the easy incorporation of MAP processors into commodity PC enclosures. It also allows for the dense packaging of multiple MAP processors into 2U-high rack-mount chassis for use in SRC high performance servers as well as custom enclosures for rugged applications.

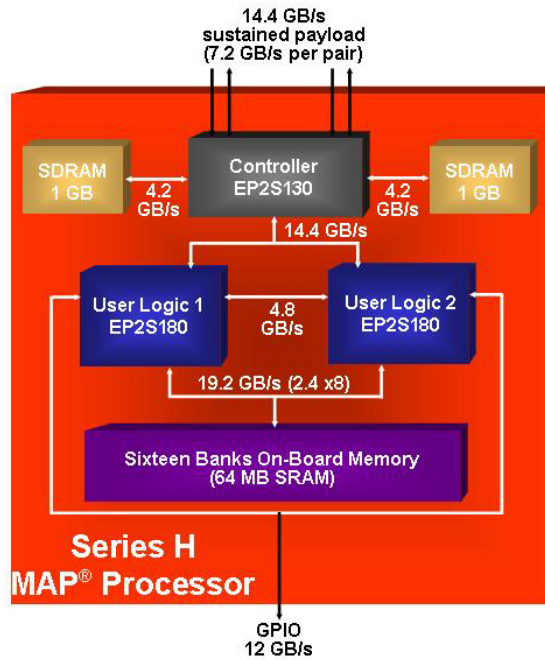


Figure 2 - Series H MAP Block Diagram

Systems can be built with a single MAP processor and microprocessor combination, or when more flexibility is desired, Multi-Ported Common Memory (MPCM) employing up to three MAP processors, and Hi-Bar® switches accommodating thousands of MAP processors can be used. The Hi-Bar switch is the SRC-proprietary scalable, high-bandwidth, low-latency switch. Each Hi-Bar module supports 64-bit addressing and has 16 input and 16 output ports to connect to 16 nodes. Microprocessors, MAP processors and Common Memory nodes can all be connected to Hi-Bar switch in any configuration as shown in Figure 3. Each input or output port sustains a yielded data payload of 3.6 GBytes/sec for an aggregate yielded data bandwidth of 57.6 GBytes/sec per 16 ports. Port-to-port latency is 180 ns with Single Error Correction and Double Error Detection (SECDED) implemented on each port. Hi-Bar switches can also be interconnected in multi-tier configurations, allowing two tiers to support 256 nodes. When used in SRC server products, each Hi-Bar switch is housed in a 1U-high, 19-inch wide rack-mountable chassis, along with its power supplies and cooling solution. In custom enclosures the Hi-Bar switch can be mounted in a variety of alternative configurations.

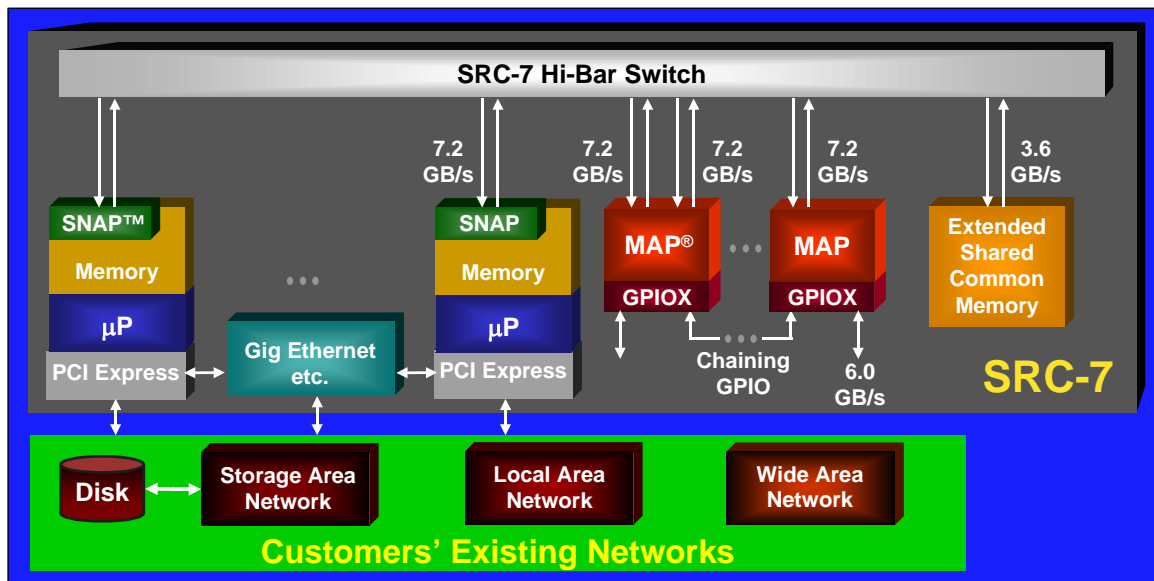


Figure 3 - Hi-Bar Switch Block Diagram

SRC servers that use the Hi-Bar crossbar switch interconnect can incorporate Common Memory nodes in addition to microprocessor and MAP nodes. Each of these Common Memory nodes contains an intelligent DMA controller and up to 32 GBs of DDR-2 SDRAM. This memory supports 64-bit addressing and can be accessed by all other MAP processor and microprocessor nodes in the system. Each node sustains memory reads and writes with 3.6 Gbytes/sec of yielded data payload bandwidth. The intelligent DMA controller is capable of performing complex DMA functions such as data transposition, strided access and scatter/gather to maximize the efficient use of the system interconnect bandwidth. Interconnect efficiencies more than 10 times greater than a cache-based microprocessor having the same bandwidth are common for these operations.

AIRBORNE SYSTEM IMPLEMENTATIONS

SRC Computers currently produces three system configurations for airborne applications. These systems utilize the same COTS modules that are used in all SRC products. UAVs using these SRC systems benefit from size, weight and power advantages as well as significant increases in computational performance.

The first of these systems is the Portable MAPstation™ system, which was developed under an Air Force contract in 2004. The latest Portable MAPstation is 2"x6 x10" convection-cooled system weighing just 4 pounds and powered by 12 VDC making it ideal for small UAV operations. When equipped with an internally mounted GPIOX card it can accommodate a variety of direct sensor inputs. For example the ADCX card supports two coherent 2 Gsample/s 10 bit A/D channels and has a low phase noise on board clock source. For image processing the Cameralink card can be used to connect to three base or one full Cameralink. For other data sources the quad port 10 Gb Ethernet card can be used. The system consumes 60 watts while providing more than an order of magnitude increase in performance than a 3 GHz Xeon® microprocessor for typical image processing and Synthetic Aperture Radar (SAR) applications. The system is software compatible with all other SRC systems, which allows applications to be developed on standard desktop systems and then deployed on the Portable MAPstation system.



Figure 4 - SRC-7 Portable MAPstation System

SRC Computers is currently updating the design of the Portable MAPstation system to utilize the latest MAP technology, a lower power microprocessor, and to have a wider environmental range. This SRC Portable MAPstation system has approximately five times the processing performance of its predecessor. It will be available both in a convection-cooled lightweight model as well as a sealed conduction cooled version. This version will have an operating range from -50C to +50C, an altitude limit in excess of 25K ft. and will meet shock and vibration requirements for single engine aircraft weighing less than 12.5K pounds. A block diagram of this MAPstation system, as shown in Figure 5, uses a Series J MAP processor in combination with an SRC-designed microprocessor module based on Intel's low-power Atom™ microprocessor called the APM processor. Typical large applications will consume approximately 80 watts. All existing GPIOX cards can be used with the SRC-7 Portable MAPstation systems.

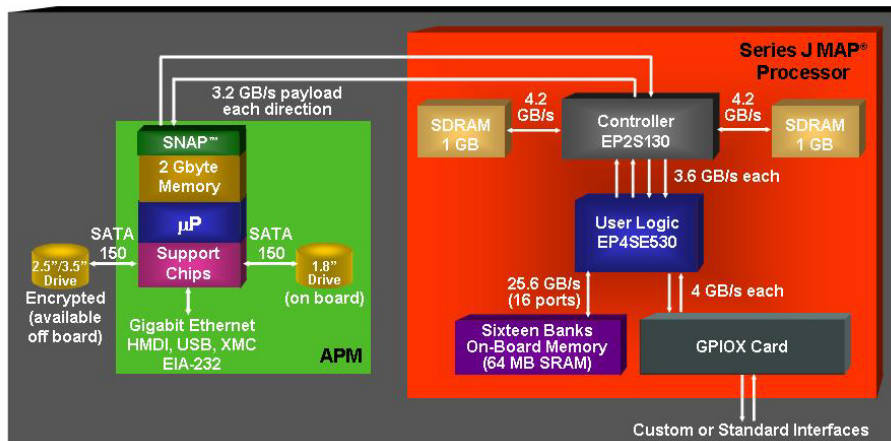


Figure 5 - SRC-7 Portable MAPstation Block Diagram

The second airborne system in production is a 10-module system, called the ATLAS™ system, designed for both manned and unmanned platforms. The ATLAS system can incorporate a customer-selected mix of low-power SRC APM processor modules, reconfigurable MAP processors and MPCMs, interconnected with the SRC high bandwidth Hi-Bar crossbar switch. All modules are interconnected with an aggregate sustained payload bandwidth of 57 GBytes/s. This 20"x17 x10.5" sealed system, as shown in Figure 6, has an overall operational weight with a typical 10-module complement of 75 pounds. The power consumption is only 500 watts for a complement of four MAP processors, two APM modules and 128 Gbytes of shared DDR-2 memory spread across four MPCM modules running a typical radar backprojection application. In addition, the ATLAS system is a powerful general-purpose processing system for ground station environments. As with all SRC products, this system uses an

unmodified Linux operating system. A version of this system is used by Lockheed Martin as the Signal Data Processor for the Army TRACER program and will be discussed in detail later in this paper.



Figure 6 - Ten Module Airborne ATLAS System

To compliment this airborne system, SRC Computers also produces a 19" rack-mountable air-cooled system called the High Density MAPstation system. This system uses all the same electronics as the ATLAS system and is ideal for application development or use in controlled environments.



Figure 7 - High Density MAPstation System

SRC Computers has developed a third new airborne solution, called the Mid-Sized Signal Data Processor (SDP) that is a variation of the other two. This mid-sized airborne system uses the SRC APM module combined with an MPCM module and two MAP processors as shown in Figure 8.

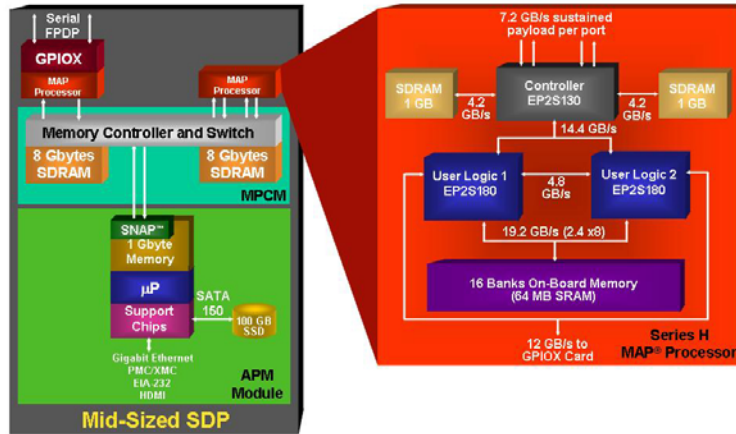


Figure 8 - Mid-Sized Signal Data Processing (SDP) Airborne System Block Diagram

This system is qualified for UAV applications to withstand an operating range from -54°C to $+60^{\circ}\text{C}$, an altitude limit in excess of 25K ft. and meets shock and vibration requirements for single engine aircraft weighing less than 12.5K pounds. The sealed system, Figure 9, has a completely passive thermal solution resulting in very high reliability.



Figure 9 - Mid-Sized Airborne System (Sealed Chassis)

THE CARTE PROGRAMMING ENVIRONMENT

To make the IMPLICIT+EXPLICIT Architecture easy for programmers to use, SRC Computers developed the Carte Programming Environment. Carte software tools support code development and execution on the hardware, as well as in emulation and simulation environments. These software tools take ANSI standard high-level language C or Fortran code and compile appropriate portions of it to run on the MAP processors and microprocessors. Everything needed to control both types of processors for a given application, is then combined by the Carte Programming Environment into a single Unified Executable. The end result is that for the first time ever, a programmer can easily use standard high-level programming languages and programming models to control a system with both reconfigurable and standard processors and achieve orders of magnitude more compute performance per processor than with microprocessors alone.

CODE DEVELOPMENT TOOLS

The SRC Carte Programming Environment supports co-development of programs written to run on microprocessors and SRC MAP processors. The MAP compiler compiles codes into configuration bit-streams that configure the MAP processor(s) while the Intel compiler compiles codes for the microprocessor.

Code to be compiled for the MAP processor is written as separate subprograms and collected into a single logic block by the Carte Programming Environment through the code in-lining process. The subprograms are called from routines executing on the microprocessor, as well as from other MAP routines. This technique permits hierarchical code development and code reuse. It also permits expressing greater parallelism within code blocks and permits optimizations across subprogram boundaries. The Carte Programming Environment then provides support for integrating the microprocessor routines, MAP routines, and libraries into a single Unified Executable that contains the microprocessor instructions, MAP logic, and all required library routines to manage the control of the program.

DEBUGGING TOOLS

The Carte Programming Environment fully supports co-development of software and hardware functional units through the use of debugger and hardware simulation tools. With the Carte Programming Environment managing both the microprocessor code as well as the simulation code, applications execute seamlessly in hardware or simulation environments. The Carte Programming Environment creates program executables that support source-level debugging through GNU's gdb debugger, the Intel[®] debugger, or any microprocessor-oriented code development tools. In debug mode, a program is executed entirely in the microprocessor. The routines coded for MAP processors are source line unchanged but retargeted for the microprocessor. This mode of development permits highly productive algorithm development and debugging. A MAP routine that is developed using debug mode will execute correctly when retargeted to the MAP processor. Debug mode can be used on any standard PC containing the Carte Programming Environment.

APPLICATIONS

SAR BACKPROJECTION

The Spotlight SAR Backprojection algorithm is considered to be the “gold standard” of the SAR imaging techniques. This section describes a study that compared the performance of a MATLAB implementation of a 2D SAR Backprojection application to a MATLAB – MAP routine implementation and to an all C Language implementation. The compute intensive MATLAB routines were converted into C Language and compiled using the Carte Programming Environment, which targets the SRC IMPLICIT+EXPLICIT Architecture with its MAP reconfigurable hardware.

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 is to order the pulses into a rectangular array and to perform 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. This backprojection algorithm calculates an exact solution for every pixel in the image, but 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, for 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 that is unavailable in any other imaging algorithm.

The backprojection computational technique is broken down into the following steps:

- Image or a slice of a 3-D object is broken down into a set of 1-D projections
- Each projection is filtered individually
- These projections are backprojected together
- Original image or cross-section is reconstructed

The backprojection algorithm is also used extensively in the medical imaging community. Applications utilizing computed tomography in the medical industry are Single Photon Emission Computerized Tomography (SPECT), Position Emission Tomography (PET), Computed Tomography (CT) Scan or Computed Axial Tomography (CAT) Scan, and Magnetic Resonance Imaging (MRI).

For more information (including details of the algorithm), please see "Spotlight Synthetic Aperture Radar" by Walter G. Carrara, et al.

The SAR backprojection application used in the study was developed under a Defense Advanced Research Projects Agency (DARPA) project called the Backhoe Challenge Problem, Public Release Number ACS 04-0273 and ASC 04-0990. The input data used in the study was synthetically generated using a simulated wideband (7-13 GHz) complex backscatter of data from a 3-D CAD model of a backhoe. The data contained 5040 1D projections of the object that were collected through 360 degrees. The computed 2D image was 1001 x 1001 pixels.

Conversion Motivation

The algorithm developers at the Air Force Research Lab (AFRL) use MATLAB for prototyping and long-term evaluation of their computational techniques. The compute time for generating a single 2D image using the entire input data volume takes approximately 1.3 – 1.5 hours based upon the microprocessor performance.

The AFRL group was very interested dramatically reducing this computation time to enhance the productivity of their algorithm developers.

Microprocessor Implementation

This implementation converted all of the original MATLAB code into the C Language. The imaging routine utilized the optimized FFTW library routines, and performance comparisons for different microprocessors are shown in Table 2 below.

SRC-7 Optimizations

The imaging routines implemented on the MAP processor 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 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 voxel in the 2D image. The architecture of the Series H MAP processor provides the ability to structure the algorithm in very novel ways. Table 1 reviews the MAP features used.

Table 1 - MAP Features

Architecture Feature	Algorithm Enhancements
Multiple DMA Paths	Concurrent DMAs of Input Data and Image Sum Array
Multiple User Logic Chips	Extension of the compute logic to handle the contribution of multiple input swaths into Image Sum Array at the same time
16 On-Board Memory Banks	Store intermediate compute results for larger number of input swaths.
Use of Global Common Memory Banks (GCM)	Using GCM allows for arbitrarily large Image Sum Arrays up to gigasample arrays
Streaming DMAs to Global Common Memory	Stream in the current Image Sum Array from GCM Bank 1, update with contribution of multiple input swaths and simultaneously stream out the Image Sum Array to GCM Bank 2.

The initial implementation on a SRC-7 Series H MAP processor used only one of the two User Logic Chips. This implementation uses a major performance enhancement by summing the contribution of four swaths at a time into every voxel in the 2D image. The two-chip version was then implemented that sums the contribution of twelve swaths into each voxel in the 2D image.

Table 2 – MAP Processor Performance

Implementation	Processor	Time (seconds)
C Code and FFTW	2.2 GHZ AMD Opteron™, 1024 Cache	153
	3.0 GHz Intel Xeon, 2048KB Cache	157
C – Series H MAP Processor	MAP using 1 User Logic Chip	10.43
	MAP using 2 User Logic Chips	3.77

Summary

Use of the backprojection algorithm in a microprocessor-based real-time SAR system was difficult due to onboard processing constraints (such as size, weight, and power). The orders of magnitude performance increase of running the SAR 2-D Backprojection algorithm on the SRC MAP architecture means that a real-time backprojection implementation is now practical, even onboard small UAV systems.

OPTICAL SENSOR IMAGE PROCESSING

Three optical image processing applications on the SRC-7 Series H MAP processor are presented in this section: image fusion (combining multispectral sensor data for image enhancement), feature tracking (tracking objects of interest across frames) and cross-correlation (identifying objects of interest). The details of each application are described and the measured performance of each implementation on a microprocessor and the MAP processor are presented.

The COTS CPU board used for performance measurement contains dual 3.0 GHz Intel Xeon CPUs (2048 KB cache) and 2 gigabytes of system memory. The SRC-7 Series H MAP processor contains dual 150 MHz Altera[®] 2S180 FPGAs with 64 megabytes of On-Board Memory (OBM) and 2 gigabytes of Global Common Memory (GCM). Input frame data rates are at the full cameralink bandwidth of 900 megabytes per second.

Six optical image sensor geometries (Table 3), from 1.9 megapixels to 112 megapixels, are used for each implementation's performance measurement.

Table 3 - Optical Image Sensor Geometries

x	y	Mpixels	Camera (Image Sensor)
1600	1200	1.9	Illunis XMV (Kodak KAI-2020)
4008	2672	11	Illunis XMV (Kodak KAI-11002)
4872	3248	16	Illunis XMV (Kodak KAI-16000)
7216	5412	39	Illunis XMV (Kodak KAF-39000)
8176	6132	50	Illunis XMV (Kodak KAF-50100)
10560	10560	112	(Dalsa / STA / USNO Astrometry)

Image Fusion

Multispectral data fusion applications combine spatial and spectral data obtained from different sensors into a single image in order to reveal information otherwise unavailable from a single sensor. For example, the fusion of two bands of infrared (IR) sensor data can distinguish large warm objects from small hot objects while a single band IR sensor cannot. Another example is the fusion of SAR sensor data with optical sensor data. SAR sensors provide spatial information about an object while optical sensors provide spectral information. The combination of spatial and spectral data enables object classification and identification.

Panchromatic sharpening is an image fusion technique that combines lower resolution red-green-blue (RGB) color image data with higher-resolution grayscale (GS) sensor data of the same target area in order to enhance image detail. This section describes a panchromatic sharpening implementation on the SRC-7 Series H MAP processor.

The basic concept of panchromatic sharpening is the higher-resolution GS pixel intensity values replace the RGB pixel intensity values for each frame. This implementation (Figure 10) assumes individual RGB and GS image frames are already co-registered and there is no need for frame resizing or cropping.

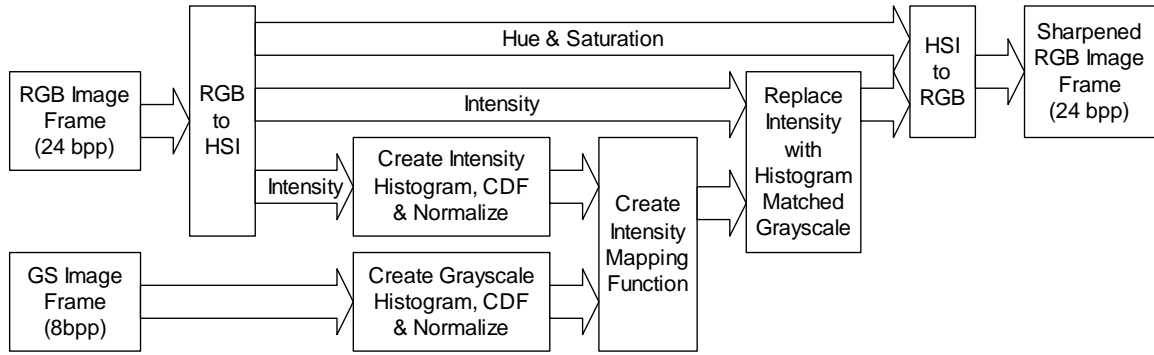


Figure 10 - Panchromatic Sharpening

A grayscale pixel's intensity is simply the pixel's 8 bit numeric value, but the intensity information is distributed among the individual RGB values for a color pixel. To obtain the intensity value for an RGB pixel, each 24 bit RGB value is transformed from the RGB color space to the Hue-Saturation-Intensity (HSI) color space. The intensity values for all pixels in both frames are then histogrammed. From these two intensity histograms, a statistical Cumulative Distribution Function (CDF) is created and then normalized for each frame. A mapping function is created from these two normalized CDF arrays to map the original color pixel intensity values to a new intensity value such that the new intensity value distribution matches the GS pixel intensity value distribution. The original intensity values are re-mapped and the new HSI image is transformed back into the RGB color space.

The CPU panchromatic sharpening application is a single threaded, serial implementation of the algorithm described above.

This Series H MAP implementation is divided into five separate stages (Figure 11). Each individual stage is fully pipelined by the Carte MAP compiler, but data dependencies force the processing into five stages. For example, stage 1 requires a complete frame histogram; stage 2 requires a complete CDF array, and so on.

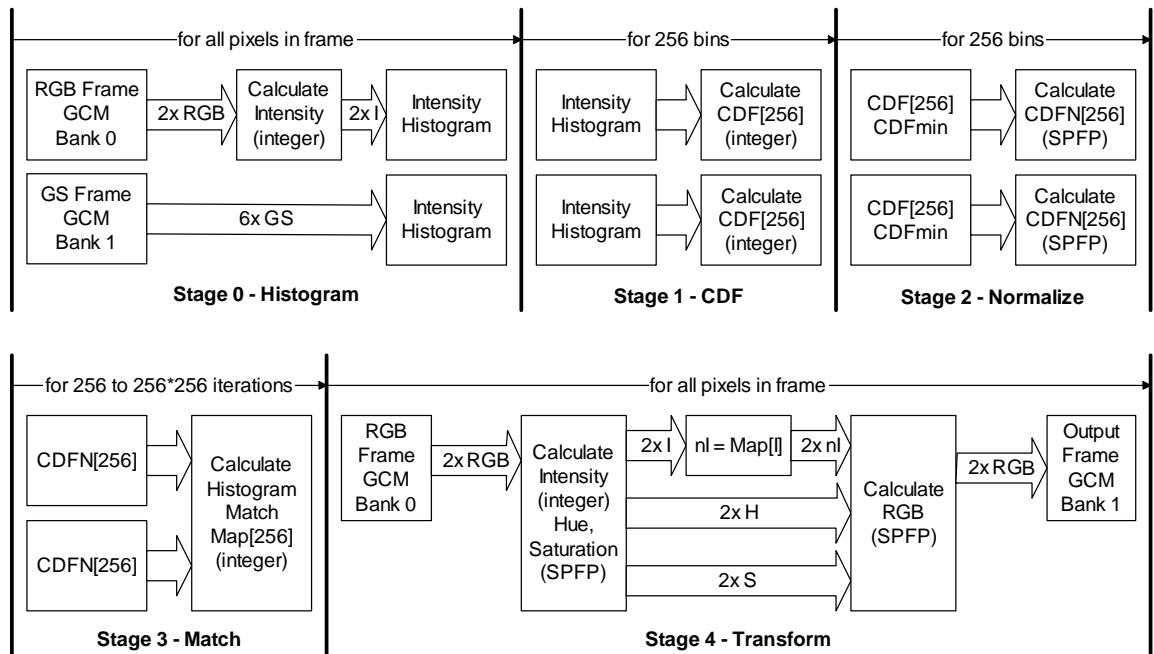


Figure 11 - MAP Multi-Stage Panchromatic Sharpening

The MAP processor's GCM Bank 0 acts as a frame buffer for the RGB image and GCM Bank 1 acts as a frame buffer for the GS image. In stage 0, two RGB and six GS pixel intensities are histogrammed in parallel every clock. The integer RGB intensity calculation is part of the RGB histogramming pipeline. After all pixel intensities for both frames are histogrammed, stage 1 calculates the CDF arrays for both histograms for all histogram bins in parallel. Stage 2 normalizes both CDF arrays in parallel, a single precision floating point (SPFP) calculation. Stage 3 uses both normalized CDF arrays to generate the histogram matching MAP array. Finally, stage 4 re-reads the RGB image data two RGB pixels per clock from GCM Bank 0 and calculates the HSI pixel values. The two integer intensity values select two new intensity values from the MAP array (generated in stage 3). The two new intensity values are cast to SPFP, and together with the two SPFP pixel hue and saturation values, are converted back to the 24 bpp RGB color space and stored in GCM Bank 1.

Table 4 shows the measured MAP processor performance in frames per second (fps) for each of the six image sensor sizes.

Table 4 - Measured Performance vs. Sensor Sizes

Sensor	Series H MAP Processor (fps)
1.9 Mpixel	75.21
11 Mpixel	12.95
16 Mpixel	9.21
39 Mpixel	3.54
50 Mpixel	2.54
112 Mpixel	1.02

Table 5 summarizes the SRC-7 architecture features that significantly enabled performance for this image processing implementation.

Table 5 - SRC-7 Architecture Features

Architecture Feature	Algorithm Enhancement
Multiple DMA Paths	Concurrent input of RGB and GS image data
Multiple DMA Paths	Concurrent image input and sharpened image output
Multiple User Logic Chips	Resources for two complete parallel computational pipelines
Dual 1 GB GDM Banks	Enables very large image frame buffers
Data Streaming	Enables pipelined parallel processing
High interconnect bandwidth	Easily enables full cameralink data rate

Feature Tracking

Visual object recognition, tracking and localization tasks in autonomous aerial vehicles require stabilized video imagery. This stabilization is achieved by image processing to estimate frame-to-frame motion due to aircraft movement. The estimated motion is removed from the video imagery so that it appears the optical sensor is not moving at all. This processing must be performed in real-time on-board the vehicle as round trip communication delays with a base station are long and there is limited video bandwidth. Feature tracking algorithms are commonly used to estimate frame-to-frame motion due to the aircraft. One popular algorithm is the pyramidal version of the Kanade-Lucas-Tomasi (KLT) feature tracker. This tracker works by minimizing the Euclidian distance between two frame's sub-images by an iterative gradient search. The algorithm is computationally efficient and well suited to real-time applications.

This section summarizes the implementation of a freely available KLT tracker (Stan Birchfield, "KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker,") on the SRC-7 Series H MAP processor

and compares its performance to the original CPU implementation. The computational kernel of the KLT tracker is an iterative horizontal and vertical convolution operation performed on two successive frames. On the MAP processor, both convolution operations are fully pipelined and executed in parallel at eight pixels per clock.

Table 6 shows the measured MAP processor performance tracking 100 features across two frames for each of the six image sensor sizes.

Table 6 - KLT Feature Tracking Performance

Sensor	Series H MAP Processor (s)
1.9 Mpixel	0.14
11 Mpixel	0.80
16 Mpixel	1.18
39 Mpixel	2.92
50 Mpixel	3.75
112 Mpixel	8.34

Table 7 summarizes the SRC-7 architecture features that significantly enabled performance for this image processing implementation.

Table 7 - SRC-7 Architecture Features

Architecture Feature	Algorithm Enhancement
Multiple DMA Paths	Concurrent input of both image frames
Multiple User Logic Chips	Resources for multiple concurrent convolution operations
Dual 1 GB GDM Banks	Enables very large image frame buffers
Data Streaming	Enables pipelined parallel processing
High interconnect bandwidth	Easily enables full cameralink data rate

Cross-Correlation

Image cross-correlation is used to locate and track a template image within a larger image. An image template is placed on a pixel in the target image and the sub-image under the template is compared to the template image for all pixels in the sub-image. Normalized cross-correlation is used to reduce brightness variance of the target image and template caused by lightning and exposure conditions. Normalization is performed every comparison step by subtracting the template and sub-image mean and dividing by the product of the standard deviations.

This section describes a spatial domain normalized cross-correlation implementation on the SRC-7 Series H MAP processor and compares its performance to a CPU implementation. Three template images (8x3, 8x6, and 8x9) are correlated to image sizes corresponding to each of the six sensor geometries. Since the template images are constant, their calculations are performed once at initialization time and stored on the CPU and MAP implementations.

Figure 12 shows a high level diagram of the normalized cross-correlation algorithm for this implementation. Except for precalculated values, all operations are performed for each image pixel and for each of the three templates. This implementation assumes the templates are already registered to the image frames. The frames contain eight bpp grayscale pixels, but these pixels are cast to floats on input and all calculations are performed with SPFP precision.

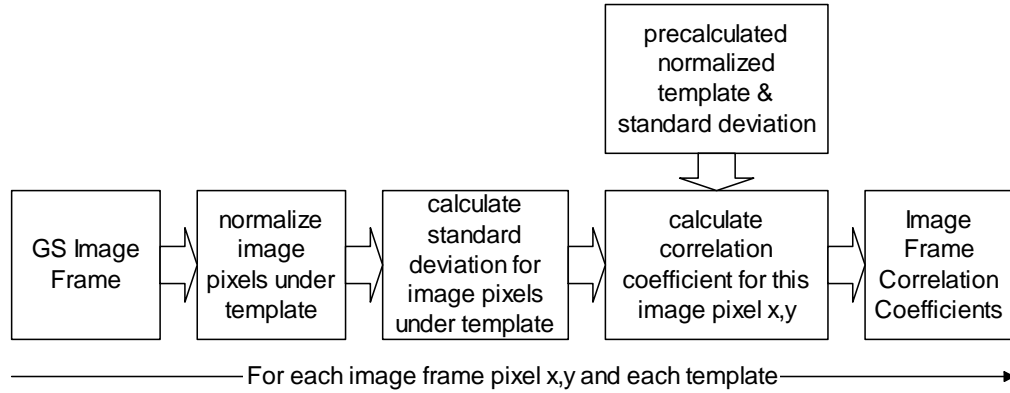


Figure 12 - Normalized Cross-Correlation Implementation

The CPU normalized cross-correlation application is a single threaded, serial implementation of the algorithm shown in Figure 12.

This Series H MAP implementation is divided into three parallel processing pipelines (Figure 13), one for each of the three templates. In Figure 13, "f(u,v)" represents a partially processed subimage area under a template of size uxv; "t(u,v)" represents a preprocessed template of size uxv; "cc(u,v)" represents a final cross-correlation value for a given template u,v. Cross-correlation values for all three templates are calculated for each image pixel.

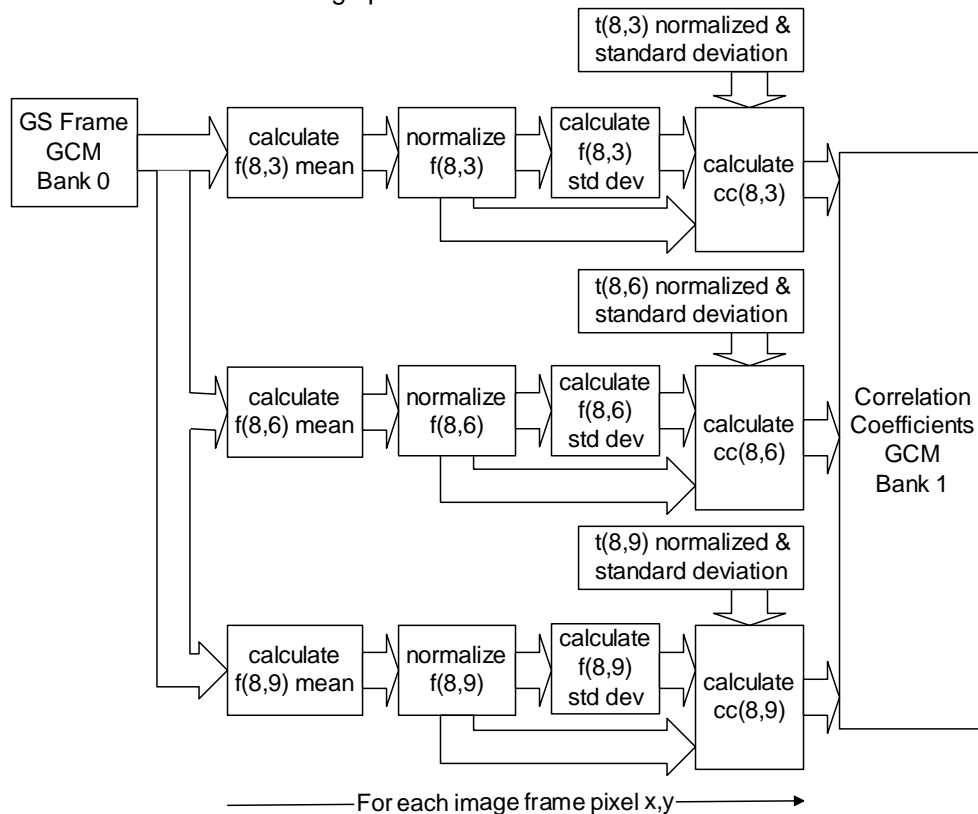


Figure 13 - Normalized Cross-Correlation Processing Pipelines

Table 8 shows the measured MAP processor performance in frames per second (fps) for each of the six image sensor sizes.

Table 8 - Normalized Cross-Correlation Performance

Sensor	Series H MAP Processor (fps)
1.9 Mpixel	78.64
11 Mpixel	14.05
16 Mpixel	9.50
39 Mpixel	3.85
50 Mpixel	3.00
112 Mpixel	1.35

Table 9 summarizes the SRC-7 architecture features that significantly enabled performance for this image processing implementation.

Table 9 - SRC-7 Architecture Features

Architecture Feature	Algorithm Enhancement
Multiple DMA Paths	Concurrent image input and coefficient output
Multiple User Logic Chips	Resources for multiple concurrent template processing
Dual 1 GB GDM Banks	Enables very large image frame buffers
Data Streaming	Enables pipelined parallel processing
High interconnect bandwidth	Easily enables full cameralink data rate

Summary

The three image processing implementations presented are each quite different and yet yielded sustained application performance from one to two orders of magnitude greater than a microprocessor system. The image fusion application is computationally intensive with floating point data and combines statistical, histogram matching and color space transform image processing building blocks. The feature tracking implementation displayed significant performance improvement for its intensive image convolution operations. Lastly, the cross-correlation implementation showcased simultaneous execution of multiple template based processing.

PROGRAM EXAMPLES

PITrackR

One example of the use of a Portable MAPstation system was the Air Force Precision Image Tracking and Registration (PITrackR) program. SRC Computers was selected to provide processors for the Precision PITrackR program by the AFRL Sensors Directorate. SRC COTS Portable MAPstation systems, with high-resolution cameras connected directly to the MAP processor's Cameralink GPIOX card, were used on this program. The resulting system provided real-time high pixel count imaging and metadata to track pilot/operator identified targets. The Portable MAPstation solutions for PITrackR weighed less than four pounds and measured only 120 cubic inches, while delivering more than enough direct sensor I/O bandwidth to support the Cameralink connection.

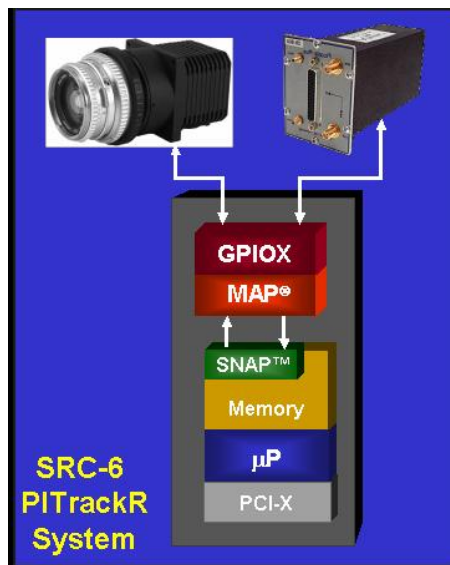


Figure 14 – SRC Portable MAPstation System for PITrackR

This system was programmed in ANSI C using the SRC Carte high-level language development environment. The ability to process the sensor information in-flight versus losing precious time while the data is relayed to a ground station is quite significant.

TRACER

The Lockheed Martin TRACER program for the U.S. Army is designed to fly on manned aircraft as well as on the General Atomics Sky Warrior UAV and utilizes the SRC ATLAS System as the SDP. Specifically the system is comprised of four MAP processors and five Global Common Memory (GCM) nodes. This configuration allows for multiple MAP processors to concurrently process data stored in the GCM nodes.

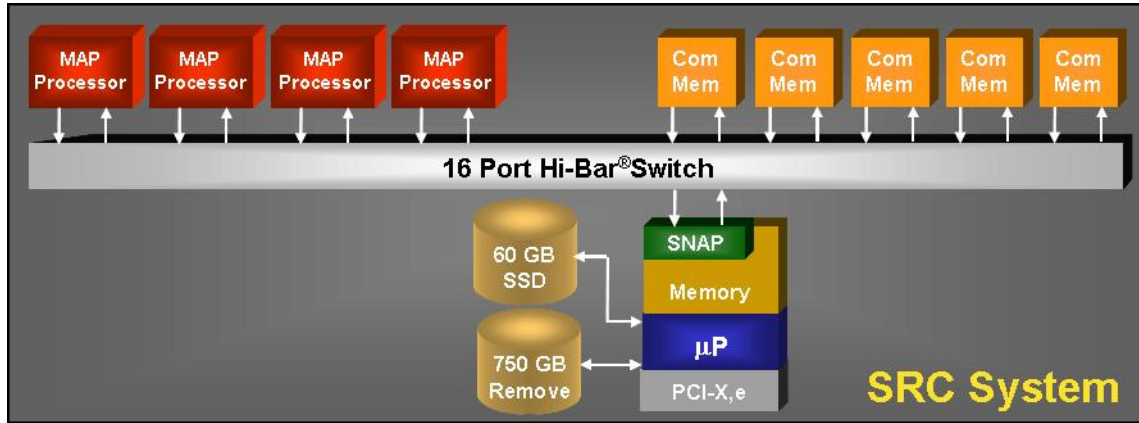


Figure 15 - TRACER Signal Data Processor

Processing Flow

The TRACER processing flow is made of several algorithms. The algorithms were written to partition the workload across several MAP processors and to use Global Common Memory nodes for storing the input and output data for each step in the processing. The processing flow is shown in Figure 16.

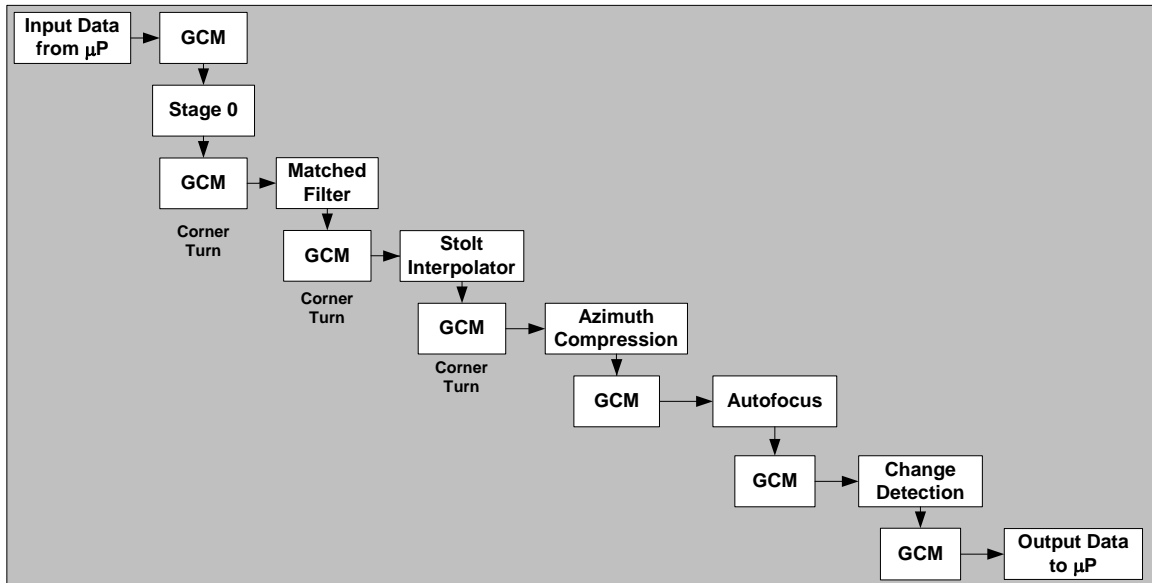


Figure 16 - TRACER Algorithm Sequence

Each processing flow stage was written to partition the workload across several MAP processors and to share Global Common Memory (GCM) nodes for storing the input and output data for each step in the processing. An example of this is shown for Stage 0 in Figure 17.

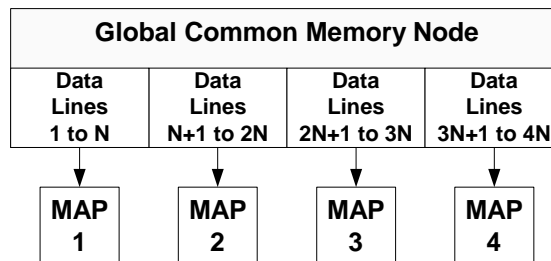


Figure 17 - Algorithm Data Movement

The data movement and compute are overlapped in time between the MAP processors. While the MAP processor 1 is processing a portion of its data, the MAP processor 2 is reading into on-board memories some of the data that it will process.

SRC Features Utilized by TRACER

The overall TRACER algorithm set comprised 250,000 lines of C codes. Three programmers ported the algorithm set to the SRC system in just three months. Performance optimizations were then made using the following architecture features.

Table 10 – SRC Architecture Features and Advantages

Architecture Feature	Advantage
Overlap DMA and compute in a MAP Processor	The time to DMA data from Global Common Memory is “hidden” behind the compute time
Stream processing	Stream processing creates an additional form of parallelism. Compute loops which are inherently sequential can be done in parallel
Streaming FFTs	The compute time for an FFT in clocks is the length of the FFT. For example, a 16384-point FFT can be computed in 16384 clocks
Complex DMAs	Complex DMAs provide the ability use specified address access patterns in the data written to Global Common Memory nodes. These forms of DMA provide the best supplied bandwidth given the data access patterns. The two forms of Complex DMA used by TRACER were the Corner Turn and 2D Sub-array

The result was an SDP with computational performance equal to 100 IBM Power PC microprocessors and a power consumption reduction from 5000 watts to just 600 watts. The system has completed all EMI, environmental, shock, and vibration qualification tests for fixed-wing single engine aircraft less than 12.5K pounds and completed flight-testing in 2010.

CONCLUSION

SRC Computers has developed and been shipping reconfigurable computer systems employing an innovative general-purpose architecture since 2002. Airborne versions of these systems have been qualified and are used on both small and large UAVs including both Arcturus and General Atomics platforms. For SAR and image processing applications these systems have demonstrated more than two orders of magnitude improved performance over microprocessor only solutions while at the same time having a reduction in power consumption and volume of approximately 90%. The ability of developers to use ANSI standard languages to port large-scale applications to the system very rapidly has also been demonstrated. SRC Computers is currently completing updated versions of its airborne systems utilizing the latest generation of MAP processors.