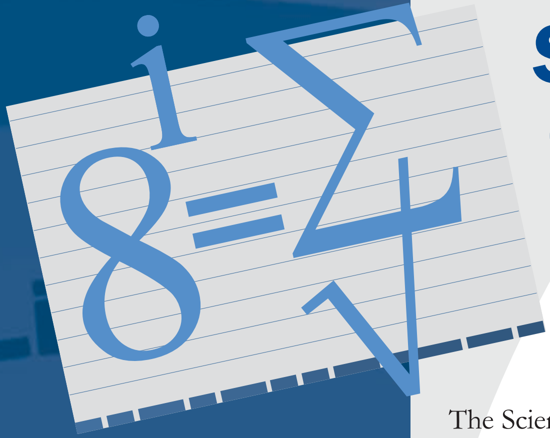
Scientific Algorithm Library (SAL)



Fastest Signal Processing Library

Highest Processor Utilization

High Throughput, Low Latency

Stable API Reduces Product Life-Cycle Costs

Embedded signal and image processing applications demand the greatest performance achievable from the processor. The Scientific Algorithm Library (SAL) from Mercury Computer Systems provides that performance through algorithm implementations that make the most efficient use of all processor resources.

The SAL is a floating-point mathematics library containing more than 600 routines required for signal processing applications such as medical imaging, radar, sonar, and signals intelligence. The most important routines are optimized for ultimate performance on AltiVec™-equipped PowerPC® microprocessors.

The Mercury SAL function categories include vector processing, matrix operations, Fast Fourier Transforms (FFTs), data conversion, signal processing, image analysis, and a wide variety of vector math operations, including vector reduction, vector-to-vector, vector-to-scalar, vector comparison, and multi-operator vector operations. The SAL also has arithmetic and logical vector functions for integer, real, double-precision, and complex data types.

High Performance

The first step toward creation of a high-throughput, low-latency signal processing library is to design the most efficient algorithms with the fewest possible instructions and computing resources. The next step is to hand-optimize or microcode routines for the target microprocessor to squeeze out the last drop of performance. The SAL represents the culmination of more than 10 years' expertise in algorithm design and microcode optimization by Mercury's staff of mathematicians, computer scientists, and applications experts.

Mercury's Scientific Algorithm Library is a powerful tool for application developers wishing to achieve maximum performance in multicomputer systems. The SAL has been optimized for each processor in a Mercury RACE++® Series system, including the AltiVec-enabled PowerPC microprocessor, and is able to take full advantage of the unique features of the AltiVec vector architecture. Most SAL functions run at least four or five times faster on a PowerPC 74xx compared to a PowerPC 750 running at the same frequency. A sample listing of function timings is provided in Table 1. Timings for other functions are available upon request.

Quality, Portability, and Productivity

SAL functions can be relied upon to provide accurate calculations. Each release of the SAL is rigorously tested to ensure accuracy, code quality, and performance. Correctness tests include producing correct answers, out-of-bounds reads and writes, and assembly code register and stack usage.

Table 1 Selected SAL Function Timings for a 500 MHz PowerPC 7410

Times in microseconds

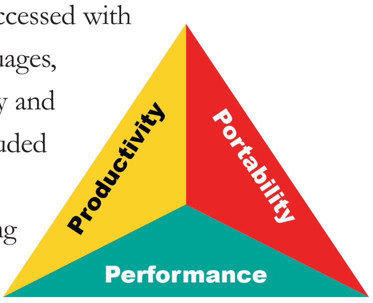
Function	Description	L1	L2	SDRAM
convx	Real convolution/correlation, 1024 output points/32 tap filter	20.98	23.34	30.38
zconvx	Split Complex convolution/correlation, 1024 output points/32 tap filter	74.47	78.52	101.17
dotprx	Real dot product, 1024 pts.	1.39	4.35	14.69
zdotprx	Split complex dot product, 1024 pts.	3.50	11.08	30.50
vmulx	Real vector multiply, 1024 pts.	1.68	7.22	25.46
zvmulx	Split complex vector multiply, 1024 pts.	3.59	15.81	39.58
fft_zriptx	Real 1D FFT, 1024 pts.	11.07	13.90	23.70
fft_ziptx	Split complex 1D FFT, 1024 pts.	19.36	24.52	40.83
fft2d_zriptx	Real 2D FFT, 32 x 32 pts.	24.01	26.96	33.86
fft2d_ziptx	Split complex 2D FFT, 32 x 32 pts.	22.30	27.50	38.73

Note:
The timings given in Table 1 are for a 500 MHz PowerPC 7410 with a 250 MHz L2 cache and 125 MHz SDRAM bus. All times are given in microseconds. SAL timings depend upon where the data in process resides and where the results are to be written. The given timings are for the cases where the data source and destination memory are identical; i.e., L1->L1, L2->L2, and SDRAM>SDRAM, but not for the mixed cases: SDRAM->L1, L2->L1, SDRAM->L2, etc. Mixed cases are usually the norm. This table shows best and worst case scenarios, with times for typical usage residing in between.

Performance tests check execution on data residing in L1 cache, L2 cache, and DRAM. The Mercury SAL is a mature product with a long history of field testing. Its error-free performance helps thousands of users in a broad range of applications.

The Mercury SAL has a history of providing a stable API that reduces product life-cycle costs and eases technology insertion. Over the years, the SAL has been ported to six processor families including i860, SHARC, and now the AltiVec-enabled PowerPC microprocessor. The insights and techniques developed through each of these iterations has carried forward to the next, resulting in a highly portable, high-quality signal processing library. The SAL application programming interface (API) is consistent between architectures and across processor generations, eliminating the need to recode for different target computers. As Mercury migrates to new processors in the future, the SAL will continue to provide high performance with a consistent API.

All SAL routines can be accessed with calls from higher-level languages, improving both productivity and portability. Binaries are included for all Mercury-supported development hosts, including Solaris™, Windows NT®, and Windows® 2000. The binaries allow users to develop applications on a supported workstation platform and then easily move them over to a Mercury multicomputer. Source files for the SAL coded in C are available for an additional fee. This allows users to develop with the SAL on any other workstation.



Because of the mathematical and computational expertise of the Mercury SAL design team, users are able to achieve greater performance levels than they could attain by developing these algorithms on their own. By leveraging the off-the-shelf performance of the 600+ functions in the SAL, application developers increase their productivity, saving development time and cost.

Getting the highest performance from the AltiVec's vector engine requires careful programming at the application level, as well as within the library functions. These details include data position, stride, and alignment. The SAL functionality, API, and usage guides help simplify these tasks to ensure optimal application performance.

SAL Functions

The SAL functions are divided across 18 categories as shown in Table 2.

Functions are available for one-dimensional and two-dimensional transforms and for mixed data types such as real-to-complex transforms and complex-to-real transforms.

Fourier transforms are critical to any signal processing system. They are useful in transforming data from the time domain to the frequency domain and vice versa. The SAL provides a wide selection of high-performance, optimized FFTs such as Real and Complex 1D, multiple 1D and 2D, and mixed radix 1D FFTs.

The SAL supports nine data types as shown in Table 3. The first five are industry-standard data types, and the remaining four are Mercury-defined.

Table 2 SAL Function Categories

1-D FFTs and Associated Window	Perform 1-D FFTs and compute a variety of associated windows
2-D FFTs and Associated Window	Perform 2-D FFTs and compute a variety of associated windows
1-D Correlation, Convolution, and Filtering	Perform 1-D correlation, convolution, and filtering operations
2-D Correlation, Convolution, and Filtering	Perform 2-D correlation, convolution, and filtering operations
2-D Image Processing	Perform a variety of specialized image processing and filtering functions such as back projection of a vector onto a 2-D image
Matrix Arithmetic	Perform arithmetic operations on matrices such as matrix multiplication, matrix transposition, and matrix inversion
Data-Type Conversion	Convert a vector of one data type to a corresponding vector of another data type (examples: integer-to-float, float-to-double)
Single Vector Generation	Fill vectors with specified values, such as all zeros or a ramp function
Single Vector Scalar Arithmetic	Combine a scalar with a vector (examples: multiply/divide a vector by a scalar, add/subtract a scalar to each element of a vector)
Single Vector Scientific	Transform each element of a vector in accordance with a specified math operation such as sine, square root, or logarithm
Single Vector Scalar Comparison	Compare each element of a given vector against a specified scalar value for thresholding, clipping, and similar operations
Single Vector Move	Move a vector from one set of locations to another
Single Vector Miscellaneous	Single vector operations that do not fit any of the preceding Single Vector function categories.
Vector-to-Scalar	Produce a scalar result from either a single vector, or a combination of two equal-length vectors (examples: vector dot-products, the sum of vector elements, and the RMS value of vector elements)
Vector-Vector Arithmetic	Arithmetically combine two vectors of equal length to produce a single-vector result (examples: vector addition/subtraction, element-by-element vector multiplication/division)
Vector-Vector Merge	Merge two equal-length vectors into a third vector
Vector-Vector Comparison	Compare the corresponding elements of two vectors and generate a third vector based upon the results of this comparison (example: create a vector containing the minimum of the corresponding elements in the two input vectors)
Vector-Vector Logical	Perform element-element, bit-wise logical operations such as AND, OR, XNOR between the two vectors, outputting the result in a third vector

Table 3 SAL Supported Data Types

Real - float	32-bit IEEE single-precision floating point
Double	64-bit IEEE double-precision floating point
Integer - Int	32-bit integer
unsigned short	16-bit integer
Character - char	8-bit integer
COMPLEX	Ordered pair of real numbers
COMPLEX_SPLIT	A pair of pointers to the real and imaginary parts
DOUBLE_COMPLEX	Ordered pair of double-precision real numbers
DOUBLE_COMPLEX_SPLIT	Double-precision real and imaginary pointers

SAL System Requirements

Compute Elements (CEs)
- **PowerPC processor family**

MC/OS™ multicomputer operating environment
- **MC/OS 5.0 or greater**

Development Hosts
- **Solaris, Windows NT, or Windows 2000**

Runtime Hosts
- **Solaris, Windows NT, Windows 2000, or VxWorks®**

Packaging

The SAL is included as part of the standard PK1 software distribution package.

Computer Systems, Inc.
MERCURY
The Ultimate Performance Machine

For more information, go to www.mc.com

199 Riverneck Road
Chelmsford, MA 01824-2820 U.S.A.
978-256-1300 · Fax 978-256-3599
800-229-2006 · http://www.mc.com
NASDAQ: MRCY

Nihon Mercury Computer Systems K.K.
No. 2 Gotanda Fujikoshi Bldg. 4F
5-23-1 Higashi Gotanda
Shinagawa-ku, Tokyo 141-0022
JAPAN
+81 3 3473 0140
Fax +81 3 3473 0141

Mercury Computer Systems, SARL
Immeuble le Montreal
19 bis, avenue du Quebec
Villebon
91951 COURTABOEUF CEDEX
FRANCE
+ 33 (0) 1 69 59 94 00
Fax + 33 (0) 69 86 10 875

Mercury Computer Systems, Ltd.
Campbell Court, Unit 19
Bramley, Tadley
HANTS RG26 5EG
UNITED KINGDOM
+ 44 1 256 880090
Fax + 44 1 25688 4004