DATASHEET

MERCURY COMPUTER SYSTEMS

MultiCore MathPack

Ultimate Performance Libraries for Signal, Image, and Data Processing

- Optimizes multicore processor resources for demanding processing-intensive applications
- Automatically utilizes additional cores for maximum performance
- Enhanced processor utilization enables high throughput and low latency
- Proven, mature, stable APIs reduce product life-cycle costs and time-to-market
- Seamless upgrade for existing single-core SAL applications

MultiCore MathPack from Mercury Computer Systems is a bundled software package that includes our latest MultiCore SAL and MultiCore VSIPL multicore libraries with our industry-leading CSAL, SAL, and VSIPL single-core libraries.

The MultiCore Scientific Algorithm Library (MultiCore SAL) lets SAL-based applications effortlessly achieve higher performance from the most advanced multicore processors in our product lines. Source-code compatibility with existing applications helps decrease time-to-market by minimizing the changes to existing applications as embedded systems adopt multicore processors.

Signal, image, and data processing applications demand the greatest performance achievable from the processor. The Scientific Algorithm Library (SAL) provides that performance through use of all processor resources. MultiCore SAL extends performance for multicore processors by automatically utilizing additional cores, and it provides similar performance to SAL per core.

MultiCore MathPack Contents and Packaging

MultiCore MathPack includes:

- Multicore libraries:

  MultiCore SAL

  MultiCore SAL with select optimizations for multicore (threaded)

  MultiCore VSIPL

  MultiCore VSIPL-Lite Plus over MultiCore SAL

- Single-core libraries:

  CSAL "C" Source code for all SAL APIs

  SAL SAL for Linux® with select optimizations for single-core

  VSIPL VSIPL-Lite Plus over SAL (where optimized) or CSAL

All these libraries are packaged into one MultiCore MathPack bundled package to simplify high-performance library ordering and to avoid determining any dependency requirements (for example, MultiCore VSIPL requires MultiCore SAL, which requires SAL). This common asset package applies across all our product offering to simplify ordering and familiarization.

MultiCore MathPack is bundled in the MultiCore Plus® Pro Edition software at a discount, and is offered a la carte with the Standard Edition software bundle.

Overview of Benefits

Mercury provides an elegant solution to the multicore problem. The days of free performance improvements by increasing clock speeds are past. Scaling performance with multiple numbers of cores requires serious effort. MultiCore SAL and MultiCore VSIPL take over this effort by managing the parallelism. There is a single thread of processing from the application perspective, but many of the SAL functions take about 1/(number of cores) the time to do the work.

MultiCore SAL eases the migration effort of single-core (serial) SAL applications to multicore (parallel) processing environments, without having to change every SAL call in your program. Because MultiCore SAL includes macros to specify whether to run a single-core or multicore version of a function, your applications can contain both single and multicore SAL calls using C or C++ for 32- or 64-bit applications.

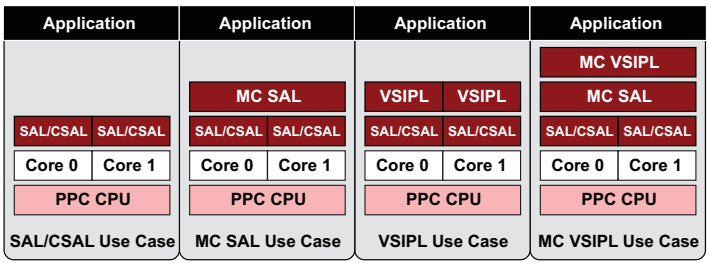


Figure 1. *MultiCore MathPack Use Cases*

MultiCore SAL
(MultiCore Scientific Algorithm Library)

MultiCore SAL contains more than 400 functions required for demanding processing-intensive applications, such as radar imaging, image exploitation, signals intelligence, image reconstruction, baggage scanning, manufacturing inspection, and others. These functions are optimized for ultimate performance on AltiVec™-equipped processors and the Freescale™ MPC864xD processors. Approximately 385 of MultiCore SAL functions are optimized for threaded (parallel) implementation on multicore processor systems.

Like SAL, MultiCore SAL has performance and development libraries. However, the MultiCore SAL development library provides TATL™ trace events for each function to automatically help determine how efficiently each of the processor cores is being utilized.

MultiCore SAL is used from the manager or application process thread only, which creates and manages one or more worker threads and optionally performs polling or interrupt-based synchronization. The worker and manager threads share the work defined by a SAL function invocation.

The typical improvement for a 2-core chip is close to 2X. Some functions scale in performance better than the number of cores, because there is more cache memory available compared to a single core.

High Performance with MultiCore SAL

Mercury designed MultiCore SAL to leverage existing industry-leading SAL optimizations to create a high-throughput, low-latency signal-processing library designed for coherent multicore processors. MultiCore SAL employs methods for managing tasks on all the available cores to minimize overhead and maximize processor utilization.

Table 1 characterizes comparative performance for selected functions on an MPC8641D processor, showing the ration of the same function running on two cores versus one core. All the functions scale nearly perfectly (maximum perfect performance would equal 2) from one core to two cores. Performance numbers of other functions are available upon request.

MultiCore SAL Portability and Productivity

SAL provides a stable application programming interface (API) that reduces product life-cycle costs and eases technology insertion. Over the years, SAL has been ported to six processor families, including i860™, SHARC®, Cell Broadband Engine™ (BE), and AltiVec-enabled Power Architecture™ microprocessors. The insights and techniques developed through each of these iterations contribute to a highly portable, high-quality signal, image, and data processing library. The SAL API is consistent across architectures and processor generations, eliminating the need to recode for different target computers. As Mercury migrates to new processors in the future, SAL will continue to provide the same high performance with a consistent API.

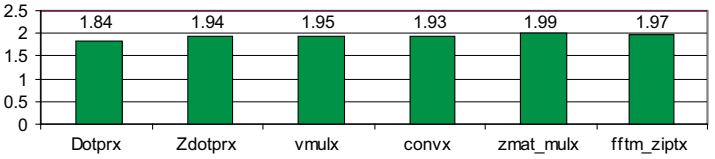
MultiCore SAL builds upon this philosophy to deliver the power of multicore processors with the consistent API that Mercury customers have depended on for years. MultiCore SAL has been architected to accommodate processors as increasing numbers of cores are developed.

Table 1. *MultiCore SAL Performance for Freescale MPC8641D*

Function	Description	1-Core* vs. 2-Core Speed-up with MC SAL
Dotprx	Real dot-product N = 32768	1.84
Zdotprx	Complex dot-product N = 32768	1.94
vmulx	Real vector multiply N = 32768	1.95
convx	Real convolution N = 16384, Filter = 16	1.93
zmat_mulx	Complex matrix-matrix multiply 256 x 256	1.99
fftm_ziptx	Complex multiple 1D FFT LOG2N = 12, m =16	1.97

*1-core timings were performed with second core idle.

MultiCoreC SAL performance: comparison of 2 core over 1 core performance for identical MultiCore SAL function



All MultiCore SAL routines can be accessed with calls from higher-level languages, improving both productivity and portability. Source files for SAL, coded in C, are available, allowing users to develop applications with SAL on their desktop and then rebuild with appropriate target libraries for deployment.

MultiCore SAL Functions

MultiCore SAL functions support a wide variety of mathematical operations critical for many applications. MultiCore SAL supports real and complex datatypes, including single- and double-precision floating-point, and 8-, 16-, and 32-bit integer.

Function categories include:

- 1-D FFTs and associated window
- 2-D FFTs and associated window
- 1-D correlation, convolution, and filtering
- 2-D correlation, convolution, and filtering
- 2-D image processing
- Matrix arithmetic
- Datatype conversion
- Single vector generation
- Single vector scalar arithmetic:
  - Single vector scientific
  - Single vector scalar comparison
  - Single vector move
  - Single vector miscellaneous
  - Vector-vector arithmetic
  - Vector-vector merge
  - Vector-vector comparison
  - Vector-vector logical

MultiCore VSIPL (MultiCore Vector, Signal, and Image Processing Library

MultiCore VSIPL contains approximately 800 entry points, including the original VSIPL-Lite standard-defined 125 functions, plus a select number of functions from the VSIPL CORE and VSIPL FULL standard listings. MultiCore VSIPL requires MultiCore SAL for the multicore advantage and maintains less than 0.5% performance impact for function call overhead.

Application programmers simply re-link their application, and MultiCore SAL and SAL automatically enable high performance.

SAL (Scientific Algorithm Library)

SAL function categories include vector processing, matrix operations, fast Fourier transforms (FFTs), data conversion, signal processing, image analysis, linear algebra; and a wide variety of vector math operations, including vector reduction, vector-to-vector, vector-to-scalar, vector comparison, and multi-operator vector operations. SAL also has arithmetic and logical vector functions for integer, real, double-precision, and complex datatypes. Recent additions include singular value decomposition (SVD), eigenvalues of Hermitian matrices, resampling, and Cholesky and QR decomposition.

High Performance with SAL

Mercury has created a high-throughput, low-latency signal processing library containing efficient algorithms with the fewest possible instructions and computing resources. Most of the library has been hand-optimized in microcode to maximize performance for the target microprocessors. SAL represents the culmination of over 10 years of expertise in algorithm design and microcode optimization by Mercury's staff of mathematicians, computer scientists, and applications experts. Table 2 is a sample listing of SAL function timings. Timings for other functions are available upon request.

SAL Portability and Productivity

All SAL routines can be accessed with calls from higher-level languages, improving both productivity and portability. Source files for SAL, coded in C, are included. This allows users to develop applications with SAL on their desktop and then to rebuild with appropriate target libraries for deployment.

SAL allows application developers to leverage the expertise of the Mercury design team to achieve better performance than they could attain by developing these algorithms on their own. By leveraging the off-the-shelf performance of the 600+ functions in SAL, application developers increase their productivity, saving development time and cost.

Achieving 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 data alignment. The SAL Reference Manual helps simplify these tasks to ensure optimal application performance.

VSIPL (Vector, Signal, and Image Processing Library)

VSIPL contains approximately 800 entry points, including the original VSIPL-Lite standard-defined 125 functions, plus a select number of functions from the VSIPL CORE and VSIPL FULL standard listings. VSIPL requires SAL for operation, because it maps to SAL, providing less than 0.5% performance impact for function call overhead.

CSAL ("C" Scientific Algorithm Library)

CSAL contains all source code in "C" for all SAL functions. CSAL allows developers to create their own functions derived from the source code for platforms that are not part of Mercury's defined embedded target platforms (such as Windows®). The intention is that resultant code be used either in conjunction with, or on Mercury-provided solutions. Binaries are included for the MPC864xD target platform, for example.

Table 2. *Selected SAL Function Timings for a 1.3 GHz MPC8641D Processor*

Function	Description	Time
convx	Real convolution/correlation, 1024 output points/32 tap filter	6.20
zconvx	Split complex convolution/correlation, 1024 output points/32 tap filter	28.14
dotprx	Real dot product, 1024 pts	0.47
zdotprx	Split complex dot product, 1024 pts	0.98
vmulx	Real vector multiply, 1024 pts	0.81
zvmulx	Split complex vector multiply, 1024 pts	1.51
fft_zriptx	Real 1D FFT, 1024 pts	3.33
fft_ziptx	Split complex 1D FFT, 1024 pts	5.86
fft2d_zriptx	Real 2D FFT, 32 x 32 pts	7.88
fft2d_ziptx	Split complex 2D FFT, 32 x 32 pts	6.873

Note: The timings given in Table 2 are for a 1.3 GHz MULTICOREP8641D processor. All times are given in microseconds.

Comprehensive Services

MultiCore MathPack is a bundled set of algorithm primitive routines that are most commonly used by developers in Mercury's served markets. Our algorithm optimization, consulting, and training services are available to help you achieve ultimate performance for your most challenging applications through the optimization of custom algorithms.

Specifications

MultiCore SAL System Requirements

Processors
MPC864xD
Operating system
Linux®, VxWorks®

SAL System Requirements

Processors
MPC74xx, MPC864xD, IBM PowerPC 970, Cell BE
Operating systems
Linux, VxWorks, Windows, MCOE™

1945.01E-DS-0409-mcmathpack

MERCURY COMPUTER SYSTEMS

Corporate Headquarters
201 Riverneck Road
Chelmsford, MA 01824-2820 USA
+1 (978) 967-1401 • +1 (866) 627-6951
Fax +1 (978) 256-3599
www.mc.com

Worldwide Locations
Mercury Computer Systems has R&D, support and sales locations in France, Germany, Japan, the United Kingdom and the United States.

For office locations and contact information, please call the corporate headquarters or visit our Web site at **www.mc.com**.

Challenges Drive Innovation™