DATASHEET

MERCURY COMPUTER SYSTEMS™

MultiCore Plus MathPack

Exploit optimized and portable Scientific Libraries for Intel, PowerPC and NVIDIA parallel processors

- High-performance libraries for signal, image and data processing
- Field-proven APIs reduce product life-cycle costs and time-to-market
- Supports heterogeneous computation resources, such as CPUs and GPUs, in a single application
- Utilizes multiple processor cores for processing-intensive applications
- Enhanced processor utilization enables high throughput and low latency
- Seamless upgrade for existing single-core SAL applications
- Open source version allows for rapid development and independent customization

Signal, image and data processing applications demand the greatest performance achievable from the processor. Mercury's Scientific Algorithm Library (SAL) is a collection of over 800 functions that provide high-performance through use of all processor resources, including co-processors such as GPUs.

MultiCore SAL (MC SAL) and MultiCore Vector Signal Image Processing Library (MC VSIPL) extends performance for multicore processors by automatically utilizing additional cores. This lets SAL-based applications effortlessly achieve higher performance from the most advanced multicore processors in Mercury's 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.

MultiCore Plus® MathPack from Mercury Computer Systems is a bundled software package that includes the latest MC SAL and MC VSIPL libraries with the industry-leading SAL and VSIPL single-core libraries as well as the "C" language implementation of SAL (CSAL).

SAL (Scientific Algorithm Library)

Mercury has created a high-throughput, low-latency signal processing library containing efficient algorithms with the fewest possible instructions and computing resources. Many key functions of the library have been hand-optimized to maximize performance for the target microprocessors. SAL represents the culmination of over 20 years of expertise in algorithm design and code optimization by Mercury's staff of mathematicians, computer scientists and applications experts. Timings for functions on different architectures are available upon request.

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. Example SAL functions include singular value decomposition (SVD), eigenvalues of Hermitian matrices, resampling, and Cholesky and QR decomposition.

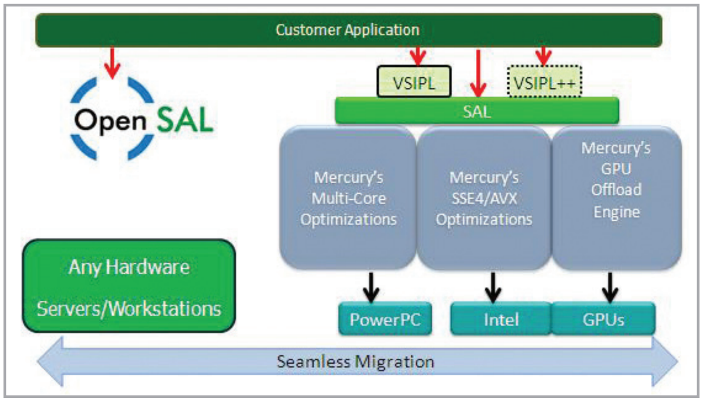

Figure 1. Mercury's SAL provides high performance, portability and software investment protection.

VSIPL (Vector Signal Image Processing Library)

VSIPL is an open-standard image and signal processing library. As part of the VSIPL Forum, Mercury helped to create and steer the direction of the VSIPL standard. Mercury's VSIPL library contains approximately 800 entry points, including the original VSIPL-Lite standard 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, it has less than 0.5% performance impact for function call overhead. It is an ideal solution for programs with VSIPL dependency.

MC SAL and MC VSIPL

Mercury provides an elegant solution to utilizing multiple cores on PowerPC and Intel processors. Typically, scaling performance with multiple cores is difficult. Using standard programming languages requires training and expertise in multi-threaded and multi-processor applications. MC SAL and MC VSIPL provide this expertise by internally managing the parallelism. There is a single thread of processing from the application perspective for ease of use; however most of the MC SAL functions achieve extremely high parallel efficiency.

MC SAL and MC VSIPL manage the cores and synchronization internally and achieve a high percentage of theoretical performance. This allows scientists and engineers to focus on the algorithm at hand, not the complexities of optimization for multiple cores. Inner-loop code written to use the single core SAL and VSIPL libraries remains completely unchanged. By calling the correct initialization function, other existing SAL and VSIPL code only needs to be recompiled to leverage the multiple cores in modern CPUs.

MC SAL Functions

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

Function categories include:

- 1-D FFTs and associated windows
- 2-D FFTs and associated windows
- 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

Figure 2 provides a sample of performance improvements achieved by MC SAL over an equivalent function running in a single core environment. As an example, MC SAL can run the vmulx function 1.95 times faster than the single core equivalent, in a 2-core processor.

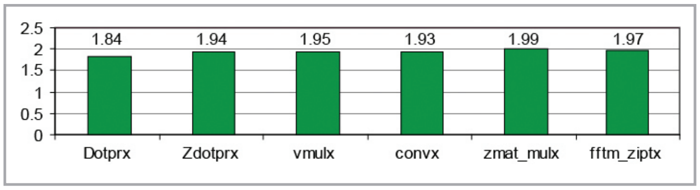

Figure 2. MC SAL performance comparison of 2 cores over 1 core for identical MC SAL function on a PowerPC 8641D.

CSAL

Included in MathPack is CSAL, a reference version of all the SAL and MC SAL APIs. CSAL contains all source code in "C" for all SAL functions. CSAL allows developers to prototype their own applications on platforms that are not part of Mercury's defined embedded target platforms, such as Windows®. CSAL provides scientists and engineers with insight into intended behavior and usage of the SAL library. Mercury also offers OpenSAL, an open source version of CSAL to allow the development community to evaluate SAL and develop applications using the rich SAL API. To learn more, check out the OpenSAL website at www.opensal.net

MathPack 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 eight processor families, including i860™, SHARC®, Cell Broadband Engine™ (BE), and AltiVec™-enabled PowerPC™ 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.

The newest architectures supported by MathPack are Intel® processors and NVIDIA® graphics processing units (GPUs).

PowerPC

Mercury's SAL library is the "gold standard" for PowerPC (aka: Power ISA, Power Architecture) math libraries. Mercury has spent years optimizing and improving algorithm performance on the PowerPC processor line, starting with the PPC603 and extending to today's dual-core PPC8641D. SAL has the fastest FFT implementations available on PowerPC processors. Faster processing directly leads to a reduced system size 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.

Intel

MathPack now has optimized performance on Intel processors. Broad SAL and VSIPL coverage has been achieved leveraging Intel's IPP and MKL libraries. In areas where IPP and MKL do not meet the performance expectations of SAL customers, Mercury has optimized routines using Intel's AVX and SSE4 streaming instructions found on modern Intel processors.

GPU

Historically, graphics processing units (GPUs) have been viewed as compelling, programmable floating-point graphics-rendering engines. The success of the GPU can be attributed to its massively parallel architecture, with up to several hundred processing cores available to take advantage of the parallelism existing in many graphics algorithms. GPUs have excelled for this purpose and are now employed in most personal computers, workstations and gaming consoles. However, the availability of embedded GPU solutions suitable to the stringent requirements of high-performance signal processing has been scarce.

With recent architectural advancements, the algorithmic scope of GPUs has grown dramatically. Non-graphics applications — such as intelligence, surveillance, reconnaissance, signals intelligence and electronic warfare — can now be addressed by GPU technology with excellent results. GPUs excel at traditional signal processing algorithms that provide more parallelism than can be exploited on a single or multi-core CPU, like the fast Fourier transform (FFT). Industry performance benchmarks on implementing GPUs in high-performance signal processing applications have shown that GPUs can obtain at least 20 times performance improvement over previous generation processors. For batch processing problems that require extreme throughput, GPUs provide a compelling solution.

The single-thread SAL programming API does not change when using specialized GPU hardware. GPUs come from many differing vendors and often require extensions to traditional programming languages to program them, such as OpenCL and CUDA. SAL preserves the traditional programming model and provides an abstraction layer that allows application developers to benefit from a high-performance library that was developed with the aid of GPU-specific languages. As a result, the developer is shielded from the requirement of rewriting code to use GPU language extensions. This enables the same SAL program to run on a desktop using CPUs and automatically "just work" when run on a system equipped with the latest GPU processors.

VSIPL++

VSIPL++ can be added to selective Mercury platforms through software alliance arrangements with Mentor Graphics. (Mentor Graphics recently acquired CodeSourcery, original developers of the Sourcery VSIPL++ product.) Sourcery VSIPL++ is a C++ library for developing high-performance signal- and image-processing applications. It utilizes sophisticated C++ techniques taking advantage of optimized libraries to achieve performance comparable to that of hand-optimized code.

Sourcery VSIPL++ is a complete implementation of the VSIPL++ API developed by the High Performance Embedded Computing Software Initiative and the VSIPL Forum to promote the goals of portability, productivity and performance simultaneously. To learn more, check out the Sourcery VSIPL++ website at http://go.mentor.com/vsiplxx.

Algorithm Optimization Services

MultiCore Plus 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.

MultiCore Plus MathPack Contents and Packaging

MultiCore MathPack includes:

Single-core libraries:

- SAL for Linux® & VxWorks with optimizations for single-core processors
- CSAL "C" Source code for all SAL APIs
- VSIPL-Lite Plus over SAL (where optimized) or CSAL

Multicore libraries:

- MC SAL with select optimizations for multicore (threaded)
- MC VSIPL-Lite Plus over MC SAL

SAL System Requirements

Processors supported

MPC74xx, MPC864xD, IBM PowerPC 970, Cell BE, IA32 and Intel64, NVIDIA GPUs

Operating systems

Linux, VxWorks, Windows, MCOE™

MC SAL System Requirements

Processors supported

MPC864xD, IA32 and Intel64

Operating system

Linux®, VxWorks®

* Environmental specifications are as installed on Mercury 6U host/carrier modules.

** As altitude increases, air density decreases and, therefore, the cooling effect of a particular number of CFM decreases. Different limits can be achieved by trading among temperature, altitude, frequency and airflow.

2862.01E-1218-DS-MCMP

MERCURY COMPUTER SYSTEMS™

INNOVATION THAT MATTERS™

Corporate Headquarters
50 Minuteman Road • Andover, MA 01810 USA
(978) 967-1401 • (866) 627-6951 • Fax (978) 256-3599

Europe Mercury Systems, LTD
Unit 1 - Easter Park, Benyon Road, Silchester, Reading
RG7 2PQ United Kingdom
+ 44 0 1189 702050 • Fax + 44 0 1189 702321