ISSPL-ALT Industry Standard Signal Processing LIbrary

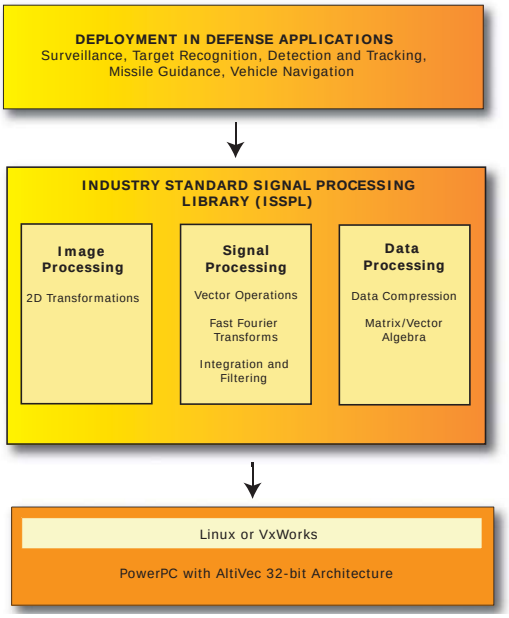
PowerPC with AltiVec Architecture

The CSPI ISSPL-ALT (Industry Standard Signal Processing Library) provides an extensive set of subroutines enabling software developers to easily create complex 32-bit signal processing algorithms for sonar, radar, and SIGINT applications.

The ISSPL-ALT is a highly optimized library of over 250 functions for signal and image processing applications for use on the 2000 and 3000 SERIES PowerPC with AltiVec based products. Compute intensive functions are hand coded to speed up execution time. The remainder use C compiled code optimized for speed.

The ISSPL-ALT reduces programming complexity by providing a single calling function for complex mathematical and signal processing routines. Both development and execution times are significantly improved. The ISSPL-ALT provides a full complement of routines to take advantage of both the AltiVec and Floating-Point units of the PowerPC. All routines are callable from C and C++ compiled programs to achieve rapid and efficient program development. The C emulations of all standard ISSPL-ALT functions are available for initial development work.

Backward compatibility with earlier versions of ISSPL protects the user's software investment.



CSPi's Industry Standard Signal Processing Library provides functions optimized for signal, image and data processing.

KEY FEATURES

- Signal Processing Library Optimized for the PowerPC with AltiVec architecture
- Over 250 functions for signal and image processing
- Compute intensive functions hand-coded to speed up execution time
- All routines are callable from C and C++ compiled programs

CSPi
High Performance Products

Industry Standard Signal Processing Library

Fast Fourier Transform Subroutines (16)

The ISSPL has fourteen functions that perform Fast Fourier Transforms on vectors, including operations in place or out of place, forward or inverse, and using complex or real data.

Basic Vector Arithmetic Subroutines (39)

From simple vector arithmetic (add, subtract, multiply, divide) to more complex operations (generate cosines, fill, interpolate) the library includes nearly forty functions for performing basic vector arithmetic and combinations thereof, such as Vector Multiply, Multiply and Subtract (vmmsb).

Non Linear Vector Arithmetic Intrinsics (48)

The library provides a range of intrinsic functions from vector logical greater than, less than, equal, not equal; to vector square root; vector sine, cosine, tangent, arcsine, arccosine, arctangent; vector truncation; vector euclidean distance; vector pythagorean; vector exponential and vector logarithm in base2 and base10; vector minimum, maximum, threshold, limit and ceiling.

Complex Vector Subroutines (47)

Among the complex vector functions in the ISSPL are: move, fill, conjugate, absolute value, negate, phase angle, magnitude, reciprocal, square root, scalar, add, multiply, divide, accumulating cross spectrum, polar to rectangular conversion and vice-versa.

Integer Vector Arithmetic (11)

Integer Vector Arithmetic functions include: negate, add, subtract, multiply, left/right shift, arithmetic right shift, inclusive/exclusive or, and xnor.

Matrix Functions (12)

The twelve matrix functions include subroutines for decomposition, row pivoting, transposition, and Weiner-Levinson Equation Solution.

Vector Reduction Subroutines (26)

Real/complex dot product, complex conjugate dot product, mean and sum of vector elements, distance between vectors squared, min/max element in a vector, min/max magnitude in a vector, and a number of offsets of first element in vector (greater than, less than, less than or equal to scalar) are among the algorithms available for vector reduction.

Format Conversion Subroutines (19)

Format Conversion subroutines for extracting real/imaginary from a complex vector, converting single to double or vice versa, performing fix and store, scale and fix operations are in the ISSPL.

Integration and Filtering Subroutines

Integration and Filtering subroutines provide algorithms for auto correlation and cross correlation across both time and frequency domains.

Miscellaneous Real Vector Subroutines (23)

These routines include: vector polynomial evaluation, random number generation, histogram, vector index, scatter and gather, vector linear interpolate, vector quadratic interpolate, table look up, Schafer's Phase Unwrapping, Hilbert transform, Envelope Detector (out of place), Vector Compress, Vector Move Greater Than Flag, Vector Logical Merge, Vector Merge Positive Test and Vector Merge Tapered, Blackman Window of Vector, Hamming Window of Vector, and Hanning Window of Vector.

Radix 3 Fast Fourier Transform Subroutines (15)

There are fifteen functions for Radix 3 Fast Fourier Transforms on vectors, including operations in place or out of place, forward or inverse, and using complex or real data.

2D Fast Fourier Transform Subroutines (12)

The 2D FFT subroutines perform Complex and Real FFT's, forward and inverse, with Row Compact and Column Compact.

Arrays of 1D Fast Fourier Transform Subroutines (4)

Array manipulation subroutines includes: Series of Complex FFT Forward and Inverse, as well as Series of Real FFT Forward and Inverse.

ISSPL PERFORMANCE HIGHLIGHTS (timings using a single thread, ISSPL-ALT Release 4.1.0.0, execution time in microseconds, 1024 pts.)		
	2000 SERIES **2942 Blade (G4 core)** **MPC7457 @ 1 GHz**	**3000 SERIES** **3120D Blade (e600 core)** **MPC8641D @ 1.33 GHz**
FAST FOURIER TRANSFORM SUBROUTINES		
Complex FFT, Forward, Split Format (in place)	10.46	7.42
Complex FFT, Forward, Split Format (out of place)	10.7	7.81
Complex FFT, Inverse, Split Format (in place)	10.46	7.40
Complex FFT, Inverse, Split Format (out of place)	10.71	7.91
Real FFT, Forward, Radix 2, Split Format (out of place)	6.4	4.84
Real FFT, Inverse, Radix 2, Split Format (out of place)	6.72	5.08
BASIC VECTOR ARITHMETIC SUBROUTINES		
Vector Ramp: y(k)=a+k*b	0.61	0.50
Vector Scalar Multiply Twice & Scalar Add: y=u*a+v*b+c	1.27	0.9975
Vector Scalar Multiply Twice & Scalar Add: y=u*a+v*b+c	1.17	0.92
NON LINEAR VECTOR ARITHMETIC INTRINSICS		
Vector Euclidean Distance (two-way): y=sqrt(u*u+v*v)	2.40	1.85
Vector Cosines, Fractions: y=cos(2*pi*u)	4.53	3.66
Vector Tangent, Fractions: y=tan(2*pi*u)	12	8.66
Vector Exponential: y=e^u	4.51	3.41
Vector Base 10 Logarithim: y=log10(u)	5.07	3.79
Vector Maximum: y=max(u,v)	1.01	0.82
Vector Clip: y=min(max(u,a),a)	0.80	0.64
Logical Vector Greater Than: if u>v then y=1.0, else y=0.0	1.20	0.95
COMPLEX VECTOR SUBROUTINES		
Complex Vector Magnitude Squared, Split Format: y=cu*conj(cu)	1.13	0.80
Complex Vector Scalar Multiply and Add, Split Format: cy=cu*ca+cv	2.55	1.82
Complex Real Vector Multiply, Split Format: cy=cu*v	2.22	1.55
Complex Vector Power and Add, Split Format: y=cu*conj(cu)*a+v*b	1.55	1.04
Complex Vector Exponential: cy=cos(u)+i*sin(u)	13.25	9.67
INTEGER VECTOR ARITHMETIC		
Vector Integer Add: iy=iu+iv	1.12	0.82
Vector Integer Multiply: iy-iu*iv	5.20	3.94
Vector Integer Inclusive or : iy=or(iu,iv); bitwise or of vectors	1.12	0.81
VECTOR REDUCTION SUBROUTINES		
Real Dot Product: a=sum(u*v)	0.62	0.51
Complex Conjugate Dot Product, Split: ca=sum(cv*conj(cu))	1.54	1.19
Sum of Vector Elements: a=sum(u)	0.4	0.36
Max Element in a Vector: a=max(u), icx=index offset at max Value	1.01	0.80
FORMAT CONVERSION SUBROUTINES		
Vector Float Integer *4: y=fl oat(iu)	0.72	0.58
Vector Float Integer*2: y=fl oat(i2u	0.70	0.55
Vector Float Byte: y=fl oat(ibu), (-128<=y<=127)	0.64	0.57
Vector Fix and Store Nearest Integer*4: iy=nint(u)	1.01	0.73
Vector Fix and Store Nearest Byte: iby=nint(u)	0.80	0.63
INTEGRATION AND FILTERING SUBROUTINES		
Difference Equation, 2 poles, 2 zeros: y(k)=v(0)*u(k)+v(1)*u(k-1)+v(2)*u(k-2)-v(3)*y(k-1)-v(4)*y(k-2)	9.84	7.41
Convd: nv=32 nd=1 / nd=2 / nd=4	18.45 / 16.8 / 19.13	13.21 / 12.64 / 13.22
Vector Linear Averaging: y=y*(a/(a+1.0))+u/(a+1.0)	1.35	1.05
MISCELLANEOUS REAL VECTOR SUBROUTINES		
Vector Gather: y=t(iv)	4.93	3.93
Vector Scatter: y(iv(k))=u(k)	5.2	3.94

2000 SERIES / 3000 SERIES Software Ordering Information

GENERAL ORDERING INFORMATION	
Software Development Kits	To help you get started with your development project CSPI offers a Software Development Kit (SDK) which includes a Single User Development License, the optimized library, example/demo code, and manuals. SDK's are available to support a Linux or VxWorks operating system environment.
Licensing	Should you wish to port your application to a non-CSPI platform utilizing a QorIQ T Series or PowerPC with Alti-Vec architecture CSPI offers a runtime/production license agreement on a case-by-case basis.
PART NUMBERS	
110282-02	2000 SERIES SDK for ISSPL-ALT VxWorks - 1. Single User Software Development License. ISSPL-ALT Library optimized for the PowerPC 7457 Architecture. Installation for VxWorks.
110283-02	2000 SERIES SDK for ISSPL-ALT VxWorks - 5. Five User Software Development License. Optimized for PowerPC 7457 Architecture. Installation for VxWorks.
110284-02	2000 SERIES SDK for ISSPL-ALT VxWorks Plus. Adds one additional user to 110282 or 110283
190321	2000 SERIES Linux SDK for ISSPL-ALT-1. Single User Software Development License. ISSPL-ALT Library optimized for the PowerPC 7457 Architecture. Installation for Linux.
190322	2000 SERIES Linux SDK for ISSPL-ALT-5. Five User Software Development License. ISSPL-ALT Library optimized for the PowerPC 7457 Architecture. Installation for Linux.
190323	2000 SERIES Linux SDK for ISSPL-ALT Plus. Adds one additional user to 190321 or 190322
110421	ISSPL-ALT Runtime License. Runtime License for one 74xx PowerPC processor. Usage must conform to CSPI's licensing agreement.
110509-01	3000 SERIES SDK for ISSPL-Linux. Single User Software Development License. Optimized ISSPL library for Freescale processors: PowerPC 8641D Architecture. Installation for Linux.
110502-01	3000 SERIES SDK for ISSPL-VxWorks. Single User Software Development License. Optimized ISSPL library for Freescale processors: PowerPC 8641D Architecture. Installation for VxWorks.
110510-01	3000 SERIES SDK for ISSPL-Additional User License. Adds one additional user to 3000 SERIES ISSPL SDK's above (110509-01 OR 110502-01) for either Linux or VxWorks.
110528-01	ISSPL-8641 Production License. Production License for use on non-CSPI MPC8641/8641D processors. One license is required for each CORE, and usage must conform to CSPI's license agreement.

The information contained herein is subject to change without prior notice. For the latest detailed information contact your representative at +1 (978) 663-7598 or visit www.cspi.com.

CSPI
High Performance Products