

VSIPL Core Lite Profile

Version 1.0



<http://www.vsipl.org>

As Approved by the VSIPL Forum

March 14, 2000

©1999-2000 Georgia Tech Research Corporation, all rights reserved.

A non-exclusive, non-royalty bearing license is hereby granted to all persons to copy, modify, distribute and produce derivative works for any purpose, provided that this copyright notice and following disclaimer appear on all copies: THIS LICENSE INCLUDES NO WARRANTIES, EXPRESSED OR IMPLIED, WHETHER ORAL OR WRITTEN, WITH RESPECT TO THE SOFTWARE OR OTHER MATERIAL INCLUDING, BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE, OR ARISING FROM A COURSE OF PERFORMANCE OR DEALING, OR FROM USAGE OR TRADE, OR OF NON-INFRINGEMENT OF ANY PATENTS OF THIRD PARTIES. THE INFORMATION IN THIS DOCUMENT SHOULD NOT BE CONSTRUED AS A COMMITMENT OF DEVELOPMENT BY ANY OF THE ABOVE PARTIES.

This material is based in part upon work supported by Ft. Huachuca/DARPA under contract No. DABT63-96-C-0060. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of Ft. Huachuca/DARPA.

The US Government has a license under these copyrights, and this material may be reproduced by or for the US Government.

VSIPL Core Lite Profile Requirements	1
Introduction.....	1
Basic Data Types.....	1
Function Profile for Core Lite.....	1
Support Functions	1
Scalar functions.....	3
Vector Elementwise Functions	3
Signal Processing Functions	5

VSIPL Core Lite Profile Requirements

Introduction

This document contains minimum requirements for a VSIPL compliant library meeting a profile called VSIPL Core Lite, or just Core Lite. This is a companion requirements document to the VSIPL specification. All functions and data types required in the Core Lite profile must meet all requirements for functions and data types as defined in the VSIPL specification.

Basic Data Types

This profile is a list of the (minimum) required functions to meet a VSIPL Core Lite profile. Implementation of the profile requires the implementor to support VSIPL data types needed to implement the functions.

To meet the minimum requirement for a VSIPL Core Lite profile the implementor need only support one float type, and one integer type.

The following scalar data types are listed as a convenience. In order to implement the functions of the Core Lite profile in accordance with the VSIPL specification these scalars must be defined. Note that *_f* and *_i* denote some implementation dependent float type and some implementation dependent integer type respectively.

Data Type	Comments
<code>vsip_scalar_vi</code>	Only scalar index for vectors is needed.
<code>vsip_scalar_bl</code>	Only scalar boolean is needed.
<code>vsip_scalar_f</code>	Only a single float type is needed.
<code>vsip_cscalar_f</code>	Only a single float type of precision matching <code>vsip_scalar_f</code> is needed.
<code>vsip_scalar_i</code>	Only a single signed, or unsigned integer type is needed.

Function Profile for Core Lite

Support Functions

All profiles must include `vsip_init` and `vsip_finalize` support functions. In addition, Core Lite must include the following VSIPL support functions.

Block Support

The following set of functions are included for block support in Core Lite.

<i>vsip_blockcreate_f</i>	<i>vsip_blockcreate_i</i>	<i>vsip_cblockcreate_f</i>
<i>vsip_blockbind_f</i>	<i>vsip_blockbind_i</i>	<i>vsip_cblockbind_f</i>
<i>vsip_blockfind_f</i>	<i>vsip_blockfind_i</i>	<i>vsip_cblockfind_f</i>
<i>vsip_blockdestroy_f</i>	<i>vsip_blockdestroy_i</i>	<i>vsip_cblockdestroy_f</i>
<i>vsip_blockadmit_f</i>	<i>vsip_blockadmit_i</i>	<i>vsip_cblockadmit_f</i>
<i>vsip_blockrelease_f</i>	<i>vsip_blockrelease_i</i>	<i>vsip_cblockrelease_f</i>
<i>vsip_blockrebind_f</i>	<i>vsip_blockrebind_i</i>	<i>vsip_cblockrebind_f</i>
<i>vsip_cstorage</i>		

Total functions are 22 for block support.

View Support

The following set of functions are included for view support in Core Lite.

<i>vsip_vbind_f</i>	<i>vsip_vbind_i</i>	<i>vsip_cvbind_f</i>
<i>vsip_vcreate_f</i>		<i>vsip_cvcreate_f</i>
<i>vsip_vdestroy_f</i>	<i>vsip_vdestroy_i</i>	<i>vsip_cvdestroy_f</i>
<i>vsip_valldestroy_f</i>		<i>vsip_cvalldestroy_f</i>
<i>vsip_vcloneview_f</i>		<i>vsip_cvcloneview_f</i>
		<i>vsip_vrealview_f</i>
		<i>vsip_vimagview_f</i>
<i>vsip_vsubview_f</i>		<i>vsip_cvsubview_f</i>
<i>vsip_vgetattrib_f</i>	<i>vsip_vgetattrib_i</i>	<i>vsip_cvgetattrib_f</i>
<i>vsip_vgetblock_f</i>		<i>vsip_cvgetblock_f</i>
<i>vsip_vputattrib_f</i>	<i>vsip_vputattrib_i</i>	<i>vsip_cvputattrib_f</i>
<i>vsip_vputoffset_f</i>		<i>vsip_cvputoffset_f</i>
<i>vsip_vputstride_f</i>		<i>vsip_cvputstride_f</i>
<i>vsip_vputlength_f</i>		<i>vsip_cvputlength_f</i>
<i>vsip_vget_f</i>		<i>vsip_cvget_f</i>
<i>vsip_vput_f</i>		<i>vsip_cvput_f</i>

Total functions are 34 for view support

Copy Support

The following copy are included as part of the Core Lite Profile.

<code>vsip_vcopy_f_f</code>	<code>vsip_vcopy_f_i</code>	<code>vsip_vcopy_i_f</code>	<code>vsip_cvcopy_f_f</code>
-----------------------------	-----------------------------	-----------------------------	------------------------------

Total functions are 4 for copy support

Scalar functions

The following functions are included for scalar support.

Return by Value	Argument Return
<code>vsip_cmplx_f</code>	<code>vsip_CMPLX_f</code>
<code>vsip_imag_f</code>	
<code>vsip_real_f</code>	

Total functions are 4 for scalar support.

Vector Elementwise Functions

The following vector and elementwise functions are included

Real Arguments	Complex Arguments	Real and Complex Arguments
<code>vsip_vatan f</code>		
<code>vsip_vatan2 f</code>		
<code>vsip_vcos f</code>		
<code>vsip_vexp f</code>		
<code>vsip_vlog f</code>		
<code>vsip_vlog10 f</code>		
<code>vsip_vsin f</code>		
<code>vsip vsqrt f</code>		
	<code>vsip cvconj f</code>	
<code>vsip vmag f</code>	<code>vsip cvmag f</code>	
	<code>vsip vcmagsq f</code>	
<code>vsip vneg f</code>	<code>vsip cvneg f</code>	
<code>vsip vrecip f</code>		
<code>vsip vsq f</code>		

Real Arguments	Complex Arguments	Real and Complex Arguments
<code>vsip vsumval <i>f</i></code>		
<code>vsip vsumsqval <i>f</i></code>		
<code>vsip vadd <i>f</i></code>	<code>vsip cvadd <i>f</i></code>	
<code>vsip svadd <i>f</i></code>		
	<code>vsip cvjdot <i>f</i></code>	
<code>vsip vdiv <i>f</i></code>		
<code>vsip svdiv <i>f</i></code>		
<code>vsip vdot <i>f</i></code>	<code>vsip cvdot <i>f</i></code>	
	<code>vsip cvjmul <i>f</i></code>	
<code>vsip vmul <i>f</i></code>	<code>vsip cvmul <i>f</i></code>	<code>vsip rcvmul <i>f</i></code>
<code>vsip svmul <i>f</i></code>	<code>vsip csvmul <i>f</i></code>	<code>vsip rscvmul <i>f</i></code>
<code>vsip vsub <i>f</i></code>	<code>vsip cvsul <i>f</i></code>	
<code>vsip vmax <i>f</i></code>		
<code>vsip vmaxval <i>f</i></code>		
<code>vsip vmin <i>f</i></code>		
<code>vsip vminval <i>f</i></code>		
<code>vsip vfill <i>f</i></code>		
<code>vsip vramp <i>f</i></code>		
		<code>vsip vcplx <i>f</i></code>
		<code>vsip vimag <i>f</i></code>
		<code>vsip vreal <i>f</i></code>

Random Elementwise Support

Core Lite requires support for uniform random elementwise vectors. The following functions are required.

<code>vsip_randcreate</code>
<code>vsip_vrandu_ <i>f</i></code>
<code>vsip_randdestroy</code>

A total of 47 functions in vector and elementwise

Signal Processing Functions

Core Lite requires support for the following signal processing functions.

Fast Fourier Transform

<i>vsip_ccffftop_f</i>	<i>vsip_ccffftop_create_f</i>	
<i>vsip_rcffftop_f</i>	<i>vsip_rcffftop_create_f</i>	<i>vsip_fft_destroy_f</i>
<i>vsip_crffftop_f</i>	<i>vsip_crffftop_create_f</i>	

FIR filter functions

<i>vsip_fir_create_f</i>	<i>vsip_cfir_create_f</i>
<i>vsip_firflt_f</i>	<i>vsip_cfirflt_f</i>
<i>vsip_fir_destroy_f</i>	<i>vsip_cfir_destroy_f</i>

Histogram Function

<i>vsip_vhisto_f</i>

A total of 14 functions in Signal Processing

A total of 127 required functions.